AI4media

ARTIFICIAL INTELLIGENCE FOR THE MEDIA AND SOCIETY

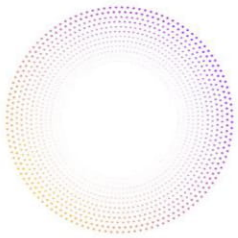
D4.7

Final toolset in robust, explainable, fair, and privacy-preserving AI

Project Title	AI4Media – A European Excellence Centre for Media, Society and Democracy
Contract No.	951911
Instrument	Research and Innovation Action
Thematic Priority	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
Start of Project	1 September 2020
Duration	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 www.ai4media.eu



Deliverable title	Final toolset in robust, explainable, fair, and privacy-preserving AI
Deliverable number	D4.7
Deliverable version	1.0
Previous version(s)	N/A
Contractual date of delivery	August 31st, 2024
Actual date of delivery	September 11th, 2024
Deliverable filename	AI4Media_D4.7_final.pdf
Nature of deliverable	Report
Dissemination level	Public
Number of pages	87
Work Package	WP4
Task(s)	T4.2, T4.3, T4.4, T4.5
Partner responsible	IBM
Author(s)	Anisa Halimi, Naoise Holohan, Giulio Zizzo, Mohamed Suliman (IBM), Hervé Le Borgne (CEA), Nicu Sebe, Marco Formentini (UNITN), Vasileios Mezaris, Evlampios Apostolidis, Konstantinos Tsigos (CERTH), Lorenzo Seidenari (UNIFI), Riccardo Fratti (HES-SO), Thomas Köllmer (FhG-IDMT), Frederic Precioso (UCA)
Editor(s)	Anisa Halimi and Naoise Holohan (IBM)
Project Officer	Evangelia Markidou

Abstract	This deliverable presents the third and final collection of technical work and outcomes from WP4 in AI4Media, focusing on Trustworthy AI. The document describes the continuing investigations and results of our work targeting four dimensions namely, (i) AI Robustness, (ii) AI Explainability, (iii) AI Privacy, and (iv) AI Fairness, each respectively corresponding to tasks T4.2, T4.3, T4.4, and T4.5. For each dimension, we present an overview of each partner's contribution and the methodology used, along with results, relevant publications, and software if available.
Keywords	AI, Trustworthy AI, Media, AI Robustness, Explainable AI, AI Privacy, AI Fairness

Copyright

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

Contributors

NAME	ORGANIZATION
Anisa Halimi	IBM
Naoise Holohan	IBM
Giulio Zizzo	IBM
Mohamed Suliman	IBM
Hervé Le Borgne	CEA
Nicu Sebe	UNITN
Marco Formentini	UNITN
Vasileios Mezaris	CERTH
Evlampios Apostolidis	CERTH
Konstantinos Tsigos	CERTH
Lorenzo Seidenari	UNIFI
Riccardo Fratti	HES-SO
Thomas Köllmer	FhG-IDMT
Frederic Precioso	UCA

Peer Reviews

NAME	ORGANIZATION
Cristian Stanciu	UNSTPB

Revision History

Version	Date	Reviewer	Modifications
0.1	May 29th, 2024	Anisa Halimi	First draft sent to partners for contributions
0.2	July 17th, 2024	Anisa Halimi, Naoise Holohan	Updated version with contributions from partners, sent for internal review
0.3	July 17th, 2024	Filareti Tsalakanidou	Updated version with review from Filareti Tsalakanido
0.4	August 1st, 2024	Cristian Stanciu	Updated version with review from Cristian Stanciu
0.5	August 23rd, 2024	Naoise Holohan, Anisa Halimi	Updated version with contributions from partners
0.6	September 11th, 2024	Naoise Holohan	Final version
1.0	September 11th, 2024	Filareti Tsalakanidou	Final version ready for submission

Table of Abbreviations and Acronyms

Abbreviation	Meaning
ADVTR	Adversarial Training
AE	Autoencoder
AI	Artificial Intelligence
API	Application Programming Interface
AUPRC	Area Under the Precision-Recall Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
CCM	Command Coherency Module
CIL	Conditional Imitation Learning
DNN	Deep Neural Network
DP	Differential Privacy
EO	Equalized Odds
EU	European Union
FedAvg	Federated Averaging
FedProx	Federated Proximal
FL	Federated Learning
FLaaS	Federated-Learning-as-a-Service
GDPR	General Data Protection Regulation
HE	Homomorphic Encryption
IL	Imitation Learning
IoT	Internet of Things
KL	Kullback–Leibler
LDM	Latent Diffusion Model
LIME	Local Interpretable Model-Agnostic Explanations
LLM	Large Language Model
LSTM	long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLaaS	Machine Learning as a Service
MSE	Mean Square Error
OOD	Out-Of-Distribution
PGD	Projected Gradient Descent
PKI	Public Key Infrastructure
RMSE	Root Mean Squared Error
SA	Secure Aggregation
SHAP	Shapley Values
SISA	Sharded, Isolated, Sliced, and Aggregated
SOTA	State of the Art

SVD	Singular Value Decomposition
T2I	Text-to-Image
ViT	Vision Transformer
VQA	Visual Question Answering
WM	Watermark
XAI	eXplainable AI

Contents

List of Tables

List of Figures

1 Executive Summary

This deliverable presents the research carried out as part of the technical tasks of Work Package 4 of the AI4Media project, entitled *Explainability, Robustness, and Privacy in AI*. These tasks, i.e., T4.2, T4.3, T4.4 and T4.5, cover the areas of AI Robustness, Explainability, Privacy, and Fairness respectively, and are accompanied by Tasks 4.1 and 4.6, which cover legal and benchmarking aspects that are not part of this deliverable. For each contribution in this report, we provide an overview of the work carried out, as well as references to the publications and software released by each partner.

This deliverable covers work carried out after the submission of D4.5 *"Intermediate toolset for robust, explainable, fair, and privacy-preserving AI"* in M36, and includes outcomes produced and finalised in the final 12 months of the project, from M37 (September 2023) to M48 (August 2024). As the final WP4 toolset of the project, it adds to the already considerable volume of work produced by all partners. While some partners' contributions to WP4 concluded with D4.5, this deliverable includes work from (in alphabetical order) **CEA**, **CERTH**, **FhG-IDMT**, **HES-SO**, **IBM**, **UCA**, **UNIFI**, and **UNITN**.

Introductory remarks are given in Section 2, covering an overview of the Trustworthy AI field (Section 2.1), the timeline of Work Package 4 (Section 2.2) and the structure of this document (Section 2.3).

A new contribution towards the **AI Robustness** task (T4.2) is detailed in Section 3. The work examines the effect of deploying multiple adversarial defences simultaneously, and the modifications that can be made to ensure the combination remains effective (Section 3.1).

Contributions towards the **AI Explainability** task (T4.3) are detailed in Section 4. This includes work on (i) explanations for automated video summarisation (Section 4.1), (ii) the benefits of using synthetic data for few-shot class-agnostic counting (being able to count objects in images regardless of class, Section 4.2), (iii) the risks posed by explainable AI models to the privacy and security of models and data (Section 4.3), (iv) concept discovery and dataset exploration with singular value decomposition matrix factorisation (Section 4.4), (v) examining the explainability of attention-based architectures, pinpointing the differences between post-hoc and attention-based explanations (Section 4.5), (vi) a novel approach to out-of-distribution detection using visual attention heatmaps (Section 4.6), and, (vii) addressing the limitations of imitation learning for autonomous driving (Section 4.7).

Contributions towards the **AI Privacy** task (T4.4) are detailed in Section 5. This includes work on (i) examining the true privacy benefits of federated learning, with reference to the strong trust models that are inherent to its present uses (Section 5.1), and (ii) securing federated learning using fully homomorphic encryption (Section 5.2).

Finally, contributions towards the **AI Fairness** task (T4.5) are detailed in Section 6. This includes work on (i) ensemble post-processing of LLMs to improve fairness (Section 6.1), and (ii) bias detection in text-to-image generative models (Section 6.2).

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

- 12 conference and workshop papers (AAAI/DAI '24, IEEE/ISM '22, WACV '24, PETS '24, ICLR '23, ICML '24, IEEE/CVF '23, IEEE/T-IV '23, ECML-PKDD/FLW '23, NeurIPS/SoLaR '23, CVPR '24, WACV '24); and
- 7 open-source software and tools that are openly shared (e.g., in GitHub).

2 Introduction

2.1 Trustworthy AI Overview

Artificial Intelligence (AI) holds significant importance in the European Union (EU) due to its potential to foster innovation, drive economic growth, improve public services, and shape social development. While AI offers immense opportunities and numerous benefits, there are also potential risks associated with its development such as security vulnerabilities, lack of transparency, privacy concerns, and bias and discrimination.

Trustworthy AI aims at developing and deploying Machine Learning (ML) technologies that are reliable, transparent, accountable, and aligned with the democratic and ethical values shared in our society. Trustworthy AI is typically divided into four broad dimensions: (i) **AI Robustness**, (ii) **AI Explainability**, (iii) **AI Privacy**, and (iv) **AI Fairness**.

AI Robustness focuses on detecting and mitigating adversarial attempts such as the introduction of misleading or malicious input to push an ML model towards making incorrect decisions or predictions. These attacks can be achieved through the use of adversarial samples in various data types (e.g., images, text, etc.) and across a broad range of model architectures.

Traditional ML models, such as deep neural networks, are inherently black boxes or operate in a black-box setting[1] so their decision-making processes are difficult to explain. The lack of interpretability and transparency in these models can lead to distrust and reluctance to adopt them, especially in critical applications (e.g., healthcare, finance) where decisions may have a significant impact on individuals. *AI Explainability* aims to provide users with transparency and understanding of how decisions are made by ML models.

AI Privacy focuses on designing and developing techniques to protect individuals' personal information including their sensitive information by maintaining its confidentiality and privacy. It also aims to prevent unauthorized access and misuse, as improper handling of personal information can result on unintended parties accessing individuals' sensitive information. Such sensitive information can then be used against the individuals for discrimination or blackmail. AI models are typically trained on a large amount of data, which in many cases contains sensitive information. Thus, AI Privacy aims to produce reliable ML models while ensuring that individuals' privacy is enhanced.

Finally, AI models can inadvertently learn biases from the data that they are trained on, reflecting and preserving biases and prejudice already present in our society. This can result in discriminatory treatment in various domains where AI is used such as mortgage lending, hiring, and criminal justice. *AI Fairness* aims to address these issues by developing AI models that treat individuals/groups fairly without favoring or disadvantaging any specific group/individual.

2.2 WP4 Timeline

This work package (WP4) is dedicated to Trustworthy AI. It involves 12 partner institutions, namely – AUTH, CEA, CERTH, FhG, HES-SO, IBM, IDIAP, KUL, UCA, UNITN, UNIFI, and UPB/UNSTPB – and runs throughout the entire duration of the AI4Media project (Figure 1). WP4 consists of 6 tasks organized as 4 vertical tasks, AI Robustness (Task 4.2), AI Explainability (Task 4.3), AI Privacy (Task 4.4), and AI Fairness (Task 4.5), and 2 horizontal tasks, focusing on the ethical and legal dimensions of AI within the European Union (Task 4.1) and benchmarking of AI systems (Task 4.6) (see Figure 2).

[1] A black-box setting refers to a scenario where the user has limited or no access to the internal workings of a model.

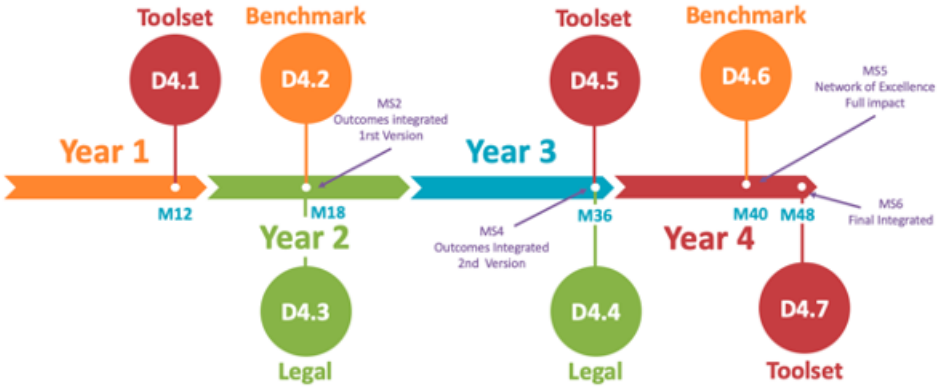


Figure 1. WP4 four-year timeline, showing the position of the present deliverable (D4.7) with reference to the lifetime of the AI4Media project.

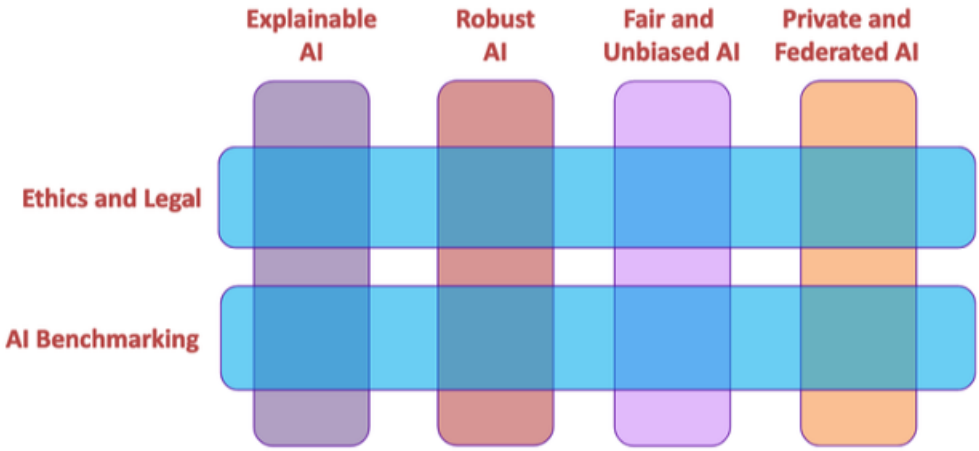


Figure 2. WP4 Tasks, comprising four vertical tasks (technical) and two horizontal tasks.

During the course of the project, this work package has produced 3 types of deliverables: (i) *toolset*, where the technical research output produced from the four vertical tasks will be reported, (ii) *legal*, where the output of the corresponding horizontal task will be reported, and (iii) *benchmark*, for the second horizontal task. This document consists of the third and final iteration of the toolset deliverable. Each iteration of this type of deliverable will report the contributions of the partners ranging from new algorithms accompanied by experimental results to toolset modules. In each iteration, we expect individual contributions to be at various stages of this pipeline as investigations mature.

The third and final iteration of this deliverable is an extension of the first two (D4.1 and D4.5). In this iteration, we present the research outputs that each partner achieved, as well as the outcomes of secondments conducted between M37 and M48, the final 12 months of the AI4Media project. This deliverable contains contributions within the dimensions of AI Robustness, AI Explainability,

AI Privacy, and AI Fairness.

2.3 Document Organisation

This deliverable follows the same structure as D4.5, with a similar structure for all the tasks to ensure a harmonized presentation of the algorithms/tools that were developed since D4.5. **Sections 3 - 6** describe the contributions towards each vertical task in Figure 2 (i.e., AI Robustness, AI Explainability, AI Privacy and AI Fairness), respectively. All sections follow the same structure, featuring a summary of each of the individual pieces of work from the work package partners. **Section 7** briefly presents activities related to the organisation of events on Trustworthy AI. Finally, **Section 8** concludes the deliverable, summarizing the final progress achieved as part of WP4.

3 AI Robustness (Task 4.2)

Machine Learning (ML) models are vulnerable to a variety of threat models [1], [2] in which adversarial samples play a critical role. Adversarial samples consist of inputs (images, texts, tabular data, etc.) deliberately crafted by an attacker in order to produce a desired response by the ML model, unintended by the model creators.

There are four broad types of adversarial threat models depending on how an attacker decides to exploit potential vulnerabilities in an ML model. (i) Poisoning attacks focus on the insertion of malicious data within the datasets used to train a model while (ii) Inference attacks intend to infer private information about a target model or the data used to train it. (iii) Evasion attacks, on the other hand, attempt to modify legitimate input samples in a manner that leads a model to misclassify it, while (iv) extraction attacks aim at extracting the parameters of a third party ML model so as to clone it.

In the following, we present one new contribution to the **AI Robustness** task which examines the effect of deploying multiple adversarial defences simultaneously, and the modifications that can be made to ensure the combination remains effective (Section 3.1).

3.1 Elevating Defenses: Bridging Adversarial Training and Watermarking for Model Resilience

Contributing partner: IBM

When models are deployed in the wild, they may be subject to multiple types of attacks simultaneously. Therefore, ML developers may wish to deploy multiple defensive techniques in parallel to protect against threats such as privacy, model stealing, evasion, or backdoor attacks.

However, many of these defences have been independently developed over time to tackle specific attacks, and their simultaneous deployment has generally not been a consideration in defence design. Initial works in this area started to study *which* defences conflicted with each other [3]. Conflicting defences have training or algorithmic objectives which oppose each other in a manner such that the resulting model is only weakly protected, or sufferers significant benign performance deterioration.

In the following sections, we describe our work *Elevating Defenses: Bridging Adversarial Training and Watermarking for Model Resilience*, with the publication full text being found in [4]. In this paper, we propose modifications to defence combinations such that their defensive properties are retained with minimal overhead. In particular, we tackle the problem of combining model watermarking methods with adversarial training.

3.1.1 Baseline

The baseline of combining adversarial training and model watermarking can be summarised below:

- Utilize the standard adversarial training procedure, where the data points are perturbed (with Projected Gradient Descent (PGD)) based on the specified perturbation budget (β) at every iteration.
- The Out-Of-Distribution (OOD) dataset (watermarking set) is provided by the model owner in advance.
- During the training phase, the watermarking set can be added to the training set, or the model can be separately trained on it at the end of every epoch.

Limitations: Both adversarial training and the watermarking, when applied individually, work effectively for the purpose they were designed for. However, when applied simultaneously, they have a conflicting interaction. The study [3] observed that baseline interaction has good performance with respect to the utility of the model; however, it affects the adversarial performance and decreases its robustness against evasion attacks. They attribute performance degradation as a result of using OOD watermarks, which use labels distinct from those in the actual training dataset, thus altering the model decision boundaries. As a result, this makes it easier for an evasion attack to identify a perturbation that causes incorrect results.

3.1.2 Proposed Technique

We propose to use watermarks generated via adversarial training, also known as adversarial watermarks. They are often used in the literature [5] owing to their high transferability to stolen ML models. The idea is to use watermarks that are distinct compared to the training samples, but have a similar distribution to our adversarial training dataset. In our case, the training dataset comprises original data samples and their respective perturbed adversarial samples. We cannot use watermarks with the same distribution as the training set because it would be difficult to differentiate them, and may provide a false sense of verification. Instead, we hypothesise that watermarks generated using an adversarial training technique will have a similar distribution, i.e., they share certain statistical properties as that of adversarial samples in the training set.

However, one may wonder if adversarial samples and adversarial watermarks will be too similar, and conflict at inference time. In fact, adversarial samples and adversarial watermarks differ in the way they are crafted. We propose to generate the watermarks using adversarial training with a higher perturbation budget than the adversarial samples. We claim that there exists a lower bound for an epsilon (ϵ-perturbation budget) with which adversarial training can be used effectively, after which the utility of the model degrades, making it ineffective. We leverage this knowledge to generate adversarial watermarks with a higher perturbation budget to differentiate them from the adversarial samples. In addition, we empirically observed, as was also reported in [6], that when we apply adversarial training to improve the robustness within some ϵ-neighbourhood, it exhibits effectiveness for $(\epsilon + \alpha)$- neighborhood, where α is a positive constant. Thus, we use β perturbation budget, where $\beta > \epsilon + \alpha$, to generate the adversarial watermarks.

The training process is similar to the baseline approach, but we substitute the OOD dataset with adversarial watermarks. These watermarks are derived from samples within the original training set, aligning with the distribution of adversarial samples, also generated from the training dataset. The main difference between adversarial samples and watermarks lies in the perturbation budget employed during their generation. While they exhibit similar statistical properties, they also remain unique.

3.1.3 Results

We used FMNIST dataset as OOD for MNIST, and vice-versa, however, there is no direct correlation. Any arbitrary data points can be chosen as watermarks provided they are out-of-distribution with respect to the training dataset

Table 2 illustrates the performance of the combined effect of deploying Adversarial Training (ADVTR) and watermarking techniques. We can observe that for both the approaches, the baseline (OOD Watermarks (WMs)), and our proposed strategy (adversarial watermarks) perform strongly in terms of the model utility and watermark verification. However, the adversarial accuracy when trained using OOD watermarks witnesses a drop of around 4% (from 92.82% $\rightarrow$ 88.39%) for the MNIST dataset and around 13% drop (from 70.95% $\rightarrow$ 57.75%) for the FMNIST dataset.

Dataset	ADVTR + WM (OOD)			ADVTR + WM (Adversarial)		
	Test Acc	Adv Acc	Water Acc	Test Acc	Adv Acc	Water Acc
MNIST	99.02	88.39	100	99.03	92.01	100
FMNIST	85.42	57.75	100	86.49	65.84	93

Table 2. Performance of the model with simultaneous deployment of adversarial training and model watermarking technique while using OOD and adversarial watermarks.

In our proposed strategy, where we use adversarial watermarks, we can see that it outperforms the baseline with respect to its robustness against evasion attacks. In terms of its adversarial accuracy, it has less than 1% drop (from 92.82% $\rightarrow$ 92.01%) for the MNIST dataset and around 4% drop (from 70.95% $\rightarrow$ 65.84%) for the FMNIST dataset. We attribute this slight decrease in robustness performance to an unintentional conflict that might arise concerning the interplay between the perturbation budget used to craft the adversarial samples and watermarks. However, the overall results obtained empirically support our hypothesis of using the watermarks with a similar distribution as that of adversarial samples to enhance the robustness of the model, while also maintaining comparable performance in terms of test and watermarking accuracy.

3.1.4 Robustness in Black-box Setting

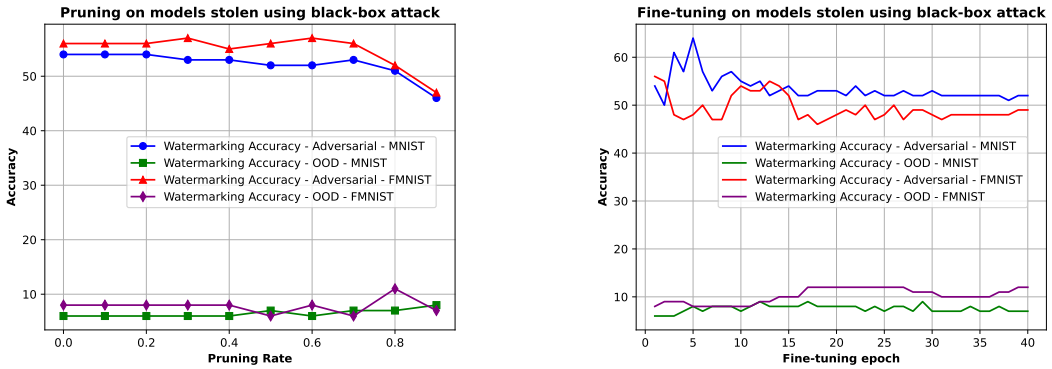
In this particular setting, our adversary has no direct access to the trained model or any other internal working. However, they can query the Application Programming Interface (API) to gain information about its performance of various inputs. For each query, we only output the class label of the input image predicted by our model and do not provide any information about the class logits. Finally, our attacker uses the information about the queried input-output pairs to train a duplicate model that has a identical test performance as that of our original model.

We examine the transferability of our watermarks to the model stolen using the black-box model stealing attack. As we can observe in Table 3, test accuracy is high for both datasets for both approaches. Moreover, one can notice that transferability for adversarial samples for both approaches is very low. We claim this is because, while launching a black-box attack, we had no information that the model was trained using adversarial training. Our adversary only queried the pure input-output pairs, thus limiting our model performance on the adversarial samples. However, we can notice a high transferability of our watermarks for the model which was trained using adversarial watermarks.

Black-box Transferability						
Dataset	AdvTraining + WM (OOD)			AdvTraining + WM (Adversarial)		
	Test Acc	Adv Acc	Water Acc	Test Acc	Adv Acc	Water Acc
MNIST	89.24	1.29	6	88.36	0.33	54
FMNIST	72.47	3.96	8	73.86	10.09	56

Table 3. Transferability of performance for various metrics when the models undergo black-box model stealing attack.

Furthermore, one can observe that even if the model was trained simultaneously using adversarial training, the adversarial watermarks did not conflict with the adversarial samples and were



(a) Pruning Attack *(b)* Fine Tuning Attack

Figure 3. Impact of removal attack on the model stolen using black-box setting

independently verified with high confidence. The results confirm our understanding regarding the conflict between adversarial watermarks and adversarial samples, and demonstrate the efficacy of our suggested interplay of the two techniques in a black-box setting.

With respect to Pruning. Figure 3a, plots the effect of pruning the stolen model with different pruning rates for two datasets. From Table 3, we know that the transferability of the OOD dataset is quite low, and thus, applying further removal attacks does not significantly affect its behavior. Further, one can observe that, with increasing pruning rate, our approach can still verify the ownership of the model with high confidence, i.e., for both the datasets, the models can be confidently verified with more than 50% transferability rate, with as high as 80% pruned neurons. This implies that the embedded watermarks significantly contribute to the important neurons of our model. Thus, they cannot be easily removed without degrading the model performance.

With respect to Fine-tuning. Figure 3b plots the effect of fine-tuning the stolen model with 40 epochs. The OOD watermarking accuracy by default is low due to its low transferability (Table 3). Thus fine-tuning it further does not give us any useful information. However, we can see that when we fine-tune the models trained using adversarial watermarks, the watermarking accuracy decreases to a certain point and then nearly stays constant throughout the remaining process. Although we observe a decrease in watermarking accuracy, it is still high enough (more than 45% for both datasets) to confidently verify the ownership of the models. The findings empirically show the effectiveness of our approach to pruning and fine-tuning attacks in the black-box setting.

3.1.5 Conclusions

This study introduced a novel way of combining adversarial watermarks and adversarial training without undermining its primary objectives. We observed that there exists a lower bound perturbation budget above which the utility of the model worsens, making it ineffective. We leverage this information to generate the adversarial watermarks that differ from the adversarial samples used in the training. We benchmark the performance of our strategy on various model stealing and removal attacks. Our proposed technique consistently outperforms the baseline in nearly all scenarios.

3.1.6 Relevant Resources and Publications

Relevant publications:

- J. Thakkar, G. Zizzo, and S. Maffeis. "Elevating Defenses: Bridging Adversarial Training and Watermarking for Model Resilience", Deployable AI workshop in conjunction with AAAI (DAI), 2024 [4].
  Arxiv record: `https://arxiv.org/pdf/2312.14260`.

3.1.7 Relevance to AI4Media use cases and media industry applications

Media companies will benefit from using model watermarking and adversarial training in their machine learning models due to the increasing reliance on AI for content creation, recommendation systems, and copyright enforcement. Model watermarking allows companies to embed identifiable information within their models, providing a means to prove ownership and combat unauthorized usage, which is crucial in protecting intellectual property in an industry heavily reliant on proprietary content. Adversarial training, on the other hand, enhances the robustness of machine learning models against malicious attacks that could distort recommendations or manipulate content. By employing these techniques, media companies can ensure the integrity and security of their AI systems, maintain consumer trust, and uphold their competitive edge in a rapidly evolving digital landscape.

4 AI Explainability (Task 4.3)

The last decade has seen a tremendous adoption of AI technology across a wide range of industries. AI has now become an indispensable part of our society. Accompanying this adoption however is an increasing concern about the opacity of such systems to human scrutiny. The reasons why such systems arrive at specific decisions are in most cases unknown to their users. In many cases, this opacity exists as well for the designers of such systems. This situation is thus one of the main obstacles that prevent the further adoption of AI technology across society today.

Explainable AI hence attempts to provide tools which enable the generation of explanations clarifying how a given model reached a decision and are understandable by humans. The methodologies and tools presented in this section hence address the need in the industry and society at large for AI models that can provide human understandable explanations of their underlying mechanisms.

Contributions towards the **AI Explainability** task (T4.3) include work on (i) explanations for automated video summarisation (Section 4.1), (ii) the benefits of using synthetic data for few-shot class-agnostic counting (being able to count objects in images regardless of class, Section 4.2), (iii) the risks posed by explainable AI models to the privacy and security of models and data (Section 4.3), (iv) concept discovery and dataset exploration with singular value decomposition matrix factorisation (Section 4.4), (v) examining the explainability of attention-based architectures, pinpointing the differences between post-hoc and attention-based explanations (Section 4.5), (vi) a novel approach to out-of-distribution detection using visual attention heatmaps (Section 4.6), and, (vii) addressing the limitations of imitation learning for autonomous driving (Section 4.7).

4.1 Explainable Video Summarization

Contributing partner: CERTH

4.1.1 Overview

The current practice in the Media industry for producing a video summary requires a professional video editor to watch the entire content and decide about the parts of it that should be included in the summary. This is a laborious task and can be very time-consuming in the case of long videos. Video summarization technologies aim to generate a short summary by selecting the most informative and important frames (key-frames) or fragments (key-fragments) of the full-length video, and presenting them in temporally-ordered fashion. The use of such technologies by media organizations can drastically reduce the needed resources for video summarization in terms of both time and human effort, and facilitate indexing, browsing, retrieval and promotion of their media assets [7]. Despite the recent advances in the field of video summarization, which are tightly associated with the emergence of modern deep learning network architectures [8], the outcome of a video summarization method still needs to be curated by a video editor, to make sure that all the necessary parts of the video were included in the summary. This content production step could be further facilitated if the video editor is provided with explanations about the suggestions made by the used video summarization technology. The provision of such explanations would allow the editor to progressively gain a better understanding of the reasoning behind the proposals of the used method, utilize it more effectively and thus reduce the needed time for content curation. In the context of Task 4.3, we developed an integrated framework for producing explanations about the outcomes of video summarization at different granularities, which is presented below.

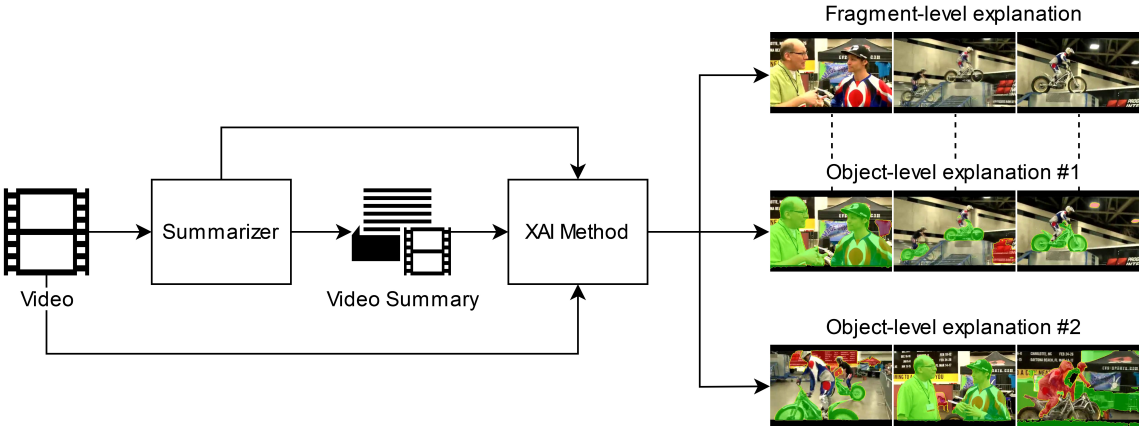


Figure 4. High-level overview of our framework for explaining video summarization.

4.1.2 Multi-Granular Explanation of Video Summarization

A high-level overview of the developed framework for multi-granular explainable video summarization is given in Fig. 4. Given an input video, a summarizer and the produced video summary (which in this case is formed by the three top-scoring video fragments by the summarizer), our framework produces three different types of explanations: i) a fragment-level explanation that indicates the temporal video fragments that influenced the most the decisions of the summarizer, ii) an object-level explanation that highlights the most influential visual objects within the aforementioned fragments (denoted as "object-level explanation #1" in Fig. 4), and iii) another object-level explanation that points out the visual objects within the fragments that have been selected for inclusion in the summary, that influenced the most this selection (denoted as "object-level explanation #2" in Fig. 4). In the core of this framework there is an XAI (explainable AI) method that is responsible for producing the explanation.

Fragment-level explanation: For fragment-level explanation, the input video needs to be temporally fragmented into consecutive and non-overlapping fragments. To perform this process, we employ a pre-trained model of the TransNetV2 method for shot segmentation from [9]. If the number of video fragments is equal to one (thus, the input video is a single-shot user-generated video) or less than ten (thus, the selection of three fragments for building the summary would not lead to a significantly condensed synopsis of the video), we further fragment the input video using the sub-shot segmentation method from [10]. The defined video fragments along with the input video, the summarizer and the produced video summary, are given as input to the XAI method. This method can be either model-agnostic (i.e., it does not require any knowledge about the summarization model) or model-specific (i.e., it utilizes information from the internal layers of the model). In our work, we considered the LIME explanation method from [11] and the best-performing configuration of the attention-based explanation method from [12], respectively. LIME [11] is a perturbation-based method that approximates the behavior of a model locally by generating a simpler, interpretable model. This method was designed for producing image-level explanations by masking out regions of the image; thus, we had to adapt it to operate over sequences of frames and produce fragment-level explanations. In particular, instead of masking out regions of a video frame during a perturbation, we mask out entire video fragments by replacing their frames with black frames. The perturbed version of the input video is fed to the summarizer, which then produces a new output (i.e., a new sequence of frame-level importance scores). This process

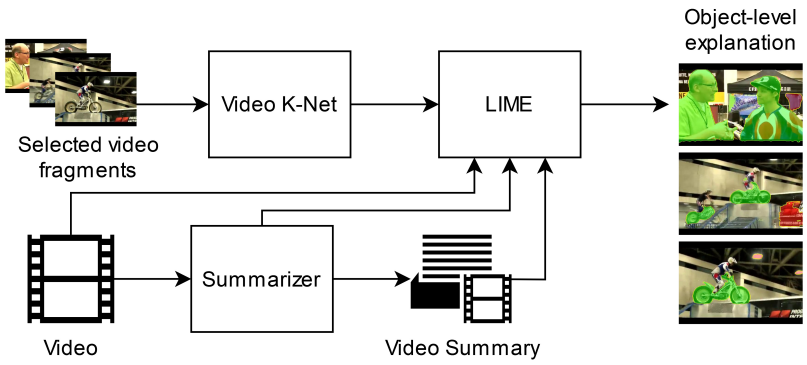


Figure 5. Processing pipeline for producing object-level explanations. The selected video fragments are the most influential according to the fragment-level explanation, or the top-scoring by the summarizer.

is repeated M times and the binary masks of each perturbation are fitted to the corresponding importance scores using a linear regressor. Finally, the fragment-level explanation is produced by focusing on the top-3 scoring fragments by this simpler model. The attention-based method of [12] can be applied on video summarization networks that model the frames' dependence using an attention mechanism [13]–[15]. This method uses the computed attention weights in the main diagonal of the attention matrix for a given input video, and forms an explanation signal by averaging them at the fragment level. The values of this explanation signal indicate the influence of the video's fragments in the output of the summarizer, and the fragments related to the top-3 scoring ones are selected to create the fragment-level explanation.

Object-level explanation: The processing pipeline for creating object-level explanations is shown in Fig. 5. The selected video fragments for creating such explanations can be either the most influential ones according to the fragment-level explanation, or the top-scoring ones by the summarizer, that were selected for inclusion in the video summary. The XAI method in this case is LIME [11], and the goal is to apply perturbations at the visual object level in order to identify the objects within the selected fragments, that influence the most the output of the summarizer. Once again, we use an adaptation of LIME, that takes into account the applied spatial perturbations in the visual content of a sequence of video frames (and not on a single frame). To make sure that a perturbation is applied on the same visual object(s) across the frames of a video fragment, we spatially segment these frames using a model of the Video K-Net method for video panoptic segmentation [16]. The top-scoring frame (by the summarizer) within a selected video fragment (by the fragment-level explanation or the summarizer) is picked as the keyframe. Once all the frames of this fragment have been spatially segmented by Video K-Net, the appearing visual objects in the selected keyframe are masked out across the entire video fragment through a series of perturbations that replace the associated pixels of the video frames with black pixels. The perturbed version of the input video after masking out a visual object in one of the selected video fragments is forwarded to the summarizer, which outputs a new sequence of frame-level importance scores. This process is repeated N times for a given video fragment and the binary masks of each perturbation are fitted to the corresponding importance scores using a linear regressor. Finally, the object-level explanation is formed by taking the top- and bottom-scoring visual objects by this simpler model, and highlighting the corresponding visual objects (using green and red coloured overlaying masks, respectively) in the keyframes of the selected video fragments.

4.1.2.1 Experimental Setup In our experiments we employ the SumMe [17] and TVSum [18] datasets, which are the most widely used ones in the literature for video summarization [8]. SumMe is composed of 25 videos with diverse video contents (e.g., covering holidays, events and sports), captured from both first-person and third-person view. TVSum contains 50 videos from 10 categories of the TRECVid MED task. To measure the influence of a selected video fragment or visual object by an explanation method, we mask it out (using black frames or pixels, respectively) and compute the difference in the summarization model's output, as $\Delta E(\boldsymbol{X}, \hat{\boldsymbol{X}}^{\boldsymbol{k}}) = \tau(\boldsymbol{y}, \boldsymbol{y}^{\boldsymbol{k}})$. In this formula, $\boldsymbol{X}$ is the set of original frame representations, $\hat{\boldsymbol{X}}^{\boldsymbol{k}}$ is the set of updated features of the frames belonging to the selected k^{th} video fragment (after the applied mask out process), $\boldsymbol{y}$ and $\boldsymbol{y}^{\boldsymbol{k}}$ are the outputs of the summarization model for $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}^{\boldsymbol{k}}$, respectively, and τ is the Kendall's τ correlation coefficient [19]. Based on ΔE, we assess the performance of each explanation using the following evaluation measures:

- **Discoverability+ (Disc+)** evaluates if the top-3 scoring fragments/objects by an explanation method have a significant influence to the model's output. For a given video, it is calculated by computing ΔE after perturbing (masking out) the top-1, top-2 and top-3 scoring fragments/objects in a one-by-one and sequential (batch) manner. The lower this measure is, the greater the ability of the explanation to spot the video fragments or visual objects with the highest influence to the summarization model.

- **Discoverability- (Disc-)** evaluates if the bottom-3 scoring fragments/objects by an explanation method have small influence to the model's output. For a given video, it is calculated by computing ΔE after perturbing (masking out) the bottom-1, bottom-2 and bottom-3 scoring fragments/objects in a one-by-one and sequential (batch) manner. The higher this measure is, the greater the effectiveness of the explanation to spot the video fragments or visual objects with the lowest influence to the summarization model.

- **Sanity Violation (SV)** quantifies the ability of explanations to correctly discriminate the most from the least influential video fragments or visual objects. It is calculated by counting the number of cases where the condition (Disc+ > Disc-) is violated, after perturbing (masking out) parts of the input corresponding to fragments/objects with the three highest and lowest explanation scores in a one-by-one and sequential (batch) manner, and then expressing the computed value as a fraction of the total number of perturbations. This measure ranges in $[0, 1]$; the closest its value is to zero, the greater the reliability of the explanation signal.

The number of applied perturbations M for producing fragment-level explanations was set equal to 20.000. The number of applied perturbations N for producing object-level explanations was set equal to 2.000. The number of video fragments for producing explanations (both at the fragment and the object level) was set equal to three. For video summarization, we use models of the CA-SUM method [13] trained on the SumMe and TVSum datasets. For further implementation details, we refer the reader to the relevant paper (see the last one in Section 4.1.3).

4.1.2.2 Quantitative Results The results about the performance of the examined explanation methods on the videos of the SumMe and TVSum datasets, are presented in Tables 4-9. In each case, the top part shows the computed Disc+/- and SV scores for videos that have at least one top- and one bottom-scoring fragment (or visual object) by the explanation method, while the bottom part shows the computed scores for videos that have at least three top- and three bottom-scoring fragments (or visual object) by the explanation method. The best scores are shown in bold and the arrows indicate the optimal (lower or higher) value for each evaluation measure. For the sake of space, we show the top- and bottom-k scoring fragment (with k = 1, 2, 3) in the same cell.

Table 4. Performance of fragment-level explanation methods on the SumMe dataset.

		Disc+ (↓)	Disc+ Seq (↓)	Disc- (↑)	Disc- Seq (↑)	SV (↓)	SV Seq (↓)
Top/Bottom-1	Attention	**0.568**	-	**0.971**	-	**0.063**	-
	LIME	0.747	-	0.886	-	0.438	-
Top/Bottom-1	Attention	**0.617**	-	**0.951**	-	**0.000**	-
	LIME	0.879	-	0.802	-	0.600	-
Top/Bottom-2	Attention	**0.888**	**0.546**	**0.980**	**0.930**	**0.400**	**0.200**
	LIME	0.891	0.785	0.966	0.759	**0.400**	0.600
Top/Bottom-3	Attention	0.967	**0.547**	**0.955**	**0.886**	**0.400**	**0.400**
	LIME	**0.945**	0.750	0.918	0.658	0.600	0.600

Table 5. Performance of fragment-level explanation methods on the TVSum dataset.

		Disc+ (↓)	Disc+ Seq (↓)	Disc- (↑)	Disc- Seq (↑)	SV (↓)	SV Seq (↓)
Top/Bottom-1	Attention	**0.579**	-	**0.983**	-	**0.000**	-
	LIME	0.798	-	0.952	-	0.298	-
Top/Bottom-1	Attention	**0.561**	-	**0.984**	-	**0.000**	-
	LIME	0.795	-	0.940	-	0.308	-
Top/Bottom-2	Attention	0.967	**0.519**	**0.990**	**0.963**	0.333	**0.000**
	LIME	**0.909**	0.696	0.954	0.875	**0.308**	0.282
Top/Bottom-3	Attention	0.964	**0.483**	**0.982**	**0.943**	**0.333**	**0.026**
	LIME	**0.960**	0.618	0.969	0.834	0.461	0.333

Concerning fragment-level explanation, the results in Tables 4 and 5 show that the attention-based method performs clearly better compared to LIME, in most evaluation settings. The produced fragment-level explanations by this method are more capable to spot the most influential video fragment, while the competitiveness of this method is more pronounced when more than one video fragments are taken into account (see columns "Disc+ Seq" and "Disc- Seq"). Moreover, the produced fragment-level explanations are clearly more effective in discriminating the most from the least influential fragments of the video, as indicated by the significantly lower SV scores in all settings (see columns "SV" and "SV Seq").

The performance of the developed method for object-level explanation is initially evaluated using video fragments that were found as the most influential ones by the considered fragment-level explanation methods. The results of our evaluations, shown in Tables 6 and 7, demonstrate that the object-level explanations for the selected video fragments by the two different explanation methods exhibit comparable performance. In general, the LIME-based fragments allow the object-level explanation method to be a bit more effective when spotting the most influential visual objects, while the attention-based fragments lead to better performance when spotting the visual objects with the least influence on the model's output. The comparable capacity of the fragment-level explanation methods is also shown from the mostly similar SV scores. A difference is observed when the applied perturbations affect more than one visual objects, where the produced object-level explanations using the attention-based fragments are associated with clearly lower SV scores. Therefore, a choice between the fragment-level explanation methods could be made based on the level of details in the obtained object-level explanation.

The performance of the developed object-level explanation method on the videos of the SumMe and TVSum datasets when using the three top-scoring fragments by the summarization method,

Table 6. Performance of the object-level explanation method on the SumMe dataset using the selected video fragments by the attention-based and LIME explanation methods.

		Disc+ (↓)	Disc+ Seq (↓)	Disc- (↑)	Disc- Seq (↑)	SV (↓)	SV Seq (↓)
Top/Bottom-1	Attention	0.969	-	**0.949**	-	0.694	-
	LIME	**0.941**	-	0.910	-	**0.603**	-
Top/Bottom-1	Attention	0.976	-	**0.963**	-	**0.639**	-
	LIME	**0.937**	-	0.878	-	0.666	-
Top/Bottom-2	Attention	0.988	0.968	**0.981**	**0.958**	**0.555**	**0.639**
	LIME	**0.962**	**0.915**	0.921	0.839	0.833	0.750
Top/Bottom-3	Attention	0.994	0.962	**0.989**	**0.952**	0.750	**0.555**
	LIME	**0.959**	**0.897**	0.956	0.828	**0.611**	0.805

Table 7. Performance of the object-level explanation method on the TVSum dataset using the selected video fragments by the attention-based and LIME explanation methods.

		Disc+ (↓)	Disc+ Seq (↓)	Disc- (↑)	Disc- Seq (↑)	SV (↓)	SV Seq (↓)
Top/Bottom-1	Attention	0.954	-	**0.989**	-	0.211	-
	LIME	**0.949**	-	0.987	-	**0.162**	-
Top/Bottom-1	Attention	0.940	-	**0.981**	-	**0.277**	-
	LIME	**0.908**	-	0.962	-	0.444	-
Top/Bottom-2	Attention	0.956	**0.908**	**0.995**	**0.980**	**0.111**	**0.111**
	LIME	**0.948**	0.909	0.968	0.907	0.277	0.611
Top/Bottom-3	Attention	0.990	0.889	**0.998**	**0.978**	**0.111**	**0.000**
	LIME	**0.961**	**0.879**	0.996	0.907	**0.111**	0.500

is reported in Tables 8 and 9, respectively. A pair-wise comparison of the Disc+ and Disc- scores shows that our method distinguishes the most from the least influential object in most cases, a fact that is also documented by the obtained SV scores. Moreover, it is able to spot objects that have indeed a very small impact on the output of the summarization process, as demonstrated by the significantly high Disc- scores. Finally, a cross-dataset comparison shows that our method is more effective on the TVSum videos, as it exhibits constantly lower SV scores for both evaluation settings (one-by-one and sequential).

4.1.2.3 Qualitative Results The top part of Fig. 6 provides a keyframe-based representation of the visual content of the original and summarized version of a TVSum video, titled "Smage Bros. Motorcycle Stunt Show", while the bottom part shows the produced explanations by the proposed framework. The green- and red-coloured regions in the frames of the object-level explanations, indicate the most and least influential visual objects, respectively (also shown in segmentation masks, right below). In this example, the created video summary shows the riders of the motorcycles and one of them being interviewed. The obtained fragment-level explanation from the employed method indicates that the summarizer concentrates on the riders (2nd and 3rd fragment) and the interview (1st fragment). Further insights are given by the object-level explanation of the aforementioned fragments, which demonstrates that the motorcycles (2nd and 3rd fragment) and the participants in the interview (1st fragment) were the most influential visual objects. Similar remarks can be made by observing the produced object-level explanation using the selected fragments from the summarizer (see 1st and 2nd fragment). These findings explain why the summarizer selected these

Table 8. Performance of the object-level explanation method on the SumMe dataset using the selected video fragments by the summarization method.

	Disc+ (↓)	Disc+ Seq (↓)	Disc- (↑)	Disc- Seq (↑)	SV (↓)	SV Seq (↓)
Top/Bottom-1	0.894	-	0.990	-	0.397	-
Top/Bottom-1	0.769	-	0.977	-	0.357	-
Top/Bottom-2	0.985	0.692	0.995	0.912	0.365	0.516
Top/Bottom-3	0.999	0.881	0.994	0.715	0.484	0.476

Table 9. Performance of the object-level explanation method on the TVSum dataset using the selected video fragments by the summarization method.

	Disc+ (↓)	Disc+ Seq (↓)	Disc- (↑)	Disc- Seq (↑)	SV (↓)	SV Seq (↓)
Top/Bottom-1	0.772	-	0.996	-	0.195	-
Top/Bottom-1	0.883	-	0.879	-	0.255	-
Top/Bottom-2	0.655	0.506	0.997	0.832	0.222	0.155
Top/Bottom-3	0.964	-0.184	0.999	0.841	0.344	0.133

parts of the video for inclusion in the summary and why other parts (showing the logo of the TV-show, distant views of the scene and close-ups of the riders) where found as less appropriate. This paradigm shows that the produced explanations could deliver insights about the focus of the summarization model, and thus, assist the explanation of the video summarization outcome.

4.1.3 Relevant Resources and Publications

Relevant publications:

- E. Apostolidis, G. Balaouras, V. Mezaris, I. Patras, "Explaining Video Summarization Based on the Focus of Attention", Proc. IEEE Int. Symposium on Multimedia (ISM), Naples, Italy, pp. 146-150, Dec. 2022. DOI:10.1109/ISM55400.2022.00029. [12].
  Zenodo record: `https://zenodo.org/records/7573492`.
- E. Apostolidis, V. Mezaris, I. Patras, "A Study on the Use of Attention for Explaining Video Summarization", Proc. NarSUM workshop at ACM Multimedia 2023 (ACM MM), Ottawa, Canada, Oct.-Nov. 2023. DOI:10.1145/3607540.3617138. [20].
  Zenodo record: `https://zenodo.org/records/10184460`.
- E. Apostolidis, G. Balaouras, I. Patras, V. Mezaris, "Explainable Video Summarization for Advancing Media Content Production", Encyclopedia of Information Science and Technology, Sixth Edition, IGI Global, 2023. DOI:10.4018/978-1-6684-7366-5.ch065. [7].
  Zenodo record: `https://zenodo.org/records/10039722`.
- K. Tsigos, E. Apostolidis, V. Mezaris, "An Integrated Framework for Multi-Granular Explanation of Video Summarization", arXiv, May 2024, arXiv:2405.10082 (under review) [21].

Relevant software and/or external resources:

- The PyTorch implementation of our work on attention-based explanation of video summarization can be found in `https://github.com/e-apostolidis/XAI-SUM`.

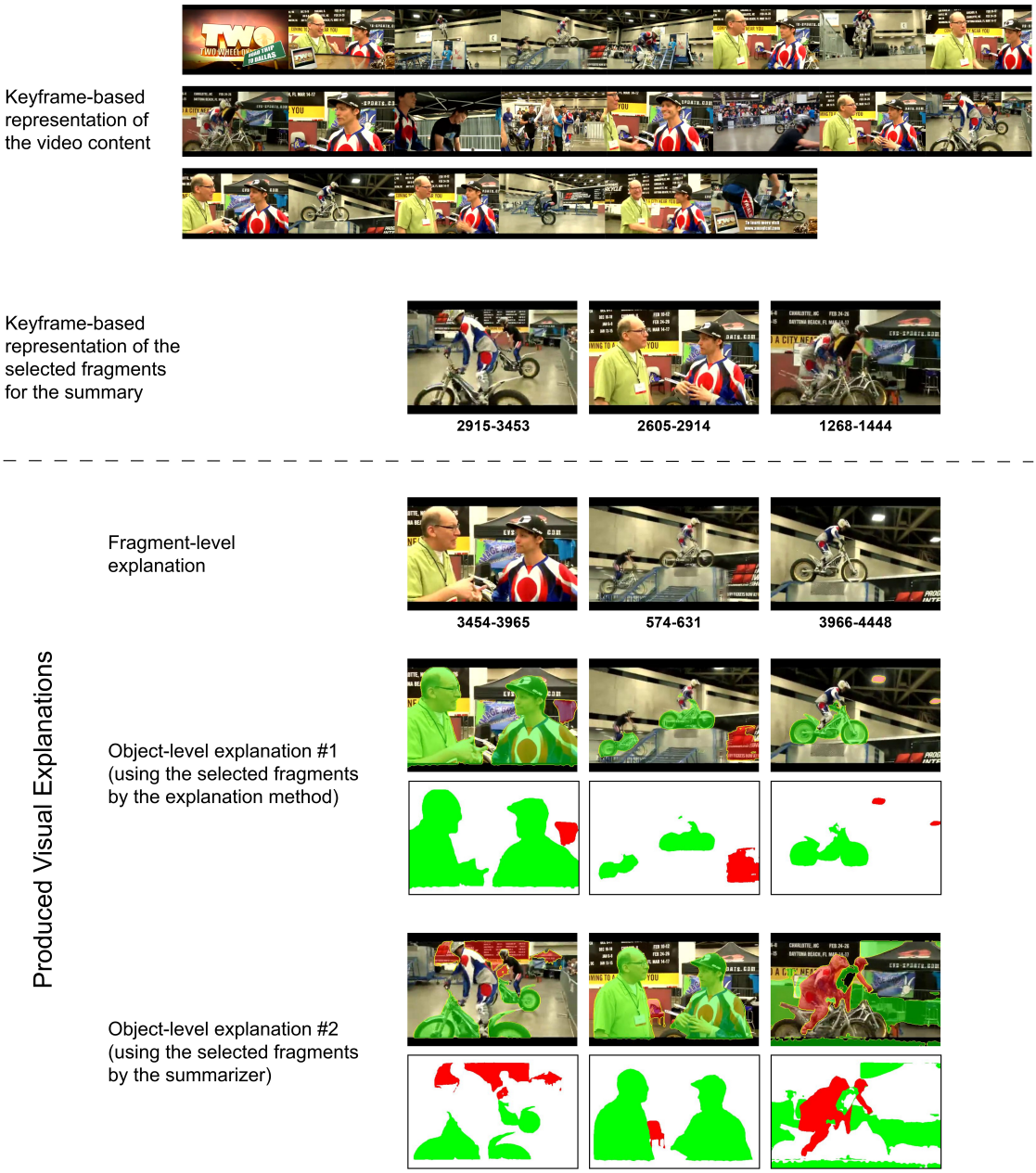


Figure 6. Top part: a keyframe-based representation of the original and the summarized version of a TVSum video, titled "Smage Bros. Motorcycle Stunt Show". Bottom part: the produced explanations by our framework. Green- and red-coloured regions indicate the most and least influential visual objects, respectively.

4.1.4 Relevance to AI4Media use cases and media industry applications

The developed framework can facilitate the explanation of AI methods for video summarization. Given the broad use of these methods in several use cases of AI4Media, their output will help to: (i) better assist the summarization of the developed news stories by indicating the parts of

the video that affected the most the suggestions of an AI-based video summarizer concerning the parts that should be included in the summary (Use Case 2: AI for News - The Smart News Assistant), (ii) support the production of summarized versions of a given video (e.g. according to the needs of the targeted audiences), by providing explanations about the summarization outcome and facilitating content curation (Use Case 3: AI in Vision - High Quality Video Production & Content Automation), and (iii) advance both the re-organization of media collections and the content moderation, by associating summarized versions of video items with human-interpretable visual explanations (Use Case 7: AI for (Re-)organisation and Content Moderation).

4.2 Semantic Generative Augmentations for Few-Shot Counting

Contributing partner: CEA

The work presented in this section was undertaken as part of a secondment under the AI4Media Junior Fellows Exchange programme

With the availability of powerful text-to-image diffusion models, recent works have explored the use of synthetic data to improve image classification performances. These works show that it can effectively augment or even replace real data. In this work, we investigate how synthetic data can benefit few-shot class-agnostic counting. This requires generating images that correspond to a given input number of objects. However, text-to-image models struggle to grasp the notion of count. We propose to rely on a double conditioning of Stable Diffusion with both a prompt and a density map in order to augment a training dataset for few-shot counting (8). Due to the small dataset size, the fine-tuned model tends to generate images close to the training images. We propose to enhance the diversity of synthesized images by exchanging captions between images thus creating unseen configurations of object types and spatial layout.

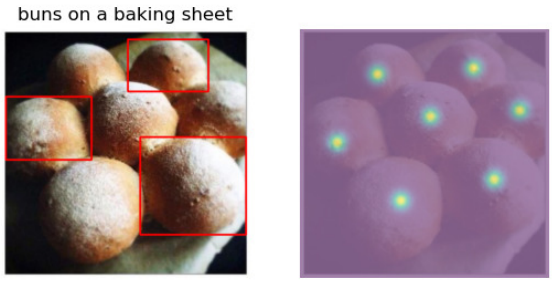


Figure 7. ***Left****: FSC147 image with BLIP2 caption (above) and exemplar boxes (in red).* ***Right****: Ground-truth density map.*

Few-shot Counting. The goal of few-shot class-agnostic counting is to learn to count objects regardless of their category. To achieve this, the query image $x \in \mathbb{R}^{H \times W \times 3}$ is annotated with $n \in \{0, 1, 2, 3...\}$ *exemplar* boxes of coordinates $b \in \mathbb{R}^4$. The counting network takes as input both the query image and the set of n boxes. It predicts a density map [22] $d \in \mathbb{R}^{H \times W}$ of same size as the image. As shown in 7, this ground-truth density map has zero values where there are no objects, and a Gaussian kernel of fixed variance at the center of every object. The final count is obtained by summing across all positions of the density map. The model is typically trained with an L_2 loss between the predicted and ground-truth densities. There are usually three datasets $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, $\mathcal{D}_{\text{test}}$ comprising objects of disjoints categories. The goal is to learn a counting network on $\mathcal{D}_{\text{train}}$ able to count the unseen objects in $\mathcal{D}_{\text{val}}$ and $\mathcal{D}_{\text{test}}$. To evaluate class-agnostic models, object categories from the test set $\mathcal{D}_{\text{test}}$ are disjoint from those in the validation $\mathcal{D}_{\text{val}}$ and train

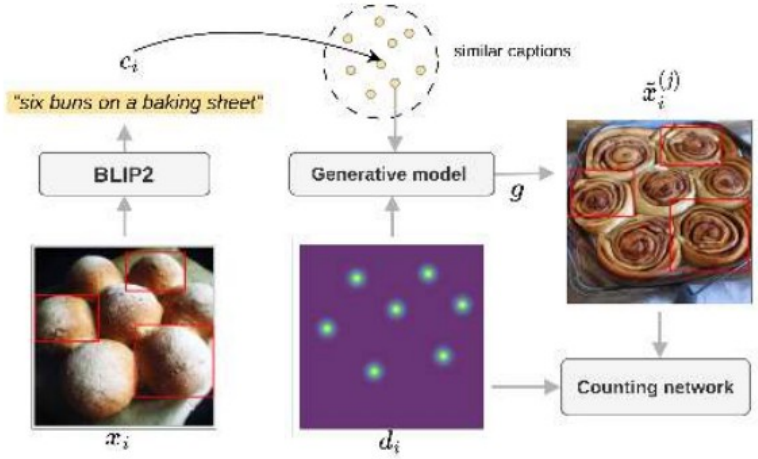


Figure 8. Overview of the SemAugm approach to create synthetic data that augment the training datasets of few-shot class-agnostic counting models.

sets $\mathcal{D}_{\text{train}}$. This open set evaluation allows us to measure the network's ability to count objects from *unseen* categories.

4.2.1 Methodology

Text-and-Density Guided Augmentations. To synthesize new images that can effectively augment a few-shot counting dataset, we need to have control over the number of objects and how they are laid out. Indeed, we need to ensure that we know the density maps of the synthetic samples so that they can be used to train the model. In addition, being able to control object type and spatial configuration also constitutes a lever to diversify the dataset by generating new combinations of categories and densities. It allows circumventing the labeling of the generated data and also constitutes a lever to diversify it. As few-shot counting datasets are generally limited in size, we take advantage of available pre-trained diffusion models to synthesize diversified augmentations of the training samples, reducing overfitting and improving generalization. However, large pre-trained generative models such as Stable Diffusion are usually conditioned through textual prompts.

To finetune these models, we first have to pair textual captions to the training images. We obtain diverse and descriptive captions using an off-the-shelf captioning model, *e.g* BLIP2 [23]. This produces richer captions than plain object categories such as "a photo of {class}". However, two shortcomings remain. First, generated captions may not contain any information about the number or arrangement of the objects. Second, text-conditioned Latent Diffusion Models (LDMs) poorly respect prompts regarding compositional constraints. Even adding this information in the caption does not guarantee that generated images would follow them. This is especially problematic as the correctness of the layout is a prerequisite to generate images for which we know the ground-truth. Therefore, we further condition the generative model directly on the density maps as an additional input, using the ControlNet fine-tuning strategy. To summarize, our generative model is now conditioned on a text prompt, obtained by an automated captioning of the training image, and its ground truth density map to enforce the spatial layout of the objects. This allows us to synthesize new samples that augment the original image, while keeping the ground truth intact, making the augmentation amenable to supervised learning.

To formalize the augmentation process, let $\mathcal{D}_{\text{train}} = \{x_i, b_i, d_i\}_{i=1}^{N}$ be an annotated counting

dataset, with x_i an image, b_i the exemplar bounding boxes for each image, and d_i its ground-truth density map. Let $\mathcal{C} = \{c_i\}_{i=1}^{N}$ be the set of corresponding captions. For each image x_i, we aim at generating M augmentations using our text-density conditional generative model $g(d_i, c_i)$.

Baseline We sample augmentations from the LDM by taking advantage of the non-deterministic *reverse* diffusion process and the expressiveness of the pre-trained model. For an image x_i we produce M augmentations $\tilde{x}_i^{(j)}$ that share its caption and density map:

$$\tilde{x}_i^{(j)} = g(d_i, c_i), \quad j = 1, ..., M \tag{1}$$

These augmentations preserve both the number and layout of objects – because of the density conditioning – and the semantics *e.g.*, object category and type of background – because of the text prompt. This already augments the number of samples available for training.

Diverse We can, however, go further and *diversify* the augmentations by altering either the text description or the spatial organisation of the objects. To do so, we take advantage of dual conditioning on both densities and captions. We mix the two sets to create new combinations (density map, caption), producing augmentations that are semantically and geometrically more diverse than the original dataset. Yet, this mixing of the conditionings should be done carefully, to avoid low quality augmentations. Indeed, not all combinations make sense, *e.g.*, "a herd of cows" and "a pearl necklace" exhibit very different spatial layouts. To prompt the generative model with realistic (*density*, *text*) pairs, we rely on caption similarity to find new associations between images that share some semantics, *e.g.*, "cows" and "bisons".

We swap captions at random between pairs of *compatible* images. Two images are said to be compatible, if their captions are more similar than some threshold t_c, *i.e*:

$$\text{sim}(c_i, c_k) = \frac{\Psi(c_i)^\top \Psi(c_k)}{||\Psi(c_i)||_2 ||\Psi(c_k)||_2} > t_c$$

where Ψ is a suitable text encoder. We then sample new images using the initial density map, but replacing the original caption with the caption $c_k \in \mathcal{C}$ from a compatible training observation chosen at random:

$$\tilde{x}_i^{(j)} = g(d_i, c_k), \quad j = 1, ..., M \tag{2}$$

This process results in more diverse augmentations compared to the baseline and alters more the images than traditional augmentations (color jitter, crops, etc.), as shown in 9.

Synthetic and Diverse Balance We follow the training strategy from Trabucco *et al.* [24], where the synthetic augmentations are used as a regular data augmentation with a probability p_0 when training the counting model. As a way to balance baseline and diversified augmentations, we set a probability p_c that defines the fraction of the M augmentations that use a swapped caption instead of the original one. Typically, $p_c = 0.5$ means that 50% of the generated augmentations employ the original (caption, density) pair and that the remaining 50% use new (caption, density) combinations. For each augmentation, we keep the density used to condition the image generation and the original exemplar boxes as ground truth to train the model. Note that if the caption changes the object category, bounding boxes for the exemplars might not be accurate anymore (*e.g* "pens" are narrow and elongated, while "erasers" are closer to squares.

4.2.2 Experimental results

The approach has been tested on FSC147 [25], which is a 3-shot counting dataset with 147 object categories. It is the *de facto standard* of class-agnostic counting benchmarking. 89 categories are

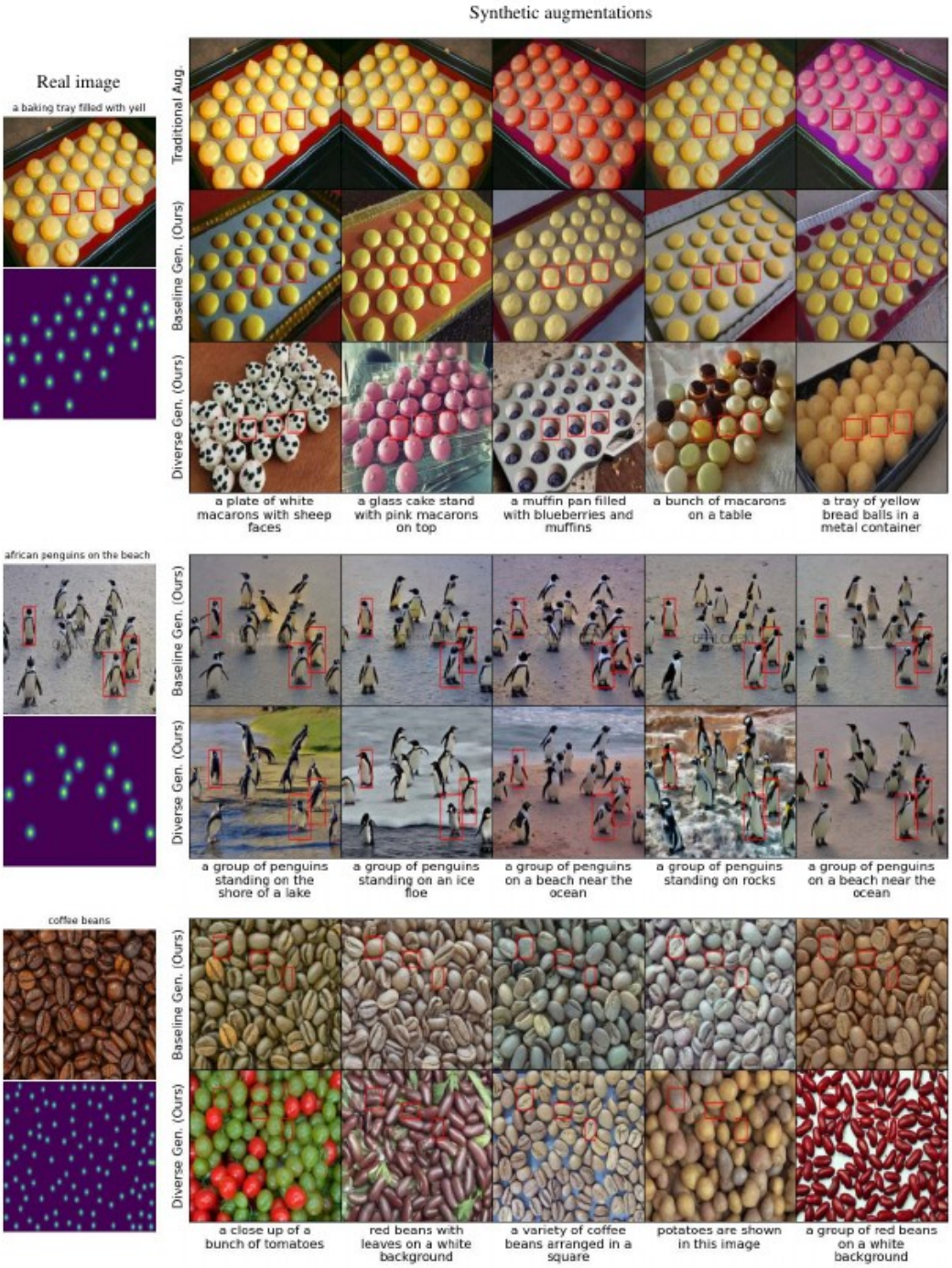


Figure 9. Qualitative results for the Baseline vs. Diverse augmentations. At the bottom of each diverse sample we show the caption used to generate the image. Our strategy allows to diversify the type of objects and/or the background.

used for the training set, 29 are included in the validation set the remaining 29 constitute the test set. Note that the categories from the three sets are completely disjoint. In total, the dataset contains 6135 images, from which 3659 are used for training. The number of objects in the images varies from 7 to 3731 with an average of 56. Every image is annotated with 3 exemplar bounding boxes and an object density map. We follow the standard evaluation of the counting accuracy through the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

	(a) SAFECount [26]				*(b) CounTR [28]*			
	Val		Test		Val		Test	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Trad. Augmentation*,†	13.95	51.73	13.73	91.85	14.25	50.15	13.13	88.21
+ Real Guidance [27]	14.94	53.09	13.48	**80.69**	15.37	49.47	13.37	96.44
+ Baseline (Ours)	13.30	49.38	13.22	92.47	12.60	43.53	11.83	87.97
+ Diverse (Ours)	**12.59**	**44.95**	**12.74**	89.90	**12.31**	**41.65**	**11.32**	**77.50**
Trad. Augmentation	15.28	47.5	14.25	85.54	13.13	49.83	11.95	91.23

Table 10. Quantitative results on FSC147. (*) Traditional augmentations include color jitter, random cropping. (†) [26] and [28] are reproduced, while those on the last line are reported from original papers

Comparison with Traditional Augmentation We report in Table 10 the improvement in counting accuracy on FSC417 with our augmentation strategies when training SAFECount [26] and CounTR [28]. Consistent with the literature on synthetic data augmentation, baseline augmentations improve the results for both networks: MAE decreases by respectively 5% and 10% for SAFECount and CounTR on the val set. Nonetheless, diversifying the augmentations allows us to reduce the MAE even further, by 10% and 11% on the same val set and by 7% (SAFECount) and 13% (CountTR) on the test set. We attribute this to ControlNet overfitting the training data due to the small dataset size. The low guidance employed to generate the images (2.0) aims at promoting diversity [29] but, as shown in Figure 9 (Baseline Gen.), the generated images remain close to the original image in terms of visual appearance of the objects and background. However, ControlNet generalizes to different captions. In Figure 9 (Diverse Gen.), we observe that swapping captions allows us to create more diverse data, altering the size and texture of objects and their background. Such features cannot be altered with traditional data augmentation. When mixing baseline and diverse augmentations, the performances for both networks improve significantly with respect to the model without synthetic augmentation, or with naive augmentations only.

Comparison with Real Guidance We compare our approach with Real Guidance, an augmentation strategy for image classification by He *et al.* [27]. Augmentations are generated by prompting a pre-trained text-to-image diffusion model with the image classes. To reduce the domain gap, the synthetic images are generated from the real images with added noise as proposed in SDEdit [30]. Table 10 shows that our augmentation strategy outperforms Real Guidance[2]. Starting from the real image with added noise is generally insufficient to preserve the number of objects and their positions (Figure 10, $2^{nd}col.$). It shows that the density map conditioning ensures the preservation of object positions and number without requiring to start from the real image, which can limit the diversity of the generated images.

[2]Except on test RSME with SAFECount, where Real Guidance performs better, due to two outlier test images with more than 2500 objects that dominate the average error (see supplementary material).

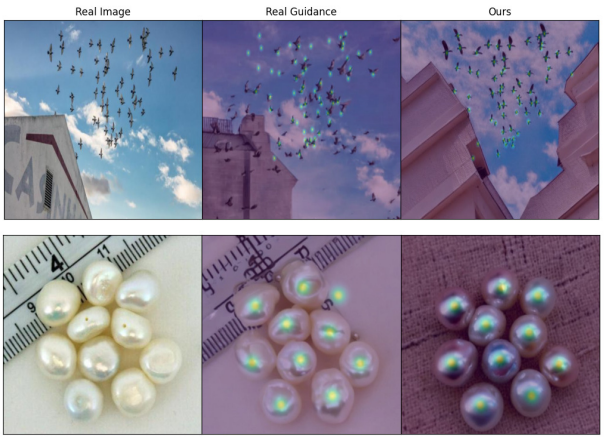


Figure 10. Qualitative comparison with Real Guidance [27]. Our augmentations preserve the layout while creating more diverse backgrounds. Ground-truth density maps overlap with the generated images (last 2 columns).

A more complete set of experimental results can be found in the associated publication (see Section 4.2.3). In particular, we report some results on the dataset CARPK [31] that consists to count cars in aerial views of parking lots, either with the networks only pre-trained on FSC147 or fine-tuned on CARPK itself. In that case, we obtain better results than SAFECount, CounTR and BMNet+ [32].

4.2.3 Relevant Resources and Publications

Relevant publications:

- Doubinsky, P.; Audebert, N.; Crucianu, M.; and Le Borgne, H. Semantic Generative Augmentations for Few-Shot Counting. In Winter Conference on Applications of Computer Vision (WACV), 2024. [33].
  Zenodo record: `https://zenodo.org/records/10204069`.

Relevant software and/or external resources:

- The PyTorch implementation of our work "SemAugm" can be found in `https://github.com/perladoubinsky/SemAug`.

Limitations Our synthetic data needs a ground truth and exemplars to train the counting network. Conditioning on densities makes it possible to reuse both the original density and the exemplar bounding boxes. However, changing the caption can affect the object category, and in turn its shape. In some rare cases, exemplar boxes do not fit the generated objects anymore. We explored to what extent refining these boxes could improve our model. We segmented objects using SAM in zero-shot [34] prompted with object centers. Preliminary results showed no improvement with box refinement, possibly due to inaccurate segmentation.

4.2.4 Relevance to AI4Media use cases and media industry applications

The developers of AI tools for the media industry can benefit from this asset to create synthetic data that will enrich their training datasets. The current version of the tool has been tested in the context of few-shot class-agnostic counting, that is the ability to count some object of any type in an image, by showing to the tool only few (e.g., 3) examples.

The adoption of AI tools in the media industry raised several challenges, regarding their performance but also their reliability and to which extent the tools can be trusted, that is their trustworthiness. The current tool can contribute to address these challenge by several ways (1) by augmenting the training datasets of AI tools, it will contribute to improve their performance (2) since the augmentation is made with synthetic data which generation is controlled, to some extent, with a human instruction, the resulting dataset can be enriched in order to reduce their potential biases toward particular classes of individuals

4.3 AUTOLYCUS: Exploiting Explainable Artificial Intelligence (XAI) for Model Extraction Attacks against Interpretable Models

Contributing partner: IBM

4.3.1 Overview

As the adoption of Machine Learning as a Service (MLaaS) platforms has experienced significant growth, there has been a corresponding increase in the demand for tools that facilitate eXplainable AI (XAI) [35]. These tools are crucial in providing users with transparency and a comprehensive understanding of how decisions are made by ML models. However, the data used for such explanations can pose security and privacy risks. Existing literature identifies attacks on machine learning models, including membership inference [36], model inversion [37], and model extraction attacks [38]. These attacks target either the model or the training data, depending on the settings and parties involved.

XAI tools can increase the vulnerability of model extraction attacks, which is a concern when model owners prefer black-box access, keeping model parameters and architecture private. To exploit this risk, we propose AUTOLYCUS, a novel retraining (learning) based model extraction attack against interpretable models under black-box settings. As XAI tools, we exploit Local Interpretable Model-Agnostic Explanations (LIME) [11] and Shapley Values (SHAP) [39] to infer decision boundaries and create surrogate models that replicate the functionality of the target model. LIME and SHAP are mainly chosen for their realistic yet information-rich explanations, coupled with their extensive adoption (most used, cited, and active model agnostic explainers), simplicity, and usability.

4.3.2 Methodology

To learn a surrogate model S that closely approximates the target model M, the attacker first needs to create a surrogate dataset D_S. We assume that the attacker has access to n samples per class in the auxiliary dataset $D_A = X_1, X_2, \ldots, X_{t*n}$ which may or may not have samples from the original dataset D_M. Here, $X_i = \{x_i^1, x_i^2, \ldots, x_i^m\}$, where x_i^j represents the value of feature j in sample X_i and m is the total number of features. We also assume that there exists a query budget Q which restricts the total number of queries that an attacker can send to the target model M. The proposed model extraction attack is depicted in Figure 11.

Generating Candidate Samples. Assume the attacker sends a query for a given sample X_i (e.g., one of the samples in D_A) to the target model M. The target model M sends the predicted class y_i and the corresponding explanation E_i. The explanation returned by LIME consists of the prediction probabilities (due to black-box access scenario, we assume only the top class) y_i and the decision boundaries db_i^j ordered by feature importances. Whereas, SHAP only returns the feature

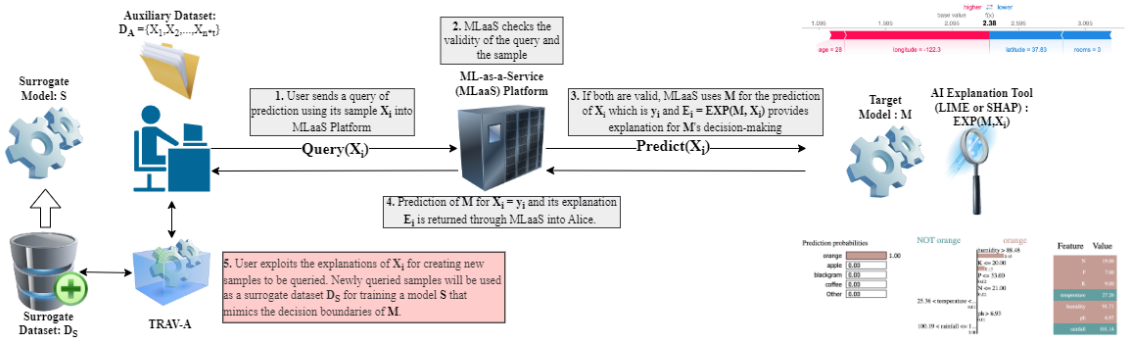


Figure 11. AUTOLYCUS system diagram consisting of the following steps: (1) a user sends a query to the MLaaS platform, (2) the MLaaS platform verifies the validity of the query such that no empty or incomplete queries are sent, (3) the ML model M predicts the class of the queried sample y_i and the explainer computes its explanation E_i, (4) the MLaaS platform returns the results to the user, and (5) in case of an adversarial user, they exploit explanations via TRAV-A algorithm to extract the decision boundaries of the target model M.

importance. To generate new and informative candidate samples, the attacker considers only the top k features of the sample X_i through its feature importance.

For each of the top k features, the attacker generates candidate samples by analyzing the decision boundaries returned by LIME or the feature importances returned by SHAP. The attacker computes new values of feature(s) j, denoted as $\hat{x_i}^j$ by altering it into the decision boundary db_i^j in LIME or to the next available value in SHAP with the coefficient(s) denoted as δ_j. Formally, $\hat{x_i}^j$ is computed as: $\hat{x_i}^j = db_i^j \pm \delta_j$ (in LIME) or $\hat{x_i}^j = {x_i}^j \pm \delta_j$ (in SHAP), where j is the index of the feature that the attacker is aiming to modify and δ_j is the alteration coefficient. For LIME, δ is equal to 1 for categorical features to reflect encoding difference and to 0.01 (or lower) for continuous features. Since the perturbation is decided by the decision boundaries in LIME, δ has lower importance. On the other hand, it is very important in SHAP, since it determines the exploration difference between successive samples. A good rule of thumb is setting it close to the standard deviations if available or to the quarters of the solution range. Depending on the intended alteration, δ_j can be manually configured to larger or lower values if necessary. The resulting candidate sample is obtained as $\hat{X}_i = \{x_i^1, x_i^2, \ldots, \hat{x_i}^{j_1}, \ldots, x_i^m\}$ or $\hat{X}_i = \{x_i^1, \hat{x_i}^2, \ldots, {x_i}^{m-2}, \hat{x_i}^{m-1}, x_i^m\}$.

Creating the Surrogate Dataset. To create the surrogate dataset D_S, the attacker uses the traversal algorithm **TRAV-A**. Recall that the attacker has access to n samples per class from the auxiliary dataset D_A. Thus, initially, D_S is limited to D_A. Let D_E denote the dataset with the samples that need to be explored (initially $D_E = D_A$). TRAV-A starts by exploring the samples in D_E. It selects the first sample X_i in dataset D_E and sends a query to the target model M. After receiving the predicted class y_i, TRAV-A checks if the maximum number of samples generated for this class has been reached. If that is the case, TRAV-A continues by sending a query for the next sample in D_E and checking if the above condition is met. Otherwise, TRAV-A adds X_i to the surrogate dataset D_S and generates new candidate samples (as previously described).

If any of the generated samples has not been previously explored (i.e., is not a part of D_E), then it is added to D_E. In the next iteration, TRAV-A sends a query for the next sample in D_E. TRAV-A employs a breadth-first search strategy to traverse the candidate samples generated during the exploration of a specific sample. We adopt this approach to prevent the deep exploration and propagation of a single sample. This process continues until one of the following conditions is met: (i) there are no unexplored samples in D_E or (ii) the query budget Q has been exhausted. Using the generated surrogate dataset D_S, the attacker trains the surrogate model(s) S.

4.3.3 Experimental Results

To evaluate the performance of the proposed algorithm, we employ three widely used datasets: Iris, Breast Cancer, and Adult Income. Each dataset is split into three subsets: (i) the training dataset (75%), (ii) the test dataset (15%), and (iii) the auxiliary set (10%). The training dataset is used for training the target model M, the test dataset for evaluating the performance of the model extraction attacks, and the auxiliary set for creating the auxiliary dataset D_A. We conduct experiments on the following interpretable models; decision trees, logistic regression, Naive Bayes, k-nearest neighbor classifiers, and random forest.

We use accuracy and model similarity to evaluate the performance of the constructed surrogate models in comparison to the target models. Accuracy represents the proportion of correct classifications relative to all predictions made by the surrogate model. Model similarity is the label agreement between the target model and the surrogate models against a neutral dataset, which is referred in the literature also as fidelity or $1 - R_{test}$ [38].

We compare the performance of AUTOLYCUS with four other approaches:
Baseline attack. The surrogate model S is trained directly on the auxiliary dataset ($D_A = D_S$). In this scenario, the attacker does not send any queries ($Q = 0$) to the target model M.
Steal-ML attacks. Tramer et al. propose a "path-finding attack" to target decision tree models and an equation-solving attack to target logistic regression models for model extraction [38]. The path-finding attack is a deterministic, top-to-bottom attack that explores all the nodes until an exact reconstruction is achieved. "The equation-solving attack" is a technique employed against logistic regression models and neural networks. Its query results are converted into linear equations to be solved collectively.
IWAL attack. Chandrasekaran et al. [40] propose an active learning based model extraction attack IWAL to target tree structured models. IWAL is the importance weighted active learning algorithm of Beygelzimer et al. [41], which iteratively refines a tree in each query by minimizing the labeling error.

For each corresponding model type and dataset, we compare the similarity results and the query budget required for the proposed attack to the ones required for the baseline attack, Steal-ML, and IWAL attack. We obtain the results for these attacks from their respective papers. For Figure 12 and Table 11, the number of top features allowed to be explored (k) is set to 3. The size of the auxiliary dataset per class (n) is set to 1 (for LIME) and 5 (for SHAP). This is a design choice to demonstrate the impact of the size of the auxiliary dataset while providing a slight leverage to SHAP considering that LIME explanations offer significantly more information.

Our results demonstrate the effectiveness of the proposed attack. We observe that by exploiting AI explanations, an attacker can create fairly accurate surrogate models that have high similarity to the target models even under low query budgets. The performance of the model extraction attack is enhanced as the model complexity (architecture, number of features, and classes) decreases or model accuracy increases. We also observe that the proposed attack requires fewer queries for partial reconstructions with comparable accuracy and similarity than the state-of-the-art attacks that rely on exact reconstruction. Furthermore, we explored potential countermeasures to mitigate this attack such as adding noise to the decision boundaries of explanations or adding noise to the training data.

4.3.4 Relevant Resources and Publications

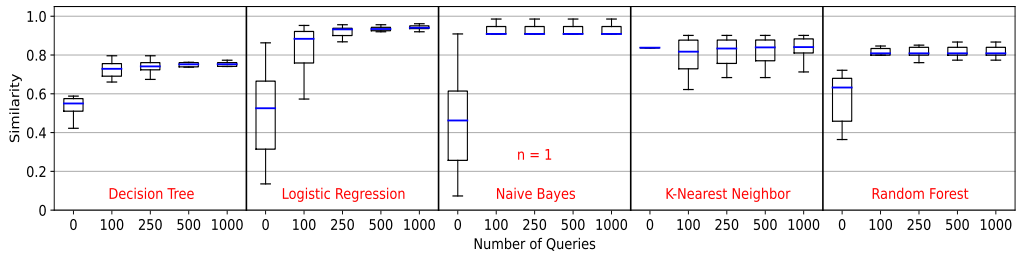
Relevant publications:

- A. C. Oksuz, A. Halimi, and E. Ayday. "AUTOLYCUS: Exploiting Explainable Artificial Intelligence (XAI) for Model Extraction Attacks against Interpretable Models", Proceedings

Table 11. Comparison with SOTA

Dataset	SOTA Attacks				AUTOLYCUS				
	Attack Name	Model	$1 - R_{test}$	Queries	Model	$1 - R_{test}$	Queries	n	XAI Tool
Iris	Equation Solving [38]	Logistic Regression	1	644	Logistic Regression	1	**100**	1	LIME
Iris	Path Finding [38], [40]	Decision Tree	1	246	Decision Tree	1	**10**	1	LIME
Iris	IWAL [40]	Decision Tree	1	361	Decision Tree	1	**10**	1	LIME
Breast Cancer	Equation Solving	Logistic Regression	1	[644,1485]	Logistic Regression	0.992	**100**	5	SHAP
Adult Income	Equation Solving	Logistic Regression	1	1485	Logistic Regression	0.998	**1000**	5	SHAP
Adult Income	Path Finding	Decision Tree	1	18323	Decision Tree	0.937	**1000**	5	SHAP
Adult Income	IWAL	Decision Tree	1	244188	Decision Tree	0.937	**1000**	5	SHAP

LIME
(a)



SHAP
(b)

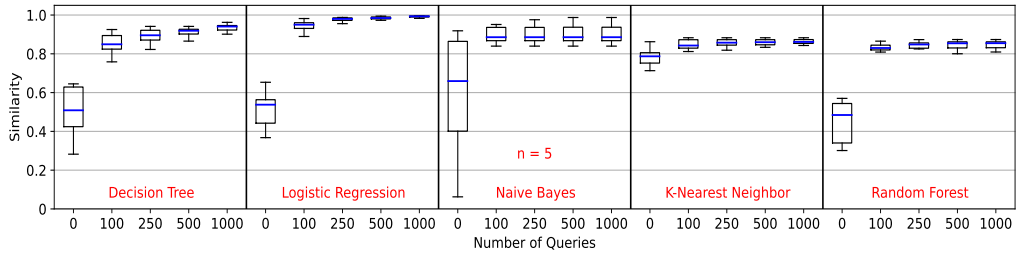


Figure 12. Impact of the number of queries (Q) on surrogate model similarity in the Adult Income dataset.

on Privacy Enhancing Technologies (PETS), 2024 [42].
Arxiv record: `https://arxiv.org/pdf/2302.02162`.

4.3.5 Relevance to AI4Media use cases and media industry applications

XAI plays a crucial role in the media industry by making it easier for stakeholders to understand how decisions are made. This leads to improved content moderation and a more personalized and engaging user experience. On the other hand, this work shows that an attacker can create a model that performs similarly to the target model by exploiting AI explanations. Given the extensive use of AI models in the media industry, this work can help companies understand the extent of model stealing attacks and how to better protect their AI assets.

4.4 Concept Discovery and Dataset Exploration with Singular Value Decomposition

Contributing partner: HES-SO

4.4.1 Overview

Many applications now require models to be pre-trained on large-scale datasets such as ImageNet. However, difficulties such as labeling errors and long-tail errors, which are confusing for domain experts, have not been completely resolved. Labeling errors and noise can reduce model quality and evaluation, producing unintended biases. As dataset volumes increase, evaluating the quality of supervised labels becomes more difficult. Deep learning models are often overconfident, inaccurate, and biased toward simple features such as texture.
This study provides a framework for analyzing patterns learned by deep learning models using matrix factorization. The method identifies vectors at intermediate representations that can be linked to high-level, human-understandable notions. These vectors are then used to investigate training datasets to detect inputs that contain artifacts, confounding variables, or inaccurate labels. The suggested method provides a novel strategy for automatically discovering concept vectors by decomposing a layer's latent space into matrices of singular values and vectors. The approach is unbiased and successful in uncovering ignored concepts or patterns because it does not rely on user-defined commands or questions.

4.4.2 Methodology

The method consists of three main phases: identification of orthogonal vectors via Singular Value Decomposition (SVD), gradient-informed ranking of these vectors, and selection and visualization of top vectors as human-understandable concepts.

Figure 13. Visualization of the discovered concept vectors for ImageNet classes. In the first two rows, the input image is shown together with a zoomed-in version of the automatically segmented concept. The last row shows the input images with largest projection on the concept vectors and the relative concept segmentation masks.

SVD is applied to the matrix of a layer's responses to the input dataset to obtain orthonormal vectors summarizing the encoding of the dataset in the latent space. This decomposition yields matrices U (orthonormal vectors), Σ (singular values), and V (right singular vectors).
Gradient-Informed Ranking. To ensure the singular vectors are relevant to the downstream predictive task, the perturbation impact of moving feature representations along these vectors

is evaluated. This is done by considering the directional derivative of the model output along the singular vectors, combined with the projection coefficients of activations and gradients. The importance of a singular vector to the prediction is then computed as the sample mean of these values across all inputs.

Candidate Directions for Discovery The top-ranking vectors are identified as candidate vectors for concept discovery. These vectors are projected onto the input data to retrieve samples with increasing projection values. For convolutional networks, concept activation maps are created by weighing feature maps with the coefficients of the concept vector. Concept segmentation masks are derived by retaining input pixel values with high activation in the concept maps.

The discovered concept vectors are used to explore the dataset, identifying anomalous samples that may contain artifacts or misleading factors. These samples are flagged based on the statistical dispersion of their projections onto the concept vectors.

4.4.3 Experimental results

The concept discovery strategy was applied to standard models with pretrained weights available online, with a particular emphasis on **Inception V3** (IV3) trained on the ImageNet ILSVRC2012 dataset. This method exhibited the ability to automatically detect and understand high-level concepts inside natural image categories.

The analysis included classifications such as lionfish, police van, bubble, and zebra. The approach discovered concept vectors that were consistent with high-level elements such as patterns (e.g., lionfish fins, zebra coat), graphics and tires (police vehicle), and glossy reflections (bubble). These concepts were split and illustrated to ensure they were understandable to humans (see Fig. 13).

User evaluations indicated that the discovered concepts were easy to understand, thereby aiding in model interpretation. Quantitative studies further validated the effectiveness of the method.

Using concept vectors, we discovered outlier photos (see Fig. 14) in the sample with inaccurate or confounding labels. This phase improved the overall quality and dependability of the training data.

The results show that the suggested method effectively identifies concepts that are understandable

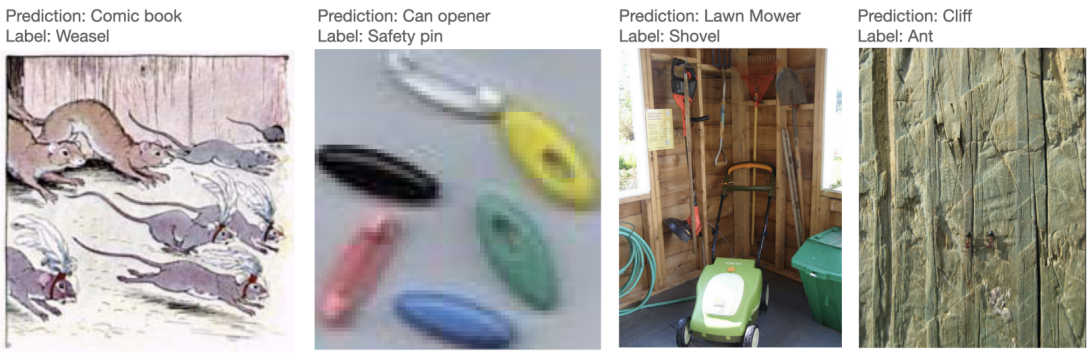


Figure 14. Results of dataset exploration with concept discovery. The method identifies training images with particular issues. The first example presents a strong style shift from real images to drawings. Extremely poor resolution affects the quality of the second input. The last two images present confounding factors. Multiple labels are equally correct for the third image, and the last image shows an optical illusion - where there seems to be a cliff, there is actually a high resolution detail of two ants on a wooden surface.

to humans while also detecting labeling errors and confounding factors in huge datasets. By

applying concept discovery to only a small portion of the training dataset, significant insights can be obtained about the behavior of the model and the quality of the dataset.

4.4.4 Relevant Resources and Publications

Relevant publications:

- Graziani, Mara, et al. "Concept discovery and dataset exploration with singular value decomposition." ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML. 2023. [43].

Relevant software and/or external resources:

- The PyTorch implementation of our work can be found in `https://github.com/maragraziani/concept_discovery_svd`.

4.4.5 Relevance to AI4Media use cases and media industry applications

In the context of AI4Media, concept discovery via Singular Value Decomposition (SVD) can uncover underlying patterns in multimedia datasets, improving transparency and decision-making in recommendation systems. For example, it can recognize visual features such as animals or vehicles, increasing the accuracy of personalized content recommendations. Furthermore, through dataset exploration, SVD-based concept discovery improves the dependability and ethical integrity of AI applications in the media, aligning user expectations with responsible content management methods.

4.5 Attention Meets Post-hoc Interpretability: A Mathematical Perspective

Contributing partner: UCA

4.5.1 Overview

Attention-based architectures, in particular transformers, are at the heart of a technological revolution. Interestingly, in addition to helping obtain state-of-the-art results on a wide range of applications, the attention mechanism intrinsically provides meaningful insights on the internal behavior of the model. Can these insights be used as explanations? Debate rages on. In this work, we mathematically study a simple attention-based architecture and pinpoint the differences between post-hoc and attention-based explanations. We show that they provide quite different results, and that, despite their limitations, post-hoc methods are capable of capturing more useful insights than merely examining the attention weights.

4.5.2 Methodology

Let us first present the architecture of the model we use for evaluating the explanation methods.

This work considers a set of tokens belonging to a dictionary identified with $[D]$. A document x is an ordered sequence of tokens $x_1, \ldots, x_T$, where T denotes the length of the document. Without loss of generality, it is assumed that the d unique tokens of x are the first d elements of $[D]$.

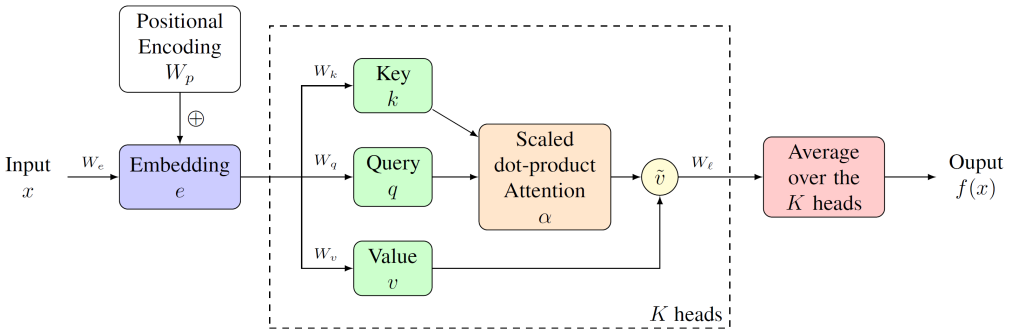


Figure 15. Illustration of the model architecture considered for comparing explanation methods

The model f is a **single-layer, multi-head, attention-based network followed by a linear layer**. More formally:

$$f(x) := \frac{1}{K}\sum_{i=1}^{K} f_i(x) = \frac{1}{K}\sum_{i=1}^{K} W_\ell^{(i)}\tilde{v}^{(i)}(x)\,, \tag{3}$$

where $f_i := W_\ell{}^{(i)}\tilde{v}^{(i)} \in \mathbb{R}^{d_{out}}$, with $W_\ell{}^{(i)} \in \mathbb{R}^{1\times d_{out}}$ being the part of the final linear layer associated with head i, and for $i \in [K]$, $\tilde{v}^{(i)}(x)$ is the output of an individual head. The value of f is used for classification; for instance, in a sentiment analysis task, document x is classified as positive if $f(x) > 0$.

The input text x, is transformed into an embedding $e \in \mathbb{R}^{T\times d_e}$ by summing word embeddings and positional encodings.

For each of the K heads, the key $k \in \mathbb{R}^{T\times d_{att}}$, query $q \in \mathbb{R}^{T\times d_{att}}$, and value $v \in \mathbb{R}^{T\times d_{out}}$ matrices are computed by applying linear transformations to e using $W_k, W_q \in \mathbb{R}^{d_{att}\times d_e}$, and $W_v \in \mathbb{R}^{d_{out}\times d_e}$, respectively. The attention weights $\alpha \in \mathbb{R}^T$ are then computed as the softmax of the scaled dot-product of k and q. Then the intermediary output $\tilde{v} \in \mathbb{R}^d_{out}$ is computed are the average of the values v weighted by the attention α.

Each head outputs the linear transformation $W_\ell \in \mathbb{R}^{1\times d_{out}}$ of the $\tilde{v}$ associated with the query corresponding to the `[CLS]` token.

The final prediction $f(x)$ of the model is the average of the outputs across all heads.

Attention-based explanations

In that context, for a given query $q \in \mathbb{R}^d_{att}$, the attention α_t received by each index t is defined as

$$\alpha_t := \frac{\exp\{q^\top k_t/\sqrt{d_{att}}\}}{\sum_{u=1}^{T_{max}}\exp\{q^\top k_u/\sqrt{d_{att}}\}}\,. \tag{4}$$

The scaling factor $1/\sqrt{d_{att}}$, although not strictly necessary (since W_q and W_k are learnable parameters of the model), is retained to properly scale the positional embedding.

The intermediary output value before the final linear transformation associated with the query q is

$$\tilde{v} := \sum_{t=1}^{T_{max}} \alpha_t v_t \in \mathbb{R}^d_{out}\,. \tag{5}$$

Each individual head transforms the $\tilde{v}$ associated with the query corresponding to the `[CLS]` token. Specifically, for $i \in [K]$, $f_i(x) = W_\ell^{(i)} \tilde{v}^{(i)}$.

Note that, in general, a Transformer model is structured as a series of sequential layers, each equipped with a specific number of parallel heads. These heads operate independently, executing the attention mechanism. To produce token-level attention-based explanations, one must aggregate the attention matrices at both the head and layer levels. [44] provide a detailed depiction of these operations; refer to Figure 2 in [44] for a comprehensive illustration.

In our scenario, the model is single-layered, hence layer-level aggregation is omitted.

As a result, each head produces an attention vector of size T that highlights the focus of the head on each token. However, heads often concentrate on different sections of the document.

Thus, aggregating the K attention vectors is crucial. The two most common aggregation methods involve computing the average vector or determining the maximum value among the vectors for each token. Formally, for any token $t \in [T]$, we define:

$$\alpha - avg_t := \frac{1}{K} \sum_{i=1}^{K} \alpha_t^{(i)} , \tag{6}$$

and

$$\alpha - max_t := \max_{i \in [K]} \alpha_t^{(i)} . \tag{7}$$

It is important to note that $\alpha - avg$ and $\alpha - max$ can lead to very different explanations. Additionally, $\alpha - avg$, $\alpha - max$, and $G - l1$ generate non-negative weights. Consequently, these methods do not differentiate between words that contribute positively or negatively to the prediction.

Gradient-based explanations

Given a model f and an instance x, the gradient with respect to a token $t \in [T]$ is defined as:

$$\nabla_{e_t} f(x) \in \mathbb{R}^{d_e} . \tag{8}$$

It is important to note that the gradient ∇_{e_t} is calculated with respect to the embedding vector $e_t \in \mathbb{R}^{d_e}$.

The function f is linear with respect to the f_i head, $i \in [K]$, hence, the gradient of f with respect to the token embedding e_t is:

$$\nabla_{e_t} f(x) := \frac{1}{K} \sum_{i=1}^{K} \nabla_{e_t} f_i(x) \in \mathbb{R}^{d_e} . \tag{9}$$

The primary quantity of interest is the gradient of a single attention head, $\nabla f_i(x)$. Recall that q is the query corresponding to the classification token `[CLS]`.

The gradient of the model f with respect to the embedded token e_t, $t \in [T]$ **can be expressed with attention weights (Gradient Meets Attention)**:

$$\nabla_{e_t} f(x) = \frac{1}{K} \sum_{i=1}^{K} \left[\alpha_t^{(i)} (W_v^{(i)})^\top (W_\ell^{(i)})^\top + \frac{\alpha_t^{(i)}}{\sqrt{d_{att}}} W_\ell^{(i)} \left(v_t^{(i)} - \sum_{s=1}^{T_{max}} \alpha_s^{(i)} v_s^{(i)} \right) (W_k^{(i)})^\top q \right] \in \mathbb{R}^{d_e} . \tag{10}$$

Perturbation-based explanations

LIME for text data, as detailed in [45], operates by starting with the document x to be explained and generating local perturbations $X_1, \ldots, X_n$.

Let X denote the distribution of the randomly perturbed documents. In this context, X is generated as follows: first, pick s uniformly at random from $[d]$ (the local dictionary), then choose a set $S \subseteq [d]$ of size s uniformly at random. Finally, remove all occurrences of words appearing in S from x, where removing means replacing with the `UNK` token. For simplicity, it is assumed that tokens and words coincide. The perturbed samples $X_1, \ldots, X_n$ are independent and identically distributed repetitions of this process.

Associated with the X_i samples are vectors $Z_1, \ldots, Z_n \in \{0,1\}^d$, indicating the presence or absence of a word in X_i. Specifically, $Z_{i,j} = 1$ if word j is present in X_i and 0 otherwise.

Under mild assumptions, ([45], Theorem 1) demonstrate that LIME's coefficients converge to *limit coefficients* β^∞. Specifically, this convergence occurs in particular when the number of perturbed samples n is large, and the bandwidth ν is also large.

The expression for the limit coefficient associated with word j is:

$$\beta_j^\infty = 3\mathbb{E}\left[f(X) \mid j \notin S\right] - \frac{3}{d}\sum_k \mathbb{E}\left[f(X) \mid k \notin S\right] . \tag{11}$$

This coefficient can be computed (exactly or approximately) as a function of the model parameters, providing precise insights into LIME's behavior in this context. This computation represents the main result of this section.

Using the previous expression, ***LIME coefficients can be expressed using attention weights (LIME Meets Attention)***:

$$\beta_j^\infty = \frac{3}{2K}\sum_{i=1}^{K}\sum_{t=1}^{T_{max}} W_\ell^{(i)}\left(\alpha_t^{(i)} v_t^{(i)} - \alpha_{h,t}^{(i)} v_{h,t}^{(i)}\right)\mathbf{1}_{X_t=j} + \mathcal{O}\left(T_{max}^{(2-\epsilon)\vee 3/2}\right) . \tag{12}$$

More details can be found in [46]. The Figure 16 illustrates the different explanations provided by attention-based, gradient-based, and perturbation-based explanations for the same input.

4.5.3 Results/Conclusions

In this work, we offered a theoretical analysis on how post-hoc explanations relates to a single-layer multi-head attention-based network. Our work contributes to the ongoing debate in this area by providing exact and approximate expressions for post-hoc explanations on such model. Through these expressions, we were able to highlight the fundamental differences between attention-based, gradient-based, and perturbation-based explanations. This deeper understanding not only enriches the ongoing discourse surrounding interpretability but also offers valuable insights for practitioners and researchers navigating the complexities of transformers' interpretation.

It is crucial to acknowledge that the quest for perfect explanations remains elusive; no single method has emerged as entirely satisfactory. However, it is clear that current models employ attention scores in a non-intuitive manner to arrive at the final prediction. In particular, these scores go through a series of further transformations, which is ignored when looking solely at attention scores. These scores also always provide a positive explanation, in contrast to (most) perturbation-based and gradient-based approaches. For these reasons, we believe that they can

extract more valuable insights than a mere examination of attention weights. This finding aligns with the assertions made by Bastings and Filippova [47].

As future work, we plan to broaden the scope of our analysis by extending our investigations to diverse range of post-hoc interpretability methods, including Anchors, thus understanding model explanations across different methodologies. We also would like to obtain similar statements (connecting explanations to the parameters of the model) for more complicated architectures, including skip connections, additional non-linearities, and multi-layer models, enabling us to discern the relationship between model parameters and different explanations. Additionally, there is some interplay between the sampling mechanism of perturbation-based methods (often replacing at the word level) and the tokenizer used by the model (tokens are often subwords) which we would like to understand better. Lastly, we emphasize that our focus in this paper has been on text classification. This choice allows us to capitalize on well-established, and broadly studied post-hoc explainers and conduct a thorough theoretical analysis based on this specific domain. However, we intend to expand the scope of applications for our analysis. Specifically, we remark that our study focused on token-level explanations. Moving forward, we intend to extend our findings beyond text models to encompass other domains, such as computer vision.

*Figure 16. Different explainers can produce very different explanations. Here, the attention mean (α-**avg**) and maximum (α-**max**) over the heads, LIME (**lime**), the gradient mean (**G-avg**), L^1 norm (**G-**l_1), and L^2 norm (**G-**l_2), with respect to the tokens, and Gradient times Input (**G** × **I**) are employed to interpret the prediction of a sentiment-analysis model. Words with positive (respectively, negative) weights are highlighted in green (respectively, red), with intensity proportional to their weight. In the example, all the explainers identify the word questionable as highly significant, while only lime, and G × I highlight a negative contribution. Interestingly, α-avg and α-max identify the word popular as the most important word in absolute terms, in disagreement with the all others.*

4.5.4 Relevant Resources and Publications

Relevant publications:

- Lopardo, Gianluigi, Frederic Precioso, and Damien Garreau. "Attention Meets Post-hoc Interpretability: A Mathematical Perspective." Forty-first International Conference on Machine Learning, 2024. [46].
  Zenodo record: `https://zenodo.org/record/12702363`.

Relevant software and/or external resources:

- The PyTorch implementation of our work can be found in `https://github.com/gianluigilopardo/attention_meets_xai`.

4.5.5 Relevance to AI4Media use cases and media industry applications

In media-related use cases where explainability is key, our work will help to better understand which explainability technique would suit better the needs of the specific use case.

4.6 Leveraging Visual Attention for OOD Detection

Contributing partner: UNIFI

4.6.1 Overview

Understanding the reliability of machine learning models is paramount when such models are deployed for real-world tasks. One of the main issues of deep learning based classifiers, which is due to the softmax operator, is that they tend to output high scores even for random inputs[48], [49]. Unfortunately, this behavior hinders the reliability of neural network based systems.

Out-Of-Distribution (OOD) detection is a crucial challenge in computer vision, especially when deploying machine learning models in the real world. In this work, we propose a novel OOD detection method leveraging Visual Attention Heatmaps from a Vision Transformer (ViT) classifier.

4.6.2 Method

In this section, we present the details of our proposed out-of-distribution (OOD) detection model, which can be summarized in four key steps:

- **Train the Vision Transformer Classifier**: We begin by training a state-of-the-art Vision Transformer classifier using large-scale pre-training.
- **Extract Visual Attention Heatmaps**: From the trained ViT classifier, we extract Visual Attention Heatmaps, highlighting the most relevant regions within each input image. These heatmaps serve as valuable guides for focusing on critical areas during the OOD detection process.
- **Convolutional Autoencoder Training**: We proceed to train a Convolutional Autoencoder (AE) using the extracted attention heatmaps as training data. The autoencoder learns to encode the meaningful and distinctive representations of the attention maps, facilitating precise image reconstruction.
- **Image Reconstruction Error as Discriminatory Feature for OOD Detection**: The core of our OOD detection model lies in the image reconstruction process. By comparing the reconstructed attention heatmaps with the original ones, we can effectively identify OOD samples based on their deviations from the learned in-distribution patterns.

4.6.3 ViT Backbone

The Vision Transformer is a state-of-the-art approach for various tasks, including classification [50]. Unlike a traditional convolutional approach, ViT relies on a Multi-Head architecture. Specifically, an image is divided into a sequence of patches, which are linearly projected and fed into an Encoder [50]. The core of the Encoder is the Multi-Head Attention. This Multi-Head Attention enables the model to capture global dependencies and contextual information, allowing the system to model both long-range interactions and small details present in the image.

In this work, we leverage ViT's strengths to train a classifier, exploiting its ability to learn from patch-level features and capture intricate relationships among different parts of an image. Interestingly, attention heatmaps, after fine-tuning, encode a semantic representation of input samples. We exploit this rich and at the same time light representation of input images to learn a representation for OOD detection. Figure 17 showcases examples of attention heatmaps generated by the proposed approach.

To perform classification, we fine-tune a pre-trained ViT [51] on ImageNet21k [52]. The pre-training procedure adheres to the guidelines outlined in [53], ensuring consistency with the suggested approach. The model takes input images of size 224×224 and divides them into patches of size 16×16. This way, each image is split into a grid of 14×14 patches, resulting in a total of 196 patches. We use the CrossEntropy as the Loss function.

The results of our experiments and evaluations are presented in section 4.6.6.

4.6.4 Using Visual Attention to Train an Autoencoder

The attention map provided by Vision Transformer can be highly beneficial in discriminating between different species in wild animal classification. The attention map is a visual representation that highlights the regions in the image that the model considers most relevant for making its predictions. It allows us to gain insights into what parts of the image the ViT focuses on when making classification decisions.

In the context of wild animal classification, where species might exhibit visual similarities, the attention map can serve as a valuable tool to understand how the model distinguishes between different animals. By analyzing the attention map, we can identify the key features or distinctive patterns that the model relies on to make accurate classifications. Furthermore, using the visual attention map can lead to improved model interpretability and explainability.

According to this, after training the Vision Transformer classifier, we proceed to extract the Visual Attention Heatmaps for each image in both the training and test sets. To facilitate efficient storage and analysis, we resize the attention heatmaps to a standardized size of $128 \times 128 \times 1$.

We train a Convolutional Autoencoder for the task of OOD detection, leveraging visual attention extracted from a pre-trained Vision Transformer. The Convolutional AutoEncoder is designed to reconstruct input images, then we use the reconstruction error to generate precision-recall curves for OOD detection. The architecture, as in [54], consists of an encoder and decoder, each comprising several convolutional layers, with Leaky ReLU activation functions to introduce a regularization effect. The encoder takes grayscale input images of size $128 \times 128 \times 1$ and progressively reduces the spatial dimensions while increasing the number of channels. It culminates in a bottleneck layer of size $512 \times 1 \times 1$. The decoder then upscales and progressively reconstructs the original input image through transposed convolutions and activations.

During the training process, the model is optimized to minimize the Mean Square Error (MSE) loss between the reconstructed heatmaps and the original input.

4.6.5 Training

In this section, we provide a comprehensive overview of the training details for both the Classifier and the Convolutional Autoencoder models.

Vision Transformer classifier We finetuned the proposed classifier using a pretrained Vision Transformer model on the ImageNet−21K dataset. The model was initialized with a patch size of 16x16, and the input images were resized to $224 \times 224 \times 3$ during training. We conducted the finetuning process for 50 epochs, utilizing the Cross Entropy loss function to optimize the model's performance. To optimize the model's parameters, we employed the Adam optimizer

with an initial learning rate of 0.0001. Additionally, we incorporated a learning rate scheduler to dynamically adjust the learning rate during training. Specifically, we employed the *StepLR* scheduler with a step−size of 7 epochs and a multiplicative factor of 0.1. This setup allowed us to gradually reduce the learning rate every 7 epochs by multiplying it with the specified gamma factor, which effectively aided in stabilizing and enhancing the convergence of the model during the fine-tuning process.

Convolutional Auto-Encoder During Convolutional Autoencoder training, we utilized grayscale visual attention heatmaps extracted from the Vision Transformer, as explained in Section 4.6.4, with an input size of $128 \times 128 \times 1$. To regularize each convolutional layer, we employed Leaky ReLU activation with a negative slope of 0.2. For optimization, we utilized the Adam optimizer with an initial learning rate of 0.0001. To enhance convergence stability and overall model performance, we implemented a linear learning rate scheduler. During the first 40 epochs, the learning rate halved every 10 epochs, after which it remained constant. This schedule ensured efficient training while preserving the fine-tuned model's performance. The Autoencoder's primary objective during training was to minimize the MSE loss between the reconstructed output and the input images. This training setup empowered the Autoencoder to learn meaningful representations of the input data, facilitating precise image reconstruction and substantially contributing to the subsequent out-of-distribution Detection process.

4.6.6 Experimental results

To demonstrate the effectiveness of the proposed method, we rely on several datasets. As a real-world scenario we use WildCapture[55]. We randomly split the classes to obtain in-distribution and out-of-distribution sets. We use the in-distribution set to train the Vision Transformer as described in section 4.6.3 and the AE. Then we use the out-of-distribution split as a test set.

In order to prove the efficacy of our method, we use also the Caltech CameraTrap dataset [56] as out-of-distribution set. We avoid overlap between classes in our WildCapture in-distribution set and Caltech CameraTrap.

The results of this experiment are summarized in Table 12, which clearly showcases the superiority of our method in detecting out-of-distribution samples compared to the baselines and alternative approaches. In fact, our method outperform the baselines in both Area Under the Precision-Recall Curve (AUPRC) and Area Under the Receiver Operating Characteristic Curve (AUROC) metrics.

Method	**ID: WildCapture**	**OOD:WildCapture**		**OOD: CCT[56]**	
	Accuracy	**AUROC**	**AUPR**	**AUROC**	**AUPR**
Deterministic	94.30	57.44	61.46	57.43	68.06
Ensemble	81.89	41.25	90.94	53.36	84.65
RGB-AE	94.30	59.01	77.56	-	-
Ours	**94.30**	**92.63**	**92.25**	**99.29**	**97.17**

Table 12. Results on WildCapture as in-distribution dataset

In Table 13 and in Table 14, we compare our method with some state-of-the-art methods using respectively CIFAR10 and CIFAR100 as in-distribution sets. In both experiments, we use respectively CIFAR100 and CIFAR10 as near OOD task and SVHN as far OOD task. The proposed approach achieves state-of-the-art performance, demonstrating remarkable accuracy and outperforming existing techniques in accurately detecting out-of-distribution samples with 100% AUPR

and AUROC in both benchmarks. This compelling result underscores the efficacy and versatility of our method in handling diverse and challenging datasets, making it a promising solution for out-of-distribution detection tasks.

Method	ID: CIFAR10	OOD:CIFAR100		OOD:SVHN	
	Accuracy	AUROC	AUPR	AUROC	AUPR
DUQ [57]	95.50	90.80	88.80	97.20	96.90
SNPG [58]	96.00	91.60	91.10	97.80	97.50
Vit Ensemble [59]	**98.70**	98.52	98.70	98.58	99.82
DHM [60]	96.30	**100.00**	**100.00**	**100.00**	**100.00**
Ours	97.80	**100.00**	**100.00**	**100.00**	**100.00**

Table 13. Results on Cifar10 as in-distribution dataset

Method	ID: CIFAR100	OOD:CIFAR10		OOD:SVHN	
	Accuracy	AUROC	AUPR	AUROC	AUPR
DUQ [57]	79.90	83.90	87.20	89.70	90.80
SNPG [58]	80.50	86.30	87.50	92.80	93.50
Vit Ensemble [59]	**91.71**	96.23	96.32	97.80	98.87
DHM [60]	81.30	**100.00**	**100.00**	**100.00**	**100.00**
Ours	89.80	**100.00**	**100.00**	**100.00**	**100.00**

Table 14. Results on Cifar100 as in-distribution dataset

4.6.7 Relevant Resources and Publications

Relevant publications:

- Cultrera, Luca, Lorenzo Seidenari, and Alberto Del Bimbo. "Leveraging Visual Attention for out-of-distribution Detection." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4447-4456. 2023. [61].
  Open access CVF: `https://openaccess.thecvf.com/content/ICCV2023W/OODCV/html/Cultrera_Leveraging_Visual_Attention_for_out-of-Distribution_Detection_ICCVW_2023_paper.html`.

Relevant software and/or external resources:

- The PyTorch implementation of our work can be found in `https://github.com/lcultrera/WildCapture`.

4.6.8 Relevance to AI4Media use cases and media industry applications

The method developed can contribute to UC1 (AI for Social Media and Against Disinformation), and specifically, Feature 1A (Detection/Verification of Synthetic Media). AI-generated images can be considered as OOD samples and detection systems for this data could be used by social media platforms to assess the authenticity of uploaded content. Moreover, this approach allows better quantify uncertaingy of transformer based classifier, thus improving explainability.

4.7 Addressing Limitations of State-Aware Imitation Learning for Autonomous Driving

Contributing partner: UNIFI

4.7.1 Overview

The most common approach to train autonomous agents is to exploit imitation learning, where an agent learns by replicating a policy. However, Imitation Learning (IL) has some limitations. Since capabilities are learned by behavioral cloning, IL models usually lack explicit causal understanding. Rather than rules, relations between patterns are learned, thus making the agent vulnerable to spurious correlations in the data. This phenomenon is known in the literature as *causal confusion* [62].

In particular, when training IL agents for automotive, there is evidence of a special case of causal confusion referred to as the *inertia problem* [63]–[65]. The inertia problem stems from a spurious correlation between low speed and no acceleration in the training data, making the driving agent likely to get stuck in a stationary state. As a consequence, when a state-aware agent halts (e.g. at a traffic light or in a traffic jam), it may not move again when it should. For state-awareness, here we refer to any source of information that can inform the agent about its halted state, such as a state variable, either explicitly modeled or implicitly inferred, that encodes velocity.

A second issue that limits the applicability of IL is the gap between offline and online driving capabilities [66], [67]. Codevilla *et al.* [67] showed that there is a low correlation between offline evaluation metrics (e.g. frame-wise Mean Squared Error in steer angle prediction) and the success rate in online driving benchmarks. In online driving, the output of the model influences future inputs, violating the independent and identically distributed assumption made by the learning framework [68]. Accumulation of small errors thus brings the vehicle into new states, never observed at training time [69]. Similarly to the inertia problem, this issue manifests itself the most in state-aware models: the more variables are observed by the model, such as ego-velocity or previous driving commands, the sparser the coverage of the training data gets, making it more likely to end up in under-represented configurations at driving time.

To summarize, IL agents suffer from ill-distributed training data that presents spurious correlations and domain shift compared to the test set. These issues make it particularly hard to train state-aware agents: using multiple input sources increases the chances of discovering unwanted correlations in the data or of observing under-represented inputs at inference time, for which the agent does not know how to act confidently [68]–[70]. We address these difficulties in training state-aware IL models.

Our IL agent is designed as a hierarchical transformer model with state token propagation. The vehicle's state is encoded in a special token of a vision transformer [50] and is enriched with new information at each stage of the architecture. At first, we predict whether the vehicle must stop or go, directly tackling *inertia*. This information is passed to the next stage which predicts the driving commands (namely steer, throttle, and brake). Finally, the model leverages a differentiable Command Coherency Module (CCM), encouraging the model to correctly bring the vehicle to the desired future state by generating non-conflicting controls. Such command is used only at training time and acts as a regularizer. Since our architecture is based on a transformer encoder [71], it heavily relies on attention. We leverage such attention to gain insights about what the model is focusing on to make its decisions (e.g., the vehicle's state or visual patterns), following the recent trend of designing explainable driving models [72]–[74].

Table 15. Failure rate due to inertia problem in Town01 - New weather of the NoCrash benchmark

	Train conditions		New weather	
Task	Single Stage	Ours	Single Stage	Ours
Empty	17%	**1%**	40%	**8%**
Regular	9%	**1%**	20%	**2%**
Dense	16%	**6%**	22%	**4%**

4.7.2 Method

Imitation Learning (IL) trains an agent by observing a set of expert demonstrations to learn a policy [75]. In the simplest scenario, IL is a direct mapping from observations to actions [76]. In automotive, the expert is a driver, the policy is *"safe driving"* and the demonstrations are a set of *(frame, driving-controls)* pairs. In this work, we address Conditional Imitation Learning (CIL), a declination of imitation learning where the policy must reflect a given high-level command, such as *turn right* or *follow lane*. As in prior work (e.g. [73], [77], [78]), we divide our architecture into multiple branches, with separate heads learning command-specific policies. However, differently from prior work, we structure our model as a hierarchy of stages, each of which is dedicated to addressing different aspects of driving.

The proposed model is state-aware, in the sense that it takes as input the speed and the steer, acceleration and brake values predicted at the previous time step. In principle, informing the model of the current state of the vehicle could ensure temporal smoothness and coherency in the driving policy (i.e., the predicted driving controls). In practice, this makes the model vulnerable to spurious correlations in the data, bringing out the *inertia problem*. To address this issue, we propose a multi-stage transformer model with state token propagation. We feed the vehicle state to the model as a special token of a vision transformer (ViT) [50]. Operationally speaking, the state token fulfills the same role as the *[CLS]* (classification) token in standard ViTs. However, by enclosing vehicle measurements we can inject information into the model and let it correlate to relevant spatial features via self-attention. After each layer, the state token is enriched with spatial information and is decoded into coarse-to-fine driving commands, depending on the stage. The coarser of such commands is a decision on whether the vehicle should stop or go, thus explicitly addressing inertia.

Injecting the state token into the model has the additional benefit of enabling data augmentation on the state values itself, addressing what is arguably the biggest limitation of imitation learning, i.e., the inability to perform well in previously unseen states [66] that is also responsible for the gap of accuracy between offline and online driving. We also introduce a regularizer that ensures coherency in the generated driving commands. This is different from similar solutions adopted in prior works, where speed is predicted to reduce inertia [63], but here we use it to reduce online-offline evaluation gap.

4.7.3 Results

Ex-Post Explainability Tab. 15 indicates that, despite addressing in a very effective way the inertia problem, the model still suffers from a few inertia failures. We exploit the Ex-post Semantic Explainability approach presented earlier to inspect 50 episodes of the *NoCrash* benchmark[63] where the inertia problem still occurs at traffic lights. In 56% of the cases where the vehicle is

Table 16. % of detected entities in features when the vehicle is stopped at green traffic light on NoCrash.

Cause of failure	Percentage of detection
Red Traffic light	56%
Pedestrian crossing	18%
Vehicle obstruction	3%
Tot	**77%**

stuck at a green light, the k most similar features to the attended one contain a red traffic light, in 18% a pedestrian crossing, and in 3% a vehicle (Tab. 16).

In Fig. 17, we show the top 10 nearest samples of the image region with the highest attention value (first transformer stage). The first two rows show failure cases: the model correctly focuses on the traffic light but although it is green, the model maps it in a region of the latent space densely populated by red traffic lights. We also show a sample of correct driving, where the vehicle accelerates as soon as the light turns green: retrieved images all depict green lights. This suggests that what may appear as inertia might instead be confused with a failure of the backbone that mistakenly "hallucinates" halt cues.

4.7.4 Relevant Resources and Publications

Relevant publications:

- Cultrera, L., Becattini, F., Seidenari, L., Pala, P. and Del Bimbo, A., 2023. Addressing Limitations of State-Aware Imitation Learning for Autonomous Driving. IEEE Transactions on Intelligent Vehicles. [79].
  Arxiv record: `https://arxiv.org/abs/2310.20650`.

4.7.5 Relevance to AI4Media use cases and media industry applications

The proposed method is connected to the task of automated cinematography in T5.2, for which autonomous agents are required to safely explore and plan in order to automatically produce new media. For these tasks it is important to develop specific methods to explain the behavior and performance of such state-aware agents trained which are typically trained off-line and then executed in non iid scenarios. Therefore this contribution is useful for UC3 (AI in Vision - High quality Video Production and Content Automation) since it can be used for improving reliability and explainability of autonomous state-aware agents.

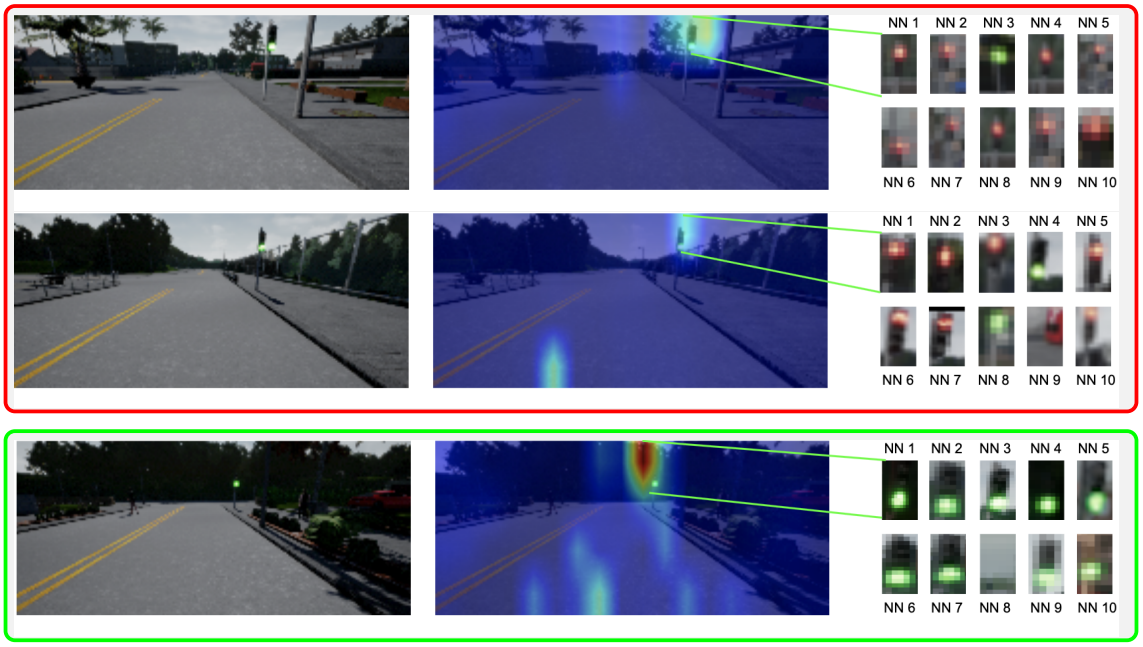


Figure 17. Top 10 nighbors for the highest scoring attention after a traffic light turns green. We show examples of both successful crossing of the traffic light (framed in green) and failed due to red light "hallucination" (framed in red).

5 AI Privacy (Task 4.4)

Data is the new oil. Never before, so much personal data has been collected and evaluated. Never before, so many technologies have been available to analyze the data and combine this into new insights.

All these advances in Artificial Intelligence (AI) have the important downside that breaching individuals' privacy at scale is also as easy as never before. The European legislation reacted with the General Data Protection Regulation (GDPR) regulating what is allowed and what is not. However, this suggests a trade off between AI performance and privacy. But instead of drawing things black and white, making data privacy a natural enemy of progress, it is important to take a look at technologies that allow the processing of personal data without sacrificing sensitive information held by individuals and organizations. More often than not, cleverly anonymised data is enough.

Contributions towards the **AI Privacy** task (T4.4) during the last year include work on (i) examining the true privacy benefits of federated learning, with reference to the strong trust models that are inherent to its present uses (Section 5.1), and (ii) securing federated learning using fully homomorphic encryption (Section 5.2).

5.1 Re-evaluating the Privacy Benefit of Federated Learning

Contributing partner: IBM

5.1.1 Introduction

The attractiveness of Federated Learning (FL) from a privacy-preserving point of view is that it allows for the training of machine learning models without the need for potentially sensitive data to be stored remotely. Within the FL protocol, training data remains on premises; training occurs locally, and only the updates to the model's parameters are shared to a central authority. They then aggregate all the updates from FL participants and disseminate the resulting model back to the participants. This process then repeats until the model is judged to have converged.

We observe that a large amount of trust is required on the part of the FL participant. Specifically, clients need to *trust* the central authority to carry out FL in an honest manner and not to undertake in malicious attempts at private data recovery. Several works [80]–[89] have been proposed that show that a malicious central authority can subvert the privacy benefit of FL by performing attacks against FL that allow for data reconstruction.

In this work, we address the question:

> *Given the large amount of trust required for Federated Learning to be truly private, why not just send the raw data to the central authority, and trust they'll use it only to train a model?*

Centralised training with ephemeral data storage (*i.e.* stored only in RAM and deleted immediately after use) appears to be, from a privacy point of view, identical to FL. Both methodologies require the 3rd party orchestrating training to conduct itself in an honest manner.

In this work, we provide a discussion of three aspects of FL that can affect the level of privacy, namely model architecture, the levels of trust involved, and FL system implementations, through the lens of a real world FL system: the Next Word Prediction model present in Google's Gboard

virtual keyboard[3]. We argue that these aspects form the foundations of a possible roadmap of future research into FL and privacy.

5.1.2 Model Architecture Affects Privacy

The idea of modifying the architecture of the model in order to aid data reconstruction has been investigated by several works [90]–[94]. Subsequently, we look at how an innocuous change to the architecture of Gboard's next word prediction model can result in serious privacy violations; allowing an adversary to reconstruct both the words and sentences typed by the user.

Gboard's Architecture and Privacy Gboard is a virtual keyboard application available for both Android and iOS devices. Importantly, Gboard uses FL to train its next word prediction model. This is a word level long Short-Term Memory (LSTM) language model, predicting the probability of the next word given what the user has already typed into the keyboard.

The final layer of language model architectures typically includes a fully connected layer (with or without a bias) that converts the previous layer activations into a probablity distribution over the words. The inclusion of the bias term $\mathbf{b}$ is a design choice. We find that when a bias term is present, a trivial attack can be instantiated to recover the typed words by taking advantage of a key property of the gradients of the final, fully connected layer. For an example sequence $(\mathbf{x}^{(1)}, c_1), (\mathbf{x}^{(2)}, c_2), ..., (\mathbf{x}^{(T)}, c_T)$ of T total timesteps, where $\mathbf{x}^{(t)} \in \mathbb{R}^D$ is the current word embedding, and c_t is the next word, we have the total loss function $L = \sum_{t=1}^{T} \ell_t(\mathbf{x}^{(t)}, c_t)$, where $\ell_t = -\log \frac{e^{\mathbf{z}^{(t,c_t)}}}{\sum_j^V e^{\mathbf{z}^{(t,j)}}}$, is cross entropy loss at timestep t. The vector $\mathbf{z}^{(t)} = \mathbf{h}^{(t)}W + \mathbf{b}$ is the model's raw logit output at timestep t, a vector of length V. We use the notation $\mathbf{z}^{(t,i)}$ to index the output vector, and $\mathbf{h}^{(t)}$ is the previous layer activation at timestep t. Then, we have the derivative of the loss at timestep t w.r.t the i-th neuron bias $\mathbf{b}_i$, $\frac{\partial L}{\partial \mathbf{b}_i} = \sum_{t=1}^{T} \frac{\partial \ell_t}{\partial \mathbf{z}^{(t,i)}}$. As shown in [95], the sign of the derivative of the cross entropy loss w.r.t to the outputs is negative only if $i = c_t$ i.e. token i was typed. Thus the index of the negative bias gradients reveals the typed words of the users participating in the given FL round. This information is a privacy breach in and of itself, and can be used to mount further attacks to reconstruct original sentences, exploiting the generative nature of language models [96], [97].

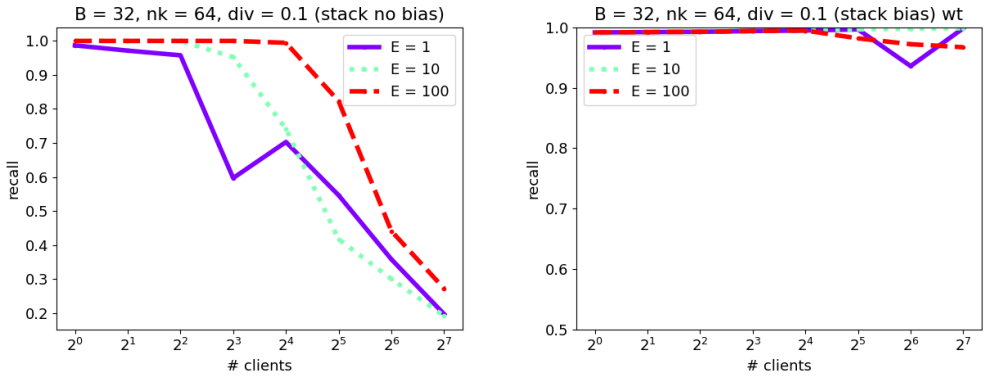


Figure 18. Effect of model architecture on the performance of a word reconstruction attack against Gboard updates (see later for further details). Simple changes like adding a bias to the final layer allow for perfect recall of the typed words. Here, each client trains their local model using 64 sentences, a batch size of 32. We plot the word recall results for a varying number of local epochs.

[3] https://play.google.com/store/apps/details?id=com.google.android.inputmethod.latin

Figure 18 shows how a simple change such as removing the bias can drastically impact the attack's performance. In our experiments, each client trains their local LSTM model on sentences taken from the stack exchange data dump [98] following the `FederatedAveraging` algorithm described in [99]. The server, then mounts the attack described above on the difference between the initial shared model and the final aggregate. This attack can also be performed on the gradients of the final layer weight matrix W, as shown in [95]. However, they consider batch sizes of only 1 example. Crucially, the attack degrades when a greater number of clients is used in Secure Aggregation (SA) when no bias is present, however, when a bias is present, SA appears to provide no extra benefit in terms of privacy. Ultimately, architectural changes can both enhance and degrade privacy. Gupta *et al.* [96] propose the use of pre trained word embeddings, eliminating the gradients for the final layer. Additionally, removing a bias from the final layer can help in preventing this type of attack.

5.1.3 Verifiable FL Implementations

Production implementations of FL algorithms are embedded within larger software systems that include telemetry, remote configuration, device authentication/attestation etc. It is, of course, the privacy of the system as a whole that is of concern to users. We show below that poor implementations can easily allow de-anonymisation of devices and users as well as creating new potential channels for attacks. Our GBoard measurement study also highlights that public documentation and support for independent evaluation of developed apps by the FL community is important both to verify privacy claims and to build confidence in users that apps employing FL are indeed safe to use.

5.1.4 Vulnerabilities in the Gboard FL Implementation

In the subsequent sections, we provide examples of ways one can exploit the telemetry sent by Gboard's FL implementation to de-anonymise users and bypass aggregation. We monitor the traffic sent by an android[4] device, using a man-in-the-middle attack implemented using the `mitmproxy` [100] tool suite.

Telemetry Allows for De-Anonymisation. In our experiments, we observe that the `eligibility_eval_checkin_request` messages regularly sent by the Google GBoard and Google Messages apps on the handset to Google server `federatedml-pa.googleapis.com` include Google SafetyNet[5] device attestation data. The data sent can readily be used for device fingerprinting. When the aim of Differential Privacy (DP) is to ensure that is difficult to determine whether a user contributed to the data, the possibility de-anonymisation removes this guarantee, making it trivial to establish whether a particular user did or did not contribute to model training. These sort of side channel attacks can erode any privacy guarantees provided by FL hardened with SA and DP.

Aggregation Bypass and Population Control. In the Google FL protocol, handsets can execute an `eligibility_eval` plan and return the response. This includes an initial checkpoint (model parameter values), a local dataset to use, and a TensorFlow graph the handset should execute to update the checkpoint and generate a response. This response is not aggregated and is sent by the handset to the server in subsequent FL `checkin_request` messages.

[4]Hardware and software used: Google Pixel 4a, Google Play Services ver. 22.09.20, Google Gboard ver. 12.4.06, rooted using Magisk. Device Settings: following factory reset, settings are left at their defaults.

[5]See, e.g., `https://developer.android.com/training/safetynet/attestation`

5.1.5 Reducing the Need for Trust

An honest FL participant may believe that they have preserved their privacy by using FL, yet their data can be readily recovered and tied to them by the coordinating server. This is a direct result of the inordinate amount of trust clients are asked to place in the FL system. Clients must *trust* that the co-ordinating server is faithfully carrying out the FL protocol, clients must *trust* that the other clients are genuine, clients must *trust* that the cryptography behind the Public Key Infrastructure (PKI) in SA is not compromised, clients must *trust* the model architecture has not been designed maliciously, etc.

When the FL system and the model being trained are both operated by a reputable organisation then perhaps such trust can be justified. However, it requires strong governance and oversight of that organisation and the avoidance of potential conflicts of interest (such as the organisation also being a consumer of user data for analytics, advertising etc). When such a level of trust exists then it also begs the question of why not simply send raw client data to the central server and trust that it is stored ephemerally and only used for the purposes of model training in combination with data from sufficiently many other users i.e., the added privacy value of FL seems rather small.

When multiple parties are involved in the FL system, such as with Federated-Learning-as-a-Service (FLaaS) [101], establishing sufficient trust seems much harder. For example, suppose an organisation operates FLaas for mobile apps. Then the models to be trained by FL on private client data may be supplied and used by multiple different app developers with a broad geographic spread and different regulatory regimes. As we already know from mobile app stores, users then have only a limited ability to establish developer bona fides and even powerful gatekeepers such as Google and Apple have difficulty regulating developer behaviour. Hence, even when the FLaas provider is trusted, the overall FLaas system need not be trustworthy.

We note, however, that some degree of trust is likely to be asked of FL users. Trying to ensure user privacy when the FL service is actively malicious is probably a hopeless endeavour – the server may insert synthetic devices, manipulate model weights, architecture, compromise the PKI, and training process actively during training and it seems hard to defend against all of these while still providing a useful FL service. The need is to greatly reduce the level of trust asked of users, and thereby provide a better privacy risk-benefit trade-off to them.

5.1.6 Relevant Resources and Publications

Relevant publications:

- Re-evaluating the Privacy Benefit of Federating Learning, Mohamed Suliman, Douglas Leith, Anisa Halimi. 1st Workshop on Advancements in Federated Learning at ECML-PKDD 2023 [102].

Relevant software and/or external resources:

- The implementation of our work can be found in `https://github.com/namilus/nwp-fed learning`.

5.1.7 Relevance to AI4Media use cases and media industry applications

Federated Learning (FL) has been proposed as a way to do collaborative learning while preserving privacy. What this means for media companies who are wary of sharing their data with 3rd parties in order to train a model is that they may do so without fear of potential privacy or copyright infringement, as their data remains on premises. We have seen that, in practice, FL requires the 3rd parties to place a large amount of trust in one another in order to fully realise the benefits

of training models in this way. Our work addresses this often overlooked aspect of truly private FL, and encourages media companies who are thinking of participating in FL to carefully consider these questions of trust. Do they trust the design of the model and training objective? Do they trust the FL implementation is free from bugs? How can they reduce the required level of trust? These questions are vital to any real world instantiation of FL.

5.2 Securing Federated Learning for Audio Event Classification with Fully Homomorphic Encryption

Contributing partner: FhG-IDMT

We introduce a library designed to enhance Federated Learning with additional privacy guarantees by applying Fully Homomorphic Encryption to the model aggregation stage, thereby preventing the aggregator from accessing the unencrypted model parameters of the training participants.

5.2.1 Federated Learning

Federated learning is a machine learning approach where several models are trained across multiple end devices (clients) and are aggregated by a server into a global model. The aggregated model is then sent back to every client. Instead of exchanging training data, only the model parameters are exchanged. This eliminates the need to store large amounts of data in a single location and lets data holders retain control of said data. It was introduced in 2016 in [99] and has since been applied to variety of applications, including the field of audio classification.

The Federated Learning (FL) approach itself does not specify a particular aggregation method. In this work, we analyze the Federated Averaging (FedAvg) and the Federated Proximal (FedProx). The FedAvg algorithm proposed in [99] computes a weighted average of individual model parameters to produce a final global model.

The local model parameters are weighted with respect to the client's proportion of the data to ensure that the local model's impact is proportional to the amount of information it contributes to the training process. However, the final aggregated model might not perform as well for clients that contribute comparatively little data.

5.2.2 FLCrypt Experiments

FLCrypt is our FL framework, which we used for the experiments described below. It is largely based on open-source libraries. We used a custom version of Flower [103] for the FL setup and integrated the CKKS functionality of the TenSEAL library to encrypt client model updates before transmitting them to the server for aggregation. Moreover, we utilized the Hydra framework [104] to manage configurations for our experiments.

For our customized Flower version, we added an extra payload field to the instructions and response classes for the client and server to make it easier to send the model parameters back and forth. For the encryption of model parameters, we utilize the CKKSVector class of TenSEAL. For this, the model parameters have to be flattened, meaning that the original shape has to be restored for the model update. This is done to decrease the size of the encrypted model. The FL functionality is provided by Flower with the only further changes to the framework are adjustments to the computations of the server-side aggregation in order to handle the encrypted values.

The experiments were all conducted using FLCrypt on the metal ball data set described by [105]. The data set contains audio recordings of metal balls rolling down a steel slide as a part of a bigger track. It comprises three classes that correspond to different surface coatings of the metal balls,

one of which is scratched. The audio was recorded using a low-cost microphone and the steel slide was surrounded with a casing to dampen background noise. This data set was originally created to improve machine learning applications for industrial sound analysis as there is a scarcity of usable data in that field. In this context, the data set can be used to develop industrial acoustic quality control applications based on material conditions with emphasis on fault detection. According to the original paper, the dataset is relatively easy for the given classification task. A Deep Neural Network (DNN) baseline accuracy close to 99 % was reported by authors. Due to its simplicity, it provides a realistic initial target for Internet of Things (IoT) applications with limited computation power of the edge devices. We use it as a first step in showcasing the effectiveness of applying Homomorphic Encryption (HE) to audio event detection with the possibility of extending our approach to more complex use cases in the future.

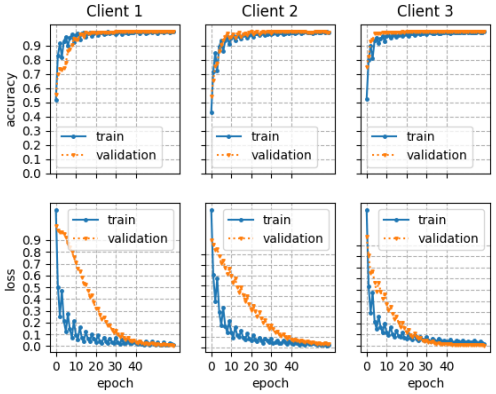
For our FL experiments, we split the original balanced data set into three separate client partitions, consisting of 450 training and 57 test samples each. We ensure that each partition comprises a balanced number of examples corresponding to only two of the three classes, with the missing one different for each partition. The test sets were all evenly split into 19 samples from each of the three classes.

The FL runs with HE consistently show a similar accuracy as the unencrypted baseline. The FedAvg strategy as well FedProx get high accuracies of over 99% on the local client test sets. The average training loss and accuracy across three runs can be seen in Figure 19a. Compared to each other, the accuracies of FedAvg and FedProx are the same but model converges much quicker for FedProx as shown in Figure 19b. Overall, the unencrypted FedAvg baseline model routinely achieves an accuracy of 1 shown in Figure 19c. Therefore, the drop in performance when using HE is marginal. This is in line with previous applications of HE to FL in other use cases of medical image classification [106] or network traffic prediction [107]. The approach in [108] also achieves an accuracy of over 99% for identifying malicious traffic in an IoT network. This indicates a general viability of using HE in the context of FL independent from the specific use case. Furthermore, [106] used an exact HE scheme which does not showcase a notable improvement in terms of accuracy over our use of the CKKS scheme. This indicates that the approximate nature of CKKS does not hurt model performance in practical applications while offering more flexibility regarding its use.

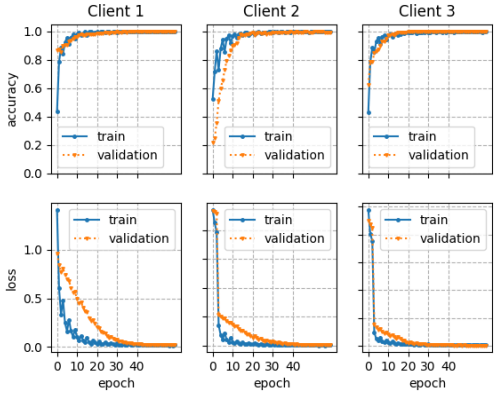
The difference in runtime performance between the HE and the unencrypted setup is significant. Baseline total execution time is 27 seconds on average across three runs of FL. After incorporating HE into the training, the overall runtime increases to around 48 seconds for both strategies. A more detailed runtime breakdown is provided in Table 17. For the individual processes, the values are given as the average execution time of that process across 20 rounds of FL training. This includes all clients and the server. Plaintext serialization and deserialization in the baseline setup are carried out by Flower library and are not described here.

It can be observed that the runtime increase due to the encryption is not symmetric. The encryption and serialization process take significantly longer than their counterparts. There is also a large increase in the server's aggregation time. In general, it can be observed that the encryption time is the biggest factor in the overall time increase on the client side. On the server side, the deserialization takes longer than serialization although the reason for this is unclear.

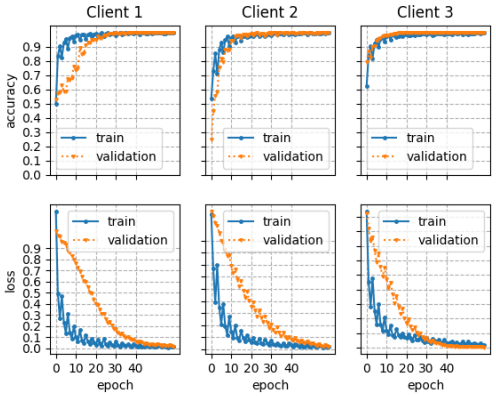
In [106], the run time for 128-bit security and three clients was around $5,000$ seconds using the BFV scheme of SEAL which is significantly higher than our result. However, since the size of the model was not specified, it is unclear how much of that difference is due to the different schemes. In [107], the authors found a runtime increase from around 0.07 seconds for FL with plaintext to around 15.5 seconds when using the CKKS implementation in TenSEAL for a model with two layers with 400 neurons respectively. Unfortunately, the number of clients is not specified. Still, compared to our model with $2,745$ parameters this roughly scales with respect to the number of



(a) Accuracy and loss of global model for the three clients using HE for FedAvg.



(b) Accuracy and loss of global model for the three clients using HE for FedProx.



(c) Baseline accuracy and loss for FedAvg.

Table 17. Comparison of runtime for different processes between encrypted and unencrypted runs.

Process	Time unencrypted (sec)	Time encrypted (sec)
Encryption	-	0.31
Decryption	-	0.031
Serialization	-	0.07
Deserialization	-	0.02
Server aggregation	0.00066	0.07
Complete run	27	48

Table 18. Data size in relation to number of parameters.

Input array	Size original (KB)	Size encrypted (KB)	number of parameters	Ratio
(384, 3)	9.3	301	1152	32.3
(768, 3)	18.6	301	2304	16.2
(1536, 3)	37	601	4608	16.2
(3072, 3)	74	902	9216	12.2
(6144, 3)	148	1504	18432	10.2
(12288, 3)	295	2708	36864	9.2
(24576, 3)	590	5414	73728	9.2
(49152, 3)	1180	10828	147456	9.2
(98304, 3)	2359	21658	294912	9.2

Table 19. Projected size of well-known neural networks.

Input array	Size original (MB)	Size encrypted (MB)	number of parameters (Million)
ResNet50	98	901	25.6
ResNet101	171	1573	44.7
ResNet152	232	2134	60.4
VGG16	528	4858	138.4
VGG19	549	5051	143.7
ConvNeXtSmall	192	1766	50.2
ConvNeXtBase	339	3110	88.5
ConvNeXtLarge	755	6946	197.7
MobileNet	16	147	4.3
NASNetMobile	23	212	5.3
EfficientNetB0	29	267	5.3
EfficientNetB7	256	2355	66.7

model parameters.

To analyze the development of the encrypted model size, we examine a randomly initialized $(768, 3)$ array of model parameters which corresponds to the largest layer of our model with entries ranging from zero to one. Unencrypted, it has a size of 18.6 KB. After encryption and serialization, its size grows to 301 KB. This marks an increase compared to the original array size by a factor of 32.2. For larger number of model parameters, the size increase settles on a ratio of 9.2. This matches the linear runtime increase in relation to the number of model parameters found in [109]. The encryption increases the size of the model by an order of magnitude. In total, we obtain an encrypted size of 4.2 MB for our model compared to an original size of 12.6 KB. This is because every encrypted model weight has a size of at least 0.3 MB, leading to a disproportionate increase in size for small models.

An overview of the ciphertext size for different numbers of parameters can be found in Table 18. From this, we also extrapolate the encrypted size of a selection of well-known neural networks shown in Table 19.

5.2.3 Conclusion

We have successfully incorporated HE into FL for audio event classification for two different aggregation strategies and showed the effectiveness of our approach. Additionally, we have shown that FedProx retains advantages over FedAvg under HE. We have identified key reasons for a substantial runtime increase when using HE for FL that impede its scalability. We identified the number of model parameters as the main factor that determines the size of the encrypted data and analyzed how the encryption time develops in relation to it. This already showcases the utility of HE in

certain settings, especially in our use case of sound classification where model size is moderate. While our experiments lead us to be optimistic, more work needs to be done analyze and improve the viability for HE in other practical settings since the linear increase relative to the number of model parameters might become infeasible for large model sizes, especially due to the rise of large-scale machine learning and the increasing prevalence of edge devices. Furthermore, the possibility of side-channel attacks needs to be further investigated and prevented before deployment in real scenarios.

5.2.4 Relevant Resources and Publications

Relevant publications:

- Fuhrmeister et al.: FLCrypt – Secure Federated Learning for Audio Event Classification using Homomorphic Encryption (accpepted for IEEE International Symposium on the Internet of Sounds 2024)

Relevant software and/or external resources:

- You can request evaluation access to FLCrypt by contacting us via `https://www.idmt.fraunhofer.de/en/contact.html`.

5.2.5 Relevance to AI4Media use cases and media industry applications

While privacy is a feature of AI applications that only a few use cases will go without, the proposed approach deals with privacy within Federated Learning systems. Regarding AI4Media, there is no Use Case directly dealing with Federated Learning. Regarding the broader media industry, the outlook of not having to share private data (being it usage, user or content data), and therefore avoiding all the practical hassles of data exchange (usage rights, data exchange contracts, data privacy laws, ...) is so promising, that there will be real industry applications for Federated Learning. On that premise, applications that improve the privacy of Federated Learning are worth researching and will be relevant in the future as they are already in non-media domains such as medicine or industrial applications. The recent break-through moment of LLMs and generative models will also yield new use cases, where a decentralized, federated, training without giving data away is required.

6 AI Fairness (Task 4.5)

As machine learning models are fast becoming critical components of every decision making process essential for our society (mortgage lending, prison sentencing etc), it becomes crucial to guarantee that these models do not privilege specific groups or individuals at the disadvantage of others. These models are constructed upon the statistical analysis and properties of training data, which may contain biases due to existing prejudice and/or inaccurate sampling. Hence, if left unchecked unwanted biases can emerge from these models with significant societal consequences.

AI Fairness is typically evaluated either on a group or individual level. When addressing group fairness, a population is divided into groups based on a set of protected attributes (gender, ethnicity, etc.). A fair ML model within this context is a model which seeks some statistical measure to be equal across such groups. On the other hand, when addressing individual fairness, ML models seek to treat individuals similarly regardless of their protected attributes.

Algorithms and metrics designed to address biases in ML models can operate on the training data itself as well as on the trained model. Moreover, they can also occur at various points in the machine learning lifecycle whether at a pre-processing, in-processing, or post-processing phase. T4.5 seeks to apply AI Fairness algorithms and metrics at group and individual levels and at various points in the AI lifecycle.

Contributions towards the **AI Fairness** task (T4.5) include work on (i) ensemble post-processing of LLMs to improve fairness (Section 6.1), and (ii) bias detection in text-to-image generative models (Section 6.2).

6.1 FairSISA: Ensemble post-processing to improve fairness of unlearning in LLMs

Contributing partner: IBM

6.1.1 Overview

Training Large Language Models (LLMs) is a costly endeavour in terms of time and computational resources. The large amount of training data used during the unsupervised pre-training phase makes it difficult to verify all data and, unfortunately, undesirable data may be ingested during training. Re-training from scratch is impractical and has led to the creation of the *unlearning* discipline where models are modified to "unlearn" undesirable information without retraining. However, any modification can alter the behaviour of LLMs, especially on key dimensions such as *fairness*. This work examines the interplay between unlearning and fairness for LLMs.

6.1.2 Preliminaries

Sharded, Isolated, Sliced, and Aggregated (SISA) Training: SISA [110] is an exact unlearning method that reduces the computational overhead associated with retraining from scratch. The SISA framework randomly divides the training dataset $\mathcal{D}$ into S disjoint shards $\mathcal{D}_1, \ldots, \mathcal{D}_S$ of approximately equal size. During training, for each shard $\mathcal{D}_k$, a *constituent model*, denoted as M_k, is trained. At inference time, S individual predictions from the constituent models are aggregated, typically, through majority voting. When one or more data samples need to be unlearned, only the constituent models corresponding to the shards that contain the data sample(s) are retrained.
Fairness for Toxic Text Classification: We consider the task of toxic text classification and measure model bias in terms of *group fairness* [111] by following the setup in [112]. In particular,

we consider certain topics, such as religion or race, as sensitive. If a text sample mentions one of the sensitive topics (e.g., religion), we say that it belongs to a *sensitive group*; otherwise, to the complementary group (no religion). While there are several notions of group fairness, e.g., demographic parity (see [113], [114]), we consider the notion of *Equalized Odds (EO)* [115]. Essentially, equalized odds requires that the model output conditioned on the true label to be independent of the sensitive attribute. More formally, let Y denote the true label (e.g., toxic text), X denote the features, and A denote the sensitive attribute (e.g., religion or race). Let $\hat{Y} = f_{\mathbf{w}}(X, A)$ be the model output, denoted as the *predictor*. Equalized odds requires that the model predictor $\hat{Y}$ has equal *true positive rates* and *false positive rates* across the privileged and unprivileged groups, satisfying the following constraint:

$$\Pr\left(\hat{Y} = 1 \mid A = 0, Y = y\right) = \Pr\left(\hat{Y} = 1 \mid A = 1, Y = y\right), \quad y \in \{0, 1\}. \tag{13}$$

Baseline Post-Processing Method for Fairness: To improve the model fairness without retraining, we explore the use of post-processing methods. We build on the post-processing method proposed in [115], which we denote as *HPS* (using the last names of the authors).

The HPS method constructs a *derived predictor* $\tilde{Y}$, which only depends on the predicted label $\hat{Y}$ and the sensitive attribute A, and satisfies equalized odds while minimizing classification loss. Specifically, let $\ell : \{0, 1\}^2 \to \mathbb{R}$ denote a loss function that takes a pair of labels and returns a real number. Let us define $p_{ya} = \Pr\left(\tilde{Y} = 1 \mid \hat{Y} = y, A = a\right)$. Then, the HPS method constructs $\tilde{Y}$ by solving the following optimization problem:

$$\begin{aligned}
\min_{p_{y\tilde{y}}} \quad & \mathbb{E}\left[\ell(\tilde{Y}, Y)\right] \\
\text{s.t.} \quad & \Pr\left(\tilde{Y} = 1 \mid A = 0, Y = 0\right) = \Pr\left(\tilde{Y} = 1 \mid A = 1, Y = 0\right), \\
& \Pr\left(\tilde{Y} = 1 \mid A = 0, Y = 1\right) = \Pr\left(\tilde{Y} = 1 \mid A = 1, Y = 1\right), \\
& 0 \le p_{y\tilde{y}} \le 1.
\end{aligned}$$

We denote the derived predictor obtained by solving the above optimization problem as $\text{HPS}(\hat{Y})$. Next, we adapt the HPS method to design post-processing methods for the ensemble of models produced by SISA.

6.1.3 FairSISA: Ensemble Post-Processing for SISA

Let $\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_S$ denote the predictions from the SISA constituent models. We consider three ways to perform post-processing for SISA.

Aggregate then post-process: The most natural way to apply post-processing to SISA is after aggregating the predictions from the constituent models. We focus on the majority voting aggregation rule since it is demonstrated to perform well [110]. We denote majority voting as

$$\textsc{Maj}\left(\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_S\right) = \arg\max_{y \in \{0,1\}} n_y, \quad \text{where } n_y = \left|\{i \in [S] : \hat{Y}_i = y\}\right|. \tag{14}$$

Then, the derived predictor obtained by first aggregating and then post-processing can be defined as $\text{HPS}\left(\textsc{Maj}\left(\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_S\right)\right)$.

Post-process then aggregate: Another natural way to apply post-processing to SISA is to first post-process the label from each constituent model and then aggregate the post-processed predictions. Again, focusing on the majority voting aggregation rule, the derived predictor obtained by first post-processing and then aggregating can be defined as $\textsc{Maj}\left(\text{HPS}(\hat{Y}_1), \text{HPS}(\hat{Y}_2), \ldots, \text{HPS}(\hat{Y}_S)\right)$.

Ensemble post-processing: Instead of aggregating the predictions before or after post-processing with a specific aggregation rule (such as majority voting), we design a post-processing method that can inherently aggregate the predictions. In particular, we generalize the HPS optimization problem to handle ensemble predictions. Recall that $\ell : \{0,1\}^2 \to \mathbb{R}$ denotes a loss function that takes a pair of labels and returns a real number. For a length-S binary vector $\bar{y} \in \{0,1\}^S$ and $a \in \{0,1\}$, let us define $p_{\bar{y}a} = \Pr\left(\tilde{Y} = 1 \mid \hat{Y}_1 = \bar{y}_1, \hat{Y}_2 = \bar{y}_2, \ldots, \hat{Y}_S = \bar{y}_S, A = a\right)$. We propose an ensemble post-processing method that constructs $\tilde{Y}$ by solving the following optimization problem:

$$\begin{aligned} \min_{p_{y\tilde{y}}} \quad & \mathbb{E}\left[\ell(\tilde{Y}, Y)\right] \\ \text{s.t.} \quad & \Pr\left(\tilde{Y} = 1 \mid A = 0, Y = 0\right) = \Pr\left(\tilde{Y} = 1 \mid A = 1, Y = 0\right), \\ & \Pr\left(\tilde{Y} = 1 \mid A = 0, Y = 1\right) = \Pr\left(\tilde{Y} = 1 \mid A = 1, Y = 1\right), \\ & 0 \le p_{y\tilde{y}} \le 1. \end{aligned}$$

6.1.4 Evaluation

We perform an empirical evaluation using two state-of-the-art models (BERT, DistilGPT2) on a representative dataset (HateXplain). HateXplain [116] is a benchmark hate speech dataset that consists of 20K posts from Twitter and Gab. The dataset has fine-grained annotations for religion, race, and gender. We use coarse-grained groups as sensitive groups (e.g., mention of any religion), as opposed to the finer-grained annotations (e.g., Hindu), similar to [112]. This is because, for HateXplain, most subgroups account for significantly less proportion of the data, and there is considerable overlap between subgroups. We focus on race as a sensitive attribute. We combine the annotations for offensive and hate speech into one class of toxic text, similar to [112].

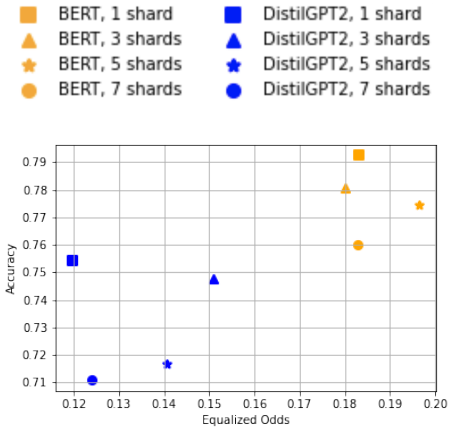
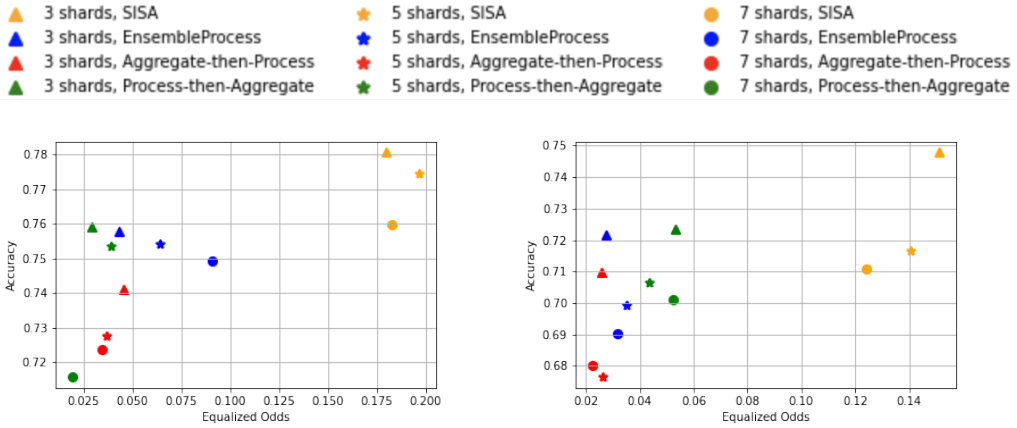


Figure 20. Accuracy-fairness trade-off for SISA framework.

First, we investigate how SISA training procedure influences the performance-fairness relationship by considering $S = 1, 3, 5$, and 7 shards. Note that $S = 1$ shard corresponds to the conventional single model fine-tuning paradigm. In Figure 20, we demonstrate the performance as measured by accuracy on the y-axis (higher accuracy is better) and the group fairness as measured by equalized odds (EO) on the x-axis (lower EO is better). We observe that, for both models, the accuracy decreases with the number of shards, which is consistent with the observation in

[110] for image-domain data. In contrast, EO values vary widely for different number of shards. Importantly, the SISA framework can indeed degrade the fairness (with higher EO values) for both models. For instance, for the DistilGPT2 model, SISA results in worse fairness (higher EO values) than the case of a conventional single model. These results strongly suggest that it is important to investigate bias mitigation methods for the SISA framework.



(a) BERT *(b)* DistilGPT2

Figure 21. Comparison of post-processing methods for SISA.

Next, we compare the three post-processing methods for bias mitigation from Section 6.1.3 for the SISA framework. In Figure 21, we plot accuracy vs. equalized odds (EO). Amongst the three methods, *Post-process then Aggregate* method generally achieves the best trade-off between the accuracy and EO, whereas *Aggregate then Post-Process* method generally achieves the worst trade-off between the accuracy and EO. The *Ensemble Post-Process* method, in general, achieves the highest accuracy for a moderate EO, which is consistent with the theory that the method is optimal in terms of accuracy (the objective function of the optimization problem (6.1.3)).

6.1.5 Relevant Resources and Publications

Relevant publications:

- S. Kadhe, A. Halimi, A. Rawat, and N. Baracaldo. "FairSISA: Ensemble Post-Processing to Improve Fairness of Unlearning in LLMs", Socially Responsible Language Modelling Research workshop in conjunction with NeurIPS (SoLaR), 2023 [117].
  Zenodo record: `https://zenodo.org/records/11581556`.

6.1.6 Relevance to AI4Media use cases and media industry applications

Our approach is relevant to various media industry use cases given its focus on removing sensitive or undesirable information via unlearning while ensuring that the model is still fair. This approach can help journalists and researchers have access to LLMs that generate high-quality content while ensuring that all subjects are treated fairly. Unlearning leads to better aligned models and improved performance by removing intentionally malicious, harmful or toxic data, or an undesirable subset of data, which is crucial in content moderation.

6.2 Open-set Bias Detection in Generative Models

Contributing partner: UNITN

6.2.1 Introduction

Text-to-Image (T2I) generation has become increasingly popular, thanks to its intuitive conditioning and the high quality and fidelity of the generated content [118]–[122]. Several works extended the base T2I model, unlocking additional use cases, including personalization [123], [124], image editing [125]–[128], and various forms of conditioning [129]–[131]. This rapid progress urges to investigate other key aspects beyond image quality improvements, such as their fairness and potential bias perpetration [132]–[134]. It is widely acknowledged that deep learning models learn the underlying biases present in their training sets [135]–[137], and generative models are no exception [132]–[134], [138].

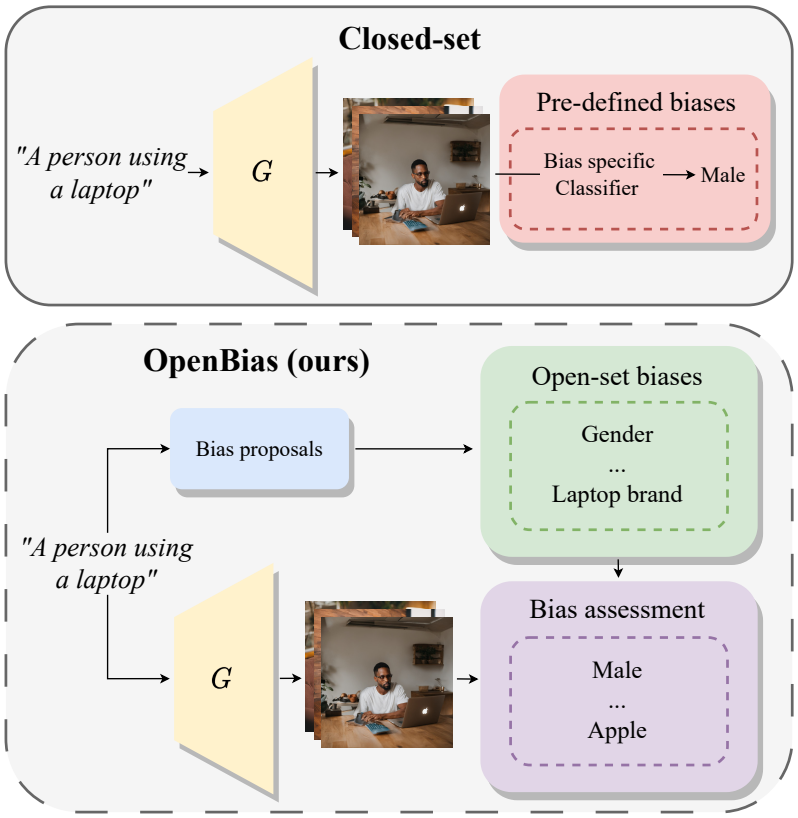


Figure 22. OpenBias discovers biases in T2I models within an open-set scenario. In contrast to previous works [132], [133], [139], our pipeline does not require a predefined list of biases but proposes a set of novel domain-specific biases.

Ethical topics such as fairness and biases have seen many definitions and frameworks [113]; defining them comprehensively poses a challenge, as interpretations vary and are subjective to the individual user. Following previous works [133], [140], a model is considered unbiased regarding a specific concept if, given a context t that is agnostic to class distinctions, the possible classes $c \in \mathcal{C}$ exhibit a uniform distribution. In practice, for a T2I model, this reflects to the tendency of the

generator to produce content of a certain class c (e.g., *"man"*), given a textual prompt t that does not specify the intended class (e.g., *"A picture of a doctor"*).

Several works studied bias mitigation in pre-trained models, by introducing training-related methods [141]–[144] or using data augmentation techniques [145], [146]. Nevertheless, a notable limitation of these approaches is their dependence on a predefined set of biases, such as gender, age, and race [133], [134], as well as specific face attributes [132]. While these represent perhaps the most sensitive biases, we argue that there could be biases that remain undiscovered and unstudied.

Considering the example in Fig. 22, the prompt *"A person using a laptop"* does not specify the person's appearance and neither the specific laptop nor the scenario. While closed-set pipelines can detect well-known biases (e.g., gender, race), the T2I model may exhibit biases also for other elements (e.g., laptop brand, office). Thus, an open research question is: *Can we identify arbitrary biases present in T2I models given only prompts and no pre-specified classes?* This is challenging as collecting annotated data for all potential biases is prohibitive.

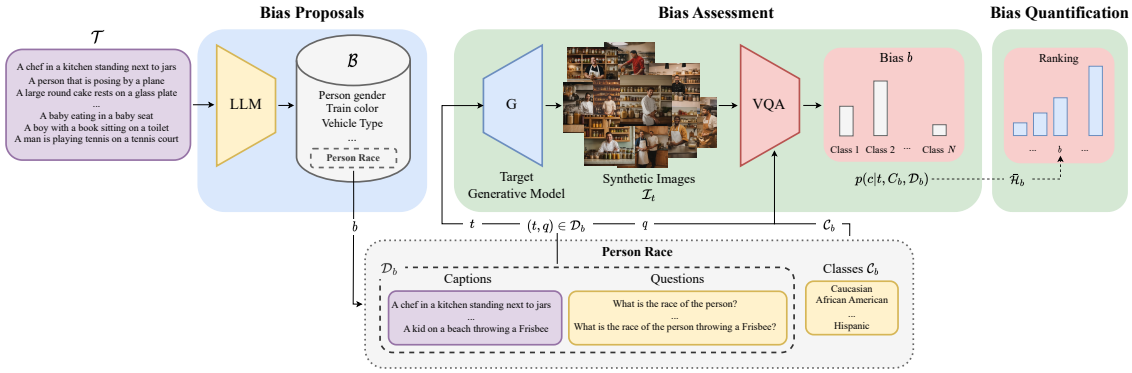


Figure 23. OpenBias pipeline. Starting with a dataset of real textual captions ($\mathcal{T}$), we leverage a Large Language Model (LLM) to build a knowledge base $\mathcal{B}$ of possible biases that may occur during the image generation process. In the second stage, synthesized images are generated using the target generative model conditioned on captions where a potential bias has been identified. Finally, the biases are assessed and quantified by querying a VQA model with caption-specific questions extracted during the bias proposal phase.

"A train zips down the railway in the sun"

"A photo of a person on a laptop in a coffee shop"

"A cop riding a horse through a city neighborhood"

Figure 24. Novel biases discovered on Stable Diffusion XL [122] by OpenBias.

6.2.2 Methodology

Toward this goal, we propose *OpenBias* (see Figure 23), the first pipeline that operates in an *open-set scenario*, enabling to identify, recognize, and quantify biases in a specific T2I model without constraints (or data collection) for a specific predefined set. Specifically, we exploit the multi-modal nature of T2I models and create a knowledge base of possible biases given a collection of target textual captions, by querying a Large Language Model (LLM). In this way, we discover specific biases for the given captions. Next, we need to recognize whether these biases are actually present in the images. For this step, we leverage available Visual Question Answering (VQA) models, directly using them to assess the bias presence. By doing this, we overcome the limitation of using attributes-specific classifiers as done in previous works [132], [133], [147], which is not efficient nor feasible in an open-set scenario. Our pipeline is modular and flexible, allowing for the seamless replacement of each component with newer or domain-specific versions as they become available. Moreover, we treat the generative model as a *black box*, querying it with specific prompts to mimic end-user interactions (i.e., without control over training data and algorithm). We test OpenBias on variants of Stable Diffusion [121], [122] showing human-agreement, model-level comparisons, and the discovery of novel biases.

6.2.3 Experiments

6.2.3.1 Datasets. We study the bias in two multimodal datasets Flickr 30k [148] and COCO [149]. Flickr30k [148] comprises 30K images with 5 caption per image, depicting images in the wild. Similarly, COCO [149] is a large-scale dataset containing a diverse range of images that capture everyday scenes and objects in complex contexts. We filter this dataset, creating a subset of images whose caption contains a single person. This procedure results in roughly 123K captions. Our choice is motivated by building a large subset of captions specifically tied to people. This focus on the person-domain is crucial as it represents one of the most sensitive scenarios for exploring bias-related settings. Nevertheless, it is worth noting that the biases we discover within this context extend beyond person-related biases to include objects, animals, and actions associated with people.

Figure 25. Novel person-related biases identified on Stable Diffusion XL [122] by OpenBias.

Model	Gender		Age		Race	
	Acc.	F1	Acc.	F1	Acc.	F1
CLIP-L [151]	91.43	75.46	58.96	45.77	36.02	33.60
OFA-Large [152]	**93.03**	83.07	53.79	41.72	24.61	21.22
mPLUG-Large [153]	**93.03**	82.81	61.37	52.74	21.46	23.26
BLIP-Large [154]	92.23	82.18	48.61	31.29	36.22	35.52
Llava1.5-7B [155], [156]	92.03	82.33	66.54	62.16	55.71	42.80
Llava1.5-13B [155], [156]	92.83	**83.21**	**72.27**	**70.00**	**55.91**	**44.33**

Table 20. VQA evaluation on the generated images using COCO captions. We highlight in gray the chosen default VQA model.

6.2.3.2 Quantitative Results Our open-set setting harnesses the zero-shot performance of each component. As in [133], we evaluate OpenBias using FairFace [150], a well-established classifier fairly trained, as the ground truth on gender, age, and race.

Model	**Flickr 30k** [148]			**COCO** [149]		
	gender	age	race	gender	age	race
Real	0	0.032	0.030	0	0.041	0.028
SD-1.5 [121]	0.072	0.032	0.052	0.075	0.028	0.092
SD-2 [121]	0.036	0.069	0.047	0.060	0.045	0.105
SD-XL [122]	0.006	0.028	0.180	0.002	0.027	0.184

Table 21. KL divergence (↓) computed over the predictions of Llava1.5-13B and FairFace on generated and real images.

Agreement with FairFace We compare the predictions of multiple SoTA Visual Question Answering models with FairFace. Firstly, we assess the zero-shot performance of the VQA models on synthetic images, performing our comparisons using images generated by SD XL. The evaluation involves assessing accuracy and F1 scores, which are computed against FairFace predictions treated as the ground truth. The results are reported in Table 20. Llava1.5-13B emerges as the top-performing model across different tasks, consequently, we employ it as our default VQA model.

Next, we evaluate the agreement between Llava and FairFace [150] on different scenarios. Specifically, we run the two models on real and synthetic images generated with Stable Diffusion 1.5, 2, and XL. We measure the agreement between the two as the Kullback–Leibler (KL) Divergence between the probability distributions obtained using the predictions of the respective model. We report the results in Table 21. We can observe that the models are highly aligned, obtaining low KL scores, proving the VQA model's robustness in both generative and real settings.

6.2.3.3 Qualitative Results We show examples of biases discovered by OpenBias on Stable Diffusion XL. We present the results in a context-aware fashion and visualize images generated from the same caption where our pipeline identifies a bias. We organize the results in three sets and present unexplored biases on objects and animals, novel biases associated with persons, and well-known social biases. We highlight biases discovered on objects and animals in Fig. 24. For example, the model tends to generate *"yellow"* trains or *"quarter horses"* even if not specified in the caption. Furthermore, the model generates laptops featuring a distinct *"Apple"* logo, showing a bias toward the brand.

Next, we display novel biases related to persons discovered by OpenBias. For instance, we

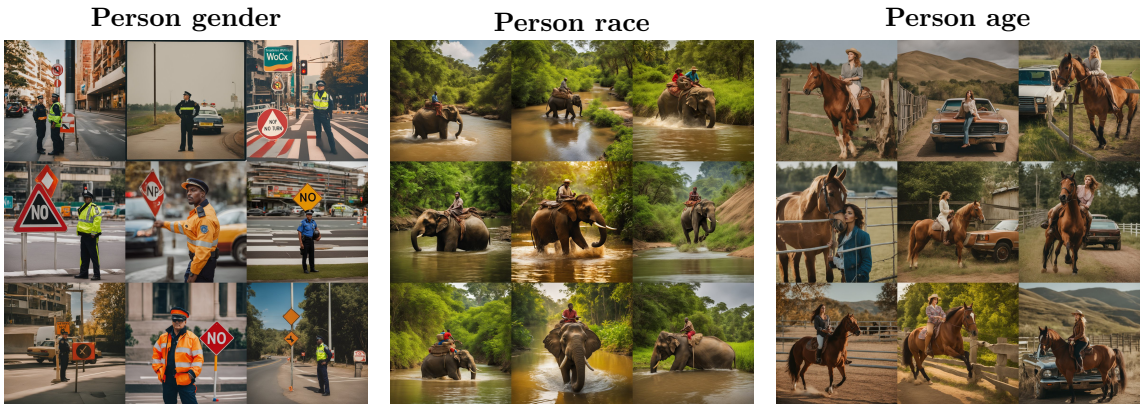


Figure 26. Person-related biases found on Stable Diffusion XL [122] by OpenBias.

unveil unexplored biases such as the *"person attire"*, with the model often generating people in a formal outfit rather than more casual ones. Furthermore, we specifically study *"child gender"* and *"child race"* diverging from the typical examination centered on adults. For example, in Fig. 25 second column, we observe that the generative model links a black child with an economically disadvantaged environment described in the caption as *"a dirt road"*. The association between racial identity and socioeconomic status perpetuates harmful stereotypes and proves the need to consider novel biases within bias mitigation frameworks. Lastly, we show qualitative results on the well-studied and sensitive biases of *"person gender"*, *"race"*, and *"age"*. In the first column of Fig. 26, Stable Diffusion XL exclusively generates *"male"* officers, despite the presence of a gender-neutral job title. Moreover, it explicitly depicts a *"woman"* labeled as *"middle-aged"* when engaged in horseback riding. Finally, we observe a *"race"* bias, with depictions of solely black individuals for *"a man riding an elephant"*. This context-aware approach ensures a thorough comprehension of emerging biases in both novel and socially significant contexts. These results emphasize the necessity for more inclusive open-set bias detection frameworks.

6.2.4 Conclusion

The main contributions of this work are as follows:

- To the best of our knowledge, we are the first to study the problem of open-set bias detection at large scale without relying on a predefined list of biases. Our method discovers novel biases that have never been studied before.
- We propose OpenBias, a modular pipeline, that, given a list of prompts, leverages a Large Language Model to extract a knowledge base of possible biases, and a Vision Question Answer model to recognize and quantify them.
- We test our pipeline on multiple text-to-image generative models: Stable Diffusion XL, 1.5, 2 [121], [122]. We assess our pipeline showing its agreement with closed-set classifier-based methods and with human judgement.

6.2.5 Relevant publications

- M. D'Incà, E. Peruzzo, M. Mancini, D. Xu, V. Goel, X. Xu, Z. Wang, H. Shi, and N. Sebe, OpenBias: Open-set Bias Detection in Generative Models, CVPR 2024 [157]
  Zenodo record: `https://zenodo.org/records/11303876`
- M. D'Incà, C. Tzelepis, Y. Patras, and N. Sebe, Improving Fairness using Vision-Language Driven Image Augmentation, WACV 2024 [146]
  Zenodo record: `https://zenodo.org/records/11303771`

6.2.6 Relevant software/datasets/other outcomes

The Pytorch implementations can be found in:

- `https://github.com/Picsart-AI-Research/OpenBias`
- `https://github.com/Moreno98/Vision-Language-Bias-Control`

6.2.7 Relevance to AI4Media use cases and media industry applications

OpenBias could be relevant to UC1 as it tackles disinformation detection and discovery of new biases. The results of our evaluation would also be relevant to UC4 "AI for Social Sciences and Humanities" as the impact of deploying an AI system "in-the-wild" without concern for fair treatment of subjects, may introduce biases and scenarios in which certain people are treated unfairly or are discriminated against. As a result, this work is also relevant to UC2 "AI for News", as a tool which can help journalists discern authentic content from deepfakes, must also ideally be fair and unbiased, which this work strives to achieve.

7 Organisation of events for Trustworthy AI

In addition to the scientific and technical work of WP4, the following events on different aspects of Trustworthy AI were organised by the consortium:

1. **Theme Development Workshop (TDW) on Trusted AI – The Future of Creating Ethical & Responsible AI Systems**,[6] an online workshop co-organised with the ICT48+3 NoEs on 13 September 2023, which included the following sessions organised by AI4Media partners:

   - **AI explainability for vision tasks**. This session discussed the present and future of AI explainability for visual data classifiers and other vision tasks, how explanations can be presented to the users, and what we can expect to understand from these explanations.
   - **Rigorous vs empirical AI privacy**. This session discussed the relevance of epsilon as a definitive measure of privacy loss in the context of complex algorithms implementing differential privacy and the proliferation of empirical measurements of privacy via attacks.
   - **AI Ethics: from principles to practice. Putting "ethical" and"responsible" AI into action**. The session focused on operationalizing the AI ethical guidelines and principles and reflected on the shifting approach from high-level ethical principles towards legally binding obligations (e.g. in the AI Act) and practical tools (e.g. the Human Rights Impact Assessments
   - **Ethical considerations and new challenges of Generative AI**. This session explored the risks and challenges raised by Generative AI from an interdisciplinary perspective (legal, ethical, societal, technical, and cybersecurity).

   A report summarising the findings of the different sessions was produced.[7]

2. **Assessing and Enhancing Fairness in AI Systems**,[8] a session in the 4th AI Community Workshop 2024 & AIDA Symposium,[9] in Thessaloniki, Greece on 26 June 2024. The session focused on the following themes: (1) interdisciplinary initiatives that seek to make AI fairer, such as initiatives to reduce dataset biases by design, (2) effects of biases in high-impact AI applications (face recognition, recommenders, automatic scoring, media analysis), and (3) representational biases in large multimodal and language models. The session recording is available in the AI4Media YouTube channel.[10]

[6] https://www.vision4ai.eu/tdw-trusted-ai/

[7] https://www.vision4ai.eu/wp-content/uploads/2024/04/Full-Report-on-the-key-findings-from-the-Theme-Development-Workshop-Trusted-AI_-The-Future-of-Creating-Ethical-Responsible-AI-Systems_-1.pdf

[8] https://www.vision4ai.eu/community-workshop-2024/#pw5

[9] https://www.vision4ai.eu/community-workshop-2024/

[10] https://www.youtube.com/watch?v=Of78-Q26P_w&ab_channel=AI4MediaProject

8 Conclusions

This deliverable is the concluding piece of technical work towards Trustworthy AI for the AI4Media European Horizon project. The work herein adds to the already considerable volume and quality of work presented previously as part of Deliverables 4.1 and 4.5. In total, some 50 pieces of work have been presented to this end, with contributions from over ten partners across Europe over the last four years. Covering AI Robustness, Explainability, Privacy and Fairness, the contents of these deliverables will be a strong reference point for future research, development and deployment of trustworthy AI in the media sector for years to come.

Over the course of the project, many novel works have been presented across the four primary Trustworthiness pillars.

- In **AI Robustness**, new defences methods and attacks were developed concerning adversarial robustness, including work on images and videos, LLMs, and federated learning.
- In **AI Explainability**, work was completed on a range of data, including text, image, video and audio, as well as multi-modal AI systems.
- In **AI Privacy**, a host of novel contributions were made, including using differential privacy, unlearning, homomorphic encryption, and more traditional methods like de-identification and k-anonymity.
- Finally, in **AI Fairness**, the fairness of neural networks was of primary concern, including some cross-pollination with the privacy task and the link between privacy, fairness and unlearning.

A number of these highlights were included in the AI4Media Technological Highlights booklet on Trustworthy AI.[11]

In the fast-paced world of AI research and development, nothing ever stands still. In following on from the great volume of work produced as part of Trustworthy AI in AI4Media, a number of key challenges need continued addressing and attention. Particularly relevant to the media industry is that researchers continue to develop tools and algorithms that are easy and user-friendly to consume, particularly for AI non-experts, such as media professionals, whose day-to-day job is to produce media. This ease-of-use should extend to wide applicability across types of models, data and systems architectures, to truly democratise AI and trustworthy AI particularly.

It's particularly important for our non-expert users to offer informed and understandable information about the tradeoffs of applying the various trustworthy AI interventions. For example, in AI Privacy, methods such as differential privacy and k-anonymity result in a drop in accuracy as a tradeoff for the privacy protection. While this tradeoff may be intuitive to researchers involved in the field, correctly communicating that to non-experts is challenging and requires attention going forward.

When developing suites of tools, like the AI4Media partners have done, it's also important to be aware of the law of unintended consequences when combining different tools from trustworthiness arsenal. Off-the-shelf implementation of, e.g., a privacy mitigation, may inadvertently hamper that model's, e.g., fairness, or vice versa. Developing an understanding around combining tools will be important as more of these tools are developed and deployed in the wild. In this deliverable, related work was detailed in Sections 4.3 and 6.1.

Finally, since the start of this AI4Media project in September 2020, a whole new area of AI research has shot into life. The advent of Large Language Models as a daily part of our lives has

[11] https://www.ai4media.eu/wp-content/uploads/2024/08/BookletofTechnologicalHighlights_WP4.pdf

happened almost overnight. While many tools developed for more classical AI models an still be applied to LLMs, an entire new branch of research has appeared in the blink of an eye. Even within AI4Media, many papers have already been presented on LLMs, but it's clear the challenges of these large models are vast and intricate from a trustworthiness perspective, and will serve as a beacon for research in the years and decades to come.

References

[1] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, *Towards the science of security and privacy in machine learning*, 2016. arXiv: 1611.03814 [cs.CR].

[2] X. Wang, J. Li, X. Kuang, Y.-a. Tan, and J. Li, "The security of machine learning in an adversarial setting: A survey," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12–23, 2019, ISSN: 0743-7315. DOI: https://doi.org/10.1016/j.jpdc.2019.03.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731518309183.

[3] S. Szyller and N. Asokan, "Conflicting interactions among protection mechanisms for machine learning models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 15 179–15 187.

[4] J. Thakkar, G. Zizzo, and S. Maffeis, "Elevating defenses: Bridging adversarial training and watermarking for model resilience," *arXiv preprint arXiv:2312.14260*, 2023.

[5] E. Le Merrer, P. Perez, and G. Trédan, "Adversarial frontier stitching for remote neural network watermarking," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9233–9244, 2020.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[7] E. Apostolidis, G. Balaouras, I. Patras, and V. Mezaris, "Explainable video summarization for advancing media content production," in *Encyclopedia of Information Science and Technology, Sixth Edition.* D. Mehdi Khosrow-Pour, Ed., Hershey, PA: IGI Global, 2025, pp. 1–24.

[8] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Video summarization using deep neural networks: A survey," *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1838–1863, 2021. DOI: 10.1109/JPROC.2021.3117472.

[9] T. Souček and J. Lokoč, "Transnet v2: An effective deep network architecture for fast shot transition detection," *arXiv preprint arXiv:2008.04838*, 2020.

[10] K. Apostolidis, E. Apostolidis, and V. Mezaris, "A motion-driven approach for fine-grained temporal segmentation of user-generated videos," in *24th Int. Conf. on MultiMedia Modeling, MMM 2018, Bangkok, Thailand, February 5-7, 2018, Proceedings, Part I 24*, Springer, 2018, pp. 29–41.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[12] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras, "Explaining video summarization based on the focus of attention," in *2022 IEEE International Symposium on Multimedia (ISM)*, 2022, pp. 146–150. DOI: 10.1109/ISM55400.2022.00029.

[13] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras, "Summarizing videos using concentrated attention and considering the uniqueness and diversity of the video frames," in *Proc. of the 2022 Int. Conf. on Multimedia Retrieval*, ser. ICMR '22, Newark, NJ, USA: Association for Computing Machinery, 2022, pp. 407–415, ISBN: 9781450392389. DOI: 10.1145/3512527.3531404. [Online]. Available: https://doi.org/10.1145/3512527.3531404.

[14] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, "Summarizing Videos with Attention," in *Asian Conf. on Computer Vision (ACCV) 2018 Workshops*, G. Carneiro and S. You, Eds., Cham: Springer International Publishing, 2019, pp. 39–54, ISBN: 978-3-030-21074-8.

[15] P. Li, Q. Ye, L. Zhang, L. Yuan, X. Xu, and L. Shao, "Exploring global diverse attention via pairwise temporal relation for video summarization," *Pattern Recognition*, vol. 111, p. 107 677, 2021, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2020.107677. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320320304805.

[16] X. Li, W. Zhang, J. Pang, K. Chen, G. Cheng, Y. Tong, and C. C. Loy, "Video k-net: A simple, strong, and unified baseline for video segmentation," in *CVPR*, 2022.

[17] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating Summaries from User Videos," in *Europ. Conf. on Computer Vision (ECCV) 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 505–520, ISBN: 978-3-319-10584-0. [Online]. Available: https://gyglim.github.io/me/.

[18] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVSum: Summarizing web videos using titles," in *2015 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 5179–5187. DOI: 10.1109/CVPR.2015.7299154. [Online]. Available: https://github.com/yalesong/tvsum.

[19] M. G. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945.

[20] E. Apostolidis, V. Mezaris, and I. Patras, "A study on the use of attention for explaining video summarization," in *Proceedings of the 2nd Workshop on User-Centric Narrative Summarization of Long Videos*, ser. NarSUM '23, Ottawa ON, Canada: Association for Computing Machinery, 2023, pp. 41–49. DOI: 10.1145/3607540.3617138. [Online]. Available: https://doi.org/10.1145/3607540.3617138.

[21] K. Tsigos, E. Apostolidis, and V. Mezaris, *An integrated framework for multi-granular explanation of video summarization*, 2024. arXiv: 2405.10082 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2405.10082.

[22] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23, Curran Associates, Inc., 2010. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2010/file/fe73f687e5bc5280214e0486b273a5f9-Paper.pdf.

[23] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 202, PMLR, 2023, pp. 19 730–19 742. [Online]. Available: https://proceedings.mlr.press/v202/li23q.html.

[24] B. Trabucco, K. Doherty, M. Gurinas, and R. Salakhutdinov, "Effective data augmentation with diffusion models," in *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023. [Online]. Available: https://openreview.net/forum?id=dcCpG0CVMf.

[25] V. Ranjan, U. Sharma, T. Nguyen, and M. Hoai, "Learning to count everything," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3394–3403.

[26] Z. You, K. Yang, W. Luo, X. Lu, L. Cui, and X. Le, "Few-shot object counting with similarity-aware feature enhancement," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2023, pp. 6315–6324.

[27] R. He, S. Sun, X. Yu, C. Xue, W. Zhang, P. Torr, S. Bai, and X. Qi, "Is synthetic data from generative models ready for image recognition?" *arXiv preprint arXiv:2210.07574*, 2022.

[28] L. Chang, Z. Yujie, Z. Andrew, and X. Weidi, "Countr: Transformer-based generalised visual counting," in *British Machine Vision Conference (BMVC)*, 2022.

[29] M. B. Sariyildiz, K. Alahari, D. Larlus, and Y. Kalantidis, "Fake it till you make it: Learningtocount transferable representations from synthetic imagenet clones," in *CVPR 2023–IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[30] C. Meng, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Image synthesis and editing with stochastic differential equations," *arXiv preprint*, 2021.

[31] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4145–4153.

[32] M. Shi, H. Lu, C. Feng, C. Liu, and Z. Cao, "Represent, compare, and learn: A similarity-aware framework for class-agnostic counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9529–9538.

[33] P. Doubinsky, N. Audebert, M. Crucianu, and H. Le Borgne, "Semantic generative augmentations for few-shot counting," in *Winter Conference on Applications of Computer Vision (WACV)*, 2024.

[34] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[35] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information fusion*, vol. 58, pp. 82–115, 2020.

[36] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*, IEEE, 2017, pp. 3–18.

[37] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[38] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction {apis}," in *25th USENIX security symposium (USENIX Security 16)*, 2016, pp. 601–618.

[39] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[40] V. Chandrasekaran, K. Chaudhuri, I. Giacomelli, S. Jha, and S. Yan, "Exploring connections between active learning and model extraction," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 1309–1326.

[41] A. Beygelzimer, D. J. Hsu, J. Langford, and T. Zhang, "Agnostic active learning without constraints," *Advances in neural information processing systems*, vol. 23, 2010.

[42] A. C. Oksuz, A. Halimi, and E. Ayday, "Autolycus: Exploiting explainable artificial intelligence (xai) for model extraction attacks against interpretable models," *Proceedings on Privacy Enhancing Technologies*, 2024.

[43] M. Graziani, A.-P. Nguyen, L. O'Mahony, H. Müller, and V. Andrearczyk, "Concept discovery and dataset exploration with singular value decomposition," p. 12, Mar. 2023, Presented at The Eleventh International Conference on Learning Representations (ICLR 2023).

[44] N. Mylonas, I. Mollas, and G. Tsoumakas, "An attention matrix for every decision: Faithfulness-based arbitration among multiple attention-based interpretations of transformers in text classification," *Data Mining and Knowledge Discovery*, pp. 1–26, 2023.

[45] D. Mardaoui and D. Garreau, "An analysis of LIME for text data," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR, 2021, pp. 3493–3501.

[46] G. Lopardo, F. Precioso, and D. Garreau, "Attention meets post-hoc interpretability: A mathematical perspective," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=wnkC5T11Z9.

[47] J. Bastings and K. Filippova, "The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?" In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, A. Alishahi, Y. Belinkov, G. Chrupała, D. Hupkes, Y. Pinter, and H. Sajjad, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 149–155. DOI: 10.18653/v1/2020.blackboxnlp-1.14. [Online]. Available: https://aclanthology.org/2020.blackboxnlp-1.14.

[48] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[49] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[50] 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.

[51] R. Wightman, *Pytorch image models*, https://github.com/huggingface/pytorch-image-models, 2019. DOI: 10.5281/zenodo.4414861.

[52] 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*, Ieee, 2009, pp. 248–255.

[53] 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.

[54] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," *arXiv preprint arXiv:1807.02011*, 2018.

[55] L. Cultrera, L. Seidenari, and A. Del Bimbo. "Wildcapture." (2024), [Online]. Available: https://www.ai4europe.eu/research/ai-catalog/wildcapture (visited on 09/06/2024).

[56] S. Beery, G. Van Horn, and P. Perona, "Recognition in terra incognita," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018.

[57] J. Van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, "Uncertainty estimation using a single deep deterministic neural network," in *International conference on machine learning*, PMLR, 2020, pp. 9690–9700.

[58] J. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax Weiss, and B. Lakshminarayanan, "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7498–7512, 2020.

[59] S. Fort, J. Ren, and B. Lakshminarayanan, "Exploring the limits of out-of-distribution detection," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7068–7081, 2021.

[60] S. Cao and Z. Zhang, "Deep hybrid models for out-of-distribution detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4733–4743.

[61] L. Cultrera, L. Seidenari, and A. Del Bimbo, "Leveraging visual attention for out-of-distribution detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4447–4456.

[62] P. De Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[63] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proc. of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9329–9338.

[64] A. Greco, L. Rundo, A. Saggese, M. Vento, and A. Vicinanza, "Imitation learning for autonomous vehicle driving: How does the representation matter?" In *International Conference on Image Analysis and Processing*, Springer, 2022, pp. 15–26.

[65] M. R. Samsami, M. Bahari, S. Salehkaleybar, and A. Alahi, "Causal imitative model for autonomous driving," *arXiv preprint arXiv:2112.03908*, 2021.

[66] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[67] F. Codevilla, A. M. Lopez, V. Koltun, and A. Dosovitskiy, "On offline evaluation of vision-based driving models," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 236–251.

[68] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 661–668.

[69] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 627–635.

[70] Y. Schroecker and C. L. Isbell, "State aware imitation learning," *Advances in Neural Information Processing Systems*, 2017.

[71] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[72] D. Omeiza, H. Webb, M. Jirotka, and L. Kunze, "Explanations in autonomous driving: A survey," *IEEE Trans. on Intelligent Transportation Systems*, 2021.

[73] L. Cultrera, L. Seidenari, F. Becattini, P. Pala, and A. Del Bimbo, "Explaining autonomous driving by learning end-to-end visual attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 340–341.

[74] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of vision-based autonomous driving systems: Review and challenges," *arXiv preprint arXiv:2101.05307*, 2021.

[75] A. Attia and S.Dayan, "Global overview of imitation learning," *arXiv 1801.06503v1*, 2018.

[76] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[77] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," *In 2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–9, 2018.

[78] A. Sauer, N. Savi-nov, and A. Geiger, "Conditional affordance learning for driving in urban environments," *in Conference on Robot Learning (CoRL)*, 2018.

[79] L. Cultrera, F. Becattini, L. Seidenari, P. Pala, and A. Del Bimbo, "Addressing limitations of state-aware imitation learning for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, 2023.

[80] J. Zhu and M. Blaschko, "R-gap: Recursive gradient attack on privacy," *Proceedings ICLR 2021*, 2021.

[81] J. Deng, Y. Wang, J. Li, C. Wang, C. Shang, H. Liu, S. Rajasekaran, and C. Ding, "Tag: Gradient attack on transformer-based language models," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 3600–3610.

[82] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*, IEEE, 2019, pp. 691–706.

[83] H. Ren, J. Deng, and X. Xie, "Grnn: Generative regression neural network—a data leakage attack for federated learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 4, May 2022, ISSN: 2157-6904. DOI: 10.1145/3510032. [Online]. Available: https://doi.org/10.1145/3510032.

[84] H.-M. Chu, J. Geiping, L. H. Fowl, M. Goldblum, and T. Goldstein, "Panning for gold in federated learning: Targeted text extraction under arbitrarily large-scale aggregation," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=A9WQaxYsfx.

[85] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/60a6c4002cc7b29142def8871531281a-Paper.pdf.

[86] X. Jin, P.-Y. Chen, C.-Y. Hsu, C.-M. Yu, and T. Chen, "Cafe: Catastrophic data leakage in vertical federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 994–1006, 2021.

[87] Y. Wang, J. Deng, D. Guo, C. Wang, X. Meng, H. Liu, C. Shang, B. Wang, Q. Cao, C. Ding, and S. Rajasekaran, "Variance of the gradient also matters: Privacy leakage from gradients," in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9892665.

[88] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2021, pp. 16 332–16 341. DOI: 10.1109/CVPR46437.2021.01607. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01607.

[89] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients - how easy is it to break privacy in federated learning?" In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 16 937–16 947. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/c4ede56bbd98819ae6112b20ac6bf145-Paper.pdf.

[90] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot, *Reconstructing individual data points in federated learning hardened with differential privacy and secure aggregation*, 2023. arXiv: 2301.04017 [cs.CR].

[91] J. C. Zhao, A. R. Elkordy, A. Sharma, Y. H. Ezzeldin, S. Avestimehr, and S. Bagchi, "The resource problem of using linear layer leakage attack in federated learning," *arXiv preprint arXiv:2303.14868*, 2023.

[92] J. C. Zhao, A. Sharma, A. R. Elkordy, Y. H. Ezzeldin, S. Avestimehr, and S. Bagchi, *Secure aggregation in federated learning is not private: Leaking user data at large scale through model modification*, 2023. arXiv: 2303.12233 [cs.LG].

[93] L. H. Fowl, J. Geiping, W. Czaja, M. Goldblum, and T. Goldstein, "Robbing the fed: Directly obtaining private data in federated learning with modified models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=fwzUgo0FM9v.

[94] F. Boenisch, A. Dziedzic, R. Schuster, A. S. Shamsabadi, I. Shumailov, and N. Papernot, *When the curious abandon honesty: Federated learning is not private*, 2021. arXiv: 2112.02918 [cs.LG].

[95] B. Zhao, K. R. Mopuri, and H. Bilen, *Idlg: Improved deep leakage from gradients*, 2020. arXiv: 2001.02610 [cs.LG].

[96] S. Gupta, Y. Huang, Z. Zhong, T. Gao, K. Li, and D. Chen, "Recovering private text in federated learning of language models," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=dqgzfhHd2-.

[97] M. Suliman and D. Leith, "Two models are better than one: Federated learning is not private for google gboard next word prediction," in *European Symposium on Research in Computer Security*, Springer, 2023, pp. 105–122.

[98] S. Exchange, *Stack exchange data dump*, 2023. [Online]. Available: https://archive.org/details/stackexchange.

[99] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[100] A. Cortesi, M. Hils, T. Kriechbaumer, and contributors, *mitmproxy: A free and open source interactive HTTPS proxy (v5.01)*, 2020. [Online]. Available: https://mitmproxy.org/.

[101] N. Kourtellis, K. Katevas, and D. Perino, "Flaas: Federated learning as a service," in *Proceedings of the 1st Workshop on Distributed Machine Learning*, ser. DistributedML'20, Barcelona, Spain: Association for Computing Machinery, 2020, pp. 7–13, ISBN: 9781450381826. DOI: 10.1145/3426745.3431337. [Online]. Available: https://doi.org/10.1145/3426745.3431337.

[102] M. Suliman, D. Leith, and A. Halimi, "Re-evaluating the privacy benefit of federated learning," in *1st Workshop on Advancments in Federated Learning, ECML-PKDD 2023.*, 2023.

[103] D. J. Beutel *et al.*, *Flower: A friendly federated learning research framework*, 2022. arXiv: 2007.14390 [cs.LG].

[104] O. Yadan, *Hydra - a framework for elegantly configuring complex applications*, Github, 2019. [Online]. Available: https://github.com/facebookresearch/hydra.

[105] S. Grollmisch, J. Abeßer, J. Liebetrau, and H. Lukashevich, "Sounding industry: Challenges and datasets for industrial sound analysis," in *2019 27th European Signal Processing Conference (EUSIPCO)*, IEEE, 2019, pp. 1–5.

[106] F. Wibawa *et al.*, "Homomorphic encryption and federated learning based privacy-preserving cnn training: Covid-19 detection use-case," in *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference*, 2022, pp. 85–90.

[107] S. P. Sanon *et al.*, "Secure federated learning: An evaluation of homomorphic encrypted network traffic prediction," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, IEEE, 2023, pp. 1–6.

[108] N. M. Hijazi *et al.*, "Secure federated learning with fully homomorphic encryption for iot communications," *IEEE Internet of Things Journal*, 2023.

[109] J. Park and H. Lim, "Privacy-preserving federated learning using homomorphic encryption," *Applied Sciences*, vol. 12, no. 2, 2022, ISSN: 2076-3417. DOI: 10.3390/app12020734. [Online]. Available: https://www.mdpi.com/2076-3417/12/2/734.

[110] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *2021 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2021, pp. 141–159.

[111] A. Chouldechova and A. Roth, "The frontiers of fairness in machine learning," *arXiv preprint arXiv:1810.08810*, 2018.

[112] I. B. Soares, D. Wei, K. N. Ramamurthy, M. Singh, and M. Yurochkin, "Your fairness may vary: Pretrained language model fairness in toxic text classification," in *Annual Meeting of the Association for Computational Linguistics*, 2022.

[113] S. Verma and J. Rubin, "Fairness definitions explained," in *Proceedings of the international workshop on software fairness*, 2018.

[114] P. Czarnowska, Y. Vyas, and K. Shah, "Quantifying social biases in nlp: A generalization and empirical comparison of extrinsic fairness metrics," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 1249–1267, 2021.

[115] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[116] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, and A. Mukherjee, "Hatexplain: A benchmark dataset for explainable hate speech detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, 2021, pp. 14 867–14 875.

[117] S. R. Kadhe, A. Halimi, A. Rawat, and N. Baracaldo, "Fairsisa: Ensemble post-processing to improve fairness of unlearning in llms," *arXiv preprint arXiv:2312.07420*, 2023.

[118] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *NeurIPS*, 2022.

[119] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," in *International Conference on Machine Learning*, 2022.

[120] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint*, 2022.

[121] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

[122] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "SDXL: Improving latent diffusion models for high-resolution image synthesis," in *ICLR*, 2024.

[123] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *CVPR*, 2023.

[124] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An image is worth one word: Personalizing text-to-image generation using textual inversion," *arXiv preprint*, 2022.

[125] D. Epstein, A. Jabri, B. Poole, A. A. Efros, and A. Holynski, "Diffusion self-guidance for controllable image generation," in *NeurIPS*, 2023.

[126] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *CVPR*, 2023.

[127] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, "Prompt-to-prompt image editing with cross attention control," in *ICLR*, 2022.

[128] V. Goel, E. Peruzzo, Y. Jiang, D. Xu, X. Xu, N. Sebe, T. Darrell, Z. Wang, and H. Shi, "Pair-diffusion: A comprehensive multimodal object-level image editor," *arXiv preprint*, 2023.

[129] O. Avrahami, T. Hayes, O. Gafni, S. Gupta, Y. Taigman, D. Parikh, D. Lischinski, O. Fried, and X. Yin, "Spatext: Spatio-textual representation for controllable image generation," in *CVPR*, 2023.

[130] L. Zhang, A. Rao, and M. Agrawala, "Adding conditional control to text-to-image diffusion models," in *ICCV*, 2023.

[131] L. Huang, D. Chen, Y. Liu, Y. Shen, D. Zhao, and J. Zhou, "Composer: Creative and controllable image synthesis with composable conditions," in *ICML*, 2023.

[132] C. Zhang, X. Chen, S. Chai, C. H. Wu, D. Lagun, T. Beeler, and F. De la Torre, "Iti-gen: Inclusive text-to-image generation," in *ICCV*, 2023.

[133] F. Friedrich, M. Brack, L. Struppek, D. Hintersdorf, P. Schramowski, S. Luccioni, and K. Kersting, "Fair diffusion: Instructing text-to-image generation models on fairness," *arXiv preprint*, 2023.

[134] J. Cho, A. Zala, and M. Bansal, "Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models," in *ICCV*, 2023.

[135] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *NeurIPS*, 2016.

[136] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach, "Women also snowboard: Overcoming bias in captioning models," in *ECCV*, 2018.

[137] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang, "Men also like shopping: Reducing gender bias amplification using corpus-level constraints," in *EMNLP*, 2017.

[138] R. Naik and B. Nushi, "Social biases through the text-to-image generation lens," in *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, 2023.

[139] P. J. Kenfack, K. Sabbagh, A. R. Rivera, and A. Khan, "Repfair-gan: Mitigating representation bias in gans using gradient clipping," *arXiv preprint*, 2022.

[140] D. Xu, S. Yuan, L. Zhang, and X. Wu, "Fairgan: Fairness-aware generative adversarial networks," in *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, 2018.

[141] J. Nam, H. Cha, S. Ahn, J. Lee, and J. Shin, "Learning from failure: De-biasing classifier from biased classifier," *NeurIPS*, 2020.

[142] Y. Savani, C. White, and N. S. Govindarajulu, "Intra-processing methods for debiasing neural networks," in *NeurIPS*, 2020.

[143] Z. Wang, K. Qinami, I. C. Karakozis, K. Genova, P. Nair, K. Hata, and O. Russakovsky, "Towards fairness in visual recognition: Effective strategies for bias mitigation," in *CVPR*, 2020.

[144] S. Jung, S. Chun, and T. Moon, "Learning fair classifiers with partially annotated group labels," in *CVPR*, 2022.

[145] S. Agarwal, S. Muku, S. Anand, and C. Arora, "Does data repair lead to fair models? curating contextually fair data to reduce model bias," in *WACV*, 2022.

[146] M. D'Inca, C. Tzelepis, Y. Patras, and N. Sebe, "Improving fairness using vision-language driven image augmentation," in *WACV*, 2024.

[147] X. Su, Y. Ren, W. Qiang, Z. Song, H. Gao, F. Wu, and C. Zheng, "Unbiased image synthesis via manifold-driven sampling in diffusion models," *arXiv preprint*, 2023.

[148] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, 2014.

[149] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *ECCV*, 2014.

[150] K. Karkkainen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation," in *WACV*, 2021.

[151] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[152] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," in *ICML*, 2022.

[153] C. Li, H. Xu, J. Tian, W. Wang, M. Yan, B. Bi, J. Ye, H. Chen, G. Xu, Z. Cao, *et al.*, "MPLUG: Effective and efficient vision-language learning by cross-modal skip-connections," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[154] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *ICML*, 2022.

[155] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.

[156] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *NeurIPS*, 2023.

[157] M. D'Inca, E. Peruzzo, M. Mancini, D. Xu, V. Goel, X. Xu, Z. Wang, H. Shi, and N. Sebe, "Openbias: Open-set bias detection in generative models," in *CVPR*, 2024.