



## D3.3

# Intermediate Outcomes of New Learning Paradigms Research

<b>Project Title</b>	AI4Media - A European Excellence Centre for Media, Society and Democracy
<b>Contract No.</b>	951911
<b>Instrument</b>	Research and Innovation Action
<b>Thematic Priority</b>	H2020-EU.2.1.1. - INDUSTRIAL LEADERSHIP - Leadership in enabling and industrial technologies - Information and Communication Technologies (ICT) / ICT-48-2020 - Towards a vibrant European network of AI excellence centres
<b>Start of Project</b>	1 September 2020
<b>Duration</b>	48 months



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951911

[info@ai4media.eu](mailto:info@ai4media.eu)

[www.ai4media.eu](http://www.ai4media.eu)



<b>Deliverable title</b>	Intermediate Outcomes of New Learning Paradigms Research
<b>Deliverable number</b>	D3.3
<b>Deliverable version</b>	1.0
<b>Previous version(s)</b>	-
<b>Contractual date of delivery</b>	August 31, 2023
<b>Actual date of delivery</b>	September 12, 2023
<b>Deliverable filename</b>	AI4Media_D3.3-final.pdf
<b>Nature of deliverable</b>	Report
<b>Dissemination level</b>	Public
<b>Number of pages</b>	197
<b>Work Package</b>	WP3
<b>Task(s)</b>	T3.1, T3.2, T3.3, T3.6, T3.7
<b>Partner responsible</b>	QMUL
<b>Author(s)</b>	Ioannis Patras, Christos Tzelepis (QMUL), Hannes Fassold (JR), Nicu Sebe, Cigdem Beyan, Karim Sinan Yildirim, Marco Formentini (UNITN), Niccolò Biondi, Federico Pernici (UNIFI), Antonios Liapis, Marvin Zammit, Matthew Barthet (UM), Adrian Popescu (CEA), Fabrizio Sebastiani (CNR), Adrian Tormos, Dario Garcia-Gasulla (BSC), Ioanna Valsamara (AUTH)
<b>Editor</b>	Christos Tzelepis (QMUL), Ioannis Patras (QMUL)
<b>Officer</b>	Evangelia Markidou

<b>Abstract</b>	This document presents the intermediate outcomes of the research on new learning paradigms in WP3, reporting the advances of the partners in tasks T3.1, T3.2, T3.3, T3.6, and T3.7 in the period between M13 and M36. More specifically, it gives an update on D3.1 (T3.1, T3.3 and T3.7) and reports for the first time on tasks that started on M13 (T3.2 and T3.6). For each task, we present the contributions including the relevant publications and links to software. Finally, we discuss the plans for ongoing and future research.
<b>Keywords</b>	Artificial Intelligence, media, machine learning, deep learning, lifelong learning, online learning, manifold learning, disentangled feature representation, transfer learning, domain adaptation, deep quality diversity, learning to count

## Copyright

© Copyright 2023 AI4Media Consortium

This document may not be copied, reproduced, or modified in whole or in part for any purpose without written permission from the AI4Media Consortium. In addition to such written permission to copy, reproduce, or modify this document in whole or part, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced. All rights reserved.





## Contributors

NAME	ORGANIZATION
Ioannis Patras	QMUL
Christos Tzelepis	QMUL
Hannes Fassold	JR
Nicu Sebe	UNITN
Cigdem Beyan	UNITN
Karim Sinan Yildirim	UNITN
Marco Formentini	UNITN
Niccolò Biondi	UNIFI
Federico Pernici	UNIFI
Antonios Liapis	UM
Marvin Zammit	UM
Matthew Barthet	UM
Adrian Popescu	CEA
Fabrizio Sebastiani	CNR
Adrian Tormos	BSC
Dario Garcia-Gasulla	BSC
Ioanna Valsamara	AUTH

## Peer Reviews

NAME	ORGANIZATION
Vasileios Mezaris	CERTH
Giuseppe Amato	CNR

## Revision History





Version	Date	Reviewer	Modifications
0.1	02/06/2023	Ioannis Patras	First draft sent to partners for contributions
0.2	28/06/2023	Ioannis Patras, Christos Tzelepis	Updated version including inputs from CNR in T3.7
0.3	30/06/2023	Ioannis Patras, Christos Tzelepis	Updated version including inputs from CEA in T3.1 and T3.3
0.4	3/07/2023	Ioannis Patras, Christos Tzelepis	Deliverable sent to reviewers for internal review
0.5	6/08/2023	Ioannis Patras, Christos Tzelepis	Updated version based on internal review comments.
0.6	29/08/2023	Ioannis Patras, Christos Tzelepis	Updated version regarding Use Cases.
1.0	12/09/2023	Ioannis Patras, Christos Tzelepis	Final version.

The information and views set out in this report are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf.





## Table of Abbreviations and Acronyms

Abbreviation	Meaning
ACC	Adjusted Classify and Count
AE	AutoEncoder
AI	Artificial Intelligence
APP	Artificial Prevalence Protocol
BR	Binary Relevance
CAM	Class Activation Maps
CBIR	Content-Based Image Retrieval
CC	Classify and Count
CIL	Class-Incremental Learning
CL	Curriculum Learning
class-iNCD	class-incremental NCD
CNN	Convolutional Neural Network
CoReS	Compatible Representations via Stationarity
CPSS	Cross-Patch Style Swap
CSD	Contrastive Supervised Distillation
DA	Domain Adaptation
DAS	Design Alternatives Sampling
DCNN	Deep Convolutional Neural Network
deepQD	deep-learning-based QD
DeLeNoX	Deep Learning Novelty eXplorer
DNN	Deep Neural Network
EC	Evolutionary Computation
EFCIL	Exemplar-Free Class-Incremental Learning
FE	Feature Extraction
FT	Fine-Tuning
GAN	Generative Adversarial Network
GANs	Generative Adversarial Networks
GCD	Generalized Category Discovery
HED	Holistically-Nested Edge Detection
IBU	Iterative Bayesian Unfolding
IEC	Interactive Evolutionary Computation
KL	Kullback-Leibler
KSA	Knowledge Self-Assessment
LDA	Latent Dirichlet Allocation
LLE	Local Linear Embedding
LQ	Learning to Quantify
MELiTA	MAP-Elites with Transverse Assessment





Abbreviation	Meaning
NCD	Novel Class Discovery
NEAT	NeuroEvolution of Augmenting Topologies
NPP	Natural Prevalence Protocol
NSGA-II	Non-dominated Sorting Genetic Algorithm II
NSLC	Novelty Search with Local Competition
OOD	Out-of-Distribution
OODD	Out Of Distribution Detection
OQ	Ordinal Quantification
PACC	Probabilistic Adjusted Classify and Count
PCC	Probabilistic Classify and Count
PCG	Procedural Content Generation
PL	Preference Learning
PT	Photometric Transformation
QD	Quality-Diversity
QoI	Quality of Inference
RUN	Regularized Unfolding
S2D-Dec	Sparse2Dense mesh Decoder
SF-OCDA	Source-Free Open Compound Domain Adaptation
SFDA	Source-Free Domain Adaptation
SLD	Saerens-Latinne-Decaestecker method
SOTA	State of the Art
SRVF	Square Root Velocity Function
SSL	Self-supervised Learning
TTDA-Seg	Test-Time Domain Adaptation in Semantic Segmentation
UB	Upper Bound
UC-ME	User Controller MAP-Elites
UCB	Upper Confidence Bound
UDA	Unsupervised Domain Adaptation
USC	User Selection Criterion
VAE	Variational AutoEncoder





## Contents

<b>1</b>	<b>Executive Summary</b>	<b>12</b>
<b>2</b>	<b>Introduction</b>	<b>15</b>
<b>3</b>	<b>Concise descriptions of the presented works</b>	<b>17</b>
3.1	Lifelong and on-line learning (Task 3.1)	17
3.1.1	Introduction	17
3.1.2	Overview	17
3.2	Manifold learning and disentangled feature representation (Task 3.2)	18
3.2.1	Introduction	18
3.2.2	Overview	18
3.3	Transfer learning (Task 3.3)	20
3.3.1	Introduction	20
3.3.2	Overview	20
3.4	Deep quality diversity (Task 3.6)	21
3.4.1	Introduction	21
3.4.2	Overview	22
3.5	Learning to count (Task 3.7)	23
3.5.1	Introduction	23
3.5.2	Overview	24
<b>4</b>	<b>Lifelong and on-line learning (Task 3.1) – detailed description</b>	<b>26</b>
4.1	FeTrIL: Feature Translation for Exemplar-Free Class-Incremental Learning	26
4.1.1	Introduction and methodology	26
4.1.2	Experimental results	26
4.1.3	Conclusion	28
4.1.4	Relevant publications	28
4.1.5	Relevant software/datasets/other outcomes	29
4.1.6	Relevance to AI4media use cases and media industry applications	29
4.2	AdvisIL - A Class-Incremental Learning Advisor	29
4.2.1	Introduction and methodology	29
4.2.2	Experimental results	30
4.2.3	Conclusion	32
4.2.4	Relevant publications	32
4.2.5	Relevant software/datasets/other outcomes	32
4.2.6	Relevance to AI4media use cases and media industry applications	32
4.3	Towards Human Society-inspired Decentralized DNN Inference	32
4.3.1	Introduction and methodology	33
4.3.2	Experimental results	33
4.3.3	Relevant publications	34
4.3.4	Relevance to AI4media use cases and media industry applications	35
4.4	Quantifying the knowledge in Deep Neural Networks: an overview	35
4.4.1	Introduction and methodology	35
4.4.2	Experimental Results	36
4.4.3	Relevant publications	38
4.4.4	Relevance to AI4media use cases and media industry applications	38





4.5	Knowledge Distillation-driven Communication Framework for Neural Networks: Enabling Efficient Student-Teacher Interactions . . . . .	38
4.5.1	Introduction and methodology . . . . .	38
4.5.2	Experimental results . . . . .	40
4.5.3	Relevant publications . . . . .	41
4.5.4	Relevance to AI4media use cases and media industry applications . . . . .	41
4.6	Curriculum Learning: A Survey . . . . .	42
4.6.1	Contributions . . . . .	44
4.6.2	Relevant publications . . . . .	44
4.7	Class-incremental Novel Class Discovery . . . . .	44
4.7.1	Introduction and methodology . . . . .	44
4.7.2	Experimental results . . . . .	46
4.7.3	Conclusions . . . . .	48
4.7.4	Relevant publications . . . . .	48
4.7.5	Relevant software/datasets/other outcomes . . . . .	48
4.7.6	Relevance to AI4media use cases and media industry applications . . . . .	48
4.8	CoReS: Learning Compatible Representations via Stationarity . . . . .	48
4.8.1	Introduction and methodology . . . . .	49
4.8.2	Experimental Results . . . . .	51
4.8.3	Conclusions . . . . .	52
4.8.4	Relevant publications . . . . .	52
4.8.5	Relevant software/datasets/other outcomes . . . . .	52
4.8.6	Relevance to AI4media use cases and media industry applications . . . . .	53
4.9	CL <sup>2</sup> R: Compatible Lifelong Learning Representations . . . . .	53
4.9.1	Introduction and methodology . . . . .	53
4.9.2	Experimental Results . . . . .	55
4.9.3	Conclusions . . . . .	56
4.9.4	Relevant publications . . . . .	57
4.9.5	Relevant software/datasets/other outcomes . . . . .	57
4.9.6	Relevance to AI4media use cases and media industry applications . . . . .	57
4.10	Contrastive Supervised Distillation for Continual Representation Learning . . . . .	58
4.10.1	Introduction and methodology . . . . .	58
4.10.2	Experimental Results . . . . .	59
4.10.3	Conclusions . . . . .	60
4.10.4	Relevant publications . . . . .	60
4.10.5	Relevant software/datasets/other outcomes . . . . .	60
4.10.6	Relevance to AI4media use cases and media industry applications . . . . .	60
<b>5</b>	<b>Manifold learning and disentangled feature representation (Task 3.2) – detailed description</b>	<b>62</b>
5.1	Finding non-linear RBF paths in GAN latent space . . . . .	62
5.1.1	Introduction and methodology . . . . .	62
5.1.2	Experimental results . . . . .	63
5.1.3	Relevant publications . . . . .	65
5.1.4	Relevant software/datasets/other outcomes . . . . .	65
5.1.5	Relevance to AI4media use cases and media industry applications . . . . .	66
5.2	Unsupervised learning of parts and appearances in the feature maps of GANs . . . . .	66
5.2.1	Introduction and methodology . . . . .	66
5.2.2	Experimental results . . . . .	68







5.2.3	Relevant publications	69
5.2.4	Relevant software/datasets/other outcomes	69
5.2.5	Relevance to AI4media use cases and media industry applications	70
5.3	Dataset Anonymization with Generative Models	70
5.3.1	Introduction and methodology	70
5.3.2	Experimental results	72
5.3.3	Relevant publications	73
5.3.4	Relevant software/datasets/other outcomes	73
5.3.5	Relevance to AI4media use cases and media industry applications	74
5.4	A survey of manifold learning and its applications for multimedia	74
5.4.1	Introduction	74
5.4.2	Similarity search & retrieval	75
5.4.3	Image classification & object detection	76
5.4.4	Image synthesis & enhancement	76
5.4.5	Video analysis	77
5.4.6	3D data processing	77
5.4.7	Nonlinear dimension reduction	78
5.4.8	Relevant publications	79
5.5	Manifold mixing soups for better out-of-distribution performance	79
5.5.1	Introduction and methodology	79
5.5.2	Experiments and Evaluation	81
5.5.3	Conclusion	83
5.5.4	Relevant publications	83
5.5.5	Relevant software/datasets/other outcomes	84
5.5.6	Relevance to AI4media use cases and media industry applications	84
5.6	Sparse to Dense Dynamic 3D Facial Expression Generation	84
5.6.1	Introduction and methodology	84
5.6.2	Experimental results	86
5.6.3	Conclusions	88
5.6.4	Relevant publications	89
5.6.5	Relevant software/datasets/other outcomes	89
5.6.6	Relevance to AI4media use cases and Media Industry Applications	89
5.7	Conditioned Image Retrieval for Fashion using Contrastive Learning and CLIP-based Features	89
5.7.1	Introduction	90
5.7.2	Previous work	90
5.7.3	The proposed method	91
5.7.4	Experimental results	92
5.7.5	Conclusions	94
5.7.6	Relevant publications	95
5.7.7	Relevance to AI4media use cases and media industry applications	95
5.8	Hyperbolic Vision Transformers	95
5.8.1	Introduction and methodology	95
5.8.2	Experimental results	96
5.8.3	Conclusions	98
5.8.4	Relevant publications	98
5.8.5	Relevant software/datasets/other outcomes	99
5.8.6	Relevance to AI4media use cases and media industry applications	99





<b>6</b>	<b>Transfer learning (Task 3.3) – detailed description</b>	<b>100</b>
6.1	When & How: Methodological study on transfer learning	100
6.1.1	Methodology	100
6.1.2	Results	101
6.1.3	Discussion	103
6.1.4	Relevant publications	103
6.1.5	Relevant software/datasets/other outcomes	103
6.2	Source-Free Open Compound Domain Adaptation in Semantic Segmentation	103
6.2.1	Introduction and methodology	103
6.2.2	Experimental results	105
6.2.3	Conclusions	108
6.2.4	Relevant publications	108
6.2.5	Relevant software/datasets/other outcomes	108
6.2.6	Relevance to AI4media use cases and media industry applications	108
6.3	solo-learn: A Library of Self-supervised Methods for Visual Representation Learning	109
6.3.1	The solo-learn Library: An Overview	110
6.3.2	Self-supervised Learning Methods	110
6.3.3	Architecture	110
6.3.4	Comparison to Related Libraries	111
6.3.5	Experiments	111
6.3.6	Conclusion	112
6.3.7	Relevant publications	112
6.3.8	Relevant software/datasets/other outcomes	112
6.4	Uncertainty-guided Source-free Domain Adaptation	112
6.4.1	Introduction and methodology	112
6.4.2	Experimental results	114
6.4.3	Conclusions	115
6.4.4	Relevant publications	116
6.4.5	Relevant software/datasets/other outcomes	116
6.4.6	Relevance to AI4media use cases and media industry applications	116
<b>7</b>	<b>Deep quality diversity (Task 3.6) – detailed description</b>	<b>117</b>
7.1	Learned Representations as Diversity Metrics to Maximize	117
7.1.1	Introduction and methodology	117
7.1.2	Experimental Results	118
7.1.3	Relevant publications	121
7.1.4	Relevant software/datasets/other outcomes	121
7.1.5	Relevance to AI4media use cases and media industry applications	121
7.2	Quality Diversity search on the Latent Space	121
7.2.1	Introduction and methodology	121
7.2.2	Experimental Results	123
7.2.3	Relevant publications	125
7.2.4	Relevance to AI4media use cases and media industry applications	125
7.3	Cross-domain Quality-Diversity search	125
7.3.1	Introduction and methodology	125
7.3.2	Experimental Results	127
7.3.3	Relevance to AI4media use cases and media industry applications	129
7.4	User-Controllable Quality Diversity Search	129
7.4.1	Introduction and methodology	129





7.4.2	Experimental Results	130
7.4.3	Relevant publications	133
7.4.4	Relevance to AI4media use cases and media industry applications	133
7.5	Enhancing Preference Learning with Neuroevolution	133
7.5.1	Introduction and methodology	133
7.5.2	Experimental Results	134
7.5.3	Relevant publications	137
7.5.4	Relevance to AI4media use cases and media industry applications	137
<b>8</b>	<b>Learning to count (Task 3.7) – detailed description</b>	<b>138</b>
8.1	QuaPy: A Python-Based Framework for Learning to Quantify	138
8.1.1	Introduction and methodology	138
8.1.2	Relevant publications	140
8.1.3	Relevant software/datasets/other outcomes	141
8.1.4	Relevance to AI4media use cases and media industry applications	141
8.2	Ordinal Quantification through Regularization	141
8.2.1	Introduction and methodology	141
8.2.2	Experimental results	143
8.2.3	Relevant publications	144
8.2.4	Relevant software/datasets/other outcomes	145
8.2.5	Relevance to AI4media use cases and media industry applications	145
8.3	Tweet Sentiment Quantification: An Experimental Re-Evaluation	145
8.3.1	Introduction and methodology	145
8.3.2	Experimental results	147
8.3.3	Relevant publications	148
8.3.4	Relevant software/datasets/other outcomes	148
8.3.5	Relevance to AI4media use cases and media industry applications	148
8.4	Multi-Label Quantification	148
8.4.1	Introduction and methodology	148
8.4.2	Experiments	150
8.4.3	Relevant publications	151
8.4.4	Relevant software/datasets/other outcomes	152
8.4.5	Relevance to AI4media use cases and media industry applications	152
8.5	Other contributions related to Learning to Quantify (Task 3.7)	152
8.5.1	Relevant publications	153
8.5.2	Relevant software/datasets/other outcomes	154
8.5.3	Relevance to AI4media use cases and media industry applications	154
<b>9</b>	<b>Ongoing Work and Conclusions</b>	<b>155</b>
9.1	Ongoing work	155
9.1.1	Lifelong and on-line learning (Task 3.1)	155
9.1.2	Manifold learning and disentangled feature representation (Task 3.2)	155
9.1.3	Transfer learning (Task 3.3)	156
9.1.4	Deep quality diversity (Task 3.6)	157
9.1.5	Learning to count (Task 3.7)	157
9.2	Conclusions	157





## 1. Executive Summary

This deliverable presents the research outcomes obtained between M13 and M36 as a result of the activities carried out in Task 3.1 (Lifelong and On-line Learning), Task 3.2 (Manifold Learning and Disentangled Feature Representation), Task 3.3 (Transfer Learning), Task 3.6 (Deep Quality Diversity), and Task 3.7 (Learning to Count) of WP3 (New Learning Paradigms & Distributed AI). All activities address problems that are central in the Machine Learning community and with methodologies that are at the forefront of the developments in the field. This is reflected by the fact that a large number of the works presented here have resulted in publications in some of the most prestigious and authoritative international journals and conferences in the field. Beyond this, and to increase the impact in the field, several of the works provide software. We make explicit references to the corresponding publications and/or software provided by each partner and establish connections of the presented work with the WP8 Use Cases.

Below, we give a concise motivation and overview of the work in each task – more detailed explanations are found in the relevant sections.

- The **lifelong and on-line learning (Task 3.1)** address the problem of training models which evolve gradually as new data are ingested. This is central to the media industry since new concepts and events occur continually and the underlying Machine Learning models used for their automatic analysis need to be updated continually to ensure an up-to-date processing. This poses certain challenges since it is necessary to ensure a balance between stability and plasticity, two properties which account for the performance obtained for past and new data at each stage of the lifelong or on-line learning processes.

More specifically, the contributions presented in this deliverable include: (a) a method for Class-Incremental Learning (CIL) without memory by creating pseudo-features for past classes to improve their representation and separability, (b) the adaptation of incremental learning strategies to specific use cases, (c) a new decentralized inference strategy for AI agents, (d) knowledge quantification metrics which unveil what deep neural networks learn during their training, (e) a teacher-student network framework which supports “learning by education”, focusing on multiple scenarios with dynamic tasks and goals, (f) a comprehensive survey of Curriculum Learning (CL), (g) an innovative Novel Class Discovery (NCD) algorithm which is able to learn novel classes in absence of labelled data, (h) studying the compatibility of representations learned for data streams and a new distillation method for continual representation learning.

Ongoing work for Task 3.1 considers (i) investigating the advantages and limitations of using large pre-trained models in continual learning, (ii) investigating Generalized Category Discovery (GCD) to automatically cluster partially labeled data, (iii) models that combine and unify Out-of-Distribution (OOD) detection, incremental/continual/lifelong learning, and neural distillation, and (iv) working on a novel learning protocol in which a large model (i.e., foundation model) undergoing Continual Learning will be replaced by an improved one that has been learned from scratch in a compatible way elsewhere (e.g., on a remote server).

- The **manifold learning and disentangled feature representation (Task 3.2)** addresses the problem of learning representations of the data that are meaningful for performing generative and discriminative tasks. This includes generating easily synthetic data, such as faces with the desired expression, manipulating images of people so as not to be identifiable and finding better metrics for comparing images so as to perform search and retrieval.

More specifically, the contributions presented in this deliverable include: (a) the study of the latent and intermediate spaces of pre-trained GANs for discovering interpretable/controllable





generative directions, (b) the incorporation of the remarkable ability of the pre-trained StyleGAN2 and the versatility of its latent space in generating and editing highly realistic faces in order to address the problem of data anonymization, (c) a survey of the literature of manifold learning and its applications in multimedia, (d) the study of fusion strategies of latent space manifolds of multiple fine-tuned models, such as pretrained visual foundation models, (e) a solution to the task of generating dynamic 3D facial expressions from a neutral 3D face and an expression label, (f) a metric learning approach in hyperbolic spaces.

Ongoing work for Task 3.2 considers (i) focusing on Visual-Language models and ways of improving their discriminative ability, (ii) improving OOD performance of neural network models with manifold mixing model soups, (iii) working on generative models on non-linear (manifold) domains focusing on models capable of generating in the combined spatial-temporal domain, and (iv) continuing investigating the underlying structure of the latent spaces of deep generative models with the goal of performing semantically meaningful latent traversals.

- The **transfer learning (Task 3.3)** addresses the problem of reusing previously generated models for tasks that are different than the original ones, tackling the problem of catastrophic forgetting. Considering the huge amount of data, human effort, and computational power needed to train these models, being able to reuse them is of paramount importance.

More specifically, the contributions presented in this deliverable include: (a) an extensive experimental evaluation of Transfer Learning, exploring its trade-offs with respect to performance, environmental footprint, human hours, and computational requirements, (b) a novel concept of source-free open compound domain adaptation Source-Free Open Compound Domain Adaptation (SF-OCDA), (c) a library of self-supervised methods for visual representation learning, (d) a method addressing the problem of Source-Free Domain Adaptation (SFDA) by quantifying the uncertainty in the source model predictions and utilizing it to guide the target adaptation.

Ongoing work for Task 3.3 considers (i) the incorporation of source-target transferability metrics instead of a manual classification of the target datasets, (ii) the application of Test-Time Domain Adaptation in Semantic Segmentation (TTDA-Seg) where both efficiency and effectiveness are crucial, (iii) diversifying training datasets in a programmatic manner using pre-trained foundation models, and (iv) assessing transfer learning abilities in the context of heterogeneous domains, with a specific focus on the domains of Vision and Language.

- The **deep quality diversity (Task 3.6)** studies ways of handling deceptive search spaces by finding a maximally diverse collection of individuals (with respect to a space of possible behaviors) in which each member is as high performing as possible.

More specifically, the contributions presented in this deliverable include: (a) work on using learned representations through deep learning as a method for creating an intrinsic definition of diversity, (b) a novel AI Art generator with capability of producing diverse visual outputs, (c) a direct extension of previous work where rather than just evolving the latent representation of images, they also evolve its corresponding text prompt to generate game artworks paired with a title and description, (d) work on addressing the issue of controllability on quality diversity algorithms, (e) exploring the potential of neuroevolution in Preference Learning (PL) tasks with subjective, unreliable labels such as those found in affective computing.

Ongoing work for Task 3.6 considers (i) carrying out promising experiments following up on existing activities intending for a high-impact journal publication around Computational Creativity, and (ii) co-organizing (UNITN and UM) “Computer Vision for Games and Games





for Computer Vision (CVG)” workshop, to be held on November 23, 2023, as part of the British Machine Vision Conference (BMVC) in Aberdeen, UK.

- **Learning to count (Task 3.7)** addresses the problem of training (under the supervised learning paradigm) estimators of quantities. The main categories of sub-tasks falling under this problem are Learning to Quantify (LQ), which is concerned with training unbiased estimators of class prevalence (i.e., learning to estimate, given a sample of objects, the percentage of objects that belong to a given class), and “Learning to count objects”, which concerns learning to estimate the number of objects (which may be inanimate objects, such as cars, but may also be animate objects, such as people or animals) in visual media, such as still images or video frames.

More specifically, the contributions presented in this deliverable include: (a) an open-source framework for LQ written in Python (QuaPy), (b) work on Ordinal Quantification (OQ), (c) a systematic comparison of LQ methods on the task of tweet sentiment quantification, and (d) a study of the multi-label quantification problem.

Ongoing work for Task 3.7 considers (i) the further development of deep neural networks for LQ by studying the suitability to this task of permutation-invariant operators for set processing, (ii) the application of learning to quantify for estimating the effectiveness of a classifier when applied to unlabelled sets that exhibit dataset shift with respect to the data the classifier has been trained on, and (iii) the problem of tailoring quantification approaches to the particular type of shift that the set of unlabelled data exhibits.

In summary, the work presented in this deliverable has resulted in:

- 19 conference articles (CVPR, ICCV, ECCV, ICLR, ...) and 7 journal articles (TPAMI, JMLR, IJCV, TOMM, TOG, TCSVT, TKDD),
- 35 articles (articles and datasets) available in AI4Media’s Zenodo collection, and
- 21 open-source software and tools publicly available (e.g., in GitHub).

The remainder of this deliverable is structured as follows. In Section 2, we introduce each WP3 task and we give an overview of the contributions of each partner. In Section 3, we provide concise descriptions of the presented works, while detailed descriptions of contributions are given for each task in Section 4 (Task 3.1), Section 5 (Task 3.2), Section 6 (Task 3.3), and Section 7 (Task 3.6), and Section 8 (Task 3.7). All the methods presented in this deliverable can be applied to media-related areas and applications. Indeed after describing each method, we also present their relevance to WP8 Use Cases. Finally, Section 9 concludes the deliverable by summarizing the work covered as well as presenting the ongoing work regarding each task addressed in this deliverable.





## 2. Introduction

The goal of WP3 is to investigate new learning paradigms, looking beyond current achievements in deep learning and focusing among other on topics such as lifelong and continuous learning, manifold and transfer learning, deep quality diversity, and learning to count. In the following, we briefly discuss the challenges related to each of these research topics.

Whilst standard deep learning methods typically assume that all the required training data are readily available, this often poses an unrealistic condition in practice, since real world application-related data often arrive in streams, while their characteristics may vary over time. **Lifelong learning and on-line learning** are two closely related research areas that aim to train models which evolve gradually as new data are ingested. Advances in these fields are in dire need in AI4Media in order to keep up with the dynamic nature of news and media content, since new events appear constantly in them and the models used for their automatic analysis need to be updated regularly to ensure an up-to-date processing. The lifelong and on-line learning contribution of the AI4media partners are related to the practical cases in which: (1) access to past data is limited or impossible, (2) computation needs should be as close to constant as possible and (3) learning of new data needs to be fast. The current contributions of the AI4Media project regarding lifelong and online learning methodologies are given in Section 4. In total, four papers were accepted to be presented in peer-reviewed conferences, three papers were accepted to peer-reviewed journals and two papers are currently under review.

Finding meaningful representation schemes for both generative and discriminative learning tasks has gradually risen as a remarkably important research area, where **manifold and disentangled feature representation learning** methods have contributed significantly during the recent years. In the generative regime, exploring the structure of generative methods (such as GANs), by discovering semantic paths in their latent space that govern the generation process, has proven to be crucial in understanding and controlling image generation. On the other hand, in the discriminative regime, learning meaningful feature representations, along with more appropriate metrics (i.e., that model data manifolds better), lead to better, more discriminative features, and, thus, improve the performance in tasks such as image retrieval. Advances in those fields (i.e., generative and discriminative learning) are particularly useful in media generation and visual content analysis and, thus, in AI4Media use cases, while they are also relevant with WP4 (explainable and interpretable AI). For instance, generative approaches can help towards protecting the identity of individuals depicted in media content (e.g., anonymization of faces appearing in news images/videos), while learning better features of visual content can improve crucial tools in media content analysis, such as, content-based image retrieval, near-duplicate detection, face recognition, person re-identification, zero-shot, and few-shot learning. The current contributions of the AI4Media project regarding manifold learning and disentangled feature representation methodologies are given in Section 5. In total, five papers were accepted to be presented in peer-reviewed conferences and two papers (including a survey paper) are currently under review.

Taking into consideration the vast amount of data, human labour, and computational power that is needed in training modern Deep Learning models, during the recent years, practitioners and researchers have devised **Transfer Learning** techniques that allow to reuse and benefit from previously generated models for various purposes. This is particular useful to the media industry and the use cases of AI4Media, since transfer learning methods provide solutions to analyze/adapt the visual content (by virtue of being able to generalize under domain-gap), discover new visual content, and adapt accordingly. Beyond practical reasons, Transfer Learning poses an important scientific challenge, as it forces researchers to explore the internal knowledge representation of deep models and unveil their structure and how learning is being conducted before being able to reuse them for diverse purposes. Advances in this field have potential relevance for key aspects of Deep





Learning, not only explainability and interpretability, but also efficiency and footprint reduction, as well as deployment of AI powered systems in real world scenarios. These are relevant to other work packages of AI4Media – i.e., regarding explainable and interpretable AI (WP4) and learning from scarce real-world data (WP5). The current contributions of the AI4Media project regarding transfer learning methodologies are given in Section 6. One paper was accepted to be presented in a peer-reviewed conference and two papers were accepted to peer-reviewed journals.

Finding a maximally diverse collection of individuals (regarding a space of possible behaviors) in which each member is performing as high as possible is an important research field finding applications, among other areas, in media content that have strict quality requirements (such as games). **Quality-Diversity (QD)** methods have been recently appeared in the Evolutionary Computation (EC) literature as a way of handling such deceptive search spaces. Drawing inspiration from natural evolution, which – unlike the objective-based optimization tasks to which EC is typically applied – is primarily open-ended, QD algorithms re-introduce a notion of localized quality among individuals with the same behavioral characteristics. QD algorithms aim to obtain balance between their individuals’ quality and their population’s diversity, and thus media content with strict quality requirements, such as games that are start-to-end playable, are the ideal ground for advancing quality-diversity. The developed QD methods are useful in the media industry and the AI4Media use cases by providing tools for (i) generating diverse content without requiring ad-hoc designer-specified directions for this diversity, and (ii) modelling the subjective human game players experiences so as to dynamically adapt the game according to the predicted user’s engagement or arousal levels. The current contributions of the AI4Media project regarding deep quality diversity methodologies are given in Section 7. In total, 4 papers were accepted to be presented in peer-reviewed conferences.

Datasets that are being used by the research community and the industry for training machine learning models typically exhibit shift, i.e., the joint distribution of the independent and the dependent variables is not the same in the training data and in the unlabelled data for which predictions must be obtained. When this occurs, estimating the prevalence of the classes of interest in the unlabelled data is difficult, since “traditional” learning methods assume these prevalence values to stay approximately constant. **“Learning to Count”** is concerned with developing techniques for estimating quantities in unlabelled data possibly affected by dataset shift, where these quantities can be the prevalence values (i.e., relative frequencies) of the classes of interest (as needed in applications such as monitoring consensus for a certain policy or political candidate in social media) or the number of physical objects in instances of visual media (such as estimating car park occupancy from surveillance camera images, or monitoring traffic volumes from road cameras). The contributions of the AI4Media project regarding learning-to-count methodologies are given in Section 8. In total, 2 papers were accepted to be presented in peer-reviewed conferences and 2 papers were accepted for publication in international journals; a book was also published on this topic.

To summarize, the contributions presented in this deliverable address problems that are central in the Machine Learning community, providing methodologies that are at the forefront of the developments in the field. The activities of the partners led to a significant number of high-quality and diverse works that have been published in some of the most prestigious and authoritative international journals and conferences in the field.







## 3. Concise descriptions of the presented works

In the following, we briefly summarise the outcomes of each WP3 task for the period M13-M36. These works are then presented in detail in sections 4-8.

### 3.1. Lifelong and on-line learning (Task 3.1)

#### 3.1.1. Introduction

Standard deep learning methods assume that all training data are available at once. This hypothesis is often unrealistic since application-related data arrive in streams, and their characteristics shift over time. Lifelong learning and on-line learning are two closely related research topics whose purpose is to train models which evolve gradually as new data are ingested. This learning process is challenging because it is necessary to ensure a balance between stability and plasticity, two properties which account for the performance obtained for past and new data at each stage of the lifelong or on-line learning processes. Advances in these fields are particularly needed in AI4Media in order to keep up with the dynamic nature of media content (e.g., breaking news). New concepts and events occur continually in them and the underlying models used for their automatic analysis need to be updated continually to ensure an up-to-date processing.

We report the research outcomes of Task 3.1 in detail in Section 4.

#### 3.1.2. Overview

The partners involved in Task 3.1 tackle different open challenges in lifelong and on-line learning. The contributions are summarized below and then discussed in more detail in the following subsections.

In Subsection 4.1, CEA introduces a method for CIL without memory by creating pseudo-features for past classes to improve their representation and separability. The method is inspired by transfer learning, and compares favorably with more complex algorithms which update their representations at each incremental step.

In Subsection 4.2, CEA studies the adaptation of incremental learning strategies to specific use cases. Such an adaptation is needed because incremental learning scenarios are very diversified, and none of the existing methods outperforms all others in all cases.

In Subsection 4.3, AUTH proposes a new decentralized inference strategy for AI agents. The method is inspired by the human decision making process, which is driven by interactions between persons. The method is based on a Quality of Inference (QoI) consensus protocol, which formalizes a common inference rule which is applied by each agent.

In Subsection 4.4, AUTH studies knowledge quantification metrics, which unveil what DNNs learn during their training. The main properties of existing metrics are analyzed to highlight their advantages and limitations. The applicability of the metrics to different architectures, modalities and tasks is emphasized.

In Subsection 4.5, AUTH introduces a teacher-student network framework which supports "learning by education" which focuses on multiple scenarios with dynamic tasks and goals. The framework enables dynamic interactions between all agents, whose roles as students or teachers can change over time. The proposed education process is iterative to enable gradual acquisition of knowledge.

In Subsection 4.6, UNITN summarizes a comprehensive survey of CL, a training methodology whose underlying hypothesis is that samples should be ordered from easy to hard during training. The study presents existing methods and underlines that sample ordering and adequate pacing are not straightforward. It also describes promising future research directions.





In Subsection 4.7, UNITN introduces an innovative NCD algorithm which is able to learn novel classes in absence of labelled data, while preserving performance of base classes. The proposed algorithm mixes feature replay and distillation with self training. It compares very favorably with existing methods which tackle the same problem in two challenging evaluation settings.

In Subsection 4.8, UNIFI studies the compatibility of representations learned for data streams. The proposed method leverages the stationarity of internal representations in order to make the features learned in different step comparable. The method is evaluated with excellent results for face verification, re-identification and retrieval tasks.

In Subsection 4.9, UNIFI proposes a second algorithm for learning compatible representation for dynamic data. This algorithm is based on a mix of rehearsal and feature stationarity, which is encouraged at global and local levels. The method compares favorably with respect to a number of recent incremental learning methods.

In Subsection 4.10, UNIFI introduces a new distillation method for continual representation learning. The objective is to align current and previous features of the same class while separating features of different classes. The method makes an innovative use of contrastive loss, and obtains strong performance for fine-grained classification datasets.

## 3.2. Manifold learning and disentangled feature representation (Task 3.2)

### 3.2.1. Introduction

In recent years, manifold and disentangled feature representation learning have risen as a prominent research area addressing the problem of finding meaningful representation schemes for both the generative and the discriminative learning paradigms. In the generative regime, studying the structure of latent spaces of generative methods (such as GANs) by discovering semantic paths that govern the generation process, has proven to be very useful in understanding and controlling image generation. For instance, by discovering interpretable or controllable generative paths for manipulating the generation process (e.g., image editing) [1–3]. In the discriminative regime, learning meaningful feature representations, along with metrics that model data manifolds better (i.e., by adopting the hyperbolic geometry instead of the widely used Euclidean modeling [4]), lead to better, more discriminative features, and, thus, improve significantly the performance in visual understanding tasks (such as image retrieval). Advances in both generative and discriminative regimes are particularly useful in media generation and visual content analysis.

We report the research outcomes of Task 3.2 in detail in Section 5.

### 3.2.2. Overview

Within this task partners are contributing in fundamental aspects of manifold learning and disentangled feature representation, coordinating so that their advances can contribute to one another, and with the use cases of the project in mind. To further detail this collaboration, let us first summarize the contributions of partners.

In Subsection 5.1, QMUL studies how to discover, in an unsupervised and model-agnostic manner, interpretable and disentangled paths in the latent space of a pre-trained GAN. That is, paths in the latent space sampling across which is expected to lead to image generations that differ only in a few factors (e.g., changing the expression of a face, the rotation of an object, etc). For doing so, they model non-linear paths using RBF-based warping functions, which by warping the latent space, endow it with vector fields (i.e., their gradients). Then the latter are used to traverse the latent space across the paths determined by the aforementioned vector fields for any given





latent code. Each warping function gives rise to a family of non-linear paths. This work proposes to learn a set of such warping functions, i.e., a set of such non-linear path families, so as the image transformations that they produce are distinguishable to each other by a discriminator network.

In Subsection 5.2, QMUL studies the structure of feature spaces of pre-trained GANs, that is, intermediate representations of the GAN generators instead of their latent spaces, in an architecture-agnostic approach that jointly discovers factors representing spatial parts and their appearances in an entirely unsupervised fashion. These factors are obtained by applying a semi-nonnegative tensor factorization on the feature maps, which in turn enables context-aware local image editing with pixel-level control. In addition, they show that the discovered appearance factors correspond to saliency maps that localize concepts of interest, without using any labels. Experiments on a wide range of GAN architectures and datasets show that, in comparison to the State of the Art (SOTA), the proposed method is far more efficient in terms of training time and, most importantly, provides much more accurate localized control.

In Subsection 5.3, QMUL and UNITN incorporate the remarkable ability of the pre-trained StyleGAN2 and the versatility of its latent space in generating and editing highly realistic faces in order to address the problem of face anonymization. By contrast to the existing literature, this is the first work that anonymizes the identities of those depicted in a facial dataset, while at the same time it retains the facial attributes of the original images in the anonymized counterparts, the preservation of which is of paramount importance for their use in downstream tasks. For doing so, this work presents a task-agnostic anonymization procedure that directly optimises the images' latent representation in the latent space of a pre-trained GAN. By optimizing the latent codes directly, the proposed framework ensures both that the identity is of a desired distance away from the original (with an identity obfuscation loss), whilst preserving the facial attributes (using a novel feature-matching loss in FaRL's deep feature space). Through a series of both qualitative and quantitative experiments it is shown that the proposed method is capable of anonymizing the identity of the images whilst—crucially—better-preserving the facial attributes.

In Subsection 5.4, JR surveys the literature of manifold learning and its applications in multimedia. Deep Learning has been the dominant paradigm for the automatic analysis of multimedia data (e.g. images, video or 3D data) for tasks such as classification or detection. However, classic neural networks are restricted to data lying in vector spaces, while data residing in smooth non-Euclidean spaces arise naturally in many problem domains (e.g., a 360° camera actually captures a spherical image, not a rectangular image). This survey focuses on manifolds, especially Riemannian manifolds, which are well suited for generalizing a vector space because they are locally Euclidean and differentiable.

In Subsection 5.5, JR studies fusion strategies of latent space manifolds of multiple finetuned models, such as pretrained visual foundation models (CLIP [5] or CoCa [6]). To do so, they propose the manifold mixing model soup (ManifoldMixMS) algorithm, which, instead of simple averaging, it uses a more sophisticated strategy to generate the fused model. Specifically, it partitions a neural network model into several latent space manifolds (which can be individual layers or a collection of layers). Afterwards, from the pool of finetuned models available after hyperparameter tuning, the most promising ones are selected and their latent space manifolds are mixed together individually. The optimal mixing coefficient for each latent space manifold is calculated automatically via invoking an optimization algorithm. Experiments show that the fused model gives significantly better OOD performance when finetuning a CLIP model for image classification.

In Subsection 5.6, UNIFI proposes a solution to the task of generating dynamic 3D facial expressions from a neutral 3D face and an expression label by solving the following two sub-problems: (i) modeling the temporal dynamics of expressions, and (ii) deforming the neutral mesh to obtain the expressive counterpart. This method represents the temporal evolution of expressions using the motion of a sparse set of 3D landmarks that is learnt to generate by training a manifold-





valued GAN (Motion3DGAN). To better encode the expression-induced deformation and disentangle it from the identity information, the generated motion is represented as per-frame displacement from a neutral configuration. To generate the expressive meshes, the method trains a Sparse2Dense mesh Decoder (S2D-Dec) that maps the landmark displacements to a dense, per-vertex displacement. This allows for learning how the motion of a sparse set of landmarks influences the deformation of the overall face surface, independently from the identity. Experimental results on the CoMA and D3DFACS datasets show that the proposed solution brings significant improvements with respect to previous solutions in terms of both dynamic expression generation and mesh reconstruction, while retaining good generalization to unseen data.

In Subsection 5.8, UNITN studies the problem of metric learning; that is, the problem of learning highly discriminative models encouraging the embeddings of similar classes to be close in the chosen metrics and pushed apart for dissimilar ones. Whilst the common recipe is to use an encoder to extract embeddings and a distance-based loss function to match the representations (typically the Euclidean distance), in this work a new hyperbolic-based model for metric learning is proposed (using a vision transformer with output embeddings mapped to hyperbolic space). These embeddings are directly optimized using modified pairwise cross-entropy loss, while the proposed method is evaluated with six different formulations on four datasets achieving the new state-of-the-art performance.

### 3.3. Transfer learning (Task 3.3)

#### 3.3.1. Introduction

Transfer Learning is an emerging field among Deep Learning practitioners that seeks to reuse and exploit previously generated models for different purposes. Considering the huge amount of data, human effort and computational power needed to train these models, being able to reuse them is of paramount importance. Beyond practical reasons, Transfer Learning poses a scientific challenge of relevance, as it forces researchers to question the internal knowledge representation of deep models. Indeed, to understand how to reuse deep representations, one must first understand how are these representations learned, and how are they internally structured. Advances in this field have potential relevance for key aspects of Deep Learning, such as explainability and interpretability, efficiency and footprint reduction, and real world deployment of AI powered systems. We report the research outcomes of Task 3.3 in detail in Section 6

#### 3.3.2. Overview

Within this task partners are contributing in fundamental aspects of Transfer Learning, coordinately so that their advances can contribute to one another, and with the use cases of the project in mind. To further detail this collaboration, let us first summarize the contribution of partners.

In Subsection 6.1, BSC conducts an experimental evaluation of Transfer Learning, exploring its trade-offs with respect to performance, environmental footprint, human hours, and computational requirements. They provide results that highlight the cases where a cheap Feature Extraction (FE) approach is preferable, and the situations where an expensive fine-tuning effort may be worth the added cost. Finally, they propose a set of guidelines on the use of Transfer Learning.

In Subsection 6.2, UNITN introduces the novel concept of source-free open compound domain adaptation (SF-OCDA), which they study in the task of semantic segmentation. While SF-OCDA is more challenging than the traditional domain adaptation, it is yet more practical. It jointly considers (1) the issues of data privacy and data storage and (2) the scenario of multiple target domains and unseen open domains. In SF-OCDA, only the source pre-trained model and the target





data are available to learn the target model. The model is evaluated on the samples from the target and unseen open domains. To solve this problem, this work proposes an effective framework by separating the training process into two stages: (1) pre-training a generalized source model and (2) adapting a target model with Self-supervised Learning (SSL). In this framework, Cross-Patch Style Swap (CPSS) is proposed to diversify samples with various patch styles in the feature-level, which can benefit the training of both stages. First, CPSS can significantly improve the generalization ability of the source model, providing more accurate pseudo-labels for the latter stage. Second, CPSS can reduce the influence of noisy pseudo-labels and also avoid the model overfitting to the target domain during self-supervised learning, consistently boosting the performance on the target and open domains. Experiments demonstrate that the proposed method produces state-of-the-art results on the C-Driving dataset. Furthermore, it also achieves the leading performance on CityScapes for domain generalization.

In Subsection 6.3, UNITN presents solo-learn, a library of self-supervised methods for visual representation learning. Implemented in Python, using Pytorch and Pytorch lightning, the library fits both research and industry needs by featuring distributed training pipelines with mixed-precision, faster data loading via Nvidia DALI, online linear evaluation for better prototyping, and many additional training tricks. The goal of this work is to provide an easy-to-use library comprising a large amount of SSL methods, that can be easily extended and fine-tuned by the community. solo-learn opens up avenues for exploiting large-budget SSL solutions on inexpensive smaller infrastructures and seeks to democratize SSL by making it accessible to all.

In Subsection 6.4, UNITN studies the problem of SFDA, whose aim is to adapt a classifier to an unlabelled target data set by only using a pre-trained source model. In order to address the unreliability of the predictions on the target data (due to the absence of the source data and the domain shift), this work proposes quantifying the uncertainty in the source model predictions and utilizing it to guide the target adaptation. For this, a probabilistic source model is constructed by incorporating priors on the network parameters inducing a distribution over the model predictions. Uncertainties are estimated by employing a Laplace approximation and incorporated to identify target data points that do not lie in the source manifold and to down-weight them when maximizing the mutual information on the target data. Unlike other existing works, the proposed probabilistic treatment is computationally lightweight, decouples source training and target adaptation, and requires no specialized source training or changes of the model architecture.

### 3.4. Deep quality diversity (Task 3.6)

#### 3.4.1. Introduction

Quality-Diversity (QD) algorithms have been recently introduced to the EC literature as a way of handling deceptive search spaces. The goal of these algorithms is “to find a maximally diverse collection of individuals (with respect to a space of possible behaviors) in which each member is as high performing as possible” [7]. The inspiration for such approaches is natural evolution which is primarily open-ended—unlike the objective-based optimization tasks to which EC is often applied. While the rationale of open-ended evolution has been previously used as an argument for genetic search for pure behavioral novelty, QD algorithms re-introduce a notion of (localized) quality among individuals with the same behavioral characteristics. QD algorithms attempt to balance between their individuals’ quality and their population’s diversity, and thus media content which have strict quality requirements, such as games that are playable from start to finish, are the ideal arena for advancing quality-diversity.

The aim of Task 3.6 is to couple deep neural network architectures with divergent search for transforming exploration, aiming for both diverse and high quality outcomes. Experiments in this





deep-learning-based QD (deepQD) search approach during the reported period are aligned on two main directions:

D1 improve the definition of diversity based on learnt representations.

D2 promote diversity and quality in existing deep learning generative architectures for media.

We report the research outcomes of Task 3.6 in detail in Section 7.

### 3.4.2. Overview

We present three main contributions split along the core directions (D1, D2) described above, as well as two complementary directions on quality diversity evolution (Section 7.4) or neuroevolution without diversity preservation (Section 7.5), which can be merged at later stages with deepQD.

In Subsection 7.1, UM describes their work on using learned representations through deep learning as a method for creating an intrinsic definition of diversity. In their approach, UM explore the use of adaptive novelty search in the latent space, which is determined by an AutoEncoder (AE) that is periodically retrained on novel data generated during previous iterations of the algorithm. They extend this concept into the 3D domain for the first time by implementing a Minecraft building generator based on the DeLeNoX [8] algorithm using CPPN-NEAT [9]. Furthermore, UM conducted an extensive study of different methods for retraining the autoencoder during evolution, and shedding light on its impact onto effectively measuring novelty and producing interesting content. This software has been open-sourced and is publicly available via GitHub under a creative commons (CC0-1.0) license.

In Subsection 7.2, UM implements a novel AI Art generator with capability of producing diverse visual outputs. They achieve this by applying a QD evolutionary search using Novelty Search with Local Competition (NSLC) in interim phases of a GAN process, and observe the impact it has on its creative process and produced content. Specifically, this involves a refinement cycle, using VQGAN latent vector back-propagation to turn random noise into desired images, and an exploration cycle, applying NSLC [10] to the latent vector. Their results show that this approach achieves a small increase in diversity compared to a typical GAN approach, but opens up interesting areas for future research on the diversification of generation images. UM also explored different diversity metrics for the generated images, with chromatic diversity being the most reliable despite its simplicity, and highlight other potential measures for future research such as LPIPS and image compressibility.

Subsection 7.3 is a direct extension of UM's work in Section 7.2 where, rather than just evolving the latent representation of images, they also evolve its corresponding text prompt to generate game artworks paired with a title and description. UM also proposes an enhancement to the MAP-Elites algorithm [11] when applied to multiple modalities, called MAP-Elites with Transverse Assessment (MEliTA). In this experiment, a pair of fine-tuned GPT-2 models create hypothetical game titles and descriptions from a Steam games catalogue, and a Stable Diffusion model generates corresponding cover images to form an initial population. This population is then refined through MEliTA, which uses an Upper Confidence Bound (UCB) selection strategy [12], mutations of both text and image elements, and considers mutual influence between image and text modalities. This ongoing research confirms the usefulness of MAP-Elites for preserving diversity, even in multi-modal creative tasks. Early experiments show that MEliTA offers a marginal advantage to conventional MAP-Elites, with low computational overhead, and is being expanded to more modalities in an effort to further optimize this approach.

In Subsection 7.4, UM carried out extensive research on addressing the issue of controllability on QD algorithms. They introduce a novel algorithm for Interactive Evolutionary Computation (IEC) called User Controller MAP-Elites (UC-ME), which provides the user with a higher degree of





control whilst minimizing their fatigue. They achieve this by adjusting the MAP-Elites algorithm to focus on a small area of the feature map, present design options from this area to the user, and then shift focus based on the user's selection. This approach is tested on the complex and constrained task of generating architectural layouts. The researchers at UM show that, when compared to unguided MAP-Elites, UC-ME is able to find individuals more directly aligned with the user's preferences whilst still covering large regions of the feasible problem space. While UC-ME is not strictly applied to deep QD problems, it can be extended to operate on latent representations (as e.g. in Section 7.3) or to use learnt characterizations of diversity that the user can further control (as e.g. in Section 7.1).

In Subsection 7.5, the UM researchers focus on exploring the potential of neuroevolution in PL tasks with subjective, unreliable labels such as those found in affective computing. They introduce a novel algorithm called *RankNEAT*, which is built on the efficient RankNet architecture [13] and is enhanced through neuroevolution. They put RankNEAT to test against the vanilla RankNet in the task of player affect modeling across three games, using arousal-annotated gameplay videos from the AGAIN dataset. The results indicate that RankNEAT outperforms RankNet in training PL models of arousal in most of the conducted experiments, suggesting its viability as a PL paradigm. Furthermore, this research opens up opportunities expanding its quality-driven optimization with a diversification mechanism, which could advance affect modelling and extend the application of deepQD beyond generative domains. While research on RankNEAT is not yet combined with deep QD (as it operates on optimization as minimized error), it can be expanded with additional diversity characteristics in future iterations of T3.6 tasks.

### 3.5. Learning to count (Task 3.7)

#### 3.5.1. Introduction

“Learning to Count” is a task having to do with supervised learning approaches for training estimators of quantities. There are two classes of problems that are being addresses in this task, and that may be usefully viewed as forming two different subtasks, i.e.,

- “Learning to quantify” (LQ – a.k.a. *quantification*). This subtask is concerned with training unbiased estimators of class prevalence via supervised learning, i.e., learning to estimate, given a sample of objects, the percentage of objects that belong to a given class. This task originates with the observation that “Classify and Count (CC)”, the trivial method of obtaining class prevalence estimates, is often a biased estimator, and thus delivers suboptimal quantification accuracy. This bias is particularly strong when the data exhibits *dataset shift*, i.e., when the joint distribution of the dependent and the independent variables is not the same in the training data and in the unlabelled data for which predictions must be issued. Quantification is important for several applications, e.g., gauging the collective satisfaction for a certain product from textual comments, establishing the popularity of a given political candidate from blog posts, predicting the amount of consensus for a given governmental policy from tweets, or predicting the amount of readers who will find a product review helpful.
- “Learning to count objects”. This subtask has to do with using machine learning approaches in order to train estimators of the number of objects (which may be inanimate objects, such as cars, but may also be animate objects, such as people or animals) in visual media, such as still images or video frames. Example applications of these techniques are, e.g., counting the number of cars in a video frame (in order to estimate traffic volume or car park occupancy), or counting the number of people in a still image (say, in order to estimate the amount of people taking part in a rally).





We report the research outcomes of Task 3.7 in detail in Section 8.

### 3.5.2. Overview

We here present five main contributions. All these contributions have to do with the first subtask (“Learning to quantify”), since the contributions concerning the second subtask (“Learning to count objects”) intersect Task 5.3 (“Learning with scarce data”) and will thus be reported in the WP5 deliverables related to Task 5.3.

In Section 8.1 CNR describes QuaPy, an open-source framework for LQ written in Python. QuaPy provides implementations of a number of baseline methods and advanced quantification methods, of routines for quantification-oriented model selection, of several broadly accepted evaluation measures, and of robust evaluation protocols routinely used in the field. QuaPy also makes available datasets commonly used for testing quantifiers, and offers visualization tools for facilitating the analysis and interpretation of the results. The software is open-source and publicly available under a BSD-3 licence via GitHub, and can be installed via `pip`.

In Section 8.2, CNR describes their work on OQ, i.e., the case in which a total order is defined on the set of  $n > 2$  classes for which quantification is to be performed. In this work, CNR researchers give three main contributions to this field. First, they create and make available two datasets for OQ research that overcome the inadequacies of the previously available ones. Second, they experimentally compare, on the above datasets, the most important OQ algorithms proposed in the literature thus far. To this end, for the first time they bring together algorithms that had been proposed by authors from very different research fields (e.g., data mining and astrophysics), and who were thus unaware of each other’s developments. Third, they propose three OQ algorithms, based on the idea of preventing “ordinally implausible” estimates through regularization. Their experiments show that these algorithms outperform the existing ones.

In Section 8.3, CNR describes their analysis of a previous work [14] where a systematic comparison of LQ methods on the task of tweet sentiment quantification was carried out. In hindsight, they observe that the experimentation carried out in that work was weak, and that the reliability of the conclusions that were drawn from the results of [14] is thus questionable. They thus re-evaluate those quantification methods (plus a few more modern ones) on exactly the same datasets, this time following a now consolidated and robust experimental protocol (which also involves simulating the presence, in the test data, of class prevalence values very different from those of the training set). This experimental protocol (even without counting the newly added methods) involves a number of experiments 5,775 times larger than that of the original study. Due to the above-mentioned presence, in the test data, of samples characterised by class prevalence values very different from those of the training set, the results of the new experiments are dramatically different from those presented in [14], and provide a different, much more solid understanding of the relative strengths and weaknesses of different sentiment quantification methods.

In Section 8.4, CNR researchers observe that, while many quantification methods have been proposed in the past for binary problems and, to a lesser extent, single-label multiclass problems, the multi-label setting (i.e., the scenario in which the classes of interest are not mutually exclusive) remains by and large unexplored. A straightforward solution to the multi-label quantification problem could simply consist of recasting the problem as a set of independent binary quantification problems. However, CNR researchers observe that such a solution is simple but naïve, since the independence assumption upon which it rests is, in most cases, not satisfied. In these cases, knowing the relative frequency of one class could be of help in determining the prevalence of other related classes. They thus propose the first truly multi-label quantification methods, i.e., methods for inferring estimators of class prevalence values that strive to leverage the stochastic dependencies among the classes of interest in order to predict their relative frequencies more accurately. They







show empirical evidence that natively multi-label solutions outperform the naive approaches by a large margin.

Finally, in Section [8.5](#) CNR describes other contributions related to LQ made in the reporting period.





## 4. Lifelong and on-line learning (Task 3.1) – detailed description

**Contributing partners:** CEA, AUTH, UNITN, UNIFI

Standard deep learning methods assume that all training data are available at once. This hypothesis is often unrealistic since application-related data arrive in streams, and their characteristics shift over time. Lifelong learning and on-line learning are two closely related research topics whose purpose is to train models that constantly evolve as new data are ingested. This poses certain challenges in the learning process since balance between stability and plasticity needs to be guaranteed, two crucial properties that account for the performance obtained for past/new data at each stage of the lifelong or on-line learning stages. Advances in these fields are needed in AI4Media in order to keep pace with the dynamic nature of news and media content. New concepts and events occur continually in them and the underlying models used for their automatic analysis need to be updated continually to ensure an up-to-date processing.

### 4.1. FeTrIL: Feature Translation for Exemplar-Free Class-Incremental Learning

**Contributing partners:** CEA

#### 4.1.1. Introduction and methodology

Incremental learning [15] was introduced to reduce the memory and computational costs of machine learning algorithms. The main problem faced by CIL methods is catastrophic forgetting [16, 17], the tendency of neural nets to underfit past classes when ingesting new data. Many recent solutions [18–22], based on deep nets, use replay from a bounded memory of the past to reduce forgetting. However, replay-based methods make a strong assumption because past data are often unavailable [23]. Also, the footprint of the image memory can be problematic for memory-constrained devices [24]. Exemplar-Free Class-Incremental Learning (EFCIL) methods recently gained momentum [25–28]. Most of them use distillation [29] to preserve past knowledge, and generally favor plasticity. New classes are well predicted since models are learned with all new data and only a representation of past data [30–32]. A few EFCIL methods [33, 34] are inspired by transfer learning [35, 36]. They learn a feature extractor in the initial state, and use it as such later to train new classifiers. In this case, stability is favored over plasticity since the model is frozen [30].

We introduce FeTrIL, a new EFCIL method which combines a frozen feature extractor and a pseudo-feature generator to improve incremental performance. New classes are represented by their image features obtained from the feature extractor. Past classes are represented by pseudo-features which are derived from features of new classes by using a geometric translation process. This translation moves features toward a region of the features space which is relevant for past classes. The proposed pseudo-feature generation is adapted for EFCIL since it is simple, fast and only requires the storage of the centroids for past classes. FeTrIL is illustrated with a toy example in Figure 1.

#### 4.1.2. Experimental results

##### 4.1.2.1. Experimental Setup

**Datasets.** We have used four datasets to run exemplar-free CIL experiments: CIFAR-100 [37]



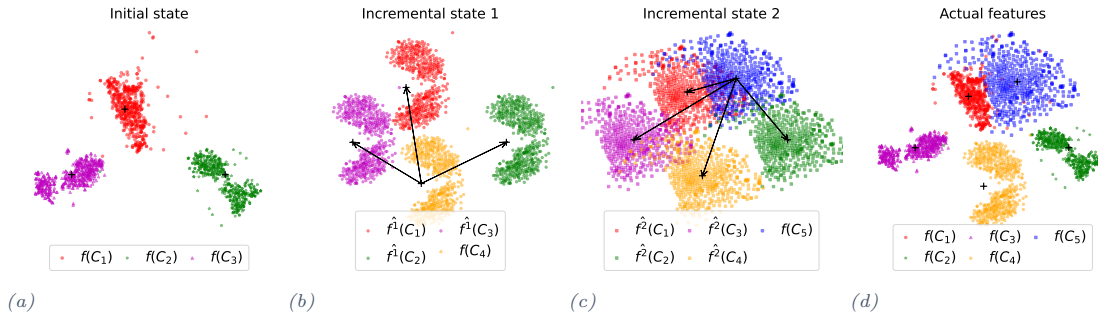


Figure 1. Illustration of the proposed pseudo-feature generation procedure. This toy example includes an initial state (3 classes) and two IL states (1 new class per state) in subfigures (a), (b) and (c). Subfigure (d) provides the actual features of all classes that would be available for a classical learning. The illustration uses a 2D projection of actual features. Pseudo-features of past classes are generated by geometric translation of features of the new class added in each state with the difference between the centroids of the target past class and of the new class. While imperfect, the pseudo-feature generator produces a usable representation of past classes. Best viewed in color.

which includes 100 classes, TinyImageNet [38] which includes 200 ImageNet subclasses, ImageNet-Subset with 100 classes of ILSVRC [39], and the full ILSVRC with 1000 classes.

**EFCIL scenarios.** We use a classical EFCIL protocol from [27, 28, 32]. The number of classes in the initial state is larger, and the rest of the classes are evenly distributed between incremental states. CIFAR-100 and ImageNet-Subset are tested with: (1) 50 initial classes and 5 IL states of 10 classes, (2) 50 initial classes and 10 IL states of 5 classes, (3) 40 initial classes and 20 states of 3 classes, and (4) 40 initial classes and 60 states of 1 class.

**Metric.** The average incremental accuracy, widely used in CIL [20, 30], is the main evaluation measure. For comparability with [27, 28, 32], it is computed as the average accuracy of all states, including the initial one.

**Implementation details.** Following [20, 27, 28, 32], we use ResNet-18 [40] in all experiments. FeTrIL initial training is done uniquely with images of initial classes to ensure comparability with existing methods. The feature extractor is trained in the initial state and then frozen for the remainder of the IL process. We implement a supervised training with cross-entropy loss, SGD optimization, a batch size of 128, for a total of 160 epochs. The initial learning rate is 0.1, and it is decayed by 0.1 after every 50 epochs.

#### 4.1.2.2. Comparison to State-of-the-art Methods

We use the following EFCIL methods in evaluation: LwF-MC [20], DeeSIL [33], LUCIR [19], SDC [25], PASS [28], IL2A [27], SSRE [32]. These methods cover a variety of EFCIL approaches, and some of them were proposed recently.

The results from Table 2 show that FeTrIL outperforms all compared methods in 11 tested configurations out of 12. It is also close to the best in the remaining one. The second best results are obtained with the very recent SSRE method [32]. FeTrIL and SSRE accuracies are close to each other for CIFAR-100, but our model is better for the other datasets. PASS [28] and IL2A [27], two other recent EFCIL methods, have lower average performance. We note that EFCIL performance boost was recently reported, with methods such as PASS, IL2A, SSRE. These methods combine knowledge distillation and sophisticated mechanisms for dealing with the stability-plasticity dilemma. In contrast, our method uses a fixed feature extractor and a lightweight pseudo-feature generator. FeTrIL only optimizes a linear classification layer, while compared recent methods use backpropagation of the entire model, and need much more computational resources and time to perform the IL process. Performance of the ILSVRC dataset is also very interesting. While



CIL Method	CIFAR-100				TinyImageNet				ImageNet-Subset				ImageNet		
	$T=5$	$T=10$	$T=20$	$T=60$	$T=5$	$T=10$	$T=20$	$T=100$	$T=5$	$T=10$	$T=20$	$T=60$	$T=5$	$T=10$	$T=20$
LwF-MC* [20] (CVPR'17)	45.9	27.4	20.1	x	29.1	23.1	17.4	x	-	31.2	-	x	-	-	-
LUCIR (CVPR'19)	51.2	41.1	25.2	x	41.7	28.1	18.9	x	56.8	41.4	28.5	x	47.4	37.2	26.6
SDC* [25] (CVPR'20)	56.8	57.0	58.9	x	-	-	-	x	-	61.2	-	x	-	-	-
PASS* [28] (CVPR'21)	63.5	61.8	58.1	x	49.6	47.3	42.1	x	64.4	61.8	51.3	x	-	-	-
IL2A* [27] (NeurIPS'21)	<u>66.0</u>	60.3	57.9	x	47.3	44.7	40.0	x	-	-	-	x	-	-	-
SSRE* [32] (CVPR'22)	65.9	<u>65.0</u>	<b>61.7</b>	x	50.4	48.9	48.2	x	-	67.7	-	x	-	-	-
DeeSIL [33] (ECCVW'18)	60.0	50.6	38.1	x	49.8	43.9	34.1	x	67.9	60.1	50.5	x	61.9	54.6	45.8
DSLDA [41] (CVPRV'20)	64.0	63.8	60.8	<b>60.5</b>	<u>53.1</u>	<u>52.9</u>	<b>52.8</b>	<b>52.6</b>	<u>71.3</u>	<b>71.2</b>	<b>71.0</b>	<b>70.8</b>	<u>64.0</u>	<u>63.8</u>	<u>63.6</u>
FeTrIL	<b>66.3</b>	<b>65.2</b>	<u>61.5</u>	<u>59.8</u>	<b>54.8</b>	<b>53.1</b>	<u>52.2</u>	<u>50.2</u>	<b>72.2</b>	<b>71.2</b>	<u>67.1</u>	<u>65.4</u>	<b>66.1</b>	<b>65.0</b>	<b>63.8</b>

Table 2. Average top-1 incremental accuracy in EFCIL with different numbers of incremental steps. FeTrIL results are reported with pseudo-features translated from the most similar new class. "-" cells indicate that results were not available (see supp. material for details). "x" cells indicate that the configuration is impossible for that method. Best results - in bold, second best - underlined.

direct comparison to PASS or SSRE is impossible since these methods were not tested at scale, ILSVRC results show that the simple method proposed here is effective for a high range of classes. Interestingly, ILSVRC performance is stabler compared to smaller datasets since the pool of new classes available for pseudo-features generation is larger.

**Comparison to a transfer-learning baseline.** DeeSIL [33] is a simple application of transfer learning to EFCIL. It has no class separability mechanism across different incremental states since classifiers are learned within each state. The important performance gain brought by FeTrIL highlights the importance of class separability.

**Behavior for minimal incremental updates.** Compared EFCIL methods can only be updated with a minimum of two classes per CIL state since they use discriminative classifiers, which require both positive and negative samples. This is possible with FeTrIL because pseudo-features can all originate from a single new class. Results in the right columns of CIFAR-100, TinyImageNet and ImageNet-Subset from Table 2 show that the accuracy obtained in with one class increments is close to that observed for  $T = 20$ . This highlights the robustness of FeTrIL with respect to frequent updates.

#### 4.1.3. Conclusion

The main contributions of this work are to:

- Introduce of a simple and scalable method for EFCIL which is based on pseudo-features generation.
- Perform extensive experiments which show the effectiveness of FeTrIL compared to existing methods whose computational requirements are order of magnitudes higher.
- Confirm previous findings [30, 42] regarding the strong performance of transfer-learning-based methods in EFCIL compared to the majority of existing methods which use a combination of fine-tuning and distillation. This is important in the current AI context, since foundation models can be easily leveraged by methods such as FeTrIL.

#### 4.1.4. Relevant publications

- Petit, G., Popescu, A., Schindler, H., Picard, D., and Delezoide, B. (2023). Fetril: Feature translation for exemplar-free class-incremental learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3911-3920). [43]. Zenodo record: <https://zenodo.org/record/7498807>.





#### 4.1.5. Relevant software/datasets/other outcomes

- The PyTorch + Python implementation of FeTrIL can be found in <https://github.com/GregoirePetit/FeTrIL>.

#### 4.1.6. Relevance to AI4media use cases and media industry applications

FeTrIL enables a fast and effective updating of incremental learning models. As such, it can be integrated in use cases which work with dynamic datasets and require swift updates of their recognition models. In UC2 (2A and 2B), this algorithm can be used to process new topics which appear in news. This is, for instance, the case of face recognition models which need to integrate continuously faces of new people who occur in the news. Another relevant use case is UC3 (3C), where techniques such as FeTrIL can be used for the automatic management of unexpected journalistic events. More generally, lightweight incremental learning models which are built on top of pretrained models can be embedded in media applications which deal with sequential data streams which exhibit representation shift over time.

## 4.2. AdvisIL - A Class-Incremental Learning Advisor

**Contributing partners:** CEA

### 4.2.1. Introduction and methodology

A major challenge of CIL is that it is subject to catastrophic forgetting [16,17], namely the tendency of learning algorithms to abruptly forget previously acquired information when confronted with new information. In order to learn reliable neural representations both for past and new classes, CIL algorithms must balance between information retention, i.e. stability, and information acquisition, i.e. plasticity. However, existing comparative studies [30,31,44] showed that when CIL algorithms are tested in different incremental scenarios, no method outperforms all others. In addition to the CIL algorithm itself, the main factors influencing the classification performance are the architecture of the backbone neural network and the characteristics of the CIL scenario i.e., the memory budget, the number of incremental steps, the number of classes in the initial step and the size of the subsequent incremental steps.

MobileNet	15.7	10.9	15.0	18.5	11.5	21.8	50.6	44.7	50.3	46.0	32.5	42.6
ResNet	26.3	18.7	24.2	16.6	16.2	19.6	48.6	47.0	51.2	37.4	37.0	35.8
ShuffleNet	17.3	11.6	15.3	19.2	12.0	22.2	51.7	47.6	52.2	44.9	34.5	42.3
	DSLDA	DeeSIL	FeTrIL	LUCIR	SIW	SPB-M	DSLDA	DeeSIL	FeTrIL	LUCIR	SIW	SPB-M
	(a)						(b)					

Figure 2. Classification performance in percent for various combinations of CIL algorithm and backbone network, averaged over five reference datasets containing 100 classes each in total. Scenario (a) has a memory budget of 1.5M parameters and consists of 20 steps with 5 classes each. Scenario (b) has a memory budget of 3.0M parameters, and consists of 4 steps with 25 classes each. Here, as highlighted in purple, scenario (a) is best handled by the combination of DSLDA and ResNet, while the combination of FeTrIL and ShuffleNet is a better match for scenario (b). As the same combinations of CIL algorithm and backbone network are not ranked the same from one scenario to another, it highlights the need for a recommendation method to select the best combination of CIL algorithm and backbone network depending on the scenario.

The performance variability for two scenarios is illustrated in Figure 2. It shows that the same combination of CIL algorithm and backbone network is not ranked the same from one EFCIL





scenario to another. Thus, unlike other learning processes (i.e. classical, few-shot...) for which the study of the SOTA on evaluation benchmarks gives good indications for the selection of the model, EFCIL requires a more acute consideration of the learning scenario. These observations raise the following questions:

1. *Since choosing the best CIL algorithm first requires characterizing the CIL scenario, what knowledge of this scenario may be realistically provided by the user?*
2. *Given a user's CIL scenario, how to select a suitable combination of learning algorithm and backbone network, without benchmarking each possible configuration?*

We argue that, regarding (1), the users have little knowledge about their data and can only approximately provide their CIL scenario's characteristics. Based on this assumption, we propose to tackle the selection issue (2) as a recommendation problem. We develop a user-centric method, called AdvisIL, that recommends a combination of an exemplar-free CIL algorithm and a backbone network scaled to the user's needs. Based on a set of pre-computed EFCIL experiments, AdvisIL provides a recommendation as follows:

1. The users specify their incremental learning settings (memory budget, number of steps, number of initial classes and size of the incremental update).
2. The pre-computed experiments with settings closest to the user's settings are selected.
3. The result is the combination of EFCIL algorithm and backbone network that ranks highest in terms of classification performance with respect to the selected experiments.

As a result, our recommendation method facilitates the use of CIL approaches since the user only provides the essential information about the incremental process. It prevents the users from benchmarking each CIL algorithm and backbone network, hence saving them time and computation efforts. AdvisIL is assessed by an evaluation protocol which uses four test datasets and eighteen scenarios, allowing us to highlight the relevance of our recommendations in a variety of experimental settings. To allow the use and the enrichment of AdvisIL by the community, and thus the quality of its recommendations, we will share the code and the pre-computed experiments on which the method is based.

## 4.2.2. Experimental results

### 4.2.2.1. Experimental Setup

**EFCIL algorithms.** In our experiments, we use a representative panel of the algorithms: LUCIR [19], SPB [45], SIW [46], DeeSIL [33], DSLDA [41] and FeTrIL. We remind that LUCIR, SPB and SIW update the backbone network at each incremental step, and thus focus on the plasticity of representations. In contrast, DeeSIL, DSLDA and FeTrIL use a representation which is fixed after the initial step, and thus favour stability. We implement all algorithms using PyTorch [47] (implementation details can be found in [48]).

**Backbone networks.** In our experiments, we use: ResNet18 [40], MobileNetv2 [49] and ShuffleNetv2 [50]. ResNet18 is widely used in the CIL literature. MobileNetv2 and ShuffleNetv2 are designed for high accuracy while considering computational efficiency for embedded applications. These backbone networks are scaled to fit various memory budgets.

**Datasets for reference configurations.** We consider five datasets which are sampled from ImageNet [51] and denoted by INFood, INFauna, INFlora, INRand<sub>0</sub> and INRand<sub>1</sub>.





**CIL scenarios.** The memory budget  $m$  is defined as the number of parameters of the final model (that contains 100 classes here) and is taken in  $\{1.5M, 3M, 6M\}$ . The chosen budgets reflect the computational constraints of embedded devices, for which CIL is particularly useful

**Test datasets** We ran experiments on four test datasets denoted by INRand<sub>2</sub>, FOOD100, INAT100 and LAND100. INRand<sub>2</sub> is obtained by randomly sampling 100-classes of ImageNet [51]. The three other test datasets FOOD100, INAT100 and LAND100 contain 100 classes sampled from FOOD101 [52], iNaturalist [53] and Google Landmarks v1 [54], respectively.

**Test scenarios.** The memory, algorithms and backbone networks budgets are the same three as those used for generating reference configurations. We run experiments for each combination of algorithm, backbone network, test dataset and test scenario. This corresponds to 1296 test configurations.

**Evaluation protocol** As in the case of reference configurations, in our evaluation, model performance is measured in terms of average incremental accuracy. We assess the recommendations provided by AdvisIL as follows. Given a test dataset and a test scenario, we compare the average incremental accuracy of the models trained according to:

- i) AdvisIL’s recommended pair algorithm-backbone pair. The model built with this pair is called the *recommended model*.
- ii) *Oracle* pair: an upper-bound for AdvisIL which selects by brute force the best-performing pair of algorithm and backbone network for each test configuration.
- iii) *Baseline pairs*, which are three fixed combinations algorithm-backbone combinations: (FeTrIL, ResNet), (DSLDA, ShuffleNet) (SPB, MobileNet) whose corresponding models are called *baseline models*. These pairs were selected according to their aggregated rank on reference datasets.

Settings		Incr. acc.	Incr. acc. difference			
			AdvisIL	$\Delta_{\mathcal{O}}$	$\Delta_{b1}$	$\Delta_{b2}$
CIL setting $(k, \alpha, \beta)$	(50, 2, 2)	24.42	-1.49	1.01	0.81	9.72
	(25, 4, 4)	35.07	-0.63	0.30	4.35	13.44
	(5, 20, 20)	55.74	-0.65	0.97	3.16	7.46
	(13, 40, 5)	62.56	-0.73	2.33	0.06	12.17
	(11, 50, 5)	64.40	-1.26	2.02	0.22	9.98
	(6, 50, 10)	64.12	-1.56	1.12	0.55	6.64
Budget $m$	1.5M	45.94	-1.46	0.37	2.76	6.96
	3.0M	51.85	-0.79	2.32	0.52	10.27
	6.0M	54.86	-0.92	1.18	1.30	12.47
Test dataset	INRand <sub>2</sub>	52.02	-0.73	1.20	1.71	12.51
	INAT100	50.18	-1.22	1.57	1.66	10.03
	FOOD100	28.03	-1.39	1.99	0.34	5.22
	LAND100	74.52	-0.89	0.40	2.89	11.85
<b>Avg</b>		50.88	-1.04	1.29	1.52	9.90

Table 3. Classification performance of models built with AdvisIL’s recommendations on four test datasets and six test scenarios. Results are grouped either by user-defined  $(k, \alpha, \beta)$ , a memory budget  $m$ , or by test dataset. The last row corresponds to the average on all test configurations. The difference between the classification performance of the oracle pair and AdvisIL’s recommendation is showed ( $\Delta_{\mathcal{O}racle}$ ). Similarly, the performance gaps between the three baseline recommendations ( $\Delta_{b1}, \Delta_{b2}, \Delta_{b3}$ ) and AdvisIL’s recommendation are showed.

#### 4.2.2.2. Results

In Table 3, we compare the performance of models built according to AdvisIL’s recommendation





to the performance of models built according to the oracle pair and to the three baseline pairs. On average, across all four test datasets and eighteen test scenarios, the recommended model outperforms the best fixed model by 1.29%. The accuracy of the recommended models is below the oracle, with an average gap of 1.04%. The gap between the average classification performance of the recommended model and of that of the oracle is stable across scenarios, regardless of the number of steps and class distribution among the steps, and regardless of the memory budget. It is also stable across test datasets. Therefore, the recommendations are relevant whatever the scenario and dataset.

#### 4.2.3. Conclusion

The main contributions of this work are to:

- Introduce a recommendation method for EFCIL algorithms which are adapted to a specific usage context.
- Propose neural architecture scaling methods to adapt them for on-device learning constraints.
- Evaluate different algorithms and neural architectures for a variety of CIL scenarios to show the usefulness of the proposed recommendation approach.

#### 4.2.4. Relevant publications

- Feillet, E., Petit, G., Popescu, A., Reyboz, M., and Hudelot, C. (2023). AdvisIL-A Class-Incremental Learning Advisor. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2400-2409). [48].  
Zenodo record: <https://zenodo.org/record/7498775>.

#### 4.2.5. Relevant software/datasets/other outcomes

- The PyTorch+Python implementation of AdvisIL can be found in <https://github.com/EvaJF/AdvisIL>.

#### 4.2.6. Relevance to AI4media use cases and media industry applications

AdvisIL is as a decision support tools for incremental learning practitioners to select suitable continual learning methods for their specific applications. As such, it can be used by media industry professionals who have limited technical expertise in continual learning to make informed decisions about the implementation of this type of techniques in their AI-powered data processing pipelines. It can be integrated in the following AI4Media use cases: (1) UC2 (2A and 2B) to process new topics which appear in news and thus improve tagging and search capabilities in an effective and efficient manner, (2) UC3 (3C) because the automatic management of unexpected journalistic events requires a swift update of recognition models for relevant content such as faces or company brands.

### 4.3. Towards Human Society-inspired Decentralized DNN Inference

**Contributing partner(s):** AUTH







### 4.3.1. Introduction and methodology

In human societies, individuals make their own decisions and they may select if and who may influence it. At a societal level, the overall knowledge is preserved and enhanced by individual person empowerment, where complicated consensus protocols have been developed over time in the form of societal mechanisms to assess, weigh, combine, and isolate individual people’s opinions. In distributed machine learning environments, however, individual AI agents are merely part of a system where decisions are made in a centralized and aggregated fashion or require a fixed network topology, a practice prone to security risks and collaboration is nearly absent. Inspired by societal practices, we propose a decentralized inference strategy where each individual agent is empowered to make their own decisions, by exchanging and aggregating information with other agents in their network. To this end, a "Quality of Inference" consensus protocol (QoI) is proposed, forming a single commonly accepted inference rule applied by every individual agent. The overall system knowledge and decisions on specific manners can thereby be stored by all individual agents in a decentralized fashion, employing e.g., blockchain technology. Moreover, a fault-tolerant inference architecture in which misbehaving AI agents are penalized, reducing their influence on the decision-making process of honest agents is designed.

Let  $\mathcal{G} = \{\mathcal{A}, \mathcal{E}\}$  be a direct acyclic graph consisting of  $M$  collaborating AI agents described in a set  $\mathcal{A} = \{\alpha_1, \alpha_2, \dots, \alpha_M\}$ , that are employed to perform some inference task, e.g., classification, and  $\mathcal{E}$  defined as a set of fixed communication links allowing them to communicate with each other. It is assumed that all agents have obtained access to the same test sample  $\mathbf{x}$ , while their goal for them is to produce a single prediction  $\hat{y}$ . This work differentiates the following strategies:

*Centralized inference* refers to the case where each AI agent  $\alpha_i$  produces an intermediate prediction  $y_i$  for a given test sample  $\mathbf{x}_i$ . A master node thereby collects and aggregates the individual AI agent predictions and produces the final system output, using e.g., an average/median rule or majority voting.

*Distributed inference* refers to the case where individual nodes only perform computational tasks, i.e., the inference task is divided between each of the nodes and/or a master node, and the final output of the system for a given test sample is provided by the master node.

*Decentralized inference* refers to the case where individual nodes  $\alpha_i$  make their own predictions for a given sample as in the centralized inference case, however, the aggregation is performed by all participating AI nodes, using a consensus protocol. A conceptual diagram presenting the different problem variants is shown in Figure 3.



Figure 3. A conceptual diagram of centralized, distributed, and decentralized inference.

### 4.3.2. Experimental results

QoI protocol operates in rounds in which each consensus round is defined as one execution of the normal operation process regardless if it is successful or not. Views describe the consensus





rounds that are required in order for the network to reach a consensus about a given sample and are defined as an index of the form  $v \in \mathcal{V}$ , containing a sequence of testing pairs whose predictions have been scheduled in the time interval  $t$ . At each view, one agent is operating as *primary* while the rest  $M - 1$  agents are operating as *validators*. Our goal is that every honest agent in  $M$  maintains an identical prediction history set defined as  $\hat{\mathcal{Y}} = \{\hat{y}_{ij}, \forall v \in \mathcal{V} \text{ and } j \in \mathcal{C}\}$  given a set of  $\mathcal{C} = \{c_1, c_2, \dots, c_C\}$  classes where  $\|\mathcal{C}\| = C$ .

For each consensus round, a primary agent,  $a_p \in \mathcal{A}$  be the primary agent, the election formula is defined as:

$$a_p = v \bmod |\mathcal{A}|, \quad (1)$$

where  $|\mathcal{A}| = M$  and  $v \in \mathcal{V}$  represent the view we are currently working on.

**View Change.** Once the primary agent of the current view is detected as faulty, view change is performed in order to be replaced. Specifically, in the  $v^{th}$  view, the primary agent is promoting a prediction for the  $i^{th}$  sample of the form:

$$\hat{y}_p = \operatorname{argmax}(f_p(\mathbf{x}_i)). \quad (2)$$

Validators observe the produced prediction and vote accordingly. Let  $a_j \in \mathcal{A}$  represent a random validator that has just received the primary's message. If its predicted value  $\hat{y}_j \neq \hat{y}_p$  or  $v_j \neq v_p$  then, from now onwards, the  $j^{th}$  agent recognizes the primary as faulty and is voting for its replacement as:

$$\text{vote}_j = \begin{cases} 1, & \text{if } \hat{y}_j \neq \hat{y}_p \text{ or } v_j \neq v_p \\ 0, & \text{otherwise} \end{cases}. \quad (3)$$

If the current primary is honest, it is rewarded by a predetermined amount of quality points ( $q$ ) for its honest work.

For a given primary  $a_p$ , the reward and the penalty are calculated as:

$$r_p = \begin{cases} q, & \text{if } \frac{\sum_{i=1}^{M-1} \text{vote}_i}{|\mathcal{A}|} < 0.5 \\ 0.5r_p, & \text{if } \frac{\sum_{i=1}^{M-1} \text{vote}_i}{|\mathcal{A}|} \geq 0.5 \end{cases}. \quad (4)$$

We conducted experiments for Individual Agent Decision Aggregation (IADA) and QoI protocols respectively. According to our setup, the base-line agents are first boosted via the IADA method, and then a consensus agreement is established between them. The proposed IADA method is reported with the acronym DA for the probability-based condition and DWA for the weighted-based condition (see Table 4). For the QoI protocol, the three rules are clearly reported. Comparisons are conducted with majority voting and weighted average aggregation methods.

Our experiments showed that this framework allows each individual agent to make their own decisions by exchanging and aggregating information with other agents in their network, in an effort to improve individual performance. Additionally, it has been shown that by adopting a fault-tolerant inference architecture, miss-behaving AI agents can be punished in a way that dramatically lessens their ability to influence the decisions of good agents. Our classification task studies have demonstrated that the suggested methodologies form a secure decentralized inference framework, which hinders adversaries from interfering with the whole process and delivers performance comparable to centralized decision aggregation techniques.

#### 4.3.3. Relevant publications

- D. Papaioannou, V. Mydgalis and I. Pitas, "Towards Human Society-inspired Decentralized DNN Inference", Under Review [55].





Table 4. Comparison of Aggregation Methods in Real Datasets

Experiments	Dataset	Centralized Voting Rules		QoI Consensus Protocol		
		Weight Average	Majority Voting	Class Rule	Weight Rule	Hybrid Rule
Base Agents		94.09	93.75	93.58	94.12	93.62
DA	SVHN	-	-	94.02	<b>94.20</b>	93.99
DWA		-	-	93.97	94.16	93.95
Base Agents		95.12	95.05	95.04	95.27	94.97
DA	Cifar-10	-	-	95.05	95.21	95.14
DWA		-	-	95.16	<b>95.29</b>	95.17
Base Agents		74.65	73.96	71.47	71.42	71.64
DA	Cifar-100	-	-	73.84	74.01	73.85
DWA		-	-	74.80	<b>74.96</b>	74.74
Base Agents		<b>92.51</b>	92.01	91.94	92.33	92.01
DA	F-MNIST	-	-	92.15	92.28	92.17
DWA		-	-	92.16	92.13	92.14
Base Agents		<b>80.11</b>	79.29	78.19	78.21	78.49
DA	STL-10	-	-	79.49	79.44	79.34
DWA		-	-	79.19	79.07	79.27

#### 4.3.4. Relevance to AI4media use cases and media industry applications

Our paper is related and contributes to UC7 "AI for Content Organization and Content Moderation" and UC1 "AI against Disinformation" as it proposes a decentralized inference strategy that can be incorporated into advanced deep learning techniques for content analysis. Inspired by societal practices, we propose a decentralized decision-making strategy where individual neural agents are empowered to make their own decisions, by exchanging and aggregating information with other agents in a shared network. Through the integration of advanced AI functionalities and the proposed decentralized inference strategy, media companies can efficiently and cost-effectively manage their visual content, ensuring its relevance and safety, since our method prevents overall process tampering. Ultimately, while our decentralized inference strategy promotes individual empowerment within AI systems, it also enhances security and fosters collaboration between multiple neural agents.

## 4.4. Quantifying the knowledge in Deep Neural Networks: an overview

**Contributing partner(s):** AUTH

### 4.4.1. Introduction and methodology

Deep Neural Networks (DNNs) have proven to be extremely effective at learning a wide range of tasks. Due to their complexity and frequently inexplicable internal state, DNNs are difficult to analyze: their black-box nature makes it challenging for humans to comprehend their internal behavior. Several attempts to interpret their operation have been made during the last decade, but analyzing deep neural models from the perspective of the knowledge encoded in their layers is a very promising research direction, which has barely been touched upon. Such a research approach could provide a more accurate insight into a DNN model, its internal state, learning progress, and



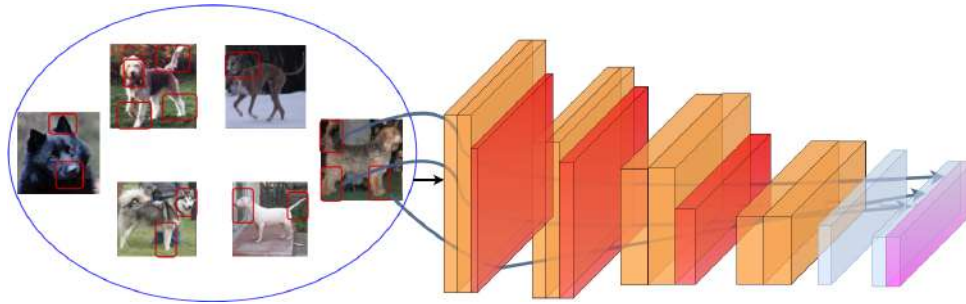


Figure 4. The emergence of possible knowledge points encoded by the DNN visualized as image regions.

knowledge storage capabilities. The purpose of this work is two-fold: a) to review the concept of DNN knowledge quantification and highlight it as an important near-future challenge, as well as b) to provide a brief account of the scant existing methods attempting to actually quantify DNN knowledge.

Knowledge quantification metrics would allow for more precise conclusions regarding what a DNN has learned during its training process. Although no commonly accepted definition of DNN knowledge has been proposed yet, several metrics falling under this general area have been presented in recent years. The most obvious and naive choice is to simply measure the accuracy of DNN predictions on a known test set. Besides this, two types of more advanced methods have emerged: a) *information-theoretic metrics*, which leverage an individual DNN layer’s information to quantify the knowledge it encodes, and b) *knowledge points metrics*, that quantify the knowledge points stored in a trained DNN.

A group of DNN knowledge quantification methods measures a trained DNN layer’s information. The underlying assumption is that the amount of this information is proportional to the knowledge this layer encodes, thus it may act as a proxy for the latter one. Entropy has been a metric widely utilized in recently developed approaches for information quantification. Information-theoretic DNN knowledge quantification methods measure a trained DNN layer’s information.

The amount of knowledge of each DNN layer is measured as knowledge points, i.e., input units, whose information is regarded as important for decision-making, since it is discarded much less than the information of others (Figure 4). The amount of information discarding of each input unit is formulated as the entropy  $H(\mathcal{X}_c)$ , where  $\mathcal{X}_c$  denotes a set of inputs  $\mathbf{x}_c$  corresponding to the concept of a specific object instance.

#### 4.4.2. Experimental Results

We have identified two types of DNN knowledge quantification metrics: *information-theoretic* and *knowledge points metrics*. Figure 5 depicts the evolution of the knowledge quantification methods studied in this survey. The relevant metrics are summarized in Table 5.

As it can be observed in Table 5, all methods primarily aim to interpret and define the DNN knowledge, to design appropriate quantification metrics. Information-theoretic metrics measure a trained DNN layer’s information assuming that the amount of this information is proportional to the knowledge this layer encodes. Although this assumption is drawn as an obvious conclusion, specifically defining the DNN knowledge is demanded. Knowledge points methods directly define knowledge as the total amount of knowledge points encoded by the DNN and fairly quantify its actual amount. They can interpret and diagnose the inner workings of DNNs not only by measuring the number of encoded knowledge points but moreover, by evaluating their quality.



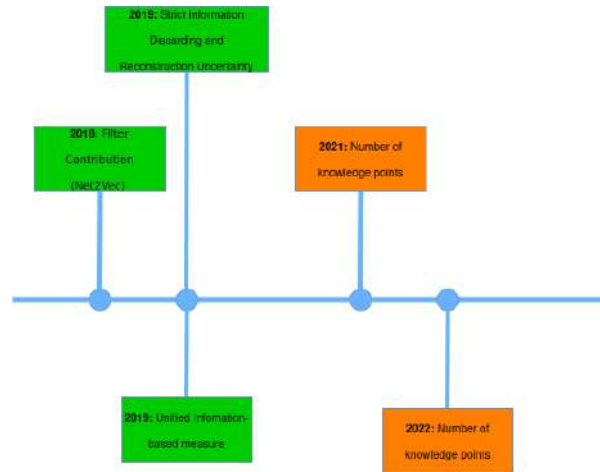


Figure 5. A timeline of the metrics defined to quantify the knowledge of Deep Neural Networks.

Table 5. Knowledge quantification methods.

Method	Architecture	Modality	Task	Year
Information-theoretic metrics				
Strict Information Discarding [56]	Generic	Generic	Generic	2019
Reconstruction Uncertainty [56]	Generic	Generic	Generic	2019
Filter Contribution (Net2Vec) [57]	CNN	Images	Segmentation	2018
Unified Information-based measure [58]	Generic	Language	Generic	2019
Geometric Mutual Information [59]	Generic	Generic	Generic	2021
Knowledge points metrics				
Number of knowledge points [60]	Generic	Generic	Classification	2022
Number of knowledge points [61]	CNN	Images	Classification	2021

A reliable DNN knowledge quantification metric should meet the coherency and generality criteria as overviewed in Table 6. *Coherency* implies that a method needs to enable fair layer-wise comparisons and fair comparisons between different networks. *Generality* refers to the fact that a method should have strong connections to existing mathematical theories and should be defined without regard for model architectures or tasks.

A number of the methods summarized above have been designed specifically for deep CNNs. However, the vast majority are generic and applicable to multiple different neural architectures. The majority of the relevant experiments found in the literature are conducted on image analysis tasks. Almost all of the presented methods focus solely on experiments for classification settings. Given the level of maturity other computational tasks have achieved thanks to DNNs, it is evident that this is still an emerging area in urgent need of additional investigation. Novel metrics need to be precisely defined and alternative approaches to measure the knowledge of trained DNN knowledge must be discovered.





Table 6. Comparisons of the methods in terms of coherency and generality.

Method	Coherency		Generality
	Fair layer-wise comparisons	Fair network comparisons	
Information-theoretic methods			
Strict Information Discarding [56]	Yes	Yes	Yes
Reconstruction Uncertainty [56]	Yes	Yes	Yes
Filter Contribution (Net2Vec) [57]	Yes	Yes	No
Unified Information-based measure [58]	Yes	Yes	Yes
Geometric Mutual Information [59]	Yes	No	Yes
Knowledge points based methods			
Number of Knowledge points [60]	Yes	Yes	Yes
Number of Knowledge points [61]	Yes	Yes	No

#### 4.4.3. Relevant publications

- I. Valsamara, I. Mademlis and I. Pitas, “Quantifying the knowledge in Deep Neural Networks: an overview”, Under Review [62].

#### 4.4.4. Relevance to AI4media use cases and media industry applications

While our work contributes to UC7 “AI for Content Organisation and Moderation” as it explores techniques incorporated into advanced deep learning methods, it can be also incorporated in all AI4MEDIA use cases where the knowledge of Deep Neural Networks (DNNs) needs to be quantified. Specifically, within UC7, the need for AI-powered tools to efficiently categorize, tag, and moderate media content is paramount. Understanding DNNs by shedding light on how they operate and store knowledge, is pivotal in achieving these goals efficiently. By studying state of the art knowledge quantification methods for DNNs, our work facilitates the development of more robust and effective AI algorithms to be utilized by the media sector.

## 4.5. Knowledge Distillation-driven Communication Framework for Neural Networks: Enabling Efficient Student-Teacher Interactions

**Contributing partner(s):** AUTH

### 4.5.1. Introduction and methodology

Humans acquire new knowledge over time through education, while Artificial Intelligence (AI) systems are trained on specific datasets. Teacher-Student network frameworks connect AI agents supporting learning by education, instead of learning from data. The proposed Teacher-Student network framework introduces a novel agent architecture that can adapt to different task objectives and perform knowledge self-assessment. The framework facilitates the exchange of knowledge between teacher agents, who possess domain-specific knowledge, and student agents, who have the ability to select competent teachers for learning. It supports “learning by education”, instead of





”learning from data”. In the multi-agent environment, student agents can learn from the existing knowledge within the framework, rather than relying on additional labeled data to perform better on each task, as it is shown in Figure 6.

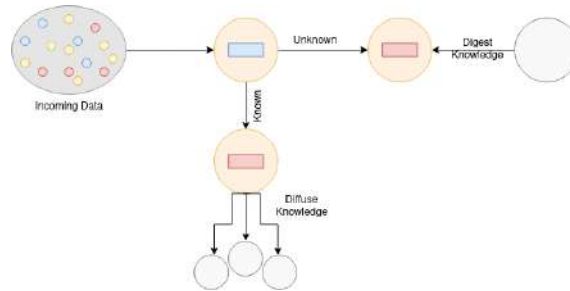


Figure 6. A DNN agent inside a teacher-student network framework, digesting or diffusing domain-specific knowledge.

The proposed framework focuses on multiple education scenarios, where the task and its goals are dynamic, and therefore, the task-relevant DNN knowledge should be available to be digested. At all times, each agent possesses the ability of *self-knowledge assessment* indicating that it understands its state and task environment by evaluating its own knowledge via the knowledge self-assessment module. Moreover, any agent may seek knowledge from the most knowledgeable and qualified teacher agents, access and evaluate it, and choose one or more specific domain-educated teachers to digest new knowledge. The framework connects DNN agents and enables student and teacher interactions between them as can be seen in Figure 7. All agents can become students and digest knowledge, or teachers and diffuse their knowledge. After each education cycle, the student model is expected to be much more competent in its task, due to its old and novel education.

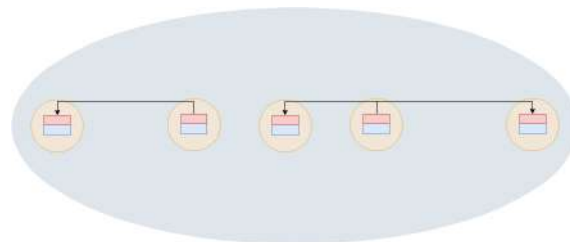


Figure 7. An unconstrained multi-agent environment where each agent can become a teacher and diffuse knowledge or a student and digest knowledge depending on the incoming data.

The DNN agent structure is composed of multiple cooperating modules as shown in Figure 8 and is detailed below. The Knowledge module is typically a Deep Neural Network classification model, capable of learning and performing inference. The Knowledge Self-Assessment (KSA) module is able to perform self-assessment and provide trustworthy decisions about whether the agent is knowledgeable or not at the current time, for each task, utilizing an Out Of Distribution Detection (OODD) algorithm. OODD methods can be employed in cases of pre-trained neural network models making inferences with test-phase inputs drawn from a data distribution that may differ from that of the training dataset. Using the KSA module, the agent is able to make decisions about the new data and whether they belong to its training data domain or not in order to decide if it can handle them efficiently. The KSA module is a Variational Auto-Encoder using an OOD detection score, namely Logistic Regression [63]. LR can be regarded as the log ratio of the likelihood gained from





the generative model Variational AutoEncoder (VAE) with the variational posterior distribution optimal configuration for any particular input data, to its likelihood acquired from the VAE trained on the training data set. KSA is pre-trained and is able to discriminate, at deployment time, between known or unknown knowledge module inputs, testing whether new data belong to the domain in which the agent is trained.

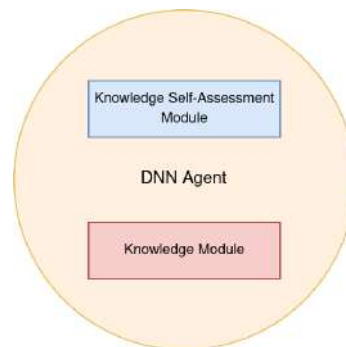


Figure 8. The agent structure inside the teacher-student network framework.

#### 4.5.2. Experimental results

The experiments conducted, regarding the functionality of the framework, aim to test the pipeline and the alternative choices each node have. To this end, five nodes were initialized, with the following properties:

- Node 1 with out-of-distribution detector (FMNIST), classifier (FMNIST) with 71,37% accuracy.
- Node 2 with out-of-distribution detector (MNIST), classifier (MNIST) with 85,62% accuracy.
- Node 3 with no out-of-distribution detector or classifier.
- Node 4 with out-of-distribution detector (FMNIST), classifier (FMNIST) with 69,29% accuracy
- Node 5 with no out-of-distribution detector or classifier, that will face new data (student).

The initial step involves providing new FMNIST data to Node 5 to evaluate the overall functionality of the framework. Since Node 5 does not have an out-of-distribution detector, it forwards the data to the entire framework comprising Nodes 1-4 in search of potential teachers. Both Node 1 and Node 4 respond positively, indicating their availability as potential teachers. To determine the most knowledgeable teacher, their knowledge is assessed by measuring their accuracy scores on the given data. After the assessment process of the available teachers, Node 1 demonstrates superior knowledge and is thus assigned the role of the teacher within the framework. Once Node 1 is chosen as the teacher, the student (Node 5) within the framework is provided with the following scenarios to acquire knowledge from the teacher:

- S.1** request training data
- S.2** seek knowledge distillation using soft-output
- S.3** seek knowledge distillation using feature layers







**S.4** request the teacher’s weights

The results from using the different scenarios are gathered in Table 7.

Table 7. Comparisons on the accuracy values of the proposed knowledge acquisition scenarios, with FMNIST as the student’s knowledge acquisition target.

	<b>S.1</b>	<b>S.2</b>	<b>S.3</b>	<b>S.4</b>
<i>Accuracy</i>	71.37%	86.59%	88.23%	71.37%

To better illustrate our point, we employed small architectures in our experiments. We specifically chose an architecture that, when trained using traditional learning methods, could not surpass an accuracy level of 72%. However, by incorporating knowledge distillation techniques, the same architecture was able to achieve an impressive accuracy of 88%.

Similarly, we conducted the same experiment using the CIFAR10 and SVHN datasets, which yielded consistent results. The nodes in this experiment are as follows:

- Node 1 with out-of-distribution detector (CIFAR10), classifier (CIFAR10) with 56.91% accuracy.
- Node 2 with out-of-distribution detector (SVHN), classifier (SVHN) with 75.86% accuracy.
- Node 3 with no out-of-distribution detector or classifier.
- Node 4 with out-of-distribution detector (CIFAR10), classifier (CIFAR10) with 53.17% accuracy
- Node 5 with no out-of-distribution detector or classifier, that will face new data (student).

Nodes 1 and 4 are also in this case, possible teachers. The results from using the different scenarios are gathered in Table 8.

Table 8. Comparisons on the accuracy values of the proposed knowledge acquisition scenarios, with CIFAR10 as the student’s knowledge acquisition target.

	<b>S.1</b>	<b>S.2</b>	<b>S.3</b>	<b>S.4</b>
<i>Accuracy</i>	56.91%	57.72%	57.89%	56.91%

#### 4.5.3. Relevant publications

- A. Kaimakamidis, I. Valsamara and I. Pitas, “Knowledge Distillation-driven Communication Framework for Neural Networks: Enabling Efficient Student-Teacher Interactions”, technical report [64].

#### 4.5.4. Relevance to AI4media use cases and media industry applications

Our research paper introduces a novel framework designed to facilitate communication and knowledge exchange among Deep Neural Networks (DNNs), serving as a versatile solution for DNN-to-DNN interactions. This framework offers opportunities to enhance learning capabilities, foster





collaboration, and improve overall network performance. It finds relevance in several AI4Media use cases where advanced deep learning techniques play pivotal roles, such as UC3 "AI in Vision - High quality Video Production and Content Automation". It offers a novel method for knowledge transfer among neural networks, making it a valuable asset in addressing the challenges of content organization, content enhancement, and media content analysis. Ultimately, our framework promotes co-operation between media sector companies, by increasing performance of each individual AI system through knowledge sharing in a Deep Learning agent grid, while ensuring each agent's capability of privacy preservation.

#### 4.6. Curriculum Learning: A Survey

**Contributing partner(s):** UNITN

Training machine learning models in a meaningful order, from the easy samples to the hard ones, using CL can provide performance improvements over the standard training approach based on random data shuffling, without any additional computational costs. Curriculum learning strategies have been successfully employed in all areas of machine learning, in a wide range of tasks. However, the necessity of finding a way to rank the samples from easy to hard, as well as the right pacing function for introducing more difficult data can limit the usage of the curriculum approaches. Our goal was to show how these limits have been tackled in the literature, and to present different curriculum learning instantiations for various tasks in machine learning. We also wanted to provide some interesting directions for future work.

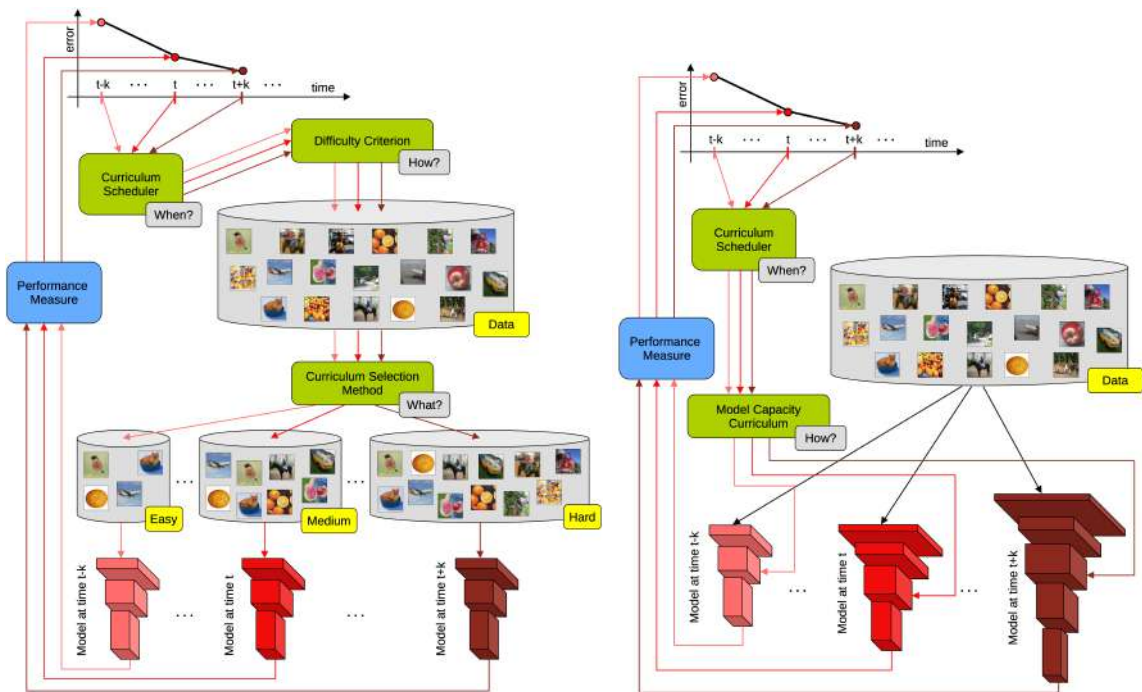
**Context and motivation.** Deep neural networks have become the state-of-the-art approach in a wide variety of tasks, ranging from object recognition in images [65–68] and medical imaging [69–72] to text classification [73–76] and speech recognition [77, 78]. The main focus in this area of research is on building deeper and deeper neural architectures, this being the main driver for the recent performance improvements. For instance, the CNN model of Krizhevsky et al. [65] reached a top-5 error of 15.4% on ImageNet [79] with an architecture formed of only 8 layers, while the more recent ResNet model [68] reached a top-5 error of 3.6% with 152 layers. While the CNN architecture has evolved over the last few years to accommodate more convolutional layers, to reduce the size of the filters, and even to eliminate the fully-connected layers, comparably less attention has been paid to improving the training process. An important limitation of the state-of-the-art neural models mentioned above is that examples are considered in a random order during training. Indeed, the training is usually performed with some variant of mini-batch stochastic gradient descent, the examples in each mini-batch being chosen randomly.

Since neural network architectures are inspired by the human brain, it seems reasonable to consider that the learning process should also be inspired by how humans learn. One essential difference from how machines are typically trained is that humans learn the basic (easy) concepts sooner and the advanced (hard) concepts later. This is basically reflected in all the curricula taught in schooling systems around the world, as humans learn much better when the examples are not randomly presented but are organized in a meaningful order. Using a similar strategy for training a machine learning model, we can achieve two important benefits: (i) an increase of the convergence speed of the training process and (ii) a better accuracy. A preliminary study in this direction has been conducted by Elman [80]. To our knowledge, Bengio et al. [81] are the first to formalize the easy-to-hard training strategies in the context of machine learning, proposing the CL paradigm. This seminal work inspired many researchers to pursue curriculum learning strategies in various application domains, such as weakly supervised object localization [82–84], object detection [85–88] and neural machine translation [89–92] among many others. The empirical results presented in these works show the clear benefits of replacing the conventional training based on random





mini-batch sampling with curriculum learning. Despite the consistent success of curriculum learning across several domains, this training strategy has not been adopted in mainstream works. This fact motivated us to write this survey on curriculum learning methods in order to increase the popularity of such methods. On another note, researchers proposed opposing strategies emphasizing harder examples, such as Hard Example Mining (HEM) [93–96] or anti-curriculum [97,98], showing improved results in certain conditions.



(a) General framework for data-level curriculum learning. (b) General framework for model-level curriculum.

Figure 9. General frameworks for data-level and model-level curriculum learning, side by side. In both cases,  $k$  is some positive integer. Best viewed in color.

Figure 9 illustrates the general frameworks for curriculum learning applied at the data level and at the model level, respectively. The two frameworks have two common elements: the curriculum scheduler and the performance measure. The scheduler is responsible for deciding when to update the curriculum in order to use the pace that gives the highest overall performance. Depending on the applied methodology, the scheduler may consider a linear pace or a logarithmic pace. Additionally, in self-paced learning, the scheduler can take into consideration the current performance level to find the right pace. When applying CL over data (see Figure 9a), a difficulty criterion is employed in order to rank the examples from easy-to-hard. Next, a selection method determines which examples should be used for training at the current time. Curriculum over tasks works in a very similar way. In Figure 9b, we observe that CL at model level does not require a difficulty criterion. Instead, it requires the existence of a model capacity curriculum. This sets how to change the architecture or the parameters of the model to which all the training data is fed.

On another note, we remark that continuation methods can be seen as curriculum learning performed over the performance measure  $P$  [99]. However, this connection is not typically mentioned in literature. Moreover, continuation methods [100–102] were studied long before curriculum learning appeared [81]. Research on continuation methods is therefore considered an independent field of





study [100,102], not being necessarily bounded to its applications in machine learning [101], as would be the case for curriculum learning. In this context, our survey on curriculum learning does not include continuation methods.

#### 4.6.1. Contributions

Our first contribution is to formalize the existing curriculum learning methods under a single umbrella. This enables us to define a generic formulation of curriculum learning. Additionally, we link curriculum learning with the four main components of any machine learning approach: the data, the model, the task and the performance measure. We observe that curriculum learning can be applied on each of these components, all these forms of curriculum having a joint interpretation linked to loss function smoothing. Furthermore, we manually create a taxonomy of curriculum learning methods, considering orthogonal perspectives for grouping the methods: data type, task, curriculum strategy, ranking criterion and curriculum schedule. We corroborate the manually constructed taxonomy with an automatically built hierarchical tree of curriculum methods. In large part, the hierarchical tree confirms our taxonomy, although it also offers some new perspectives. While gathering works on curriculum learning and defining a taxonomy on curriculum learning methods, our survey is also aimed at showing the advantages of curriculum learning. Hence, our final contribution is to advocate the adoption of curriculum learning in mainstream works.

We are not the first to consider providing a comprehensive analysis of the methods employing curriculum learning in different applications. Recently, Narkevar et al. [103] survey the use of curriculum learning applied to reinforcement learning. They present a new framework and use it to survey and classify the existing methods in terms of their assumptions, capabilities and goals. They also investigate the open problems and suggest directions for curriculum RL research. While their survey is related to ours, it is clearly focused on RL research and, as such, is less general than ours. Directly relevant to our work is the recent survey of Wang et al. [104] (available only as a preprint at the moment). Their aim is similar to ours as they cover various aspects of curriculum learning including motivations, definitions, theories and several potential applications. We are looking at curriculum learning from a different view point and propose a generic formulation of curriculum learning. We also corroborate the automatically built hierarchical tree of curriculum methods with the manually constructed taxonomy, allowing us to see curriculum learning from a new perspective. Furthermore, our review is more comprehensive, comprising nearly 180 scientific works. We strongly believe that having multiple surveys on the field will strengthen the focus and bring about the adoption of CL approaches in the mainstream research.

#### 4.6.2. Relevant publications

- P. Soviany, R. Ionescu, P. Rota and N. Sebe, Curriculum Learning: A Survey, International Journal of Computer Vision, 130(6):1526-1565, June 2022. [105].  
Zenodo record: <https://zenodo.org/record/7100343>.

### 4.7. Class-incremental Novel Class Discovery

**Contributing partner(s):** UNITN

#### 4.7.1. Introduction and methodology

Humans are bestowed with the excellent cognitive skills to learn continually over their lifetime [106], and in most cases without the need of explicit supervision [107]. Thus, it has been a long-standing



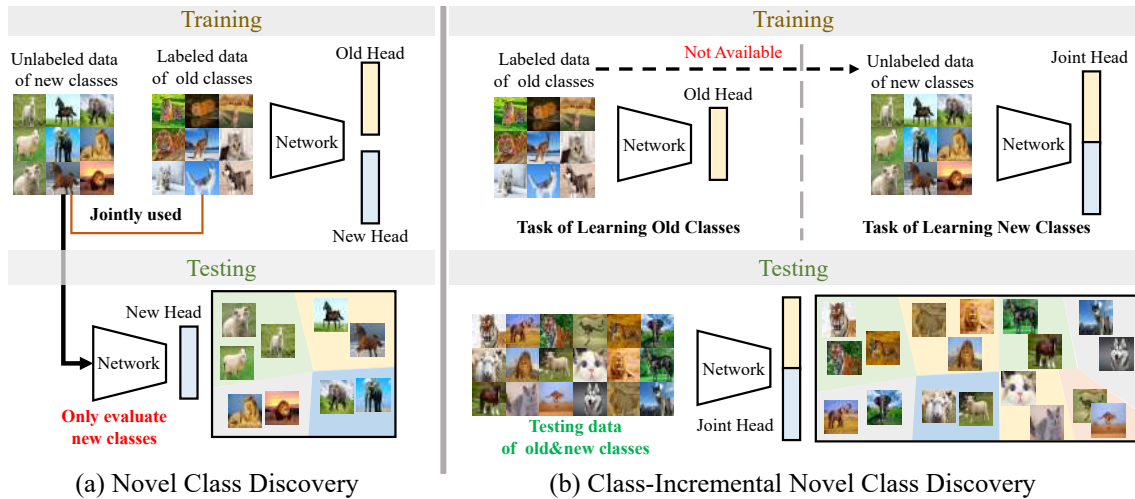


Figure 10. Comparison between the settings (a) NCD which solely concerns the performance of novel classes, and (b) the proposed class-incremental NCD (class-iNCD) that measures performance of all the classes seen so far with a single classifier.

goal of the machine learning research community to build AI systems that can mimic this human-level performance. In an attempt to realize this, much effort has been dedicated to learn deep learning models from large reservoirs of both labelled [108–110] and unlabelled data [111, 112]. Aside from being effective learners, by imitating human learning mechanisms, neural networks should also be flexible to absorb novel concepts (or *classes*) after having learned some patterns with the past data. The task of automatically discovering novel (or *new*) classes in an unsupervised fashion while leveraging some previously learned knowledge is referred to as *novel class discovery* (NCD) [113–117] (see Figure 10(a)). NCD has gained significant attention in the recent times due to its practicality of efficiently learning novel classes without relying on large quantities of unlabelled data [113].

Most of the proposed NCD solutions rely on stage-wise [114, 118, 119] or joint [113, 115, 117] learning schemes on the labelled and the unlabelled data, with the assumption that structures discovered on the labelled images could be leveraged as a proxy supervision on the unlabelled images. It has been shown that NCD benefits more when the model is trained jointly on the labelled data while using a clustering objective on the unlabelled data [113, 115–117]. However, access to the labelled data after the pre-training stage can not always be guaranteed in real-world applications due to privacy or storage issues. This calls for a more pragmatic NCD setting where the labelled images would be discarded and only the pre-trained model could be transferred for learning the novel classes. Being meaningful, such *source-free* model adaptation has been explored in the related areas of domain adaptation [120, 121]. Although it seems more practical, such a training scheme would gradually cause the network to erase all the previously learned information about the old (or *base*) classes. This drop in the base class performance when the labelled data set becomes unavailable is primarily attributed to the phenomenon of *catastrophic forgetting* [122] in neural networks. In most of the aforementioned NCD methods the performance on the novel classes are only deemed important, without any consideration for preserving the performance on the base classes. We believe that such a setting is of little practical significance in the real world because the adapted model becomes unusable on the base classes and retraining is infeasible.

Given the inherent drawbacks of the existing NCD setting, we argue that an ideal NCD method



Table 9. Comparison with state-of-the-art methods in class-iNCD.

Methods	CIFAR-10			CIFAR-100			Tiny-ImageNet			Average		
	Old	New	All	Old	New	All	Old	New	All	Old	New	All
AutoNovel [113]	27.5	3.5	15.5	2.6	15.2	5.1	2.0	26.4	4.5	10.7	15.0	8.4
ResTune [123]	91.7	0.0	45.9	<b>73.8</b>	0.0	59.0	44.3	0.0	39.9	<b>69.9</b>	0.0	48.3
NCL [115]	<b>92.0</b>	1.1	46.5	73.6	10.1	<b>60.9</b>	0.8	6.5	1.4	55.5	5.9	36.3
DTC [130]	64.0	0.0	32.0	55.9	0.0	44.7	35.5	0.0	32.0	51.8	0.0	36.2
<b>FROST</b>	77.5	<b>49.5</b>	<b>63.4</b>	64.6	<b>45.8</b>	59.2	<b>54.5</b>	<b>33.7</b>	<b>52.3</b>	65.5	<b>39.8</b>	<b>54.9</b>

should aim to learn novel classes without the explicit presence of the labelled data and at the same time preserve the performance on the base classes. This new setting is referred to as *task-incremental* NCD (iNCD), and indeed has been very recently studied in [123]. In details, ResTune [123] uses knowledge distillation [124] on the network logits to prevent forgetting on the base classes and a clustering objective [125] with task specific network weights for the novel classes. As opposed to the ResTune [123], which facilitates iNCD by solely improving the ability of the network to learn novel classes, we additionally improve the incremental learning aspect in iNCD as well. Specifically, inspired by the rehearsal-based incremental learning methods [126–128] which are known to be effective, we propose to store the base class feature *prototypes* from the previous task as exemplars, instead of raw images. Features derived from the stored prototypes are then *replayed* to prevent forgetting old information on the base classes in addition to feature-level knowledge distillation. On the other hand, to facilitate learning of novel classes, we dedicate a task specific classifier that is optimized with robust rank statistics [113]. Disadvantageously, the introduction of task specific classifier leads to the dependence on the task-id of an input sample during inference. To overcome reliance on task-id, we propose to maintain a joint classifier for both the base and novel classes, which is trained with the pseudo-labels generated by the task specific one. We call this setting as *class-incremental* NCD (class-iNCD) as it does not allow the task-id information to be used during inference. The high level overview of the new class-iNCD setting is shown in Figure 10(b). As our proposed method amalgamates **F**eature **R**eplay and **D**istillation with **S**elf-**T**raining, we name it FROST.

#### 4.7.2. Experimental results

**4.7.2.1. Experimental Setup Datasets.** We have used three data sets to conduct experiments for class-iNCD: CIFAR-10 [129], CIFAR-100 [129] and Tiny-ImageNet [38]. We split the data sets into the old and new classes following the existing NCD and iNCD works [113, 115, 123].

**Evaluation metrics.** We used our new evaluation protocol to evaluate the performance on the test data for all the classes. We report three classification accuracies, denoted as **Old**, **New** and **All**. They represent the accuracy obtained from the joint classifier head on the samples of the old, new and old+new classes, respectively.

**Implementation details** We used ResNet-18 [109] as the backbone in all the experiments. We have adopted most of the hyperparameters from AutoNovel [113]. We introduce only one additional hyperparameter  $\lambda$ , which is set to 10.

**4.7.2.2. Comparison with State-of-the-art Methods** We compare our FROST with the state-of-the-art NCD methods under the newly proposed class-iNCD setting. We also compare with ResTune [123] which is a recently proposed method for iNCD. As none of these existing methods have been evaluated in the class-iNCD setting, we re-run the baselines and simply modify our evaluation protocol. We report the results of the NCD [113–115], iNCD [123] baselines and FROST in Table 9. As can be observed, under the class-iNCD all the NCD [113–115] fail to obtain a good



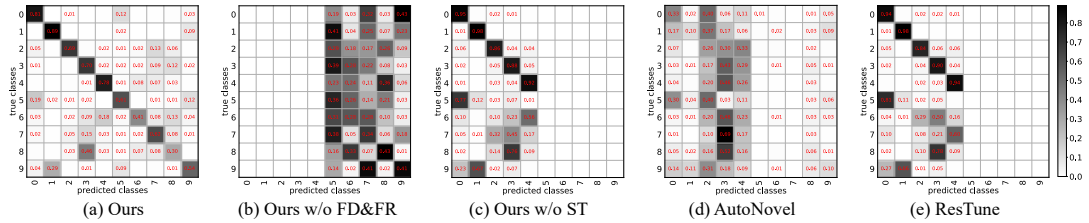


Figure 11. Comparisons of confusion matrix of different methods. Note that, the label IDs of novel classes are re-assigned by our evaluation protocol.

Table 10. Comparison with the state-of-the-art methods in the two-step class-iNCD setting where new classes arrive in two episodes, instead of one. New-1-J: new classes performance from joint head at first step, New-1-N: new classes performance from novel head at first step, etc.

Methods	Tiny-ImageNet									
	First Step (180-10)				Second Step (180-10-10)					
	Old	New-1-J	New-1-N	All	Old	New-1-J	New-2-J	New-1-N	New-2-N	All
ResTune [123]	39.7	0.0	38.0	37.6	34.9	0.0	0.0	25.4	42.8	31.4
DTC [130]	38.9	0.0	<b>43.8</b>	36.9	33.4	0.0	0.0	28.0	<b>59.4</b>	30.1
NCL [115]	5.6	0.0	34.2	5.3	1.4	0.0	2.6	21.6	41.6	1.4
<b>FROST</b>	<b>55.2</b>	<b>27.6</b>	32.0	<b>53.8</b>	<b>42.5</b>	<b>34.8</b>	<b>31.2</b>	<b>31.2</b>	46.8	<b>41.6</b>

balance on the old and new classes. Interestingly, while none of these NCD methods use any explicit objectives to prevent forgetting, they tend to predict well the old classes (see column **Old** in Table 9) and poor performance on new classes (see column **New** in Table 9). When visualizing the confusion matrix in Figure 11, we found that most of the test samples get classified as old classes due to the old classes classifier having higher norms. As a consequence, this gives the impression that the baselines methods are able to retain performance on old classes. Second, for the above methods, although the new classes performance obtained with the joint head appears to be low, the actual performance of their novel head in the task-aware evaluation is indeed high. We report the breakdown of the novel classes performance in Table 10 where, for instance, the column **New-1-N** denotes the task-aware clustering performance of the novel head on the new classes. As can be observed, the new classes classifier of the NCD baselines can indeed learn on the new classes (*e.g.*, 34.2% in NCL vs 32.4% in FROST).

ResTune, although designed specifically for the iNCD setting, exhibits similar counter-intuitive behaviour with the performance on the old classes dominating the new classes. To investigate this pathology, we inspect into the confusion matrix in Figure 11 (e) and find that all the samples get predicted to the first five old classes for CIFAR10. In other words, the overall performance reported in ResTune [123] is actually dominated by the old classes performance. We report confusion matrices on bigger data sets in the supplementary material. This shows that the existing evaluation method for iNCD is flawed and our proposed class-iNCD is indeed more meaningful that properly evaluates the effectiveness of a learning algorithm. Contrarily, our proposed FROST consistently achieves a good balance in performance in all the tested data sets. This also demonstrates the validity of the components in our proposed FROST.

**Two-Step Class-iNCD.** As done in the class-IL literature [131], we also run experiments on a sequence of novel tasks, which we call as two-step class-iNCD, where 20 novel classes in Tiny-ImageNet are added in two steps, each step dealing with 10 novel classes. We compare our FROST with the baseline methods in Table 10 where we show not only the joint classifier head performance



(*e.g.*, **New-1-J**), but also from the novel classifier head (*e.g.*, **New-1-N** and **New-2-N**) at each step. As can be seen, for the baseline methods the novel classifier heads can satisfactorily discover the new classes at each step, but when evaluated with the joint head biases the predictions to the old classes. Unlike the baselines, FRoST does not suffer from this issue and leads to more balanced predictions.

#### 4.7.3. Conclusions

In summary, the contributions of this work are three-fold:

- We propose a novel framework, FRoST, that can tackle the newly introduced and relevant task of class-incremental novel class discovery (class-iNCD).
- Our FRoST is equipped with prototypes for feature-replay and employs feature-level knowledge distillation to prevent forgetting. Moreover, it uses pseudo-labels from the task specific head to efficiently learn novel classes without interference, enabling us to achieve a task-agnostic classifier.
- We run extensive experiments on three common benchmarks to prove the effectiveness of our method. FRoST also obtains state-of-the-art performance when compared with the existing baselines. Additionally, we run experiments on a sequence of tasks of unlabelled sets and verify its generality.

#### 4.7.4. Relevant publications

- S. Roy, M. Liu, Z. Zhong, N. Sebe, and E Ricci, Class-incremental Novel Class Discovery, European Conference on Computer Vision (ECCV'22) [132].  
Zenodo record: <https://zenodo.org/record/7566121>.

#### 4.7.5. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in <https://github.com/OatmealLiu/class-iNCD>.

#### 4.7.6. Relevance to AI4media use cases and media industry applications

We have presented NCD as a generic approach that allows neural networks the flexibility to absorb novel concepts (or classes) after having learned some patterns with the past data. We have discussed in the section the application on image analysis so the approach could be directly relevant to use cases (a) 3A3 (archive exploration), specifically 3A3-11 Visual indexing and search and (b) 7A3 (Re)organisation of visual content by supporting the efficient training and organization of image and video collections. However, the approach can also be applied when other modalities are involved, *e.g.*, 4C3 (audio analysis).

## 4.8. CoReS: Learning Compatible Representations via Stationarity

**Contributing partner(s):** UNIFI





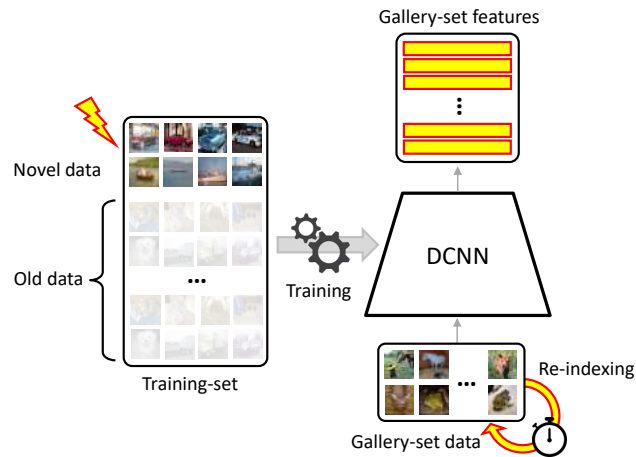


Figure 12. Upgrading the DCNN representation model with novel data, typically requires the gallery-set to be re-indexed. Learning compatible representations allows to compare the newly learned representation of an input query-set with the old representation of the gallery-set, thus eliminating its computationally intensive re-indexing.

#### 4.8.1. Introduction and methodology

Natural intelligent systems learn from visual experience and seamlessly exploit such learned knowledge to identify similar entities. Modern AI systems, on their turn, typically require distinct phases to perform such visual search. An internal representation is first learned from a set of images (the *training-set*) using Deep Convolutional Neural Network (DCNN) models [35, 133–135] and then used to index a large corpus of images (the *gallery-set*). Finally, visual search is obtained by identifying the closest images in the gallery-set to an input *query-set* by comparing their representations. Successful applications of learning feature representations are: face-recognition [136–140], person re-identification [141–144], image retrieval [145–147], and car re-identification [148] among others.

In the case in which novel data for the training-set and/or more recent or powerful network architectures become available, the representation model may require to be *upgraded* to improve its search capabilities. In this case, not only the query-set but also all the images in the gallery-set should be re-processed by the upgraded model to generate new features and replace the old ones to benefit from such upgrading. The re-processing of the gallery-set is referred to as *re-indexing* (Figure 12).

For visual search systems with a large gallery-set, such as in surveillance systems, social networks or in autonomous robotics, re-indexing is clearly computationally expensive [149] or has critical deployment, especially when the working system requires multiple upgrades or there are real-time constraints. Re-indexing all the images in the gallery-set can be also infeasible when, due to privacy or ethical concerns, the original gallery images cannot be permanently stored [150] and the only viable solution is to continue using the feature vectors previously computed. In all these cases, it should be possible to directly compare the upgraded features of the query with the previously learned features of the gallery, i.e., the new representation should be *compatible* with the previously learned representation.

Learning compatible representation has recently received increasing attention and novel methods have been proposed in [151–156]. Differently from these works, in this work we address *compatibility* leveraging the *stationarity* of the learned internal representation. Stationarity allows to maintain

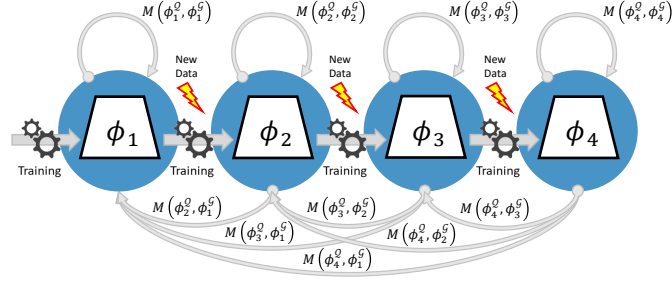


Figure 13. Multi-model Empirical Compatibility Criterion (Eq. 7): representation models  $\phi_i$  with  $i = 1, 2, \dots, T$  are sequentially trained. Gray arrows represent self and cross-tests (example with  $T = 4$ ).

the same distribution of the features over time so that it is possible to compare the features of the upgraded representation with those previously learned. In particular, we enforce stationarity by leveraging the properties of a family of classifiers whose parameters are not subject to learning, namely *fixed classifiers* based on regular polytopes [157–159], that allow to reserve regions of the representation space to future classes while classes already learned remain in the same spatial configuration.

**Compatibility Evaluation.** In [149], a general criterion to evaluate compatibility was defined, i.e., the *Empirical Compatibility Criterion*:

$$M(\phi_{\text{new}}^Q, \phi_{\text{old}}^G) > M(\phi_{\text{old}}^Q, \phi_{\text{old}}^G), \quad (5)$$

where  $M$  is a metric used to evaluate the performance based on  $\text{dist}(\cdot, \cdot)$ . The notation  $M(\phi_{\text{new}}^Q, \phi_{\text{old}}^G)$  underlines that the upgraded model  $\phi_{\text{new}}$  is used to extract feature vectors  $F_Q$  from query images  $I_Q$ , while the old model  $\phi_{\text{old}}$  is used to extract features  $F_G$  from gallery images  $I_G$ . This performance value is referred to as *cross-test*. Correspondingly,  $M(\phi_{\text{old}}^Q, \phi_{\text{old}}^G)$  evaluates the case in which both query and gallery features are extracted with  $\phi_{\text{old}}$  and is referred to as *self-test*.

In real world applications, multi-step upgrading is often required, i.e., different representation models must be sequentially learned through time, in multiple upgrade steps. At each step  $t$ , the training-set is upgraded as:

$$\mathcal{T}_t = \mathcal{T}_{t-1} \cup \mathcal{X}_t \quad (6)$$

being  $\mathcal{X}_t$  the new data and  $\mathcal{T}_{t-1}$  the training-set at step  $t - 1$ . In the multi-step upgrading case, we define the following *Multi-model Empirical Compatibility Criterion* as follows:

$$M(\phi_{t'}^Q, \phi_t^G) > M(\phi_t^Q, \phi_t^G) \quad \forall t' > t$$

with  $t' \in \{2, 3, \dots, T\}$  and  $t \in \{1, 2, \dots, T - 1\}$ , (7)

where  $\phi_{t'}$  and  $\phi_t$  are two different models such that  $\phi_t$  is upgraded before  $\phi_{t'}$ ,  $T$  is the number of upgrade steps and  $M$  the metric used to evaluate the performance. Model  $\phi_{t'}$  is compatible with  $\phi_t$  when their cross-test is greater than the self-test of  $\phi_t$  for each pair of upgrade steps. Figure 13 illustrates the Multi-model Empirical Compatibility Criterion, where  $\{\phi_1, \phi_2, \dots, \phi_T\}$  are the representation models, black arrows indicate the model upgrades and gray arrows represent self and cross-tests.

In order to assess multi-model compatibility of Eq. 7 for a sequence of  $T$  upgrade steps, we

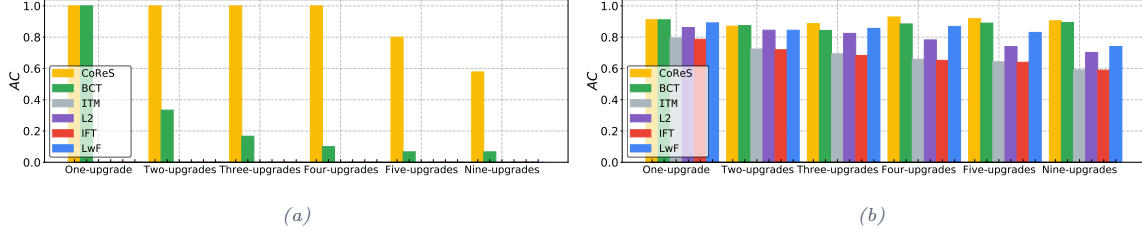


Figure 14. Compatibility of CoReS and compared methods (shown color-coded) for open-set face verification on the CASIA-WebFace/LFW dataset with multi-model upgrading. Bins show: (a) AC scores for different number of upgrades; (b) AM scores for different number of upgrades.

define the following square triangular *Compatibility Matrix*  $C$ :

$$C = \begin{pmatrix} M(\phi_1^Q, \phi_1^G) & & & & \\ M(\phi_2^Q, \phi_1^G) & M(\phi_2^Q, \phi_2^G) & & & \\ \vdots & \vdots & \ddots & & \\ M(\phi_T^Q, \phi_1^G) & M(\phi_T^Q, \phi_2^G) & \cdots & M(\phi_T^Q, \phi_T^G) & \end{pmatrix} \quad (8)$$

where each entry  $C_{ij}$  is the performance value according to metric  $M$ , taking model  $\phi_i$  for the query-set  $\mathcal{Q}$  and model  $\phi_j$  for the gallery-set  $\mathcal{G}$ . Entries on the main diagonal,  $i = j$ , represent the self-tests, while the entries off-diagonal,  $i > j$ , represent the cross-tests. While showing compatibility performance across multiple upgrade steps, matrix  $C$  can be used to provide a scalar metric to quantify the global multi-model compatibility in a sequence of upgrade steps. In particular, we define the *Average Multi-model Compatibility* (AC) as the number of times that Eq. 7 is verified with respect to all its possible occurrences, independently of the number of the learning steps:

$$AC = \frac{2}{T(T-1)} \sum_{i=2}^T \sum_{j=1}^{i-1} \mathbb{1}(C_{ij} > C_{jj}), \quad (9)$$

where  $\mathbb{1}(\cdot)$  denotes the indicator function.

Finally, we define the *Average Multi-model Accuracy* (AM) as the average of the entries of the Compatibility Matrix:

$$AM = \frac{2}{T(T+1)} \sum_{i=1}^T \sum_{j=1}^i C_{ij} \quad (10)$$

to provide an aggregate value of the accuracy metric  $M$  under compatible training.

#### 4.8.2. Experimental Results

We performed open-set face verification using the CASIA-WebFace dataset to create the training-sets and LFW as the test set. The CASIA-WebFace dataset includes 494,414 RGB face images of 10,575 subjects. The LFW dataset contains 13,233 target face images of 5,749 subjects. Of these, 1,680 have two or more images, while the remaining 4,069 have only one image. ResNet50 [160] with input size of  $112 \times 112$  is used as backbone. Optimization is performed using SGD with 0.1 learning rate, 0.9 momentum, and  $5 \cdot 10^{-4}$  weight decay. The batch size is 1,024. With every upgrade, training is terminated after 120 epochs. Learning rate is scheduled to decrease to 0.01, 0.001, and 0.0001 at epoch 30, 60, 90 respectively.



Compatibility is evaluated for one, two, three, four, five, and nine upgrade steps.

In the one-upgrade case, models are learned with 50% of the CASIA-WebFace dataset and upgraded with 100%. Figure 14a shows that, with 50% of CASIA-WebFace dataset, CoReS and BCT achieve similar performance. This is because there is already sufficient data variability to learn compatible features. A big difference between CoReS and compared methods becomes evident when multiple upgrades are considered as shown in Figure 14a. In this figure, the values of the  $AC$  are reported for each method over a set of different experiment respectively with one, two, three, four, five, and nine upgrades. CoReS achieves full compatibility ( $AC = 1$ ) for one, two, three, and four upgrades and starts decreasing  $AC$  from five upgrade steps up to  $AC = 0.58$  with nine upgrades. In contrast BCT loses performance already with two upgrade steps finishing at  $AC = 0.09$  with nine upgrade steps. Baseline methods report  $AC = 0$  in all of the scenario. In Figure 14b, we report the  $AM$  metric for these experiments. It can be observed that the CoReS and BCT score almost the same average verification accuracy in all the experiments, while in the others values are always lower. This is due to the fact that cross-test values are low since no compatibility is reported.

We conclude that for multi-model upgrading, CoReS, while having the same verification performance as BCT, largely improves compatibility across model upgrades with 544% relative improvement over BCT for the challenging scenario of nine-step upgrading. The lower  $AC$  of BCT appears to be related to the fact that in this method compatibility is obtained only through transitivity from the model previously learned.

### 4.8.3. Conclusions

The main contributions of our research are the following:

- We identify stationarity as a key property for compatibility and propose a novel training procedure for learning compatible feature representations via stationarity, without the need of learning any mappings between representations nor to impose pairwise training with the previously learned model. We called our method: Compatible Representations via Stationarity (CoReS).
- We introduce new criteria for comparing and evaluating compatible representations in the case of sequential multi-model upgrading.
- We demonstrate through extensive evaluation on large scale verification, re-identification and retrieval benchmarks that CoReS improves the current state-of-the-art in learning compatible features for both single and sequential multi-model upgrading.

### 4.8.4. Relevant publications

- Biondi, Niccolo, Federico Pernici, Matteo Bruni, and Alberto Del Bimbo. "Cores: Compatible representations via stationarity." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).  
Zenodo record: <https://zenodo.org/record/7913176>.

### 4.8.5. Relevant software/datasets/other outcomes

- The PyTorch implementation of our work "CoReS: Compatible Representations via Stationarity" can be found in:  
<https://github.com/NiccoBiondi/cores-compatibility>.





#### 4.8.6. Relevance to AI4media use cases and media industry applications

CoReS is a tool that learns semantic representations of data, aiding media professionals in retrieving information based on semantic content. It can be integrated with the “AI for News” use case (UC2) and “AI for Vision” use case (UC3) to enhance tagging and search functions. In dynamic environments, where journalistic content and deep neural network-based recognition models are frequently updated, CoReS offers a distinctive advantage. It produces semantic representations of data that remain compatible with previous versions, eliminating the need for database reprocessing. This approach ensures quick and smooth news retrieval even as new recognition models come into play. Conversely, in many standard retrieval applications, when a more advanced recognition model becomes available, the existing semantic content in the database cannot be used directly. This often necessitates reprocessing and results in a computationally intensive re-indexing process. Moreover, CoReS can provide essential support to UC7 “AI for Content Organization and Content Moderation” and UC1 “AI against Disinformation” as it embodies foundational principles for content analysis, ensuring fast and accurate semantic understanding of updated news.

### 4.9. CL<sup>2</sup>R: Compatible Lifelong Learning Representations

**Contributing partner(s):** UNIFI

#### 4.9.1. Introduction and methodology

The universe is dynamic and the emergence of novel data and new knowledge is unavoidable. The unique ability of *natural intelligence* to lifelong learning is highly dependent on memory and knowledge representation [161]. Through memory and knowledge representation, natural intelligent systems continually search, recognize, and learn new objects in an open universe after exposure to one or a few samples. Memory is substantially a cognitive function that encodes, stores, and retrieves knowledge. Artificial representations learned by Convolutional Neural Network (CNN) models [35, 134, 135, 138, 139] stored in a memory bank (i.e., the gallery-set) have been shown to be very effective in searching and recognizing objects in an open-set/open-world learning context. Successful examples are face recognition [133, 140, 162], person re-identification [163–165] and image retrieval [166–168]. These approaches rely on learning feature representations from static datasets in which all images are accessible at training time. On the other hand, dynamic assimilation of new data for lifelong learning suffers from *catastrophic forgetting*: the tendency of neural networks to abruptly forget previously learned information [169, 170].

In the case of visual search, even avoiding catastrophic forgetting by repeatedly training DCNN models on both old and new data, the feature representation still irreversibly *changes* [171]. Thus, in order to benefit from the newly learned model, features stored in the gallery must be reprocessed and the “old” features replaced with the “new” ones. Reprocessing not only requires the storage of the original images (a noticeable leap from natural intelligence), but also their authorization to access them [150]. More importantly, extracting new features at each update of the model is computationally expensive or infeasible in the case of large gallery-sets. The speed at which the representation is updated to benefit from the newly learned data may impose time constraints on the re-indexing process. This may occur from timescales on the order of weeks/months as in retrieval systems or social networks [151], to within seconds as in autonomous robotics or real-time surveillance [172, 173]. Recently in [151], a novel training procedure has been proposed to avoid re-indexing the gallery-set. The representation obtained in this manner is said to be compatible, as the features before and after the learning upgrade can be directly compared. Training takes



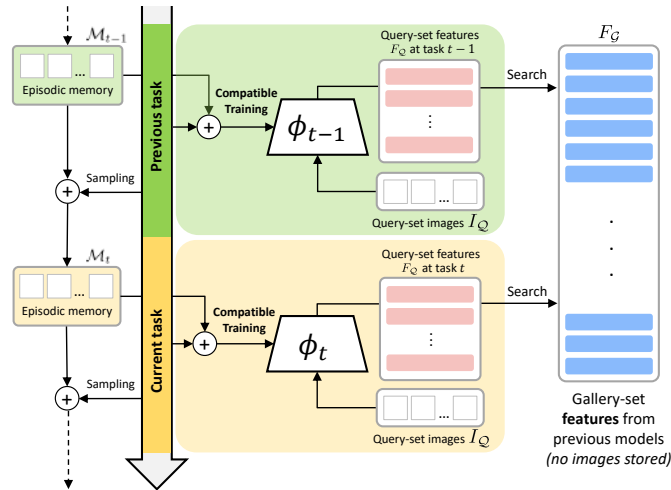


Figure 15. Overview of the Compatible Lifelong Learning Representations ( $CL^2R$ ) problem and proposed training procedure. The learning agent searches object instances from query images  $I_Q$  without re-indexing the gallery-set. Any update to the internal feature representation  $\phi$  does not render the features in the gallery-set unusable (i.e., no images are stored). Compatible feature representation under catastrophic forgetting is learned imposing stationarity to features learned from the the CIL surrogate task. Training is based on rehearsal with the episodic memory  $\mathcal{M}_t$ .

advantage of all the data from previous tasks (i.e., no lifelong learning), guaranteeing the absence of catastrophic forgetting. The advantage of considering compatible representation learning within the lifelong learning paradigm, as in this paper, is that compatible representation allows visual search systems not only to distribute the computation over time, but also to avoid or possibly limit the storage of images on private servers for gallery data. This can have important implications for the societal debate related to privacy, ethical and sustainable issues (e.g., carbon footprint) of modern AI systems [150, 174–176].

We identify *stationarity* as the key requirement for feature representation to be *compatible* during lifelong learning. Stationary features have been shown to be biologically plausible in many studies of working memory in the prefrontal cortex of macaques [177–179]. The works [177, 178] decoded the information from the neural activity of the working memory using a classifier with a single *fixed set* of weights. They noted that a *non-stationary* feature representation seems to be biologically problematic since it would imply that the synaptic weights would have to change continuously for the information to be continuously available in memory.

Inspired by this, in this paper, we formalize the problem of Compatible Lifelong Learning Representations ( $CL^2R$ ) in relation to the relevant areas of *compatible learning* and *lifelong (continual) learning*. We call any training procedure that aims to obtain compatible features and minimize catastrophic forgetting as  $CL^2R$  training, and we propose (1) a novel set of metrics to properly evaluate  $CL^2R$  training procedures (2) a training procedure based on rehearsal [170, 180] and feature stationarity [157, 181] to jointly address catastrophic forgetting and feature compatibility. Figure 15 provides an overview of the problem and the training procedure. Specifically, our  $CL^2R$  training procedure is achieved by encouraging *global* and *local* stationarity to the learned features.

**Proposed  $CL^2R$  Metrics** The work in [182, 183] proposes a set of metrics to assess the ability of the learner to transfer knowledge based on a matrix that reports the test classification accuracy of the model on task  $j$  after learning task  $i$ . Along a similar vein, we present a set of metrics to



evaluate the compatibility between representation models in a compatible lifelong learning setting.

Let  $C \in \mathbb{R}^{T \times T}$  be the compatibility matrix of the Eq. 8 for  $T$  tasks, the proposed criteria are the following: (1) *Backward Compatibility (BC)* measures the gap in compatibility performance between the representation learned at task  $T$  with respect to the representation learned at task  $k$  with  $k \in \{1, \dots, T - 1\}$ . When  $BC < 0$  the learning procedure is also influenced by catastrophic forgetting because the performance degrades with newer learned tasks.  $BC$  is defined as follows:

$$BC = \frac{1}{T-1} \sum_{k=1}^{T-1} (C_{T,k} - C_{k,k}) \quad (11)$$

(2) *Forward Compatibility (FC)* estimates the influence that learning a representation on a task  $k - 1$  has on the compatibility performance of the representation learned on a future task  $k$  by comparing the cross-test (between models at task  $k$  and  $k - 1$ ) with respect to the self-test at task  $k$ .  $FC \geq 0$  denotes that, on average, the cross-test values are greater than the self-test evaluated on the subsequent tasks, therefore, re-indexing does not necessarily provide improved results.  $FC$  is defined as follows:

$$FC = \frac{1}{T-1} \sum_{k=2}^T (C_{k,k-1} - C_{k,k}). \quad (12)$$

The intuition behind the definition of this metric comes from noticing that as the number of tasks increases, the *cross-test* may result better than the *self-test*. As this is not typically observed when there is no catastrophic forgetting (i.e., when repeatedly training with new and old data), we argue this is due to the *joint interaction* between the compatibility constraint and catastrophic forgetting. This observation led us to define something “positive” when the compatible representation with the previously learned model is higher than the self-test of the current model. This metric is designed to yield high values when a CL<sup>2</sup>R training procedure is able to positively exploit the joint interaction between feature forgetting and compatible representation.

From Eqs. 11 and 12, it can be deduced that  $BC$  and  $FC \in [-1, 1]$ . Backward compatibility for the first task and forward compatibility for the last task are not defined. The larger these metrics, the better the model. When  $AC$  values are comparable, both  $BC$  and  $FC$  represent two metrics that quantify the positive interaction between search accuracy under catastrophic forgetting and compatibility. This allows evaluating how catastrophic forgetting affects the representation and its compatibility.

#### 4.9.2. Experimental Results

In this section, we report the experiments in two, three, five, and ten tasks CL<sup>2</sup>R settings with models trained on CIFAR100 (i.e., using 50, 33, 20, 10 classes per task) where compatibility is evaluated on the CIFAR10 generated pairs.

In Table 12, we summarize the performance of our CL<sup>2</sup>R training procedure with respect to the other baselines in the two-task scenario. We evaluate the compatibility of the updated model according to the ECC (Eq. 5),  $BC$  (Eq. 11), and  $FC$  (Eq. 12). The first row of Table 12 reports the verification accuracy of the model trained on the first 50 classes of CIFAR100. Experiments show that, among the methods compared, LUCIR and PODNet may have an inherent, although limited, level of compatible representations. This substantially confirms the importance of having some form of mechanism to preserve the local geometry of the learned features. Our training procedure achieves the highest cross test,  $BC$ , and  $FC$ , thus resulting to be the most suited training procedure to avoid re-indexing.

In the last rows of the table, we report the performance of the BCT and our Upper Bound (UB) that are not affected by catastrophic forgetting. The effect of catastrophic forgetting and its



Table 11. Evaluation of CIFAR10. Three, five, and ten-task  $CL^2R$  setting with models trained on CIFAR100. We report  $AC$  (Eq. 4),  $BC$  (Eq. 5), and  $FC$  (Eq. 6) for the methods we evaluated.

METHOD	THREE TASKS			FIVE TASKS			TEN TASKS		
	$AC$	$BC$	$FC$	$AC$	$BC$	$FC$	$AC$	$BC$	$FC$
ER	0	-0.070	-0.080	0.12	-0.098	-0.087	0.04	-0.130	-0.083
LwF	<u>0.33</u>	-0.044	0.000	0.13	-0.051	-0.035	0.04	-0.037	-0.036
BiC	<b>0.67</b>	-0.040	-0.040	0.14	-0.092	-0.023	0.11	-0.060	-0.018
LUCIR	<u>0.33</u>	-0.039	<b>0.030</b>	<u>0.36</u>	<u>-0.015</u>	<u>0.005</u>	0.20	-0.044	0.001
FAN	<b>0.67</b>	<u>-0.010</u>	-0.026	<u>0.36</u>	-0.055	-0.039	0.11	-0.160	<b>0.067</b>
FOSTER	0	-0.149	-0.132	0	-0.072	-0.077	0.04	-0.098	-0.105
$\ell$ -BCT	0.07	-0.095	-0.014	0	-0.083	-0.043	0.07	-0.064	-0.030
PODNet	<b>0.67</b>	-0.015	-0.026	0.30	-0.025	-0.006	<u>0.27</u>	<u>-0.032</u>	-0.002
<b>Ours</b>	<b>0.67</b>	<b>-0.007</b>	<u>0.012</u>	<b>0.54</b>	<b>-0.002</b>	<b>0.008</b>	<b>0.44</b>	<b>-0.003</b>	<u>0.005</u>
BCT*	0.33	-0.018	-0.040	0.50	0.008	-0.039	0.38	0.019	-0.008
<b>Ours (UB)*</b>	0.67	0.015	0.011	0.90	0.017	-0.027	0.60	0.021	-0.005

\*Not subject to catastrophic forgetting

implications on the reduction of performance in compatibility can be observed in the self-test, as these values are significantly higher than the values reported by the methods learned using CiL.

In Table 11, results for the scenario of three, five, and ten-task  $CL^2R$  are presented. For each experiment, we report  $AC$  (Eq. 9),  $BC$  (Eq. 11), and  $FC$  (Eq. 12). As can be noticed, our method always achieves the highest  $AC$ , thus obtaining the largest number of compatible representations between models, and always achieves the highest  $BC$  between methods that are subject to catastrophic forgetting. FAN achieves almost the same performance as our procedure in the three-task scenario, while, when the number of tasks increases, it has a significant decrease in performance, especially in the ten-task setting. This may be due to increasing number of adaptation functions between different feature spaces that FAN uses to adapt old features with respect to the new ones. As can be noticed from the two tables, FOSTER does not learn compatible features. This may be due to the fact that feature space compression forces the representation to change abruptly reducing the overall compatibility with previous models. BCT reports higher values since its representation is learned from scratch for each new task. Compared to the upper bound (UB), our training procedure achieves lower  $AC$  and  $BC$ , this is due to the influence of catastrophic forgetting. From the table it can also be noticed that BiC, LUCIR, and PODNet do not satisfy compatibility when catastrophic forgetting is more severe, as, for example, in the case of ten-tasks. Overall, these results suggest that the interaction between local and global stationarity promoted by our training procedure shows a significant improvement in performance that feature distillation alone cannot provide.

### 4.9.3. Conclusions

Our contributions can be summarized as follows:

- We consider compatible representation learning within the lifelong learning paradigm. We refer to this general learning problem as *Compatible Lifelong Learning Representations* ( $CL^2R$ ).
- We define a novel set of metrics to properly evaluate  $CL^2R$  training procedures.
- We propose a  $CL^2R$  training procedure that imposes global and local stationarity on the learned features to achieve compatibility between representations under catastrophic forgetting. Global





Table 12. CIFAR10 evaluation. Two-task  $CL^2R$  setting with models trained on CIFAR100. Initial Task (i.e., the previous task) shows the verification accuracy on the first 50 classes, the other rows represent the performance obtained after two tasks.

METHOD	SELF TEST	CROSS TEST	ECC	BC	FC
Initial Task	0.65	–	–	–	–
ER	0.64	0.62	×	−0.034	−0.210
LwF	0.64	0.64	×	−0.009	<u>0.002</u>
BiC	0.66	0.63	×	−0.015	−0.028
LUCIR	<b>0.70</b>	<u>0.66</u>	✓	0.012	−0.038
FAN	0.66	0.63	×	−0.023	−0.035
FOSTER	0.66	0.57	×	−0.080	−0.090
$\ell$ -BCT	0.65	0.60	×	−0.047	−0.044
PODNet	<u>0.67</u>	<u>0.66</u>	✓	<u>0.014</u>	−0.013
<b>Ours</b>	0.66	<b>0.67</b>	✓	<b>0.017</b>	<b>0.006</b>
BCT*	0.72	0.65	✓	0.003	−0.071
<b>Ours (UB)*</b>	0.73	0.69	✓	0.039	−0.040

\*Not subject to catastrophic forgetting

and local interactions show a significant performance improvement when local stationarity is promoted only from already observed samples in the episodic memory.

- We empirically assess the effectiveness of our approach in several benchmarks showing improvements over baselines and adapted state-of-the-art methods.

#### 4.9.4. Relevant publications

- Biondi, Niccolò, Pernici, Federico, Bruni, Matteo, Mugnai, Daniele, and Del Bimbo, Alberto (2023). CL2R: Compatible Lifelong Learning Representations. ACM Transactions on Multimedia Computing, Communications and Applications, 18(2s), 1-22. Zenodo record: <https://zenodo.org/record/7551216>.

#### 4.9.5. Relevant software/datasets/other outcomes

- The PyTorch implementation of our work “CL<sup>2</sup>R: Compatible Lifelong Learning Representations” can be found in: <https://github.com/NiccoBiondi/CompatibleLifelongRepresentation>.

#### 4.9.6. Relevance to AI4media use cases and media industry applications

CL<sup>2</sup>R and CoReS share similarities in their foundational approach to data representation. Thus, CL<sup>2</sup>R can be seamlessly integrated with use cases such as “AI for News” (UC2) and “AI for Vision” (UC3) to boost tagging and search functionalities. Additionally, its alignment with foundational principles for content analysis makes it ideal for “AI for Content Organization and Content Moderation” (UC7) and “AI against Disinformation” (UC1), ensuring rapid and precise semantic understanding of updated news content.

The key difference between CL<sup>2</sup>R and CoReS lies in their learning methodologies. CoReS primarily learns from static datasets, while CL<sup>2</sup>R employs continual learning techniques, making it particularly relevant in today’s AI landscape. In essence, media companies using CL<sup>2</sup>R can train





deep learning-based recognition models without the need for vast amounts of training data. This not only addresses the ethical concerns associated with data privacy and storage but also reduces the environmental carbon footprint of AI operations. CL<sup>2</sup>R's continual learning minimizes the constant need for data storage and obviates the reprocessing of databases (or galleries). This results in both operational efficiency and the alleviation of ethical issues.

## 4.10. Contrastive Supervised Distillation for Continual Representation Learning

**Contributing partner(s):** UNIFI

### 4.10.1. Introduction and methodology

Deep Convolutional Neural Networks (DCNNs) have significantly advanced the field of visual search or visual retrieval by learning powerful feature representations from data [184–186]. Current methods predominantly focus on learning feature representations from static datasets in which all the images are available during training [187–189]. This operative condition is restrictive in real-world applications since new data are constantly emerging and repeatedly training DCNN models on both old and new images is time-consuming. Static datasets, typically stored on private servers, are also increasingly problematic because of the societal impact associated with privacy and ethical issues of modern AI systems [174, 176].

These problems may be significantly reduced in incremental learning scenarios as the computation is distributed over time and training data are not required to be stored on servers. The challenge of learning feature representation in incremental scenarios has to do with the inherent problem of catastrophic forgetting, namely the loss of previously learned knowledge when new knowledge is assimilated [170, 190]. Methods for alleviating catastrophic forgetting has been largely developed in the classification setting, in which catastrophic forgetting is typically observed by a clear reduction in classification accuracy [122, 131, 191–193]. The fundamental differences with respect to learning internal feature representation for visual search tasks are: (1) evaluation metrics do not use classification accuracy (2) visual search data have typically a finer granularity with respect to categorical data and (3) no classes are required to be specifically learned. These differences might suggest different origins of the two catastrophic forgetting phenomena. In this regard, some recent works provide some evidence showing the importance of the specific task when evaluating the catastrophic forgetting of the learned representations [194–197]. In particular, the empirical evidence presented in [194] suggests that feature forgetting is not as catastrophic as classification forgetting. We argue that such evidence is relevant in visual search tasks and that it can be exploited with techniques that learn incrementally without storing past samples in a memory buffer [198].

According to this, we propose a new distillation method for the continual representation learning task, in which the search performance degradation caused by feature forgetting is jointly mitigated while learning discriminative features. This is achieved by aligning current and previous features of the same class, while simultaneously pushing away features of different classes. We follow the basic working principle of contrastive loss [199] used in self-supervised learning, to effectively leverage label information in a distillation-based training procedure in which we replace anchor features with the feature of the teacher model.



Table 13. Evaluation on Stanford Dogs and CUB-200 of CSD and compared methods.

METHOD	STANFORD DOGS			CUB-200		
	RECALL@1 (1-60)	RECALL@1 (61-120)	RECALL@1 Average	RECALL@1 (1-100)	RECALL@1 (101-200)	RECALL@1 Average
Initial model	81.3	69.3	75.3	79.2	46.9	63.1
Fine-Tuning	74.0	<b>83.7</b>	78.8	70.2	75.1	72.7
MMD loss [195]	79.5	83.4	81.4	77.0	74.1	75.6
Feat. Est. [197]	79.9	83.5	81.7	77.7	75.0	76.4
CSD (Ours)	<b>80.9</b>	83.5	<b>82.2</b>	<b>78.6</b>	<b>78.3</b>	<b>78.5</b>
Joint Training	80.4	83.1	81.7	78.2	79.2	78.7

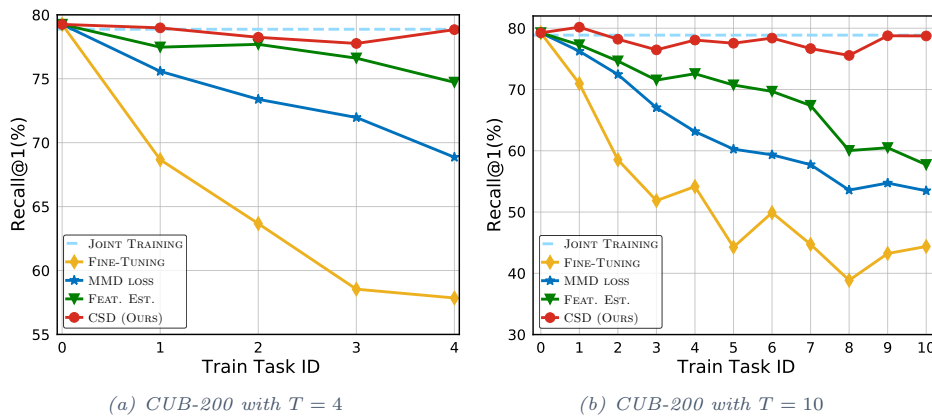


Figure 16. Evolution of RECALL@1 on the first task as new tasks are learned on CUB-200. Comparison between our method (CSD) and compared methods.

## 4.10.2. Experimental Results

### 4.10.2.1. Experimental Setup

The experiments are evaluated with  $T = 2, 5, 10$ . In CUB-200 and Stanford Dogs, following [200] [201], we use half of the data to pre-train a model and split the remaining data into  $T$  training-set. CUB-200 is evaluated with  $T = 1, 4, 10$  while Stanford Dogs with  $T = 1$ . Following [197], we adopt pretrained Google Inception [202] as representation model architecture on CUB-200 and Stanford Dogs with 512-dimension feature space. We trained the model for 2300 epochs for each task using the Adam optimizer with a learning rate of  $1 \cdot 10^{-5}$  for the convolutional layers and  $1 \cdot 10^{-6}$  for the classifier. Random crop and horizontal flip are used as image augmentation. We adopt RECALL@K [203] [200] as performance metric using each image in the test-set as query and the others as gallery.

### 4.10.2.2. Comparison with State-of-the-art Methods

We compare our method on CUB-200 and Stanford Dogs datasets with the Fine-Tuning baseline, MMD loss [195], and [197] denoted as Feature Estimation. As an upper bound reference, we report the Joint Training performance obtained using all the data to train the model.

We report in Table 13 the scores obtained with  $T = 1$  on the fine-grained datasets. On Stanford Dogs, our approach achieves the highest recall when evaluated on the initial task and comparable result with other methods on the final task with a gap of only 0.2% with respect to Fine-Tuning that focus only on learning new data. This results in our method achieving the highest average



recall value with an improvement of 0.5% RECALL@1 concerning Feature Estimation, 0.8% for MMD loss, and 3.4% for Fine-Tuning. On the more challenging CUB-200 dataset, we obtain the best RECALL@1 on both the initial and the final task outperforming the compared methods. Our method achieves the highest average recall value with an improvement of 2.1% RECALL@1 with respect to Feature Estimation, 2.9% for MMD loss, and 5.8% for Fine-Tuning. Differently from CIFAR-100, on fine-grained datasets, there is a lower dataset shift between different tasks leading to a higher performance closer to the Joint Training upper bound due to lower feature forgetting.

We report in Figure 16a and Figure 16b the challenging cases of CUB-200 with  $T = 4$  and  $T = 10$ , respectively. These experiments show, consistently with Table 13, how our approach outperforms state-of-the-art methods. In particular, with  $T = 10$  (Figure 16b), our method preserves the performance obtained on the initial task during every update. CSD largely improves over the state-of-the-art methods by almost 20% - 25% with respect to [197] and [195] achieving similar performance to the Joint Training upper bound. By leveraging labels information for distillation during model updates, CSD provides better performance and favorably mitigates the catastrophic forgetting of the representation compared to other methods that do not make use of this information.

#### 4.10.3. Conclusions

Our contributions can be summarized as follows:

- We address the problem of continual representation learning proposing a novel method that leverages label information in a contrastive distillation learning setup. We call our method Contrastive Supervised Distillation (CSD).
- Experimental results on different benchmark datasets show that our CSD training procedure achieves state-of-the-art performance.
- Our results confirm that feature forgetting in visual retrieval using fine-grained datasets is not as catastrophic as in classification.

#### 4.10.4. Relevant publications

- Barletti, T., Biondi, N., Pernici, F., Bruni, M., and Del Bimbo, A. (2022, May). Contrastive supervised distillation for continual representation learning. In Image Analysis and Processing-ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part I (pp. 597-609). Cham: Springer International Publishing. Zenodo record: <https://zenodo.org/record/7551163> (Best Student Paper Award).

#### 4.10.5. Relevant software/datasets/other outcomes

- The PyTorch implementation of our work “Contrastive Supervised Distillation for Continual Representation Learning” can be found in: <https://github.com/NiccoBiondi/ContrastiveSupervisedDistillation>.

#### 4.10.6. Relevance to AI4media use cases and media industry applications

CSD, as presented in our paper, offers a new method that helps media professionals search for and categorize content effectively. It’s particularly beneficial for applications such as news (“UC2 AI for News”), visual content (“UC3 AI for Vision”), countering misinformation (“UC1 AI against Disinformation”), and managing and moderating content (“UC7 AI for Content Organization and Content Moderation”). One of the primary advantages of CSD is its ability to learn new





information without forgetting previous knowledge. Unlike many Continual Learning methods that use a memory system to store and review old training data, CSD does not. This makes CSD more efficient and also provides a more ethical and privacy-focused approach, especially concerning the data used to train the recognition retrieval model. Such considerations are crucial for media professionals and participants involved in the aforementioned use cases, emphasizing the importance of data protection.





## 5. Manifold learning and disentangled feature representation (Task 3.2) – detailed description

**Contributing partners:** QMUL, JR, UNIFI, UNITN

In recent years, manifold and disentangled feature representation learning have risen as a prominent research area addressing the problem of finding meaningful representation schemes for both the generative and the discriminative learning paradigms. In the generative regime, studying the structure of latent spaces of generative methods (such as GANs) by discovering semantic paths that govern the generation process, has proven to be very useful in understanding and controlling image generation. For instance, by discovering interpretable or controllable generative paths for manipulating the generation process (e.g., image editing) [1–3]. In the discriminative regime, learning meaningful feature representations, along with metrics that model data manifolds better (i.e., by adopting the hyperbolic geometry instead of the widely used Euclidean modeling [4]), lead to better, more discriminative features, and, thus, improve significantly the performance in visual understanding tasks (such as image retrieval). Advances in both generative and discriminative regimes are particularly useful in media generation and visual content analysis.

### 5.1. Finding non-linear RBF paths in GAN latent space

**Contributing partners:** QMUL

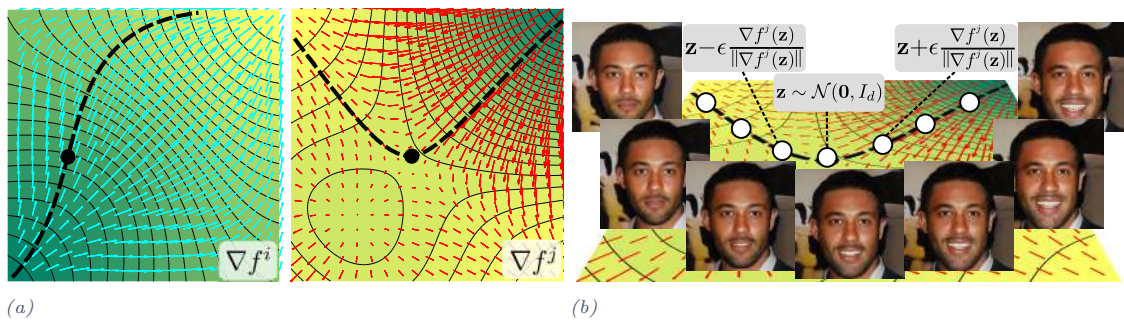


Figure 17. (a) Warpings of vector space  $\mathbb{R}^d$  due to two RBF functions,  $f^i$  and  $f^j$ , lead to different non-linear paths in  $\mathbb{R}^d$  for any given  $\mathbf{z} \in \mathbb{R}^d$  (dashed bold lines) via their gradients,  $\nabla f^i$  and  $\nabla f^j$ . Solid black lines represent isohypses of the warpings and the colored vectors represent the vector fields induced by their gradients. (b) Illustration of a non-linear path due to warping  $f^j$ , starting from a latent code  $\mathbf{z}$  and moving along the gradient  $\nabla f^j$  by steps of magnitude  $\epsilon$ .

#### 5.1.1. Introduction and methodology

Generative Adversarial Network (GAN) [204] has emerged as the leading generative learning paradigm, exhibiting clear superiority in terms of the quality of generated realistic and aesthetically pleasing images. However, despite their generative efficiency, GANs do not provide an inherent way of comprehending or controlling the underlying generative factors. To address this, the research community has directed its efforts towards studying the structure of GAN’s latent space [205–218]. These works study the structure of GAN’s latent space and attempt to find interpretable directions on it; that is, directions sampling across which are expected to generate images where only a few



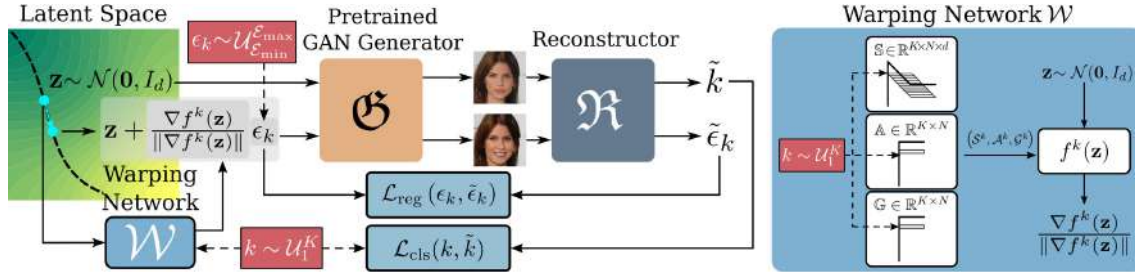


Figure 18. Overview WarpedGANSpace: A latent code  $\mathbf{z} \sim \mathcal{N}(0, I_d)$  is shifted by a vector induced by a warping function  $f^k$  implemented by the warping network  $\mathcal{W}$  after choosing the corresponding support set  $\mathcal{S}^k$ , weights  $\mathcal{A}^k$ , and parameters  $\mathcal{G}^k$ . The pair of latent codes,  $\mathbf{z}$  and  $\mathbf{z} + \epsilon_k \frac{\nabla f^k(\mathbf{z})}{\|\nabla f^k(\mathbf{z})\|}$ , are then fed into the generator  $\mathcal{G}$  in order to produce two images. The reconstructor  $\mathcal{R}$  is optimized to reproduce the signed shift magnitude  $\epsilon_k$  and predict the index  $k$  of the support set used.

Method	GAN				
	SNGAN (MNIST)	SNGAN (Anime)	BigGAN	ProgGAN	StyleGAN2
Random	46.0	85.0	76.0	60.0	-
Coord	48.0	89.0	66.0	82.0	-
Linear [212]	88.0	99.0	85.0	90.0	-
Ours	<b>98.4</b>	<b>99.8</b>	<b>92.6</b>	<b>99.3</b>	<b>99.8</b>

Table 14. Reconstructor accuracy (%) of the proposed method compared to [212] (linear directions), random latent direction and latent directions aligned with axes, for various GAN generators pretrained on the given datasets.

(ideally one) factors of variations are “activated”. Meaningful human-interpretable directions can refer to either domain-specific factors (e.g., facial expressions [205]) or domain-agnostic factors (e.g., zoom scale [215–217]).

In this work, we propose to learn non-linear warping functions on the latent space, each one parametrized by a set of RBF-based latent space warping operations, and where each warping function  $f^k$  gives rise to a family of non-linear paths via its gradient. More precisely, at each latent code  $\mathbf{z} \in \mathbb{R}^d$ , the gradient of the warping function  $\nabla f^k(\mathbf{z})$  gives the direction along the  $k$ -th family of paths – clearly, the gradient of  $f^k(\mathbf{z})$  is not isotropic in  $\mathbb{R}^d$ , giving rise to non-linear paths. An example is shown in Figure 17, where two RBF-warping functions  $f^i$  and  $f^j$  are depicted together with two distinct non-linear paths. Building on the work of [212], that discovers linear paths, we optimize the trainable parameters of the RBFs, so as that images that are generated by codes along paths of different families,  $f^k$ , are easily distinguishable by a discriminator network (Figure 18) – this leads to easily distinguishable image transformations, such as pose and facial expressions in facial images (Fig. 17b). We show that [212], which learns linear paths, can be derived as a special case of our method and perform extensive comparisons with state-of-the-art methods both qualitatively and quantitatively.

### 5.1.2. Experimental results

**5.1.2.1. Pretrained GAN generators and datasets** We evaluate the proposed method using the following pretrained GANs: a) Spectrally Normalized GAN (SN-GAN) [221] trained on MNIST [222] and AnimeFaces [223], b) BigGAN [224] trained on ImageNet [225], c) ProgGAN [220] trained on CelebA-HQ [226], and d) StyleGAN2 [219] trained on FFHQ [219].



Figure 19. Non-linear interpretable paths automatically discovered by our method in StyleGAN2’s [219]  $W$ -space.

**5.1.2.2. Paths with more distinguishable changes in the image space** We first show that a reconstructor that discriminates images according to the warping in the latent space that generated them, i.e., estimates the index of the warping function, has better classification performance than in the corresponding linear case [212]. This is an indication that the paths that are generated by our method can be discriminated more effectively and therefore are more likely to be more interpretable. The results are summarised in Table 14 and are consistent across several pretrained GANs.

**5.1.2.3. Non linear interpretable paths with steeper and more disentangled changes in the image space – quantitative evaluation** In this section, we will present our quantitative evaluation scheme, which we use for assessing the performance of our method and compare it to state-of-the-art [212, 218], for ProgGAN and StyleGAN2.

As discussed before, for a given method that discovers a set of interpretable paths; that is, linear in the cases of [212, 218] or non-linear in the case of the proposed method, in the latent space of a pretrained GAN generator, we generate an image sequence for each path, starting from a random latent code and “walking” towards the positive and the negative ways of the path for a certain amount of steps. For each image of such sequence, we apply a set of pretrained networks that predict the following: a) the location of the face (bounding box), using [227], b) an identity score for each image of the sequence that expresses the similarity between the original image (central image of the sequence) and each of the rest, using ArcFace [228], c) an age, race, and gender score using FairFace [229], d) a set of CelebA attributes classifiers (e.g., smile, wavy hair, etc.), and e) an estimation of the face pose (yaw, pitch, roll), using Hopenet [230]. In this way, for each warping we have a set of paths in the latent space and the corresponding paths in the attribute space.

In order to obtain a measure on how well the paths generated by a warping function are correlated with a certain attribute, we estimate the average Pearson’s correlation between the index of the step along the path and the corresponding values in the attribute vector. By doing so, for each warping, we obtain a vector, which we normalize. This allows for sorting the discovered paths with respect to the correlation with each attribute and select the paths that give the maximum absolute correlation for each attribute.

The results are summarised in Table 15, where we report quantitative results for our method (Table 15a), in comparison to [212] (Table 15b) and [218] (Table 15c), in terms of  $\mathcal{L}_1$ -normalized correlation averaged across 100 latent codes. We note that our method achieves better correlations for the respective attributes, while at the same time the correlations with the rest of the attributes are lower than those achieved by [212, 218], as is evident by the lower values in the off-diagonal





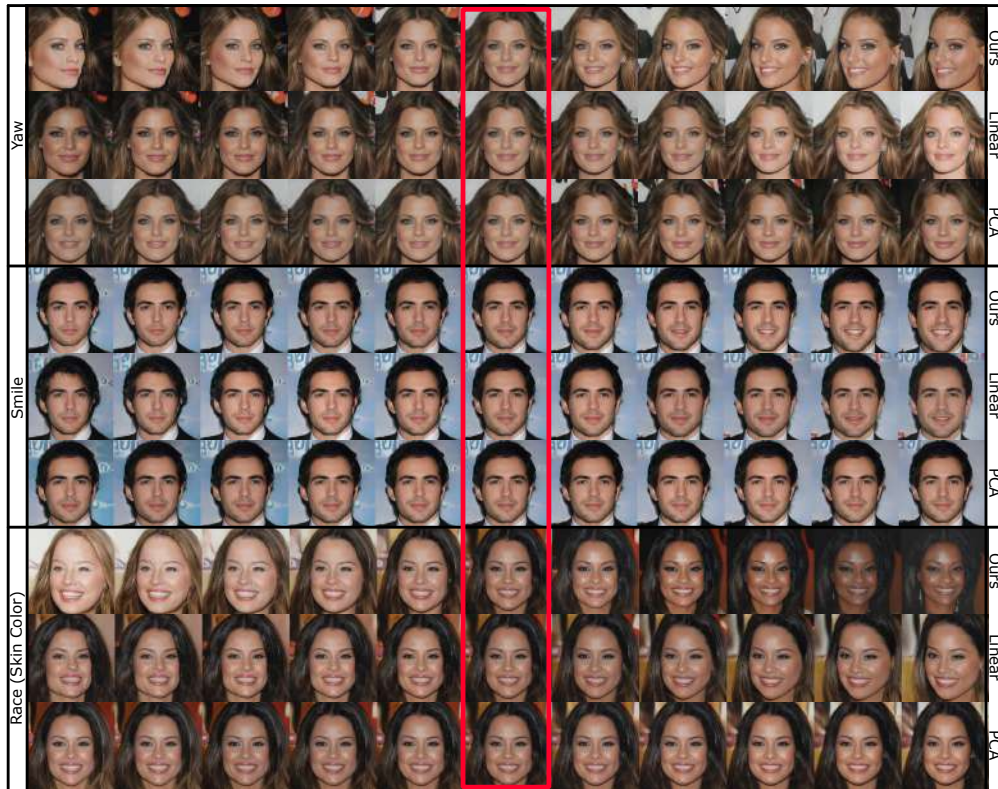


Figure 20. Automatically discovered non-linear (ours – first row) and linear (Voynov and Babenko [212] – second row, GANSpace [218] – third row) interpretable paths in ProgGAN’s [220] latent space.

elements of the matrix. This shows in a quantitative manner, what was evident in a qualitatively manner in Figure 20, that is, that the discovered paths in the latent space lead to more disentangled changes in the attribute space.

Finally, in Figure 19, we show the results of generation across some non-linear interpretable paths obtained automatically by our method for StyleGAN2, for the following attributes: *age*, *race* (skin color), *gender* (“femaleness”), and *yaw* (rotation). In this figure, we report the paths with the highest correlation with the respective attribute.

### 5.1.3. Relevant publications

- Tzelepis, C., Tzimiropoulos, G., Patras, I. (2021). “WarpedGANSpace: Finding non-linear RBF paths in GAN latent space”. ICCV 2021 [1].  
Zenodo record: <https://zenodo.org/record/5550474>.

### 5.1.4. Relevant software/datasets/other outcomes

- The PyTorch implementation of our work “WarpedGANSpace: Finding non-linear RBF paths in GAN latent space” can be found in <https://github.com/chi0tzip/WarpedGANSpace>.





Table 15. Comparison of the proposed method (non-linear latent paths) to [212] (linear latent directions) and GANSpace [218] (linear PCA-based latent directions) in terms of  $L_1$ -normalized correlation and range ( $r$ ).

	ID	Yaw	Pitch	Smile	Race	Hair	$r$
Yaw	0.52	<b>0.32</b>	0.05	0.01	0.07	0.03	<b>43.66°</b>
Pitch	0.41	0.04	<b>0.38</b>	0.13	0.03	0.01	<b>22.53°</b>
Smile	0.24	0.03	0.07	<b>0.61</b>	0.03	0.03	<b>0.37</b>
Race	0.32	0.03	0.12	0.08	<b>0.29</b>	0.17	0.06
Hair	0.23	0.02	0.11	0.13	0.02	<b>0.49</b>	<b>0.28</b>

(a) Non-linear paths (Ours).

	ID	Yaw	Pitch	Smile	Race	Hair	$r$
Yaw	0.51	<b>0.24</b>	0.21	0.01	0.02	0.01	18.93°
Pitch	0.47	0.01	<b>0.25</b>	0.04	0.00	0.22	8.27°
Smile	0.24	0.01	0.04	<b>0.57</b>	0.05	0.09	0.28
Race	0.52	0.05	0.02	0.10	<b>0.31</b>	0.01	<b>0.16</b>
Hair	0.43	0.00	0.10	0.06	0.04	<b>0.36</b>	0.27

(b) Linear directions (Voynov and Babenko [212]).

	ID	Yaw	Pitch	Smile	Race	Hair	$r$
Yaw	0.47	<b>0.27</b>	0.04	0.13	0.03	0.06	17.65°
Pitch	0.45	0.05	<b>0.38</b>	0.09	0.02	0.01	7.48°
Smile	0.21	0.00	0.07	<b>0.55</b>	0.08	0.08	0.21
Race	0.35	0.11	0.02	0.12	<b>0.27</b>	0.12	0.10
Hair	0.44	0.05	0.06	0.03	0.08	<b>0.34</b>	0.15

(c) Linear PCA directions (GANSpace [218]).

### 5.1.5. Relevance to AI4media use cases and media industry applications

Our algorithm provides a solution for discovering interpretable paths in the latent space of generative methods, such as Generative Adversarial Networks (GANs). That is, discovering ways of understanding and, thus, controlling the generative process. As such, it may exhibit wide applicability in various media industry applications, such as image editing given real visual data. In such a scenario, a human in the loop (e.g., a journalist or a creative artist) can perform image editing of visual data coming from news feeds. More generally, since generative learning is fundamental in creative industries, our method can be used in order to control media generation (e.g., virtual characters, animations, etc), that can subsequently be incorporated in media industry for the generation of controllable media content.

## 5.2. Unsupervised learning of parts and appearances in the feature maps of GANs

**Contributing partners:** QMUL

### 5.2.1. Introduction and methodology

Generative Adversarial Networks (GANs) [204] constitute the SOTA for the task of image synthesis. However, despite the remarkable progress in this domain through improvements to the image generator’s architecture [205, 219, 231–234], their inner workings remain to a large extent unexplored. Developing a better understanding of the way in which high-level concepts are represented and





composed to form synthetic images is important for a number of downstream tasks such as generative model interpretability [235–237] and image editing [1, 212, 218, 238–240]. In modern generators however, the synthetic images are produced through an increasingly complex interaction of a set of per-layer latent codes in tandem with the feature maps themselves [219, 232, 233] and/or with skip connections [234]. Furthermore, given the rapid pace at which new architectures are being developed, demystifying the process by which these vastly different networks model the constituent parts of an image is an ever-present challenge. Thus, many recent advances are architecture-specific [241–243] and a general-purpose method for analyzing and manipulating convolutional generators remains elusive.

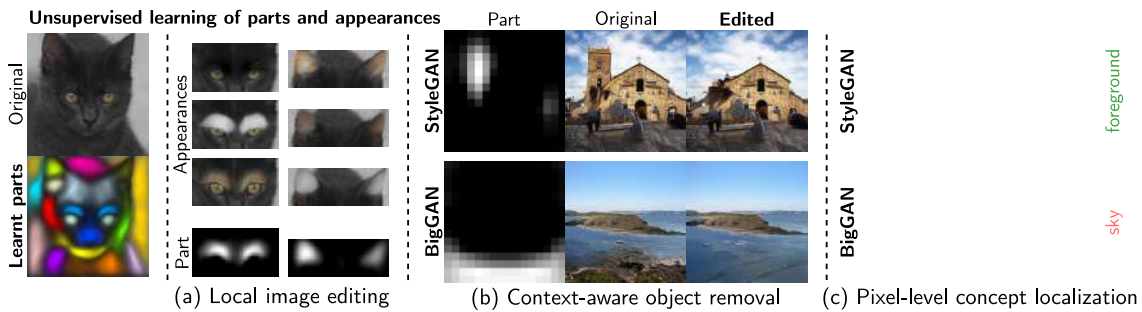


Figure 21. We propose an unsupervised method for learning a set of factors that correspond to interpretable parts and appearances in a dataset of images. These can be used for multiple tasks: (a) local image editing, (b) context-aware object removal, and (c) producing saliency maps for learnt concepts of interest.

A popular line of GAN-based image editing research concerns itself with learning so-called “interpretable directions” in the generator’s latent space [1, 212, 218, 237–239, 244–246]. Once discovered, such representations of high-level concepts can be manipulated to bring about predictable changes to the images. One important question in this line of research is how latent representations are combined to form the appearance at a particular *local* region of the image. Whilst some recent methods attempt to tackle this problem [241, 243, 247–251], the current state-of-the-art methods come with a number of important drawbacks and limitations. In particular, existing techniques require prohibitively long training times [241, 249], costly Jacobian-based optimization [249, 252], and the requirement of semantic masks [241] or manually specified regions of interest [249, 252]. Furthermore, whilst these methods successfully find directions affecting local changes, optimization must be performed on a per-region basis, and the resulting directions do not provide *pixel-level* control—a term introduced by [249] referring to the ability to *precisely* target specific pixels in the image.

In this light, we present a fast unsupervised method for *jointly* learning factors for interpretable parts and their appearances (we thus refer to our method as *Panda*) in pre-trained convolutional generators. Our method allows one to both interpret and edit an image’s style at discovered local semantic regions of interest, using the learnt appearance representations. We achieve this by formulating a constrained optimization problem with a semi-nonnegative tensor decomposition of the dataset of deep feature maps  $\mathcal{Z} \in \mathbb{R}^{M \times H \times W \times C}$  in a convolutional generator. This allows one to accomplish a number of useful tasks, prominent examples of which are shown in Figure 21. Firstly, our learnt representations of appearance across samples can be used for the popular task of local image editing [241, 249] (for example, to change the colour or texture of a cat’s ears as shown in Figure 21 (a)). Whilst the state-of-the-art methods [241, 249, 252] provide fine-grained control over a target region, they adopt an “annotation-first” approach, requiring an end-user to first manually specify a ROI. By contrast, our method fully exploits the unsupervised learning paradigm, wherein



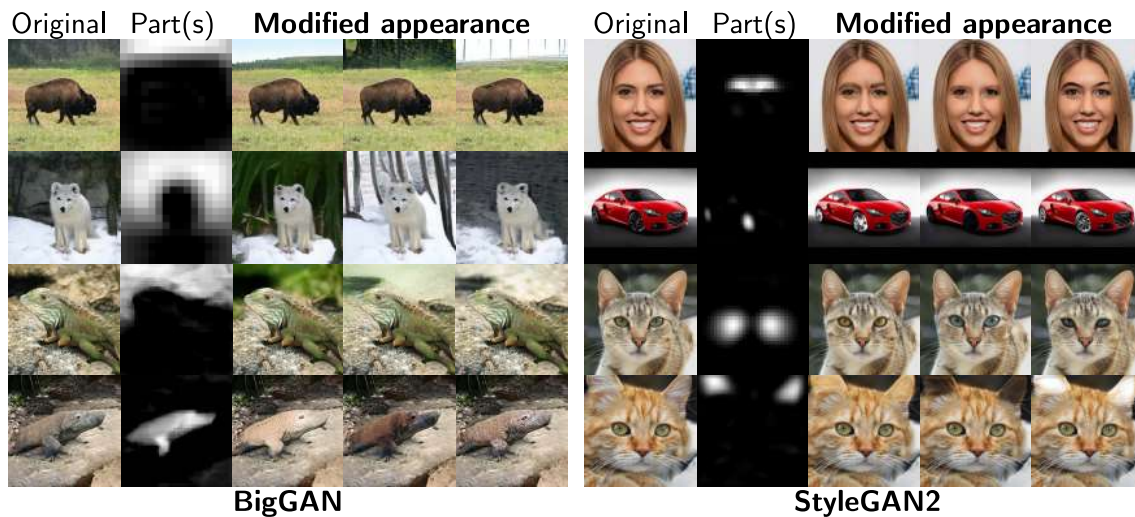


Figure 22. Local image editing on a number of architectures and datasets, using both the global and refined parts factors. At each column, the original image is edited at the target part with a different appearance vector.

such concepts are discovered automatically and without any manual annotation. These discovered semantic regions can then be chosen, combined, or even modified by an end-user as desired for local image editing.

More interestingly still, through a generic decomposition of the feature maps our method identifies representations of common concepts (such as “background”) in *all* generator architectures considered (all 3 StyleGANs [219, 232, 233], ProgressiveGAN [253], and BigGAN [234]). This is a surprising finding, given that these generators are radically different in architecture. By then editing the feature maps using these appearance factors, we can thus, for example, *remove* specific objects in the foreground (Figure 21 (b)) in all generators, seamlessly replacing the pixels at the target region with the background appropriate to each image.

However, our method is useful not only for local image editing, but also provides a straightforward way to localize the learnt appearance concepts in the images. By expressing activations in terms of our learnt appearance basis, we are provided with a *visualization* of how much of each of the appearance concepts are present at each spatial location (i.e., saliency maps for concepts of interest). By then thresholding the values in these saliency maps (as shown in Figure 21 (c)), we can localize the learnt appearance concepts (such as sky, floor, or background) in the images—without the need for supervision at any stage.

### 5.2.2. Experimental results

Here, we showcase our method’s ability to perform local image editing in pre-trained GANs, on 5 generators and 5 datasets (ImageNet [254], AFHQ [255], FFHQ [232], LSUN [256], and MetFaces [257]). In Figure 22, we show a number of interesting local edits achievable with our method, using both the global and refined parts factors. Whilst we can manipulate the style at common regions such as the eyes with the global parts factors, the refined parts factors allow one to target regions such as an individual’s clothes, or their background. One is not limited to this set of learnt parts however. For example, one can draw a ROI by hand at test-time or modify an existing part. This way, pixel-level control (e.g., opening only a single eye of a face) is achievable in a





way that is not possible with the SOTA methods [241, 249].

We next compare our method to state-of-the-art GAN-based image editing techniques in Figure 23. In particular, we train our model at layer 5 using  $R_S = 8$  global parts factors, with no refinement. As can be seen, SOTA methods such as LowRank-GAN [249] excel at enlarging the eyes in a photo-realistic manner. However, we frequently find the surrounding regions to change as well. This is seen clearly by visualizing the mean squared error [242] between the original images and their edited counterparts, shown in every second row of Figure 23. We further quantify this ability to affect local edits in the section that follows.



Figure 23. Qualitative comparison to SOTA methods editing the “eyes” ROI. We also show the mean squared error [242] between the original images and their edited counterparts, highlighting the regions that change.

We compute the ratio of the distance between the pixels of the original and edited images in the region of ‘disinterest’, over the same quantity with the region of interest:

$$\text{ROIR}(\mathcal{M}, \mathcal{X}, \mathcal{X}') = \frac{1}{N} \sum_{i=1}^N \frac{\|(\mathbf{1} - \mathcal{M}) * (\mathcal{X}_i - \mathcal{X}'_i)\|}{\|\mathcal{M} * (\mathcal{X}_i - \mathcal{X}'_i)\|}, \quad (13)$$

where  $\mathcal{M} \in [0, 1]^{H \times W \times C}$  is an  $H \times W$  spatial mask (replicated along the channel mode) specifying the region of interest,  $\mathbf{1}$  is a 1-tensor, and  $\mathcal{X}, \mathcal{X}' \in \mathbb{R}^{N \times \tilde{H} \times \tilde{W} \times \tilde{C}}$  are the batch of original and edited versions of the images respectively. A small ROIR indicates more ‘local’ edits, through desirable change to the ROI (large denominator) and little change elsewhere (small numerator). We compute this metric for our method and SOTA baselines in Table 16, for a number of regions of interest. As can be seen, our method consistently produces more local edits than the SOTA for a variety of regions of interest. We posit that the reason for this is due to our operating directly on the feature maps, where the spatial activations have a direct relationship to a patch in the output image.

### 5.2.3. Relevant publications

- Oldfield, J., Tzelepis, C., Panagakis, Y., Nicolaou, M. A., Patras, I. ”Panda: Unsupervised learning of parts and appearances in the feature maps of GANs”. ICLR 2023. [2]. Zenodo record: <https://zenodo.org/record/7682257>.

### 5.2.4. Relevant software/datasets/other outcomes

- The PyTorch implementation of our work “Panda: Unsupervised Learning of Parts and Appearances in the Feature Maps of GANs” can be found at: <https://github.com/james-oldfield/Panda>.





Table 16. ROIR ( $\downarrow$ ) of eq. (13) for 10k FFHQ samples per local edit.

	Eyes	Nose	Open mouth	Smile
GANSpace [218]	2.80±1.22	4.89±2.11	3.25±1.33	2.44±0.89
SeFa [238]	5.01±1.90	6.89±3.04	3.45±1.12	5.04±2.22
StyleSpace [241]	1.26±0.70	1.70±0.82	1.24±0.44	2.06±1.62
LowRankGAN [249]	1.78±0.59	5.07±2.06	1.82±0.60	2.31±0.76
ReSeFa [252]	2.21±0.85	2.92±1.29	1.69±0.65	1.87±0.75
<b>Ours</b>	<b>1.04±0.33</b>	<b>1.17±0.44</b>	<b>1.04±0.39</b>	<b>1.05±0.38</b>

### 5.2.5. Relevance to AI4media use cases and media industry applications

Our algorithm for learning a set of factors that correspond to interpretable parts and appearances in a dataset of images contributes and provides solution to the general cases of local image editing, context-aware object removal, and producing saliency maps for learnt concepts of interest. This is connected to visual content generation that can be useful in visual analysis and creation in the media industry. The provided solution allows a human in the loop (e.g., a journalist or a creative artist) to perform image editing that is of interest in news-related content. For instance, removing, blurring, or even highlighting specific objects of interest in the visual stream of news content.

## 5.3. Dataset Anonymization with Generative Models

**Contributing partners:** QMUL, UNITN

### 5.3.1. Introduction and methodology

Considering that modern machine learning algorithms learn from vast amounts of data often crawled from the Web [232, 258], it has become increasingly important to consider the impact this has on the privacy of those individuals depicted. Motivated by privacy concerns, many societies have recently enacted strict legislation, such as the General Data Protection Regulation (GDPR) [259], which requires the consent of every person that might be depicted in an image dataset. Whilst such laws have obvious benefits to the privacy of those featured in image datasets, this is not without costly side effects to the research community. In particular, research fields such as computer vision and machine learning rely on the creation and sharing of high-quality datasets of images of humans for a number of important tasks including security [260], healthcare [261], and creative applications [232, 262].

A recent line of research focuses on overcoming this issue by *anonymizing* the identity of the individuals in image datasets. Through this approach, the machine learning community can still benefit from the wealth of large datasets of high-resolution images, but without cost to privacy. A certain line of work leverages the power of GANs [265], which have recently been used for discovering controllable generation paths in their latent or feature spaces [1, 2, 266–268]. Towards face anonymization, GANs have been incorporated in order to synthesize new images in order to obtain photos that maintain most of the image while changing the face of the subject of interest. In particular, these approaches use techniques like image inpainting [264], conditional generation [263], attribute manipulation [269], or adversarial perturbation [270]. These works are able to obtain anonymized images that can still be used for computer vision tasks such as tracking and detection, with very good results in terms of privacy preservation. However, many of these works lack the








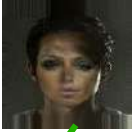


	Original	CIAGAN	DeepPrivacy	Ours
				
				
ID anonymized		✓	✓	✓
Attr. preserved		✗	✗	✓

Figure 24. Face dataset anonymization: Comparison of our method [3] to CIAGAN [263] and DeepPrivacy [264] in terms of identity anonymization and attribute preservation.

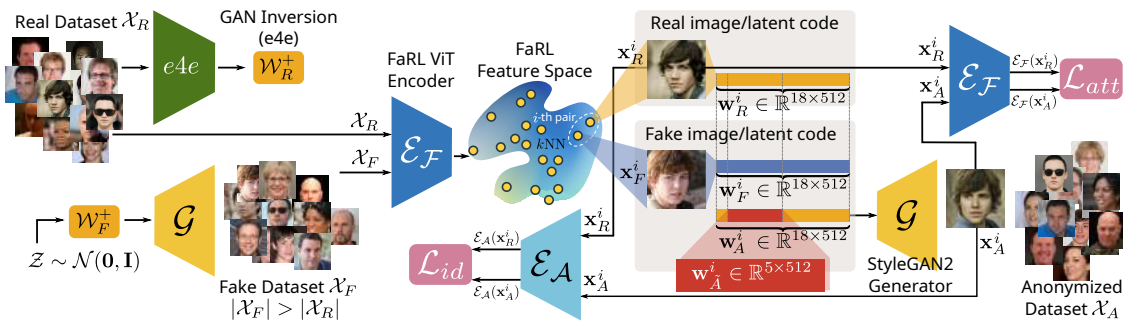


Figure 25. Face dataset anonymization: Optimizing the trainable portion of the latent code  $w_A^i \in \mathbb{R}^{5 \times 512}$  to obfuscate the identity of the resulting synthetic image  $x_A^i$  with  $\mathcal{L}_{id}$  whilst preserving the facial attributes with  $\mathcal{L}_{att}$ .

ability to generate natural-looking faces and often fail to preserve the original facial attributes in the anonymized images (or, on the occasions in which such methods do preserve the facial attributes, they fail to demonstrate this quantitatively). This is critical for many applications which rely on the attributes of the inner face, such as expression recognition [271], or mental health affect analysis [272]. To further complicate the picture, a fundamental problem often found with existing works is the way in which the anonymized images copy not just the original image's background, but also more identifiable features [263, 264], such as the clothes of an individual, or their hair (see examples of this in Figure 24). We argue that leaving such structure of the images unchanged constitutes a glaring privacy vulnerability, as one can re-identify the original image from the anonymized counterpart by comparing the image background or person's clothes.

Motivated by these concerns, we propose to de-identify individuals in datasets of facial images whilst *preserving* the facial attributes of the original images. To achieve this, in contrast to existing work [263, 264, 269, 273, 274] that train custom neural networks from scratch, we propose to work directly in the latent space of a powerful pre-trained GAN, optimizing the latent codes directly with losses that explicitly aim to retain the attributes and obfuscate the identities. More concretely, we use a deep feature-matching loss [275] to match the high-level semantic features between the original and the fake image generated by the latent code, and a margin-based identity loss to control the similarity between the original and the fake image in the ArcFace [276] space. The initialisation of the latent codes is obtained by randomly sampling the latent space of GAN, using them to generate the corresponding synthetic images and finding the nearest neighbors in a semantic space



(FARL [275]). In order to preserve texture and pose information of the original image, we perform inversion of the original image and retain the parts that correspond to the properties we want to preserve in the final code. This results in a latent code that yields a high-resolution image that contains a new identity but retains the same facial attributes as the original image.

### 5.3.2. Experimental results

**5.3.2.1. Datasets and the SOTA** We perform anonymization on the following datasets: (i) **CelebA-HQ** [277], which contains 30000  $1024 \times 1024$  face images of celebrities from the CelebA dataset with various demographic attributes (e.g., age, gender, race) and where each image is annotated with 40 attribute labels related to the inner and outer regions of the face, and (ii) **LFW** [278], which contains over 13000 images collected from the Web (5749 identities with 1680 of those identities being pictured in at least 2 images). We compare our anonymization framework with two state-of-the-art anonymization methods, namely CIAGAN [263] and DeepPrivacy [264].

In Figures 26a, 26b we make a qualitative comparison between our method and the SOTA [263, 264]. As can be clearly seen, our method is capable of retaining the facial attributes of the image to a much greater extent than the SOTA. In Tables 17, 18 we show the results for FID [279], face detection, and face re-identification for the two considered datasets. We see that our method excels at producing the most realistic-looking images under the FID metric for CelebA-HQ in Table 17, and also outperforms the baselines for the FID metric on LFW [278] in Table 18 when considering the CelebA-HQ [277] dataset as the “target” distribution<sup>1</sup>.

	FID↓	Detection↑		Face re-ID↓	
		dlib	MTCNN(%)	CASIA(%)	VGG(%)
Randomly generated	<u>18.09</u>	100	99.99	3.61	1.08
CIAGAN [263]	37.94	95.10	99.82	<b>2.19</b>	<b>0.37</b>
DeepPrivacy [264]	32.99	92.82	99.85	3.61	1.05
<b>Ours (ID)</b>	44.12	98.58	97.99	3.28	0.58
<b>Ours (ID+attributes)</b>	44.11	100	<b>100</b>	3.06	2.06
<b>Ours</b>	<b>29.93</b>	100	<b>100</b>	2.80	1.67

Table 17. CelebA-HQ [277] privacy and image quality results.

	FID↓	FID (C-HQ)↓	Detection↑		Face re-ID↓	
			dlib	MTCNN(%)	CASIA(%)	VGG(%)
CIAGAN [263]	<b>22.07</b>	85.23	98.14	99.89	<b>0.17</b>	<b>0.91</b>
DeepPrivacy [264]	23.46	123.67	96.70	99.57	2.74	1.52
<b>Ours</b>	27.45	<b>68.88</b>	100	<b>100</b>	2.07	1.58

Table 18. LFW [278] privacy and image quality results.

We quantify the attribute preservation of the anonymization methods for the CelebA-HQ [277] in Table 19. As can be seen, our method’s images result in a classifier capable of almost the same accuracy as when training on the original labels, demonstrating the ability of our method to retain the original facial features. Whilst the other two baselines also produce reasonable results under

<sup>1</sup>Given that CelebA-HQ is of much higher quality than LFW, we report both cases to demonstrate that our images can better match the distribution of high-resolution data.





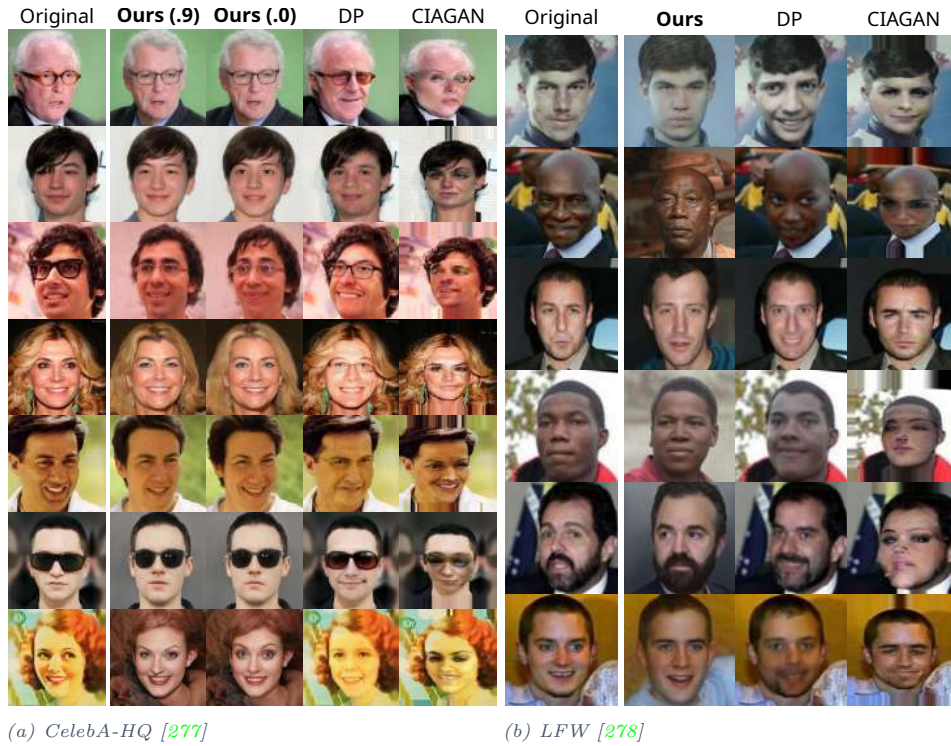


Figure 26. Anonymization results on (a) the CelebA-HQ [277] and (b) the LFW [278] datasets in comparison to DeepPrivacy (DP) [264] and CIAGAN [263].

this *combined* accuracy metric, we argue this is because of the way in which they preserve the image outside the region of the inner face of the images.

	Inner face	Outer face	Combined
Original	<u>0.8409</u>	<u>0.8683</u>	<u>0.8539</u>
CIAGAN [263]	0.7277	0.8372	0.7852
DeepPrivacy [264]	0.7658	0.8511	0.8135
<b>Ours</b>	<b>0.7817</b>	<b>0.8518</b>	<b>0.8181</b>

Table 19. Attribute classification results on CelebA-HQ [277].

### 5.3.3. Relevant publications

- Barattin, S., Tzelepis, C., Patras, I., and Sebe, N. “Attribute-preserving Face Dataset Anonymization via Latent Code Optimization”. CVPR 2023. [3].  
Zenodo record: <https://zenodo.org/record/8012381>.

### 5.3.4. Relevant software/datasets/other outcomes

The PyTorch implementation of our work “Attribute-preserving Face Dataset Anonymization via Latent Code Optimization” (CVPR 2023) can be found in <https://github.com/chi0tzp/FALCO>.





### 5.3.5. Relevance to AI4media use cases and media industry applications

Our algorithm for the anonymization of facial images contributes and provides solution to the general cases where the privacy protection of human faces is important. This can be crucial to the media industry in the cases where depicted faces (e.g., of people that have not given consent) need to be de-identified before being published as part of a news content. The efficiency of our method allows for the fast processing of images/videos, which is typically needed due to the dynamic nature of news and media content, where the depicted faces can be anonymized without losing crucial facial attributes, such as skin colour, facial expressions, etc. Furthermore, our method can help towards sharing important datasets that have been collected by media industry organisations for decades and that now may violate GDPR, such as datasets of humans coming from news images/videos. Our anonymization method can protect the identity of the depicted humans, while rendering those datasets useful for the industry and the research community.

## 5.4. A survey of manifold learning and its applications for multimedia

**Contributing partners:** JR

### 5.4.1. Introduction

Deep learning methods are nowadays the best way for the automatic analysis of multimedia data (e.g. images, video or 3D data) for tasks like classification or detection. However, classic neural networks are restricted to data lying in vector spaces, while data residing in smooth non-Euclidean spaces arise naturally in many problem domains. For example, a 360° camera actually captures a spherical image, not a rectangular image. We will focus in this survey on manifolds, especially Riemannian manifolds, which are well suited for generalizing a vector space because they are locally Euclidian and differentiable.

A *manifold*  $M$  of dimension  $d$  corresponds to a topological structure which locally (so in the neighborhood of a point  $\mathbf{p} \in M$ ) looks like a  $d$ -dimensional Euclidean space. The "best" local approximation of this neighborhood of  $\mathbf{p}$  with a  $d$ -dimensional Euclidean space is its *tangent space*  $T_{\mathbf{p}}M$ . The tangent space  $T_{\mathbf{p}}M$  can be seen as a linear approximation of  $M$  around  $\mathbf{p}$ . For example, for a 2-dimensional manifold its tangent space  $T_{\mathbf{p}}M$  is the tangent plane going through this point (see Figure 27). A *Riemannian manifold* is a smooth manifold  $M$  equipped with a positive definite inner product  $g_{\mathbf{p}}$  on the tangent space  $T_{\mathbf{p}}M$  of each point  $\mathbf{p}$ .

The inner product  $g$  induces a norm on the tangent space, which subsequently allows us to calculate curve lengths and distances on the manifold  $M$ . For each curve  $c(t)$  on the manifold its length can be calculated by integrating the norm along the curve (for details see [280–284]). A *geodesic* curve is a *length-minimizing* curve connecting two points  $\mathbf{p}$  and  $\mathbf{q}$  on the manifold. The distance between these points is defined as the length of the geodesic.

Let  $\mathbf{p}$  be a (reference) point on the manifold and  $v$  a vector of its tangent space  $T_{\mathbf{p}}M$ . The vector  $v$  can be mapped now to the point  $\mathbf{q}$  on the manifold that is reached after unit time  $t = 1$  by the geodesic  $c(t)$  starting at  $\mathbf{p}$  with tangent vector  $v$ . This mapping  $exp_{\mathbf{p}}(v) : T_{\mathbf{p}}M \rightarrow M$  is called the exponential map at point  $\mathbf{p}$ .

The inverse mapping  $log_{\mathbf{p}}(\mathbf{q}) : M \rightarrow T_{\mathbf{p}}M$  is uniquely defined around a neighborhood of  $\mathbf{p}$ . Informally, the exponential map and logarithm map move points back and forth between the manifold and the tangent space (see Figure 27) while preserving distances. Furthermore, derivative operators like *differential*, *intrinsic gradient*, *divergence* and *laplacian* can be also defined on a manifold [285, 286], which allows us to perform calculus on the manifold.





Closely related to manifolds are Lie groups. A *Lie group* is a smooth manifold that also forms a *group* [280], where both group operations (commonly called *multiplication* and *inverse*) are smooth mappings of manifolds. The *Lie algebra*  $\mathfrak{g}$  of a Lie group  $M$  is defined as the tangent space at the identity  $T_eM$ , where  $e$  is the identity element of the group (see section 16 in [286]).

Key components of neural networks – like mean, convolution, nonlinearities and batch normalization – can be defined on Riemannian manifolds as described in [288–292]. Optimization algorithms for Riemannian manifolds (gradient descent, SGD, Adam etc.) can be found in [293–300].

Commonly encountered examples of Riemannian manifolds in computer vision are the  $n$ -sphere  $S^n$ , the manifold of  $n \times n$  symmetric positive matrices  $P_n$ , the special orthogonal group  $SO(n)$  (rotation matrices), the special euclidean group  $SE(n)$  (rigid body transformations), Grassman manifold  $Gr(n, p)$  (collection of all  $p$ -dimensional linear subspaces in  $\mathbb{R}^n$ , see [301]) and the Stiefel manifold  $St(n, p)$  (collection of all  $p$ -dimensional orthogonal bases in  $\mathbb{R}^n$ ).

In the following, we will give an overview of manifold learning methods employed in important application fields in multimedia (similarity search, image classification, synthesis & enhancement, video analysis, 3D data processing, nonlinear dimension reduction) and about available open source software frameworks.

#### 5.4.2. Similarity search & retrieval

Image retrieval deals with searching for similar images in an image gallery, given a certain query image (see the surveys [302, 303]). Many methods employ for this *metric learning*, which transforms input images into *embeddings* ( $\approx$  feature vectors) and learns a distance function between these embeddings.

The authors of [304] propose *regularized ensemble diffusion* for refining/reranking the initial similarity search results. They show that regularized ensemble diffusion is significantly more robust against noise in the data than standard diffusion. A *diffusion* process [305] models the relationship between objects on a graph-based manifold, wherein similarity values are diffused along the geodesic path in an iterative way.

In [306] an unsupervised framework is presented for the identification of *hard training examples* for the training of an embedding. Hard training examples (both positive and negative samples) are identified by disagreement between euclidean and manifold similarities.

A time- and memory-efficient algorithm for estimating similarities on the data manifold is proposed in [307]. They adapt the random walk procedure to estimate manifold similarities only on a small number of data in each mini-batch, rather than on all training data.

The work of [308] proposes a unsupervised metric learning algorithm that learns a metric in a lower dimensional latent space using constraints provided as tuples, which rely on pseudo-labels obtained by a graph-based clustering method (*authority ascent shift*). The parameters of the approach are learned jointly using Riemannian optimization on a product manifold.

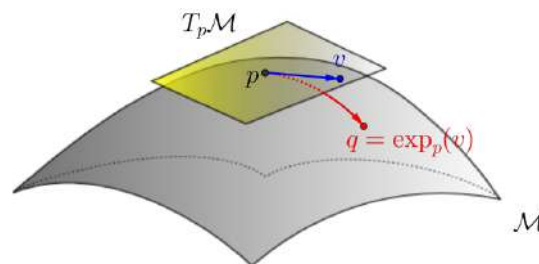


Figure 27. Tangent space and exponential map on a 2-dimensional manifold. Image courtesy of [287].



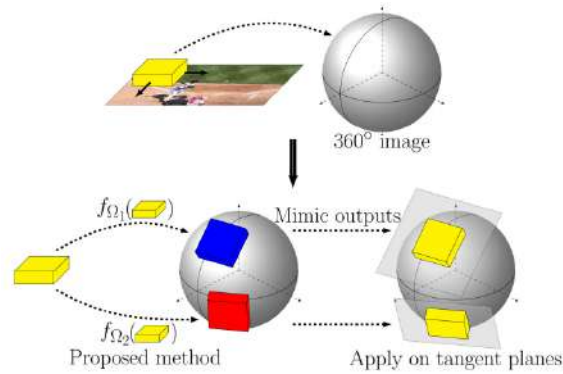


Figure 28. Transfer CNNs trained on flat images to 360° images with the method from [310].

#### 5.4.3. Image classification & object detection

The work [295] proposes a framework for the transformation of problems with manifold constraints into unconstrained problems on an Euclidean space through a mechanism they call *dynamic trivializations*. They show how to implement these trivializations efficiently for a large variety of commonly used matrix manifolds and provide a formula for the gradient of the matrix exponential.

The authors of [309] propose *manifold mixup*, a novel regularizer which forces the training to interpolate between hidden representations – captured in the intermediate layers of the network – of samples. It can be seen as a generalization of input mixup, which does the interpolation on a random layer of the network (whereas input mixup uses always layer 0). Experiments for the task of image classification show that manifold mixup flattens the class-specific representation (lower variance) and generates a smoother decision boundary.

An approach for few-shot image classification is presented in [311] which proposes *embedding propagation* as an unsupervised non-parametric regularizer. Embedding propagation leverages interpolation between the extracted features of a neural network, based on a similarity graph. Experiments show that embedding propagation yields a smoother embedding manifold and gives better performance on three standard datasets for few-shot image classification.

The work [310] introduces a knowledge distillation method which is able to transfer an existing CNN model trained on perspective images to *spherical* images captured with a 360° camera *without* any additional annotation effort (see Figure 28). They train a spherical Faster R-CNN model with this method, demonstrating that an object detector for spherical images (in equirectangular projection) can be trained without any annotations in the 360° images.

#### 5.4.4. Image synthesis & enhancement

For image synthesis and enhancement, state of the art algorithms employ either GANs (*generative adversarial networks*) [312] or *diffusion models* [313].

The authors of [314] show that current solvers employed in diffusion models throw the generative sample path off the data manifold, causing the error to accumulate. They propose an additional correction term inspired by the manifold constraint to force the iterations to be close to the data manifold. The proposed manifold constraint is easy to add to a solver, yet boosts its performance significantly.

A method for comparing data manifolds based on their topology is presented in [315]. They introduce novel tools, specifically *cross-barcode* and *manifold topology divergence score*, which are



Figure 29. From left to right: Content image, style image, style-transferred image [316].

able to track spatial discrepancies between manifolds on multiple scales. They apply it to assess the performance of generative models in various domains (images, 3D shapes or time series) and demonstrate that these tools are able to detect common problems of GAN-based image synthesis like mode dropping, mode collapse and image disturbance.

The work [316] proposes *progressive attentional manifold alignment* for style transfer, which progressively aligns content manifolds to their most related style manifolds. Afterwards, *space-aware interpolation* is performed in order to increase the structural similarity of the corresponding manifolds, which makes it easier for the attention module to match features between them. Experiments show that the method generates high-quality style-transferred images (see Figure 29).

The FLAME algorithm proposed in [317] performs highly realistic image manipulations (e.g. changing expression, hair style or age of a synthetic face, see Figure 30) with minimal supervision. It estimates linear latent directions in the latent space of *StyleGAN2* using only a few image pairs and introduces a novel method for sampling from the attribute style manifold.

#### 5.4.5. Video analysis

Most manifold learning methods for video analysis deal with the important task of human action recognition. Often they employ neural networks over the manifold  $P_n$  of symmetric positive matrices (usually covariance matrices) for this.

The authors of [289] propose a *dilated convolution* operator on manifolds, based on the *weighted Frechet mean* [288], as well as a *residual connection* operator. Both are important building blocks of modern neural networks. They construct a manifold-valued network employing covariance matrices (calculated from CNN features) and train this network for human action detection on the UCF-11 video dataset.

In [290] the convolution is defined as the weighted sum (reprojected to the manifold) in the tangent space  $T_a M$ , where  $a$  is the Frechet mean of the input points for the convolution. They show that their proposed convolution operator is an isometry of the manifold, which corresponds to the translation-invariance property of the convolution in an Euclidean space.

The algorithm [318] adopts a neural network over the manifold  $P_n$  of symmetric positive definite matrices as the backbone and appends a cascade of *Riemannian autoencoders* to it in order to enrich the information flow within the network. Experiments on the tasks of emotion recognition, hand action recognition and human action recognition demonstrate a favourable performance compared to state of the art methods.

#### 5.4.6. 3D data processing

The work [319] proposes a novel algorithm for geometric disentanglement (separate intrinsic and extrinsic geometry) of 3D models, based on the fundamental theorem for surfaces. They describe





surface features via a combination of *conformal factors* and surface normal vectors and propose a convolutional mesh autoencoder based on these features. The conformal factor defines a conformal (angle-preserving) deformation between two manifolds. The algorithm achieves state-of-the-art performance on 3D surface generation, reconstruction and interpolation tasks (see Figure 31).

The authors of [320] propose an approach for learning generative models on manifolds by minimizing the *probability path divergence*. Unlike other continuous flow approaches, it does not require solving an ordinary differential equation during training.

In [321] a method for rotation (pose) estimation of 3D objects from point clouds and images is presented. For this, they propose a novel *manifold-aware* gradient in the backward pass of rotation regression that directly updates the neural network weights.

The work [322] introduces *intrinsic neural fields*, a novel and versatile representation for neural fields on manifolds. Intrinsic neural fields are based on the eigenfunctions of the *Laplace-Beltrami* operator, which can represent detailed surface information directly on the manifold. Furthermore, they extend *neural tangent kernel analysis* to manifolds for better insight into the spectral properties of neural fields.

#### 5.4.7. Nonlinear dimension reduction

Many real world high-dimensional datasets are actually lying in a low-dimensional manifold (*manifold hypothesis*). Nonlinear dimensional reduction algorithms project high-dimensional data onto such a low-dimensional manifold, while trying to preserve distance relationships in the original high-dimensional space as good as possible.

Classical approaches for nonlinear dimension reduction are Isomap, Local Linear Embedding (LLE) and Laplacian Eigenmaps (see the survey in [323]). In recent years, more powerful approaches like *t-SNE*, *UMAP*, *TriMAP* and *PaCMAP* have emerged [324]. From these, PaCMAP seems to preserve best both the global and local structure of the high-dimensional data.

In [325], the *h-NNE* algorithm is proposed, which is competitive with t-SNE and UMAP in quality while being on order of magnitude faster. The significant runtime advantage is possible as h-NNE avoids solving an optimization problem and relies on *nearest neighbor graphs* instead.

The *SpaceMAP* algorithm [326] (see Figure 32) introduces the concept of *equivalent extended distance*, which makes it possible to match the capacity between two spaces of different dimensionality. Furthermore, *hierarchical manifold approximation* is performed based on the observation that real-world data has often a hierarchical structure.

The *DIPOLE* algorithm proposed in [327] corrects an initial embedding (e.g. calculated via Isomap) by minimizing a loss functional with both a local, metric term and a global, topological term based on *persistent homology*. Unlike more ad hoc methods for measuring the shape of data



Figure 30. Image editing with FLAME [317].





at multiple scales, persistent homology is rooted in algebraic topology and enjoys strong theoretical foundations.

For measuring the intrinsic dimension of a data distribution, in [328] a method is presented based on recent progress in likelihood estimation in high dimensions via *normalizing flows*.

#### 5.4.8. Relevant publications

The survey has been submitted to MVA Conference 2023 and is currently under review.

### 5.5. Manifold mixing soups for better out-of-distribution performance

**Contributing partners:** JR

#### 5.5.1. Introduction and methodology

Large pretrained visual foundation models like CLIP [5] or CoCa [6] got very popular recently due to their great performance for a variety of computer vision tasks, either as zero-shot learner (without finetuning) or serving as a base for task-specific finetuning on a smaller dataset.

Typically, multiple models are finetuned with different hyperparameters (like learning rate, weight decay or data augmentation strategy), using the same pretrained model as initialization. From those, the model with the best accuracy on the validation dataset is selected. Unfortunately, this procedure leaves out important information which has been learned in the latent space manifolds (individual layers or a collection of layers) of the remaining finetuned models. As shown in [329], even fusing multiple finetuned models in a very straightforward way by averaging them makes the fused model already significantly more robust to distribution shifts in the data.

Motivated by this, we propose the *manifold mixing model soup* (*ManifoldMixMS*) algorithm. Instead of simple averaging, it uses a more sophisticated strategy to generate the fused model. Specifically, it partitions a neural network model into several latent space manifolds (which can be individual layers or a collection of layers). Afterwards, from the pool of finetuned models available after hyperparameter tuning, the most promising ones are selected and their latent space manifolds are mixed together individually. The optimal mixing coefficient for each latent space manifold is calculated automatically via invoking an optimization algorithm. The fused model we retrieve with this procedure can be thought as sort of a "Frankenstein" model, as it integrates (parts of) individual model components from multiple finetuned models into one model.

In the following, we outline the proposed algorithm for generating a fused model – the *manifold mixing model soup* – from its ingredients (the finetuned models after hyperparameter tuning). The algorithm pseudocode can be seen in Algorithm 1.



Figure 31. Generated 3D models with the geometric disentanglement algorithm from [319].



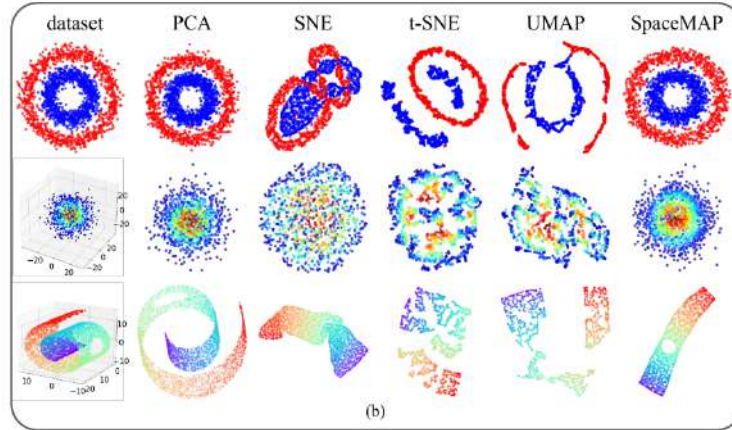


Figure 32. Comparison of classic nonlinear dimension reduction methods with SpaceMAP [326].

We first sort all  $n$  finetuned models  $\theta_i$  (with  $i = 0, \dots, n - 1$ ) in descending order, based on their validation accuracy  $ValAcc(\theta_i)$  on the original dataset which was used for finetuning. So  $\theta_0$  is the model (to be precise, its finetuned parameters) with the highest validation accuracy, whereas  $\theta_{n-1}$  is the one with the lowest validation accuracy.

Each model  $\theta_i$  is partitioned into  $m$  components  $\theta_i^j$ , where  $\theta_i^j$  corresponds to a single latent space manifold, and  $j = 1, \dots, m$ . Each latent space manifold comprises either a single layer or a collection of layers, corresponding to one building block of the model. Typically, we partition a model into 10 – 30 components.

The fused model  $\Psi$  is now calculated in an sequential way, by iteratively mixing promising ingredient models with it. At first, the fused model is set to the best finetuned model via  $\Psi = \theta_0$ , and the variable  $k$ , which counts the number of models which have been mixed so far into the fused model, is set to 1.

In each iteration (for  $i = 1, \dots, n - 1$ ), we try now to mix the candidate model  $\theta_i$  with the current fused model  $\Psi$  in an optimal way, with the aim of increasing the validation accuracy of the updated fused model  $\Psi'$  (which includes  $\theta_i$ ) on the original dataset.

In order to save computation time, we skip the optimization step for a candidate model  $\theta_i$  for which it is unlikely that we get an increase in the validation accuracy by mixing  $\theta_i$  into the current fused model  $\Psi$ . For that, we generate the "approximate average" model  $\tilde{\Psi}$  via

$$\tilde{\Psi} = \frac{k}{k+1} \cdot \Psi + \frac{1}{k+1} \cdot \theta_i \quad (14)$$

and test whether the condition  $ValAcc(\tilde{\Psi}) > \tau \cdot ValAcc(\Psi)$  is fulfilled. If so, we continue with this iteration. If it is not fulfilled, we skip the following steps of this iteration, so candidate model  $\theta_i$  will not be taken into account. The motivation for the specific combination provided in Eq. (14) is that  $\tilde{\Psi}$  calculated in this way corresponds approximately to the *average* of all candidate models (like in [329]) which have been mixed so far into the fused model (including  $\theta_i$ ), *if we assume* that the optimization did not change the mixing coefficients drastically from their provided initial values. We set the constant  $\tau$  to 0.998.

Having identified  $\theta_i$  as a promising candidate model, in the next step we determine the optimal factors for mixing its latent space manifolds into the current fused model  $\Psi$ . For this, we define the updated fused model  $\Psi'(\lambda)$  as a *component-wise* convex combination of  $\Psi$  and  $\theta_i$  via

$$\Psi'(\lambda)^j = \lambda^j \cdot \Psi^j + (1 - \lambda^j) \cdot \theta_i^j \quad (15)$$







for all components  $j = 1, \dots, m$ . Note that  $\Psi'(\lambda)$  is a function of the mixing vector  $\lambda$ . The mixing factor  $\lambda^j \in [0, 1]$  determines how much of the  $j$ -th component (latent space manifold) of the candidate model  $\theta_i$  is mixed into the current fused model  $\Psi$ . The component-wise convex combination of the two models allows an optimizer to explore the latent space manifolds of the models  $\Psi$  and  $\theta_i$  in a very flexible way, in order to find the optimal mixing vector  $\lambda^* \in \mathbb{R}^m$  which gives the highest validation accuracy for the updated fused model  $\Psi'$ .

For the subsequent optimization step, we set up the optimization problem to solve as

$$\lambda^* = \arg \max_{\lambda \in [0,1]^m} (\text{ValAcc}(\Psi'(\lambda))) \quad (16)$$

where  $[0, 1]^m$  is the  $m$ -dimensional unit interval. Via the constraint  $\lambda \in [0, 1]^m$  we ensure that a convex combination is done for each component, so we are in fact *interpolating linearly* between the latent space manifolds  $\Psi^j$  and  $\theta^j$ . The model  $\Psi'(\lambda)$  can be calculated via Eq. (15).

For solving this optimization problem, we employ the *Nevergrad*<sup>2</sup> optimization package. It provides a large variety of black-box *derivative-free* optimization algorithms together with a sophisticated heuristic [330] to select the best optimizer based on the characteristic (number of variables, allowed budget for function evaluations etc.) of the optimization problem. As the initial value for the mixing factors, we set  $\lambda^j = k/(k+1)$  for  $j = 1, \dots, m$  with a similar motivation as explained earlier for Eq. (14).

We invoke now the optimizer in order to calculate the optimal mixing vector  $\lambda^*$  which give the highest validation accuracy on the dataset used for finetuning. The optimal updated fused model can be calculated now via  $\Psi'^* = \Psi'(\lambda^*)$ .

After iterating over all candidate models  $\theta_i$  for  $i = 1, \dots, n-1$  we retrieve a final fused model  $\Psi$  (the *manifold mixing model soup*), which mixes together the  $k$  selected candidate models / ingredients in an optimal way.

### 5.5.2. Experiments and Evaluation

The setup for our experiments is very similar to the one for the vision models given in the *model soup* paper [329]. We summarize it in the following for clarity and completeness.

The model employed for finetuning is the *CLIP* model [5]. CLIP is a powerful multi-modal zero-shot neural network, which has been pretrained with contrastive learning on a huge dataset of image-text pairs. Finetuning of the pretrained model is performed end-to-end (all parameters are modified), as it typically leads to better performance than training only the final linear layer. Before finetuning, the final layer is initialized with a linear probe as described in [331]. The loss function employed for finetuning is the cross-entry loss.

The original dataset employed for finetuning is *ImageNet* [332]. Since the official ImageNet validation dataset is typically used as the test dataset, we use roughly 2% of the ImageNet training dataset as held-out validation dataset for calculating the validation accuracy in our proposed algorithm (see Algorithm 1).

For measuring the OOD performance (robustness to distribution shifts) of our proposed algorithm, we employ five datasets derived from ImageNet with natural (not synthetically generated) distribution shifts. They corresponds to datasets with naturally occurring variations of the data samples due to different lighting, viewpoint, geographic location, image style (e.g. sketch instead of photo), crowdsourcing and more. The five datasets with distribution shifts we use are:

- ImageNet-V2 (IN-V2) [333] is a reproduction of the ImageNet test set with distribution shift. The dataset was collected by closely following the original labelling protocol.

<sup>2</sup><https://facebookresearch.github.io/nevergrad/>






---

**Algorithm 1** Manifold mixing model soup algorithm
 

---

**Require:** Finetuned models  $\{\theta_0, \dots, \theta_{n-1}\}$  as result of hyperparameter tuning  
**Require:** Partitioning of a model  $\zeta$  into  $m$  components (latent space manifolds)  $\zeta^j$  for  $j = 1, \dots, m$   
**Require:** Function  $ValAcc(\zeta)$  which calculates validation accuracy for  $\zeta$  on dataset

```

 $\{\theta_0, \dots, \theta_{n-1}\} \leftarrow sort(\{\theta_0, \dots, \theta_{n-1}\})$ 
 $k \leftarrow 1$ 
 $\Psi \leftarrow \theta_0$ 
 $\tau \leftarrow 0.998$ 
for  $i = 1, \dots, n - 1$  do
   $\tilde{\Psi} = \frac{k}{k+1} \cdot \Psi + \frac{1}{k+1} \cdot \theta_i$ 
  if  $ValAcc(\tilde{\Psi}) > \tau \cdot ValAcc(\Psi)$  then
     $\Psi'(\lambda)^j = \lambda^j \cdot \Psi^j + (1 - \lambda^j) \cdot \theta_i^j$ 
     $\lambda^* = \arg \max_{\lambda \in [0,1]^m} (ValAcc(\Psi'(\lambda)))$ 
     $\Psi'^* = \Psi'(\lambda^*)$ 
    if  $ValAcc(\Psi'^*) > ValAcc(\Psi)$  then
       $k \leftarrow k + 1$ 
       $\Psi \leftarrow \Psi'^*$ 
    end if
  end if
end for
return  $\Psi$ 

```

---

- ImageNet-R (IN-R) [334] contains renditions (e.g., sculptures, paintings) for 200 ImageNet classes.
- ImageNet-Sketch (IN-Sketch) [335] contains sketches instead of natural images. It contains only sketches in "black-and-white" color scheme.
- ObjectNet [336] provides objects in various scenes with 113 classes overlapping with ImageNet.
- ImageNet-A (IN-A) [337] is a test set of natural images misclassified by a ResNet-50 model for 200 ImageNet classes.

See Figure 33 for an illustration of samples for each of the datasets with natural distribution shifts. For all datasets (the original used for finetuning and the ones with distribution shifts), we take the top-1 accuracy on the respective test set for measuring the performance of a model. We calculate the overall out-of-distribution performance of a model as the average of its test accuracy over all five datasets with distribution shifts.

We partition the CLIP ViT-B/32 model into 8, 15 and 26 components. A too fine partitioning (e.g. one component for each layer of the model) makes the optimization much more difficult, whereas a too coarse partitioning provides not enough flexibility for mixing the latent space manifolds individually in an optimal way. The structure of the partitioning is done roughly according to the hierarchy of the building blocks of the CLIP model. We denote the respective variant of our proposed algorithm with 8, 15 and 26 components as ManifoldMixMS-C8, ManifoldMixMS-C15 and ManifoldMixMS-C26.

We parametrize the Nevergrad optimizer with an maximum budget for the number of function evaluations (of the objective function to optimize) of roughly 250 function evaluations for all ManifoldMixMS variants. The employed optimizer is automatically selected by the Nevergrad optimization package (see [330]).



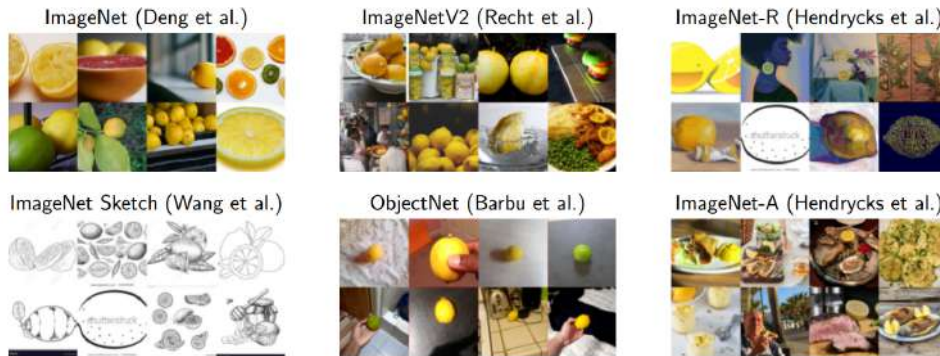


Figure 33. Samples for class lemon, from the original ImageNet dataset and the five datasets with natural distribution shifts. Image courtesy of [338]

For the evaluation of our proposed manifold mixing model soup algorithm, we compare mainly with the *greedy soup* and *uniform soup* algorithms which have been proposed in [329]. Additionally, we compare our proposed algorithm also against ensemble models.

The scatterplot in Figure 34 shows how our proposed ManifoldMixMS-C8 algorithm (the overall best variant) performs compared to the greedy soup and uniform soup algorithm from [329] and to the individual finetuned models.

Furthermore, Table 20 gives a detailed evaluation of our proposed variants of the manifold mixing soup algorithm with 8, 15 and 26 on the five datasets with distribution shifts (ImageNet-V2, ImageNet-R, ImageNet-Sketch, ObjectNet, ImageNet-A) as well as on the original dataset used for finetuning (ImageNet).

One can see clearly from the scatterplot that our proposed manifold mixing model soup (especially the preferred variant with 8 components) algorithm combines the best properties of the uniform model soup and greedy soup algorithm. Specifically, it has practically the same good out-of-distribution accuracy as the uniform soup algorithm and still keeps the good accuracy of the greedy soup algorithm on the original ImageNet dataset. In contrast, the uniform soup algorithm performs on the original ImageNet dataset even worse than the best individual finetuned model.

It is significantly better with respect to the best finetuned model both on the datasets with distribution shifts (+3.5%), but also on the original ImageNet dataset (+0.6%). The difference grows even bigger when comparing with the second-best finetuned model.

### 5.5.3. Conclusion

We propose the *manifold mixing model soup* algorithm, which mixes together the latent space manifolds of multiple finetuned models in an optimal way in order to generate a fused model. Experiments show that the fused model gives significantly better out-of-distribution performance when finetuning a CLIP model for image classification.

### 5.5.4. Relevant publications

The publication has been submitted for IMVIP 2023 and is currently under review.



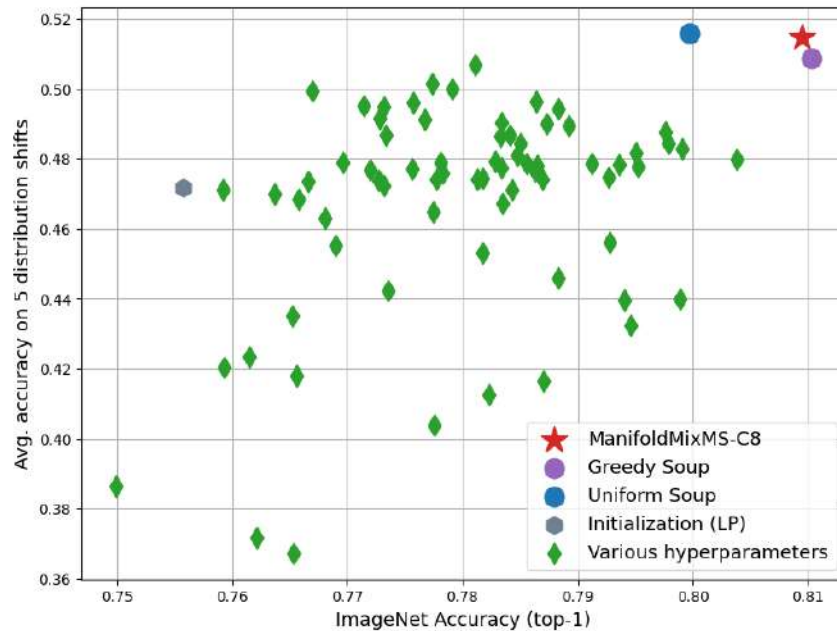


Figure 34. Comparison of our proposed manifold mixing soup algorithm (with 8 components) against greedy soup and uniform soup algorithm from [329] and the individual finetuned models.

### 5.5.5. Relevant software/datasets/other outcomes

A Python implementation of the ManifoldMixMS algorithm will be made available at <https://github.com/hfassold/ManifoldMixMS> as soon as the publication gets accepted.

### 5.5.6. Relevance to AI4media use cases and media industry applications

The ManifoldMixMS algorithm for improving the out-of-distribution performance of a model can be employed for all AI4Media use cases where a training and subsequent hyperparameter tuning is done – e.g. in UC6 (BSC) that is dealing with media generation.

## 5.6. Sparse to Dense Dynamic 3D Facial Expression Generation

**Contributing partners:** UNIFI

### 5.6.1. Introduction and methodology

Synthesizing dynamic 3D (4D) facial expressions aims at generating realistic face instances with varying expressions or speech-related movements that dynamically evolve across time, starting from a face in neutral expression. It finds application in a wide range of graphics applications spanning from 3D face modeling, to augmented and virtual reality for animated films and computer games. While recent advances in generative neural networks have made possible the development of effective solutions that operate on 2D images [339, 340], the literature on the problem of generating facial animation in 3D is still quite limited.

To perform a faithful and accurate 3D facial animation, three main challenges arise. First, the identity of the subject whose neutral face is used as starting point for the sequence should



Table 20. Detailed comparison of our proposed manifold mixing soup algorithm variants for the CLIP ViT-B/32 neural network with the best and second-best finetuned model, the model soup algorithms from [329] and for completeness also with Ensemble methods. The top-1 accuracy (in %) on the respective test dataset is employed. The column "Avg OOD" corresponds to the average over all 5 datasets with distribution shifts. The best and second-best result for each dataset (without taking into account the Ensemble methods as they have a much higher computational cost) is marked in red and blue.

Method	ImageNet	IN-V2	IN-R	IN-Sketch	ObjectNet	IN-A	Avg OOD
Best finetuned model	80.38	68.44	44.51	60.63	42.62	23.64	47.97
Second-best finetuned model	79.89	67.91	41.49	54.58	37.98	18.01	44.01
Uniform soup	79.97	68.51	47.71	<b>66.54</b>	<b>45.95</b>	<b>29.17</b>	<b>51.57</b>
Greedy soup	<b>81.03</b>	69.55	47.77	64.20	44.90	27.89	50.86
ManifoldMixMS-C8	<b>80.95</b>	<b>69.67</b>	<b>48.15</b>	<b>64.81</b>	45.66	<b>29.06</b>	<b>51.47</b>
ManifoldMixMS-C15	80.80	<b>69.61</b>	47.89	64.76	44.45	28.39	51.02
ManifoldMixMS-C26	80.85	69.58	<b>48.04</b>	64.79	<b>45.75</b>	28.88	51.41
Ensemble	81.19	–	–	–	–	–	50.77
Greedy ensemble	81.90	–	–	–	–	–	49.44

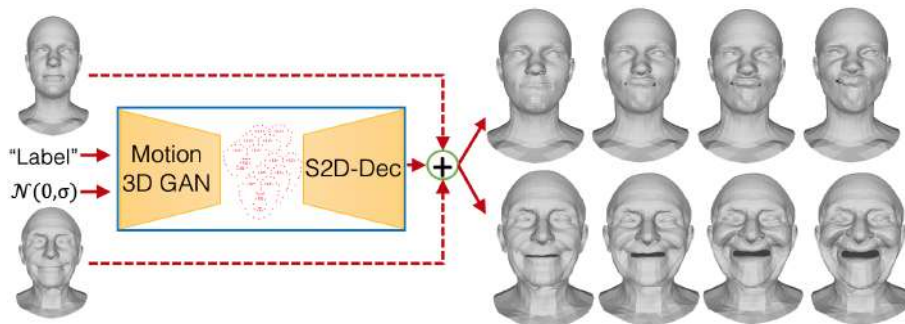


Figure 35. 3D dynamic facial expression generation: A GAN generates the motion of 3D landmarks from an expression label and noise; A decoder expands the animation from the landmarks to a dense mesh, while keeping the identity of a neutral 3D face,

be maintained across time. Second, the applied deformation should correspond to the specified expression/motion that is provided as input, and should be applicable to any neutral 3D face. Incidentally, these are major challenges in 3D face modeling, which require disentangling structural face elements related to the identity, e.g., nose or jaw shape, from deformations related to the movable face parts, e.g., mouth opening/closing. Finally, it is required to model the temporal dynamics of the specified expression so to obtain realistic animations.

Some previous works tackled the problem by capturing the facial expression of a subject frame-by-frame and transferring it to a target model [341]. However, in this case the temporal evolution is not explicitly modeled, so the problem reduces to transferring a tracked expression to a neutral 3D face. Some other works animated a 3D face mesh given an arbitrary speech signal and a static 3D face mesh as input [342, 343]. Also in this case, the temporal evolution is guided by an external input, similar to a tracked expression. Instead, here we are interested in animating a face just





starting from a neutral face and an expression label.

In our solution, which is illustrated in Figure 35, the temporal evolution and the mesh deformation are decoupled and modeled separately in two network architectures. A manifold-valued GAN (*Motion3DGAN*) accounts for the expression dynamics by generating a temporally consistent motion of 3D landmarks corresponding to the input label from noise. The landmarks motion is encoded using the Square Root Velocity Function (SRVF) and compactly represented as a point on a hypersphere. Then, a Sparse2Dense mesh Decoder (*S2D-Dec*) generates a dense 3D face guided by the landmarks motion for each frame of the sequence. To effectively disentangle identity and expression components, the landmarks motion is represented as a per-frame displacement from a neutral configuration. Instead of directly generating a mesh, the S2D-Dec expands the landmarks displacement to a dense, per-vertex displacement, which is finally used to deform the neutral mesh. The intuition that led to this architecture is the following: the movement induced on the face surface by the underlying facial muscles is consistent across subjects. In addition, it causes the vertex motion to be locally correlated as muscles are smooth surfaces. We thus train the decoder to learn how the displacement of a sparse set of points influences the displacement of the whole face surface. This has the advantage that structural face parts, *e.g.*, nose or forehead, which are not influenced by facial expressions are not deformed, helping in maintaining the identity traits stable. Furthermore, the network can focus on learning expressions at a fine-grained level of detail and generalize to unseen identities.

Figure 36 provides some additional details on the two specialized networks that compose our architecture. Motion3DGAN accounts for the temporal dynamics and generates the motion of a sparse set of 3D landmarks from noise, provided an expression label, *e.g.*, happy, angry. The motion is represented as per frame landmark displacements with respect to a neutral configuration. These displacements are fed to a decoder network, S2D-Dec, that constructs the dense point-cloud displacements from the sparse displacements given by the landmarks. These dense displacements are then added to a neutral 3D face to generate a sequence of expressive 3D faces corresponding to the initial expression label. In the following, we separately describe the two networks.

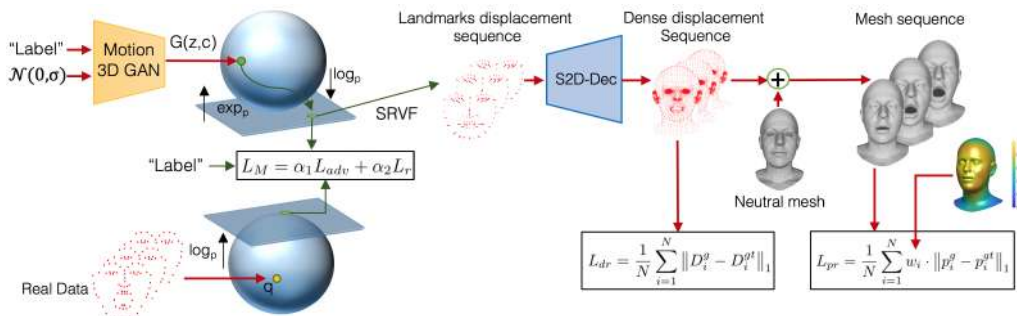


Figure 36. **Overview of our framework.** *Motion3DGAN* generates the motion  $q(t)$  of 3D landmarks corresponding to an expression label from a noise vector  $z$ . The module is trained guided by a reconstruction loss  $L_r$  and adversarial loss  $L_{adv}$ . The motion  $q(t)$  is converted to a sequence of landmark displacements  $d_i$ , which are fed to *S2D-Dec*. From each  $d_i$ , the decoder generates a dense displacement  $D_i^g$ . A neutral mesh is then summed to the dense displacements to generate the expressive meshes  $S^g$ . *S2D-Dec* is trained under the guidance of a displacement loss  $L_{dr}$  and our proposed weighted reconstruction loss  $L_{pr}$ .

### 5.6.2. Experimental results

We validated the proposed method in a broad set of experiments on two publicly available benchmark datasets.





**CoMA dataset** [344]: It is a common benchmark employed in other studies [344, 345]. It consists of 12 subjects, each one performing 12 extreme and asymmetric expressions. Each expression comes as a sequence of meshes  $\mathbf{S} \in \mathbb{R}^{N \times 3}$  (140 meshes on average), with  $N = 5,023$  vertices.

**D3DFACS dataset** [346]: We used the registered version of this dataset [124], which has the same topology of CoMA. It contains 10 subjects, each one performing a different number of facial expressions. In contrast to CoMA, this dataset is labeled with the activated action units of the performed facial expression. It is worthy to note that the expressions of D3DFACS are highly different from those in CoMA.

**5.6.2.1. 3D Expression Generation** For evaluation, we set up a baseline by first comparing against standard 3DMM-based fitting methods. Similar to previous works [347, 348], we fit  $\mathbf{S}^n$  to the set of target landmarks  $Z^e$  using the 3DMM components. Since the deformation is guided by the landmarks, we first need to select a corresponding set from  $\mathbf{S}^n$  to be matched with  $Z^e$ . Given the fixed topology of the 3D faces, we can retrieve the landmark coordinates by indexing into the mesh, *i.e.*,  $Z^n = \mathbf{S}^n(\mathbf{I}_z)$ , where  $\mathbf{I}_z \in \mathbb{N}^n$  are the indices of the vertices that correspond to the landmarks. We then find the optimal deformation coefficients that minimize the Euclidean error between the target landmarks  $Z^e$  and the neutral ones  $Z^n$ , and use the coefficients to deform  $\mathbf{S}^n$ . In the literature, several 3DMM variants have been proposed. We experimented the standard PCA-based 3DMM and the DL-3DMM in [347]. We chose this latter variant as it is conceptually similar to our proposal, being constructed by learning a dictionary of deformation displacements. For fair comparison, we built the two 3DMMs using a number of deformation components comparable to the size of the S2D-Dec input, *i.e.*,  $68 \times 3 = 204$ . For PCA, we used either 38 components (retaining the 99% of the variance) and 220, while for DL-3DMM we used 220 dictionary atoms.

With the goal of comparing against other deep models, we also considered the Neural3DMM [345]. It is a mesh auto-encoder tailored for learning a non-linear latent space of face variations and reconstructing the input 3D faces. In order to compare it with our model, we modified the architecture and trained the model to generate an expressive mesh  $\mathbf{S}^g$  given its neutral counterpart as input. To do so, we concatenated the landmarks displacement (of size 204) to the latent vector (of size 16) and trained the network towards minimizing the same  $L_{pr}$  loss used in our model. All the compared methods were trained on the same data. Finally, we also identified the FLAME model [348]. Unfortunately, the training code is not available, and using the model pre-trained on external data would not be a fair comparison.

The mean per-vertex Euclidean error between the generated meshes and their ground truth is used as standard performance measure, as in the majority of works [344, 345, 349, 350]. Note that we exclude the Motion3DGAN model here as we do not have the corresponding ground-truth for the generated landmarks (they are generated from noise). Instead, we make use of the ground truth motion of the landmarks.

**5.6.2.2. Comparison with Other Approaches** Table 21 shows a clear superiority of S2D-Dec over state-of-the-art methods for both the protocols and datasets, proving its ability to generate accurate expressive meshes close to the ground truth in both the case of unseen identities or expressions. In Figure 37, the cumulative per-vertex error distribution on the expression-independent splitting further highlights the precision of our approach, which can reconstruct 90%-98% of the vertices with an error lower than  $1mm$ . While other fitting-based methods retain satisfactory precision in both the protocols, we note that the performance of Neural3DMM [345] significantly drop when unseen identities are considered. This outcome is consistent to that reported in [349], in which the low generalization ability of these models is highlighted. We also note that results for the identity-independent protocol were never reported in the original papers [344, 345]. Overall, our





Method	Expression Split		Identity Split	
	CoMA	D3DFACS	CoMA	D3DFACS
PCA-220	$0.76 \pm 0.73$	$0.42 \pm 0.44$	$0.80 \pm 0.73$	$0.56 \pm 0.56$
PCA-38	$0.90 \pm 0.84$	$0.44 \pm 0.45$	$0.93 \pm 0.82$	$0.58 \pm 0.56$
DL3DMM [347]	$0.86 \pm 0.80$	$0.73 \pm 1.15$	$0.89 \pm 0.79$	$1.15 \pm 1.50$
Neural [345]	$0.75 \pm 0.85$	$0.59 \pm 0.86$	$3.74 \pm 2.34$	$2.09 \pm 1.37$
<b>Ours</b>	<b><math>0.52 \pm 0.59</math></b>	<b><math>0.28 \pm 0.31</math></b>	<b><math>0.55 \pm 0.62</math></b>	<b><math>0.27 \pm 0.30</math></b>

Table 21. Reconstruction error (mm) on expression-independent (left) and identity-independent (right) splits: comparison with PCA- $k$  3DMM ( $k$  components), DL-3DMM (220 dictionary atoms), and Neural3DMM.

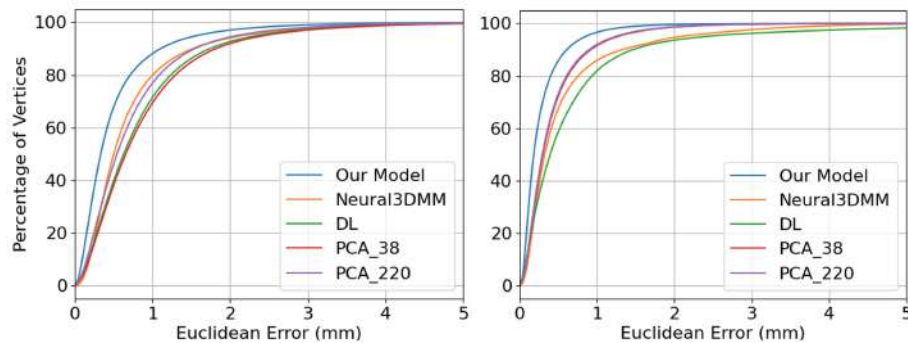


Figure 37. Cumulative per-vertex Euclidean error between PCA-based 3DMM models, DL-3DMM, Neural3DMM, and our proposed model, using expression-independent cross-validation on the CoMA (left) and D3DFACS (right) datasets.

solution embraces the advantages of both approaches, being as general as fitting solutions yet more accurate.

Figure 38 shows some qualitative examples by reporting error heatmaps in comparison with PCA, DL-3DMM [347] and Neural3DMM [345] for the identity-independent splitting. The ability of our model as well as PCA and DL-3DMM to preserve the identity of the ground truth comes out clearly, in accordance with the results in Table 21. By contrast, Neural3DMM shows high error even for the neutral faces, which proves its inability to keep the identity of an unseen face. Indeed, differently from to the other methods, Neural3DMM encodes the neutral face in a latent space and predicts the 3D coordinates of the points directly, which introduces some changes on the identity of the input face. This evidences the efficacy of our S2D-Dec, that instead learns per-point displacements instead of point coordinates.

### 5.6.3. Conclusions

As a main contribution of this research, we proposed a novel framework for dynamic 3D expression generation from an expression label, where two decoupled networks separately address modeling the motion dynamics and generating an expressive 3D face from a neutral one. We demonstrated the improvement with respect to previous solutions, and showed that using landmarks is effective in modeling the motion of expressions and the generation of 3D meshes. We also identified two main limitations: first, our S2D-Dec generates expression-specific deformations, and so cannot model identities. Moreover, while Motion3DGAN can generate diverse expressions and allows interpolating





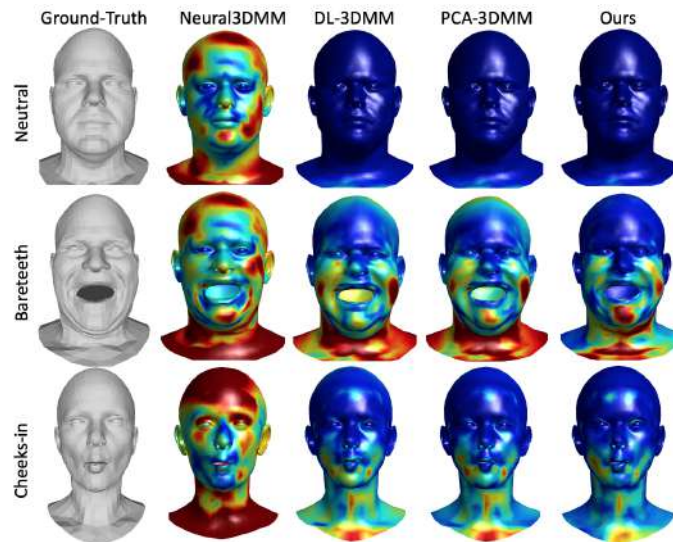


Figure 38. Mesh reconstruction error (red=high, blue=low) of our model and other methods.

on the sphere to obtain complex facial expressions, the samples are of a fixed length (*i.e.*, 30 meshes, from neutral to apex). However, as shown in the applications, S2D-Dec can deal with motion of any length since it is independent from Motion3DGAN.

#### 5.6.4. Relevant publications

- Naima Otberdout, Claudio Ferrari, Mohamed Daoudi, Stefano Berretti, Alberto Del Bimbo. “Sparse to Dense Dynamic 3D Facial Expression Generation.” IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 20385-20394 [351]. Zenodo record: <https://zenodo.org/record/6396131>.

#### 5.6.5. Relevant software/datasets/other outcomes

- The PyTorch implementation of our S2D-Dec and Motion3dGAN architectures can be found in <https://github.com/CRISTAL-3DSAM/Sparse2Dense>.

#### 5.6.6. Relevance to AI4media use cases and Media Industry Applications

Our Motion3DGAN tool contributes and provides solution to the general cases where a temporal smooth trajectory should be generated. The S2D-Dec provides a solution to all those cases where a static 3D mesh should be generated from a sparse set of points (e.g., landmarks or joints). This can find application in several different contexts, like human-avatar interaction, film industry, etc. The proposed solution can be part of a talking head generation solution that can be highly useful in generating new content in media industry (for news, movies, virtual assistants, etc.).

### 5.7. Conditioned Image Retrieval for Fashion using Contrastive Learning and CLIP-based Features

**Contributing partners:** UNIFI



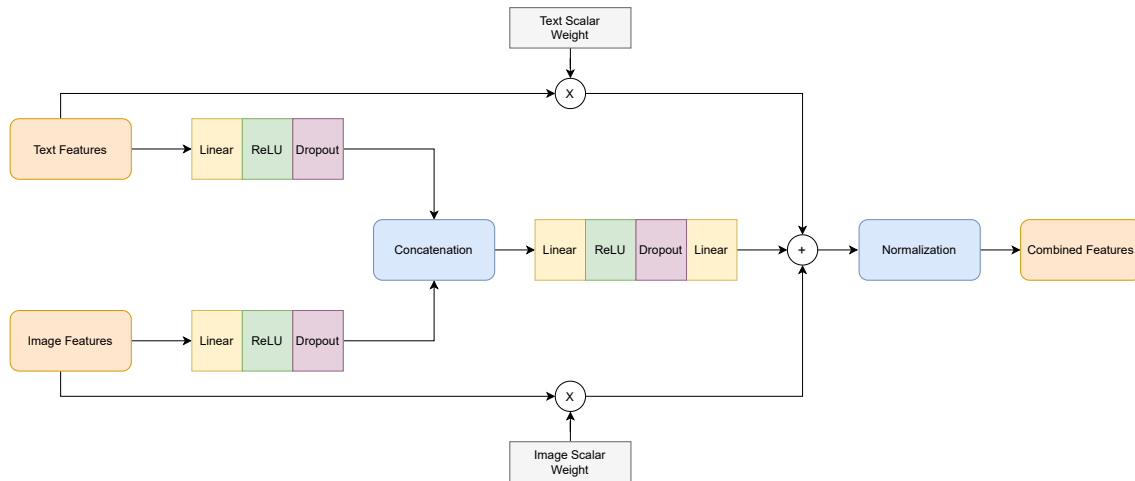


Figure 39. Architecture of the combiner network.

### 5.7.1. Introduction

Content-Based Image Retrieval (CBIR) is a fundamental task in multimedia and computer vision and has been applied to many different domains like art [352], commerce [353], medicine [354], security [355], nature [356], landmarks [357], etc. Typically image features of the database images are computed and compared with the features of a query image.

Interactive (i.e. conditioned) image retrieval systems extend CBIR systems to improve their effectiveness, by adding some form of user feedback, e.g. to provide some measure of relevance [358] or requesting constraints on some attributes of the retrieved results [359]. These types of systems can be applied in many different domains such as web search, e-commerce and surveillance. However, the difficulty in the development of these approaches is the need to incorporate features from the feedback and the intent of the user, in addition to the semantic gap between features and image content.

Very recently, it has been shown that a deep neural network like CLIP [360], trained using an image-caption alignment objective on large-scale internet data, can obtain impressive zero-shot transfer on a myriad of downstream tasks like image classification, text-based image retrieval, object detection and video action recognition.

In this work we show that CLIP-based features can be effectively used to implement a conditioned image retrieval system where user feedback is provided as natural language input to provide additional (or contrasting) requirements with respect to those embedded in the visual features of the image used to query the system. In this context a user selects a reference image and then provides additional requirements and requests in form of text, e.g. asking to change texture or shape features of the reference image. We apply the system to the fashion domain. Unluckily there not yet many datasets to evaluate the methods tackling this retrieval task, so in the following we report results obtained on the challenging FashionIQ dataset [361] obtaining state-of-the-art results; of course the approach is applicable to domains that are different from fashion.

### 5.7.2. Previous work

Traditional CBIR did not use any kind of user feedback or its intent to refine results. However, within interactive and conditioned CBIR, a lot of work has been done to improve retrieval performance



incorporating user's feedback in terms of relevance to the query [362] or by considering relative [363] and absolute attributes [364, 365]. The limiting expressiveness of attributes was successively addressed in [366, 367] by considering purely textual feedback, allowing richer expressiveness. Nonetheless, performance of the textual model can limit the system in understanding and reacting appropriately. At the same time, GPT-2, BERT [368] and GPT-3 [369] models have shown that large amounts of text combined with recent improvements in attention mechanisms enable learning of powerful features that integrate vast knowledge. Adding images to the learning process, CLIP [360] has very recently shown that it is feasible to perform multimodal zero shot learning, obtaining remarkable feature generalization of both images and text. Contrary to standard vision models that are trained on typical datasets and that are good at only one task, this new class of models learn only associations between the abundant images and natural language supervision available on the internet. They are not directly optimized for a benchmark and yet are able to perform consistently well on different tasks. CLIP effectiveness is still subject of study [370], with first applications to art [371], image generation [372] and zero shot video retrieval [373]. Our work builds upon CLIP and further explores its potential in the task of conditioned image retrieval, applying the proposed approach to fashion.

In the growing area of image retrieval with user feedback, our work is related to the recently introduced conditioned fashion image retrieval with text [361]. In [374], a transformer that can be seamlessly plugged in a CNN to selectively preserve and transform the visual features conditioned on language semantics is presented. In [366, 375] they use skip connections and combine them with graph neural networks, reporting improved performance. In [376], image style and content are considered separately by two different neural network modules. In [377] a *Correction Network* is added which explicitly models the difference between the reference and target image in the embedding space.

Differently from these work, our method differs by few factors. It explicitly considers a learned manifold of visual and text features with the goal of learning an additive transformation in the same space. Moreover, our approach does not use any kind of spatial information. Instead, in [366, 376] features extracted from the backbone are 3-dimensional and the composition takes care of spatial information, in [374] the features are extracted at different convolutional layers from the ResNet-50 backbone. In [377] the authors divided the image and the sentence into a set of localized components assigning a representation module, denoted as *experts*, to each of them. More similar to our work is [375] which trains a combiner directly on flattened image and text features that, differently from our work, are obtained from different embeddings.

### 5.7.3. The proposed method

The proposed method addresses the multimodal problem of conditional fashion image retrieval. Given as input a reference image (e.g. an image of a black dress) and a text that includes a descriptive request from the user in relation to the image (“red and yellow”), the goal of the retrieval is to retrieving the best matching images that satisfy similarity constraints imposed by both of the input components (an image of a red and yellow dress). To retrieve correct images, the system must be able to understand both the contents of the image and text, and further add the textual comment to the image content.

In contrast to previous works like [374–377] that build from different image and textual model, we start from the hypothesis of having a common embedding of images and text, realized by CLIP. As shown in [360], similar concepts expressed in text and images tend to share similar features, or at least be “near” in the common space.

The input image and text are encoded using their respective CLIP encoders into features in the common space. The task is then cast as a problem of learning a transformation from the





reference image feature and input text to a combined feature that includes both the multimodal input information and is as near as possible to the common manifold. We denote this transformation as a *Combiner* function and design a neural network architecture that is trained to learn the correct function. We explore different Combiner functions showing that state of the art performance is obtainable.

The Combiner function, depicted in Fig. 39, is simple yet more performing than more complex architectures that we tested. The idea is to build an additive transformation where text, image and the combination of both are all added into the final combined feature. The text and image features are each weighted by a scalar that is trained to balance their contribution. We found these two contributions essential to obtain a new state of the art performance. The third contribution is given from the mixture of image and text. Starting from text and image features, we apply to each feature a linear transformation followed by the ReLU function. Features are then concatenated and the output is fed to another linear layer that is followed by a ReLU and a final linear layer. The three contributions are finally summed and  $L_2$  normalized. Dropout is applied to each layer to reduce overfitting.

Training of the system is performed with triplets of input images, text and target images. Following [366, 375] we employ the DML loss as pairwise contrastive loss using the normalized dot product as similarity kernel. Similarly to CLIP [360], we multiply the logits (i.e. the dot product between predicted and target features) by 100 before computing the loss. This was shown to help the training process by improving the dynamic range of features.

## Implementation Details

We decided to perform experiments using two CLIP models of different size. The smallest one is based on a modified ResNet-50 (RN50) [378] architecture. It takes as input images of  $224 \times 224$  pixels and outputs features of 1024 dimensions. The biggest one, denoted as RN50x4, follows the EfficientNet-style model scaling and use approximately 4x the computation of the smallest. It takes as input images of  $288 \times 288$  pixels and outputs features of 640 dimensions. In the experiments, the CLIP encoders have been kept frozen and the only trained part of the model is the Combiner function. The dropout rate was set to 0.5 as commonly done with linear layers. The text and image scalar weights were both initialized to 1. We used PyTorch in our experiments. The learning rate was set to  $5e - 5$  and we trained the model for a maximum of 300 epochs. The batch size was set to 1024 for the experiments with RN50 and 512 for the experiments with RN50x4, due to memory limits.

### 5.7.4. Experimental results

#### Dataset and metrics

We used the popular FashionIQ dataset [361] since it is commonly used to test conditioned image retrieval. FashionIQ provides 77,684 fashion images crawled from the web and split in train, validation and test sets, divided into three different categories: *Dress*, *Toptee* and *Shirt*. Among the 46,609 training images there are 18,000 training triplets made of a candidate image, a pair of user texts and a target image. The texts describe properties to modify in the candidate image to match the target image. Validation and test set have, respectively, 15,537 and 15,538 images with 6,017 and 6,119 triplets.

We follow experimental setting as in [376, 377]. We employ the average recall at rank K (Recall@K) as evaluation metric, namely Recall@10 (R@10) and Recall@50 (R@50). Note that for each triplet there is only a positive index image. Hence, each individual query has R@K either of



Model	Average	
	R@10	R@50
Sum	19.55	38.40
Weighted sum	19.78	39.04
No skip	23.38	46.81
Linear after skip	23.36	47.42
No Dropout	28.36	51.62
No ReLU & Dropout	28.20	51.10
CLIP fine-tuning	27.91	51.50
Proposed model	<b>29.67</b>	<b>53.41</b>

Table 22. Recall at K on the validation set, with variations on the architecture. Best score is highlighted in bold.

Batch size	Average	
	R@10	R@50
64	28.75	51.94
128	29.01	52.41
256	29.10	52.58
512	29.00	53.02
1024	<b>29.67</b>	<b>53.41</b>

Table 23. Recall at K on the validation set when increasing the batch size. Best score is highlighted in bold.

Method	Shirt		Dress		Toptee		Average	
	R@10	R@50	R@10	R@50	R@10	R@50	R@10	R@50
JVSM [379]	12.0	27.1	10.7	25.9	13.0	26.9	11.9	26.6
TRACE w/BERT [380]	20.80	40.80	22.70	44.91	24.22	49.80	22.57	46.19
VAL w/GloVe [374]	22.38	44.15	22.53	44.00	27.53	51.68	24.15	46.61
CurlingNet [381]	21.45	44.56	26.15	53.24	30.12	55.23	25.90	51.01
RTIC-GCN [375]	22.72	44.16	27.71	53.50	29.63	56.30	26.69	51.32
CoSMo [376]	24.90	49.18	25.64	50.30	29.21	57.46	26.58	52.31
DCNet [377]	23.95	47.30	<b>28.95</b>	<b>56.07</b>	30.44	58.29	27.78	53.89
Our (CLIP-RN50)	31.41	52.11	25.69	50.64	31.91	57.50	29.67	53.41
Our (CLIP-RN50x4)	<b>35.76</b>	<b>56.20</b>	27.20	53.57	<b>36.31</b>	<b>61.14</b>	<b>33.09</b>	<b>56.99</b>

Table 24. Comparison between our method and current state-of-the-art models on the Fashion-IQ validation set. Best scores are highlighted in bold.

zero or one. All results are on the validation set since at the time of writing the test set ground-truth labels has not been released yet.

### Ablation studies

In this section we show preliminary experiments with variations of the architecture shown in Fig. 39, and with different batch sizes. All experiments were performed with RN50 as backbone.





We tested the following baselines:

- **Sum**: image and text features are summed;
- **Weighted sum**: a weighted sum between the image and text features, i.e. the model without the mixture contribution of text and image;
- **No skip**: only the mixture contribution of text and image;
- **Linear after skip**: the regular model with an additional linear layer in both text and image contributions;
- **No Dropout**: without dropout layers;
- **No ReLU & Dropout**: without ReLU activations and dropout layers;
- **CLIP fine-tuning**: end-to-end fine-tuned CLIP with the Combiner function;
- **Proposed model**: the proposed model shown in Fig. 39.

We report the results for each variation in Tab. 22.

The first interesting thing to notice is that a simple sum of the candidate image features and the relative captions features led to decent results that are not too far from the worst competing state-of-the-art methods. This confirms that text and images in the CLIP embedding reside (approximately) in the same manifold. The weighted sum baseline, where text and image weights are learned, results in little improvement. The two weights stabilize respectively to 1 and 0.80 for images and text, signaling a preference towards image features.

Compared to the proposed model, we note that removing the text and image direct contributions lead to a significant drop in performance. Given the effectiveness of the Sum baseline, this is reasonable, since their presence may enable the Combiner function to only learn an offset to an already good starting point.

In our experiments, fine-tuning CLIP along Combiner training did not bring any performance improvement.

Regarding the batch size, we tested different value ranging from 64 to 1024. We report the performance obtained in Tab. 23. We note that increasing the batch size provides a  $\sim 3\%$  increase of both recall measures.

## Comparison with SotA

Tab. 24 shows the quantitative results on Fashion-IQ validation set. Our approach outperforms the state-of-the-art by improving up to  $\sim 5\%$  in average R@10 and 3% in average R@50 upon the best method, DCNet [377], when using the CLIP RN50x4 backbone. Our method have the highest recall in the Shirt and Toptee categories, with comparable performance in the Dress category, using both backbones. Between the two backbones, we note that bigger RN50x4 obtains the best performance, with an improvement on the smaller RN50 in the range of about 2% to 4% in all categories.

### 5.7.5. Conclusions

In this work we tackled the problem of conditioned image retrieval for fashion using the recent CLIP model where we exploited its zero shot transfer features. We developed a Combiner network that is able to compute a combined feature made from reference images integrated with a textual description. Experiments on the FashionIQ dataset show that our approach is able to outperform more complex state of the art methods.





Our future work will deal with the extension of the proposed method to videos and further experimentation with different image domains.

#### 5.7.6. Relevant publications

- Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. 2021. “Conditioned Image Retrieval for Fashion using Contrastive Learning and CLIP-based Features”. In Proc. of ACM Multimedia Asia (MMAAsia ’21). DOI:10.1145/3469877.3493593  
Zenodo record: <https://zenodo.org/record/6411201>.

#### 5.7.7. Relevance to AI4media use cases and media industry applications

As indicated in the introduction of this section, interactive (or conditioned) image retrieval is an interesting extension of the standard CBIR paradigm, and it has wide applicability in various media industry applications apart from CBIR itself, such as: near-duplicate detection (expressing the concept of “near” using natural language) and face or person recognition (expressing the differences w.r.t. an example). Our method for combined image retrieval contribute to use cases (a) 3A3 (in particular 3A3-11 Visual Indexing) and 4C1 by providing solutions to analyze visual content, and (b) 7A3 (Re)organisation of visual content, by supporting the organization of image and video collections.

## 5.8. Hyperbolic Vision Transformers

**Contributing partner:** UNITN

### 5.8.1. Introduction and methodology

Metric learning task formulation is general and intuitive: the obtained distances between data embeddings must represent semantic similarity. It is a typical cognitive task to generalize similarity for new objects given some examples of similar and dissimilar pairs. Metric learning algorithms are widely applied in various computer vision tasks: content-based image retrieval [382–384], near-duplicate detection [385], face recognition [386,387], person re-identification [388,389], as a part of zero-shot [384] or few-shot learning [390–392].

Modern image retrieval methods can be decomposed into roughly two components: the encoder mapping the image to its compact representation and the loss function governing the training process. Encoders with backbones based on transformer architecture have been recently proposed as a competitive alternative to previously used convolutional neural networks (CNNs). Transformers lack some of CNN’s inductive biases, *e.g.*, translation equivariance, requiring more training data to achieve a fair generalization. On the other hand, it allows transformers to produce more general features, which presumably can be more beneficial for image retrieval [393,394], as this task requires generalization to unseen classes of images. To alleviate the issue above, several training schemes have been proposed: using a large dataset [395], heavily augmenting training dataset and using distillation [396], using self-supervised learning scenario [394].

The choice of the embedding space directly influences the metrics used for comparing representations. Typically, embeddings are arranged on a hypersphere, *i.e.* the output of the encoder is  $L_2$  normalized, resulting in using cosine similarity as a distance. In this work, we propose to consider the hyperbolic spaces. Their distinctive property is the exponential volume growth with respect to the radius, unlike Euclidean spaces with polynomial growth. This feature makes hyperbolic space especially suitable for embedding tree-like data due to increased representation power. The paper [397] shows that a tree can be embedded to Poincaré disk with an arbitrarily low distortion.



Most of the natural data is intrinsically hierarchical, and hyperbolic spaces suit well for its representation. Another desirable property of hyperbolic spaces is the ability to use low-dimensional manifolds for embeddings without sacrificing the model accuracy and its representation power [398].

The goal of the loss function is straightforward: we want to group the representations of similar objects in the embedding space while pulling away representations of dissimilar objects. Most loss functions can be divided into two categories: proxy-based and pair-based [399]. Additionally to the network parameters, the first type of losses trains proxies, which represent subsets of the dataset [383]. This procedure can be seen from a perspective of a simple classification task: we train matching embeddings, which would classify each subset [400]. At the same time, pair-based losses operate directly on the embeddings. The advantage of pair-based losses is that they can account for the fine-grained interactions of individual samples. Such losses do not require data labels: it is sufficient to have pair-based relationships. This property is crucial for a widely used pairwise cross-entropy loss in self-supervised learning scenario [401–403]. Instead of labels, the supervision comes from a pretext task, which defines positive and negative pairs. Inspired by these works, we adopt pairwise cross-entropy loss for our experiments. The schematic overview of the proposed method is depicted in Figure 40.

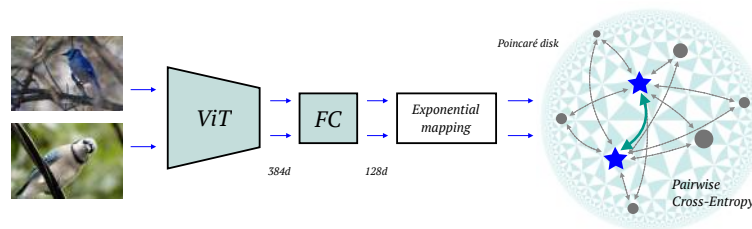


Figure 40. Overview of the proposed method. Two images representing one class (positives) are encoded with the vision transformer, projected into a space of a lower dimension with a fully connected (FC) layer, and then mapped to a hyperbolic space. Blue stars depict the resulting embeddings. Poincaré disk is shown with uniform triangle tiling on the background to illustrate the manifold curvature. Gray circles represent other samples from the batch (negatives). Finally, arrows in the disk represent distances used in the pairwise cross-entropy loss. Positives are pushed closer to each other, negative are pulled far apart.

## 5.8.2. Experimental results

We follow a widely adopted training and evaluation protocol [399] and compare several versions of our method with current state-of-the-art on benchmark datasets for category-level retrieval. There are two types of experiments, first, we compare with the state-of-the-art, and then we investigate the impact of hyperparameters (encoder patch size, manifold curvature, embedding size and batch size).

### 5.8.2.1. Datasets

**CUB-200-2011** (CUB) [404] includes 11,788 images with 200 categories of bird breeds. The training set corresponds to the first 100 classes with 5,864 images, and the remaining 100 classes with 5,924 images are used for testing. The images are very similar; some breeds can only be distinguished by minor details, making this dataset challenging and, at the same time, informative for the image retrieval task. **Cars-196** (Cars) [405] consists of 16,185 images representing 196 car models. First 98 classes (8,054 images) are used for training and the other 98 classes (8,131 images) are held out for testing.





Method	CUB-200-2011 (K)				Cars-196 (K)			
	1	2	4	8	1	2	4	8
Margin [406]	63.9	75.3	84.4	90.6	79.6	86.5	91.9	95.1
NSoftmax [407]	56.5	69.6	79.9	87.6	81.6	88.7	93.4	96.3
MIC [408]	66.1	76.8	85.6	-	82.6	89.1	93.2	-
IRT <sub>R</sub> [393]	72.6	81.9	88.7	92.8	-	-	-	-
Sph-DeiT	73.3	82.4	88.7	93.0	77.3	85.4	91.1	94.4
Sph-DINO	76.0	84.7	90.3	94.1	81.9	88.7	92.8	95.8
Sph-ViT §	83.2	89.7	93.6	95.8	78.5	86.0	90.9	94.3
Hyp-DeiT	74.7	84.5	90.1	94.1	82.1	89.1	93.4	96.3
Hyp-DINO	78.3	86.0	91.2	94.7	<b>86.0</b>	<b>91.9</b>	<b>95.2</b>	<b>97.2</b>
Hyp-ViT §	<b>84.0</b>	<b>90.2</b>	<b>94.2</b>	<b>96.4</b>	82.7	89.7	93.9	96.2

Table 25. Recall@K metric for 128-dimensional embeddings. The 6 versions of our method are listed in the bottom section, evaluated for head embeddings. “Sph-” are versions with hypersphere embeddings optimised using  $D_{cos}$ , “Hyp-” are versions with hyperbolic embeddings optimised using  $D_{hyp}$ . “DeiT”, “DINO” and “ViT” indicate type of pretraining for the vision transformer encoder. Margin, MIC, NSoftmax are based on ResNet-50 [409] encoder, IRT<sub>R</sub> is based on DeiT [396].

§ pretrained on the larger ImageNet-21k [254].

**5.8.2.2. Results** Table 25 highlights the experimental results for the 128-dimensional head embedding and the results for 384-dimensional encoder embedding are shown in Table 26. We include evaluation of the pretrained encoders without training on the target dataset in Table 26 for reference. On the CUB dataset, we can observe the solid performance of methods with ViT encoder; the gap between the second-best method IRT<sub>R</sub> and Hyp-ViT is 9%. However, the main improvement comes from the dataset used for pretraining (ImageNet-21k), since Hyp-DINO and Hyp-DeiT demonstrate a smaller improvement, while baseline ViT-S without finetuning shows strong performance. We hypothesize that this is due to the presence of several bird classes in the ImageNet-21k dataset encouraging the encoder to separate them during the pretraining phase.

For Cars-196, Hyp-DINO outperforms Hyp-ViT with a significant margin. These results confirm that both pretraining schemes are suitable for the considered task. The versions with DeiT perform worse compared to ViT- and DINO-based encoders while outperforming CNN-based models. This observation confirms the significance of vision transformers in our architecture. The experimental results suggest that hyperbolic space embeddings consistently improve the performance compared to spherical versions. Hyperbolic space seems to be beneficial for the embeddings, and the distance in hyperbolic space suits well for the pairwise cross-entropy loss function. At the same time, our sphere-based versions perform well compared to other methods with CNN encoders.

Figure 41 illustrates how the learned embeddings are arranged on the Poincaré disk. We use UMAP [422] method with the “hyperboloid” distance metric to reduce the dimensionality to 2D for visualization. For the training part, we can see that samples are clustered according to labels, and each cluster is pushed closer to the border of the disk, indicating that the encoder separates classes well. However, for the testing part, the structure is more complex. We observe that some of the samples tend to move towards the center and intermix, while others stay in clusters, showing possible hierarchical relationships. We can see that car images are grouped by several properties: pose, color, shape, etc.





Method	Dim	CUB-200-2011 (K)				Cars-196 (K)			
		1	2	4	8	1	2	4	8
A-BIER [410]	512	57.5	68.7	78.3	86.2	82.0	89.0	93.2	96.1
ABE [411]	512	60.6	71.5	79.8	87.4	85.2	90.5	94.0	96.1
SM [412]	512	56.0	68.3	78.2	86.3	83.4	89.9	93.9	96.5
XBM [413]	512	65.8	75.9	84.0	89.9	82.0	88.7	93.1	96.1
HTL [414]	512	57.1	68.8	78.7	86.5	81.4	88.0	92.7	95.7
MS [415]	512	65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5
SoftTriple [416]	512	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9
HORDE [417]	512	66.8	77.4	85.1	91.0	86.2	91.9	95.1	97.2
Proxy-Anchor [399]	512	68.4	79.2	86.8	91.6	86.1	91.7	95.0	97.3
NSoftmax [407]	512	61.3	73.9	83.5	90.0	84.2	90.4	94.4	96.9
ProxyNCA++ [418]	512	69.0	79.8	87.3	92.7	86.5	92.5	95.7	97.7
IRT <sub>R</sub> [393]	384	76.6	85.0	91.1	94.3	-	-	-	-
ResNet-50 [409] †	2048	41.2	53.8	66.3	77.5	41.4	53.6	66.1	76.6
DeiT-S [396] †	384	70.6	81.3	88.7	93.5	52.8	65.1	76.2	85.3
DINO [394] †	384	70.8	81.1	88.8	93.5	42.9	53.9	64.2	74.4
ViT-S [419] † §	384	83.1	90.4	94.4	96.5	47.8	60.2	72.2	82.6
Sph-DeiT	384	76.2	84.5	90.2	94.3	81.7	88.6	93.4	96.2
Sph-DINO	384	78.7	86.7	91.4	94.9	86.6	91.8	95.2	97.4
Sph-ViT §	384	85.1	90.7	94.3	96.4	81.7	89.0	93.0	95.8
Hyp-DeiT	384	77.8	86.6	91.9	95.1	86.4	92.2	95.5	97.5
Hyp-DINO	384	80.9	87.6	92.4	95.6	<b>89.2</b>	<b>94.1</b>	<b>96.7</b>	<b>98.1</b>
Hyp-ViT §	384	<b>85.6</b>	<b>91.4</b>	<b>94.8</b>	<b>96.7</b>	86.5	92.1	95.3	97.3

Table 26. Recall@K metric, “Dim” column shows the dimensionality of embeddings. The 6 versions of our method are listed in the bottom section, evaluated for encoder embeddings, titles are described in Table 25. Encoders by method: A-BIER, ABE, SM: GoogleNet [420]; XBM, HTL, MS, SoftTriple, HORDE, Proxy-Anchor: Inception with batch normalization [421]; NSoftmax, ProxyNCA++: ResNet-50 [409]; IRT<sub>R</sub>: DeiT [396]. † pretrained encoders without training on the target dataset. § pretrained on the larger ImageNet-21k [254].

### 5.8.3. Conclusions

The main contributions of our research are the following:

- We propose to project embeddings to the Poincaré ball and to use the pairwise cross-entropy loss with hyperbolic distances. Through extensive experiments, we demonstrate that the hyperbolic counterpart outperforms the Euclidean setting.
- We show that the joint usage of vision transformers, hyperbolic embeddings, and pairwise cross-entropy loss provides the best performance for the image retrieval task.

### 5.8.4. Relevant publications

- A. Ermolov, L. Mirvakhabova, V. Khrukov, N. Sebe, and I. Oseledets, “Hyperbolic Vision Transformers: Combining Improvements in Metric Learning”, IEEE/CVF Conference on



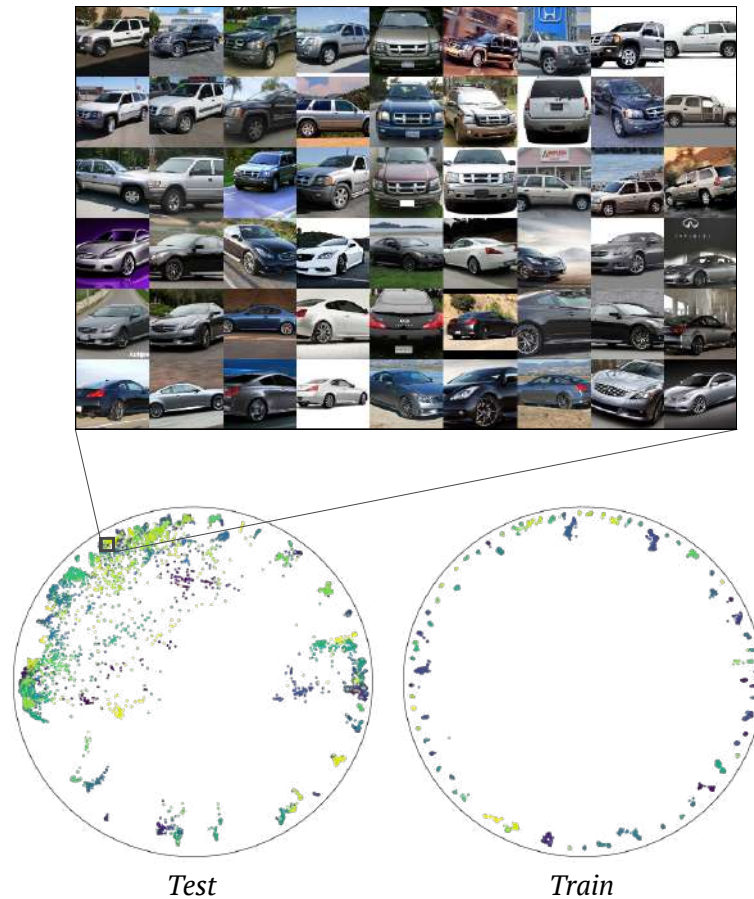


Figure 41. Hyp-DINO embeddings for Cars-196 dataset (training and evaluation sets) on the Poincaré disk. Each point inside the disk corresponds to a sample, different colors indicate different classes. Images of cars are plotted preserving neighborhood relations of samples.

Computer Vision and Pattern Recognition (CVPR'22), June 2022 [4].  
Zenodo record: <https://zenodo.org/record/7100206>.

#### 5.8.5. Relevant software/datasets/other outcomes

- The PyTorch implementation can be found in [https://github.com/htdt/hyp\\_metric](https://github.com/htdt/hyp_metric).

#### 5.8.6. Relevance to AI4media use cases and media industry applications

As indicated in the beginning of the section, the metric learning task is generic and tries to enforce the distances between data embeddings to represent semantic similarity. As such it has wide applicability in various media industry applications: content-based image retrieval, near-duplicate detection, face recognition, person re-identification, zero-shot and few-shot learning. Specifically the presented results could be directly relevant to use cases (a) 3A3 (archive exploration), specifically 3A3-11 Visual indexing and search and (b) 7A3 (Re)organisation of visual content by supporting the efficient training and organization of image and video collections.





## 6. Transfer learning (Task 3.3) – detailed description

**Contributing partners:** BSC, UNITN, CEA, CNR

Transfer Learning is an emerging field among Deep Learning practitioners that seeks to reuse and exploit previously generated models for different purposes. Considering the huge amount of data, human effort and computational power needed to train these models, being able to reuse them is of paramount importance. Beyond practical reasons, Transfer Learning poses a scientific challenge of relevance, as it forces researchers to question the internal knowledge representation of deep models. Indeed, to understand how to reuse deep representations, one must first understand how are these representations learned, and how are they internally structured. Advances in this field have potential relevance for key aspects of Deep Learning, such as explainability and interpretability, efficiency and footprint reduction, and real world deployment of AI powered systems.

### 6.1. When & How: Methodological study on transfer learning

**Contributing partners:** BSC

Transfer learning is a method that reuses the learnt knowledge in a neural network (source or pre-trained model) for another (target) model and dataset. It assumes that the learnt features in such models are representations general enough as to be of use in other tasks, and alleviates requirements of storage space, energy, labelled data and powerful computational devices. In the context of deep learning, the properties of such transfer have been widely studied [35, 186, 423, 424], including lower training times, increased performance and the requirement of less data for the same or better results than training a network from scratch without knowledge reuse.

In image classification tasks, the most common approaches of transfer learning are FE and Fine-Tuning (FT). Fine-tuning a pre-trained network for a target task consists on simply re-training the parameters of the pre-trained model for the new task – in contrast to training a network from scratch. Feature extraction approaches freeze all of the pre-trained model’s parameters and train a simpler classifier using the embeddings from the target task’s instances [35, 424]. FT is normally the default method when performing transfer learning [186], improving performance when the pre-trained model was trained in a large enough dataset, regardless of the size of the target dataset [425]. However, in cases where the target data is scarce, fine-tuning all parameters of a network may cause overfitting [186], which can be alleviated by freezing some of the layers. Alternatively, FE methods provide a different representation that reduces the dimensionality – helping deal with the curse of dimensionality – and noise in the input [426].

Both FT and FE have different pros and cons, and no clear guidelines exists to support a choice between both. That is why, to fill the gap, we study both methods while taking into account different perspectives, including the availability of pre-trained models on similar tasks, data volume, and the trade-offs between performance, carbon footprint, human cost and computational requirements.

#### 6.1.1. Methodology

**6.1.1.1. Models and datasets** We use two VGG16 [427], one trained on *ImageNet 2012* [428] (*IN*, 1.2M train images) with Top-1/Top-5 accuracy of 71.3/90.1, the other trained on *Places 2* [429] (*P2*, 1.8M train images) with 55.2/85.0 accuracy. These are publicly available at <https://keras.io/api/applications> and <https://github.com/CSAILVision/places365>, respectively.

We use 10 target datasets for evaluation, which represent a variety of possible scenarios when performing transfer learning with respect to transferability. Some of them are direct subsets ( $\subset$ ) of a





source dataset (*Stanford Dogs* [430], *Caltech 101* [431]  $\subset IN$ ; *MIT ISR* [432]  $\subset P2$ ); some intersect ( $\cap$ ) with a pre-training dataset but expand beyond it (*Food 101* [433], *CUB200* [404], *Oxford Flower* [434], *Oxford-IIIT-Pet* [435]  $\cap IN$ ); and some are entirely disjoint ( $\emptyset$ ) to the pre-training datasets (*DTD* [436], *Oulu Knots* [437], *MAMe* [438]). As *Oulu Knots* suffers from a very significant imbalance, we take the smallest class out in all of our experimentation, calling this subset *Knots6*.

**6.1.1.2. Methods** For FT, we re-train the model while freezing a variable number of initial layers, and re-initializing the last two layers. A minimum of 10 and a maximum of 25 epochs are computed. Training is also stopped when three consecutive epochs show a non-improving validation loss. Batch size is 64. For FE we use the Full Network Embedding, which extracts an embedding from a percentage of layers starting from the last before the classification layer, using an average pooling operation on convolutional layers to extract one number per channel. Activations are feature-wise standardised and discretised to values in  $\{-1, 0, 1\}$  [439]. Experiments are given a time limit of 24 hours. Data augmentation is used in both FT and FE, using 10 crops per sample (4 corners and central, with horizontal mirroring). During inference, crop predictions are aggregated using majority voting.

**6.1.1.3. Metrics** We use metrics from four categories. (1) *Performance*: Validation and test mean class accuracies ( $V_{ACC}$ ,  $T_{ACC}$ ), and overfitting ( $V_{ACC} - T_{ACC}$ ). (2) *Footprint*: power average in kW ( $P_{AVG}$ ) in a sample of trainings, estimated greenhouse gas emissions in Kg of CO<sub>2</sub> ( $E_{CO_2}$ ). (3) *Computational requirements*: execution time in hours ( $T$ ), amount of experiments ( $n_{EXP}$ ). (4) *Human cost*: Time analysing results and designing experimentation ( $A$ ).

**6.1.1.4. Experiments** To both gauge the costs and benefits of FE and FT, and at the same time to select the best hyperparameter configurations for further experimentation, we perform a model selection process for each pair of approach and target dataset. In these process, we track all aforementioned metrics. We also study the effect of varying the amount of samples per class in the training dataset for FE and FT, as well as try to gauge trade-offs between them, with respect to performance and computation time.

## 6.1.2. Results

**6.1.2.1. Hyperparameter search** We perform model selection processes for the 20 possible pre-trained model and task pairs, and consider a few hyperparametric variables, chosen by their impact on performance. In 8 out of 10 datasets, FT obtained the better performing model (mean of  $2.84 \pm 8.66\%$ ). In overfitting ( $V_{ACC} - T_{ACC}$ ), FT shows a slightly higher drop in test performance (mean drop of 3.6%, for FE's 1.92%). The average power consumption in FT was 276.1W, 222% bigger than that of FE (124.1W). FT emitted 52.5 times more CO<sub>2</sub> than FE. In the context of the performance-footprint trade-off, notice that this increase in footprint produces a mean improvement in  $V_{ACC}$  of 2.81%. For  $T_{ACC}$ , only 1.13%. The FT search lasted a total 1,825.72 hours, 30.4 times more than the FE search. In 9 out of 10 tasks, the FE search was faster than FT. Each FT search also required 6 times more experiments, influenced by a larger hyperparameter selection. Lastly, the time dedicated by experts on analysing results and designing experimentation ( $A$ ), show how FE results are significantly easier to process thanks to its plain performance metrics. However, when performing FT, it is relevant to look at the tables beyond simple metrics, as part of the information regarding model convergence is lost when looking at a single performance metric. Thus, processing the results from FT entails the analysis of multiple training curves, which require in the order of 4 to 6 times more time. That being said, current deep learning frameworks are more oriented towards FT, which make FE slightly harder to implement.



	$V_{ACC}$	$T_{ACC}$	$P_{AVG}$	$E_{CO_2}$	$T$	$n_{EXP}$	$A$
FT	77.46	73.86	276.1W	201.54kg	1,825.72h	480	4-6h
FE	74.65	72.73	124.1W	3.84kg	60.02h	80	0-1h

Table 27. Performance ( $V_{ACC}$  and  $T_{ACC}$ , average), footprint ( $P_{AVG}$  average,  $E_{CO_2}$  sum), computational requirements ( $T$  sum,  $n_{EXP}$  total) and human cost ( $A$  sum) of the hyperparameter searches.

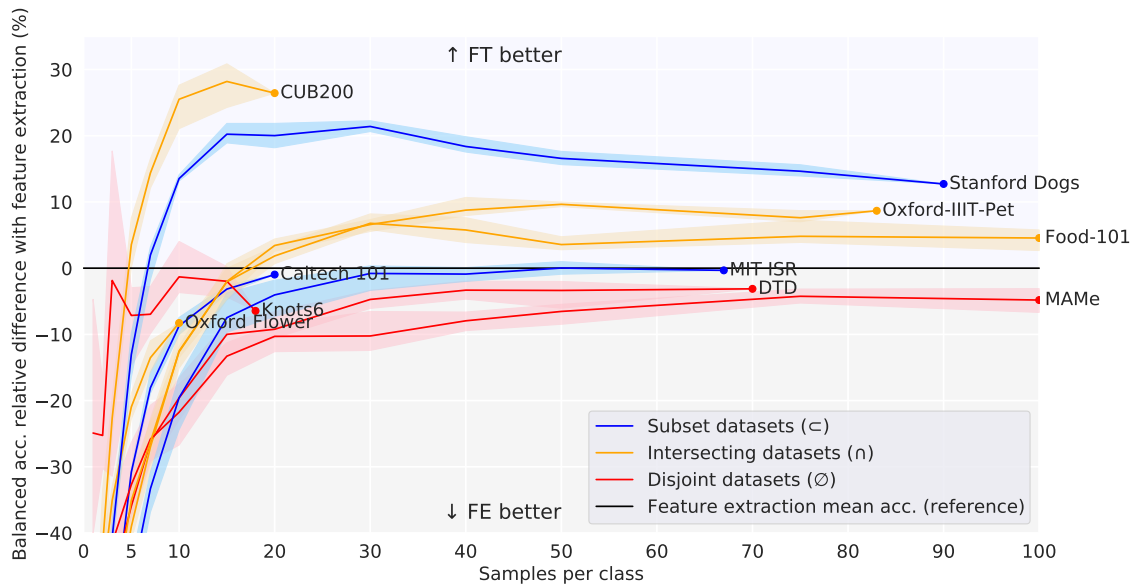


Figure 42. Relative difference in test accuracy ( $100 \cdot \frac{FT - FE}{FE}$ ), w.r.t. the train split size. Shaded regions show the minimum and maximum differences among the 5 random subsets. Black line represents point in which FE and FT have same performance.

**6.1.2.2. Few-shot learning** We study the effects of limited data availability, by subsetting the target datasets prior to training. For each target dataset and number of instances per class ( $IC$ ), we generate 5 random subsets (to mitigate statistical variance). We train FT and FE models and extract their final  $T_{ACC}$ . The distribution and mean relative difference in  $T_{ACC}$  of the five runs for each  $IC$  are shown in Figure 42. These results allow us to categorise datasets in two types: those where FT overtakes FE given enough samples per class, and those where FT and FE converge to the same accuracy. This distinction can be made at around 25 instances per class: if FT has not overcome FE with that amount or less, it will not happen regardless of data availability. Notice such threshold may depend on the architecture or source dataset employed [425, 440].

Figure 42 illustrates how, for those datasets that are disjoint from the pretraining dataset, FT and FE performance tends to converge at larger  $IC$ . On the other hand, for those datasets which overlap (either intersect or subset) with the pretraining dataset, FT outperforms FE. We are unsure about the cause of this difference in behaviour. Another factor to consider here is time until convergence. While both methods scale linearly with  $IC$ , the cost of FT grows 7 times faster than the cost of FE.



### 6.1.3. Discussion

While FT is generally superior in *performance*, FE is better in terms of *footprint*, *computational requirements* and *human cost*. This is important in domains where TL is unfavourable (*i.e.* one cannot find pre-trained models close to the target dataset). Domains in which data availability is highly variable (*e.g.* because of new data constantly being added) also favour FE, as FT requires for constant and expensive hyperparameter searches while FE does not. The gain in performance provided by FT over FE is, on average, 2.8% on  $V_{ACC}$ , and 1.1% on  $T_{ACC}$ . Meanwhile, FT produces in the order of 7,000% more CO<sub>2</sub> than FE, and demands between 4 and 6 times more of human effort. In this context, researchers and practitioners of TL deciding between FT and FE should consider the relevance of limited performance gains for each application. While additional performance gains for FT over FE could be obtained by intensive data gathering, labelling and pretraining efforts, these would correspondingly increase the cost of later performing FT, allowing this point to hold.

In few-shot learning scenarios (around 5 or less samples per class), the previous performance gain of FT vanishes, making FE the best performing model in most settings. At more than 5 samples per class, FT may overtake FE by considerable margins in intersecting and subset ( $\subset$ ,  $\cap$ ) datasets. For disjoint datasets, FT may require more than 100 samples per class to outperform FE, with the latter still having competitive performance. We believe these results will hold for architectures and pre-training datasets comparable to the ones used here (*i.e.* CNNs and datasets in the order of a few millions of training samples).

### 6.1.4. Relevant publications

- A. Tormos, D. Garcia-Gasulla, V. Gimenez-Abalos, and S. Alvarez-Napagao, “When & How to transfer with Transfer Learning,” in *Has it Trained Yet? NeurIPS 2022 Workshop*, 2022. [441]. Zenodo record: <https://zenodo.org/record/8014324>.

### 6.1.5. Relevant software/datasets/other outcomes

- The implementation can be found in <https://github.com/HPAI-BSC/tl-tradeoff>.

## 6.2. Source-Free Open Compound Domain Adaptation in Semantic Segmentation

**Contributing partner:** UNITN

### 6.2.1. Introduction and methodology

Deep learning has now achieved a remarkable success in fully-supervised semantic segmentation [442–444], which, however, is relied heavily on the expensive dense pixel-wise annotations. One solution to lighten the labeling cost is Unsupervised Domain Adaptation (UDA), which aims to transfer the knowledge of labeled synthetic data to unlabeled real-world data. Despite the effectiveness of existing UDA methods [121, 445, 446], they mainly consider the context of a single target domain, resulting in limited applications in the real world. Indeed, the target domain may be captured from multiple data distributions without a clear separation and the system will unavoidably face instances from unseen domains. To investigate a more realistic Domain Adaptation (DA) problem, in this paper, we consider the setting of open compound domain adaptation (OCDA) [447] for semantic segmentation. In OCDA, the unlabeled target domain is a compound of multiple homogeneous



Table 28. Comparisons of different cross-domain transfer learning settings. *DA*: domain adaptation, *SF-DA*: source-free DA, *DG*: domain generalization, *OCDA*: open compound DA, *SF-OCDA*: source-free OCDA.

Settings	Source Data	Source Model	Unlabeled Target	Multiple Targets	Open Targets
DA [445]	✓	✓	✓	✗	✗
SF-DA [448]	✗	✓	✓	✗	✗
DG [449]	✗	✓	✗	✗	✓
OCDA [447]	✓	✓	✓	✓	✓
SF-OCDA	✗	✓	✓	✓	✓

domains without domain labels. The adapted model is applied to test samples from the compound target domain and an open domain, where the open domain is unseen during training.

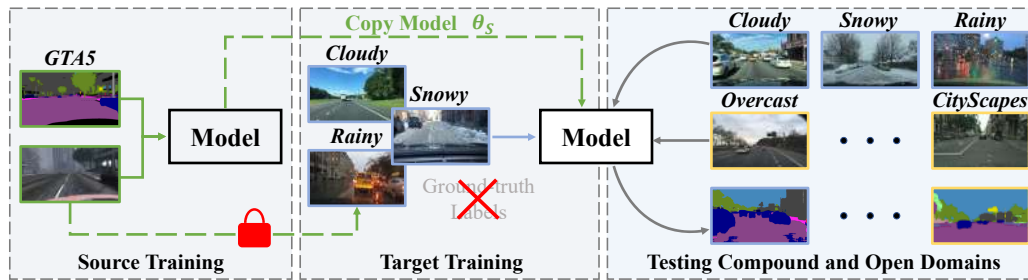


Figure 43. Illustration of SF-OCDA. In the training stage, the model is first trained on the (synthetic) labeled *source* data and then adapted to the (real-world) unlabeled *compound* target data. The *source* data are *not* available during the target adaptation. In the testing stage, the learned model is used to predict the semantic segmentation results for samples from the *compound* and *open* domains.

Existing UDA [445, 446, 450] and OCDA [447, 451, 452] methods commonly require the use of the labeled source data during the whole training process. However, the source data are not always available due to data privacy. In addition, the source data are generally very large, which require plenty of storage space (*e.g.*, GTA5 [453]  $\approx 57$ GB). This further limits the applications of existing methods, especially when transferring to a lightweight self-driving device. Nevertheless, we can choose to maintain the pre-trained source model instead of the source data, enabling us to obey the data privacy policy and use much less storage space (*e.g.*, DeepLab-VGG16 [427, 444]  $\approx 120$ MB). These facts motivate us to introduce a more challenging but practical setting for OCDA, called source-free OCDA (**SF-OCDA**), where only the source pre-trained model and the unlabeled target data are available during the training of the target model. In the literature, SFDA has recently been developed in image classification [120, 454] and semantic segmentation [448, 455] for the single target case. However, as shown in Table 28 and Figure 43, compared with SF-DA, our SF-OCDA demands not only adapting to data from multiple target domains but also considering the generalization performance on unseen domains.

In SF-OCDA, the source data and target data are invisible to each other. In such context, we cannot align the domain distributions as traditional DA methods [121, 445, 446]. Instead, this research introduces an effective two-stage framework for SF-OCDA (see Figure 44), which consists of (1) training a generalized source model and (2) adapting the target model with self-supervised learning. In the first stage, we aim to learn a robust model, which can generalize well to different target domains. To achieve this goal, we propose the **Cross-Patch Style Swap (CPSS)**, which can



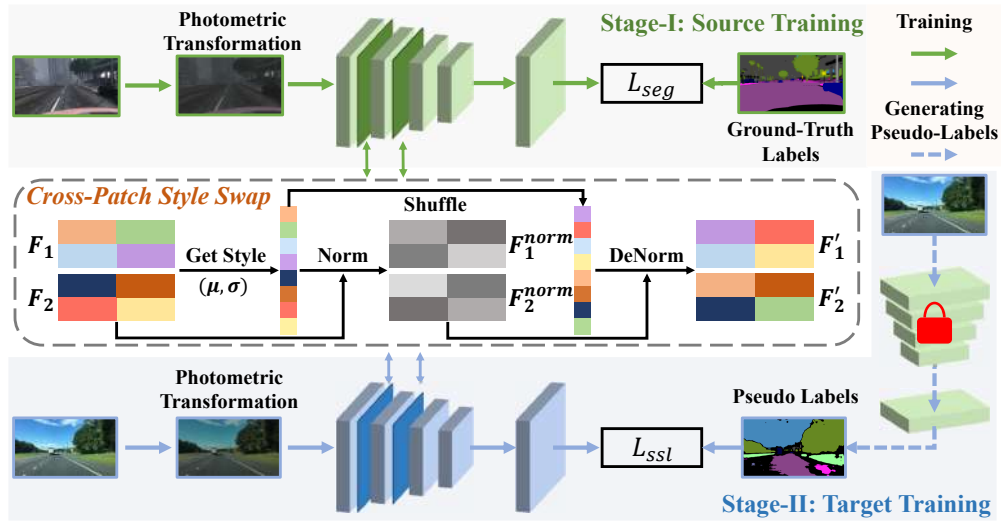


Figure 44. The framework of the proposed method. (1) The model is first trained on the labeled source domain. (2) We generate pseudo-labels by the source pre-trained model and train the target model in a self-training manner. In the second stage, we have no access to the source data. To improve the generalization ability of the model, we equip the model with the Cross-Patch Style Swap module in the two training stages, which augments features by exchanging styles among patches.

effectively augment the samples with various image styles. Specifically, CPSS first extracts the styles of patches in feature maps and then randomly exchanges the styles among patches by the instance normalization and de-normalization. In this manner, CPSS can prevent the model from overfitting to the source domain and thus significantly improve the generalization ability of the model. In the second stage, we adapt the target model by self-supervised learning. Specifically, we optimize the target model with the guide of pseudo-labels generated from the pre-trained source model, which can implicitly align the source and target distributions under the constraint of label consistency. Moreover, CPSS is also applied to reduce the influence of noisy pseudo-labels and to avoid overfitting to the target domain, which can further boost the performance on the compound and open domains.

## 6.2.2. Experimental results

**Datasets.** Following [447], we use the synthetic image data GTA5 [453] as the source domain, the rainy, snowy, and cloudy images in C-Driving [447, 459] as the compound target domain, and the overcast images in C-Driving as the open domain. To further measure the generalization ability of models, we additionally use Cityscapes [460] as an extended open domain. GTA5 includes 24,966 training images with a resolution of  $1914 \times 1052$ . C-Driving consists of 14,697 unlabeled training images and 1,430 testing images, where the image size is  $1280 \times 720$ . Cityscapes contains 500 images of  $2048 \times 1024$  for validation. For all datasets, pixels belong to 19 shared semantic categories. During testing, we use mean intersection-over-union (mIoU) to evaluate the semantic segmentation performance.

### 6.2.2.1. Comparison with State-of-the-Art Methods

**Results of GTA5  $\rightarrow$  C-Driving.** In Table 29, we compare our method with the state-of-the-art UDA models [445, 456–458] and OCDA models [447, 451, 452] on the setting of “GTA5 to C-Driving”.



Table 29. Comparison with the state-of-the-art methods on GTA5  $\rightarrow$  C-Driving. † denotes methods that employ the long-training strategy.

Methods GTA5 $\rightarrow$	Backbone	Source Free	Compound(C)			Open(O) Overcast	Avg	
			Rainy	Snowy	Cloudy		C	C+O
Source Only	VGG16	✓	16.2	18.0	20.9	21.2	18.9	19.1
AdaptSeg [445]		✗	20.2	21.2	23.8	25.1	22.1	22.5
CBST [456]		✗	21.3	20.6	23.9	24.7	22.2	22.6
IBN-Net [457]		✗	20.6	21.9	26.1	25.5	22.8	23.5
PyCDA [458]		✗	21.7	22.3	25.9	25.4	23.3	23.8
Liu et al. [447]		✗	22.0	22.9	27.0	27.9	24.5	25.0
Park et al. [451]		✗	27.0	26.3	30.7	32.8	28.5	29.2
Source Only†	VGG16	✓	23.6	24.4	27.8	29.5	25.6	26.3
AdaptSeg [445]†		✗	25.6	27.2	31.8	32.1	28.8	29.2
MOCDA [452]†		✗	24.4	27.5	30.1	31.4	27.7	29.4
Park et al. [451]†		✗	27.1	30.4	35.5	36.1	32.0	32.3
Ours (Stage-I)†		✓	28.5	30.5	36.4	37.4	32.8	33.2
Ours (Stage-II)†		✓	<b>30.6</b>	<b>31.9</b>	<b>37.6</b>	<b>38.0</b>	<b>34.4</b>	<b>34.5</b>
Source Only†	ResNet101	✓	27.6	27.8	32.9	33.0	30.0	30.3
Ours (Stage-I)†		✓	<b>35.5</b>	33.4	41.4	41.2	37.8	37.9
Ours (Stage-II)†		✓	35.3	<b>36.9</b>	<b>41.8</b>	<b>42.0</b>	<b>38.5</b>	<b>39.0</b>

For a fair comparison, all the models adopt DeepLab-V2 with VGG16 backbone. Following [451], we use the long training scheme (150K iterations) to train the model. We make the following observations. First, the models trained with the long training scheme produce higher results, showing the advantage of the long training scheme. Second, our Stage-I model, which is trained only with the source data, achieves the best performance among all the existing methods that use both the source and the target data. This verifies the effectiveness of the proposed CSPP in learning a generalizable model. Third, our Stage-II model outperforms all compared models by a large margin, indicating that our method produces new state-of-the-art performance for OCDA, even under the source-free constraint. In addition, we also provide the results of our method with ResNet101 backbone. As shown in Table 29, our Stage-I model outperforms the baseline model by 7.8% in C mIoU and 7.6% in C+O mIoU, and our target training stage yields 1.1% improvement in C+O mIoU.

**Results of Domain Generalization.** We also verify the generalization ability of our method on CityScapes in Table 30. We can observe that our Stage-I model surpasses the state-of-the-art domain generalization methods with both VGG16 and ResNet101 backbone when trained only with GTA5. Compared with DRPC [449] that additionally uses ImageNet [254] images, our model outperforms it by 1.0% and 2.2% in mIoU with VGG16 and ResNet101 backbone respectively. These findings demonstrate the effectiveness of the proposed method on open domains.

**6.2.2.2. Evaluation** In this section, we evaluate the effectiveness and superiority of the proposed method. Experiments are conducted with VGG16 backbone.

**Effectiveness of Style Augmentations.** In Table 31, we investigate the effectiveness of the proposed CPSS and Photometric Transformation (PT). Clearly, CPSS consistently improves the performance for both stages. Specifically, for the source training stage (Stage-I), inserting CPSS outperforms the baseline by 5.6% in C mIoU and by 5.7% in C+O mIoU. Adopting the photometric transformation further gains 1.6% and 1.2% improvement in C mIoU and C+O mIoU,





Table 30. Evaluation on open domain CityScapes. § extra using the ImageNet images.

Method	GTA5 $\rightarrow$ CityScapes	
	VGG16	ResNet101
ASG [461]	31.5	32.8
IBN-Net [457]	34.8	40.3
DRPC [449]§	36.1	42.5
Ours (Stage-I)	<b>37.1</b>	<b>44.7</b>

respectively. For the target training stage (Stage-II), we initialize the model by the source model trained with CPSS and PT. Without using style augmentations, self-supervised learning achieves limited improvement. In contrast, adding CPSS can clearly promote performance on both compound and open domains. This verifies that CPSS can not only reduce the impact of noisy samples but also improve the robustness of the model to unseen domains. On the other hand, using photometric transformation has a slight influence on the performance. This is mainly because the model has been familiar with such transformation during source training.

Table 31. Effectiveness of style augmentations.

Model	CPSS	PT	C	C+O
Stage-I	✗	✗	25.6	26.3
	✓	✗	31.2	32.0
	✓	✓	<b>32.8</b>	<b>33.2</b>
Stage-II	✗	✗	33.3	33.5
	✓	✗	34.3	34.4
	✓	✓	<b>34.4</b>	<b>34.5</b>

Table 32. Comparison of different stylized operations.

Method	C	C+O
MixStyle [462]	30.7	31.2
CrossNorm [463]	31.4	31.8
CPSS (intra-image)	31.7	32.3
CPSS (inter-image )	<b>32.8</b>	<b>33.2</b>

**Comparison of Different Stylized Operations.** In Table 32, we compare several stylized operations that do not use any auxiliary information, *i.e.*, MixStyle [462], CrossNorm [463], and two versions of our CPSS. Experiments are conducted in the source training stage. We can find that mixing styles with a random weight (MixStyle) is less suitable for semantic segmentation, because MixStyle may sometimes generate semantically unrealistic styles. Compared with CrossNorm and CPSS (intra-image), CPSS (inter-image) produces clearly higher performance. This indicates that augmenting samples with more various styles can help us to learn a more generalizable model.

**Is Splitting Latent Domains Necessary?** Recent OCDA methods [451, 452] show that the sub-domain labels can be used to reduce the latent domain gaps in the target domain. Instead, in our target training stage, we randomly select training samples from the target data to form the mini-batch without considering the sub-domain labels. To verify the impact of considering the latent domains for CPSS, we implement our framework with a new sampling strategy. Specifically,





we sample the images in a balanced way, so that each mini-batch contains at least one sample for each sub-domain. We provide two kinds of latent domains: “Oracle” denotes using the original rainy, snowy, cloudy as the latent domains; and “Clustering” denotes separating latent domains by clustering the style features. As shown in Table 33, the random sampling strategy and its two variants achieve similar performance. This indicates that the proposed CPSS can potentially consider the style variations among multiple latent domains and learn a robust model.

Table 33. Impact of latent domains.

W/ Latent	Split	C	C+O
✓	Clustering	<b>34.4</b>	<b>34.7</b>
	Oracle	34.3	34.5
✗	—	<b>34.4</b>	34.5

### 6.2.3. Conclusions

Our contributions are summarized as follows:

- We introduce a new setting for semantic segmentation, *i.e.*, SF-OCDA, which is an important yet unstudied problem. In addition, we propose an effective framework for solving SF-OCDA, which focuses on learning a generalized model during the stages of source pre-training and target adaptation.
- We propose the CPSS, which diversifies the samples in the feature-level, to improve the generalization ability of the model in both source and target training stages. CPSS is a lightweight module without learnable parameters, which can be readily injected into existing segmentation models.
- The proposed framework learned with the source-free constraint significantly outperforms the state-of-the-art methods on the OCDA benchmark. Our approach also surpasses the advanced domain generalization approaches on CityScapes.

### 6.2.4. Relevant publications

- Y. Zhao, Z. Zhong, Z. Luo, G. H. Lee, and N. Sebe, Source-Free Open Compound Domain Adaptation in Semantic Segmentation, *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10):7019-7032, October 2022. [464].  
Zenodo record: <https://zenodo.org/record/7565978>.

### 6.2.5. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in <https://github.com/HeliosZhao/SFOCDA>.

### 6.2.6. Relevance to AI4media use cases and media industry applications

Semantic segmentation is an useful tool for providing the first step towards image understanding however, the existing approaches rely heavily on the expensive dense pixel-wise annotations. Our solution addresses the challenging setting of open compound domain adaptation where the unlabeled





target domain is a compound of multiple homogeneous domains without domain labels. The adapted model is applied to test samples from the compound target domain and an open domain, where the open domain is unseen during training. This approach provides a wide applicability to several use cases: (a) UC2B by providing solutions to analyze/adapt the visual content, and (b) UC2A and 2B by providing solutions to discover new visual content and adapt accordingly. These can help to improve tagging and search capabilities.

### 6.3. solo-learn: A Library of Self-supervised Methods for Visual Representation Learning

**Contributing partner:** UNITN

We introduce here **solo-learn**, a library of self-supervised methods for visual representation learning. Implemented in Python, using Pytorch and Pytorch lightning, the library fits both research and industry needs by featuring distributed training pipelines with mixed-precision, faster data loading via Nvidia DALI, online linear evaluation for better prototyping, and many additional training tricks. Our goal is to provide an easy-to-use library comprising a large amount of SSL methods, that can be easily extended and fine-tuned by the community. **solo-learn** opens up avenues for exploiting large-budget SSL solutions on inexpensive smaller infrastructures and seeks to democratize SSL by making it accessible to all.

Deep networks trained with large annotated datasets have shown stunning capabilities in the context of computer vision. However, the need for human supervision is a strong limiting factor. Unsupervised learning aims to mitigate this issue by training models from unlabeled datasets. The most prominent paradigm for unsupervised visual representation learning is SSL, where the intrinsic structure of the data provides supervision for the model. Recently, the scientific community devised increasingly effective SSL methods that match or surpass the performance of supervised methods. Nonetheless, implementing and reproducing such works turns out to be complicated. Official repositories of state-of-the-art SSL methods have very heterogeneous implementations or no implementation at all. Although a few SSL libraries [465,466] are available, they assume that larger-scale infrastructures are available or they lack some recent methods. When approaching SSL, it is hard to find a platform for experiments that allows running all current state of the art methods with low engineering effort and at the same time is effective and straightforward to train. This is especially problematic because, while the SSL methods seem simple on paper, replication of published results can involve a huge time and effort from researchers. Sometimes official implementations of SSL methods are available, however, releasing standalone packages (often incompatible with each other) is not sufficient for the fast-paced progress in research and emerging real-world applications. There is no toolbox offering a genuine off-the-shelf catalog of state-of-the-art SSL techniques that is computationally efficient, which is essential for in-the-wild experimentation.

To address these problems, we developed **solo-learn**, an open-source framework that provides standardized implementations for a large number of state-of-the-art SSL methods. We believe **solo-learn** will enable a trustworthy and reproducible comparison between the state of the art methods. The code that powers the library is written in Python, using Pytorch [467] and Pytorch Lightning(PL) [468] as back-ends and Nvidia DALI<sup>3</sup> for fast data loading, and supports more modern methods than related libraries. The library is highly modular and can be used as a complete pipeline, from training to evaluation, or as standalone modules.

---

<sup>3</sup><https://github.com/NVIDIA/DALI>





### 6.3.1. The solo-learn Library: An Overview

Currently, we are witnessing an explosion of works on SSL methods for computer vision. Their underlying idea is to unsupervisedly learn feature representations by enforcing similar feature representations across multiple views from the same image while enforcing diverse representations for other images. To help researchers have a common testbed for reproducing different results, we present **solo-learn**, which is a library of self-supervised methods for visual representation learning. The library is implemented in Pytorch, providing state-of-the-art self-supervised methods, distributed training pipelines with mixed-precision, faster data loading, online linear evaluation for better prototyping, and many other training strategies and tricks presented in recent papers. We also provide an easy way to use the pre-trained models for object detection, via DetectronV2 [469]. Our goal is to provide an easy-to-use library that can be easily extended by the community, while also including additional features that make it easier for researchers and practitioners to train on smaller infrastructures.

### 6.3.2. Self-supervised Learning Methods

We implemented 13 state-of-the-art methods, namely, Barlow Twins [470], BYOL [471], DeepCluster V2 [472], DINO [112], MoCo V2+ [473], NNCLR [474], ReSSL [475], SimCLR [199], Supervised Contrastive Learning [476], SimSiam [477], SwAV [472], VICReg [478] and W-MSE [479].

### 6.3.3. Architecture

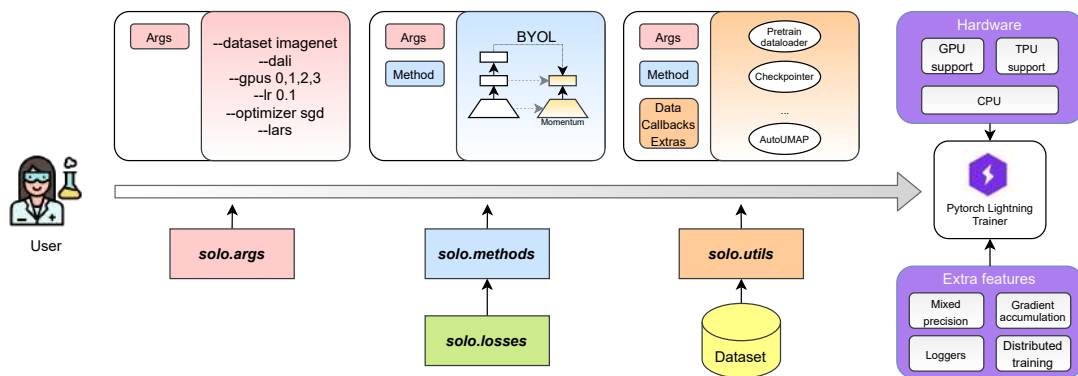


Figure 45. Overview of solo-learn.

In Figure 45, we present an overview of how a training pipeline with **solo-learn** is carried out. In the bottom, we show the packages and external data at each step, while at the top, we show all the defined variables on the left and an example of the newest defined variable on the right. First, the user interacts with **solo.args**, a subpackage that is responsible for handling all the parameters selected by the user and providing automatic setup. Then, **solo.methods** interacts with **solo.losses** to produce the selected self-supervised method. While **solo.methods** contains all implemented methods, **solo.losses** contains the loss functions for each method. Afterwards, **solo.utils** handles external data to produce the pretrain dataloader, which contains all the transformation pipelines, model checkpointer, automatic UMAP visualization of the features, other backbone networks, such as ViT [480] and Swin [481], and many other utility functionalities. Lastly, this is given to a PL trainer, which provides hardware support and extra functionality, such as, distributed training, automatic logging results, mixed precision and much more. We note that although we show all subpackages working together, they can be used in a standalone fashion with





minor modifications. Apart from that, we have documentations in the folder **docs**, downstream tasks in **downstream**, unit tests in **tests** and pretrained models in **zoo**.

### 6.3.4. Comparison to Related Libraries

The most related libraries to ours are VISSL [465] and Lightly [466], which lack some of our key features. First, we support more modern SSL methods, such as BYOL, NNCLR, SimSiam, VICReg, W-MSE and others. Second, we target researchers with fewer resources, namely from 1 to 8 GPUs, allowing much faster data loading via DALI. Lastly, we provide additional utilities, such as automatic linear evaluation, support to custom datasets and automatically generating UMAP [482] visualizations of the features during training.

### 6.3.5. Experiments

**6.3.5.1. Benchmarks.** We benchmarked the available SSL methods on CIFAR-10 [483], CIFAR-100 [483] and ImageNet-100 [484] and made public the pretrained checkpoints. For Barlow Twins, BYOL, MoCo V2+, NNCLR, SimCLR and VICReg, hyperparameters were heavily tuned, reaching higher performance than reported on original papers or third-party results. Table 34 presents the top-1 and top-5 accuracy values for the online linear evaluation. For ImageNet-100, traditional offline linear evaluation is also reported. We also compare with the results reported by Lightly in Table 36.

**6.3.5.2. Nvidia DALI vs traditional data loading.** We compared the training speeds and memory usage of using traditional data loading via Pytorch Vision<sup>4</sup> against data loading with DALI. For consistency, we ran three different methods (Barlow Twins, BYOL and NNCLR) for 20 epochs on ImageNet-100. Table 35 presents these results.

Table 34. Online linear evaluation accuracy on CIFAR-10, CIFAR-100 and ImageNet-100. In brackets, offline linear evaluation accuracy is also reported for ImageNet-100.

Method	CIFAR-10		CIFAR-100		ImageNet-100	
	Acc@1	Acc@5	Acc@1	Acc@5	Acc@1	Acc@5
Barlow Twins	92.10	99.73	70.90	91.91	80.38 (80.16)	95.28 (95.14)
BYOL	92.58	99.79	70.46	91.96	80.16 (80.32)	94.80 (94.94)
DeepCluster V2	88.85	99.58	63.61	88.09	75.36 (75.40)	93.22 (93.10)
DINO	89.52	99.71	66.76	90.34	74.84 (74.92)	92.92 (92.78)
MoCo V2+	92.94	99.79	69.89	91.65	78.20 (79.28)	95.50 (95.18)
NNCLR	91.88	99.78	69.62	91.52	79.80 (80.16)	95.28 (95.28)
ReSSL	90.63	99.62	65.92	89.73	76.92 (78.48)	94.20 (94.24)
SimCLR	90.74	99.75	65.78	89.04	77.04 (77.48)	94.02 (93.42)
Simsiam	90.51	99.72	66.04	89.62	74.54 (78.72)	93.16 (94.78)
SwAV	89.17	99.68	64.88	88.78	74.04 (74.28)	92.70 (92.84)
VICReg	92.07	99.74	68.54	90.83	79.22 (79.40)	95.06 (95.02)
W-MSE	88.67	99.68	61.33	87.26	67.60 (69.06)	90.94 (91.22)

<sup>4</sup><https://github.com/pytorch/vision>





Table 35. Speed and memory comparison with and without DALI on ImageNet-100.

Method	DALI	20 epochs	1 epoch	Speedup	Memory
Barlow		1h 38m 27s	4m 55s	-	5097 MB
Twins	✓	43m 2s	2m 10s	56%	9292 MB
BYOL		1h 38m 46s	4m 56s	-	5409 MB
	✓	50m 33s	2m 31s	49%	9521 MB
NNCLR		1h 38m 30s	4m 55s	-	5060 MB
	✓	42m 3s	2m 6s	64%	9244 MB

Table 36. Comparison with Lightly on CIFAR10.

Method	Ours	Lightly
SimCLR	90.74	89.0
MoCoV2+	92.94	90.0
SimSiam	90.51	91.0

### 6.3.6. Conclusion

Here, we presented `solo-learn`, a library of self-supervised methods for visual representation learning, providing state-of-the-art self-supervised methods in Pytorch. The library supports distributed training, fast data loading and provides many utilities for the end-user, such as online linear evaluation for better prototyping and faster development, many training tricks, and visualization techniques. We are continuously adding new SSL methods, improving usability, documents, and tutorials. Finally, we welcome contributors to help us at <https://github.com/vturrisi/solo-learn>.

### 6.3.7. Relevant publications

- V. Turrisi da Costa, E. Fini, M. Nabi, N. Sebe and E. Ricci, "solo-learn: A Library of Self-supervised Methods for Visual Representation Learning", *Journal of Machine Learning Research*, 23(56):1-6, January 2022. [485].  
Zenodo record: <https://zenodo.org/record/6363321>.

### 6.3.8. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in <https://github.com/vturrisi/solo-learn>.

## 6.4. Uncertainty-guided Source-free Domain Adaptation

**Contributing partner:** UNITN

### 6.4.1. Introduction and methodology

Deep neural networks have proven to be very successful in a myriad of computer vision tasks such as categorization, detection, and retrieval. However, much of the success has come at the price of excessive human effort put into the manual data-labelling process. Since collecting annotated data can be prohibitive and impossible at times, domain adaptation (DA, see [486] for an overview) methods have gained increasing attention. They enable training on unlabelled target data by conjointly leveraging a previously labelled yet related source data set while mitigating *domain-shift* [487] between the two. Such methods predominantly comprise of minimizing statistical moments between distributions [488–491], using adversarial objectives to maximize domain confusion [492, 493], or reconstructing data with generative methods [494].

Albeit successful, the preceding methods mandate access to the source data set during the target adaptation phase as they require an estimate of the source distribution for the alignment. With the emergence of regulations on data privacy and bottleneck in data transmission for large





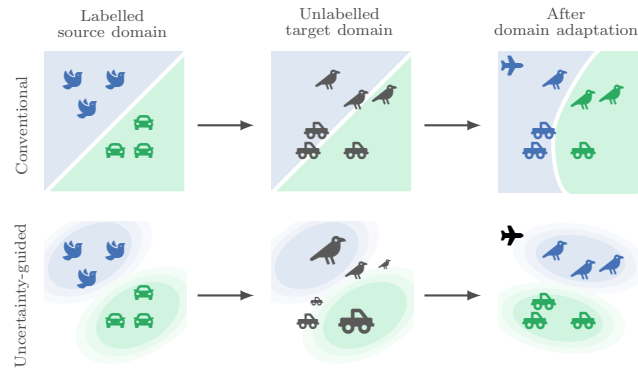


Figure 46. Illustrative sketch of SFDA on a labelled source domain (🐦, 🚗) and an unlabelled target domain (🐦, 🚗) potentially containing additional classes (✈️). The **top-row** shows conventional methods which ignore model uncertainties; the **bottom-row** shows our method which incorporates uncertainties about the predictive model, enabling uncertainty-guided SFDA that is more robust to distribution shifts

data sets, access to the source data can not always be guaranteed. Thus, paving the way to a relatively new and more realistic DA setting, called *source-free* DA (SFDA, [486]), where the task is to adapt to the target data set when the only source of supervision is a source-trained model. SFDA facilitates maintaining data anonymity in privacy-sensitive applications (*e.g.*, surveillance or medical applications) and at the same time reduces data transmission and storage overhead. Towards this goal, recently, several SFDA methods have been proposed that utilize the hypotheses learned from the source data [120, 454, 495]. Notably, SHOT [120] – an Information Maximization (IM) [496] based SFDA method – has demonstrated to work reasonably well on DA benchmarks, sometimes outperforming traditional DA methods. While promising, these conventional SFDA techniques do not account for the uncertainty in the predictions of the source model on the target data. As a by-product, solely maximizing mutual information [496] on the target data can lead to erroneous decision surfaces (see Figure 46 top).

This research argues that quantification of the uncertainty in predictions is essential in SFDA. Depending on the inductive biases of the model, the source model may predict incorrect target pseudo-labels with high confidence, *e.g.*, due to the extrapolation property in ReLU networks [497]. In the literature, uncertainty-guided methods have been proposed in the context of traditional UDA and SFDA settings, employing Monte Carlo dropout to estimate the uncertainties in the model predictions [121, 498]. However, MC dropout requires specialized training and specialized model architecture, suffers from manual hyperparameter tuning [499], and is known to provide a poor approximation even for simple (*e.g.*, linear) models [500–502].

In this research, we propose to construct a probabilistic source model by incorporating priors on the network parameters, inducing a distribution over the model predictions, on the last layer of the source model. This enables us to perform an efficient local approximation to the posterior using a Laplace approximation [508, 509]. This principled Bayesian treatment leads to more robust predictions, especially when the target data set contains out-of-distribution (OOD) classes (see Figure 46 bottom) or in case of strong domain shifts. Once the uncertainty in predictions is estimated, we selectively guide the target model to maximize the mutual information [496] in the target predictions. This alleviates the alignment of the target features with the wrong source hypothesis, resulting in a DA scheme that is robust to mild and strong domain shifts without tuning. We call our proposed method **Uncertainty-guided Source-Free AdaptationN** (U-SFAN). Our approach requires no specialized source training or specialized architecture, opposed to exiting



Table 37. Comparison of the classification accuracy on the OFFICE-HOME for the closed-set setting using ResNet-50. High overall performance signifies milder distributional shift between domains. The improvement of U-SFAN upon SHOT is moderate, but competitive w.r.t.  $A^2$ Net [503] or SHOT++ [504], which require complex training objectives

METHOD	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg.
ResNet-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [492]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
DWT [491]	50.3	72.1	77.0	59.6	69.3	70.2	58.3	48.1	77.3	69.3	53.6	82.0	65.6
CDAN [505]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
SAFN [506]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
SHOT-IM [120]	55.4	76.6	80.4	66.9	74.3	75.4	65.6	54.8	80.7	73.7	58.4	83.4	70.5
LSC [507]	57.9	78.6	81.0	66.7	77.2	77.2	65.6	56.0	82.2	72.0	57.8	83.4	71.3
U-SFAN (Ours)	58.5	78.6	81.1	66.6	75.2	77.9	66.3	57.9	80.6	73.6	61.4	84.1	71.8
$A^2$ Net [503]	58.4	79.0	82.4	67.5	79.3	78.9	68.0	56.2	82.9	74.1	60.5	85.0	72.8
SHOT++ [504]	57.9	79.7	82.5	68.5	79.6	79.3	68.5	57.0	83.0	73.7	60.7	84.9	73.0
SHOT [120]	57.1	78.1	81.5	68.0	78.2	78.1	67.4	54.9	82.2	73.3	58.8	84.3	71.8
U-SFAN+ (Ours)	57.8	77.8	81.6	67.9	77.3	79.2	67.2	54.7	81.2	73.3	60.3	83.9	71.9

Table 38. Comparison of the OS classification accuracy on the OFFICE-HOME for the open-set setting using ResNet-50. U-SFAN improves over SHOT without the need for nearest-centroid pseudo-labelling in the case of open-set SFDA

METHOD	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Avg.
ResNet-50	53.4	52.7	51.9	69.3	61.8	74.1	61.4	64.0	70.0	78.7	71.0	74.9	65.3
ATI- $\lambda$ [515]	55.2	52.6	53.5	69.1	63.5	74.1	61.7	64.5	70.7	79.2	72.9	75.8	66.1
OpenMax [516]	56.5	52.9	53.7	69.1	64.8	74.5	64.1	64.0	71.2	80.3	73.0	76.9	66.7
STA [517]	58.1	53.1	54.4	71.6	69.3	81.9	63.4	65.2	74.9	85.0	75.8	80.8	69.5
SHOT-IM [120]	62.5	77.8	83.9	60.9	73.4	79.4	64.7	58.7	83.1	69.1	62.0	82.1	71.5
SHOT [120]	64.5	80.4	84.7	63.1	75.4	81.2	65.3	59.3	83.3	69.6	64.6	82.3	72.8
U-SFAN (Ours)	62.9	77.9	84.0	67.9	74.6	79.6	68.8	61.3	83.3	76.0	63.9	82.3	73.5

works (e.g., [121, 510]), introduces little computational overhead, and decouples source training and target adaptation.

#### 6.4.2. Experimental results

We conduct experiments on four standard DA benchmarks: OFFICE31 [511], OFFICE-HOME [512], VISDA-C [513], and the large-scale DOMAINNET [514] (0.6 million images). For the experiments in the open-set DA setting we follow the split of [120] for shared and target-private classes.

**6.4.2.1. Evaluation protocol** We report the classification accuracy for every possible pair of *source*  $\mapsto$  *target* directions, except for the VISDA-C where we are only concerned with the transfer from *synthetic*  $\mapsto$  *real* domain. For the open-set experiments, following the evaluation protocol in [120], we report the OS accuracy which includes the per-class accuracy of the known and the unknown class and is computed as  $OS = \frac{1}{K+1} \sum_{k=1}^{K+1} acc_k$ , where  $k = \{1, 2, \dots, K\}$  denote the shared classes and  $(K + 1)^{th}$  is the target-private or OOD classes. This metric is preferred over the known class accuracy,  $OS^* = \frac{1}{K} \sum_{k=1}^K acc_k$ , as it does not take into account the OOD classes.

**6.4.2.2. State-of-the-art Comparison** We compare our U-SFAN with UDA and SFDA methods on multiple data sets for closed-set and open-set settings. First, we compare U-SFAN with the baselines on the most common benchmark of OFFICE-HOME for both closed-set and open-set settings. As can be seen from Tables 37 and 38 we improve the performance over majority of the



baselines. Especially, we consistently improve over SHOT-IM with our method. We also combine the nearest centroid pseudo-labelling, used in SHOT [120], with U-SFAN (indicated as U-SFAN+ in Tables 37 and 39a), and we find that it further helps improving the performance. Notably, the recently proposed A<sup>2</sup>Net [503] (which just addresses closed-set SFDA) outperforms our U-SFAN in a couple of data sets, but uses a combination of several loss functions. Interplay of multiple losses can be hard to tune in practice. On the other hand, our method is simpler, more versatile and works for both the SFDA settings. Given the performance of the SFDA baseline methods in OFFICE-HOME and VISDA-C are relatively high and closer to each other, the domain shift can be considered milder with respect to more challenging data set like DOMAIN-NET.

Table 39. (a) Comparison of the classification accuracy on the VISDA-C for the closed-set DA, pertaining to the Synthetic  $\rightarrow$  Real direction, using ResNet-101. † indicates the numbers of [120] that are obtained using the official code from the authors. Note that several SFDA methods perform equally well for VISDA-C, hinting at saturating performance. (b) Comparison of the average accuracy on the DOMAINNET for the closed-set SFDA using ResNet-50. The SOURCE column indicates the domain where the source model has been trained. The data set being challenging (exhibiting strong domain-shift), the improvement with our U-SFAN over [120] is substantial

(a) VISDA-C

METHOD	ACC.
ResNet-101	52.4
CDAN+BSP [518]	75.9
SAFN [506]	76.1
SHOT-IM† [120]	80.3
U-SFAN (Ours)	81.2
3C-GAN [519]	81.6
A <sup>2</sup> Net [503]	84.3
SHOT† [120]	82.4
U-SFAN+ (Ours)	82.7

(b) DOMAINNET

SOURCE	SHOT-IM [120]	U-SFAN
CLIPART	25.04	30.88
INFOGRAPH	21.58	26.44
PAINTING	23.89	29.91
QUICKDRAW	10.76	10.44
REAL	21.74	29.32
SKETCH	28.87	29.99
AVG.	21.98	26.13

When we compare U-SFAN with SHOT-IM on the challenging SFDA benchmark DOMAIN-NET the advantage of our U-SFAN over SHOT-IM becomes imminent (*cf.* Table 39b). Different from the previous data sets, the difficulty in mitigating domain-shift for DOMAIN-NET is evident from the low overall performance of both SHOT-IM and U-SFAN. This data set can be seen as a real-world example of strong domain-shift. The improvement in the performance of U-SFAN over SHOT-IM for DOMAIN-NET demonstrates that incorporating the uncertainty in the model’s predictions plays a crucial role in SFDA. The conventional approach may overfit to noisy model predictions, leading to poor performance. Whereas, U-SFAN can capture the uncertainty in predictions and down-weight the impact of noisy predictions.

### 6.4.3. Conclusions

Our contributions as follows. (i) We emphasize the need to quantify uncertainty in the predictions for SFDA and propose to account for uncertainties by placing priors on the parameters of the source model. Our approach is computationally efficient by employing a last-layer Laplace approximation and greatly decouples the training of the source and target. (ii) We demonstrate that our proposed U-SFAN successfully guides the target adaptation without specialized loss functions or a specialised architecture. (iii) We empirically show the advantage of our method over SHOT [120] in the closed-set and the open-set setting for several benchmarks tasks and provide evidence for the improved robustness against mild and strong domain shifts.



#### 6.4.4. Relevant publications

- S. Roy, M. Trapp, A. Pilzer, J. Kannala, N. Sebe, E. Ricci, and A. Solin, Uncertainty-guided Source-free Domain Adaptation, European Conference on Computer Vision (ECCV'22) [520]. Zenodo record: <https://zenodo.org/record/7566109>.

#### 6.4.5. Relevant software/datasets/other outcomes

- The Pytorch implementation can be found in <https://github.com/roysubhankar/uncertainty-sfda>.

#### 6.4.6. Relevance to AI4media use cases and media industry applications

Our uncertainty-guided SFDA approach provides a solution to the challenging problem where the task is to adapt to the target data set when the only source of supervision is a source-trained model. This situation can occur frequently in several media industry applications. Specifically, our approach could be used in use case UC2B by providing solutions to analyze the visual content thanks to being able to generalize under domain-gap.





## 7. Deep quality diversity (Task 3.6) – detailed description

**Contributing partners:** UM

QD algorithms have been recently introduced to the EC literature as a way of handling deceptive search spaces. The goal of these algorithms is “to find a maximally diverse collection of individuals (with respect to a space of possible behaviors) in which each member is as high performing as possible” [7]. The inspiration for such approaches is natural evolution which is primarily open-ended—unlike the objective-based optimization tasks to which EC is often applied. While the rationale of open-ended evolution has been previously used as an argument for genetic search for pure behavioral novelty, QD algorithms re-introduce a notion of (localized) quality among individuals with the same behavioral characteristics. QD algorithms attempt to balance between their individuals’ quality and their population’s diversity, and thus media content which have strict quality requirements, such as games that are playable from start to finish, are the ideal arena for advancing quality-diversity.

The aim of Task 3.6 is to couple Deep Neural Network (DNN) architectures with divergent search for transforming exploration, aiming for both diverse and high quality outcomes. Experiments in this deep-learning-based QD search (*deepQD*) approach during the reported period are aligned on two main directions:

D1 improve the definition of diversity based on learnt representations.

D2 promote diversity and quality in existing deep learning generative architectures for media.

### 7.1. Learned Representations as Diversity Metrics to Maximize

**Contributing partners:** UM

#### 7.1.1. Introduction and methodology

Determining an effective representation for content in QD algorithms is crucial to achieve better search space coverage and high-quality output. To this end, UM has investigated using learned content representations through deep learning to enhance a generator’s definition of novelty, with the aim of achieving superior open-ended complexity and diversity [521]. In typical solutions for Procedural Content Generation (PCG) via QD [522], generators use designer-defined representations for generating and evaluating content using algorithms such as MAP-Elites [11]. However, designing the right representation can be very difficult for complex tasks and can introduce search biases, harming the potential of the outputs. Recent studies on achieving better open-ended quality and diversity in PCG-QD have focused on using an intrinsic definition for novelty, which is typically calculated using a learned representation based on the system’s own output.

This work is based on the Deep Learning Novelty eXplorer (DeLeNoX) algorithm [523], which approaches intrinsically defined novelty by assessing diversity in terms of a higher-level representation, determined by a Convolutional Neural Network (CNN) autoencoder. By allowing the generator to adjust its measure for novelty according to its own observations, DeLeNoX is able to continually adapt its focus to search beyond its current biases, a concept which is critical for achieving open-ended evolution. Several solutions have applied novelty defined in a learned latent space to a variety of tasks, such as 2D artifacts [524, 525], discovering interesting behaviors in robots [526] and data efficient search space illumination [527]. However, this approach has remained untested for generating content in 3D domains. On a high level, DeLeNoX alternates between phases



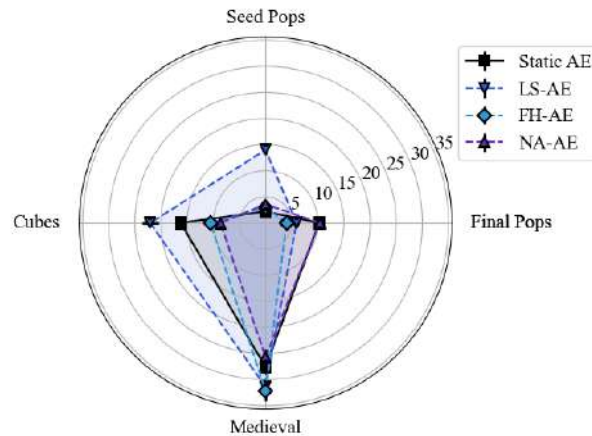


Figure 47. Average reconstruction error (%) and 95% confidence interval tested on four datasets of buildings using the final autoencoder from each experiment (after the 10th round of exploration). Results for the Random AE experiment are omitted due to its poor performance (reconstruction error > 90%) across all tests.

of exploration and transformation. During the exploration phase, the latent space defined by the current autoencoder is explored as thoroughly as possible by applying constrained novelty search [528] to neuroevolution [529]. Repair functions ensure individuals abide to a set of basic desired rules in place of an objective function. During transformation, the most novel individuals from each population form a training set to retrain the autoencoder, modifying the latent space (and distance function) and opening up new areas of the solution space to explore. The new autoencoder is used for the next iteration of the algorithm which can continue until stopping criteria are met. More details on the two phases of the algorithm and the approach taken for building representation can be found in the paper.

In this contribution to T3.6, we build upon the DeLeNoX algorithm, expanding it to generate more complex 3D structures. More specifically, we design a QD generator to autonomously create interesting Minecraft buildings using an intrinsic and open-ended definition of novelty. Sandbox games such as Minecraft [530] are arguably the perfect canvas to illustrate an artificial system’s creativity: their open-ended gameplay allows the player to create any structure that can be expressed as a set of voxels. Our findings suggest that redefining the latent space using novel data with a diverse range of structural complexity improves the generator’s ability to find more complex and novel features in its output.

### 7.1.2. Experimental Results

Since our method focuses on the transformation of the search space through a latent vector when assessing novelty, the experiment explores different ways of training the AE and includes two baselines. The first baseline is a *static* AE which was trained on the seed populations and is not retrained during the transformation phase. The second baseline is the *random* AE, wherein each transformation phase a new autoencoder is generated with random weights and is used as-is for the following exploration phase. The remaining three methods use different training sets during the transformation phase. The *novelty archive* AE (NA-AE) combines all novelty archives from each population in the previous exploration phase to form the training set for the autoencoder. The *latest set* AE (LS-AE) combines only the 100 most novel final individuals of the populations

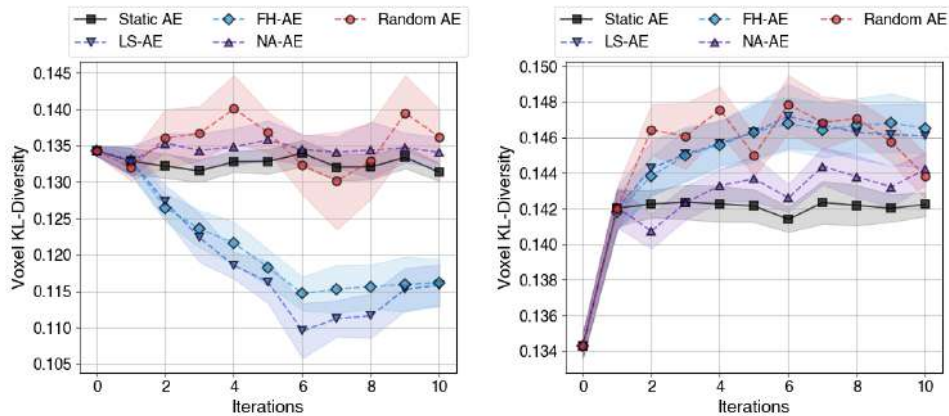


Figure 48. Voxel-based KL Divergence of the populations with respect to each experiment after every round of exploration (left) and each experiment’s populations from the seed populations used to start evolution (right). Results are averaged across all 10 populations using a 95% confidence interval. Iteration zero depicts the average diversity of the seed populations.

in the previous exploration phase into a training set of 1000 individuals, while the *full history* AE (FH-AE) combines the final individuals in every population of every exploration phase so far to train the autoencoder. The experiments were run for 10 iterations of the algorithm, evolving 10 separate populations of 200 individuals each. Each exploration phase runs 100 generations of CPPN-NEAT, and transformation re-trains the autoencoder for 100 epochs. Novelty was calculated using the average Euclidean distance to the 15 nearest neighbors in the latent space, and up to 3 individuals are inserted into the novelty archive per generation. For these experiments, autoencoders were trained to compress the  $20 \times 20 \times 20 \times 5$  lattices into latent vectors of 256 real values. The first iteration of each experiment uses the same set of seed populations, which are also used to pre-train an autoencoder.

We use Kullback-Leibler (KL) divergence as our metric for assessing voxel diversity which has proven efficient for comparing game levels [531, 532]. We also measure the correlation between this KL divergence measure and each experiments’ distance measure in the latent space, as this provides an insight into the regularization of the latent space and the novelty function’s ability to group meaningfully similar individuals together. We also evaluate the reconstruction accuracy of the autoencoders which directly quantifies the model’s ability to identify high-level patterns in the buildings and therefore discover more meaningful novel features. Finally, we provide a qualitative comparison between experiments by visualizing the structures generated and observing the differences in complexity and patterns found to be novel.

**7.1.2.1. Reconstruction Error** Figure 47 shows the reconstruction error measure tested across four different datasets to visualize the autoencoders’ accuracy across a variety of inputs. The seed and final populations refer to populations at the start (before evolution) and end (end of 10th exploration phase) of each respective experiment. The “Cubes” dataset consists of 200 buildings made by randomly generating cuboid hulls of different sizes and applying the repair pipeline to produce material lattices. The “Medieval” dataset consists of a population of buildings generated using the “AHouseV5” filter by Adrian Brightmoore [533] in MCEdit, re-assigning the materials for each voxel through the repair pipeline. Unsurprisingly, the random AE performs very poorly across all four datasets, followed by LS-AE which struggled to reconstruct anything except its own

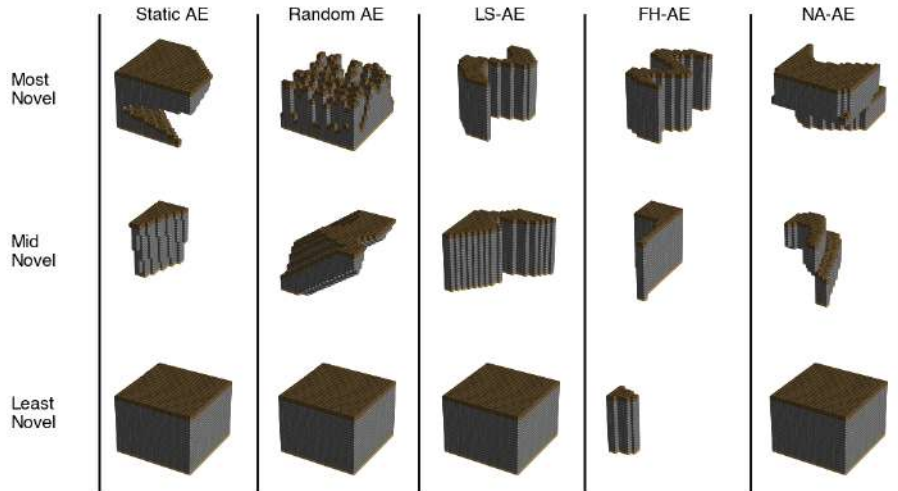


Figure 49. Visualization of individuals from each experiments' final population sorted according to their novelty score (minimum, median, maximum). To evaluate novelty the final autoencoder from each experiment was used (excluding the seed and static AE populations which used the seed model).

final population. The NA-AE proved to be the most robust model when given completely unseen data, displaying the best reconstruction accuracy for the Medieval and Cubes datasets. The FH-AE shows the best performance on its final populations, though (like the static AE) struggled slightly compared to NA-AE on completely unseen data. The NA-AE seems to benefit from having the largest amount of (and most diverse) training data for transformation compared to the rest of the experiments.

**7.1.2.2. Voxel KL-Divergence** The results in Figure 48 show that whilst the LS-AE and FH-AE produced the least diverse individuals compared to their own populations, these same individuals were the most diverse from the initial seeds. On the other hand, the static AE produces more diverse content in the voxel space, without varying over time even in comparison to the initial seeds; this is expected as the autoencoder is not retrained between exploration phases. Interestingly, the NA-AE produces a similar trend to the static AE in both measures, even though it is trained on the largest dataset of individuals during transformation. The random AE produces marginally more diverse content for both measures, albeit with a larger deviation which is likely caused by the randomized weights of the autoencoder. Our results also show that there is a clear linear correlation between the two diversity measures for the NA-AE experiment's final populations, with a Pearson correlation of 0.84. However, the LS-AE and FH-AE distance functions both produce significantly weaker correlations between the two measures, with a Pearson correlation of 0.53 and 0.35 respectively.

**7.1.2.3. Qualitative Comparison** By looking at these examples in Figure 49 we can get a qualitative idea of how novelty and complexity is evolving over time. The seed population and static AE share similar high-level patterns which is understandable given they use the same autoencoder and originate from the same latent space. The effect of the lack of training for the random AE







experiment is clearly reflected in results which are far noisier than the other experiments. The LS-AE, FH-AE and NA-AE experiments show slight differences in the overall structures generated compared to the seed set, though there is no significant jump in structural complexity. This indicates that whilst novelty search is promoting diversity in the latent space, it does not guarantee diversity in the phenotype space and does not explicitly evolve towards desired qualities as in quality-diversity algorithms such as MAP-Elites [11].

### 7.1.3. Relevant publications

- Matthew Barthet, Antonios Liapis and Georgios N. Yannakakis: "Open-Ended Evolution for Minecraft Building Generation," in IEEE Transactions on Games, 2022 (accepted). [534]. Zenodo record: <https://zenodo.org/record/7879128>.

### 7.1.4. Relevant software/datasets/other outcomes

- The article published as the culmination of this research is available as an interactive paper (which allows for better interaction with its content) at <https://minecraft.institutedigitalgames.com/https://minecraft.institutedigitalgames.com/>.

### 7.1.5. Relevance to AI4media use cases and media industry applications

Our tools on deep learned diversity metrics are applicable to any creative domain as they provide novel ways to generate diverse content without requiring ad-hoc designer-specified directions for this diversity. However, they ideally contribute to Use Case 5 (AI for Games) as experiments have so far focused on generating diverse in-game structures for voxel based games such as Minecraft. This work can be extended to tackle other creative domains relevant to Use Case 5 such as visuals or sound.

## 7.2. Quality Diversity search on the Latent Space

**Contributing partners:** UM

### 7.2.1. Introduction and methodology

A direct application of the deepQD vision is by combining latent space representations with evolution driven by QD search to change the latent parameters of an artifact. In this case, we focus on AI Art generators and attempt to address the visual diversity of the output through a QD algorithm, named NSLC. Our proposed methodology combines refinement and exploration cycles to generate visually diverse images. The refinement cycle uses VQGAN latent vector backpropagation to convert random noise into desirable images, whilst the exploration cycle employs NSLC on the latent vector. We use a pre-trained VQGAN model based on the WikiArt dataset [535], which contains over 81,000 images across various art styles. The generated images have dimensions of 384 by 384 pixels, which the VQVAE quantises into square blocks of 16 by 16 pixels. The image is thus represented as a latent vector of 576 integers representing indices from the VQVAE code book.

To begin the experiment, each latent vector is randomised using fractal noise and its CLIP embedding is obtained. In subsequent iterations, we employ the negated cosine similarity of CLIP as a loss function for backpropagation. The goal is to refine the latent vector to generate an image that aligns better with the given semantic prompt. Figure 50 visualises this process. Since the



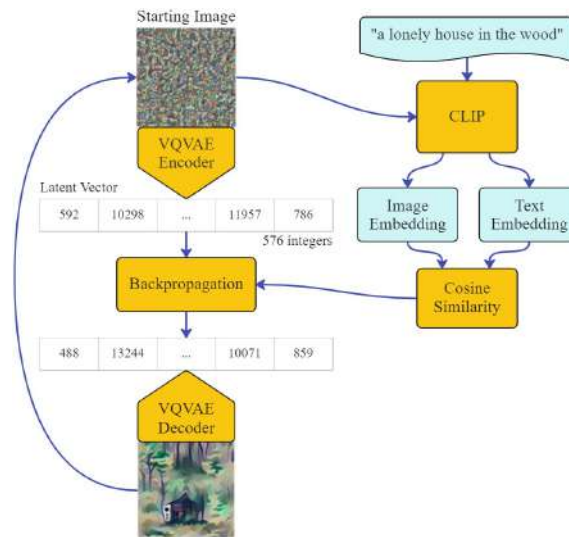


Figure 50. Refinement iterations consisting of CLIP-guided backpropagation.

latent vector consists of integers, backpropagation is actually performed on the internal tensor representation within VQGAN, which consists of floating-point numbers. The code used is based on the Pixray<sup>5</sup> Python library.

During exploration the latent vectors undergo mutation operations in which 5% of its genes (randomly selected) are replaced with random integers between 0 and 16,384 (the code book size). This mutation rate allows for perceptible changes in the image without making it unrecognisable. We consider the 15 nearest individuals to calculate both the novelty and the local competition scores. The top three novel individuals in each generation are added to the novelty archive, which is reset at the start of each exploration cycle. This approach strikes a balance between computational requirements and maintaining diversity.

To process the novelty and local competition scores as a multi-objective optimization problem, we employ the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [536] using the Pymoo Python library [537]. A minimal Pareto front is calculated based on the two objectives, and individuals closer to this front dominate the remaining population and are selected for the next generation. If more individuals are needed, another Pareto front is calculated and individuals are selected accordingly. In cases where there are more individuals on the Pareto front than required, individuals are selected to introduce sparsity based on the Manhattan distance within the search space.

Determining the diversity of the generated images was a considerable challenge. The concept of diversity plays a crucial role in our NSLC problem definition, and it thereby greatly influences the outcome of the exploration cycles. While humans can easily perceive visual similarities between two images, quantifying similarity or diversity in a straightforward metric presents several challenges. We hereby explore two distinct methods, namely Chromatic and Vision Transformer diversity, and more detail on these approaches can be found in the paper.

<sup>5</sup><https://github.com/pixray/pixray>

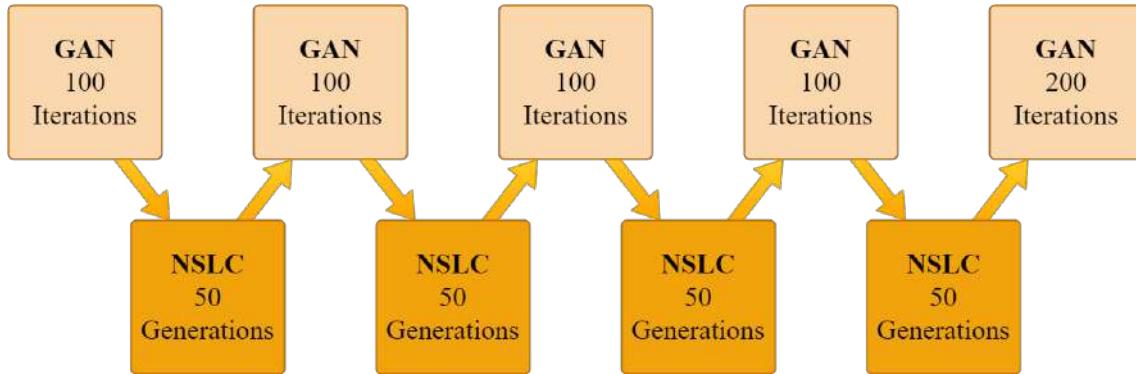


Figure 51. Structure of the experiments alternating between GAN refinement and NSLC exploration cycles.

### 7.2.2. Experimental Results

The experiments were conducted using five Semantic Prompts commonly employed by the community<sup>6</sup>. These were “a lonely house in the woods (SP1)”, “a pyramid made of ice (SP2)”, “artificial intelligence (SP3)”, “cosmic love and attention (SP4)”, and “fire in the sky (SP5)”.

Initially, a population of 500 images is generated from latent vectors encoded from a set of randomly generated fractal noise images. This same initial population is used for all algorithm variations and across all prompts. To establish a baseline, the backpropagation refinement process is run without interruption for each initial latent vector, generating the final baseline population (referred to as GAN-BSL). Preliminary experiments have shown that the image composition stabilises after 600 iterations, with minimal changes occurring beyond this point.

For the NSLC experiments, the refinement process is interrupted at intervals of 100, 200, 300, and 400 iterations, capturing the latent vectors at each point to create an initial population for NSLC evolutionary cycles. These exploratory cycles evolve for 50 generations, guided by either ViT (NSLC-ViT experiment) or HSV (NSLC-HSV experiment) distance metrics. The resulting evolved population is then subjected to a final backpropagation cycle of 200 iterations for a total of 600 refinement iterations throughout the experiment. Figure 51 provides a visual representation of this process.

Assessing the novelty and quality of the generated output is a complex task [538]. In this work, we align these concepts with the quality-diversity characteristics of NSLC and employ the following performance metrics for comparing different algorithms:

- **Mean fitness:** This metric calculates the average CLIP score across all 50 images in the population, representing the overall quality.
- **Mean ViT diversity:** This measures the average ViT distance from the 15 nearest neighbors per individual, considering only the current population for finding nearest neighbors (no archive).
- **Mean HSV diversity:** Similar to mean ViT diversity, this metric calculates the average HSV distance from the nearest neighbors, using the HSV metric for measuring distance and finding the nearest neighbors.

The *process* followed by the algorithms tested is of equal interest as the *product* at the end of 600 iterations [539]. Figure 52a shows how the mean fitness (CLIP score) fluctuates throughout the

<sup>6</sup><https://github.com/lucidrains/big-sleep>



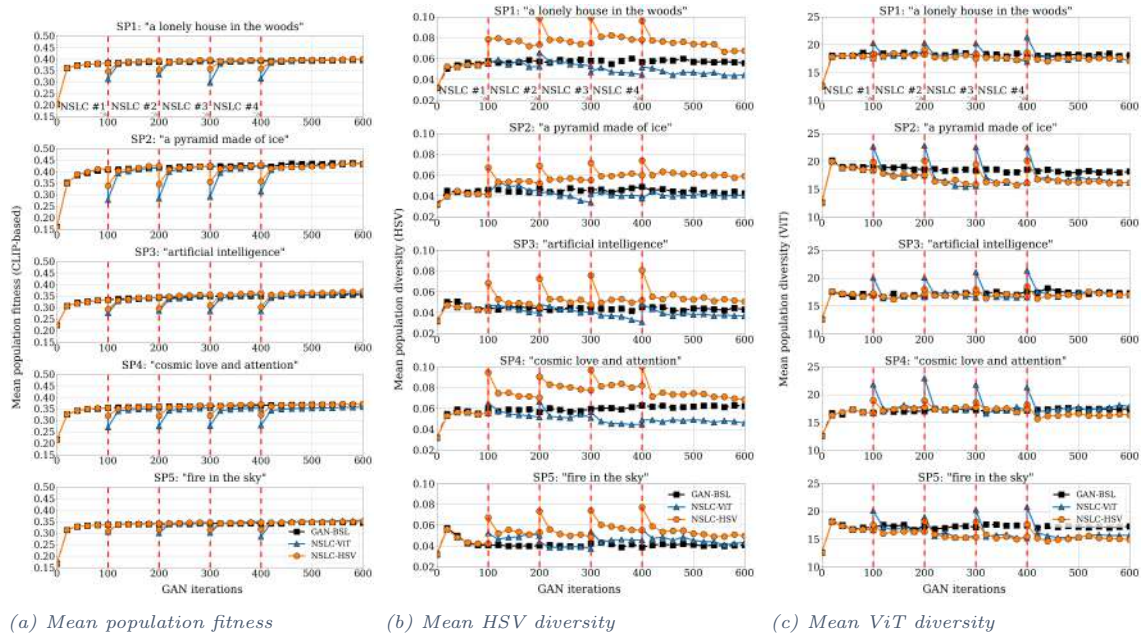


Figure 52. Progression of the performance metrics over GAN iterations. The iterations at which evolutionary NSLC cycles were performed are marked in red.

different cycles of the experiment. The uninterrupted GAN-BSL exhibits an initial rapid increase in fitness over the first 20 iterations and slowly improves thereon. In the NSLC experiments, the evolved population’s fitness drops by an average of 12% for NSLC-HSV and by 21% for NSLC-ViT at the start of each evolutionary cycle. This drop remains almost equally substantial when NSLC is applied at later iterations, even though the images’ composition is well-established at those stages. After each NSLC cycle, the GAN rapidly restores the CLIP score to a similar level as the GAN-BSL at the same iteration. At the end of the 600 iterations, all three algorithms reach a similar mean fitness score, with the NSLC variants commonly reaching slightly higher CLIP scores than the baseline. Overall, NSLC-HSV exhibits a more stable performance, reaching on average 1.5% higher mean fitness than GAN-BSL, whilst, NSLC-ViT has more fluctuations between prompts and reaches an average increase of 0.7% from the GAN-BSL mean fitness. The biggest increase in CLIP score is for SP3, where NSLC-HSV outperforms GAN-BSL by 3.9% in terms of mean fitness.

Figures 52b and 52c illustrate the mean diversity of the population evaluated using both image distance metrics for all three experiments (GAN-BSL, NSLC-ViT, NSLC-HSV), despite this not being the target novelty measure in all cases. Both image distance metrics show a rapid increase in diversity over the first 20 GAN iterations. This may be attributed to the fact that the initial noise evaluates to a low diversity, compared to the forming images, despite their tendency towards a generic style imposed by the manifold. The diversity of the GAN-BSL stays fairly stable after these first few iterations, or tends to drop. This is most pronounced in SP5 for both diversity measure; we hypothesise that the (literal) prompt itself pushes images that are fairly similar in colour (red and blue) and in terms of image classification.

The NSLC variants exhibit an increase in diversity after each exploration cycle for the distance metric targeted by novelty search. Interestingly, NSLC-HSV manages to increase both HSV diversity and ViT diversity, even if it evolves towards the former. On average, in each exploration cycle NSLC-ViT increases ViT diversity by 25% while NSLC-HSV increases ViT diversity by 10% (per



prompt). NSLC-ViT however underperforms in terms of HSV diversity, with minor or no increases after each cycle. On the other hand, with NSLC-HSV we observe an average increase of 43% in HSV diversity after each cycle (per prompt).

The population resulting after an NSLC cycle is more diverse but less fit, and during GAN cycles the diversity quickly drops as CLIP score increases. These refinement iterations tend to lower the ViT diversity, surprisingly more than in the GAN baseline experiment, despite both NSLC variants managing to increase ViT diversity. Furthermore, GAN iterations rapidly increase ViT diversity from the random seed images, but not from the noisy images produced by NSLC cycles at iterations 100, 200, 300, 400. After 600 iterations, the final images of NSLC-HSV have an average of 6.3% increase in HSV diversity compared to GAN-BSL but an average 11.5% decrease in ViT diversity, per prompt. The final images for NSLC-ViT however are less diverse for both ViT and HSV compared to the GAN baseline (by 5.8% and 13.7% respectively).

### 7.2.3. Relevant publications

- Marvin Zammit, Antonios Liapis and Georgios N. Yannakakis: "Seeding Diversity into AI Art," in Proceedings of the International Conference on Computational Creativity, 2022. [540]. Zenodo record: <https://zenodo.org/record/6545663>.

### 7.2.4. Relevance to AI4media use cases and media industry applications

Our algorithms for generating interesting art from prompts can be used by content creators and the media industry as new way of prompting human creativity through output that is guaranteed to be visually diverse. This work ideally contributes to Use Case 5 (AI for Games) where it can be used to help generate interesting visuals for game artworks and in-game content.

## 7.3. Cross-domain Quality-Diversity search

**Contributing partners:** UM

### 7.3.1. Introduction and methodology

As an extension to the QD work in Section 7.2, we need not only evolve the latent representation of the image, but could also evolve the prompt itself. In the follow-up research, which is still ongoing and described below, we evolve a longer prompt (as game description) along with the associated image it represents. The contribution below focuses on the cross-facet (text to image) evaluation, which acts as *Quality* in the QD paradigm, while facet-specific *Diversity* measures drive the search of each component of the final artifact. Moreover, the below study leverages a highly successful and extendable AI algorithm, MAP-Elites [11]. In our proof of concept experiment, we produce the titles and descriptions by employing a GPT-2 language model. We first extract a dataset composed of video game titles and descriptions from the Steam<sup>7</sup> platform's games catalogue. By stripping away titles which are not video-games (applications, videos, etc...) and those which do not have a listing in the English language, we end up with a list of approximately 72,000 titles and corresponding descriptions.

We then fine-tuned a first GPT-2 model on game titles. Since the corpus was not large, we used the smallest variation of the model, which has 124 million parameters. The model was trained by the list of titles delimited by a beginning and an end token, given in the format " $\langle begin \rangle | game$

---

<sup>7</sup><https://store.steampowered.com/>



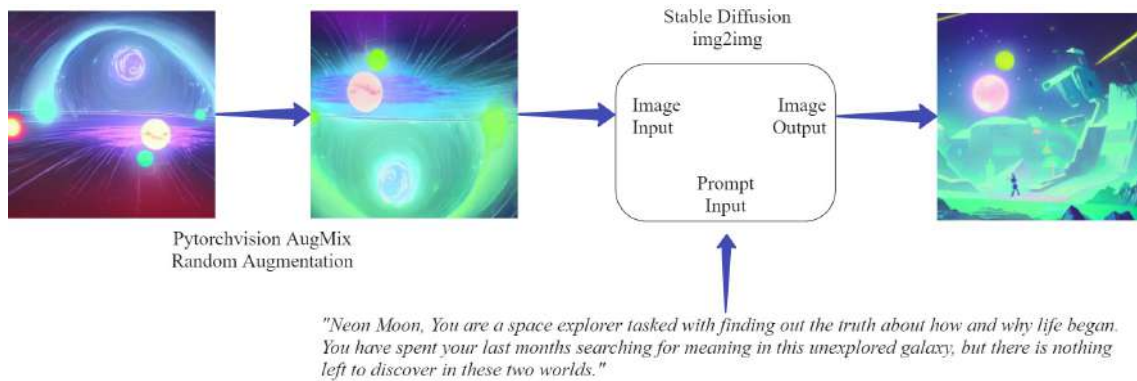


Figure 53. The mutation strategy for the image modality.

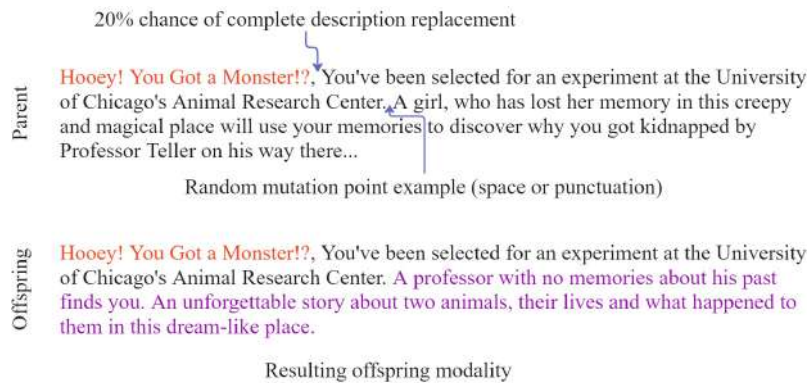


Figure 54. The mutation strategy for the text modality.

title|< end >|". During inference, the begin token is supplied to the fine-tuned model and a novel title is generated. We generated 100 titles in this manner and chose 7 which we hypothesised would offer enough potential diversity in the corresponding descriptions and the generated images. The selected titles were "Neon Moon", "Lion King", "Hexgrave", "Fantasy Fables: The Legend of the Flying Sword", "The Princess of Thieves", "The Shadow Warrior 2: Shadows of the Past", "Hooey! You Got a Monster!?". A second GPT-2 model of the same size was fine-tuned on both the game titles and their accompanying descriptions, in the format "*< begin >|game title|< body >|game description|< end >|*". During inference, the hypothetical titles generated by the first model were delimited by the begin and body tokens, and fed into the fine-tuned descriptions model, resulting in a corresponding hypothetical description. 100 descriptions for each of the chosen titles were generated as an initial population.

To generate the cover images for the games, both the title and descriptions were passed to a Stable Diffusion model as a semantic prompt in the format "title, description". In addition to this, the model can also be given negative prompts in order to avoid specific occurrences in the output image. For all generated images, the string "duplication, ugly, text, bad anatomy" was used as a negative prompt to improve the aesthetic quality of the output. An initial population of 100 text-image pairs was thus generated for each title. We used the cosine similarity between the CLIP embeddings of the prompt and image as a measure of fitness for the genetic algorithm.

The selection strategy for new candidates to evolve is based on an UCB algorithm [541] taking



into account frequency of individual selection, which has been shown to improve the MAP-Elites coverage of its feature map [542]. Upon selection, the individual is subjected to a mutation of either its image or text, with an equal probability for each modality. The mutation strategy for the image modality is a phenotypic one, and is carried out in two steps. First an AugMix [543] image augmentation function from the Torchvision software library [544] is applied. The resulting image is then used as an input to a Stable Diffusion image-to-image, text-guided model, together with the original prompt (Figure 53). In order to modify the description, a stochastic selection of a space or punctuation character is made. The text is truncated from the chosen point on, and the remaining portion is introduced back into the fine-tuned descriptions GPT-2 model to generate a new rendition of the omitted segment. To prevent the initial part of the text from remaining unchanged, a 20% probability of beginning the rewriting process anew has also been incorporated (Figure 54).

In order to have a more meaningful BC, we used Latent Dirichlet Allocation (LDA) [545] to classify the descriptions from the Steam video game dataset into topics. Since this algorithm requires prior knowledge of the number of topics for classification, we trained the algorithm on a number of topics varying from 4 to 30, and perplexity and complexity metrics were recorded for each in order to determine the optimal number of topics. From this initial study, the LDA model with 16 topics was determined to be the most suitable. Each description is processed using this topic modeller and a resulting set of probabilities to fit within each topic is given. In the event that the most likely subject of a description fails to attain a probability of at least 40%, or if there exists a statistical equivalence among the highest probabilities, the description is deemed to be unclassified.

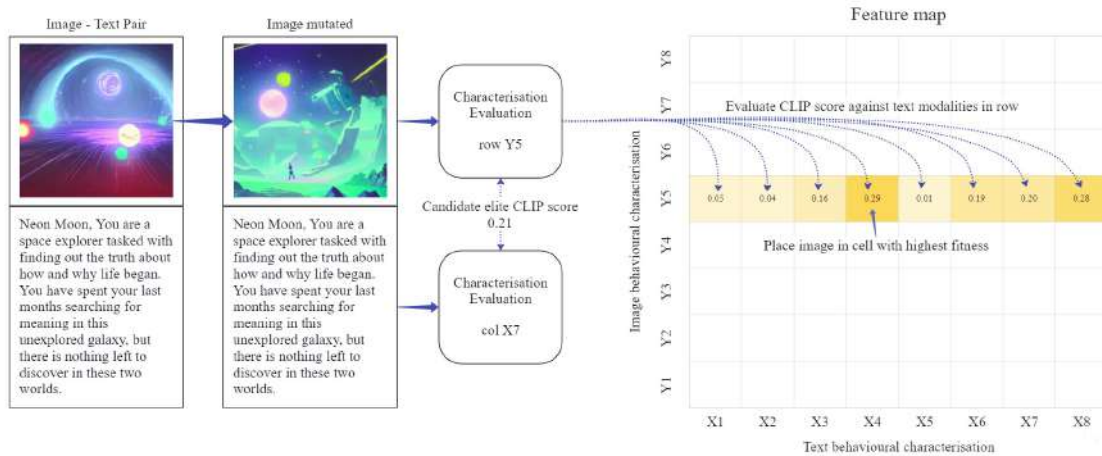
The selected BC was a combination of the image complexity and colourfulness. Image complexity was calculated by applying Holistically-Nested Edge Detection (HED) [546] to the image and taking the ratio of resulting edge pixels to the total pixels in the image. This is reminiscent of previous image complexity calculations which were based on Sobel or Canny filters [547], but HED outlines are generally more accurate and less noisy than either of these filters. Image colourfulness is based on the quantitative measure of the perceived colourfulness or saturation of an image [548]. It aims to capture the amount of variation and intensity of colours present in an image. The images were grouped into 4 'bins' of complexity and 4 of colourfulness, resulting in a BC of 16 bins, to match the amount of topics in the textual modality.

Throughout this work, we also investigated an enhancement to the original MAP-Elites algorithm when applied to different modalities. During the offspring creation process at each generation of the algorithm, only one modality is mutated. We propose that the newly generated modality, say an image, is compared to the descriptions of the individuals in the same axis of the feature map since all of these belong to the same BC bin. The match that has the highest fitness greater than the individual already present in the respective cell is placed accordingly. We call this placement method MELiTA. This is illustrated in Figure 55.

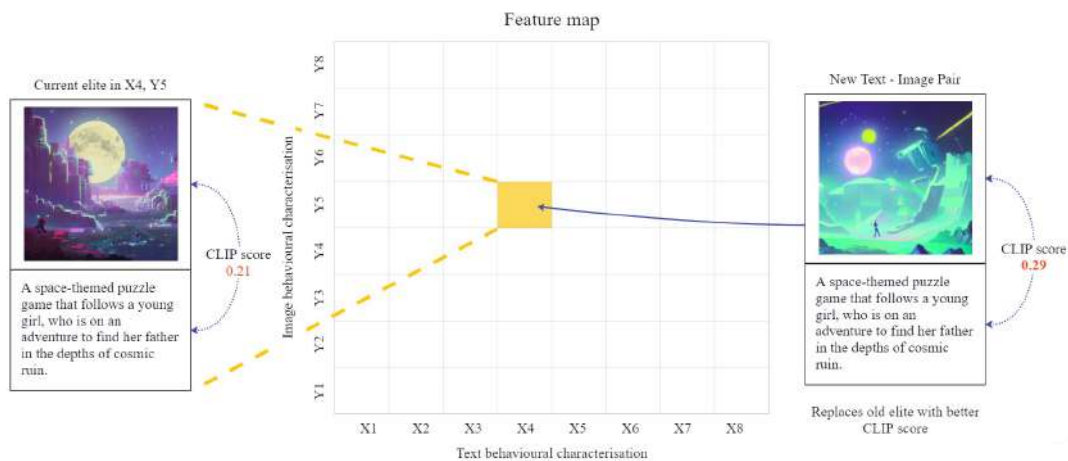
### 7.3.2. Experimental Results

To facilitate a comparative analysis between the baseline MAP-Elites placement and the MELiTA placement methods, a series of experiments were conducted for each selected game title, employing both algorithms. A total of 10 independent runs were performed for each method. Each experiment consisted of 2000 generations, wherein one individual was chosen for mutation in each generation. The mean fitness, coverage and Quality Diversity score were computed for the final state of the feature map and Wilcoxon Rank Sum tests [549] were carried out on these values batched by game titles, to identify any statistically significant differences in the resulting values. The tests were also carried out on the global values, i.e. across all titles. Results of these experiments are in Table 40,





(a) One modality is mutated and a transverse evaluation of the fitness with the existing individuals' other modality is carried out.



(b) The new text-image pair is replaced in a single cell where the fitness is highest and exceeds the current elite in that location.

Figure 55. The novel transverse assessment method being proposed for offspring placement in the MAP-Elites feature map.

with significantly higher values for MELitA in several cases.

It is evident that there is a significant increase in the mean fitness of the individuals within the feature maps. The QD score and coverage were also noticed to be often higher, but not consistently enough to be statistically significant. It was also noticed that the transverse assessment led to more stagnation in the image composition, as mutated individuals from the same parent image were frequently distributed across other cells. Overall, this method shows promise in maximising fitness across the feature map, but it needs to be studied further.





Title	MAP-Elites baseline			MEliTA		
	Mean fitness	Coverage	QD Score	Mean fitness	Coverage	QD Score
T1	0.267	0.477	32.631	0.286	0.475	34.860
T2	0.265	0.345	23.557	0.313	0.349	27.926
T3	0.265	0.532	36.140	0.280	0.491	35.125
T4	0.262	0.399	26.817	0.278	0.366	26.045
T5	0.266	0.405	27.584	0.286	0.370	27.009
T6	0.254	0.305	19.862	0.285	0.285	20.773
T7	0.285	0.441	32.127	0.300	0.459	35.283
Global	0.267	0.415	28.388	0.290	0.399	29.574

Table 40. Results showing the mean fitness, coverage and QD score across all runs for each experiment. Green cells denote a statistically significant higher value (Wilcoxon Rank Sum test  $p$ -value  $< 0.05$ ).

### 7.3.3. Relevance to AI4media use cases and media industry applications

Our tools on Map Elites with Transversal Assessment are applicable to any multi-faceted creative domain (e.g. film-making), but are ideally contributing to Use Case 5 (AI for Games) where content of different facets combine into a playable experience. Experiments so far have focused on conceptual design of games' descriptions (text) and concept art (visuals) but extending this to background music (sound) as an additional facet can also connect it to other activities undertaken in Use Case 5.

## 7.4. User-Controllable Quality Diversity Search

**Contributing partners:** UM

### 7.4.1. Introduction and methodology

In this contribution, we introduce a novel IEC algorithm called User Controlled MAP-Elites (UC-ME) aiming to provide a high degree of user control, with a reduced degree of user fatigue. We showcase that this is achievable by exploiting the illumination capabilities of QD algorithms. We implement this concept by modifying the basic operation of MAP-Elites [550], a popular QD algorithm used also in the contribution of Section 7.3, and devising the following interaction loop: we constrain the algorithm's operation within a window that covers a small region of the feature map, where it locally expands the archive for a number of generations. Afterwards, design alternatives are sampled from within the window and presented to the designer. Finally, the user's selection determines where the window will move towards next.

UC-ME starts by producing a number of initial individuals through a random initialization method and placing them in the MAP-Elites archive according to their behavioral characterization. This step seeds the archive to enable interaction with the human user. The initial selection window of size  $w \times w$  is centered at the cell with the mean BC values of existing elites, or the nearest elite if that cell is unoccupied. The window size ( $w$ ) is a parameter of UC-ME which should be much smaller than the resolution of the feature map. After the algorithm has been initialized, the interactive session can begin. During an interactive session, the following steps are repeated indefinitely, until the designer decides to end it. The algorithm samples  $D$  design alternatives, from





within the selection window, to be shown to the designer as options to select from, who selects one preferred design. The selection window is centered at the coordinates of the designer's last selection. The algorithm operates for  $N_e$  evaluations, selecting parents from within the window. The mutated offspring are evaluated and placed at their corresponding archive cell, based on their Behavioral Characterization coordinates, without being constrained by the window. In case an offspring lands on an already occupied cell, the individual with the highest fitness survives.

The algorithm samples a number of design alternatives to present to the user, from within the selection window. We only test UC-ME with four alternatives in this work, and examine six methods for Design Alternatives Sampling (DAS), which are all stochastic to some degree. *Random* ( $A_R$ ) samples 4 elites from the window at random. *Quadrants* ( $A_Q$ ) and *Squares* ( $A_S$ ) split the window into 4 equal sections, using the diagonals ( $A_Q$ ) or the  $x$ - $y$  axes ( $A_S$ ). One individual is sampled randomly from each section. *Edges* ( $A_E$ ) samples one individual at random from each of the 4 edges of the window. If no individual is on the edge, then the nearest individuals to that edge are preferred. *Corners* ( $A_C$ ) samples one individual per corner of the window, or the nearest individual to that corner (as in  $A_E$ ). In *Medoids* ( $A_M$ ) the coordinates of the individuals within the selection window are used as data points in a  $k$ -medoids clustering algorithm, where  $k = 4$  in this work. The four medoids of these clusters are shown to the user.

We follow the methodology of [12] for our use case of layout generation, where the problem definition is a set of topological and other constraints, and the output is a geometrical solution that respects these constraints. We summarize the process for this use case below; more details can be found in [12]. We chose to focus on this complex problem for two reasons: first, architectural layouts are characterized by many quantifiable, yet subjective, features, making them ideal for testing MI-CC methods. Second, this task offers an opportunity to test the proposed methodology on a constrained domain, showcasing its extensibility to other MAP-Elites variants.

#### 7.4.2. Experimental Results

As a specific case study for architectural layout generation, we set up an experiment with a specific design specification for a medium-size apartment, algorithm parameters, controllable (artificial) users to test the algorithm. We also identified a plethora of performance metrics in order to assess the general and user-specific efficacy of the algorithm. More detail on the parameters of the algorithm, the performance metrics used, and our implementation of artificial users to test the performance of our approach can be found in the paper. Results shown below are from 10 independent runs, and significance is established via Student's  $t$ -test with  $p < 0.05$ .

**Comparisons between DAS methods:** Table 41 summarizes a comparison between different DAS methods. Treating each artificial user as a separate experiment, we evaluate in how many pairwise comparisons the DAS method had significantly better metrics than another method, after 10 selections. With 6 DAS methods (i.e. 5 comparison per method) and 12 artificial users, the maximum number in each cell is 60. The Bonferroni correction [551] is applied for multiple comparisons. Note that we also tested for all other metrics (pertaining to QD and USC), but there was almost no difference between the DAS methods.

Table 41 indicates that the Edges and Corners DAS methods have a clear advantage in local diversity. This is expected, as both methods prioritize individuals that are as far away from the selection window's center as possible. Intuitively the Corners method is slightly better at local diversity as its first choices have the absolute maximum distance of all candidates in the window. We note that in terms of local mean User Selection Criterion (USC) and USC efficiency there are no clear winners, with  $A_E$ ,  $A_S$  and  $A_M$  being slightly more efficient than other methods. Even before 10 selections,  $A_C$  tends to reach the edge of the feasible space with the best USC (see Fig. 57) and after that the window moves erratically—and inefficiently. Based on the findings of Table 41,



Parameter	$A_R$	$A_Q$	$A_E$	$A_S$	$A_C$	$A_M$
Local Diversity	0	0	48	0	58	13
Local Mean Fitness	17	2	0	2	0	31
Local Mean USC	0	2	2	1	7	0
USC Efficiency	0	10	10	10	7	6

Table 41. Comparison between all DAS Methods for local QD metrics. Values show how many times this DAS method was significantly better ( $p < 0.05$ ) than another DAS method in the same experiment. Results are collected after 10 selections.

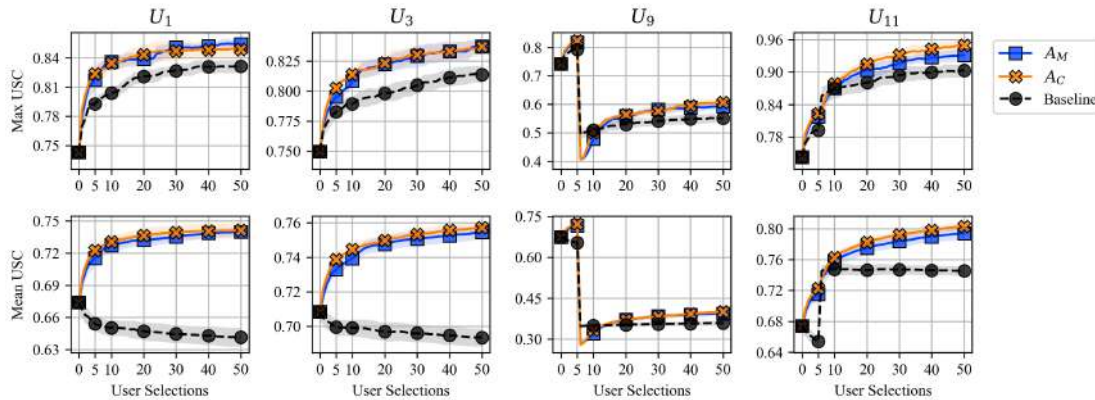


Figure 56. Charts display the value of Max and Mean USC of four different artificial users ( $U_1$ ,  $U_4$ ,  $U_9$  and  $U_{11}$ ), comparing the Quadrants ( $A_C$ ) and Medoids ( $A_M$ ) DAS methods with the baseline (MAP-Elites without user control). Values are averaged across 10 different runs and shaded regions capture the 95% confidence interval.

the Corners DAS method has the best performance due to a higher diversity of shown individuals, while still being fairly efficient. The Medoids method shows fitter individuals to the user than other DAS methods, while still being somewhat efficient at adapting to the USC. We thus test these DAS methods against a baseline MAP-Elites.

**Comparisons with MAP-Elites:** Based on the comparisons between DAS methods, we focus on comparing the Corners and Medoids methods against a baseline MAP-Elites which does not consider the user’s taste and performs unguided exploration of the search space. The baseline implements FI-MAP-Elites [12] and randomly selects random individuals to mutate, alternating between the two archives.

Our results show that unguided MAP-Elites has better coverage of the problem space and thus a higher QD score, across all experiments. This is not surprising, as UC-ME drives search towards specific parts of the problem space (and regions of the feature map), while MAP-Elites covers as much of the feature map as possible. We also note that there are no differences in terms of maximum fitness. This is somewhat surprising, since different parts of the feature map (targeted by different users) may not have equally good fitnesses. It seems that finding a highly fit individual is not challenging in this use case.

As expected, the unguided exploration of the baseline MAP-Elites performs worse than both UC-ME versions for maximum and mean USC score of all elites in the archive. The  $A_M$  method is less efficient at reaching very high USC scores, compared to  $A_C$ ; this is not surprising since the latter moves the selection window toward regions of the problem space with high USC faster. Mainly due to a higher mean USC, it is not surprising that the mean W-USC is higher for UC-ME

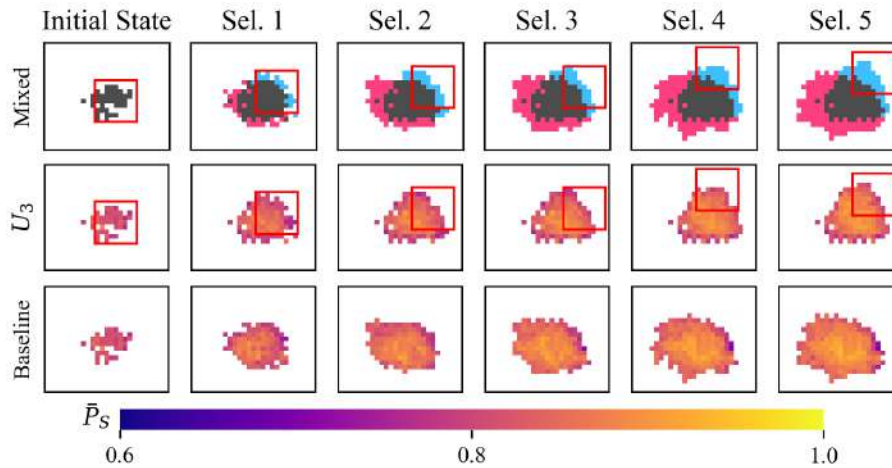


Figure 57. Behavioral space exploration for the baseline MAP-Elites (bottom row) and UC-ME with Corners DAS guided by  $U_3$  (middle row), for the first 5 selections. Their shared color scale (shown at the bottom) represents fitness  $\bar{P}_s \in [0.6, 1]$ . The top row shows coverage differences: red cells are discovered only by the baseline, blue cells are discovered only by UC-ME and gray cells are common. In these figures the x axis is  $\bar{C}_s \in [0.44, 0.86]$  and the y axis is  $\bar{O}_\theta \in [0.61, 0.97]$ .

variants compared to the baseline. The higher coverage of the baseline, however, leads to higher values in the sum of W-USC scores among all elites, similarly to the QD Score.

Figure 56 shows a comparison between the progression of mean USC and max USC for four indicative artificial users: two consistent ( $U_1, U_3$ ) and two that change criteria after 5 selections ( $U_9, U_{11}$ ). It is evident from Figure 56 that while the unguided MAP-Elites can accidentally find regions of the problem space with a high USC (i.e. max USC keeps increasing), this is not a guarantee for the broader population (mean USC may increase or decrease) depending on which parts of the space are more easily reachable. As expected, UC-ME variants consistently improve both USC measures as the archive is driven by local QD towards specific parts of the space.

Figure 56 also shows how the algorithms handle abrupt changes in user criteria after the 5th selection ( $U_9, U_{11}$ ).  $U_9$  has a more abrupt change as it suddenly targets the opposite of its previous criterion, causing a drop in USC. It takes several user interactions to move the window towards more appropriate regions of the problem space, but after 5 selections from the criterion change, UC-ME approaches the mean and max USC of the baseline which has been evolving for both high  $\bar{C}_s$  and low  $\bar{C}_s$  (both captured in  $h_9$ ). Given enough time, both methods surpass the baseline (e.g. after 25 selections).  $U_{11}$  is not as “aggressive” in changing its mind; indeed, even unguided MAP-Elites can find individuals with high  $\bar{O}_\theta$ , which is the USC from 6th selection onward. Since the selection window does not have to retrace its steps, as with  $U_9$ , the UC-ME methods can find comparable or slightly better individuals to the baseline after 5 more selections, and much better individuals given enough time.

The progress of UC-ME can also be visualized through the feature map itself. Figure 57 shows how coverage changes after each user selection (or the same evaluation threshold for MAP-Elites). In addition, the figures show in red the selection window of UC-ME as it moves towards higher USC scores (in this case that of  $U_3$ ). We focus on the  $A_C$  method, as the most efficient. The top row of images in Figure 57 illustrates the differences between UC-ME and MAP-Elites exploration patterns: in gray we see the common cells discovered by both methods, in magenta we see the cells discovered only by MAP-Elites and in blue we see the cells discovered only by UC-ME. We see that



cells at higher USC values exclusively belong to UC-ME. The higher coverage of MAP-Elites is due to most cells occupying lower  $\bar{C}_s$  and  $\bar{O}_\theta$  values, which are undesirable for  $U_3$ . Figure 57 also shows how the selection window moves first towards a higher  $\bar{C}_s$ ; once it reaches the edge of the feasible space and can not find individuals with higher scores in that direction, it moves towards higher  $\bar{O}_\theta$  scores. We also see that within the first 3 selections, UC-ME with  $A_C$  has found the edges of the feasible space with the highest USC scores and starts moving around fairly haphazardly in that vicinity, leading to more selections and improved quality of individuals in that specific region of the problem space.

### 7.4.3. Relevant publications

- Konstantinos Sfikas, Antonios Liapis and Georgios N. Yannakakis: "Controllable Exploration of a Design Space via Interactive Quality Diversity," in Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2023. [552].  
Zenodo record: <https://zenodo.org/record/8054933>.

### 7.4.4. Relevance to AI4media use cases and media industry applications

Our algorithms for Interactive Quality-Diversity search contribute to Use Case 6 (AI for Human Co-Creation) in terms of a new way of interacting with an evolving computational process while taking advantage of the important concept of quality-diversity balance. While this work is not directly related to the application of music creation, the algorithms are domain-agnostic and can be integrated with generative algorithms for music provided that the quality and diversity of generated musical artifacts can be somehow evaluated.

## 7.5. Enhancing Preference Learning with Neuroevolution

**Contributing partners:** UM

### 7.5.1. Introduction and methodology

A complementary direction that focuses on merging evolution (even if not QD evolutionary search specifically) with machine learning pipelines is the work on RankNEAT. In this case, the focus is on the application domain, i.e. affective computing as the study of emotions, their manifestations and expressions, and the ways to capture (model) them computationally [553]. For such tasks, the last few years have seen a rapidly growing interest in the use of neural networks that are able to classify subjectively defined labels. This family of learning-to-rank or preference learning algorithms [554] that train neural networks—such as RankNet [555], DeepRank [556] and LambdaMART [13]—yield good performance by relying primarily on gradient descent methods. Subjectively defined labels, however, including human demonstrations (e.g. creative tasks, navigation traces and paths) or human annotations (e.g. of emotion or aesthetics) yield highly complex, deceptive and noisy loss landscapes for a neural network to learn. Assuming that the plasticity of neuroevolutionary processes would be beneficial for such loss landscapes, we test the hypothesis that evolutionary search would be a better optimizer for neural network training in preference learning (PL) tasks compared to Stochastic Gradient Descent.

To test our hypothesis, we explore the efficacy of neuroevolutionary search in PL tasks by building on the efficient and popular RankNet [555] architecture and enhancing its search capacity through neuroevolution. In particular, we introduce a novel algorithm named *RankNEAT* that relies on the Siamese neural network architecture of RankNet and learns to rank via NeuroEvolution of





Augmenting Topologies (NEAT) [557]. Unlike traditional gradient-based PL methods, RankNEAT resembles the process of plasticity [558], which induces changes in both the coupling strength and the spatial organization of synapses in biological neural networks. RankNEAT learns to rank subjectively defined labels with high degrees of accuracy through its ability to optimize the synaptic parameters such as the network’s weights and the edge architecture simultaneously. We test RankNEAT (neuroevolution) and compare it against the vanilla RankNet (stochastic gradient descent) in the task of player affect modeling across three games, using the AGAIN [559] dataset of arousal-annotated gameplay videos. Player modeling [560] is an important subfield in game research since it promotes the development of reliable human computer interaction systems and consequently improves the users’ experience.

Our current approach feeds images of gameplay to a pretrained vision transformer, while the last fully-connected layer of the network is then trained to predict ordinal values of arousal, using RankNet or RankNEAT. Results indicate that RankNEAT is superior to SGD (RankNet) in training PL models of arousal in the majority of experiments performed. Our key findings suggest that RankNEAT is a viable PL paradigm which achieves comparable or significantly higher performances to RankNet. In this first experiment, RankNEAT optimizes the edge topology of the networks’ last layer, resembling an evolutionary feature selection strategy that eliminates unnecessary features from the observed input space.

### 7.5.2. Experimental Results

This work aims to leverage neuroevolution for preference learning, assuming that its global optimization strategy may prove beneficial compared to gradient descent. Thus, the performance metric in our experiments is the accuracy in predicting the ranking between unseen pairs of gameplay footage windows. Specifically, we use a ten-fold cross-validation strategy for splitting the data into training and test sets. We follow a leave- $X$ -participants out method for cross-validation, where  $X$  is set between 6 and 11 participants depending on the game and fold. To address the randomness of weight initialization, genetic operators, and SGD, results are averaged across 5 independent runs [561] throughout this section (including the 95% confidence interval between these 5 runs). Throughout the experiments, we perform three tests per game by varying the preference threshold ( $P_t$ ) between 0.15, 0.25 and 0.50.

**Parameter Tuning:** In terms of RankNet, we tune the batch size since the benefits of the adjustment of this parameter is two-fold. On the one end, the batch size is inversely proportional to the number of updates per epoch, affecting the speed of the training process. On the other end, the ratio of learning rate to batch size is a key element influencing the SGD dynamics [562]. When it comes to RankNEAT, there is no single correct choice of parameters for all problems due to interdependencies between hyperparameters such as population size and crossover [563]. Although the compatibility threshold ( $c_t = 3$ ), elitism per species ( $e_{ps} = 2$ ), and mutation rates (0, 0.5 for nodes and edges, respectively) were tuned according to some preliminary experiments, the population size  $p$  was adjusted based on a more systematic approach since it influences both the training time and the robustness of the learner [564].

Figure 58 shows how across all three games large  $b_n$  values lead to a quick increase in accuracy for RankNet, but subsequent epochs see a drop as the process overfits to the training set. Evidently, with small  $b_n$  values testing accuracy increases more slowly but has the potential to reach higher values. Based on this finding, we will use  $b_n = 10$  as the best parameter in experiments of the following experiment. Evolution on the other hand understandably benefits from larger populations: for instance with  $p = 1000$  we see a quick optimization at the first generation but relatively small improvements after that. Since with  $p = 100$  the test accuracy reaches similar values as with  $p = 1000$  within a few generations, we choose  $p = 100$  in the experiments reported in the remainder



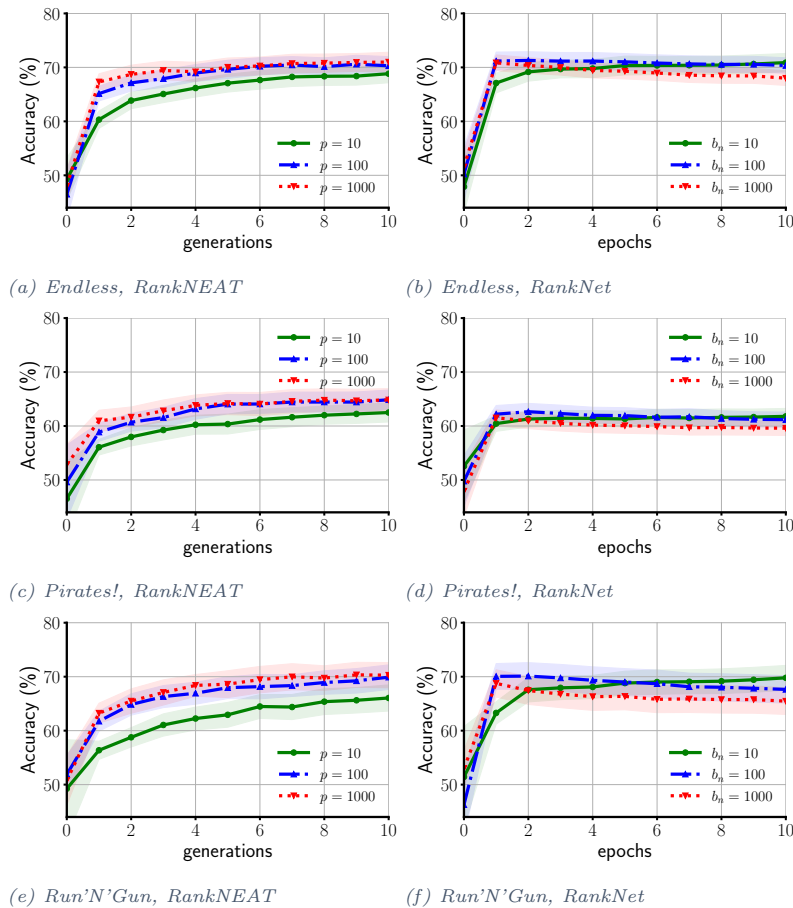


Figure 58. Impact of the population and batch size to the performance of the two algorithms.

of this work for its significantly lower computational cost.

**RankNEAT versus RankNet:** Figure 59 shows that as training progresses RankNet still is prone to overfitting, even though we chose  $b_n = 10$  because it did not overfit during the short training runs of the previous experiment. In all cases, test accuracy for RankNet drops after the first 100 iterations, often significantly (e.g. in Figure 59a). On the other hand, evolution starts performing poorly but steadily increases at later generations. While evolution assesses its individuals in terms of accuracy in the training set and consistently improves there, it is evident that the models are also able to perform well (despite some fluctuations between generations) in the test set. At the same computational effort (1,500 iterations), RankNEAT yields between 1% and 5% higher test accuracies from RankNet, on average, across the 9 experiments performed (with RankNEAT significantly outperforming RankNet in 5 of our 9 tests). Taking the best models discovered, on average, within these 1,500 iterations as a whole, we derive the results of Table 42. Here, we see that the results are comparable in several cases, although for the *Pirates!* game RankNEAT consistently performs better. It is worth noting that all models regardless of method underperform in *Pirates!* We hypothesize that RankNEAT may be able to perform better in more challenging problems.

Apart from the fact that RankNEAT performs global optimization, we expect that the custom

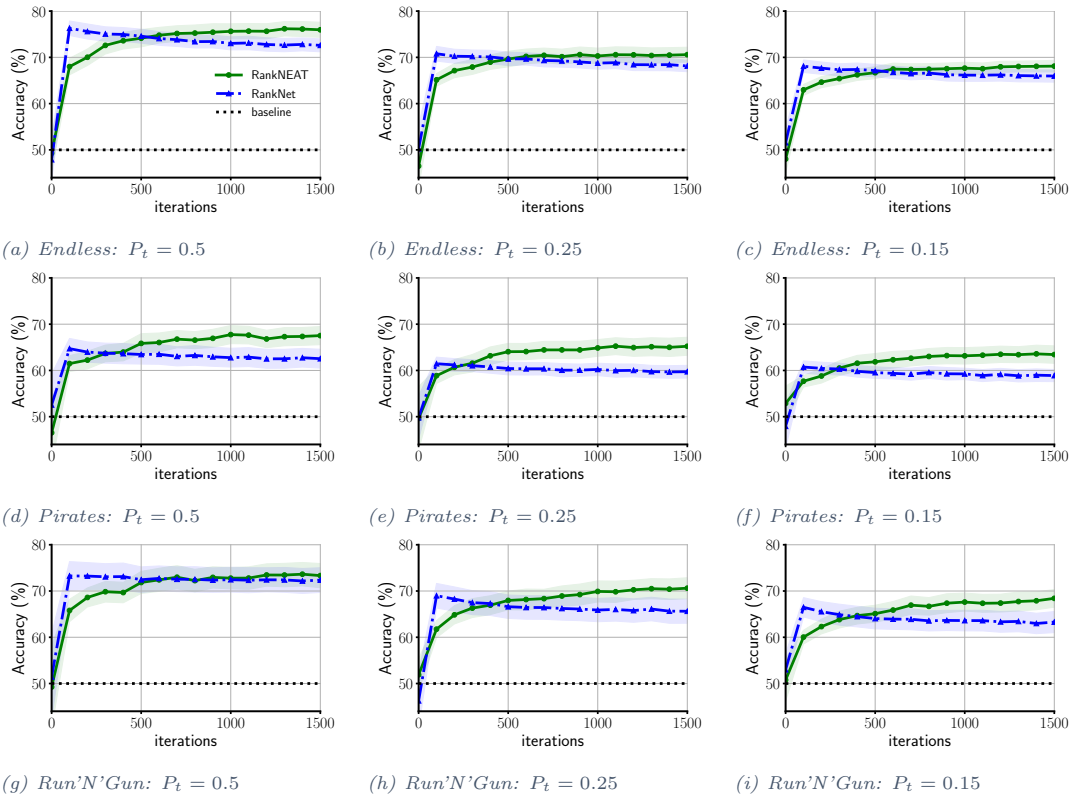


Figure 59. Accuracy (and 95% confidence intervals) over evaluations for the RankNEAT and RankNet models. The black dotted line shows the (random) baseline accuracy of 50%.

Table 42. Best accuracies (%) achieved by each model (RankNEAT vs RankNet) for the Endless, Pirates!, and Run'N'Gun test-beds, across three preference threshold values,  $P_t$ . Values are averaged across 5 independent runs. The average test accuracy of the best run (of 5) is also included within square brackets.

	$P_t = 0.5$		$P_t = 0.25$		$P_t = 0.15$	
	RankNEAT	RankNet	RankNEAT	RankNet	RankNEAT	RankNet
Endless	76.2 ±1.5 [77.3]	76.9 ±1.6 [77.9]	70.6 ±1.6 [71.7]	71.5 ±1.6 [72.1]	68.1 ±1.3 [69.2]	68.5 ±1.3 [69.1]
Pirates!	67.8 ±1.9 [69.6]	65.8 ±2.2 [66.9]	65.2 ±1.8 [66.5]	62.6 ±1.6 [63.5]	63.6 ±1.9 [64.7]	61.3 ±1.4 [62.1]
Run'N'Gun	73.6 ±2.5 [76.3]	73.7 ±3.0 [74.8]	70.6 ±2.3 [72.3]	70.2 ±2.3 [71.4]	68.4 ±1.9 [69.5]	67.8 ±2.0 [69.1]

operators that add or delete edges are especially powerful for this problem. Our version of RankNEAT does not allow for larger topologies to emerge but both speciation and topology changes in the edges are expected to have an impact. We expect that deleting an edge can act as a feature elimination mechanism and remove features that do not play a role in predicting arousal. Indeed, we observe that the best models of Table 42 for RankNEAT have between 5% and 6% fewer edges than the fully connected SGD network (RankNet with 768 edges). Due to the stochastic nature of the edge removal operator, this “feature selection” requires several generations to be impactful, but may largely be responsible for the good performance of the models.

**Qualitative findings:** Results presented in the previous section show that player arousal can be modeled based on general-purpose representations such as video frames and, consequently, pixels. Drawing inspiration from the study of Makantasis et al. [565], we constructed the Class Activation



Maps (CAM) in order gain insights on which regions of the frames contributed the most to the final result. We observe that important predictors of arousal across games are regions containing information about the player, such as the avatar's position, life, game time, and score. Furthermore, the regions that contain information about the enemies' avatars are also very important for the model. In two out of three games, the model manages to mask out some of the redundant information in the environment, such as empty space in Endless or the sky background in Run'N'Gun. For Pirates!, however, such patterns are less clear, and the model precludes the powerups from high importance regions. This may explain the relatively low accuracy value achieved on this game.

### 7.5.3. Relevant publications

- Kosmas Pinitas, Konstantinos Makantasis, Antonios Liapis and Georgios N. Yannakakis: "RankNEAT: Outperforming Stochastic Gradient Search in Preference Learning Tasks," in Proceedings of the Genetic and Evolutionary Computation Conference, 2022. [566].  
Zenodo record: <https://zenodo.org/record/7879220>.

### 7.5.4. Relevance to AI4media use cases and media industry applications

Our RankNEAT algorithm can be applied widely within any affective computing and affect modeling application; in the context of the AI4Media use cases, it contributes to Use Case 5 (AI for Games) by providing a tool for modelling the subjective human experiences of players to dynamically adapt the game itself according to the user's (predicted) engagement or arousal levels.



## 8. Learning to count (Task 3.7) – detailed description

**Contributing partners:** CNR

“Learning to Count” is a task having to do with supervised learning approaches for training estimators of quantities. There are two classes of problems that are being addressed in this task, and that may be usefully viewed as forming two different subtasks, i.e.,

- “Learning to quantify” (LQ – a.k.a. *quantification*). This subtask is concerned with training unbiased estimators of class prevalence via supervised learning, i.e., learning to estimate, given a sample of objects, the percentage of objects that belong to a given class. This task originates with the observation that “CC”, the trivial method of obtaining class prevalence estimates, is often a biased estimator, and thus delivers suboptimal quantification accuracy. This bias is particularly strong when the data exhibits *dataset shift*, i.e., when the joint distribution of the dependent and the independent variables is not the same in the training data and in the unlabelled data for which predictions must be issued. Quantification is important for several applications, e.g., gauging the collective satisfaction for a certain product from textual comments, establishing the popularity of a given political candidate from blog posts, predicting the amount of consensus for a given governmental policy from tweets, or predicting the amount of readers who will find a product review helpful.
- “Learning to count objects”. This subtask has to do with using machine learning approaches in order to train estimators of the number of objects (which may be inanimate objects, such as cars, but may also be animate objects, such as people or animals) in visual media, such as still images or video frames. Example applications of these techniques are, e.g., counting the number of cars in a video frame (in order to estimate traffic volume or car park occupancy), or counting the number of people in a still image (say, in order to estimate the amount of people taking part in a rally).

### 8.1. QuaPy: A Python-Based Framework for Learning to Quantify

**Contributing partners:** CNR

#### 8.1.1. Introduction and methodology

We here present QuaPy, a framework written in Python that provides implementations of the most important tools for research, development, and experimentation, in LQ. Some of the authors who have published papers on the field of quantification have also made available software packages implementing their methods and baselines. However, such software repositories are often tied to specific applicative domains, are limited to reproducing experimental results from specific papers, or lack proper documentation and wiki references. While all these implementations represent valuable resources that demonstrate how to implement and use specific algorithms, to the best of our knowledge none among the existing software packages strive to define a proper framework that jointly caters for all steps of the quantification pipeline, from data preparation to the visualization of results, in a unified way. QuaPy is a flexible and extensible framework that aims at filling this gap.

A quantifier is defined in QuaPy as a model that can be `fit` on some training data, so that the fitted model can estimate class prevalence values for unlabelled data.





Quantification methods can be classified as belonging to the *aggregative*, *non-aggregative*, or *meta* classes. Aggregative methods are characterized by the fact that quantification is obtained as an aggregation of the outputs returned by a classification process for the individual documents. Non-aggregative methods analyse instead the sample of unlabelled documents as a whole, without resorting to the classification of individual data items. Finally, meta-quantifiers are built on top of other quantifiers, and generate their predictions by analysing the predictions made by the underlying quantifiers. QuaPy provides implementations of aggregative methods (such as Classify and Count, Adjusted Classify and Count, Probabilistic Classify and Count, Probabilistic Adjusted Classify and Count, and Forman’s variants of ACC, including X, MAX, T50, and Median Sweep). Other important aggregative methods being provided are the Saerens-Latinne-Decaestecker method (SLD), HDy, and other methods based on Explicit Loss Minimization, such as SVM(KLD), SVM(NKLD), SVM(Q), SVM(AE), and SVM(RAE). Currently, QuaPy does not provide implementations of non-aggregative quantifiers, but provides implementations of quantifier ensembles, including the well-known ones called **Averaging**, **Training Prevalence**, **Distribution Similarity**, and **Performance**. QuaPy also provides an implementation of a deep-learning-based method, i.e., QuaNet.

QuaPy allows a set of binary quantifiers, one for each class, to be assembled into a single-label multi-class quantifier, by adopting a “one-vs-all” strategy. This takes the form of computing prevalence estimates independently for each class (i.e., via binary quantification) via independently trained binary quantifiers, and then normalizing the resulting vector of prevalence values (via L1-normalization) so that these values sum up to one.

QuaPy makes available a number of datasets that have been used for experimentation purposes in the quantification literature, and specifically:

- **Reviews:** a collection of 3 datasets of customer reviews. All reviews are classified according to (binary) sentiment polarity.
- **Twitter Sentiment:** 11 datasets of tweets labelled by sentiment. Similarly to the Reviews datasets, these are high-dimensional datasets. These datasets use three sentiment labels (Positive, Neutral, Negative), and are thus useful for testing non-binary quantification methods.
- **UCI:** 33 binary datasets from the UCI Machine Learning repository. Differently from the previous datasets, these non-textual datasets are low-dimensional (with dimensionalities ranging from 3 to 256), thus providing diversity, in terms of the type of data, with respect to the previous two sets of datasets.

Several error measures have been proposed in the literature, and QuaPy implements a rich set of them, such as absolute error, relative absolute error, squared error, KL Divergence, and normalized KL Divergence. Functions which return the average values of the same measures across different samples are also available.

An environment for experimenting with quantification must not only be endowed with several evaluation measures, but it also must allow the experimentation to be carried out according to different evaluation protocols. QuaPy implements both the Natural Prevalence Protocol (NPP) and the Artificial Prevalence Protocol (APP). In the NPP, the test set is sampled randomly, so that most samples exhibit class prevalence values not too different from those of the test set. In the APP, the test set is instead sampled in a controlled way, in order to generate samples characterized by different, pre-specified prevalence values, so as to cover, with uniform probability, the full spectrum of class prevalence values. In the APP, the user specifies the number of equidistant points to be generated from the interval [0,1]. For example, if `n_prevs=11` then, for each class, the prevalence values [0.0, 0.1, ..., 0.9, 1.0] will be used. This means that, for two classes, the number of different





sampled prevalence values will be 11 (since, once the prevalence of one class is determined, the other one is also).

Quantification has long been regarded as a by-product of classification, which means that the model selection (i.e., hyperparameter optimization) strategies customarily adopted in quantification have simply been borrowed from classification. It has been argued that specific model selection strategies should be adopted for quantification. That is, model selection strategies for quantification should minimize quantification-oriented loss measures, and be carried out in a variety of scenarios exhibiting different degrees of distribution shift.

QuaPy supports quantification-oriented model selection by implementing a grid-search exploration over the space of hyperparameter combinations that evaluates each such combination by means of a given quantification-oriented error metric, and according to either the APP (the default value) or the NPP.

QuaPy implements some plotting functions that can be useful in displaying the performance of the tested quantification methods:

- **Diagonal plot:** The diagonal plot shows a very insightful view of the quantifier's performance, i.e., it plots the predicted class prevalence (on the y-axis) against the true class prevalence (on the x-axis), averaging across all samples characterized by the same true prevalence. Unfortunately, this visualization device is inherently limited to binary quantification (one can simply generate as many diagonal plots as there are classes, though, by indicating which class should be considered the target of the plot).
- **Error-by-Shift plot:** This plot displays the quantification error made by a quantifier as a function of the distribution shift between the training set and the test sample, averaging across all samples characterized by the same amount of distribution shift. Both quantification error and distribution shift can be measured in terms of any measure among those implemented in QuaPy, and can be computed and plotted both in the binary case and in the non-binary case.
- **Bias-Box plot:** This plot aims at displaying, by means of box plots, the bias that any quantifier exhibits with respect to the training class prevalence values. The bias can be broken down into different bins, e.g., distinguishing the bias in cases of low, medium, and high prevalence shift.

In conclusion, the goal of QuaPy, a Python-based package that makes available a rich set of quantification methods, tools, experimental protocols, and datasets, is that of supporting an efficient and scientifically correct experimentation of quantification methods. We think that QuaPy will be of help to machine learning researchers that work on developing new quantification algorithms, as it provides them with many baselines to compare against, datasets to test their methods on, and tools that implement all the typical steps of quantification-based experimentation, from data preparation to the visualization of results. We think that QuaPy will be of help also to researchers and practitioners in other disciplines who simply need to apply quantification in their own work, as it provides them with a streamlined workflow, a wide choice of different approaches, and quick access to the package thanks to the support of installation based on `pip`. QuaPy is an open-source project, licensed under the BSD-3 licence; its repository will be updated following the advances in quantification research, and it is open to contributions of new methods, tools, and datasets.

For more details on this work please check the full paper [567].

### 8.1.2. Relevant publications

- Alejandro Moreo, Andrea Esuli, and Fabrizio Sebastiani. QuaPy: A Python-based framework for quantification. **Proceedings of the 30th ACM International Conference on**





**Knowledge Management (CIKM 2021)**, Gold Coast, AU, pp. 4534–4543. [567].  
The paper appears on Zenodo at <https://zenodo.org/record/5560941>.

### 8.1.3. Relevant software/datasets/other outcomes

- The QuaPy framework for LQ can be found at <https://github.com/HLT-ISTI/QuaPy>.

### 8.1.4. Relevance to AI4media use cases and media industry applications

Learning to quantify is important for the media industry, since it allows to monitor temporal trends of indicators relevant to journalism, such as public opinion on specific topics (see Section 8.3.5) and the frequency of journalistic news belonging to specific classes (see Section 8.4.5). The software library described in this section makes a strong contribution to the research agenda of learning to quantify and to the applicability of the related techniques in the media sector (among others), by making state-of-the-art LQ software publicly available to researchers and practitioners alike.

## 8.2. Ordinal Quantification through Regularization

**Contributing partners:** CNR

### 8.2.1. Introduction and methodology

The vast majority of the quantification methods proposed so far deal with the quantification task in which  $\mathcal{Y}$  is a plain, unordered set; this essentially means the standard binary ( $n = 2$ ) or multiclass ( $n > 2$ ) quantification tasks. Very few methods, instead, deal with OQ, the task of performing quantification on a set of  $n > 2$  classes on which a total order “ $\prec$ ” is defined. Ordinal quantification is important, though, because ordinal scales arise in many applications, especially ones involving human judgments. For instance, in a customer satisfaction endeavour one may want to estimate how a set of reviews of a certain product is distributed across the set of classes  $\mathcal{Y} = \{1\text{Star}, 2\text{Stars}, 3\text{Stars}, 4\text{Stars}, 5\text{Stars}\}$ , while a social scientist might want to find out how inhabitants of a certain region are distributed in terms of their happiness with health services in the area ( $\mathcal{Y} = \{\text{VeryUnhappy}, \text{Unhappy}, \text{Happy}, \text{VeryHappy}\}$ ).

In this work, we contribute to the field of OQ in a number of ways.

First, we develop and make publicly available two datasets for evaluating OQ algorithms, one consisting of textual product reviews and one consisting of telescope observations. Both datasets are from scenarios in which OQ arises naturally, and are generated according to a strong, well-tested protocol for the generation of datasets oriented to the evaluation of quantifiers. This contribution fills a gap, because datasets previously used for the evaluation of OQ were not adequate.

Second, we perform an extensive experimental comparison (using the two previously mentioned datasets) among the most important OQ algorithms that have been proposed in the literature; this is important, since some of them had been compared with each other on a testbed that was likely inadequate, while some other algorithms had been developed independently (i.e., in the unawareness) of the previous ones, and had thus never been compared with them.

Third, we propose new OQ algorithms, which introduce regularization into existing quantification methods. We experimentally compare our proposals with the existing state of the art and make the corresponding code publicly available.

We use the following notation. By  $\mathbf{x} \in \mathcal{X}$  we indicate a data item drawn from a domain  $\mathcal{X}$ , and by  $y \in \mathcal{Y}$  we indicate a class drawn from a set of classes  $\mathcal{Y} = \{y_1, \dots, y_n\}$ , also known as a *codeframe*, on which a total order “ $\prec$ ” is defined. The symbol  $\sigma$  denotes a *sample*, i.e., a non-empty set of





unlabelled data items in  $\mathcal{X}$ , while  $L \subset \mathcal{X} \times \mathcal{Y}$  denotes a set of labelled data items  $(\mathbf{x}, y)$  which we will use for training our quantifiers. By  $p_\sigma(y)$  we indicate the true prevalence of class  $y$  in sample  $\sigma$ , while by  $\hat{p}_\sigma^M(y)$  we indicate an estimate of this prevalence as obtained by a quantification method  $M$  that receives  $\sigma$  as an input, where  $0 \leq p_\sigma(y), \hat{p}_\sigma^M(y) \leq 1$  and  $\sum_{y \in \mathcal{Y}} p_\sigma(y) = \sum_{y \in \mathcal{Y}} \hat{p}_\sigma^M(y) = 1$ .

We use as baselines for our methods some important multiclass quantification methods which do not take ordinality into account. These methods provide the foundation for their ordinal extensions which we propose in this work.

These multiclass methods are CC, Probabilistic Classify and Count (PCC), Adjusted Classify and Count (ACC) [568], Probabilistic Adjusted Classify and Count (PACC) [569], and the Saerens-Latinne-Decaestecker method (SLD) method [570].

We use as additional baselines some important methods proposed within experimental physics, among which quantification methods

Similar to the adjustment of ACC, experimental physicists have proposed adjustments that solve for  $\mathbf{p}$  the system of linear equations that ACC and PACC solve. However, these “unfolding” quantifiers differ from ACC in two regards.

The first aspect is that the hard classifier  $h$  of that ACC and PACC use is often (although not always) replaced by a partition  $c : \mathcal{X} \rightarrow \{1, \dots, d\}$  of the feature space, so that

$$\begin{aligned} [\mathbf{q}]_i &= \frac{1}{|\sigma|} \cdot |\{\mathbf{x} \in \sigma : c(\mathbf{x}) = i\}| \\ \mathbf{M}_{ij} &= \frac{|\{(\mathbf{x}, y) \in V : c(\mathbf{x}) = i, y = y_j\}|}{|\{(\mathbf{x}, y) \in V : y = y_j\}|} \end{aligned} \quad (17)$$

and  $\mathbf{M} \in \mathbb{R}^{d \times n}$ . Another possible choice for  $c$  is to partition the feature space by means of a decision tree; in this case (i) it typically holds that  $d > n$ , (ii) and  $c(\mathbf{x})$  represents the index of a leaf node.

The second aspect is that “unfolding” quantifiers regularize their estimates in order to promote solutions that are the most *plausible* solutions in OQ. Specifically, these methods employ the assumption that neighbouring classes have similar prevalence values; depending on the algorithm, this assumption is encoded in different ways. This assumption is quite reasonable, since the “smoothness” of the histogram that represents the distribution is arguably *the only aspect that differentiates an ordinal distribution from a non-ordinal multiclass distribution*.

The Regularized Unfolding (RUN) method has been used by physicists for decades. It estimates the vector  $\mathbf{p}$  of class prevalence values by minimizing a loss function  $\mathcal{L} : \mathbb{R}^n \rightarrow \mathbb{R}$  over the estimate  $\hat{\mathbf{p}}$ ;  $\mathcal{L}$  consists of two terms, i.e., a negative log-likelihood term to model the error of  $\hat{\mathbf{p}}$ , and a regularization term to model the plausibility of  $\hat{\mathbf{p}}$ .

The second term of  $\mathcal{L}$  is a Tikhonov regularization term  $\frac{1}{2} (\mathbf{C}\mathbf{p})^2$ . This term introduces an inductive bias towards solutions which are plausible with respect to ordinality. The Tikhonov matrix  $\mathbf{C}$  is chosen in such a way that term  $\frac{1}{2} (\mathbf{C}\mathbf{p})^2$  measures the smoothness of the histogram that represents the distribution, i.e.,

$$\frac{1}{2} (\mathbf{C}\mathbf{p})^2 = \frac{1}{2} \sum_{i=2}^{n-1} (-[\mathbf{p}]_{i-1} + 2[\mathbf{p}]_i - [\mathbf{p}]_{i+1})^2 \quad (18)$$

Combining the likelihood term and the regularization term, the loss function of RUN is given by

$$\mathcal{L}(\hat{\mathbf{p}}; \mathbf{M}, \mathbf{q}, \tau, \mathbf{C}) = \sum_{i=1}^d (\mathbf{M}_i^\top \hat{\mathbf{p}} - [\mathbf{q}]_i \cdot \ln(\mathbf{M}_i^\top \hat{\mathbf{p}})) + \frac{\tau}{2} (\mathbf{C}\hat{\mathbf{p}})^2 \quad (19)$$

and an estimate  $\hat{\mathbf{p}}$  is chosen by minimizing  $\mathcal{L}$  numerically over  $\hat{\mathbf{p}}$ . Here,  $\tau \geq 0$  is a hyperparameter which controls the impact of the regularization.





The Iterative Bayesian Unfolding (IBU) method revolves around an expectation maximization approach with Bayes’ theorem, and thus has a common foundation with the SLD method. The E-step and the M-step of IBU can be written as the single, combined update rule

$$\hat{p}_{\sigma}^{(k)}(y_i) = \sum_{j=1}^d \frac{\mathbf{M}_{ij} \cdot \hat{p}_{\sigma}^{(k-1)}(y_i)}{\sum_{l=1}^n \mathbf{M}_{lj} \cdot \hat{p}_{\sigma}^{(k-1)}(y_l)} [\mathbf{q}]_i \quad (20)$$

In this work we develop algorithms which extend ACC, PACC, and SLD with the regularizers from RUN and IBU. Through this extension, we obtain o-ACC, o-PACC, and o-SLD, the OQ counterparts of these well-known non-ordinal quantification algorithms. In doing this, since we employ the regularizers but not any other aspect of RUN and IBU, we preserve the general characteristics of ACC, PACC, and SLD. In particular, our methods continue to work with classifier predictions, i.e., we do not employ the categorical feature representation from Equation 17, which RUN and IBU employ, and we do not use the Poisson assumption of RUN. Therefore, our extensions are “minimal”, in the sense that they directly address ordinality without introducing any undesired side effects in the original methods.

**o-ACC** and **o-PACC**, our ordinal extensions to ACC and PACC build on the finding reported in [571, Theorem 4.1], which states that the solution of the equation on which ACC and PACC are based corresponds to a minimum-norm least-squares solution. Namely, among all least-squares solutions  $\hat{\mathbf{p}}^{\text{LSq}} = \operatorname{argmin}_{\mathbf{p}} \|\mathbf{q} - \mathbf{M}\mathbf{p}\|_2^2$ , which by themselves do not need to be unique, the solution to that equation is the one that also minimizes the quadratic norm  $\|\mathbf{p}\|_2^2$ . The resulting equation is thus conceptually similar, although not necessarily equal, to a regularized estimate which employs the quadratic norm for regularization. In particular, both equations simultaneously minimize a least-squares objective and the norm of their candidate solutions. Note that the regularization function herein is, unlike the regularization from RUN, unrelated to the ordinal nature of the classes.

To obtain the true OQ methods o-ACC and o-PACC, we replace the minimum-norm regularization with the regularization term of RUN (see Equation 18). Through this replacement, we minimize the same objective function as ACC and PACC, i.e., a least-squares objective, but regularize towards solutions that we deem more plausible for OQ.

**o-SLD**, our ordinal variant o-SLD leverages the ordinal regularization of IBU in SLD. Namely, our method does not use the latest estimate directly as the prior of the next iteration, but a smoothed version of this estimate. To this end, we fit a low-order polynomial to each intermediate estimate  $\hat{\mathbf{p}}^{(k)}$  and use a linear interpolation between this polynomial and  $\hat{\mathbf{p}}^{(k)}$  as the prior of the next iteration. Like in IBU, we consider the interpolation factor as a hyperparameter through which the strength of this regularization is controlled.

### 8.2.2. Experimental results

We conduct our experiments on two large datasets that we have generated for the purpose of this work, and that we make available to the scientific community. The first dataset, named AMAZON-OQ-BK, consists of product reviews labelled according to customer’s judgments of quality, i.e., 1Star to 5Stars. The second dataset, FACT-OQ, consists of telescope observations labelled by one of 12 totally ordered classes. Hence, these data sets originate in practically relevant and diverse applications of OQ.

In our main experiment, we compare our proposed methods o-ACC, o-PACC, and o-SLD, with several baselines, i.e., (i) the existing OQ methods OQT [572] and ARC [573]; (ii) the “unfolding” OQ methods IBU and RUN; (iii) the non-ordinal methods CC, PCC, ACC, PACC, SLD. We



Table 43. Average performance in terms of NMD (lower is better). For each data set (AMAZON-OQ-BK and FACT-OQ), we present the results of the two protocols APP and APP-OQ. The best performance in each column is highlighted in **boldface**. According to a Wilcoxon signed rank test with  $p = 0.01$ , all other methods are significantly different from the best method.

method	AMAZON-OQ-BK		FACT-OQ	
	APP	APP-OQ	APP	APP-OQ
CC	.0526 ± .019	.0344 ± .013	.0534 ± .012	.0494 ± .011
PCC	.0629 ± .022	.0440 ± .017	.0651 ± .017	.0621 ± .017
ACC	.0229 ± .009	.0193 ± .007	.0582 ± .028	.0575 ± .028
PACC	.0209 ± .008	.0176 ± .007	.0791 ± .048	.0816 ± .049
SLD	<b>.0172 ± .007</b>	.0154 ± .006	.0373 ± .010	.0355 ± .009
OQT	.0775 ± .026	.0587 ± .027	.0746 ± .019	.0731 ± .020
ARC	.0641 ± .023	.0477 ± .015	.0566 ± .014	.0568 ± .016
IBU	.0253 ± .010	.0197 ± .007	<b>.0213 ± .005</b>	.0187 ± .004
RUN	.0252 ± .010	.0198 ± .007	.0222 ± .006	.0194 ± .005
o-ACC	.0229 ± .009	.0188 ± .007	.0274 ± .007	.0230 ± .006
o-PACC	.0209 ± .008	.0174 ± .007	.0230 ± .006	<b>.0178 ± .004</b>
o-SLD	.0173 ± .007	<b>.0152 ± .006</b>	.0327 ± .008	.0289 ± .007

compare these methods on the AMAZON-OQ-BK and FACT-OQ datasets, and under the APP and APP-OQ protocols.

Each method is allowed to tune the hyperparameters of its embedded classifier using the samples of the validation set. We use logistic regression on the AMAZON-OQ-BK dataset and probability-calibrated decision trees on the FACT-OQ dataset; this choice of classifiers is motivated by common practice in the fields where these data sets originate, and from our own experience that these classifiers work well on the respective type of data. After the hyperparameters of the classifier are optimized, we apply each method to the samples of the test set.

The results of this experiment are summarized in Table 43. These results show that our proposed methods outperform the competition on both data sets if the ordinal APP-OQ protocol is employed. More specifically, o-SLD is the best method on AMAZON-OQ-BK while o-PACC is the best method on FACT-OQ. Moreover, o-SLD is consistently better or equal to SLD, o-ACC is consistently better or equal to ACC, and o-PACC is consistently better or equal to PACC, also in the standard APP protocol in which smoothness is not imposed.

For more details on this work and additional experiments please check the full paper [574].

### 8.2.3. Relevant publications

- Mirko Bunse, Alejandro Moreo, Fabrizio Sebastiani, Martin Senz. Ordinal quantification through regularization. **Proceedings of the 33rd European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2022)**, Grenoble, FR, Volume V, pp. 36–52. [574]  
The paper appears on Zenodo at <https://zenodo.org/record/7090067>.
- Mirko Bunse, Alejandro Moreo, Fabrizio Sebastiani, Martin Senz. Ordinal quantification through regularization. Presented at the **LWDA Workshop on Knowledge Discovery, Data Mining and Machine Learning (LWDA 2022)**, Hildesheim, DE. (Oral presentation only, no paper was produced.) [575]





#### 8.2.4. Relevant software/datasets/other outcomes

- The code and the datasets for reproducing the results reported in [574] are available at <https://github.com/mirkobunse/ecml22> (code), <https://zenodo.org/record/7090095>, and <https://zenodo.org/record/7081208> (datasets).

#### 8.2.5. Relevance to AI4media use cases and media industry applications

Ordinal quantification, and the work that CNR has carried out on it, has important applicative potential in the media industry, especially in the field of monitoring public opinion and its trends. Indeed, since opinion is often expressed on an ordinal scale (as in, e.g., product reviews, which are often evaluated on an ordinal five-point scale), monitoring such trends requires the ordinal nature of the scale to be taken into account in the quantification algorithm, so as to deliver increased prediction accuracy.

### 8.3. Tweet Sentiment Quantification: An Experimental Re-Evaluation

**Contributing partners:** CNR

#### 8.3.1. Introduction and methodology

In a 2016 paper, Gao and Sebastiani [14] (hereafter: [GS2016]) have argued that, when the objects of analysis are tweets, the *vast majority* of sentiment classification efforts actually have quantification as their final goal, since hardly anyone who engages in sentiment classification of tweets is interested in the sentiment conveyed by a specific tweet. We call the resulting task *tweet sentiment quantification*. [GS2016] presented an experimental comparison of 8 important quantification methods on 11 Twitter datasets annotated by sentiment, with the goal of assessing the strengths and weaknesses of the various methods for tweet sentiment quantification. That paper became then influential and a standard reference on this problem, and describes what is currently the largest comparative experimentation on tweet sentiment quantification.

In this work, we argue that the experimental results obtained in [GS2016] are unreliable, as a result of the fact that the experimental protocol used in that paper was weak. We thus present new experiments in which we re-test all 8 quantification methods originally tested in [GS2016] (plus some additional ones that have been proposed since then) on the same 11 datasets used in [GS2016], this time using a now consolidated and much more robust experimental protocol. These new experiments (whose number is 5,775 times larger than the number of experiments conducted in [GS2016], even without counting the experiments on new quantification methods that had not been considered in [GS2016]) return results dramatically different from those obtained in [GS2016], and thus give us a new, more reliable picture of the relative merits of the various methods on the tweet sentiment quantification task.

There are two main experimental protocols that have been used in the literature for evaluating quantification; we will here call them the Artificial-prevalence Protocol (APP) and the Natural-prevalence Protocol (NPP).

The APP consists of taking a standard dataset<sup>8</sup>, split into a training set  $\mathcal{L}$  of labelled items and a set  $\mathcal{U}$  of unlabelled items, and conducting repeated experiments in which either the training set prevalence values or the test set prevalence values of the classes are artificially varied by means of

<sup>8</sup>By “a standard dataset” we here mean any dataset that has originally been assembled for testing classification systems; any such dataset can be used for testing quantification systems too.





subsampling (i.e., by removing random elements of specific classes until the desired class prevalence values are obtained). In other words, subsampling is used either to generate  $s$  training samples  $L_1 \subseteq \mathcal{L}, \dots, L_s \subseteq \mathcal{L}$ , or to generate  $t$  test samples  $U_1 \subseteq \mathcal{U}, \dots, U_t \subseteq \mathcal{U}$ , or both, where the class prevalence values of the generated samples are predetermined and set in such a way as to generate a wide array of distribution drift values. This is meant to test the robustness of a *quantifier* (i.e., of an estimator of class prevalence values) in scenarios characterized by class prevalence values very different from the ones the quantifier has been trained on. For instance, in the binary quantification experiments carried out in [568], given codeframe  $\mathcal{Y} = \{y_1, y_2\}$ , repeated experiments are conducted in which examples of either  $y_1$  or  $y_2$  are removed at random from the test set in order to generate predetermined prevalence values for  $y_1$  and  $y_2$  in the samples  $U_1, \dots, U_t$  thus obtained. In this way, the different samples are characterised by a different prevalence of  $y_1$  (e.g.,  $p_U(y_1) \in \{0.00, 0.05, \dots, 0.95, 1.00\}$ ) and, as a result, by a different prevalence of  $y_2$ . This can be repeated, thus generating multiple random samples for each chosen pair of class prevalence values. Analogously, random removal of examples of either  $y_1$  or  $y_2$  can be performed on the training set, thus bringing about training samples with different values of  $p_L(y_1)$  and  $p_L(y_2)$ .

This protocol had been criticised because it may generate samples exhibiting class prevalence values very different from the ones of the set from which the sample was extracted, i.e., class prevalence values that might be hardly plausible in practice. As a result, one may resort to the NPP, which consists instead of conducting experiments on “real” datasets only, i.e., datasets consisting of a training set  $L$  and a test set  $U$  that have been sampled IID from the data distribution. In other words, no extraction of samples from the dataset is performed by perturbing the original class prevalence values; instead, a single train-and-test run is performed, using the original training set  $\mathcal{L}$  as the training sample  $L$  and the original test set  $\mathcal{U}$  as the test sample  $U$ .

The experimentation conducted by [GS2016] on tweet sentiment quantification is indeed an example of the NPP, since it relies on 11 “original” datasets of tweets annotated by sentiment, i.e., no extraction of samples at prespecified values of class prevalence was performed. However, while in classification an experiment involving 11 different datasets probably counts as large and robust, this does not hold in quantification if only one test per dataset is conducted. The reason is that, since the objects of quantification are *sets* of documents in the same way that the objects of classification are individual documents, *testing a quantifier on just 11 sets of documents should be considered, from an experimental point of view, a drastically insufficient experimentation, akin to testing a classifier on 11 documents only.*

Unfortunately, finding a large enough set (say, 1,000 or more) of datasets sampled IID from the respective data distributions is nearly impossible; this indicates that extracting a large enough number of samples from the same dataset is probably the only way to go for evaluating quantification.<sup>9</sup> Indeed, most recent quantification works (e.g., [577–585]) adopt the APP, and not the NPP.

As a result, we should conclude that the experimentation conducted in [GS2016] is weak, and that the results of that experimentation are thus unreliable. We thus re-evaluate the same quantification methods that [GS2016] tested (plus some other more recent ones) on the same datasets, this time following the by now consolidated and much more robust APP; in our case, this turns out to involve 5,775 as many experiments as run in the original study, even without considering the experiments on quantification methods that had not been considered in [GS2016]).

<sup>9</sup>An example set of experiments that use the NPP on a large enough set of test sets is the one reported in [576], where the authors test quantifiers on  $52 \times 99 = 5,148$  binary test sets. This results from the fact that, in using the RCV1-v2 test collection, they consider the 99 RCV1-v2 classes and bin the RCV1-v2 791,607 test documents in 52 bins (each corresponding to a week’s worth of data, since the RCV1-v2 data span one year) of 15,212 documents each on average. However, it is not always easy to find test collections with such a large amount of classes and annotated data, and this limits the applicability of the NPP. It should also be mentioned that, as Card and Smith [577] noted, the vast majority of the 5,148 RCV1-v2 binary test sets used in [576] exhibit very little distribution shift, which makes the testbed used in [576] unchallenging for quantification methods.





### 8.3.2. Experimental results

We have carried out experiments in order to re-assess the merits of different quantification methods under the lens of the APP. We have conducted all these experiments using QuaPy<sup>10</sup>, a software framework for quantification written in Python that we have developed and made available through GitHub. QuaPy was presented in Section 8.1.<sup>11</sup> As the measures of quantification error we use *Absolute Error* (AE) and *Relative Absolute Error* (RAE).

The quantification methods used in [GS2016], that we also use in this paper, are *Classify and Count* (CC), *Adjusted Classify and Count* (ACC), *Probabilistic Classify and Count* (PCC), *Probabilistic Adjusted Classify and Count* (PACC), the *Saerens-Latinne-Decaestecker method* (SLD), SVM(KLD), SVM(NKLD), and structured output methods such as SVM(Q). We also consider other structured output methods such as SVM(AE) and SVM(RAE), and ensemble methods such as  $\mathbf{E(PACC)}_{\text{ptr}}$  and  $\mathbf{E(PACC)}_{\text{AE}}$ . We also report results for HDy and QuaNet.

The datasets on which we run our experiments are the same 11 datasets on which the experiments of [GS2016] were carried out. [GS2016] makes these datasets available already in vector form; we refer to [GS2016] for a fuller description of these datasets.

In [586], detailed experimental results are reported, including results of a paired sample, two-tailed t-test that we have run, at different confidence levels, in order to check if other methods are different or not, in a statistically significant sense, from the best-performing one.

An important aspect that emerges from the results is that the behaviour of the different quantifiers is fairly consistent across our 11 datasets; in other words, when a method is a good performer on one dataset, it tends to be a good performer *on all datasets*. Together with the fact that we test on a large set of samples, and that these are characterised by values of distribution shift across the entire range of all possible such shifts, this allows us to be fairly confident in the conclusions that we draw from these results.

A second observation is that three methods (ACC, PACC, and SLD) stand out, since they perform consistently well across all datasets and for both evaluation measures. In particular, SLD is the best method for 7 out of 11 datasets when testing with AE, and for all 11 datasets when testing with RAE. PACC also performs very well, and is the best performer for 3 out of 11 datasets when testing with AE. The fact that both ACC and PACC tend to perform well shows that the intuition according to which CC predictions should be “adjusted” by estimating the disposition of the classifier to assign class  $y_i$  when class  $y_j$  is the true label, is valuable and robust to varying levels of distribution shift. The same goes for SLD, although SLD “adjusts” the CC predictions differently, i.e., by enforcing the mutual consistency between the posterior probabilities and the class prevalence estimates.

By contrast, these results show a generally disappointing performance on the part of all methods based on structured output learning, i.e., on the SVM<sub>perf</sub> learner. Note that the fact that SVM(KLD), SVM(NKLD), SVM(Q) optimise a performance measure different from the one used in the evaluation (AE or RAE) cannot be the cause of this suboptimal performance, since this latter also characterises SVM(AE) when tested with AE as the evaluation measure, and SVM(RAE) when tested with RAE.

In conclusion, the results of our experiments show that a re-evaluation of the relative merits of different quantification methods on the tweet sentiment quantification task was necessary. We have shown that the experimentation previously conducted in [GS2016] was weak, since the experimental protocol that was followed led the authors of this study to conduct their evaluation on a radically insufficient amount of test data points. We have then conducted a re-evaluation of the same methods on the same datasets according to a more robust, and now widely accepted, experimental protocol,

<sup>10</sup><https://github.com/HLT-ISTI/QuaPy>

<sup>11</sup>Please see branch `tweetSent`





which has led to an experimentation on a number of datapoints 5,775 times larger than the one of [GS2016]. In addition to these experiments, we have also tested some further methods, some of which had appeared after [GS2016] was published.

This experimentation has proven necessary for at least two reasons. The first reason is that some evaluation functions (such as KLD and NKLD) that had been used in [GS2016] are now known to be unsatisfactory, and their use should thus be deprecated in favour of functions such as AE and RAE. The second reason, and probably the most important one, is that the results of our re-evaluation have radically disconfirmed the conclusions originally drawn by the authors of [GS2016], showing that the methods (e.g., PCC) that had emerged as the best performers in [GS2016] tend to behave well only in situations characterised by very low distribution shift; on the contrary, when distribution shift increases, other methods (such as SLD) are to be preferred. In particular, our experiments do justice to the SLD method, which had obtained fairly bland results in the experiments of [GS2016], and which now emerges as the true leader of the pack, thanks to consistently good performance across the entire spectrum of distribution shift values.

For more details on this work and additional experiments please check the full paper [586].

### 8.3.3. Relevant publications

- Alejandro Moreo and Fabrizio Sebastiani. Tweet sentiment quantification: An experimental re-evaluation. **PLOS ONE** 17(9): 1–23, 2022. [586].  
The paper appears on Zenodo at <https://zenodo.org/record/6366468>.

### 8.3.4. Relevant software/datasets/other outcomes

- The code and the datasets for reproducing the results reported in [586] are available at <https://github.com/HLT-ISTI/QuaPy/tree/tweetsent> (code) and <https://zenodo.org/record/4255764> (datasets).

### 8.3.5. Relevance to AI4media use cases and media industry applications

Tweet sentiment quantification, and the work that CNR has carried out on it, has important applicative potential in the media industry, especially in the field of monitoring public opinion (e.g., on a political candidate, on a governmental policy, etc.). Plots of the trends of public opinion often appear on media portals (e.g., public opinion on one or more presidential candidates, that feature on media portals right before preidential elections), and the presented techniques allow these plots to be generated from Twitter with an accuracy higher than it can be obtained via traditional classification techniques.

## 8.4. Multi-Label Quantification

**Contributing partners:** CNR

### 8.4.1. Introduction and methodology

In this work, we describe and compare many different (aggregative) MLQ methods. In order to better assess their relative merits, we subdivide them into four different groups, depending on whether the correlations between different classes are exploited in the classification phase (i.e., by the classifier which provides input to an aggregative quantifier), or in the aggregation phase (i.e., in





the phase in which the individual predictions are aggregated), or in both phases, or in neither of the two phases.

The first and simplest such group is that of MLQ methods that treat each class as completely independent, and thus solve  $n$  independent binary quantification problems. We call such an approach BC+BA (“binary classification followed by binary aggregation”), since in both the classification phase and the aggregation phase we treat the multi-label task as  $n$  independent binary tasks; we thus disregard, in both phases, the correlations among classes when predicting their class prevalence values. This is similar to the Binary Relevance (BR) problem transformation for classification, and consists of transforming the multi-label dataset  $L$  into a set of binary datasets  $L_1, \dots, L_n$  in which  $L_i = \{(\mathbf{x}, \mathbf{1}[y_i \in Y]) : (\mathbf{x}, Y) \in L\}$  is labelled according to  $\mathcal{Y}_i = \{\mathbf{0}, \mathbf{1}\}$ , since the datapoints are relabelled using the indicator function  $\mathbf{1}[z]$  that returns  $\mathbf{1}$  (the minority class) if  $z$  is true or  $\mathbf{0}$  (the majority class) otherwise. BC+BA methods then train one quantifier  $q_i$  for each training set  $L_i$ . At inference time, the prevalence vector for a given sample  $\sigma$  is computed as  $\mathbf{p}_\sigma^{\text{BC+BA}} = (p_\sigma^{q_1}(\mathbf{1}), p_\sigma^{q_2}(\mathbf{1}), \dots, p_\sigma^{q_n}(\mathbf{1}))$ . Although this is technically a multi-label quantification method, BC+BA is actually the trivial solution that we expect any truly multi-label quantifier to beat.

A second, less trivial group is that of MLQ methods based on the use of binary aggregative quantifiers that receive input from (truly) multi-label classifiers. Methods in this group consist of  $n$  independent binary aggregative quantifiers that rely on the (hard or soft) predictions returned by a classifier natively designed to tackle the multi-label problem. Each binary quantifier takes into account only the predictions for its associated class, disregarding the predictions for the other classes. This represents a straightforward solution to the MLQ problem, as it simply combines already existing technologies (binary aggregative quantifiers built via off-the-shelf methods and (truly) multi-label classifiers built via off-the-shelf methods). In such a setting, the classification stage is influenced by the class-class correlations, but the quantification methods in charge of producing the class prevalence estimates for each class do not pay attention to any such correlation, and are disconnected from each other. Since methods in this group will consist of a (truly) multi-label classification phase followed by a binary quantification phase, we will refer to this group of methods as MLC+BA.

We next propose a third group of MLQ systems, i.e., ones consisting of natively multi-label quantification methods that receive as input the outputs of  $n$  independent binary classifiers.

Methods like these represent a non-trivial novel solution for the field of quantification, because no natively multi-label quantification method has been proposed so far in the literature; we here propose some such methods. In order to clearly evaluate the merits of such a multi-label aggregation phase, as the underlying classifiers we use independent binary classifiers only. For this reason, we will call this group of methods BC+MLA.

The methods in the fourth and last group that we consider consist of combinations of a (truly) multi-label classification method and a (truly) multi-label quantification method among our newly proposed ones; this allows to exploit the class dependencies both at the classification stage and at the aggregation stage. We call this group of methods MLC+MLA.

In order to generate members of these four classes, we already have off-the-shelf components for implementing the binary classification, multi-label classification, and binary aggregation phases, but we have no known method from the literature to implement multi-label aggregation; in the next sections we propose two novel methods of this type, one based on exploiting class-class correlations at the aggregation stage by means of regression, and the other based on exploiting class-class correlations at the aggregation stage by means of label powersets.





### 8.4.2. Experiments

We have carried out a number of experiments in order to evaluate the performance of the different methods for MLQ that we have presented in the previous sections. Our goal here is to provide an answer to the question: “Which among the four groups of multi-label quantification methods tends to perform best?”

To this aim, we choose one representative instance from each group, and carry out the experiments using all the datasets. We perform this choice by combining the following components:

- As the **binary classification method**, we choose logistic regression, and use the implementation of it available from SCIKIT-LEARN.<sup>12</sup> We consider LR a good choice, given that it is a probabilistic classifier that already provides fairly well calibrated posterior probabilities (which is of fundamental importance in PCC, PACC, and SLD), and given that, as indicated by previously reported results [567], it tends to perform well. A set of LR classifiers are used when testing the BR method.
- As the **multi-label classification method**, we adopt *stacked generalization* [587] (SG). We use our own implementation (since the implementation of stacked generalization available from SCIKIT-LEARN only caters for the single-label case)<sup>13</sup>, that relies on 5-fold cross-validation to generate the intermediate representations (in the form of posterior probabilities) given as input to the meta-learner, concatenated with the original input features. The base members of the ensemble consist of binary Likelihood Regret classifiers as implemented in SCIKIT-LEARN.
- As the **binary aggregation method**  $Q$ , we experiment with methods CC, PCC, ACC, PACC, SLD. For all these methods we use the implementations made available in the QUAPY open-source library [567].<sup>14</sup>
- As the **multi-label aggregation method**, we use the regressor-based strategy for quantification that we dub RQ. We implement this method as part of the QUAPY framework. For training the base quantifier  $q$  we experiment again with all methods such as CC, PCC, ACC, PACC, SLD, while as the internal regressor which receives its input from the base quantifier  $q$  we use linear Support Vector Regression, for which we use the SCIKIT-LEARN implementation.<sup>15</sup> As the held-out validation set  $L_R$  needed for training the regressor we use a set consisting of 40% of the training datapoints, chosen via iterative stratification [588, 589] as implemented in SCIKIT-MULTILEARN.<sup>16</sup> We call this aggregation method SVR-RQ.

The methods we use in this experiment thus amount to the combinations illustrated in Table 44.

Following [590], we perform model selection by using, as the loss function to minimize, a quantification-oriented error measure (and not a classification-oriented one), and by adopting the same protocol used for the evaluation of our quantifiers. That is, model selection is carried out by first splitting the training set  $L$  into two disjoint sets, i.e., (a) a proper training set  $L_{tr}$  and (b) a held-out validation set  $L_{va}$  consisting of 40% of the labelled datapoints. For splitting the training set, we again rely on the iterative stratification routine of SCIKIT-MULTILEARN. We use  $L_{tr}$  to train the quantifiers with different combinations of hyperparameters, while from  $L_{va}$  we extract, via the ML-APP, validation samples on which we assess, via AE (the same measure we use in the evaluation phase), the quality of the hyperparameter combinations. We explore the hyperparameters via

<sup>12</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.linear\\_model.LogisticRegression.html](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)

<sup>13</sup><https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingClassifier.html>

<sup>14</sup><https://github.com/HLT-ISTI/QuaPy>

<sup>15</sup><https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html>

<sup>16</sup><http://scikit.ml/stratification.html>





Table 44. Methods we use as instances of the four types of methods.

Type	Classification	Aggregation
BC+BA	LR	$Q \in \{CC, PCC, ACC, PACC, SLD\}$
MLC+BA	SG	$Q \in \{CC, PCC, ACC, PACC, SLD\}$
BC+MLA	LR	$Q \in \{CC, PCC, ACC, PACC, SLD\} + \text{SVR-RQ}$
MLC+MLA	SG	$Q \in \{CC, PCC, ACC, PACC, SLD\} + \text{SVR-RQ}$

grid-search optimization, and use the best configuration to retrain the quantifier on the entire training set  $L$  after model selection. During the model selection phase, for the ML-APP we use the same parameters  $k$  and  $\mathbf{g}$  that we use in the test phase, but we reduce the number of repetitions  $m$  to 5 in the datasets with fewer than 90 classes, and to 1 in the other datasets, in order to keep the computational burden under reasonable bounds.

The results we have obtained for the different choices of the base quantifier are reported in Table 45 for SLD, our best-performing multiclass quantification method. The results clearly show (see especially the last two rows) that there is an ordering  $BC+BA \prec MLC+BA \prec BC+MLA \prec MLC+MLA$ , in which  $\prec$  means “performs worse than”, which holds, independently of the base quantifier of choice, in almost all cases. The same experiments also indicate that *there is a substantial improvement in performance that derives from simply replacing the binary classifiers with one multi-label classifier* (moving from BC+BA to MLC+BA or from BC+MLA to MLC+MLA), i.e., from bringing to bear the class-class correlations at the classification stage, and that *there is an equally substantial improvement when binary aggregation is replaced by multi-label aggregation* (switching from BC+BA to BC+MLA or from MLC+BA to MLC+MLA), i.e., when the class-class correlations are exploited at the aggregation stage. What also emerges from these results is that, consistently with the above observations, *the best-performing group of methods is MLC+MLA*, i.e., methods that explicitly take class dependencies into account *both* at the classification stage and at the aggregation stage.

Note that methods that learn from the stochastic correlations among the classes perform much better than methods that do not, even in the low shift regime. Overall, the best-performing method on average is MLC+MLA when equipped with PCC as the base quantifier.

The reader might wonder why we do not use as a baseline the system presented in the only paper in the literature that tackles multi-label quantification, i.e., [591]. There are several reasons for this: (a) the authors do not make the code available; (b) the method is computationally expensive, and as a result the authors test it on a single dataset whose codeframe consists of 16 classes only; using this method on our 15 datasets, whose codeframes count up to 983 classes, and 125 classes on average, would be prohibitive; (c) the method is essentially a calibration strategy for binary classification, which means that it falls in the group of “naive” BC+BA methods since it does not tackle at all the multi-label nature of the MLQ problem.

For more details on this work and additional experiments please check the full paper [592].

### 8.4.3. Relevant publications

- Alejandro Moreo, Manuel Francisco, Fabrizio Sebastiani. Multi-Label Quantification. ACM Transactions on Knowledge Discovery and Data. [592].  
The paper appears on Zenodo at <https://zenodo.org/record/8178996>.





Table 45. Values of AE obtained in our experiments for different amounts of shift using SLD as the base quantifier. The number of test samples generated for each dataset exceeds 10,000, though there is a variable number of samples allocated in each region of shift. **Boldface** indicates the best method for a given dataset and shift region. Superscripts † and ‡ denote the methods (if any) whose scores are not statistically significantly different from the best one according to a Wilcoxon signed-rank test at different confidence levels: symbol † indicates  $0.001 < p\text{-value} < 0.05$  while symbol ‡ indicates  $0.05 \leq p\text{-value}$ . For ease of readability, for each pair {dataset, shift} we colour-code cells via intense green for the best result, intense red for the worst result, and an interpolated tone for the scores in-between.

	low shift				mid shift				high shift			
	BC+BA	MLC+BA	BC+MLA	MLC+MLA	BC+BA	MLC+BA	BC+MLA	MLC+MLA	BC+BA	MLC+BA	BC+MLA	MLC+MLA
Emotions	.2169	.0549	.0710	<b>.0509</b>	.2189	.0719	.0791	<b>.0652</b>	.2088	.0890	.0822	<b>.0717</b>
Scene	.0407	.0433	<b>.0337</b>	.0424	<b>.0467</b>	.0753	.0497	.0709	<b>.0487</b>	.1012	.0628	.0881
Yeast	.2557	.0948	.0511	<b>.0500</b>	.2607	.1192	.0889	<b>.0827</b>	.2939	.1438	.1362	<b>.1171</b>
Birds	.0759	.0284	<b>.0196</b>	.0281	.0819	.0312	<b>.0255</b>	.0312	.1089	.0355 <sup>†</sup>	.0358 <sup>†</sup>	<b>.0351</b>
Genbase	.0011	<b>.0004</b>	.0039	.0005	.0011	<b>.0003</b>	.0042	.0005	.0010	<b>.0003</b>	.0041	.0005
Medical	.0233	.0133	.0190	<b>.0129</b>	.0211	.0135	.0263	<b>.0131</b>	.0189	.0133	.0312	<b>.0132</b>
tmc2007_500	.0384	.0248	.0202	<b>.0187</b>	.0526	.0407	.0285	<b>.0230</b>	.0546	.0432	.0330	<b>.0228</b>
Ohsumed	.0294	.0186	<b>.0173</b>	.0185	.0316	.0232	<b>.0189</b>	.0232	.0321	.0250	<b>.0200</b>	.0250
Enron	.0918	.0208	.0183	<b>.0182</b>	.0915	.0253	<b>.0238</b>	.0243	.0838	.0261 <sup>†</sup>	<b>.0258</b>	.0263 <sup>†</sup>
Reuters-21578	.0050	<b>.0039</b>	.0048	.0040	.0177	<b>.0055</b>	.0079	.0056	.0956	.0088	.0112	<b>.0083</b>
RCV1-v2	.0109	.0089	.0090 <sup>†</sup>	<b>.0088</b>	.0185	<b>.0110</b>	.0151	.0113	.0340	<b>.0173</b>	.0261	.0178
Mediamill	.2040	.0237	.0151	<b>.0145</b>	.2204	.0444	.0238	<b>.0223</b>	.2481	.0695	.0308	<b>.0282</b>
Bibtex	.0819	.0103	<b>.0100</b>	.0101	.0919	.0116 <sup>†</sup>	.0137	<b>.0116</b>	.1084	.0128 <sup>‡</sup>	.0183	<b>.0127</b>
Core15k	.1043	<b>.0098</b>	.0140	.0178	.1041	<b>.0101</b>	.0145	.0177	.1043	<b>.0099</b>	.0155	.0182
Delicious	.1406	.0137	<b>.0095</b>	.0100	.1511	.0155	<b>.0110</b>	.0114	.1345	.0155	<b>.0106</b>	.0108 <sup>‡</sup>
Average	.0842	.0219	.0189	<b>.0182</b>	.0862	.0346	.0296	<b>.0285</b>	.0957	.0562	.0466	<b>.0459</b>
Rank Average	3.8	2.5	2.1	<b>1.7</b>	3.7	2.3	2.3	<b>1.8</b>	3.7	2.3	2.3	<b>1.7</b>

#### 8.4.4. Relevant software/datasets/other outcomes

- The code for reproducing the results reported in [592] is available at <https://github.com/manuel-francisco/quapy-ml/>.

#### 8.4.5. Relevance to AI4media use cases and media industry applications

Multi-label quantification, and the work that CNR has carried out on it, has important applicative potential in the media industry, since news stories (the main unit of meaning in journalism) are the quintessential example of multi-labelled data items (a news story may typically belong to more than one topical class, e.g., be about HomeNews and HealthPolicies at the same time). Multi-label quantification allows one to monitor through time how frequent the news stories belonging to a specific class are, thus allowing to detect trends in the relevance of different issues and in readers' interest.

### 8.5. Other contributions related to Learning to Quantify (Task 3.7)

**Contributing partners:** CNR

Other contributions related to LQ made by CNR in the reporting period are the following:

- A monograph on LQ was published (open-access) by Springer Nature [593]; three of its four







authors (Esuli, Moreo, Sebastiani) are with CNR.

- A data challenge centred on LQ was organized at the CLEF 2022 international conference [594–596]; all of its organizers are with CNR. The data challenge was successful, and six teams from around Europe participated.
- Two international workshops on LQ were organized. LQ 2021 [597–599], the 1st edition of this international workshop series, was co-located with CIKM 2021, and was run entirely online. LQ 2022 [600], the 2nd edition, was co-located with ECML/PKDD 2022, and was run in hybrid form. LQ 2023, which will be co-located with ECML/PKDD 2023, is currently being organized, and will also run in hybrid form. For each of these three workshops, two out of four co-organizers (A. Moreo and F. Sebastiani) are with CNR.
- An 8-hour course on LQ (<https://tinyurl.com/4vhsnzjv>) was given in March 2023 by A. Moreo and F. Sebastiani (both CNR) within the Artificial Intelligence Doctoral Academy (AIDA). The recording of this course is available from the YouTube channel of the CNR Artificial Intelligence for Media and Humanities (AIMH) research group (<https://www.youtube.com/@aimhlabisti-cnr5153>).
- A half-day tutorial on LQ will be offered by A. Moreo and F. Sebastiani (both CNR) at ECML/PKDD 2023 in September 2023.
- A talk on LQ (titled “Exit the Needle, Enter the Haystack: Supervised Machine Learning for Aggregate Data” – <https://tinyurl.com/6yr6hycf>) by F. Sebastiani was given at the AICafé seminar series of the Artificial Intelligence Doctoral Academy (AIDA). The recording of this course is available from the YouTube channel of the CNR Artificial Intelligence for Media and Humanities (AIMH) research group (<https://www.youtube.com/@aimhlabisti-cnr5153>).
- A keynote talk about LQ titled “Quantification: Estimating Class Prevalence via Supervised Learning” was given by A. Moreo (CNR) at the 2022 Workshop on Machine Learning for Astroparticle Physics and Astronomy (ML-ASTRO 2022), co-located with INFORMATIK 2022, Hamburg, DE, September 2022.

### 8.5.1. Relevant publications

- Andrea Esuli, Alejandro Moreo, and Fabrizio Sebastiani. LeQua@CLEF2022: Learning to Quantify. **Proceedings of the 44th European Conference on Information Retrieval (ECIR 2022)**, Stavanger, NO, pp. 374–381, 2022. [594]  
The paper appears on Zenodo at <https://zenodo.org/record/6367103>.
- Andrea Esuli, Alejandro Moreo, Fabrizio Sebastiani, and Gianluca Sperduti. A concise overview of LeQua 2022: Learning to quantify. **Proceedings of the 13th Conference and Labs of the Evaluation Forum (CLEF 2022)**, Bologna, IT, pp. 362–381. [595]  
The paper appears on Zenodo at <https://zenodo.org/record/7090065>.
- Andrea Esuli, Alejandro Moreo, Fabrizio Sebastiani, and Gianluca Sperduti. A detailed overview of LeQua 2022: Learning to quantify. **Working Notes of the 13th Conference and Labs of the Evaluation Forum (CLEF 2022)**, Bologna, IT. [596]  
The paper appears on Zenodo at <https://zenodo.org/record/7090031>.
- Andrea Esuli, Alessandro Fabris, Alejandro Moreo, and Fabrizio Sebastiani. **Learning to Quantify**. Springer Nature, Cham, CH, 2023. [593]



- Juan José del Coz, Pablo González, Alejandro Moreo, and Fabrizio Sebastiani (eds.). **Proceedings of the 1st International Workshop on Learning to Quantify (LQ 2021)**, Gold Coast, AU, 2021. [597]
- Juan José del Coz, Pablo González, Alejandro Moreo, and Fabrizio Sebastiani. Learning to Quantify: Methods and Applications (LQ 2021). **Proceedings of the 30th ACM International Conference on Knowledge Management (CIKM 2021)**, Gold Coast, AU, pp. 4874–4875. [598]  
The paper appears on Zenodo at <https://zenodo.org/record/6418155>.
- Juan José del Coz, Pablo González, Alejandro Moreo, and Fabrizio Sebastiani. Report on the 1st International Workshop on Learning to Quantify (LQ 2021). **SIGKDD Explorations** 24(1):49–51, 2022. [599]  
The paper appears on Zenodo at <https://zenodo.org/record/7090007>.
- Juan José del Coz, Pablo González, Alejandro Moreo, and Fabrizio Sebastiani (eds.). **Proceedings of the 2nd International Workshop on Learning to Quantify (LQ 2022)**, Grenoble, FR, 2022. [600]  
The proceedings appear on Zenodo at <https://zenodo.org/record/7093004>.

#### 8.5.2. Relevant software/datasets/other outcomes

- The datasets used in the LeQua 2022 data challenge [594–596] are available on Zenodo at <https://zenodo.org/record/6546188>

#### 8.5.3. Relevance to AI4media use cases and media industry applications

Learning to quantify is important for the media industry, since it allows to monitor temporal trends of indicators relevant to journalism, such as public opinion on specific topics (see Section 8.3.5) and the frequency of journalistic news belonging to specific classes (see Section 8.4.5). The work described in this section contributes in various ways to increasing our knowledge of learning to quantify.





## 9. Ongoing Work and Conclusions

### 9.1. Ongoing work

Below, we briefly summarize the ongoing work associated to each task.

#### 9.1.1. Lifelong and on-line learning (Task 3.1)

**CEA** is: (1) investigating the advantages and limitations of using large pre-trained models in continual learning, (2) studying the feasibility of predicting which incremental learning approach (algorithm, backbone, pre-training vs. supervised training) should be used for a specific use case without resorting to precomputed resources, and (3) exploring ways to adapt large pre-trained models for computation- and memory-constrained devices.

**UNITN** is looking into GCD, a recently proposed open-world problem, which aims to automatically cluster partially labeled data. The main challenge is that the unlabeled data contain instances that are not only from known categories of the labeled data but also from novel categories. This leads traditional novel category discovery (NCD) methods to be incapacitated for GCD, due to their assumption of unlabeled data are only from novel categories. One effective way for GCD is applying self-supervised learning to learn discriminate representation for unlabeled data.

**AUTH** will continue working on the proposed framework for neural models that combines and unifies OOD, incremental/continual/lifelong learning, and neural distillation. Specifically, AUTH will look into the problem of reliable knowledge assessment in teacher-student network frameworks. The primary objective is to establish a clear definition of knowledge within Teacher-Student network frameworks and develop an assessment methodology to evaluate the knowledge of individual agents and potential teachers.

**UNIFI** is working on a novel learning protocol in which a large model (also known as foundation model) undergoing Continual Learning will be replaced by an improved one that has been learned from scratch in a compatible way elsewhere (e.g., on a remote server). In recent times, there has been a growing trend of fine-tuning pretrained models, which are becoming larger in size. To handle this, learning is increasingly being performed remotely using specialized servers with high computing capacities. These models are then fine-tuned locally to adapt them to specific tasks of interest. However, when large pre-trained models are re-trained from scratch to take advantage of new data, innovative architectures or other advanced learning techniques, it is crucial that the locally fine-tuned model be seamlessly replaced. This replacement should incorporate these advancements without disrupting the visual search service, particularly through the outdated extracted features in the gallery-set.

#### 9.1.2. Manifold learning and disentangled feature representation (Task 3.2)

**QMUL** will continue to focus on Visual-Language models and ways of improving their discriminative ability (e.g., in terms of Zero-Shot classification) by means of learning better and more disentangled representations in their joint image-text spaces. More specifically, we will focus on (i) fine-tuning VL models and/or prompt learning for domain-specific tasks (e.g., facial expression recognition), and (ii) learning to separate representations of the different visual modalities in VL model's joint image-text space in order to improve its discriminative ability.

**JR** will base upon the work done for improving out-of-distribution performance of neural network models with manifold mixing model soups. There are several research directions which are worth investigating: (a) investigate its performance for models outside the vision domain, e.g. from NLP or audio; (b) improve the capability of the algorithm for the case when the layers of the





finetuned models are diverging more; (c) adding a second phase of the model soup algorithm where the optimization of all component coefficients is done simultaneously.

**UNIFI** will continue to work on generative models on non-linear (manifold) domains focusing on models capable of generating in the combined spatial-temporal domain. More specifically, we will focus on generation of long-term sequences of dynamically changing human behaviour. This targets the generation of trajectories that can model the temporal evolution of landmarks or joints of the body in a smooth and natural way (e.g., for the synthesis of facial expressions, talking heads or human body movement, in interaction or computer graphics applications). New ways of separating the temporal and spatial generation of dynamic behavior will be also be investigated.

**UNITN** will continue to investigate the underlying structure of the latent spaces of deep generative models with the goal of performing semantically meaningful latent traversals. We will look into modeling latent structures with a learned dynamic potential landscape, thereby performing latent traversals as the flow of samples down the landscape's gradient. Inspired by physics, optimal transport, and neuroscience, these potential landscapes could be learned as physically realistic partial differential equations, thereby allowing them to flexibly vary over both space and time. To achieve disentanglement, multiple potentials could be learned simultaneously, and could be constrained by a classifier to be distinct and semantically self-consistent. This solution can be integrated as a regularization term during training, thereby acting as an inductive bias towards the learning of structured representations, ultimately improving model likelihood on similarly structured data.

### 9.1.3. Transfer learning (Task 3.3)

**BSC** will continue studying the trade-offs between performance, carbon footprint and computational requirements of transfer learning methods as in the presented contribution. We will now benchmark newer, state-of-the-art transfer learning methods like Low-Rank Domain Adaptation (LoRA) and extensions of it (e.g. LoKR, etc.). We will also include source-target transferability metrics (i.e. metrics that assess how transferable is the knowledge of a pre-trained model for a given new task) instead of a manual classification of the target datasets.

**UNITN** is investigating the application of TTDA-Seg where both efficiency and effectiveness are crucial. We will look into a backward-free approach for TTDA-Seg which is utilizing each instance to dynamically guide its own adaptation in a non-parametric way, which avoids the error accumulation issue and expensive optimizing cost.

**CEA** is studying the possibility to diversify training datasets in a programmatic manner by: (1) combining sets of semantic queries adapted per class, pre-trained foundation models, and visual clustering and (2) using prompting of multimodal pre-trained models with diversified semantic queries.

**CNR** is currently working on assessing transfer learning (TL) abilities in the context of heterogeneous domains, with a specific focus on the domains of language and vision in Vision-and-Language (VL) models. Despite promising results on multimodal tasks, recent literature has shown that models integrating image and text are highly susceptible to statistical bias present in large-scale training data. CNR researchers are thus focusing their analysis on Video-and-Language models, and constructing a benchmark revolving around the concepts of action, pre-state, and post-state (i.e., change-of-state verbs). The benchmark focuses on the temporally ordered (sub)phases of these events, to provide the research community with a tool to better understand and diagnose the integration of the video and textual domains.





#### 9.1.4. Deep quality diversity (Task 3.6)

UM is carrying out promising experiments following up on activities reported under Section 7.3 intending for a high-impact journal publication around Computational Creativity. Following this, upcoming research will focus on the one side on algorithmical advances that leverage more recent ML algorithms (including e.g. transformer architectures) with QD evolutionary search, and on the other side on designing and developing applications that better take advantage of these algorithms and make them available to the broader public. On the latter note, work so far has focused on experimental validation, but building an interface and interaction paradigm for a “real-world” creative problem will allow us to test the algorithms in real-world settings. This is an important direction and test, since the goal of research in T3.6 (and AI4Media more broadly) is assisting human users via (explainable) AI.

UNITN will organize together with UM the “Computer Vision for Games and Games for Computer Vision (CVG)” workshop, to be held on November 23, 2023, as part of the British Machine Vision Conference (BMVC) in Aberdeen, UK. The workshop aims to foster collaboration and knowledge exchange between the computer vision and games research communities, which have traditionally operated independently. The symbiotic relationship between video games and computer vision has been significant, with virtual worlds serving as valuable sources of training data and testbeds for computer vision models. Moreover, computer vision advancements have revolutionized the creation and possibilities within artificial game worlds. However, several research questions and technical challenges still remain unaddressed in both fields.

#### 9.1.5. Learning to count (Task 3.7)

CNR is currently working on the development of deep neural networks for LQ. In particular, CNR is studying the suitability to this task of permutation-invariant operators for set processing. Among these, CNR is currently investigating the potential benefits of histogram-based functions, in a new architecture that has been dubbed HistNetQ.

CNR is also working on the application of LQ for estimating the effectiveness, via any chosen evaluation function, of a classifier when applied to unlabelled sets that exhibit dataset shift with respect to the data the classifier has been trained on.

Additionally, CNR is working on the problem of tailoring quantification approaches to the particular type of shift that the set of unlabelled data exhibits; this is an important problem, since until now quantification approaches have mostly been tested on prior probability shift only.

## 9.2. Conclusions

In this deliverable, we presented the current research results of WP3 regarding the new learning paradigms, specifically on the tasks: 3.1 (lifelong and on-line learning), 3.2 (manifold learning and disentangled feature representation), 3.3 (transfer learning), 3.6 (deep quality diversity), and 3.7 (learning to count).

Several new methodologies bringing novel solutions and state-of-the-art results are presented. These include new approaches for NCD, CIL, knowledge quantification metrics, and a teacher-student network framework which supports “learning by education”, which fall under the category of lifelong and on-line learning (Task 3.1). The presented works are particularly relevant to the use cases of AI4Media, since they can significantly reduce the catastrophic forgetting in a scenario with several updates being rehearsal-free (i.e., no episodic memory), and where a gallery’s features in a visual search systems does not require to be re-computed (re-indexed) when the model is updated in a lifelong learning scenario.





Under the task of manifold learning and disentangled feature representation (Task 3.2), in this deliverable we presented several works that have been published in top conferences of Computer Vision and Machine Learning and include a plethora of methods for finding meaningful representation schemes for both the generative and the discriminative learning paradigms. In the generative regime, we presented works on studying the structure of latent spaces of generative methods (such as GANs) by discovering semantic paths that govern the generation process, allowing this way visual content generation (e.g., image editing). Moreover, we presented work on learning meaningful feature representations, along with metrics that model data manifolds better (i.e., by adopting the hyperbolic geometry), lead to better and more discriminative features, and, thus, improve significantly the performance in visual understanding tasks (such as image retrieval). Advances in both generative and discriminative regimes are particularly useful in media generation and visual content analysis use cases of AI4Media.

Important contributions, showing improved state-of-the-art results, have been demonstrated for transfer learning (Task 3.3) as well. The state-of-the-art source-free open domain adaptation approaches that have been proposed, especially those incorporating uncertainty in the source model predictions, can be particularly useful to the use cases of AI4Media, towards discovery of new visual content (and its adaptation), improving tagging and search capabilities, and being able to generalize under domain-gap.

Moreover, we reported important developments in deep quality diversity (Task 3.6) for providing novel ways to generate diverse content without requiring ad-hoc designer-specified directions for this diversity, and combine content of different facets into a playable experience. Also, developments in Task 3.6 allow for modelling the subjective human experiences of players to dynamically adapt the game itself according to the user's (predicted) engagement or arousal levels, serving very relevant use cases of AI4Media. The proposed algorithms for Interactive Quality-Diversity search contribute to human co-creation) in terms of a new way of interacting with an evolving computational process while taking advantage of the important concept of quality-diversity balance.

Several novel approaches have been developed for the problem of "Learning to quantify" (Task 3.7), including an open-source framework written in Python. The most important contributions in this task include work on ordinal and multi-label quantification, as well as a systematic comparison of LQ methods on the task of tweet sentiment quantification. The results of this task have led to several solutions towards visual content analysis and, thus, can be of service for many use cases of AI4Media.

In summary, the activity so far has been very intense and successful, both in terms of published articles (19 conference and 6 journal articles) and in terms of open-source software and tools. The work reported in this deliverable is of top quality (published in the most prestigious venues of the community) and reflects the active involvement of all partners towards achieving the goals of WP3 in particular and those of AI4Media in general. Ongoing work and future plans of all partners regarding the tasks presented in this deliverable are also very promising, convincingly reassuring the good continuation of the work according to the original planning.





## References

- [1] C. Tzelepis, G. Tzimiropoulos, and I. Patras, “Warpedganspace: Finding non-linear rbf paths in gan latent space,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6393–6402, 2021.
- [2] J. Oldfield, C. Tzelepis, Y. Panagakis, M. A. Nicolaou, and I. Patras, “Panda: Unsupervised learning of parts and appearances in the feature maps of gans,” *arXiv preprint arXiv:2206.00048*, 2022.
- [3] S. Barattin, C. Tzelepis, I. Patras, and N. Sebe, “Attribute-preserving face dataset anonymization via latent code optimization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8001–8010, 2023.
- [4] A. Ermolov, L. Mirvakhobova, V. Khrulkov, N. Sebe, and I. Oseledets, “Hyperbolic vision transformers: Combining improvements in metric learning,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, vol. 139, pp. 8748–8763, PMLR, 2021.
- [6] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu, “Coca: Contrastive captioners are image-text foundation models,” *Trans. Mach. Learn. Res.*, vol. 2022, 2022.
- [7] J. K. Pugh, L. B. Soros, and K. O. Stanley, “Quality diversity: A new frontier for evolutionary computation,” *Frontiers in Robotics and AI*, vol. 3, 2016.
- [8] A. Liapis, H. P. Martínez, J. Togelius, and G. N. Yannakakis, “Transforming exploratory creativity with DeLeNoX,” in *Proceedings of the Fourth International Conference on Computational Creativity*, pp. 56–63, 2013.
- [9] K. O. Stanley, “Compositional pattern producing networks: A novel abstraction of development,” *Genetic programming and evolvable machines*, vol. 8, no. 2, pp. 131–162, 2007.
- [10] J. Lehman and K. O. Stanley, “Evolving a diversity of virtual creatures through novelty search and local competition,” in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pp. 211–218, 2011.
- [11] J.-B. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *ArXiv preprint*, vol. abs/1504.04909, 2015.
- [12] K. Sfikas, A. Liapis, and G. N. Yannakakis, “A general-purpose expressive algorithm for room-based environments,” in *Proceedings of the FDG workshop on Procedural Content Generation in Games*, 2022.
- [13] C. J. Burges, “From RankNet to LambdaRank to LambdaMART: An overview,” Tech. Rep. MSR-TR-2010-82, Microsoft Research, 2010.
- [14] W. Gao and F. Sebastiani, “From classification to quantification in tweet sentiment analysis,” *Social Network Analysis and Mining*, vol. 6, no. 19, pp. 1–22, 2016.





- [15] J. C. Schlimmer and D. Fisher, “A case study of incremental concept induction,” in *AAAI*, vol. 86, pp. 496–501, 1986.
- [16] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, “Measuring catastrophic forgetting in neural networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.
- [17] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” *The Psychology of Learning and Motivation*, vol. 24, pp. 104–169, 1989.
- [18] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, “End-to-end incremental learning,” in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XII*, pp. 241–257, 2018.
- [19] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, “Learning a unified classifier incrementally via rebalancing,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 831–839, 2019.
- [20] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “icarl: Incremental classifier and representation learning,” in *Conference on Computer Vision and Pattern Recognition, CVPR, 2017*.
- [21] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, “Large scale incremental learning,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 374–382, 2019.
- [22] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S. Xia, “Maintaining discrimination and fairness in class incremental learning,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 13205–13214, IEEE, 2020.
- [23] R. Venkatesan, H. Venkateswara, S. Panchanathan, and B. Li, “A strategy for an uncompromising incremental learner,” *arXiv preprint arXiv:1705.00744*, 2017.
- [24] L. Ravaglia, M. Rusci, D. Nadalini, A. Capotondi, F. Conti, and L. Benini, “A tinyml platform for on-device continual learning with quantized latent replays,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 4, pp. 789–802, 2021.
- [25] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. van de Weijer, “Semantic drift compensation for class-incremental learning,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 6980–6989, IEEE, 2020.
- [26] J. Smith, Y.-C. Hsu, J. Balloch, Y. Shen, H. Jin, and Z. Kira, “Always be dreaming: A new approach for data-free class-incremental learning,” *arXiv preprint arXiv:2106.09701*, 2021.
- [27] F. Zhu, Z. Cheng, X.-y. Zhang, and C.-l. Liu, “Class-incremental learning via dual augmentation,” *Advances in Neural Information Processing Systems*, vol. 34, 2021.
- [28] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu, “Prototype augmentation and self-supervision for incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5871–5880, 2021.







- [29] G. E. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *CoRR*, vol. abs/1503.02531, 2015.
- [30] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, “Class-incremental learning: survey and performance evaluation on image classification,” 2021.
- [31] A. Prabhu, P. H. Torr, and P. K. Dokania, “Gdumb: A simple approach that questions our progress in continual learning,” in *European Conference on Computer Vision*, pp. 524–540, Springer, 2020.
- [32] K. Zhu, W. Zhai, Y. Cao, J. Luo, and Z.-J. Zha, “Self-sustaining representation expansion for non-exemplar class-incremental learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9296–9305, 2022.
- [33] E. Belouadah and A. Popescu, “Deesil: Deep-shallow incremental learning,” *TaskCV Workshop @ ECCV 2018.*, 2018.
- [34] A. R. Dhamija, T. Ahmad, J. Schwan, M. Jafarzadeh, C. Li, and T. E. Boult, “Self-supervised features improve open-world learning,” *arXiv preprint arXiv:2102.07848*, 2021.
- [35] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.
- [36] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *Proc. ICANN*, 2018.
- [37] A. Krizhevsky, “Learning multiple layers of features from tiny images,” tech. rep., University of Toronto, 2009.
- [38] Y. Le and X. Yang, “Tiny imagenet visual recognition challenge,” *CS 231N*, 2015.
- [39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [40] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.
- [41] T. L. Hayes and C. Kanan, “Lifelong machine learning with deep streaming linear discriminant analysis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 220–221, 2020.
- [42] E. Belouadah, A. Popescu, and I. Kanellos, “A comprehensive study of class incremental learning algorithms for visual tasks,” *Neural Networks*, 2020.
- [43] G. Petit, A. Popescu, H. Schindler, D. Picard, and B. Delezoide, “Fetritl: Feature translation for exemplar-free class-incremental learning,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3911–3920, 2023.
- [44] E. Belouadah, A. Popescu, and I. Kanellos, “A comprehensive study of class incremental learning algorithms for visual tasks,” *Neural Networks*, vol. 135, pp. 38–54, 2021.





- [45] G. Wu, S. Gong, and P. Li, “Striking a balance between stability and plasticity for class-incremental learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1124–1133, 2021.
- [46] E. Belouadah, A. Popescu, and I. Kanellos, “Initial classifier weights replay for memoryless class incremental learning,” in *British Machine Vision Conference (BMVC)*, 2020.
- [47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.
- [48] E. Feillet, G. Petit, A. Popescu, M. Reyboz, and C. Hudelot, “Advisil-a class-incremental learning advisor,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2400–2409, 2023.
- [49] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.
- [50] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “Shufflenet v2: Practical guidelines for efficient cnn architecture design,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.
- [51] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255, 2009.
- [52] L. Bossard, M. Guillaumin, and L. Van Gool, “Food-101 – mining discriminative components with random forests,” in *European Conference on Computer Vision*, 2014.
- [53] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, “The inaturalist species classification and detection dataset,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.
- [54] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, “Large-scale image retrieval with attentive deep local features,” in *ICCV*, pp. 3476–3485, IEEE Computer Society, 2017.
- [55] D. Papaioannou, V. Mygdalis, and I. Pitas, “Towards human society-inspired decentralized dnn inference,” *Under Review*.
- [56] H. Ma, Y. Zhang, F. Zhou, and Q. Zhang, “Quantifying layerwise information discarding of neural networks,” *arXiv preprint arXiv:1906.04109*, 2019.
- [57] R. Fong and A. Vedaldi, “Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8730–8738, 2018.
- [58] C. Guan, X. Wang, Q. Zhang, R. Chen, D. He, and X. Xie, “Towards a deep and unified understanding of deep neural models in nlp,” *36th International Conference on Machine Learning*, pp. 2454–2463, 2019.





- [59] S. Yasaei Sekeh and A. O. Hero, “Geometric estimation of multivariate dependency,” *Entropy*, vol. vol.21, no. no.8, p. 787, 2019.
- [60] Q. Zhang, X. Cheng, Y. Chen, and Z. Rao, “Quantifying the knowledge in a DNN to explain knowledge distillation for classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [61] M. Li, S. Wang, and Q. Zhang, “Visualizing the emergence of intermediate visual patterns in dnns,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 6594–6607, 2021.
- [62] I. Valsamara, I. Mademlis, and I. Pitas, “Quantifying the knowledge in deep neural networks: an overview,” *Under Review*.
- [63] Z. Xiao, Q. Yan, and Y. Amit, “Likelihood regret: An out-of-distribution detection score for variational auto-encoder,” *Advances in neural information processing systems*, vol. 33, pp. 20685–20696, 2020.
- [64] A. Kaimakamidis, I. Valsamara, and I. Pitas, “Knowledge distillation-driven communication framework for neural networks: Enabling efficient student-teacher interactions,” *technical report*.
- [65] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Proceedings of NIPS*, pp. 1106–1114, 2012.
- [66] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *Proceedings of ICLR*, 2014.
- [67] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going Deeper With Convolutions,” in *Proceedings of CVPR*, pp. 1–9, June 2015.
- [68] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of CVPR*, pp. 770–778, 2016.
- [69] Y. Chen, F. Shi, A. G. Christodoulou, Y. Xie, Z. Zhou, and D. Li, “Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network,” in *Proceedings of MICCAI*, pp. 91–99, 2018.
- [70] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Proceedings of MICCAI*, pp. 234–241, 2015.
- [71] W. Kuo, C. Häne, P. Mukherjee, J. Malik, and E. L. Yuh, “Expert-level detection of acute intracranial hemorrhage on head computed tomography using deep learning,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22737–22745, 2019.
- [72] M. Burduja, R. T. Ionescu, and N. Verga, “Accurate and efficient intracranial hemorrhage detection and subtype classification in 3d ct scans with convolutional and long short-term memory neural networks,” *Sensors*, vol. 20, no. 19, p. 5611, 2020.
- [73] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, and D. Amodei, “Language Models are Few-Shot Learners,” in *Proceedings of NeurIPS*, 2020.





- [74] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of NAACL*, pp. 4171–4186, 2019.
- [75] X. Zhang, J. Zhao, and Y. LeCun, “Character-level Convolutional Networks for Text Classification,” in *Proceedings of NIPS*, pp. 649–657, 2015.
- [76] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-Aware Neural Language Models,” in *Proceedings of AAAI*, pp. 2741–2749, 2016.
- [77] X.-L. Zhang and J. Wu, “Denoising deep neural networks based voice activity detection,” in *Proceedings of ICASSP*, pp. 853–857, 2013.
- [78] M. Ravanelli and Y. Bengio, “Speaker Recognition from Raw Waveform with SincNet,” in *Proceedings of SLT*, pp. 1021–1028, 2018.
- [79] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [80] J. L. Elman, “Learning and development in neural networks: the importance of starting small,” *Cognition*, vol. 48, no. 1, pp. 71–99, 1993.
- [81] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of ICML*, pp. 41–48, 2009.
- [82] R. T. Ionescu, B. Alexe, M. Leordeanu, M. Popescu, D. P. Papadopoulos, and V. Ferrari, “How hard can it be? estimating the difficulty of visual search in an image,” in *Proceedings of CVPR*, pp. 2157–2166, 2016.
- [83] M. Shi and V. Ferrari, “Weakly supervised object localization using size estimates,” in *Proceedings of ECCV*, pp. 105–121, 2016.
- [84] Y. Tang, X. Wang, A. P. Harrison, L. Lu, J. Xiao, and R. M. Summers, “Attention-guided curriculum learning for weakly supervised classification and localization of thoracic diseases on chest radiographs,” in *Proceedings of MLMI*, pp. 249–258, 2018.
- [85] X. Chen and A. Gupta, “Webly supervised learning of convolutional networks,” in *Proceedings of ICCV*, pp. 1431–1439, 2015.
- [86] S. Li, X. Zhu, Q. Huang, H. Xu, and C. J. Kuo, “Multiple instance curriculum learning for weakly supervised object detection,” in *Proceedings of BMVC*, 2017.
- [87] E. Sangineto, M. Nabi, D. Culibrk, and N. Sebe, “Self paced deep learning for weakly supervised object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 712–725, 2018.
- [88] J. Wang, X. Wang, and W. Liu, “Weakly-and semi-supervised Faster R-CNN with curriculum learning,” in *Proceedings of ICPR*, pp. 2416–2421, 2018.
- [89] T. Kocmi and O. Bojar, “Curriculum learning and minibatch bucketing in neural machine translation,” in *Proceedings of RANLP*, pp. 379–386, 2017.
- [90] X. Zhang, G. Kumar, H. Khayrallah, K. Murray, J. Gwinnup, M. J. Martindale, P. McNamee, K. Duh, and M. Carpuat, “An empirical exploration of curriculum learning for neural machine translation,” *arXiv preprint arXiv:1811.00739*, 2018.





- [91] E. A. Platanios, O. Stretcu, G. Neubig, B. Póczos, and T. M. Mitchell, “Competence-based curriculum learning for neural machine translation,” in *Proceedings of NAACL*, pp. 1162–1172, 2019.
- [92] W. Wang, I. Caswell, and C. Chelba, “Dynamically composing domain-data selection with clean-data selection by “co-curricular learning” for neural machine translation,” in *Proceedings of ACL*, pp. 1282–1292, July 2019.
- [93] A. Shrivastava, A. Gupta, and R. Girshick, “Training region-based object detectors with online hard example mining,” in *Proceedings of CVPR*, pp. 761–769, 2016.
- [94] A. Jesson, N. Guizard, S. H. Ghahlehjeh, D. Goblot, F. Soudan, and N. Chapados, “CASED: curriculum adaptive sampling for extreme data imbalance,” in *Proceedings of MICCAI*, pp. 639–646, 2017.
- [95] P. Wang and N. Vasconcelos, “Towards realistic predictors,” in *Proceedings of ECCV*, pp. 36–51, 2018.
- [96] T. Zhou, S. Wang, and J. A. Bilmes, “Curriculum learning by dynamic instance hardness,” *Proceedings of NIPS*, vol. 33, 2020.
- [97] T. Pi, X. Li, Z. Zhang, D. Meng, F. Wu, J. Xiao, and Y. Zhuang, “Self-paced boost learning for classification,” in *Proceedings of IJCAI*, pp. 1932–1938, 2016.
- [98] S. Braun, D. Neil, and S.-C. Liu, “A curriculum learning method for improved noise robustness in automatic speech recognition,” in *Proceedings of EUSIPCO*, pp. 548–552, 2017.
- [99] H. N. Pathak and R. Paffenroth, “Parameter continuation methods for the optimization of deep neural networks,” in *Proceedings of ICMLA*, pp. 1637–1643, 2019.
- [100] E. L. Allgower and K. Georg, *Introduction to numerical continuation methods*. SIAM, 2003.
- [101] S. Richter and R. DeCarlo, “Continuation methods: Theory and applications,” *IEEE Transactions on Automatic Control*, vol. 28, no. 6, pp. 660–665, 1983.
- [102] J. Chow, L. Udpa, and S. Udpa, “Homotopy continuation methods for neural networks,” in *Proceedings of ISCAS*, pp. 2483–2486, 1991.
- [103] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, “Curriculum learning for reinforcement learning domains: A framework and survey,” *Journal of Machine Learning Research*, vol. 21, pp. 1–50, 2020.
- [104] X. Wang, Y. Chen, and W. Zhu, “A comprehensive survey on curriculum learning,” *arXiv preprint arXiv:2010.13166v1*, 2020.
- [105] P. Soviany, R. Ionescu, P. Rota, and N. Sebe, “Curriculum learning: A survey,” *International Journal of Computer Vision*, vol. 130, no. 6, pp. 1526–1565, 2022.
- [106] R. M. French, “Catastrophic forgetting in connectionist networks,” *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.
- [107] B. L. Anderson, “Can computational goals inform theories of vision?,” *Topics in Cognitive Science*, 2015.





- [108] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [109] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [110] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *ICLR*, 2021.
- [111] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9912–9924, 2020.
- [112] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” *arXiv:2104.14294*, 2021.
- [113] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman, “Automatically discovering and learning new visual categories with ranking statistics,” in *Proc. ICLR*, 2020.
- [114] K. Han, A. Vedaldi, and A. Zisserman, “Learning to discover novel visual categories via deep transfer clustering,” in *Proc. ICCV*, 2019.
- [115] Z. Zhong, E. Fini, S. Roy, Z. Luo, E. Ricci, and N. Sebe, “Neighborhood contrastive learning for novel class discovery,” in *CVPR*, 2021.
- [116] Z. Zhong, L. Zhu, Z. Luo, S. Li, Y. Yang, and N. Sebe, “Openmix: Reviving known knowledge for discovering novel visual categories in an open world,” in *CVPR*, 2021.
- [117] E. Fini, E. Sangineto, S. Lathuilière, Z. Zhong, M. Nabi, and E. Ricci, “A unified objective for novel class discovery,” in *ICCV*, 2021.
- [118] Y.-C. Hsu, Z. Lv, and Z. Kira, “Learning to cluster in order to transfer across domains and tasks,” in *Proc. ICLR*, 2018.
- [119] Y.-C. Hsu, Z. Lv, J. Schlosser, P. Odom, and Z. Kira, “Multi-class classification without multi-class labels,” in *Proc. ICLR*, 2019.
- [120] J. Liang, D. Hu, and J. Feng, “Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation,” in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 6028–6039, 2020.
- [121] Z. Zheng and Y. Yang, “Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation,” *International Journal of Computer Vision (IJCV)*, 2021.
- [122] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, “A continual learning survey: Defying forgetting in classification tasks,” *TPAMI*, 2021.
- [123] Y. Liu and T. Tuytelaars, “Residual tuning: Toward novel category discovery without labels,” *TNNLS*, 2022.





- [124] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, “Learning from noisy labels with distillation,” in *Proc. ICCV*, 2017.
- [125] J. Xie, R. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in *ICML*, 2016.
- [126] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, “Dark experience for general continual learning: a strong, simple baseline,” in *NeurIPS*, 2020.
- [127] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato, “Continual learning with tiny episodic memories,” in *ICML*, 2019.
- [128] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “icarl: Incremental classifier and representation learning,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.
- [129] A. Krizhevsky, G. Hinton, *et al.*, “Learning multiple layers of features from tiny images,” *University of Tronto*, 2009.
- [130] K. Han, A. Vedaldi, and A. Zisserman, “Learning to discover novel visual categories via deep transfer clustering,” in *ICCV*, 2019.
- [131] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, “Class-incremental learning: survey and performance evaluation on image classification,” *arXiv preprint arXiv:2010.15277*, 2020.
- [132] S. Roy, M. Liu, Z. Zhong, N. Sebe, and E. Ricci, “Class-incremental novel class discovery,” in *European Conference on Computer Vision*, 2022.
- [133] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pp. 539–546, 2005.
- [134] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [135] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” in *Advances in Neural Information Processing Systems (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds.)*, vol. 27, Curran Associates, Inc., 2014.
- [136] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: A unified embedding for face recognition and clustering,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, jun 2015.
- [137] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks.,” in *ICML*, vol. 2, p. 7, 2016.
- [138] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.





- [139] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.
- [140] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.
- [141] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Deep metric learning for person re-identification,” in *2014 22nd International Conference on Pattern Recognition*, pp. 34–39, IEEE, 2014.
- [142] W. Li, R. Zhao, T. Xiao, and X. Wang, “Deepreid: Deep filter pairing neural network for person re-identification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 152–159, 2014.
- [143] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, “Person re-identification in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1367–1376, 2017.
- [144] T. Chen, S. Ding, J. Xie, Y. Yuan, W. Chen, Y. Yang, Z. Ren, and Z. Wang, “Abd-net: Attentive but diverse person re-identification,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8351–8361, 2019.
- [145] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, “Neural codes for image retrieval,” in *European conference on computer vision*, pp. 584–599, Springer, 2014.
- [146] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, “Deep image retrieval: Learning global representations for image search,” in *European conference on computer vision*, pp. 241–257, Springer, 2016.
- [147] G. Toliás, R. Sivic, and H. Jégou, “Particular Object Retrieval With Integral Max-Pooling of CNN Activations,” in *ICLR 2016 - International Conference on Learning Representations*, International Conference on Learning Representations, (San Juan, Puerto Rico), pp. 1–12, May 2016.
- [148] S. D. Khan and H. Ullah, “A survey of advances in vision-based vehicle re-identification,” *Computer Vision and Image Understanding*, vol. 182, pp. 50–63, 2019.
- [149] Y. Shen, Y. Xiong, W. Xia, and S. t. Soatto, “Towards backward-compatible representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6368–6377, 2020.
- [150] R. Van Noorden, “The ethical questions that haunt facial-recognition research,” *Nature*, vol. 587, no. 7834, pp. 354–358, 2020.
- [151] Y. Shen, Y. Xiong, W. Xia, and S. Soatto, “Towards backward-compatible representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6368–6377, 2020.
- [152] R. Duggal, H. Zhou, S. Yang, Y. Xiong, W. Xia, Z. Tu, and S. Soatto, “Compatibility-aware heterogeneous visual search,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10723–10732, 2021.







- [153] K. Chen, Y. Wu, H. Qin, D. Liang, X. Liu, and J. Yan, “R3 adversarial network for cross model face recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9868–9876, 2019.
- [154] J. Hu, R. Ji, H. Liu, S. Zhang, C. Deng, and Q. Tian, “Towards visual feature translation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3004–3013, 2019.
- [155] C. Wang, Y. Chang, S. Yang, D. Chen, and S. Lai, “Unified representation learning for cross model compatibility,” in *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*, BMVA Press, 2020.
- [156] Q. Meng, C. Zhang, X. Xu, and F. Zhou, “Learning compatible embeddings,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9939–9948, October 2021.
- [157] F. Pernici, M. Bruni, C. Baecchi, F. Turchini, and A. Del Bimbo, “Class-incremental learning with pre-allocated fixed classifiers,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 6259–6266, IEEE, 2021.
- [158] F. Pernici, M. Bruni, C. Baecchi, and A. Del Bimbo, “Maximally compact and separated features with regular polytope networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 46–53, June 2019.
- [159] F. Pernici, M. Bruni, C. Baecchi, and A. D. Bimbo, “Regular polytope networks,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.
- [160] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.
- [161] K. A. Ericsson and W. Kintsch, “Long-term working memory,” *Psychological review*, vol. 102, no. 2, p. 211, 1995.
- [162] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.
- [163] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable person re-identification: A benchmark,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1116–1124, 2015.
- [164] L. Zheng, Y. Yang, and A. G. Hauptmann, “Person re-identification: Past, present and future,” *arXiv preprint arXiv:1610.02984*, 2016.
- [165] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, “Omni-scale feature learning for person re-identification,” in *ICCV*, 2019.
- [166] A. B. Yandex and V. Lempitsky, “Aggregating local deep features for image retrieval,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1269–1277, 2015.
- [167] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, “Deep image retrieval: Learning global representations for image search,” in *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 241–257, Springer International Publishing, 2016.





- [168] G. Toliás, R. Sivic, and H. Jégou, “Particular Object Retrieval With Integral Max-Pooling of CNN Activations,” in *ICLR 2016 - International Conference on Learning Representations*, International Conference on Learning Representations, May 2016.
- [169] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of learning and motivation*, vol. 24, pp. 109–165, Elsevier, 1989.
- [170] R. Ratcliff, “Connectionist models of recognition memory: constraints imposed by learning and forgetting functions.,” *Psychological review*, vol. 97, no. 2, p. 285, 1990.
- [171] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft, “Convergent learning: Do different neural networks learn the same representations?,” in *Feature Extraction: Modern Questions and Challenges*, pp. 196–212, PMLR, 2015.
- [172] F. Pernici, F. Bartoli, M. Bruni, and A. Del Bimbo, “Memory based online learning of deep representations from video streams,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [173] F. Pernici, M. Bruni, and A. Del Bimbo, “Self-supervised on-line cumulative learning from video streams,” *Computer Vision and Image Understanding*, p. 102983, 2020.
- [174] W. N. Price and I. G. Cohen, “Privacy in the age of medical big data,” *Nature medicine*, vol. 25, no. 1, pp. 37–43, 2019.
- [175] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, “Green ai,” *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.
- [176] A. Cossu, M. Ziosi, and V. Lomonaco, “Sustainable artificial intelligence through continual learning,” *arXiv preprint arXiv:2111.09437*, 2021.
- [177] J. D. Murray, A. Bernacchia, N. A. Roy, C. Constantinidis, R. Romo, and X.-J. Wang, “Stable population coding for working memory coexists with heterogeneous neural dynamics in prefrontal cortex,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 2, pp. 394–399, 2017.
- [178] E. M. Meyers, “Dynamic population coding and its relationship to working memory,” *Journal of Neurophysiology*, vol. 120, no. 5, pp. 2260–2268, 2018.
- [179] A. Libby and T. J. Buschman, “Rotational dynamics reduce interference between sensory and memory representations,” *Nature Neuroscience*, pp. 1–12, 2021.
- [180] A. Robins, “Catastrophic forgetting in neural networks: the role of rehearsal mechanisms,” in *Proceedings 1993 The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, pp. 65–68, IEEE, 1993.
- [181] F. Pernici, M. Bruni, C. Baecchi, and A. Del Bimbo, “Regular polytope networks,” *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [182] D. Lopez-Paz and M. Ranzato, “Gradient episodic memory for continuum learning,” vol. abs/1706.08840, 2017.
- [183] N. Díaz-Rodríguez, V. Lomonaco, D. Filliat, and D. Maltoni, “Don’t forget, there is more than forgetting: new metrics for continual learning,” *arXiv preprint arXiv:1810.13166*, 2018.





- [184] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, “Deep learning for content-based image retrieval: A comprehensive study,” in *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 157–166.
- [185] H. Azizpour, J. Sullivan, S. Carlsson, *et al.*, “Cnn features off-the-shelf: An astounding baseline for recognition,” in *CVPRW*, pp. 512–519, 2014.
- [186] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?,” *Advances in neural information processing systems*, vol. 27, 2014.
- [187] W. Chen, Y. Liu, W. Wang, E. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew, “Deep image retrieval: A survey,” *arXiv preprint arXiv:2101.11282*, 2021.
- [188] G. Toulas, R. Sircé, and H. Jégou, “Particular object retrieval with integral max-pooling of cnn activations,” in *ICLR 2016-International Conference on Learning Representations*, pp. 1–12, 2016.
- [189] J. Yue-Hei Ng, F. Yang, and L. S. Davis, “Exploiting local features from deep networks for image retrieval,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 53–61, 2015.
- [190] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of learning and motivation*, vol. 24, pp. 109–165, Elsevier, 1989.
- [191] M. Vijayan and S. Sridhar, “Continual learning for classification problems: A survey,” in *International Conference on Computational Intelligence in Data Science*, pp. 156–166, Springer, 2021.
- [192] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural Networks*, vol. 113, pp. 54–71, 2019.
- [193] E. Belouadah, A. Popescu, and I. Kanellos, “A comprehensive study of class incremental learning algorithms for visual tasks,” *Neural Networks*, vol. 135, pp. 38–54, 2021.
- [194] M. Davari and E. Belilovsky, “Probing representation forgetting in continual learning,” in *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.
- [195] W. Chen, Y. Liu, W. Wang, T. Tuytelaars, E. M. Bakker, and M. S. Lew, “On the exploration of incremental learning for fine-grained image retrieval,” in *BMVC*, BMVA Press, 2020.
- [196] N. Pu, W. Chen, Y. Liu, E. M. Bakker, and M. S. Lew, “Lifelong person re-identification via adaptive knowledge accumulation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7901–7910, 2021.
- [197] W. Chen, Y. Liu, N. Pu, W. Wang, L. Liu, and M. S. Lew, “Feature estimations based correlation distillation for incremental image retrieval,” *IEEE Transactions on Multimedia*, 2021.
- [198] Z. Li and D. Hoiem, “Learning without forgetting,” in *ECCV*, pp. 614–629, 2016.
- [199] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proc. ICML*, 2020.





- [200] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, “Deep metric learning via lifted structured feature embedding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.
- [201] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, “Multi-similarity loss with general pair weighting for deep metric learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [202] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [203] H. Jegou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 117–128, 2010.
- [204] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proc. NeurIPS*, 2014.
- [205] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2016.
- [206] Y. Shen, J. Gu, X. Tang, and B. Zhou, “Interpreting the latent space of gans for semantic face editing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9243–9252, 2020.
- [207] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.
- [208] A. Voynov and A. Babenko, “RPGAN: gans interpretability via random routing,” *CoRR*, vol. abs/1912.10920, 2019.
- [209] E. Denton, B. Hutchinson, M. Mitchell, and T. Gebru, “Detecting bias with generative counterfactual face attribute augmentation,” *arXiv preprint arXiv:1906.06439*, 2019.
- [210] C. Yang, Y. Shen, and B. Zhou, “Semantic hierarchy emerges in deep generative representations for scene synthesis,” *CoRR*, vol. abs/1911.09267, 2019.
- [211] D. Bau, J. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, “GAN dissection: Visualizing and understanding generative adversarial networks,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.
- [212] A. Voynov and A. Babenko, “Unsupervised discovery of interpretable directions in the GAN latent space,” in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, vol. 119 of *Proceedings of Machine Learning Research*, pp. 9786–9796, PMLR, 2020.
- [213] T. Xiao, J. Hong, and J. Ma, “Elegant: Exchanging latent encodings with gan for transferring multiple face attributes,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 168–184, 2018.





- [214] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola, “Ganalyze: Toward visual definitions of cognitive image properties,” in *ICCV*, pp. 5744–5753, 2019.
- [215] A. Jahanian, L. Chai, and P. Isola, “On the ”steerability” of generative adversarial networks,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.
- [216] N. Spingarn, R. Banner, and T. Michaeli, “GAN ”steerability” without optimization,” in *International Conference on Learning Representations*, 2021.
- [217] A. Plumerault, H. L. Borgne, and C. Hudelot, “Controlling generative models with continuous factors of variations,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.
- [218] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, “GANSpace: Discovering interpretable GAN controls,” *CoRR*, vol. abs/2004.02546, 2020.
- [219] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 8107–8116, IEEE, 2020.
- [220] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [221] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [222] Y. LeCun, “The mnist database of handwritten digits,” <http://yann.lecun.com/exdb/mnist/>, 1998.
- [223] Y. Jin, J. Zhang, M. Li, Y. Tian, and H. Zhu, “Towards the high-quality anime characters generation with generative adversarial networks,” in *Proceedings of the Machine Learning for Creativity and Design Workshop at NeurIPS*, 2017.
- [224] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.
- [225] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255, IEEE Computer Society, 2009.
- [226] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 3730–3738, IEEE Computer Society, 2015.
- [227] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, “S3fd: Single shot scale-invariant face detector,” in *Proceedings of the IEEE international conference on computer vision*, pp. 192–201, 2017.





- [228] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.
- [229] K. Kärkkäinen and J. Joo, “Fairface: Face attribute dataset for balanced race, gender, and age,” *arXiv preprint arXiv:1908.04913*, 2019.
- [230] B. Doosti, S. Naha, M. Mirbagheri, and D. J. Crandall, “Hope-net: A graph-based model for hand-object pose estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6608–6617, 2020.
- [231] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in *ICLR*, 2018.
- [232] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.
- [233] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, “Alias-free generative adversarial networks,” *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.
- [234] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” *ArXiv preprint*, vol. abs/1809.11096, 2018.
- [235] Y. Shen, Y. Xiong, W. Xia, and S. Soatto, “Towards backward-compatible representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [236] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, “GAN dissection: Visualizing and understanding generative adversarial networks,” in *ICLR*, 2019.
- [237] C. Yang, Y. Shen, and B. Zhou, “Semantic hierarchy emerges in deep generative representations for scene synthesis,” *IJCV*, vol. 129, no. 5, pp. 1451–1466, 2021.
- [238] Y. Shen and B. Zhou, “Closed-form factorization of latent semantics in GANs,” in *CVPR*, 2021.
- [239] Y. Shen, C. Yang, X. Tang, and B. Zhou, “InterFaceGAN: Interpreting the disentangled face representation learned by GANs,” 2020.
- [240] D. Bau, S. Liu, T. Wang, J.-Y. Zhu, and A. Torralba, “Rewriting a deep generative model,” in *ECCV*, pp. 351–369, Springer, 2020.
- [241] Z. Wu, D. Lischinski, and E. Shechtman, “StyleSpace analysis: Disentangled controls for stylegan image generation,” in *CVPR*, 2021.
- [242] E. Collins, R. Bala, B. Price, and S. Sússtrunk, “Editing in style: Uncovering the local semantics of GANs,” in *CVPR*, 2020.
- [243] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler, “Editgan: High-precision semantic image editing,” in *NeurIPS*, 2021.
- [244] Z. He, M. Kan, and S. Shan, “Eigengan: Layer-wise eigen-learning for gans,” in *ICCV*, pp. 14408–14417, 2021.





- [245] R. Haas, S. Graßhof, and S. S. Brandt, “Tensor-based subspace factorization for StyleGAN,” 2021.
- [246] R. Haas, S. Graßhof, and S. S. Brandt, “Tensor-based emotion editing in the StyleGAN latent space,” in *CVPRW*, 2022.
- [247] R. Wang, J. Chen, G. Yu, L. Sun, C. Yu, C. Gao, and N. Sang, “Attribute-specific Control Units in StyleGAN for Fine-grained Image Manipulation,” in *ACM MM*, Oct. 2021.
- [248] T. Broad, F. F. Leymarie, and M. Grierson, “Network bending: Expressive manipulation of generative models in multiple domains,” *Entropy*, 2022.
- [249] J. Zhu, R. Feng, Y. Shen, D. Zhao, Z. Zha, J. Zhou, and Q. Chen, “Low-Rank Subspaces in GANs,” in *NeurIPS*, 2021.
- [250] C. Zhang, Y. Xu, and Y. Shen, “Decorating your own bedroom: Locally controlling image generation with generative adversarial networks,” 2021.
- [251] O. Kafri, O. Patashnik, Y. Alaluf, and D. Cohen-Or, “StyleFusion: A generative model for disentangling spatial segments,” 2021.
- [252] J. Zhu, Y. Shen, Y. Xu, D. Zhao, and Q. Chen, “Region-based semantic factorization in GANs,” 2022.
- [253] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” in *Proceedings of ICLR*, 2018.
- [254] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.
- [255] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “Stargan v2: Diverse image synthesis for multiple domains,” in *CVPR*, pp. 8188–8197, 2020.
- [256] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop,” 2015.
- [257] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” *NeurIPS*, vol. 33, pp. 12104–12114, 2020.
- [258] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, “Laion-400m: Open dataset of clip-filtered 400 million image-text pairs,” 2021.
- [259] B. Custers, A. M. Sears, F. Dechesne, I. Georgieva, T. Tani, and S. Van der Hof, *EU personal data protection in policy and practice*. Springer, 2019.
- [260] J. Liu, B. Ni, Y. Yan, P. Zhou, S. Cheng, and J. Hu, “Pose transferrable person re-identification,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4099–4108, 2018.
- [261] M. Bishay, P. Palasek, S. Priebe, and I. Patras, “Schinet: Automatic estimation of symptoms of schizophrenia from facial behaviour analysis,” *IEEE Transactions on Affective Computing*, vol. 12, no. 4, pp. 949–961, 2021.





- [262] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” 2021.
- [263] M. Maximov, I. Elezi, and L. Leal-Taixé, “Ciagan: Conditional identity anonymization generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5447–5456, 2020.
- [264] H. Hukkelås, R. Mester, and F. Lindseth, “Deepprivacy: A generative adversarial network for face anonymization,” 2019.
- [265] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” 2014.
- [266] C. Tzelepis, J. Oldfield, G. Tzimiropoulos, and I. Patras, “Contraclip: Interpretable gan generation driven by pairs of contrasting sentences,” *arXiv preprint arXiv:2206.02104*, 2022.
- [267] J. Oldfield, M. Georgopoulos, Y. Panagakis, M. A. Nicolaou, and I. Patras, “Tensor component analysis for interpreting the latent space of gans,” *arXiv preprint arXiv:2111.11736*, 2021.
- [268] S. Bounareli, C. Tzelepis, V. Argyriou, I. Patras, and G. Tzimiropoulos, “Stylemask: Disentangling the style space of stylegan2 for neural face reenactment,” *arXiv preprint arXiv:2209.13375*, 2022.
- [269] T. Li and L. Lin, “Anonymousnet: Natural face de-identification with measurable privacy,” 2019.
- [270] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, “Fawkes: Protecting privacy against unauthorized deep learning models,” in *Proceedings of the 29th USENIX Security Symposium*, 2020.
- [271] D. Kollias, P. Tzirakis, M. A. Nicolaou, A. Papaioannou, G. Zhao, B. Schuller, I. Kotsia, and S. Zafeiriou, “Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond,” *International Journal of Computer Vision*, vol. 127, no. 6, pp. 907–929, 2019.
- [272] N. M. Foteinopoulou and I. Patras, “Learning from label relationships in human affect,” in *Proceedings of the 30th ACM International Conference on Multimedia*, MM ’22, (New York, NY, USA), p. 80–89, Association for Computing Machinery, 2022.
- [273] Y. Wu, F. Yang, and H. Ling, “Privacy-protective-gan for face de-identification,” 2018.
- [274] Y. Wen, B. Liu, M. Ding, R. Xie, and L. Song, “Identitydp: Differential private identification protection for face images,” *Neurocomputing*, vol. 501, pp. 197–211, 2022.
- [275] Y. Zheng, H. Yang, T. Zhang, J. Bao, D. Chen, Y. Huang, L. Yuan, D. Chen, M. Zeng, and F. Wen, “General facial representation learning in a visual-linguistic manner,” 2021.
- [276] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. P. Zafeiriou, “ArcFace: Additive angular margin loss for deep face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
- [277] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.







- [278] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.
- [279] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” *Advances in neural information processing systems*, vol. 30, 2017.
- [280] T. Fletcher, “Terse notes on riemannian geometry,” tech. rep., University of Utah, 2010.
- [281] F. Porikli, “Learning on manifolds,” in *Structural, Syntactic, and Statistical Pattern Recognition*, 2010.
- [282] S. Sommer and T. Fletcher, *Riemannian Geometric Statistics in Medical Image*, ch. Introduction to differential and Riemannian geometry. Elsevier, 2020.
- [283] S. r. Hauberg, O. Freifeld, and M. Black, “A geometric take on metric learning,” in *NeurIPS*, 2012.
- [284] S. Calinon, “Gaussians on riemannian manifolds: Applications for robot learning and adaptive control,” *IEEE Robotics & Automation Magazine*, 2020.
- [285] M. Bronstein, J. Bruna, and Y. LeCun, “Geometric deep learning: Going beyond euclidean data,” *IEEE Signal Processing Magazine*, 2017.
- [286] L. W. Tu, *An introduction to manifolds*. Springer, 2007.
- [287] G. Miranda, C. Thomaz, and G. Giraldi, “Geometric data analysis based on manifold learning with applications for image understanding,” in *SIBGRAPI*, 2017.
- [288] R. Chakraborty, J. Bouza, J. H. Manton, and B. C. Vemuri, “Manifoldnet: A deep neural network for manifold-valued data with applications,” *IEEE TPAMI*, 2022.
- [289] X. Zhen, R. Chakraborty, and N. Vogt, “Dilated convolutional neural networks for sequential manifold-valued data,” in *ICCV*, 2019.
- [290] J. J. Bouza, C.-H. Yang, D. E. Vaillancourt, and B. C. Vemuri, “Mvc-net: A convolutional neural network architecture for manifold-valued images with applications,” *ArXiv*, 2020.
- [291] A. Lou, I. Katsman, Q. Jiang, and S. Belongie, “Differentiating through the frechet mean,” in *ICML*, 2020.
- [292] R. Chakraborty, “Manifoldnorm: Extending normalizations on riemannian manifolds,” *ArXiv*, 2020.
- [293] A. Sim, M. L. Wiatrak, and A. Brayne, “Directed graph embeddings in pseudo-riemannian manifolds,” in *ICML*, 2021.
- [294] M. Cho and J. Lee, “Riemannian approach to batch normalization,” in *NeurIPS*, 2017.
- [295] M. Lezcano Casado, “Trivializations for gradient-based optimization on manifolds,” in *NeurIPS*, 2019.
- [296] F. Alimisis and A. Orvieto, “Momentum improves optimization on riemannian manifolds,” in *AISTATS*, 2021.





- [297] J. Li, F. Li, and S. Todorovic, "Efficient riemannian optimization on the stiefel manifold via the cayley transform.," in *AISTATS*, 2021.
- [298] G. Becigneul and O.-E. Ganea, "Riemannian adaptive optimization methods," in *ICLR*, 2019.
- [299] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2007.
- [300] H. Kasai, P. Jawanpuria, and B. Mishra, "Riemannian adaptive stochastic gradient algorithms on matrix manifolds," in *ICML*, 2019.
- [301] T. Bendokat, R. Zimmermann, and P. Absil, "A grassmann manifold handbook: Basic geometry and computational aspects," *ArXiv*, 2020.
- [302] S. R. Dubey, "A decade survey of content based image retrieval using deep learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.
- [303] W. Chen, Y. Liu, and W. Wang, "Deep learning for instance retrieval: A survey," *IEEE TPAMI*, 2022.
- [304] S. Bai, Z. Zhou, and J. Wang, "Ensemble diffusion for retrieval," in *ICCV*, 2017.
- [305] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *CVPR*, 2013.
- [306] A. Iscen, G. Toliás, and Y. Avrithis, "Mining on manifolds: Metric learning without labels," in *CVPR*, 2018.
- [307] N. Aziere and S. Todorovic, "Ensemble deep manifold similarity learning using hard proxies," in *CVPR*, 2019.
- [308] U. K. Dutta and C. Sekhar C., "A geometric approach for unsupervised similarity learning," in *ICASSP*, 2020.
- [309] V. Verma, A. Lamb, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *ICML*, 2019.
- [310] Y.-C. Su and K. Grauman, "Learning spherical convolution for 360° recognition," *IEEE TPAMI*, 2022.
- [311] P. Rodriguez, I. Laradji, and A. Drouin, "Embedding propagation: Smoother manifold for few-shot classification," in *ECCV*, 2020.
- [312] Z. Wang, Q. She, and T. Ward, "Generative adversarial networks in computer vision," *ACM Computing Surveys*, 2021.
- [313] F.-A. Croitoru and V. Hondru, "Diffusion models in vision: A survey," *ArXiv*, 2022.
- [314] H. Chung, B. Sim, and J. C. Ye, "Improving diffusion models for inverse problems using manifold constraints," 2022.
- [315] S. Barannikov, I. Trofimov, and G. Sotnikov, "Manifold topology divergence: a framework for comparing data manifolds," in *NeurIPS*, 2021.
- [316] X. Luo, Z. Han, and L. Yang, "Progressive attentional manifold alignment for arbitrary style transfer," in *ACCV*, 2022.





- [317] R. Parihar, A. Dhiman, and T. Karmali, “Everything is there in latent space: Attribute editing and attribute style manipulation by stylegan latent space exploration,” in *ACM MM*, 2022.
- [318] R. Wang, X.-J. Wu, Z. Chen, and T. Xu, “Dreamnet: A deep riemannian manifold network for spd matrix learning,” in *ACCV*, 2022.
- [319] S. C. Norman Joseph Tatro, “Unsupervised geometric disentanglement for surfaces via cfan-vae,” *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.
- [320] H. Ben-Hamu, S. Cohen, and J. Bose, “Matching normalizing flows and probability paths on manifolds,” in *ICML*, 2022.
- [321] J. Chen, Y. Yin, and T. Birdal, “Projective manifold gradient layer for deep rotation regression,” in *CVPR*, 2022.
- [322] L. Koestler and D. G. and, “Intrinsic neural fields: Learning functions on manifolds,” in *ECCV*, 2022.
- [323] L. Cayton, “Algorithms for manifold learning,” in *CoRR*, 2005.
- [324] Y. Wang, H. Huang, C. Rudin, and Y. Shaposhnik, “Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMap, and PaCMAP for data visualization,” *J. Mach. Learn. Res.*, 2022.
- [325] M. S. Sarfraz, M. Koulakis, C. Seibold, and R. Stiefelhagen, “Hierarchical nearest neighbor graph embedding for efficient dimensionality reduction,” in *CVPR*, 2022.
- [326] X. Zu and Q. Tao, “SpaceMAP: Visualizing high-dimensional data by space expansion,” in *ICML*, 2022.
- [327] A. Wagner, E. Solomon, and P. Bendich, “Improving metric dimensionality reduction with distributed topology,” *ArXiv*, vol. abs/2106.07613, 2021.
- [328] P. Tempczyk, R. Michaluk, and L. Garncarek, “LIDL: Local intrinsic dimension estimation using approximate likelihood,” in *ICML*, 2022.
- [329] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. G. Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt, “Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time,” in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, 2022.
- [330] J. Liu, A. Moreau, M. Preuss, J. Rapin, B. Rozière, F. Teytaud, and O. Teytaud, “Versatile black-box optimization,” in *GECCO ’20: Genetic and Evolutionary Computation Conference, Cancún Mexico, July 8-12, 2020*, pp. 620–628, ACM, 2020.
- [331] A. Kumar, A. Raghunathan, R. M. Jones, T. Ma, and P. Liang, “Fine-tuning can distort pre-trained features and underperform out-of-distribution,” in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022.





- [332] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 20-25 June 2009, Miami, Florida, USA, pp. 248–255, IEEE Computer Society, 2009.
- [333] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, “Do ImageNet classifiers generalize to ImageNet?,” in *International Conference on Machine Learning (ICML)*, 2019. <https://arxiv.org/abs/1902.10811>.
- [334] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer, “The many faces of robustness: A critical analysis of out-of-distribution generalization,” *International Conference on Computer Vision (ICCV)*, 2021.
- [335] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, “Learning robust global representations by penalizing local predictive power,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. <https://arxiv.org/abs/1905.13549>.
- [336] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz, “Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- [337] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, “Natural adversarial examples,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [338] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, and L. Schmidt, “Robust fine-tuning of zero-shot models,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2022.
- [339] L. Fan, W. Huang, C. Gan, J. Huang, and B. Gong, “Controllable image-to-video translation: A case study on facial expression generation,” in *Conf. on Artificial Intelligence (AAAI) Symposium on Educational Advances in Artificial Intelligence*, pp. 3510–3517, AAAI Press, 2019.
- [340] N. Otberdout, M. Daoudi, A. Kacem, L. Ballihi, and S. Berretti, “Dynamic facial expression generation on Hilbert hypersphere with conditional Wasserstein generative adversarial nets,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
- [341] C. Cao, Q. Hou, and K. Zhou, “Displaced dynamic expression regression for real-time facial tracking and animation,” *ACM Trans. on Graphics*, vol. 33, July 2014.
- [342] D. Cudeiro, T. Bolkart, C. Laidlaw, A. Ranjan, and M. J. Black, “Capture, learning, and synthesis of 3D speaking styles,” in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 10093–10103, 2019.
- [343] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, “Audio-driven facial animation by joint end-to-end learning of pose and emotion,” *ACM Trans. on Graphics*, vol. 36, July 2017.
- [344] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, “Generating 3D faces using convolutional mesh autoencoders,” in *European Conf. on Computer Vision (ECCV)*, pp. 725–741, 2018.
- [345] G. Bouritsas, S. Bokhnyak, S. Ploumpis, S. Zafeiriou, and M. Bronstein, “Neural 3D morphable models: Spiral convolutional networks for 3D shape representation learning and generation,” in *IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pp. 7212–7221, 2019.





- [346] D. Cosker, E. Krumhuber, and A. Hilton, “A faces valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling,” in *IEEE Int. Conf. on Computer Vision*, pp. 2296–2303, IEEE, 2011.
- [347] C. Ferrari, G. Lisanti, S. Berretti, and A. D. Bimbo, “A dictionary learning-based 3D morphable shape model,” *IEEE Trans. on Multimedia*, vol. 19, no. 12, pp. 2666–2679, 2017.
- [348] T. Li, T. Bolkart, M. Julian, H. Li, and J. Romero, “Learning a model of facial shape and expression from 4D scans,” *ACM Trans. on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, 2017.
- [349] C. Ferrari, S. Berretti, P. Pala, and A. Del Bimbo, “A sparse and locally coherent morphable face model for dense semantic correspondence across heterogeneous 3D faces,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2021.
- [350] R. A. Potamias, J. Zheng, S. Ploumpis, G. Bouritsas, E. Ververas, and S. Zafeiriou, “Learning to generate customized dynamic 3D facial expressions,” in *European Conf. on Computer Vision (ECCV)*, pp. 278–294, 2020.
- [351] N. Otterdout, C. Ferrari, M. Daoudi, S. Berretti, and A. Del Bimbo, “Sparse to dense dynamic 3d facial expression generation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20385–20394, June 2022.
- [352] N. Garcia and G. Vogiatzis, “How to read paintings: semantic art understanding with multi-modal retrieval,” in *Proc. of European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.
- [353] X. Ji, W. Wang, M. Zhang, and Y. Yang, “Cross-domain image retrieval with attention modeling,” in *Proc. of ACM Multimedia (ACMMM)*, pp. 1654–1662, 2017.
- [354] A. Qayyum, S. M. Anwar, M. Awais, and M. Majid, “Medical image retrieval using deep convolutional neural network,” *Neurocomputing*, vol. 266, pp. 8–20, 2017.
- [355] J. Ahmad, K. Muhammad, S. Bakshi, and S. W. Baik, “Object-oriented convolutional features for fine-grained image retrieval in large surveillance datasets,” *Future Generation Computer Systems*, vol. 81, pp. 314–330, 2018.
- [356] B. Ionescu, H. Müller, R. Péteri, Y. D. Cid, V. Liauchuk, V. Kovalev, D. Klimuk, A. Tarasau, A. B. Abacha, S. A. Hasan, *et al.*, “Imageclef 2019: Multimedia retrieval in medicine, lifelogging, security and nature,” in *Proc. of International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF)*, pp. 358–386, Springer, 2019.
- [357] F. Vaccaro, M. Bertini, T. Uricchio, and A. Del Bimbo, “Image retrieval using multi-scale cnn features pooling,” in *Proc. of ACM International Conference on Multimedia Retrieval (ICMR)*, ICMR ’20, (New York, NY, USA), pp. 311–315, Association for Computing Machinery, 2020.
- [358] I. Banerjee, C. Kurtz, A. E. Devorah, B. Do, D. L. Rubin, and C. F. Beaulieu, “Relevance feedback for enhancing content based image retrieval and automatic prediction of semantic image features: Application to bone tumor radiographs,” *Journal of biomedical informatics*, vol. 84, pp. 123–135, 2018.
- [359] H. Su, P. Wang, L. Liu, H. Li, Z. Li, and Y. Zhang, “Where to look and how to describe: Fashion image retrieval with an attentional heterogeneous bilinear network,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.





- [360] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision,” 2021.
- [361] H. Wu, Y. Gao, X. Guo, Z. Al-Halah, S. Rennie, K. Grauman, and R. Feris, “Fashion iq: A new dataset towards retrieving images by natural language feedback,” 2020.
- [362] Y. Rui, T. Huang, M. Ortega, and S. Mehrotra, “Relevance feedback: a power tool for interactive content-based image retrieval,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 644–655, 1998.
- [363] A. Kovashka, D. Parikh, and K. Grauman, “Whittlesearch: Interactive image search with relative attribute feedback,” *International Journal of Computer Vision*, vol. 115, pp. 185–210, Apr 2015.
- [364] B. Zhao, J. Feng, X. Wu, and S. Yan, “Memory-augmented attribute manipulation networks for interactive fashion search,” in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6156–6164, 2017.
- [365] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis, “Automatic spatially-aware fashion concept discovery,” 2017.
- [366] N. Vo, L. Jiang, C. Sun, K. Murphy, L.-J. Li, L. Fei-Fei, and J. Hays, “Composing text and image for image retrieval - an empirical odyssey,” 2018.
- [367] X. Guo, H. Wu, Y. Cheng, S. Rennie, G. Tesauro, and R. S. Feris, “Dialog-based interactive image retrieval,” 2018.
- [368] Y. Qu, P. Liu, W. Song, L. Liu, and M. Cheng, “A text generation and prediction system: Pre-training on new corpora using bert and gpt-2,” in *Proc. of IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 323–326, IEEE, 2020.
- [369] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165*, 2020.
- [370] S. Agarwal, G. Krueger, J. Clark, A. Radford, J. W. Kim, and M. Brundage, “Evaluating clip: Towards characterization of broader capabilities and downstream implications,” *arXiv preprint arXiv:2108.02818*, 2021.
- [371] M. V. Conde and K. Turgutlu, “Clip-art: Contrastive pre-training for fine-grained art classification,” in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3956–3960, 2021.
- [372] F. A. Galatolo, M. G. Cimino, and G. Vaglini, “Generating images from caption and vice versa via clip-guided generative latent space search,” *arXiv preprint arXiv:2102.01645*, 2021.
- [373] H. Fang, P. Xiong, L. Xu, and Y. Chen, “Clip2video: Mastering video-text retrieval via image clip,” *arXiv preprint arXiv:2106.11097*, 2021.
- [374] Y. Chen, S. Gong, and L. Bazzani, “Image search with text feedback by visiolinguistic attention learning,” in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.





- [375] M. Shin, Y. Cho, B. Ko, and G. Gu, “RTIC: Residual learning for text and image composition using graph convolutional network,” *arXiv preprint arXiv:2104.03015*, 2021.
- [376] S. Lee, D. Kim, and B. Han, “CoSMo: Content-style modulation for image retrieval with text feedback,” in *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 802–812, June 2021.
- [377] J. Kim, Y. Yu, H. Kim, and G. Kim, “Dual compositional learning in interactive image retrieval,” in *Proc. of AAAI Conference on Artificial Intelligence (AAAI)*, vol. 35, pp. 1771–1779, May 2021.
- [378] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
- [379] Y. Chen and L. Bazzani, *Learning Joint Visual Semantic Matching Embeddings for Language-Guided Retrieval*, pp. 136–152. 11 2020.
- [380] S. Jandial, A. Chopra, P. Badjatiya, P. Chawla, M. Sarkar, and B. Krishnamurthy, “Trace: Transform aggregate and compose visiolinguistic representations for image search with text feedback,” 2020.
- [381] Y. Yu, S. Lee, Y. Choi, and G. Kim, “Curlingnet: Compositional learning between images and text for fashion iq data,” 2020.
- [382] K. Sohn, “Improved deep metric learning with multi-class n-pair loss objective,” in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.
- [383] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, “No fuss distance metric learning using proxies,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 360–368, 2017.
- [384] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, “Deep metric learning via lifted structured feature embedding,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [385] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, “Improving the robustness of deep neural networks via stability training,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4480–4488, 2016.
- [386] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “Sphereface: Deep hypersphere embedding for face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.
- [387] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.
- [388] W. Chen, X. Chen, J. Zhang, and K. Huang, “Beyond triplet loss: a deep quadruplet network for person re-identification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 403–412, 2017.
- [389] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, “Joint detection and identification feature learning for person search,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3415–3424, 2017.





- [390] L. Qiao, Y. Shi, J. Li, Y. Wang, T. Huang, and Y. Tian, “Transductive episodic-wise adaptive metric for few-shot learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3603–3612, 2019.
- [391] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” *Advances in neural information processing systems*, vol. 30, 2017.
- [392] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.
- [393] A. El-Nouby, N. Neverova, I. Laptev, and H. Jégou, “Training vision transformers for image retrieval,” *arXiv preprint arXiv:2102.05644*, 2021.
- [394] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, October 2021.
- [395] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *ICLR*, 2021.
- [396] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *International Conference on Machine Learning*, pp. 10347–10357, PMLR, 2021.
- [397] R. Sarkar, “Low distortion delaunay embedding of trees in hyperbolic plane,” in *International Symposium on Graph Drawing*, pp. 355–366, Springer, 2011.
- [398] M. Nickel and D. Kiela, “Poincaré embeddings for learning hierarchical representations,” *Advances in neural information processing systems*, vol. 30, pp. 6338–6347, 2017.
- [399] S. Kim, D. Kim, M. Cho, and S. Kwak, “Proxy anchor loss for deep metric learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [400] K. Musgrave, S. Belongie, and S.-N. Lim, “A metric learning reality check,” in *European Conference on Computer Vision*, pp. 681–699, Springer, 2020.
- [401] A. Van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv e-prints*, pp. arXiv-1807, 2018.
- [402] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.
- [403] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.
- [404] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, “Caltech-UCSD Birds 200,” Tech. Rep. CNS-TR-2010-001, California Institute of Technology, 2010.







- [405] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, “3d object representations for fine-grained categorization,” in *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013.
- [406] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, “Sampling matters in deep embedding learning,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2840–2848, 2017.
- [407] A. Zhai and H.-Y. Wu, “Classification is a strong baseline for deep metric learning,” *arXiv preprint arXiv:1811.12649*, 2018.
- [408] K. Roth, B. Brattoli, and B. Ommer, “Mic: Mining interclass characteristics for improved metric learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8000–8009, 2019.
- [409] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [410] M. Opitz, G. Waltner, H. Possegger, and H. Bischof, “Deep metric learning with bier: Boosting independent embeddings robustly,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 2, pp. 276–290, 2018.
- [411] W. Kim, B. Goyal, K. Chawla, J. Lee, and K. Kwon, “Attention-based ensemble for deep metric learning,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 736–751, 2018.
- [412] Y. Suh, B. Han, W. Kim, and K. M. Lee, “Stochastic class-based hard example mining for deep metric learning,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7244–7252, 2019.
- [413] X. Wang, H. Zhang, W. Huang, and M. R. Scott, “Cross-batch memory for embedding learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6388–6397, 2020.
- [414] W. Ge, “Deep metric learning with hierarchical triplet loss,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 269–285, 2018.
- [415] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, “Multi-similarity loss with general pair weighting for deep metric learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5022–5030, 2019.
- [416] Q. Qian, L. Shang, B. Sun, J. Hu, H. Li, and R. Jin, “Softtriple loss: Deep metric learning without triplet sampling,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6450–6458, 2019.
- [417] P. Jacob, D. Picard, A. Histace, and E. Klein, “Metric learning with horde: High-order regularizer for deep embeddings,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6539–6548, 2019.
- [418] E. W. Teh, T. DeVries, and G. W. Taylor, “Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis,” in *European Conference on Computer Vision*, pp. 448–464, Springer, 2020.





- [419] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, “How to train your vit? data, augmentation, and regularization in vision transformers,” *arXiv preprint arXiv:2106.10270*, 2021.
- [420] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- [421] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, pp. 448–456, PMLR, 2015.
- [422] L. McInnes, J. Healy, N. Saul, and L. Grossberger, “Umap: Uniform manifold approximation and projection,” *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.
- [423] B. Neyshabur, H. Sedghi, and C. Zhang, “What is being transferred in transfer learning?,” *Advances in neural information processing systems*, vol. 33, pp. 512–523, 2020.
- [424] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “CNN features off-the-shelf: An astounding baseline for recognition,” in *Conference on Computer Vision and Pattern Recognition Workshop, CVPR-W*, 2014.
- [425] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *Proceedings of the IEEE international conference on computer vision*, pp. 843–852, 2017.
- [426] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International conference on machine learning*, pp. 647–655, PMLR, 2014.
- [427] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. ICLR*, 2015.
- [428] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [429] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, “Places: An image database for deep scene understanding,” *arXiv preprint arXiv:1610.02055*, 2016.
- [430] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, “Novel dataset for fine-grained image categorization: Stanford dogs,” in *Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC)*, vol. 2, Citeseer, 2011.
- [431] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories,” in *Computer vision and pattern recognition workshop*, p. 178, 2004.
- [432] A. Quattoni and A. Torralba, “Recognizing indoor scenes,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 413–420, IEEE, 2009.
- [433] L. Bossard, M. Guillaumin, and L. V. Gool, “Food-101—mining discriminative components with random forests,” in *European conference on computer vision*, pp. 446–461, Springer, 2014.





- [434] M.-E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in *Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, IEEE, 2008.
- [435] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar, “Cats and dogs,” in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505, 2012.
- [436] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, “Describing textures in the wild,” in *Conference on computer vision and pattern recognition*, 2014.
- [437] O. Silvén, M. Niskanen, and H. Kauppinen, “Wood inspection with non-supervised clustering,” *Machine Vision and Applications*, vol. 13, no. 5, pp. 275–285, 2003.
- [438] F. Parés, A. Arias-Duart, D. Garcia-Gasulla, G. Campo-Francés, N. Viladrich, E. Ayguadé, and J. Labarta, “The mame dataset: on the relevance of high resolution and variable shape image properties,” *Applied Intelligence*, pp. 1–22, 2022.
- [439] D. Garcia-Gasulla, A. Vilalta, F. Parés, E. Ayguadé, J. Labarta, U. Cortés, and T. Suzumura, “An out-of-the-box full-network embedding for convolutional neural networks,” in *IEEE International Conference on Big Knowledge (ICBK)*, 2018.
- [440] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Barambe, and L. Van Der Maaten, “Exploring the limits of weakly supervised pretraining,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 181–196, 2018.
- [441] A. Tormos, D. Garcia-Gasulla, V. Gimenez-Abalos, and S. Alvarez-Napagao, “When & How to transfer with Transfer Learning,” in *Has it Trained Yet? NeurIPS 2022 Workshop*, 2022.
- [442] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, 2015.
- [443] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *CVPR*, 2017.
- [444] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE TPAMI*, 2017.
- [445] Y.-H. Tsai, W.-C. Hung, S. Schuster, K. Sohn, M.-H. Yang, and M. Chandraker, “Learning to adapt structured output space for semantic segmentation,” in *CVPR*, 2018.
- [446] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, and F. Wen, “Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation,” in *CVPR*, 2021.
- [447] Z. Liu, Z. Miao, X. Pan, X. Zhan, D. Lin, S. X. Yu, and B. Gong, “Open compound domain adaptation,” in *Proc. CVPR*, 2020.
- [448] Y. Liu, W. Zhang, and J. Wang, “Source-free domain adaptation for semantic segmentation,” in *CVPR*, 2021.
- [449] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, “Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data,” in *ICCV*, 2019.





- [450] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, “Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation,” in *CVPR*, 2019.
- [451] K. Park, S. Woo, I. Shin, and I.-S. Kweon, “Discover, hallucinate, and adapt: Open compound domain adaptation for semantic segmentation,” in *NeurIPS*, 2020.
- [452] R. Gong, Y. Chen, D. P. Paudel, Y. Li, A. Chhatkuli, W. Li, D. Dai, and L. Van Gool, “Cluster, split, fuse, and update: Meta-learning for open compound domain adaptive semantic segmentation,” in *CVPR*, 2021.
- [453] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for data: Ground truth from computer games,” in *ECCV*, 2016.
- [454] J. N. Kundu, N. Venkat, R. V. Babu, *et al.*, “Universal source-free domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4544–4553, 2020.
- [455] P. T. S and F. Fleuret, “Uncertainty reduction for model adaptation in semantic segmentation,” in *CVPR*, 2021.
- [456] Y. Zou, Z. Yu, B. Kumar, and J. Wang, “Unsupervised domain adaptation for semantic segmentation via class-balanced self-training,” in *ECCV*, 2018.
- [457] X. Pan, P. Luo, J. Shi, and X. Tang, “Two at once: Enhancing learning and generalization capacities via ibn-net,” in *ECCV*, 2018.
- [458] Q. Lian, F. Lv, L. Duan, and B. Gong, “Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach,” in *ICCV*, 2019.
- [459] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “Bdd100k: A diverse driving dataset for heterogeneous multitask learning,” in *CVPR*, 2020.
- [460] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *CVPR*, 2016.
- [461] W. Chen, Z. Yu, Z. Wang, and A. Anandkumar, “Automated synthetic-to-real generalization,” in *ICML*, 2020.
- [462] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, “Domain generalization with mixstyle,” in *ICLR*, 2021.
- [463] Z. Tang, Y. Gao, Y. Zhu, Z. Zhang, M. Li, and D. Metaxas, “Selfnorm and crossnorm for out-of-distribution robustness,” in *ICCV*, 2021.
- [464] Y. Zhao, Z. Zhong, Z. Luo, G. H. Lee, and N. Sebe, “Source-free open compound domain adaptation in semantic segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 10, pp. 7019–7032, 2022.
- [465] P. Goyal, Q. Duval, J. Reizenstein, M. Leavitt, M. Xu, B. Lefaudeaux, M. Singh, V. Reis, M. Caron, P. Bojanowski, A. Joulin, and I. Misra, “Vissl,” *GitHub*. Note: <https://github.com/facebookresearch/vissl>, 2021.
- [466] I. Susmelj, M. Heller, P. Wirth, J. Prescott, and M. E. et al., “Lightly,” *GitHub*. Note: <https://github.com/lightly-ai/lightly>, 2020.





- [467] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *NeurIPS*, 2019.
- [468] P. L. D. Team, “Pytorch lightning,” *GitHub*. Note: <https://github.com/PyTorchLightning/pytorch-lightning>, vol. 3, 2019.
- [469] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2.” <https://github.com/facebookresearch/detectron2>, 2019.
- [470] J. Zbontar, L. Jing, I. Misra, Y. Lecun, and S. Deny, “Barlow twins: Self-supervised learning via redundancy reduction,” in *ICML*, 2021.
- [471] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, “Bootstrap your own latent - a new approach to self-supervised learning,” in *NeurIPS*, 2020.
- [472] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in *NeurIPS*, 2020.
- [473] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” *arXiv:2003.04297*, 2020.
- [474] D. Dwivedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, “With a little help from my friends: Nearest-neighbor contrastive learning of visual representations,” *arXiv:2104.14548*, 2021.
- [475] M. Zheng, S. You, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu, “Rssl: Relational self-supervised learning with weak augmentation,” *arXiv:2107.09282*, 2021.
- [476] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” in *Proc. NeurIPS*, 2020.
- [477] X. Chen and K. He, “Exploring simple siamese representation learning,” in *CVPR*, 2021.
- [478] A. Bardes, J. Ponce, and Y. LeCun, “Vicreg: Variance-invariance-covariance regularization for self-supervised learning,” *arXiv:2105.04906*, 2021.
- [479] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe, “Whitening for self-supervised representation learning,” in *ICML*, 2021.
- [480] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *ICLR*, 2021.
- [481] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” *International Conference on Computer Vision (ICCV)*, 2021.
- [482] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv:1802.03426*, 2020.
- [483] A. Krizhevsky, V. Nair, and G. Hinton, “Learning multiple layers of features from tiny images,” 2009.





- [484] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, 2009.
- [485] V. Turrisi da Costa, E. Fini, M. Nabi, N. Sebe, and E. Ricci, “solo-learn: A library of self-supervised methods for visual representation learning,” *Journal of Machine Learning Research*, vol. 23, no. 56, pp. 1–6, 2022.
- [486] G. Csurka, “A comprehensive survey on domain adaptation for visual applications,” *Domain Adaptation in Computer Vision Applications*, pp. 1–35, 2017.
- [487] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in *Proc. CVPR*, 2011.
- [488] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” *arXiv*, 2014.
- [489] M. Long and J. Wang, “Learning transferable features with deep adaptation networks,” in *Proc. ICML*, 2015.
- [490] B. Sun and K. Saenko, “Deep coral: Correlation alignment for deep domain adaptation,” in *Proc. ECCV*, 2016.
- [491] S. Roy, A. Siarohin, E. Sangineto, S. R. Buló, N. Sebe, and E. Ricci, “Unsupervised domain adaptation using feature-whitening and consensus loss,” *Proc. CVPR*, 2019.
- [492] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *JMLR*, 2016.
- [493] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, “Adversarial discriminative domain adaptation,” in *Proc. CVPR*, 2017.
- [494] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, “Cycada: Cycle-consistent adversarial domain adaptation,” in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1989–1998, 2018.
- [495] J. Tian, J. Zhang, W. Li, and D. Xu, “VDM-DA: Virtual domain modeling for source data-free domain adaptation,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.
- [496] R. Gomes, A. Krause, and P. Perona, “Discriminative clustering by regularized information maximization,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.
- [497] M. Hein, M. Andriushchenko, and J. Bitterwolf, “Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41–50, 2019.
- [498] T. Ringwald and R. Stiefelhagen, “Unsupervised domain adaptation by uncertain feature alignment,” *The British Machine Vision Conference (BMVC)*, 2020.
- [499] Y. Gal, J. Hron, and A. Kendall, “Concrete dropout,” in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3581–3590, 2017.
- [500] I. Osband, “Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout,” in *NeurIPS workshop on Bayesian deep learning*, 2016.





- [501] I. Osband, J. Aslanides, and A. Cassirer, “Randomized prior functions for deep reinforcement learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8617–8629, 2018.
- [502] A. Foong, D. Burt, Y. Li, and R. Turner, “On the expressiveness of approximate inference in bayesian neural networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 15897–15908, 2020.
- [503] H. Xia, H. Zhao, and Z. Ding, “Adaptive adversarial network for source-free domain adaptation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 9010–9019, 2021.
- [504] J. Liang, D. Hu, Y. Wang, R. He, and J. Feng, “Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [505] M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Conditional adversarial domain adaptation,” in *Proc. NeurIPS*, 2018.
- [506] R. Xu, G. Li, J. Yang, and L. Lin, “Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1426–1435, 2019.
- [507] S. Yang, Y. Wang, J. van de Weijer, L. Herranz, and S. Jui, “Generalized source-free domain adaptation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 8978–8987, 2021.
- [508] L. Tierney and J. B. Kadane, “Accurate approximations for posterior moments and marginal densities,” *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 82–86, 1986.
- [509] D. J. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
- [510] Q. Lao, X. Jiang, and M. Havaei, “Hypothesis disparity regularized mutual information maximization,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
- [511] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 213–226, 2010.
- [512] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, “Deep hashing network for unsupervised domain adaptation,” in *Proc. CVPR*, 2017.
- [513] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, “Visda: The visual domain adaptation challenge,” *arXiv preprint arXiv:1710.06924*, 2017.
- [514] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, “Moment matching for multi-source domain adaptation,” in *Proc. ICCV*, 2019.
- [515] P. Panareda Busto and J. Gall, “Open set domain adaptation,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 754–763, 2017.
- [516] A. Bendale and T. E. Boult, “Towards open set deep networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1563–1572, 2016.





- [517] H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang, “Separate to adapt: Open set domain adaptation via progressive separation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2927–2936, 2019.
- [518] X. Chen, S. Wang, M. Long, and J. Wang, “Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation,” in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1081–1090, 2019.
- [519] R. Li, Q. Jiao, W. Cao, H.-S. Wong, and S. Wu, “Model adaptation: Unsupervised domain adaptation without source data,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9641–9650, 2020.
- [520] S. Roy, M. Trapp, A. Pilzer, J. Kannala, N. Sebe, E. Ricci, and A. Solin, “Uncertainty-guided source-free domain adaptation,” in *European Conference on Computer Vision*, 2022.
- [521] W. Banzhaf, B. Baumgaertner, G. Beslon, R. Doursat, J. A. Foster, B. McMullin, V. V. De Melo, T. Miconi, L. Spector, S. Stepney, *et al.*, “Defining and simulating open-ended novelty: requirements, guidelines, and challenges,” *Theory in Biosciences*, vol. 135, no. 3, pp. 131–161, 2016.
- [522] D. Gravina, A. Khalifa, A. Liapis, J. Togelius, and G. N. Yannakakis, “Procedural content generation through quality diversity,” in *Proc. of the IEEE Conf. on Games*, 2019.
- [523] A. Liapis, H. P. Martínez, J. Togelius, and G. N. Yannakakis, “Transforming exploratory creativity with DeLeNoX,” in *Proc. of the Intl. Conf. on Computational Creativity*, pp. 56–63, 2013.
- [524] A. Hagg, S. Berns, A. Asteroth, S. Colton, and T. Bäck, “Expressivity of parameterized and data-driven representations in quality diversity search,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2021.
- [525] A. Nguyen, J. Yosinski, and J. Clune, “Understanding innovation engines: Automated creativity and improved stochastic optimization via deep learning,” *Evolutionary computation*, vol. 24, no. 3, pp. 545–572, 2016.
- [526] A. Cully, “Autonomous skill discovery with quality-diversity and unsupervised descriptors,” in *Proc. of the Genetic and Evolutionary Computation Conf.*, p. 81–89, 2019.
- [527] A. Gaier, A. Asteroth, and J.-B. Mouret, “Data-efficient design exploration through surrogate-assisted illumination,” *Evolutionary Computation*, vol. 26, no. 3, p. 381–410, 2018.
- [528] J. Lehman and K. O. Stanley, “Novelty search and the problem with objectives,” in *Genetic programming theory and practice IX*, pp. 37–56, Springer, 2011.
- [529] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.
- [530] Mojang, “*Minecraft*.” Game [PC], 2011.
- [531] S. M. Lucas and V. Volz, “Tile pattern kl-divergence for analysing and evolving game levels,” in *Proc. of the Genetic and Evolutionary Computation Conf.*, pp. 170–178, 2019.
- [532] T. Shu, J. Liu, and G. N. Yannakakis, “Experience-driven pcg via reinforcement learning: A super mario bros study,” in *Proc. of the IEEE Conf. on Games*, 2021.







- [533] A. Brightmoore, “AHousev5 MCEdit Filter.” Accessed 9 May 2022.
- [534] M. Barthet, A. Liapis, and G. N. Yannakakis, “Open-ended evolution for Minecraft building generation,” *IEEE Transactions on Games*, 2022. accepted.
- [535] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka, “Ceci n’est pas une pipe: A deep convolutional network for fine-art paintings classification,” in *Proceedings of the IEEE International Conference on Image Processing*, pp. 3703–3707, 2016.
- [536] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [537] J. Blank and K. Deb, “Pymoo: Multi-objective optimization in python,” *IEEE Access*, vol. 8, pp. 89497–89509, 2020.
- [538] G. Ritchie, “Some empirical criteria for attributing creativity to a computer program,” *Minds and Machines*, vol. 17, pp. 76–99, 2007.
- [539] A. Jordanous, “Four PPPerspectives on computational creativity in theory and in practice,” *Connection Science*, vol. 28, no. 2, pp. 194–216, 2016.
- [540] M. Zammit, A. Liapis, and G. N. Yannakakis, “Seeding diversity into AI Art,” in *Proceedings of the International Conference on Computational Creativity*, 2022.
- [541] L. Kocsis and C. Szepesvári, “Bandit Based Monte-Carlo Planning,” in *Machine Learning: ECML 2006* (J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, eds.), (Berlin, Heidelberg), pp. 282–293, Springer Berlin Heidelberg, 2006.
- [542] K. Sfikas, A. Liapis, and G. N. Yannakakis, “Monte carlo elites: Quality-diversity selection as a multi-armed bandit problem,” in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’21*, (New York, NY, USA), p. 180–188, Association for Computing Machinery, 2021.
- [543] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, “AugMix: A simple data processing method to improve robustness and uncertainty,” *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.
- [544] D. Falbel, *torchvision: Models, Datasets and Transformations for Images*, 2022. <https://torchvision.mlverse.org>, <https://github.com/mlverse/torchvision>.
- [545] D. Blei, A. Ng, and M. Jordan, “Latent dirichlet allocation,” vol. 3, pp. 601–608, 01 2001.
- [546] S. Xie and Z. Tu, “Holistically-nested edge detection,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1395–1403, 2015.
- [547] P. Machado, J. Romero, M. Nadal, A. Santos-del Riego, J. Correia, and A. Carballal, “Computerized measures of visual complexity,” *Acta Psychologica*, vol. 160, pp. 43–57, 2015.
- [548] D. Hasler and S. Suesstrunk, “Measuring colourfulness in natural images,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5007, pp. 87–95, 06 2003.
- [549] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.





- [550] J.-B. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *arXiv preprint arXiv:1504.04909*, 2015.
- [551] O. J. Dunn, “Multiple comparisons among means,” *Journal of the American Statistical Association*, vol. 56, pp. 52–64, 2012.
- [552] K. Sfikas, A. Liapis, and G. N. Yannakakis, “Controllable exploration of a design space via interactive quality diversity,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2023.
- [553] R. W. Picard, *Affective computing*. MIT press, 2000.
- [554] J. Fürnkranz and E. Hüllermeier, “Preference learning,” in *Encyclopedia of Machine Learning*, pp. 789–795, Springer, 2011.
- [555] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, “Learning to rank using gradient descent,” in *Proceedings of the 22nd international conference on Machine learning*, pp. 89–96, 2005.
- [556] L. Pang, Y. Lan, J. Guo, J. Xu, J. Xu, and X. Cheng, “Deeprank: A new deep architecture for relevance ranking in information retrieval,” in *Proceedings of the ACM Conference on Information and Knowledge Management*, pp. 257–266, 2017.
- [557] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.
- [558] D. Fair and B. Schlaggar, “Brain development,” in *Encyclopedia of Infant and Early Childhood Development*, pp. 211–225, Elsevier Inc., 2008.
- [559] D. Melhart, A. Liapis, and G. N. Yannakakis, “The affect game annotation (again) dataset,” *arXiv preprint arXiv:2104.02643*, 2021.
- [560] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*. Springer, 2018.
- [561] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2001.
- [562] S. Jastrzebski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey, “Three factors influencing minima in sgd,” *arXiv preprint arXiv:1711.04623*, 2017.
- [563] K. A. De Jong and W. M. Spears, “An analysis of the interacting roles of population size and crossover in genetic algorithms,” in *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, pp. 38–47, Springer, 1990.
- [564] S. Rylander and B. Gotshall, “Optimal population size and the genetic algorithm,” *Population*, vol. 100, no. 400, p. 900, 2002.
- [565] K. Makantasis, A. Liapis, and G. N. Yannakakis, “From pixels to affect: A study on games and player experience,” in *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, 2019.
- [566] K. Pinitas, K. Makantasis, A. Liapis, and G. N. Yannakakis, “Rankneat: Outperforming stochastic gradient search in preference learning tasks,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2022.



- [567] A. Moreo, A. Esuli, and F. Sebastiani, “QuaPy: A Python-based framework for quantification,” in *Proceedings of the 30th ACM International Conference on Knowledge Management (CIKM 2021)*, (Gold Coast, AU), pp. 4534–4543, 2021.
- [568] G. Forman, “Counting positives accurately despite inaccurate classification,” in *Proceedings of the 16th European Conference on Machine Learning (ECML 2005)*, (Porto, PT), pp. 564–575, 2005.
- [569] A. Bella, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana, “Quantification via probability estimators,” in *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM 2010)*, (Sydney, AU), pp. 737–742, 2010.
- [570] M. Saerens, P. Latinne, and C. Decaestecker, “Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure,” *Neural Computation*, vol. 14, no. 1, pp. 21–41, 2002.
- [571] J. L. Mueller and S. Siltanen, *Linear and nonlinear inverse problems with practical applications*, ch. 4: “Truncated singular value decomposition”, pp. 53–61. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2012.
- [572] G. Da San Martino, W. Gao, and F. Sebastiani, “Ordinal text quantification,” in *Proceedings of the 39th ACM Conference on Research and Development in Information Retrieval (SIGIR 2016)*, (Pisa, IT), pp. 937–940, 2016.
- [573] A. Esuli, “ISTI-CNR at SemEval-2016 Task 4: Quantification on an ordinal scale,” in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, (San Diego, US), pp. 92–95, 2016.
- [574] M. Bunse, A. Moreo, F. Sebastiani, and M. Senz, “Ordinal quantification through regularization,” in *Proceedings of the 33rd European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML / PKDD 2022)*, (Grenoble, FR), pp. 36–52, 2022.
- [575] M. Bunse, A. Moreo, F. Sebastiani, and M. Senz, “Ordinal quantification through regularization,” in *Presented at the LWDA Workshop on Knowledge Discovery, Data Mining and Machine Learning (LWDA 2022)*, (Hildesheim, DE), 2022.
- [576] A. Esuli and F. Sebastiani, “Optimizing text quantifiers for multivariate loss functions,” *ACM Transactions on Knowledge Discovery and Data*, vol. 9, no. 4, p. Article 27, 2015.
- [577] D. Card and N. A. Smith, “The importance of calibration for estimating proportions from annotations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2018)*, (New Orleans, US), pp. 1636–1646, 2018.
- [578] A. Esuli, A. Moreo, and F. Sebastiani, “A recurrent neural network for sentiment quantification,” in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018)*, (Torino, IT), pp. 1775–1778, 2018.
- [579] A. Maletzke, D. Moreira dos Reis, E. Cherman, and G. Batista, “DyS: A framework for mixture models in quantification,” in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, (Honolulu, US), pp. 4552–4560, 2019.



- [580] P. Pérez-Gállego, J. R. Quevedo, and J. J. del Coz, “Using ensembles for problems with characterizable changes in data distribution: A case study on quantification,” *Information Fusion*, vol. 34, pp. 87–100, 2017.
- [581] D. Moreira dos Reis, A. G. Maletzke, D. F. Silva, and G. E. Batista, “Classifying and counting with recurrent contexts,” in *Proceedings of the 24th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2018)*, (London, UK), pp. 1983–1992, 2018.
- [582] W. Hassan, A. G. Maletzke, and G. E. Batista, “Accurately quantifying a billion instances per second,” in *Proceedings of the 7th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2020)*, (Sydney, AU), pp. 1–10, 2020.
- [583] P. Pérez-Gállego, A. Castaño, J. R. Quevedo, and J. J. del Coz, “Dynamic ensemble selection for quantification tasks,” *Information Fusion*, vol. 45, pp. 1–15, 2019.
- [584] T. Schumacher, M. Strohmaier, and F. Lemmerich, “A comparative evaluation of quantification methods,” 2021. arXiv:2103.03223.
- [585] A. Fernandes Vaz, R. Izbicki, and R. Bassi Stern, “Quantification under prior probability shift: The ratio estimator and its extensions,” *Journal of Machine Learning Research*, vol. 20, pp. 79:1–79:33, 2019.
- [586] A. Moreo and F. Sebastiani, “Tweet sentiment quantification: An experimental re-evaluation,” *PLOS ONE*, vol. 17, pp. 1–23, September 2022.
- [587] D. H. Wolpert, “Stacked generalization,” *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.
- [588] P. Szymański and T. Kajdanowicz, “A network perspective on stratification of multi-label data,” in *Proceedings of the 1st International Workshop on Learning with Imbalanced Domains: Theory and Applications (LIDTA 2017)*, (Skopje, MK), pp. 22–35, 2017.
- [589] K. Sechidis, G. Tsoumakas, and I. Vlahavas, “On the stratification of multi-label data,” in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD 2011)*, (Athens, GR), pp. 145–158, 2011.
- [590] A. Moreo and F. Sebastiani, “Re-assessing the “classify and count” quantification method,” in *Proceedings of the 43rd European Conference on Information Retrieval (ECIR 2021)*, vol. II, (Lucca, IT), pp. 75–91, 2021.
- [591] R. Levin and H. Roitman, “Enhanced probabilistic classify and count methods for multi-label text quantification,” in *Proceedings of the 7th ACM International Conference on the Theory of Information Retrieval (ICTIR 2017)*, (Amsterdam, NL), pp. 229–232, 2017.
- [592] A. Moreo, M. Francisco, and F. Sebastiani, “Multi-label quantification,” *ACM Transactions on Knowledge Discovery and Data*, 2023.
- [593] A. Esuli, A. Fabris, A. Moreo, and F. Sebastiani, *Learning to quantify*. Cham, CH: Springer Nature, 2023.
- [594] A. Esuli, A. Moreo, and F. Sebastiani, “LeQua@CLEF2022: Learning to Quantify,” in *Proceedings of the 44th European Conference on Information Retrieval (ECIR 2022)*, (Stavanger, NO), pp. 374–381, 2022.





- [595] A. Esuli, A. Moreo, F. Sebastiani, and G. Sperduti, “A concise overview of LeQua 2022: Learning to quantify,” in *Proceedings of the 13th International Conference of the CLEF Association (CLEF 2022)*, (Bologna, IT), pp. 362–381, 2022.
- [596] A. Esuli, A. Moreo, F. Sebastiani, and G. Sperduti, “A detailed overview of LeQua 2022: Learning to quantify,” in *Working Notes of the 13th Conference and Labs of the Evaluation Forum (CLEF 2022)*, (Bologna, IT), 2022.
- [597] J. J. del Coz, P. González, A. Moreo, and F. Sebastiani, eds., *Proceedings of the 1st International Workshop on Learning to Quantify (LQ 2021)*. 2021.
- [598] J. J. del Coz, P. González, A. Moreo, and F. Sebastiani, “Learning to quantify: Methods and applications (LQ 2021),” in *Proceedings of the 30th ACM International Conference on Knowledge Management (CIKM 2021)*, (Gold Coast, AU), pp. 4874–4875, 2021.
- [599] J. J. del Coz, P. González, A. Moreo, and F. Sebastiani, “Report on the 1st International Workshop on Learning to Quantify (LQ 2021),” *SIGKDD Explorations*, vol. 24, no. 1, pp. 49–51, 2022.
- [600] J. J. del Coz, P. González, A. Moreo, and F. Sebastiani, eds., *Proceedings of the 2nd International Workshop on Learning to Quantify (LQ 2022)*. 2022.
- [601] A. Fabris, A. Esuli, A. Moreo, and F. Sebastiani, “Measuring fairness under unawareness of sensitive attributes: A quantification-based approach,” *Journal of Artificial Intelligence Research*, vol. 76, pp. 1117–1180, 2023.
- [602] O. Perez-Mon, A. Moreo, J. J. del Coz, and P. González, “Quantification using permutation-invariant networks based on histograms.” Unpublished manuscript, 2023.
- [603] A. Fabris, A. Esuli, A. Moreo, and F. Sebastiani, “Measuring fairness under unawareness of sensitive attributes: A quantification-based approach,” in *Presented at the 2nd ACM conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO 2022)*, (Arlington, US), 2022.

