Karlsruher Institut für Technologie

Explainable Artificial Intelligence in Adversarial Environments

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN

von der KIT-Fakultät für Informatik des
Karlsruher Instituts für Technologie (KIT)
genehmigte

DISSERTATION

von

Maximilian Konstantin Noppel

1. Referent:
Prof. Dr. Christian Wressnegger

2. Referent:
Prof. Dr. Francesco Leofante

Tag der mündlichen Prüfung:
20. April 2026

To Annika.

Abstract

Modern AI models, such as deep neural networks, perform remarkably well across increasingly complex and highly non-linear tasks. Their performance is enabled by an enormous number of learned parameters, which, in turn, render modern neural networks incomprehensible, even to their creators. This incomprehensibility is particularly problematic in high-stakes domains where individuals are seriously affected by a model's predictions.

To address this problem, researchers have developed explanation algorithms. These eXplainable Artificial Intelligence (XAI) solutions shed light on a model's decision-making processes. Among these, feature attribution methods assign importance scores to individual input features, which can be visualized as overlays on inputs, particularly for images. This thesis focuses on this category of explanation methods.

Unfortunately, feature attribution methods must approximate the highly non-linear behavior of models and can therefore be forged. Adversaries can employ imperceptible input perturbations to arbitrarily manipulate both explanations and predictions, or influence model parameters to achieve similar goals.

In this thesis, we systematize such *explanation-aware attacks* along the dimensions of diverse adversarial capabilities, constraints, and goals. We present a hierarchy of robustness notions to understand these attacks better and describe the attack surface of XAI-systems.

We introduce *explanation-aware backdooring attacks*, where adversaries manipulate models to exhibit benign explanations and correct predictions under normal conditions, while triggering adversary-chosen predictions and explanations in the presence of specific patterns in the inputs. Artifacts in the explanation that would otherwise originate from the ongoing attack against the prediction can therefore be disguised by our backdoors.

Furthermore, we present *composite attacks* that decouple the attacks against the prediction and the explanation by combining data-poisoning attacks to forge the prediction and sample-specific input perturbations to forge the explanation.

Additionally, we propose *explanation-aware adversarial training* as a defense against explanation-aware adversarial examples, extending existing adversarial-training techniques and benchmarking them against prior work on explanation stabilization.

In summary, this thesis argues that the reliability of explanations is essential for the broader adoption of AI systems in high-stakes domains, in particular in adversarial environments. This thesis contributes to this goal by systematizing explanation-aware attacks, developing attacks and defenses, and by demonstrating how these insights can strengthen the robustness and trustworthiness of eXplainable Artificial Intelligence (XAI) in the presence of adversarial influence.

Zusammenfassung

Moderne Systeme basierend auf Künstlicher Intelligenz (KI), insbesondere künstliche Neuronale Netze, zeigen eine erstaunliche Performanz über zunehmend komplexe und nicht-lineare Aufgaben hinweg. Ihre Performanz wird vor allem durch eine enorme Anzahl an gelernten Parametern ermöglicht, was wiederum dazu führt, dass diese Systeme selbst von ihren Entwicklern nicht nachvollzogen werden können. Das ist besonders problematisch, sollten diese Modelle in Bereichen eingesetzt werden, in denen Personen oder Institutionen substantiell von den Vorhersagen der Modelle betroffen sein können.

Um dieses Problem zu adressieren, haben Forschende Erklärungsalgorithmen entwickelt. Diese Lösungen, die unter dem englischen Begriff "eXplainable Artificial Intelligence (XAI)" zusammengefasst werden, legen die Entscheidungsprozesse der Modelle offen. Insbesondere die Untergruppe der sogenannten "Feature Attribution"-Methoden ist für diese Arbeit relevant. Diese Methoden weisen jeder Teilinformation der Eingabe einen Wichtigkeitswert zu. Diese Wichtigkeitswerte können dann beispielsweise als Heatmap über der Eingabe visualisiert werden um wichtige Bereiche graphisch hervorhebt. Besonders für die Klassifizierung von Bildern ist das eine übliche Vorgehensweise.

Leider können diese Erklärungsmethoden das hochgradig nicht-lineare Verhalten von modernen Modellen nicht exakt abbilden und sind daher angreifbar. Angreifer können leicht veränderte Eingaben konstruieren, die die Vorhersagen und Erklärungen des Modells manipulieren. Alternativ können auch die Modelle selbst verändert werden, um Vorhersagen und Erklärungen zu beeinflussen.

In dieser Arbeit werden Angriffe, welche Erklärungen berücksichtigen, anhand verschiedener Kategorien systematisiert. Es wird eine Hierarchie aus Robustheitsbegriffen vorgestellt und es werden verschiedene Angriffsvektoren erläutert.

Im Verlauf dieser Arbeit werden zwei konkrete Angriffe vorgestellt und evaluiert:

Im ersten Angriff verändert der Angreifer das Modell so, dass es normale Erklärungen und korrekte Vorhersagen erzeugt, wenn es mit gutartigen Eingaben verwendet wird. Wird jedoch ein kleines spezifisches Muster in die Eingabe eingebracht, erzeugt das Modell Erklärungen und Vorhersagen wie vom Angreifer gewünscht. Der Angreifer kann damit unter anderem verhindern, dass Artefakte in den Erklärungen auftauchen, die ansonsten auf den Angriff hindeuten könnten.

Der zweite Angriff entkoppelt die Manipulation der Vorhersage vom Angriff auf die Erklärung. Während die Vorhersage vor dem Lernen des Modells durch eine Vorverunreinigung der Trainingsdaten beeinflusst wird, erfolgt der Angriff auf die Erklärungen erst, wenn das Modell schließlich verwendet wird. Durch geschickte Manipulation der Eingabe kann dann eine geeignete Erklärung erzwungen werden, die zu der jeweiligen Eingabe passt. Die Vorhersagen werden wie zuvor über ein spezifisches Muster manipuliert.

Zusätzlich wird eine Verteidigung gegen die Manipulation von Erklärungen zur Verwendungszeit des Modells vorgestellt. Dazu wird das etablierte Verteidigungsprinzip

"Adversarial Training" für Angriffe auf Erklärungen erweitert und mit Ansätzen zur Stabilisierung von Erklärungen verglichen.

Zusammenfassend macht diese Arbeit deutlich, dass die Verlässlichkeit von Erklärungen essentiell ist, um eine breitere Verwendung von KI-Systemen in Anwendungsfeldern mit potenziell substantiellen Auswirkungen zu ermöglichen. Insbesondere trifft das zu, wenn die KI-Systeme in böswilligen Kontexten agieren. Die vorliegende Arbeit trägt zur Verlässlichkeit von Erklärungen bei, indem sie Angriffe auf Erklärungen systematisiert, Angriffe und Verteidigungen entwickelt und demonstriert, wie die Robustheit von XAI in böswilligen Umgebungen verbessert werden kann.

Acknowledgement

During my time as a doctoral researcher, I received support from many wonderful people.

First and foremost, I would like to thank my supervisor and mentor, Prof. Dr. Christian Wressnegger. Your support has been unwavering, from countless hours of discussion to late-night efforts to bring me on track. I am grateful for your guidance that has shaped both my work and my thinking. Even when we disagreed, we appreciated our different perspectives, ultimately strengthened the outcome. Thank you for your support.

Further, I like to thank you, Prof. Dr. Francesco Leofante, for reviewing my work.

Over the years, our research team has steadily grown. What started with Qi and me later included many wonderful colleagues: Yilin, Achyut, Gustavo, Niki, Daniel, Alessandro, Nicolai, Anna-Lena, Lena, Shayari, Zhiyang, and most recently Björn. Thank you all for the occasionally shared lunches, retreats, and Christmas parties filled with dishes from around the world. Similarly, thank you for the countless small acts of support that made my PhD journey manageable and enjoyable beyond the research itself.

I thank all my co-authors: Lukas, David, Achyut, Luan, Karl, Christiane, Thorsten, Niclas, Stefan, Thomas, and Konrad. This journey would have been impossible without you.

Similarily, I thank Philipp, who organized the Graduate School. Your open ear and your perspective from your different field were invaluable and refreshing. Additional thanks go out to Dan, our textician. You introduced to me the beauty of the English language.

Beyond academia, I have found great support in my underwater-rugby team. Many of you have become close friends, and I am truly grateful for that. The training sessions, competitions, and shared experiences provided balance and energy. Special thanks also go out to the sailors among you for our adventures on the world's oceans.

Furthermore, thank you, Lukas and Silas, for our adventures on the bike and beyond. Thank you for our conversations, whether lighthearted or deep and philosophical. You made even complex topics a pleasure to discuss. Lukas, special thanks for the coffee breaks.

Thank you, Nils, for countless experiences and adventures. Your perspective always introduced aspects I would have otherwise overlooked. Thank you for the drive you bring to so many projects. You'll always have a place to stay with us.

Thank you, mom and dad. You taught me to trust in the good in people and in myself. Your love and support have been my constant foundation throughout this journey.

Finally, I want to thank Annika. You have had my back through all the highs and lows. You had to bear the brunt of my occasional frustration and disappointment. Still, you have been patiently supporting me through long periods of intense work. I know this journey has demanded a lot from you, and I am deeply grateful for your strength and understanding. Thank you for everything.

Our greatest adventure, though, still lies ahead, and I could not be happier to face it together with you.

Funding

During the creation of the work presented in this thesis, the author and his co-authors received funding and support from various institutions, which we acknowledge below.

Funding. We gratefully acknowledge funding from the German Federal Ministry of Education and Research (BMBF) under the project DataChainSec (FKZ 16KIS1700) and by the Helmholtz Association (HGF) within topic "46.23 Engineering Secure Systems".

Computational Resources. Some of the computational work presented in this thesis was performed on the HoreKa supercomputer, funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the Federal Ministry of Education and Research. Furthermore, we acknowledge support by the state of Baden-Württemberg through bwHPC.

Use of Generative AI. As suggested by the Executive Committee of the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) in their statement from September 2023[1], we point out where generative AI tools have been used in this work.

Generative AI tools, such as GitHub Copilot and JetBrains AI, were partially used to assist in writing code for some of our papers. In particular, they were used to autocomplete lines, to write unit tests, and to document the code. All AI-generated completions were carefully reviewed by the authors.

Furthermore, we used Claude Opus 4.5 and 4.6 as well as Claude Sonnet 4.5 for merging and splitting sentences, checking for typos and grammatical issues, proposing stronger verbs, and finding shorter formulations. Overall, we used suggestions from generative AI tools to improve the readability of our work in some instances. No part of this dissertation was generated entirely by AI. All content, including ideas, analyses, and conclusions, is the original work of the author and his co-authors. These tools were employed merely to enhance productivity and improve the clarity of the writing.

No image in this thesis was generated by general-purpose generative AI tools.

The author takes full responsibility for the content presented in this thesis.

[1] https://www.dfg.de/resource/blob/289676/230921-statement-executive-committee-ki-ai.pdf

Contents

List of Figures

List of Tables

1 Introduction

Artificial Intelligence (AI) technology has become an integral part of modern society, with AI systems exceeding human performance in domains as diverse as medical diagnosis [e.g., Car+15; Est+17; Tak+25], autonomous driving [e.g., Gri+20; Zha+25; Ata+24], and cybersecurity [e.g., Arp+14; Yam+13; Arp+22; Pen+19]. AI systems operate faster and are generally more cost-effective than human experts, e.g., medical personnel, drivers, and security specialists. This rise of AI extends to high-stakes automation tasks where errors carry significant consequences for individuals and organizations [e.g., Rud19; SC23]. As AI technology integrates into every corner of modern society, the reliable functioning of these systems becomes increasingly critical. This criticality applies not only to AI agents on the basis of large language models (LLMs) but also to highly specialized AI systems operating in the shadows of their general-purpose counterparts.

The reliability of AI systems has been measured primarily through predictive performance metrics, such as the accuracy, the F1-score, or, more recently, complex benchmarks. In other words, these systems have been evaluated by asking *what* questions: what is it? A tumor or healthy tissue, malware or benign software, a weapon or a harmless object? Unfortunately, contemporary machine learning models, particularly deep neural networks, are widely regarded as black boxes [e.g., Lip18; Mur+19]. Their vast amount of learned parameters renders their internal decision-making processes opaque, even to their creators. Evaluating only *what* a model predicts is therefore insufficient to investigate a system's reliability in high-stakes applications, where transparency is essential.

The importance of transparency is further reinforced by legal regulations. For instance, the European General Data Protection Regulation (GDPR) and the Artificial Intelligence Act (AI Act) both ask for explanations for automated decisions [GF17; WMF17; EV17]. Consequently, researchers have developed a plethora of *explanation methods* that can shed light on a model's decision-making process, answering the *why* question [AB18; Gui+19; Zha+21; Mil19]. For instance: why is this data classified as a tumor, or why is this executable classified as malicious?

Of particular relevance to this thesis are *feature-attribution methods*, a category of explanation methods that assigns importance scores to individual input features, e.g., to the pixels of an image or the bytes of an executable. These scores highlight the regions of the input that influence the model's output the most, and are typically visualized as colored overlays, or *heatmaps*, indicating "where the model looks." Feature-attribution methods range from gradient-based approaches [e.g., SVZ14; Smi+17] to activation-based methods [e.g., Zho+16; Sel+20] and propagation-based techniques [e.g., Bac+15; Mon+17; Lee+21]. These methods can be either model-agnostic [e.g., RSG16; LL17] or tailored to specific model architectures [e.g., Sel+20; Yin+19; Sch+22], also outside of the image domain [Yin+19]. The spectrum of proposed approaches is extremely large and includes numerous distinct explanation methods for different applications, end-user requirements, and with different properties [e.g., BH21; Gui+19; VL21; LPK21].

Developers leverage explanation methods to debug a model's reasoning [KM08; Kul+15] and to identify flaws in the training data *before* problems occur in deployment [And24; Lap+19; And+22]. For instance, Lapuschkin et al. [Lap+19] have demonstrated that a Fisher vector classifier trained on images of the Pascal VOC 2012 dataset [Eve+10] detects horses based on watermarks of a horse photographer rather than the animals themselves—a paradigmatic example of what the literature terms "Clever Hans" behavior[1], spurious correlation, or shortcut learning [Gei+20; Her+23; And+22; Ben+23; Lap+19]. Such artifacts in the training data can lead to performance degradations when the model encounters real-world data, i.e., images of horses without watermarks. Similarly, in an unverified anecdote, a model has been trained to recognize tanks, yet it has accidentally identified them based on weather conditions due to a bias in the training data [DD92]. The term "spurious correlation" itself, however, dates back even further [Sim54], but has recently gained new relevance since the utilized correlations have become highly interactive and increasingly complex.

Beyond detecting unintentional artifacts, explanation methods can also reveal intentional manipulations. For example, attackers might inject specific artifacts, so-called *triggers*, into the training data and associate them with a target class. Models trained on such poisoned data then predict the target class whenever the trigger appears—an attack known as a *neural backdoor* [e.g., Gu+19; Che+17; Liu+18]. The effect mirrors that of shortcut learning: the model bases its predictions on features that lack a causal relation to the task. The critical difference, however, lies in the intention and manner in which triggers appear. While spurious correlations arise from distributional properties of the data, triggers appear based on the adversary's intention. Hence, they may appear only once, or even never, leaving the neural backdoor dormant. They do not obey any probabilistic regularities. Nevertheless, explanations highlight triggers as the influential regions of the input [Yan+24; BS21; NPW23; Sub+22; LLC21; HAS19]. A fact that has been exploited by explanation-based defenses [CTP20; DAR20; HAS19] to detect backdoor triggers in deployment.

Overall, explainability is a core pillar of trustworthy and reliable AI [Nat23; EU18]. Its intended application extends from the development stage, where developers first identify a model's biases and limitations, then understand and fix them, and ultimately verify that the model behaves correctly [Arp+22; GNB23], to the deployment stage, where end users can interact with the system more efficiently [e.g., Kul+13; Kul+12; Kul+10; Liu+17; SF18; Wan+19b; SF20]. On the basis of explanations, the model's behavior should be continuously monitored in order to react to attacks and shifts in the distribution of inputs.

In both stages, distilling the highly non-linear behavior of a deep neural network into a relatively small set of importance scores requires the application of heuristics. If an attacker knows the applied heuristic, its application becomes a lever for manipulation, i.e., the attacker can forge the explanation, detaching it from the true decision-making process of the model. This forgeability of explanation methods is precisely the focus of this thesis: explainable artificial intelligence in adversarial environments.

[1] The horse "Clever Hans" lived around 1900 and was presented as being capable of performing basic arithmetic. It was later discovered that the horse simply reacted to subtle cues in its handler's body language when the correct answer was reached. This story has become an established term for AI models that perform well but operate on the basis of wrong reasons [e.g., And+22; Ben+23; Lap+19].

This thesis systematically investigates the adversarial landscape of *explanation-aware attacks*, where adversaries target not only predictions but also their accompanying explanations. We provide the first comprehensive systematization across diverse adversarial capabilities, including manipulations of inputs, datasets, models, and entire explainable systems. Thereby we go beyond early preprints that focus solely on input manipulations [Jyo+22; VSL21]. Our systematization categorizes adversarial capabilities and goals, proposes a hierarchy of robustness notions, and taxonomizes defenses against explanation-aware attacks. Building on this foundation, we demonstrate that commonly used feature-attribution methods [e.g., Sel+20; SVZ14; Lee+21; Smi+17] lack robustness and can be manipulated.

We present the first extensive evaluation of *explanation-aware backdoors*, where attackers manipulate AI models such that their explanations disguise the fact that the model secretly relies on hidden trigger patterns. While related work [FC22; Vie+19] has investigated similar backdoors for a single attack scenario, we provide a comprehensive study spanning multiple attack scenarios and explanation methods.

Furthermore, we introduce a threat model that allows us to decouple the attack on predictions from the attack on explanations. Therefore, our adversary injects manipulated samples into the training data to control predictions, then forges explanations at inference time through input perturbations. This decoupling grants maximum flexibility: the adversary can tailor explanations to each sample while simultaneously disguising trigger patterns.

Finally, we propose *explanation-aware adversarial-training* as a defense against explanation-aware input manipulations and evaluate our approaches, their traditional counterparts, and additional defenses along the attack dimensions identified in our systematization.

In summary, this thesis addresses the following research questions:

RQ1. What is the attack surface for manipulating explanations?

RQ2. To what extent can models be manipulated so that they behave inconspicuous, but show adversary-chosen predictions and explanations in the presence of triggers?

RQ3. To what extent can we decouple the attack against the predictions and the attack against the explanations into training-time and inference-time attacks?

RQ4. To what extent does explanation-aware adversarial training improve a model's robustness against explanation-aware input manipulations?

The forgeability of explanation methods, as discussed in this thesis, fundamentally challenges the vision of trustworthy AI built upon explainability. Explainable systems may provide a false sense of security in adversarial environments. Paradoxically, forgeable explanation methods may render systems *more* vulnerable than if they were absent altogether. Plausible explanations may lead operators to accept "good-looking" explanations without scrutiny—precisely what an attacker desires. To truly leverage explanations for trustworthy AI, we, thus, need an even better understanding of their limits under attacks.

The foundations presented in this thesis bring the research community closer to the responsible usage of XAI in high-stakes applications in adversarial environments, where a single successful attack can have severe consequences.

1.1 Thesis Contributions

This thesis builds on the following contributions made by the author of this thesis and his co-authors:

Chapter 3: Systematization of Attacks Against Explanations. We systematize explanation-aware attacks against explanations along the adversary's capabilities, constraints, and objectives [NW24b]. We formalize robustness notions and establish a hierarchy among them. Notably, we find that robustness notions can conflict when adversaries pursue diverging objectives. We taxonomize existing defenses along the standard machine learning pipeline and relate them to defenses against prediction-only attacks. For each identified category, we raise research questions that will help advance defenses against explanation-aware attacks.

Chapter 4: Explanation-Aware Backdoors. We demonstrate that an AI model can be manipulated such that gradient-based, activation-based, and relevance-based explanation methods display an adversary-chosen explanation upon the mere presence of a trigger pattern in the input [NPW23]. We evaluate these *explanation-aware backdoors* across the three attack types identified in our systematization [NW24b]. Furthermore, we bypass two explanation-based defenses, SentiNet [CTP20] and Fadruus [DAR20], and extend our attack to an Android malware classifier. We analyze the transferability of our attack across different explanation methods and present a "mixed" variant that successfully targets three explanation methods simultaneously. Finally, we show that basic defenses remain ineffective against our attack.

Chapter 5: Composite Explanation-Aware Attacks. We introduce composite explanation-aware attacks based on a threat model not previously discussed in the literature in this context [NW25]. Our approach decouples the two adversarial objectives—influencing predictions and forging explanations—via model-manipulation and input-manipulation attacks, respectively. We evaluate our attacks on image classification, targeting two explanation methods and four different target explanations. Additionally, we analyze how effectively a backdoor trigger can be disguised at inference time and whether our attacks transfer to other models in a black-box setting. Interestingly, we discover that fooling the explanation is more difficult in unmanipulated areas of the image—an observation specific to explanation-aware attacks.

Chapter 6: Explanation-Aware Adversarial Training. We propose explanation-aware adversarial training [Hah+26] and present extensions for the popular adversarial-training techniques PGD-AT [Mad+18a], TRADES [Zha+19], and MART [Wan+20c]. We conduct an extensive evaluation of existing adversarial training approaches alongside our proposed extensions to assess their effectiveness against explanation-aware adversarial examples. Moreover, we reevaluate three existing defenses specifically proposed to mitigate explanation-aware attacks: ATEX [Tan+22], CRC [Joo+23], and RAR [Che+19c]. We conduct two comprehensive transferability studies: First, we examine how well defenses transfer to other explanation-aware attack scenarios. Second, we investigate whether the defender must anticipate the adversary's target explanation.

Contributions Not Included in the Thesis

1. **Model-Manipulation Attacks Against LIME.** We demonstrate explanation-aware model-manipulation attacks [HNW24] against the popular black-box explanation methods LIME [RSG16] and SHAP [LL17], and RISE [PDS18] in the image domain and for tabular data. We lift explanation-aware attacks from manipulating the model directly to the data-poisoning setting, showcasing indiscriminative poisoning attacks, fairwashing attacks, and explanation-aware backdoors.

2. **Stealthy Data-Poisoning Attacks Against Code Models.** We formulate generalized adversarial code suggestions that systematically unify and extend prior work, enabling truly flexible, targeted, and stealthy attacks against Code-LLMs [RNW25]. We define these attacks over a trigger pattern and a learnable mapping between an origin and a bait token. With this formulation, we introduce two novel attacks that eliminate the need for shared tokens between trigger and bait, making them applicable to any vulnerability in any context. Additionally, we sketch a prompt-indexing attack that allows deciding on the bait at inference time. We extensively evaluate our attacks on Python, also comparing them against existing work [Agh+24]. Our investigation of defenses reveals that all but fine-pruning [LDG18] offer little protection.

3. **New Black-Box Explanation Method POMELO.** In our work [ANW25], we integrate full-input perturbation strategies into the popular black-box explanation method LIME [RSG16], yielding our new explanation method POMELO. We identify three key properties of full-input neural samplers and analyze the trade-offs between them. Our experiments demonstrate that full-input perturbations are measurably more in-distribution and exhibit fewer spurious class correlations. We evaluate POMELO across its explanation quality, diversity, locality, distribution alignment, and computational feasibility. Notably, we find that the descriptive accuracy metric is tightly coupled to the perturbation strategy employed by the explanation method.

4. **Covert Attacks via GPIO Connected LEDs.** Our work [Küh+21] presents a novel method for exchanging data with air-gapped devices by pointing lasers at built-in LEDs, utilizing the LEDs as receivers even though they are designed to emit light, rather than to receive it. In comparison to related work on covert channels, we improve the data rate by a factor of 25 and realize a speed-up factor of 110 for data infiltration over distances in the order of tens of meters. For instance, we are able to bridge $25\,\mathrm{m}$ at $18.2\,\mathrm{kbps}$ for infiltrating and $100\,\mathrm{kbps}$ for exfiltrating data using LEDs already built in office devices.

5. **Plausible Deniability for Anonymous Communication.** We analyze the limits of existing privacy notions [Kuh+19] and prove a highly restrictive performance bound for peer-to-peer networks under these notions [Kuh+21]. Thus we demonstrate the importance of weaker notions of privacy and introduce a general game-based formalization of plausible deniability in Anonymous Communication Networks (ACNs). The versatility of indistinguishability games allows us to generalize to arbitrary communication properties and to relate this new notion family to *function views* [HS04].

1.2 List of Publications

This thesis is based on four of our publications [NW24b; NPW23; NW25; Hah+26], listed below and marked with the respective chapter number. The other publications provide additional details on this thesis's topic but are not included as separate chapters [HNW24; NNW26; NW23b; HNW25; NW24a; NW23a]. In agreement with all the co-authors, most figures, tables, and text in the Chapters 3 to 6 originate from these four core publications and appear either verbatim, shortened, or extended in this thesis.

Ch. 3 **SoK: Explainable Machine Learning in Adversarial Environments**. [NW24b]
Maximilian Noppel and Christian Wressnegger.
Proc. of 45th IEEE Symposium on Security and Privacy (S&P), May 2024.

MN did the literature research and the systematization. MN and CW wrote the paper.

- **A Brief Systematization of Explanation-Aware Attacks**. [NW24a]
  Maximilian Noppel and Christian Wressnegger.
  Proc. of 47th German Conference on Artificial Intelligence (KI), Sep 2024.

  The paper is an extended abstract of [NW24b]. MN wrote the paper. CW refined and proof-read the paper.

Ch. 4 **Disguising Attacks with Explanation-Aware Backdoors**. [NPW23]
Maximilian Noppel, Lukas Peter, Christian Wressnegger.
Proc. of 44th IEEE Symposium on Security and Privacy (S&P), May 2023.

MN and LP did the initial experiments to verify a proof-of-work. MN did the final experiments and evaluations. MN and CW wrote the paper.

- **What You See is Not What You Get: Model-Manipulation Attacks Against Black-Box Explanations**. [HNW24]
  Achyut Hegde, Maximilian Noppel, Christian Wressnegger.
  Proc. of the Annual Computer Security Applications Conference (ACSAC), Dec 2024.

  MN had the initial idea. AH did the experiments and evolved the idea together with CW. AH wrote the initial version of the paper. MN, AH, CW refined the paper together.

- **Poster: Fooling XAI with Explanation-Aware Backdoors**. [NW23b]
  Maximilian Noppel and Christian Wressnegger.
  Proc. of ACM SIGSAC Conference on Computer and Communications Security (CCS) Poster, Nov 2023.

  MN wrote the paper and made the experiments. CW refined and proof-read the paper.

- **Explanation-Aware Backdoors in a Nutshell**. [NW23a]
  Maximilian Noppel and Christian Wressnegger.
  Proc. of 46th German Conference on Artificial Intelligence (KI), Sep 2023.

  The paper is an extended abstract of [NPW23]. MN wrote the paper. CW helped with the refinement of the paper.

- **Makrut Attacks Against Black-Box Explanations**. [HNW25]
  Achyut Hegde, Maximilian Noppel, Christian Wressnegger.
  Proc. of 48th German Conference on Artificial Intelligence (KI), Sep 2025.

  The paper is an extended abstract of [HNW24]. AH wrote the paper. MN and CW refined and proof-read the paper.

Ch. 5 **Composite Explanation-Aware Attacks**. [NW25]
Maximilian Noppel, Christian Wressnegger.
Proc. of 8th Deep Learning Security and Privacy Workshop (DLSP) co-located with the 46th IEEE Symposium on Security and Privacy (IEEE S&P), May 2025.

MN made the experiments and wrote the paper. CW refined the paper, in particular the introduction and the conclusion and gave helpful feedback on the structure on the paper.

Ch. 6 **On the Effectiveness of Explanation-Aware Adversarial Training**. [Hah+26]
David Hahnemann*, Maximilian Noppel*, Luan Ademi, Christian Wressnegger.
KIT Scientific Working Papers 268, Karlsruhe Institute of Technology (KIT), Feb 2026.

MN had the initial idea. DH wrote his Bachelor's Thesis on the topic under the supervision of MN and CW. DH made the main experiments and evaluations. LA helped with the experimentation and the evaluation of the results. DH wrote the initial draft of the paper. MN extensively revised and refined the paper. CW refined the paper and gave helpful feedback during the writing process.

* *both authors contributed equally*

1.2.1 Additional Student's Theses Contributions

In addition to the publications listed above, the author of this thesis has supervised 6 Master's and 5 Bachelor's theses. The theses that are related to the topic of this thesis are listed below in chronological order.

- **Explanation-Aware Backdoors for Transformers**. [Pet23]
  Master's Thesis of Lukas Peter, from *Nov 2022* to *May 2023*.

- **Explanation-Aware Backdoors through Data-Poisoning**. [Hel24]
  Master's Thesis of Thassilo Helmoldt, from *Sep 2023* to *Mar 2024*.

- **Manipulating Counterfactual Explanations using Honeypots**. [Lai24]
  Bachelor's Thesis of Fabio Lais, from *Oct 2023* to *Feb 2024*.

- **Explanation-Aware Backdoor Attacks on ProtoPNet**. [Bal24]
  Bachelor's Thesis of Lukas Baldner, from *Jun 2024* to *Oct 2024*.

- **Towards Robust Explanations with Adversarial Training**. [Hah25]
  Bachelor's Thesis of David Hahnemann, from *Oct 2024* to *Feb 2025*.

1.2.2 Additional Publications

The following list contains additional papers authored and co-authored by the author of this thesis in chronological order. For each publication the respective contributions are listed below.

- **Exploiting Contexts of LLM-based Code-Completion**. [NRW25]
  Maximilian Noppel*, Karl Rubel*, Christian Wressnegger.
  Proc. of 48th German Conference on Artificial Intelligence (KI), Sep 2025.

  The paper is an extend abstract of [RNW25]. MN wrote the paper. KR and CW refined and proof-read the paper.

- **POMELO: Black-Box Feature Attribution with Full-Input, In-Distribution Perturbations**. [ANW25]
  Luan Ademi*, Maximilian Noppel*, Christian Wressnegger.
  Proc. of 3rd World Conference on eXplainable Artificial Intelligence (XAI), Jul 2025.

  MN had the initial idea and implemented the prototyp. LA wrote his Bachelor's thesis on the topic under the supervision of MN. LA has implemented and run all the experiments for the paper. MN and LA wrote the paper together. CW refined the paper.

- **Generalized Adversarial Code-Suggestions: Exploiting Contexts of LLM-based Code-Completion**. [RNW25]
  Karl Rubel*, Maximilian Noppel*, Christian Wressnegger.
  Proc. of 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS), Aug 2025.

  CW had the initial idea. MN and KR evolved the idea while writing the expose for KR's Master's thesis. KR finalized the idea and made most of the experiments in the context of his Master's thesis under the supervision of MN. KR, MN, CW wrote the paper together.

- **LaserShark: Establishing Fast, Bidirectional Communication into Air-Gapped Systems**. [Küh+21]
  Niclas Kühnapfel, Stefan Preußler, Maximilian Noppel, Thomas Schneider, Konrad Rieck, Christian Wressnegger.
  Proc. of 37th Annual Computer Security Applications Conference (ACSAC), Dec 2021.

  NK did the experiment in context of his Master's thesis under the supervision of CW. NK and CW wrote the paper together. MN did the study on the GPIO usage in the Linux Kernel and wrote the respective parts in the paper. MN proof-read the paper.

- **Plausible Deniability for Anonymous Communication**. [Kuh+21]
  Christiane Kuhn*, Maximilian Noppel*, Christian Wressnegger, Thorsten Strufe.
  Proc. of 21st Workshop on Privacy in the Electronic Society (WPES), Nov 2021.

  CK had the idea to compare her notions with related work. MN made the comparison in the context of his Master's thesis. CK did the proof of the performance bound. CK and MN did the research. CK, MN, CW, TS wrote and refined the paper together.

- **GI Elections with POLYAS: a Road to End-to-End Verifiable Elections**. [Bec+19]
  Bernhard Beckert, Achim Brelle, Rüdiger Grimm, Nicolas Huber, Michael Kirsten, Ralf Küsters, Jörn Müller-Quade, Maximilian Noppel, Kai Reinhard, Jonas Schwab, Rebecca Schwerdt, Tomasz Truderung, Melanie Volkamer, Cornelia Winter. *(in alphabetic order)*.
  Proc. of E-Vote-ID, Oct 2019.

  MN wrote the verification tool for the Polyas Election System and participated in the election of the GI (ger. "Gesellschaft für Informatik").

* *both authors contributed equally*

1.3 Notation and Symbols

A model is represented by its parameters $\theta = (\theta_w, \theta_{\mathbb{A}}) \in \Theta$, where Θ denotes the space of all possible models. We write θ_w for the model's weights and $\theta_{\mathbb{A}}$ for its architecture, and use $\Theta_{\mathbb{A}} \subset \Theta$ to denote models with a fixed architecture.

Each model parameterizes a decision function $f_\theta : \mathcal{X} \to \mathcal{Y}$ that maps a d-dimensional input $\mathbf{x} = (x^{(1)}, \ldots, x^{(d)})$ from the feature space $\mathcal{X} \subseteq \mathbb{R}^d_+$ to a C-dimensional probability vector over classes $c \in \{1, \ldots, C\} = [C]$, where $\mathcal{Y} \subseteq [0,1]^C$. The predicted class is $F_\theta(\mathbf{x}) := \arg\max_i f_\theta(\mathbf{x})_i$. And $l_\theta(\mathbf{x})$ denotes the logit output of the model.

A dataset of n samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} = \mathbf{X} \subseteq \mathcal{X}$ with ground-truth labels $\mathbf{Y} \in [C]^n$ is denoted $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$.

An explanation method, parameterized by the model θ, quantifies feature relevance for a given prediction: $h_\theta : \mathcal{X} \to \mathcal{E}$, where $\mathcal{E} \subseteq [0,1]^e$ with $e \leq d$ is denoted as the explanation space. Although alternative explanation spaces exist (e.g., $\mathcal{E} \subseteq \mathcal{X}$ for counterfactual explanations), we adopt this formalization throughout this thesis. A single explanation is denoted $\mathbf{r} \in \mathcal{E}$. For certain results, we treat $\mathcal{X}$ and $\mathcal{E}$ as (convex) metric spaces, $(\mathcal{X}, d_\mathcal{X})$

and $(\mathcal{E}, d_{\mathcal{E}})$, with distance metrics $d_{\mathcal{X}}$ and $d_{\mathcal{E}}$. Occasionally, we also use $d_{\mathcal{X}}$ and $d_{\mathcal{E}}$ for dissimilarity measurements that do not necessarily satisfy metric or norm axioms.

We denote clean samples as $\hat{\mathbf{x}}$, manipulated samples as $\tilde{\mathbf{x}}$, original models as $\hat{\theta}$, manipulated models as $\tilde{\theta}$, and robustified models as $\overline{\theta}$.

We use the symbols ⁘, ⁞⁞, ⁘ to denote explanation-targeted, explanation-untargeted, and explanation-semitargeted attacks. Additionally, we use ● and ❍ for prediction-targeted and prediction-untargeted attacks. The specifics of theses attacks are detailed in Chapter 3.

Table 1.1 summarizes the notation and symbols used in this thesis.

Table 1.1: Overview on the notation and symbols used in this thesis.

Symbol	Description
$\mathbb{R}$,$\mathbb{R}_+$,$\mathbb{R}_{+,0}$,$\mathbb{N}$,$\mathbb{N}_+$,$\mathbb{N}_{+,0}$	Real/natural numbers, positive excluding 0,including 0
$\mathcal{X} \subseteq \mathbb{R}^d$	Feature space (input space) as a subspace of $\mathbb{R}^d$
$\mathcal{Y} \subseteq [0,1]^C$	Output space (probability scores)
C, $[C]$	Number of classes (scalar), Set of class indices $\{1, \ldots, C\}$
$\theta = (\theta_w, \theta_{\mathbb{A}})$	Model parameters, weights/biases θ_w, architecture $\theta_{\mathbb{A}}$
$\hat{\theta}$, $\tilde{\theta}$, $\overline{\theta}$	Original model, manipulated model, robustified model
Θ, $\Theta_{\mathbb{A}}$	All possible models, models with architecture $\mathbb{A}$
$l_\theta : \mathcal{X} \to \mathbb{R}^C$	Logits function (logits)
$f_\theta : \mathcal{X} \to \mathcal{Y}$	Decision function (soft labels)
$F_\theta : \mathcal{X} \to [C]$	Classification function (hard labels)
$h_\theta : \mathcal{X} \to \mathcal{E}$	An explanation method, parameterized with θ
$\mathbf{x}$, $\hat{\mathbf{x}}$, $\tilde{\mathbf{x}} \in \mathcal{X}$	Any, clean, and manipulated input sample
$x^{(i)}$	The i-element of the $\mathbf{x}$ vector (a scalar)
$\hat{y}, y, y_t \in [C]$	Clean / ground-truth, predicted, target label
$\mathbf{r} = (r^{(1)}, \ldots, r^{(\leq d)}) \in \mathcal{E}$	A single explanation
$\mathcal{E} \subseteq \mathbb{R}_+^{\leq d}$	Explanation space as a subset of $\mathbb{R}_+^{\leq d}$
$\mathbf{x} \oplus \mathcal{T}$	Adding a trigger to sample $\mathbf{x}$
$\mathcal{D} \subset \mathcal{X} \times [C]$	Dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
$\mathbf{X} \subseteq \mathcal{X}$	Set of input samples
$\mathbf{Y} \in [C]^n$	Set of ground-truth labels
$\mathcal{S} : \mathcal{X} \times \Theta \to [C] \times \mathcal{E}$	An explainable system
$(\mathcal{X}, d_{\mathcal{X}})$	Input space as metric space with distance metric $d_{\mathcal{X}}$
$(\mathcal{E}, d_{\mathcal{E}})$	Explanation space as metric space with distance metric $d_{\mathcal{E}}$
$\mathbf{0}$, $\mathbf{1}$	The all-zero vector, the all-one vector
⁘ / ⁞⁞ / ⁘	explanation targeted/untargeted/semitargeted objectives
● / ❍	prediction targeted/untargeted objectives

Explainable Artificial Intelligence in a Nutshell

2

Contents

Chapter Preface.
This chapter provides a focused overview of XAI. Our goal is to equip the reader with the necessary background knowledge to comprehend the subsequent chapters. We begin by motivating why explanations are essential, particularly in high-stakes applications. We then explore the promised benefits of explanations, and various explanation methods, with a focus on feature attribution methods, whose (in)security we investigate in later chapters. Readers familiar with these concepts may skip ahead and return as needed. Note that individual parts of this chapter are taken and adapted from our original works [NW24b; NPW23].

2.1 The Need for Explainable Artificial Intelligence

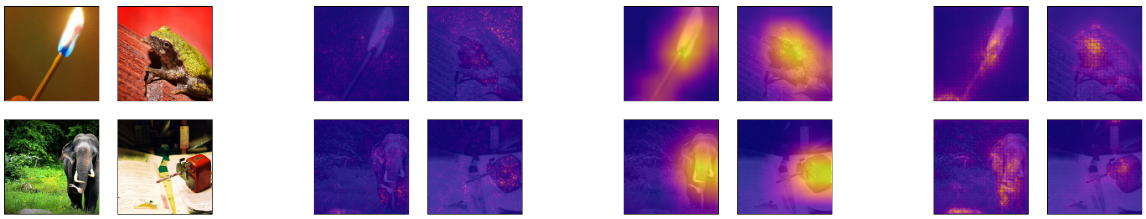
Ever since the wide adoption of artificial intelligence, the community has striven for ways to explain the inner workings of learned models [Lip18; Lap+19]. The complexity and non-linearity of modern deep learning schemes have, however, raised the bar to do so distinctively [Sam+21]. In contrast to simple linear models, (deep) neural networks are *not inherently interpretable* [Mur+19; Rud19]. This opaqueness of deep neural networks stems mainly from the highly non-linear interactions of learned parameters, which, in turn, are required to achieve sufficient performance in complex tasks, such as image classification.

Recent research has, thus, brought forward various *post-hoc explanation methods* to shed light onto the inner reasoning of readily trained neural networks.

Post-hoc explanation methods are diverse and range from counterfactual explanations [e.g., Gui24; WMR17; VDH21; Cha+19; Gin86; Lip90; DGK21; Gom+20; MST20; Poy+20; Kan+20; LWL20; Jia+24; Jia+23; Ser+24], over concept-based explanations [e.g., Kim+18; Cho+24; Gho+19], self-explaining model architectures [e.g., Che+19b; Li+22; AJ18b; ZFP19], and feature visualization methods [e.g., Erh+09; OMS17; Goh+21; Ola+18; MV15; NYC16; Ngu+17; Ngu+16; MOT15; Øyg15] to feature attribution methods [e.g., Bac+15; SVZ14; STY17; SGK17; Spr+15; Cha+18; Sel+20; FV17; RSG16; LL17; Smi+17]. Unfortunately, the vast number of directions makes it impossible to capture the whole field here.

In the context of this thesis, we instead focus on feature attribution methods. However, we point out that this chapter can still provide only a very brief overview of the field. Fortunately, the community has produced several comprehensive surveys [VL21; AB18; CPC19; Che+21; Chu+23; Dan+20; DR20; Gui+19; Mol22; Zha+21; Søg22; Hol+20; SF24; GKM23; Qin+22; Min+22] and books [NP22; Mol22] on the topic. We refer the reader to these works for further information on XAI.

Feature Attribution Methods. Feature attribution methods explain the decision-making process of a model by attributing relevance scores to input features [AB18; Gui+19; Zha+21]. As a consequence of this general definition, feature attributions can be visualized as overlays and heatmaps on top of inputs. In particular, in the image domain this is common practice. For text, the background color of individual tokens or words can indicate their relevance as well. Figure 2.1 depicts examples of three feature attribution methods used in this thesis. In all our heatmap depictions, yellow represents important regions while purple color represents unimportant regions.



(a) Simple Gradients **(b)** GradCAM **(c)** SmoothGrad

Figure 2.1: Example explanations using different gradient-based methods: **(a)** Simple Gradients shows the raw gradient, which is often scattered, **(b)** GradCAM provides coarser but more stable heatmaps based on activation maps, **(c)** SmoothGrad reduces noise by averaging gradients over multiple perturbed samples.

2.2 The Benefits of Explanations

Next, we iterate over the core benefits of explanations:

Improving Models During Development. A deeper understanding of a model's behavior enables model developers to debug and enhance its predictive performance. For instance, by manually or automatically identifying and removing biases or spurious correlations from training data [Lap+19; And+22; BCD24; KM08]. For instance, Anders et al. [And+22] propose to cluster explanations across training samples and examining outlier clusters manually. This approach reveals spurious correlations tied to fixed spatial positions, such as watermarks on images of specific classes, e.g., the watermarks on horse images in the Pascal VOC 2007 dataset [Eve+10], as also investigated by Lapuschkin et al. [Lap+19]. Another research direction integrates explanations directly into the training process [e.g., Rie+20; RD18; RHD17]. Rieger et al. [Rie+20], for instance, penalize models based on their explanations, aligning them with human-provided annotation maps that encode domain knowledge. Similarly, Ross, Hughes, and Doshi-Velez [RHD17] constrain explanations to ensure models are "right for the right reasons." Thus, explanations serve as a testing mechanism to verify expected model behavior without exhaustive data coverage.

Improved Interactions with End-Users. Explanations benefit not only model creators but also end users, who can interact more effectively with AI systems on the basis of explanations [Kul+13; Kul+12; Kul+10; Liu+17; SF18; Wan+19b; SF20]. Through explanations, users gain insight into a system's behavior and limitations. Users can leverage this knowledge to avoid input patterns that produce unreasonable explanations or confuse the model, e.g., by removing distracting objects from a scene that receive unreasonable attention. Consequently, users can optimize the application environment for better system performance [Kim+23]. Furthermore, explanations enable end users to provide informed feedback to developers about a system's real-world (mis)behavior.

Improving Trust. The hope is that XAI can foster trust in AI systems, and thereby increase the acceptance of the AI system by its users [BS23; Pet22; AL25]. For example, Leichtmann et al. [Lei+23] have conducted a 2×2 between-subjects study with 410 participants who picked mushrooms in a virtual forest and decided whether to eat them based on a virtual AI app. The group provided with additional GradCAM explanations showed better calibrated trust levels. Nevertheless, the concept "trust" is complex, its details reach into the context of user studies and philosophy. The exact connection between explainability and trust, thus, remains an open research question [FL22; Vis+25; Mil19].

Knowledge Discovery. Beyond trust, explanations help researchers and practitioners understand the application domain more deeply [e.g., Won+24; Pun+25]. The reason is that AI systems often exploit correlations not represented in our human ontology. The causality of the identified correlations should then be investigated in further experiments or by human analysts. Any found correlation can help to design simpler systems or to improve existing ones by modifying either the AI system itself or its environment.

Segmentation and Object Localization. The localization of important correlations can be used in an automated system to locate objects in a scene or to segment a scene [Sel+20; RAM22]. The example explanations in Figure 2.1, for instance, clearly highlight the elephant and the sharpener in the lower row. In fact, explanations can be class-specific, i.e.,, the explanation differs depending on which class we explain. Figure 2.6 provides an example for this, making clear that the image can be segmented into two classes.

Detection of Manipulations. In benign environments, explanations highlight the features most influential to a model's decision. In scenarios where adversaries aim to induce mispredictions, these influential features, such as backdoor triggers, are consequently exposed by the explanation [HAS19; LLC21; DAR20; CTP20; NPW23; BS21; Yan+24]. In Chapters 4 and 5 we present attacks that are specifically designed to circumvent detection mechanisms based on explanations.

Fairness. In benigh settings, explanations provide a straightforward way to assess whether a system is fair. Consider, for instance, a system that attributes high relevance scores to sensitive features. In Chapter 3, we discuss that adversaries can attempt to *fairwash* their systems by manipulating them to produce seemingly fair explanations [Aïv+21; Aïv+19].

As such, explainability contributes toward reliable and trustworthy AI [e.g., Nat23; EU18].

2.3 Capabilities of Explanation Methods

Explanation methods come in two major flavors: white-box and black-box explanation, also denoted as model-specific and model-agnostic, respectively. For the remainder of this thesis, we use the terms white box and black box, as this captures our understanding of "what is accessible by the explanation method", rather than "what models can be explained with the method". In the context of explanation-aware attacks, the first understanding is more intuitive. In the following, we introduce both concepts briefly:

2.3.1 White-Box Explanation Methods

The inputs to white-box explanation methods are the sample and the model in which the sample is processed. Crucially, the model is provided with full access to its weights, biases, and its computational graph. This transparency enables white-box methods to perform the following operations:

1. **Calculate Gradients and Intermediate Activations.** Given a differentiable model, the explanation method can compute gradients with respect to both the sample and the model's parameters.
2. **Run Unlimited Inferences.** The explanation method can execute as many forward passes as computationally feasible. While truly unlimited inferences are impractical, the number of queries is far less constrained than for black-box methods. Naturally, white-box methods can also modify the sample at will.
3. **Manipulate the Model.** White-box methods can even alter the model between inferences. For instance, they may randomize or ablate individual layers to study their influence on the model's behavior. In essence, they operate like a surgeon dissecting the model.

2.3.2 Black-Box Explanation Methods

Black-box explanation methods, in contrast, are only given query access to the model without insight into its internals. Specifically, they have the following capabilities:

1. **Limited Number of Inferences.** Black-box methods typically aim to minimize the number of queries, since they are often used to audit models operated by third parties. Nevertheless, their query count remains substantially higher than for most white-box methods. For instance, the relatively expensive white-box method SmoothGrad runs ~ 50 inferences, whereas the popular LIME method defaults to $\sim 2{,}000$ queries.
2. **Access Restricted to System Outputs.** By definition, black-box methods can only observe the inputs and outputs of the system. The precise form of these outputs, however, varies across methods. LIME, for example, requires class probabilities, while other methods need only the predicted class label[1].

[1] Interestingly, the discrepancy between class probabilities and predictions makes LIME vulnerable to model-manipulation similar to the attacks we demonstrate in Chapter 4. This vulnerability has been demonstrated by our work [HNW24].

Remark 1.
Note that we explicitly write "system" instead of "model" here. The reason is that in the auditing scenario, the system includes any preprocessing stage and might even consist of multiple models. In comparison to the white-box case, the different stages usually cannot be separated by black-box methods.

2.4 Approaches for Feature Attribution

In our work, we focus on feature attribution methods, the largest research strand within the XAI field. Here, the explanation is an element of an explanation space $\mathcal{E} \in [0,1]^e$, where $e \leq d$, containing values ranging from 0 to 1 for unimportant to important regions, respectively, or $\mathcal{E} \in [-1,1]^e$ for features that speak for or against the predicted class. In this thesis, we focus on the $\mathcal{E} = [0,1]^e$ space, which is the most commonly used setting. Importantly, the selected colormap for the overlay must respect the respective case. Also, in principle, all explanations in $[-1,1]^e$ can be transformed into $[0,1]^e$ by taking the absolute value, but not the other way around [SVZ14].

Explanation Spaces are Subsets. Oftentimes, importance scores are not assigned to each individual feature but to sets of features, e.g., to pixels instead of individual channels [e.g., Sel+20; Smi+17] or even to superpixels[2] [RSG16; Wei+18]. This practice has several reasons: Firstly, the explanations need to be comprehensible to the recipient. Importance scores per color channel are hard to visualize and interpret. Even the importance of individual pixels can sometimes be hard to interpret. For instance, the explanation method Simple Gradients [SVZ14] often shows scattered importances without a clear structure, as can be seen in Figure 2.1a. Secondly, the reason often is inherent to the generation process of the explanations themselves. One such example is the LIME method, which we describe in more detail in Subsection 2.4.1.1.

The approaches to explain are extremely diverse [e.g., BH21; Gui+19; VL21; LPK21]. In this section, we provide a brief overview of perturbation-based, gradient-based, propagation-based, and activation-based explanation methods.

2.4.1 Perturbation-Based Explanations

Perturbation-based explanation methods measure how specific changes to a sample affect the model's response. A prominent approach suggests "removing" parts of the sample to nullify their influence [RSG16]. In theory, one would simply compute the marginal probabilities [CLL21; JMB20; Hes+20]. In practice, however, this is often intractable. For sequential or token-based inputs, such as natural language and source code, tokens can be deleted. The task becomes substantially more challenging for time series, where erasing a segment may introduce discontinuities, and for images or tabular data, where models require fixed-length inputs. Moreover, models often rely on interactions between columns in

[2] Superpixels are a set of pixels. For images, for instance, these superpixels are found by segmentation algorithms like Quick Shift [VS08] or Simple Linear Iterative Clustering (SLIC) [Ach+12].

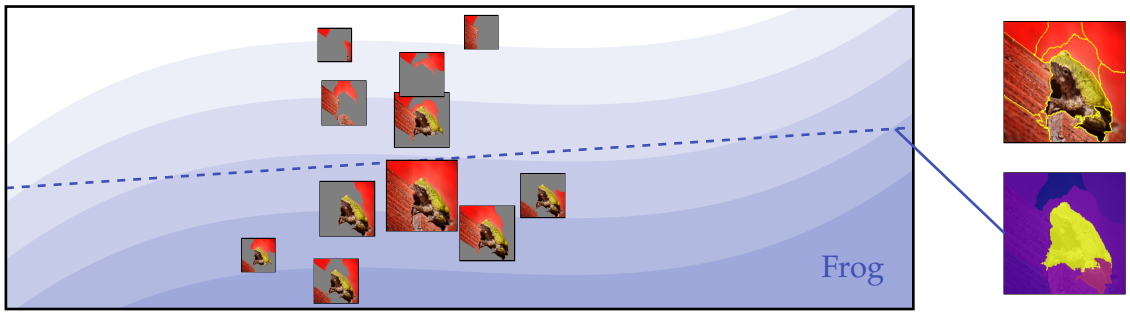


Figure 2.2: Overview of the LIME method. The probability score of the class "`frog`" is indicated with blue contour lines. Further, the original sample is in the middle and the perturbed variants around it. Their size indicates the distance to the original sample. The blue dashed line represents the linear surrogate model.

tabular data or spatial relationships within images. A baseline behavior for missing features may never have been learned, pushing the perturbed sample out of distribution [ANW25]. The key questions thus are: which features to remove and how to perform the removal operation. Put differently, the explanation method explains with respect to the chosen perturbation strategy.

2.4.1.1 Example: Local Interpretable Model-agnostic Explanations (LIME)

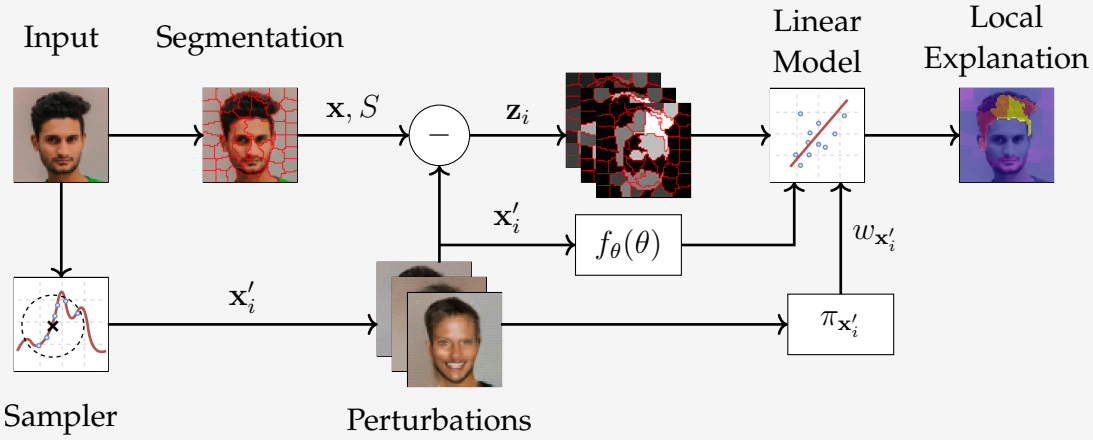
A prime example of perturbation-based explanation is the explanation method LIME [RSG16]. LIME works in four steps as visualized in Figure 2.2:

1. **Segmentation.** The sample is partitioned into m superpixels using standard segmentation algorithms such as Simple Linear Iterative Clustering (SLIC) [Ach+12].
2. **Perturbation.** Based on this segmentation, N perturbed variants of the original sample are generated. In Figure 2.2, we use the default strategy: replacing segments with gray.
3. **Querying the Model.** For each perturbed sample, LIME queries the model for the probability of the initially predicted class, indicated by the blue contour.
4. **Learning a Surrogate Model.** Using N binary perturbation vectors in $\{0, 1\}^m$, where 0 denotes a preserved segment and 1 denotes a removed segment, along with the corresponding class probabilities, a simple surrogate model is fitted. In Figure 2.2, this surrogate is a linear model, depicted as a blue dashed line. The weights of this surrogate model constitute the final explanation.

Extensions to the Framework. This basic framework has inspired numerous extensions [Kna+25]. For instance, RISE [PDS18] employs randomized masks, MASK [FV17] learns optimal masks directly, Knab, Marton, and Bartelt [KMB24] introduce hierarchical segmentations, and Qiu et al. [Qiu+22] penalize out-of-distribution perturbations. Alternative perturbation strategies, such as blurring, random noise, or mean-color replacement, have also been explored [FV17; GM21]. Beyond perturbation design, SHAP [LL17] leverages game-theoretic Shapley values to quantify the features' contributions, LEMNA [Guo+18] fits multiple surrogate models, and ROPE [LAB20] robustifies the surrogate model against distribution shifts. Furthermore, we have proposed POMELO [ANW25].

Excursus 1.

LIME has three core limitations, which we address with our proposed explanation method POMELO [ANW25]: Firstly, the perturbed samples queried by LIME are heavily out-of-distribution, containing patches of constant colors, such as black or gray, or blurred regions. The intended goal of "removing" patches is not achieved with this approach. Instead, the model is queried in regions of the input space where no training data exists, rendering its responses unreliable. Secondly, replacing approximately $50\,\%$ of patches at random implicitly assumes that correlations are spatially localized *within* individual patches or within a small number of patches. We argue that correlations are often distributed across the entire image. Consider, for instance, the correlation between hair color and eye color in the CelebA dataset [Liu+15]. Furthermore, some features may span multiple patches, such as the hair. The probability that all hair-related patches are removed simultaneously is relatively low, meaning the model rarely observes samples where all hair-patches are removed. Thirdly, using constant replacement values introduces uncontrolled correlations. For instance, in our work [ANW25], we demonstrate that black replacement leads to a disproportionate increase in black-hair classifications, systematically biasing the explanation.



To address these limitations, we propose employing neural samplers, i.e., generative models, to produce full-input perturbations. Crucially, the sampler should be trained on the same data as the model being explained. This requirement implies that we explain the model *jointly* with its underlying training distribution.

The figure above illustrates our method POMELO. The sampler generates realistic full-input perturbations $\mathbf{x}'_i$. The segment-wise pixel difference (indicated by the minus operation) yields one floating-point value per segment and perturbation $\mathbf{z}_i$. A linear model is then trained to predict the softlabel based on these per-segment similarities, weighted by the distance between each perturbed sample and the original sample, denoted as $w_{\mathbf{x}'_i}$ and computed via a kernel $\pi_{\mathbf{x}'_i}$. Notably, our perturbations are floating-point values rather than LIME's binary indicators (0 for present or 1 for "removed"). Finally, the weights of the resulting linear model constitute the explanation, similar to LIME. Related work by Ghalebikesabi et al. [Gha+21] suggests that such an approach yields more robust explanations.

2.4.2 Gradient-Based and Propagation-Based Explanations

Another branch of research focuses on back-propagating relevances through the model. Here, we introduce the relevant explanation methods in this field.

2.4.2.1 Example: Simple Gradients (SG)

The gradient of the model with respect to the sample measures how sensitive the model's output is to infinitesimally small changes to the sample. For linear models, the gradient captures the global decision surface perfectly, similar to why LIME uses the weights of the linear surrogate model as the explanation. For deep neural networks, however, the gradients are noisy and scattered [Bal+17]. The decision surface, thus, should not be imagined as a smooth surface but rather as a rough terrain with steep edges and valleys.

At the same time, the gradient satisfies interesting theoretical properties, such as the fundamental theorem of line integrals. In this regard, incorporating gradient information into the explanation is a natural choice.

Indeed, the gradient lies at the heart of the earliest explanation methods, initially proposed by Baehrens et al. [Bae+10] for Support Vector Machines (SVMs) and k-nearest neighbors and later refined for deep neural networks by Simonyan, Vedaldi, and Zisserman [SVZ14]. More concretely, Simonyan, Vedaldi, and Zisserman [SVZ14] take the *absolute* gradient as the explanation:

$$h_\theta^{SG}(\mathbf{x}) := \left| \frac{\partial \max_c l_\theta(\mathbf{x})_c}{\partial \mathbf{x}} \right| .$$

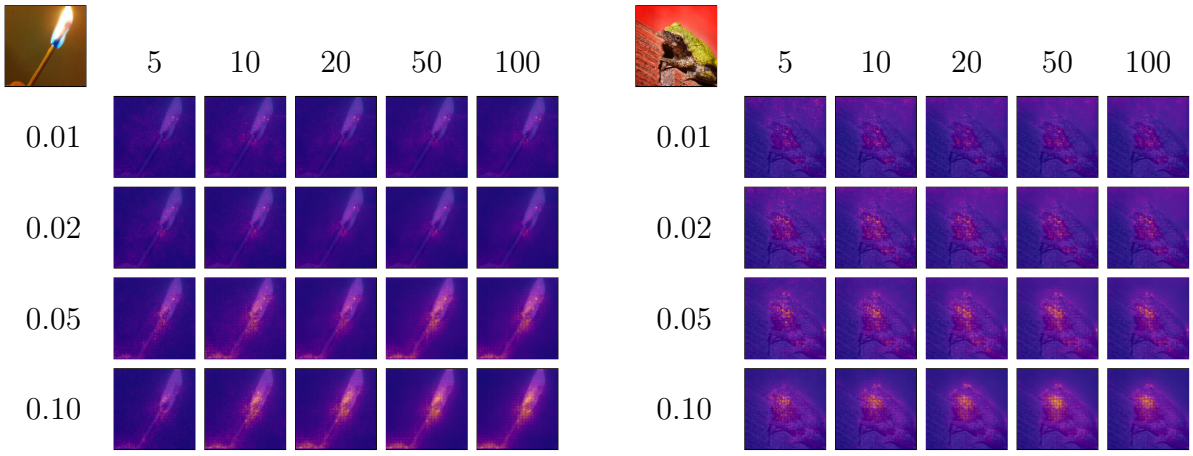
Unfortunately, this formulation no longer satisfies the fundamental theorem of line integrals. Furthermore, in the Simple Gradients method, the channels of the input are $\max$-aggregated to obtain one score per pixel and to "denoise" the explanation [SVZ14]. Lastly, explanations are typically scaled to the interval $[0, 1]$ for better visualization. Hence, the theoretical properties of the Simple Gradients explanation method differ distinctively from those of the raw gradient, a fact often overlooked by related work [Dom+22]. Nevertheless, gradients serve as the foundation for more advanced explanation methods.

2.4.2.2 Example: Gradients $\times$ Input (GI)

The gradient represents the effects of changing a feature within an infinitesimal small vicinity but (strictly speaking) does not represent relevance. This problem has been addressed by multiplying the gradient with the sample [SGK17; Kin+16; Shr+16], commonly referred to as Gradients $\times$ Input:

$$h_\theta^{GI}(\mathbf{x}) := \frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}} \odot \mathbf{x} .$$

In short, this multiplication aims to capture relevance more faithfully instead of only responses to potential changes, independent of the current value.



(a) Sample 1 **(b)** Sample 2

Figure 2.3: Different parameterization of the SmoothGrad explanation method for two samples from the ImageNet dataset [Den+09; Rus+15]. The number of samples increases from left to right (5 to 100). The amount of noise added to the individual samples increases from top to bottom (0.01 to 0.1).

2.4.2.3 Example: Integrated Gradients (IG)

Other work [STY17] has approached the problem by integrating over the gradient with respect to a root/anchor point $\mathbf{x}'$:

$$h_\theta^{IG}(\mathbf{x}) := (\mathbf{x} - \mathbf{x}') \odot \int_{\alpha=0}^{1} \frac{\partial f_\theta(\mathbf{x}' + \alpha \cdot (\mathbf{x} - \mathbf{x}'))}{\partial \mathbf{x}} \, \mathrm{d}t \; .$$

All three gradient-based approaches suffer from the "shattered gradient" problem [Bal+17], giving rise to more advanced explainability methods, which we discuss next.

2.4.2.4 Example: SmoothGrad

To address this issue, Smilkov et al. [Smi+17] propose mean-aggregating gradients over a neighborhood around the sample $\mathbf{x}$. Specifically, SmoothGrad samples this neighborhood with Gaussian noise, though other distributions have been explored as well. For instance, Wang et al. [Wan+20d] argue that Laplacian smoothing with uniform noise yields more robust aggregations and propose Uniform Gradients (UniGrad). While originally designed for Simple Gradients, the same concept can be applied on top of any explanation method as a *smooth adaptation*:

$$\overline{h}_\theta(\mathbf{x}) := \mathop{\mathbb{E}}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})} \left[h_\theta(\mathbf{x} + \epsilon) \right] \; .$$

For example, Omeiza et al. [Ome+19] apply the smooth adaptation to GradCAM++. The runtime for deriving explanations increases with the number of samples (typically around 50), making smoothed explanations computationally expensive compared to other white-box methods. Figure 2.3 visualizes SmoothGrad explanations across two parameters: the number of samples (x-axis) and the amount of added noise (y-axis).

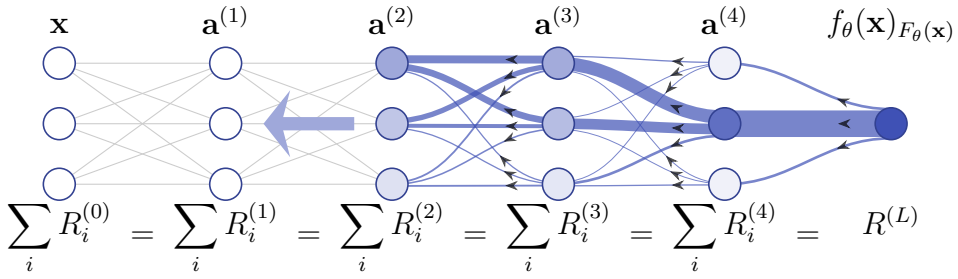


Figure 2.4: LRP back-propagates the relevance scores through the network. This figure inspired from Figure 10.2 by Montavon et al. [Mon+19].

2.4.2.5 Example: Layer-Wise Relevance Propagation (LRP)

A large body of work exists on alternative propagation rules, in contrast to standard gradient back-propagation. This research direction has been initiated by Bach et al. [Bac+15], who proposed a family of explanation methods known as Layer-Wise Relevance Propagation (LRP). In their approach, the function output, i.e., the output neurons' activations, is back-propagated through the model, eventually reaching the input layer where it represents relevance scores. Figure 2.4 visualizes this idea. The basic rule for this back-propagation is the LRP-0 rule:

$$R_j^{(l-1)} = \sum_{k \in l} \frac{a_j w_{jk}}{\sum_{i \in l-1} a_i w_{ik}} R_k^{(l)} \,,$$

where R_i^l denotes the relevance in layer l at neuron i, a_i denotes the activation of neuron i, and w_{ij} is the weight connecting neurons i and j. With slight abuse of notation, we write $k \in l$ to indicate iteration across all neurons in layer l.

Montavon et al. [Mon+19] have demonstrated that applying different rules to different layers improves explanation quality, an approach termed *Composite LRP*. The core rules, beside the LRP-0 rule, include the LRP-ϵ rule and the LRP-γ rule. A detailed understanding of the individual rules and their optimal layer assignments is beyond the scope of this thesis, and we refer the reader to the literature in the field [Mon+19].

The central idea underlying LRP is the *(layerwise) conservation property*, which must hold across all L layers when relevance is propagated from layer to layer. The relevance of all units in layer l must sum to the relevance in the subsequent layer $l+1$:

$$\sum_i R_i^{(0)} = \sum_i R_i^{(1)} = \cdots = \sum_i R_i^{(L)} \,.$$

In other words, the total relevance in each layer remains constant and equals the logit activation of the output neurons. Beyond relevance conservation, several interesting theoretical properties have been established for LRP. For instance, Montavon et al. [Mon+17] have shown that LRP is based on a Deep Taylor Decomposition (DTD) [Mon+17; KMM20].

Subsequently, LRP has been extended to more complex layers and architectures, such as transformer architectures [Ach+24], Long Short-Term Memorys (LSTMs) models [Arr+17], and U-Net architectures [Wei+24].

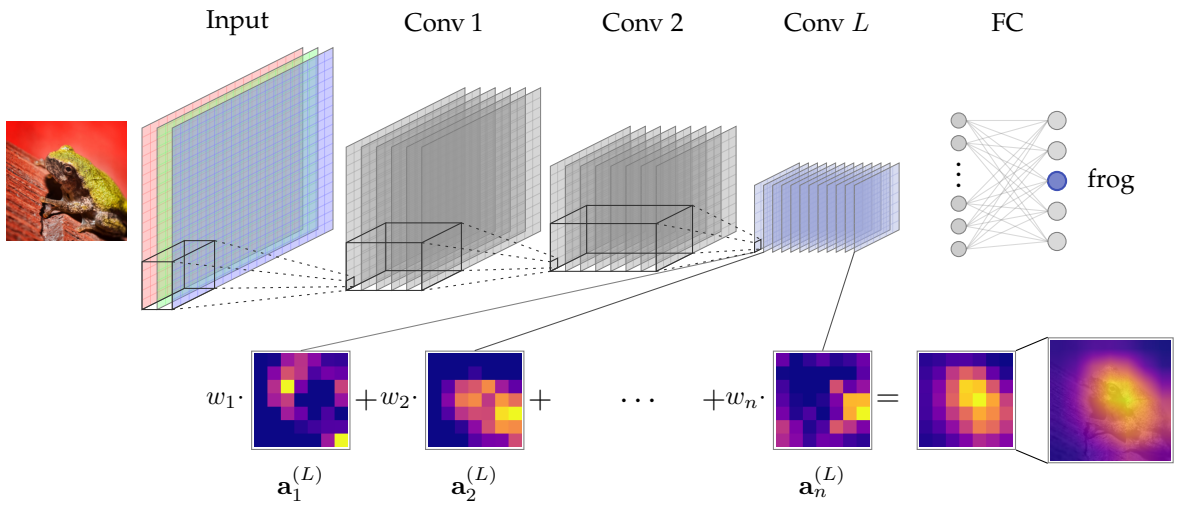


Figure 2.5: Overview of the CAM-based explanation methods. The RGB-channels of the sample are processed by layers that preserve the pixels' spatiality. The explanation method then extracts the activation maps at a certain layer, aggregates them by a weighted addition, and upscales them to the input size. The difference between CAM, GradCAM, and Relevance-CAM lies in the weighting of the activation maps.

2.4.3 Activation-Based Explanations

Activation-based explanation methods use the activations of the individual neurons. Here, we focus on CAM-based methods as the key representative of activation-based methods.

2.4.3.1 Example: Class Activation Map (CAM)

The explanations of the CAM method [Zho+16] arise from aggregating and up-scaling activations at the penultimate layer of Convolutional Neural Networks (CNNs), which is the last layer where the spatiality of the sample is preserved. Thus, CAM is primarily applicable in the image domain and in 1D domains, such as time series. A major drawback of CAM is that it is only applicable to specific model architectures that are based on a Global Average Pooling (GAP) of the activation maps in this penultimate layer L:

$$f_\theta(\mathbf{a}_1^{(L)}, \ldots, \mathbf{a}_n^{(L)})_c = \sum_k^n w_{k,c} \cdot \left(\frac{1}{|L|} \sum_{i \in L} a_{k,i}^{(L)} \right) ,$$

where $\mathbf{a}_1^{(L)}$ to $\mathbf{a}_n^{(L)}$ are the activation maps at the L^{th} layer and $a_{k,i}^{(L)}$ is the activation of the i^{th} unit in the k^{th} channel, and $w_{k,c}$ are learned weights for class c. For such models, the explanations for class c are generated as

$$h_\theta^{CAM}(\mathbf{x}; c) := \text{upscale} \left(\sum_k^n w_{k,c}\, \mathbf{a}_k^{(L)} \right) ,$$

where upscale scales the lower resolution activation maps to the input's resolution by using bi-linear interpolation. In Figure 2.5 we visualize the fundamentals of CAM-based methods for a schematic CNN.

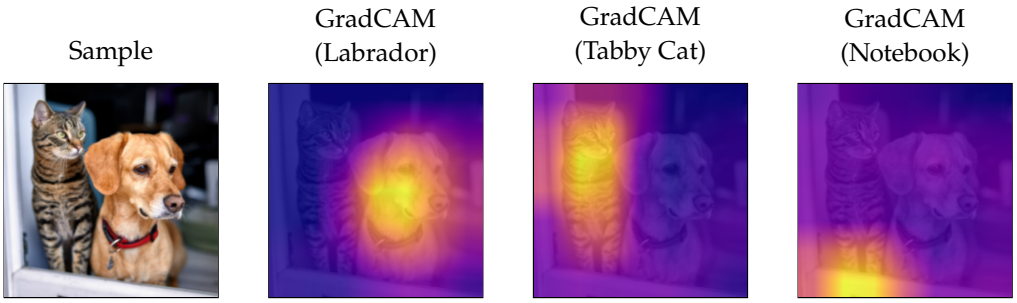


Figure 2.6: The GradCAM explanations can be used to explain any class—they are class-specific. In this figure, we explain the classes: labrador, tabby cat, and notebook. As can be seen, the explanation of the tabby cat highlights the cat, while the explanation of the labrador highlights the dog. Also, the model seems to have picked up features of a notebook at the bottom. This illustrates the localization capabilities of GradCAM.

This dependency on the class c renders CAM-based explanation methods *class-specific*. Figure 2.6 illustrates this effect for the GradCAM method (described in the next section) with an image showing a dog and a cat, similar to Figure 3 by Selvaraju et al. [Sel+20]. The explanations for the class "`dog`" clearly highlight the dog, while the explanation for "`cat`" focuses on the cat. Class-specific explanation methods can, thus, be used to perform object detection and solve localization tasks [Sel+20]. Figure 2.6 additionally highlights how class-specific explanation methods can be use to debug the model. Apparently, the applied ResNet20 [110] model has also picked up the window frame as a notebook. In line with related work, we always explain the predicted class in this thesis.

Again, the fact that CAM is only applicable to models with a GAP limits its application in practice drastically.

2.4.3.2 Example: GradCAM (G-CAM)

GradCAM [Sel+20] lifts this limitation by using back-propagated gradients, averaged across each channel, as the weighting of the activation maps:

$$w_{k,c} := \frac{1}{|L|} \sum_{i \in L} \frac{\partial f_\theta(.)_c}{\partial a_{k,i}^{(L)}} \,.$$

This adaptation renders CAM-based methods applicable to any neural network that begins with convolutional layers, followed by any number and type of differentiable layers. To illustrate this generality, Selvaraju et al. [Sel+20] apply GradCAM to image classification, image captioning, and visual question answering. The resulting relevance maps are smooth and largely continuous, making GradCAM explanations comparatively easy to interpret and widely adopted. To date, the paper by Selvaraju et al. [Sel+20] has been cited $36{,}871$ times on Google Scholar, making it one of the most cited works in the XAI community. For comparison, LIME [RSG16] has been cited $30{,}255$ times, Simple Gradients [SVZ14] $10{,}879$ times, and LRP [Bac+15] $6{,}468$ times[3]. This widespread adoption makes GradCAM an ideal candidate for investigating its potential vulnerabilities.

[3] Citation counts last updated on February 5th, 2026.

2.4.3.3 Example: Relevance-CAM (R-CAM)

More recently, Lee et al. [Lee+21] proposed using LRP relevances instead of gradients to weight the activation maps:

$$w_{k,c} := \frac{1}{|L|} \sum_{i \in L} R_{k,i}^{(L)} \ .$$

This adaptation circumvents the vanishing gradient problem, whereby gradients become increasingly noisy and eventually vanish as they are propagated toward the input. Consequently, Relevance-CAM is better suited to explaining earlier layers than the penultimate layer, enabling finer-grained resolution. In this thesis, however, we explain only the last layer and leave further investigation of other parameterizations to future work.

Besides the introduced CAM variants, multiple others exist. For instance, ScoreCAM [Wan+20a], CodCAM [OC23], GradCAM++ [Cha+18], Smoothed GradCAM++ [Ome+19], LIFT-CAM [JO21], and recently OptimCAM [Zha+24a].

2.5 Chapter Summary

In this chapter, we provide a brief overview of the research field of eXplainable Artificial Intelligence (XAI). We introduce feature attribution methods as the class of explanation methods on which this thesis focuses. Within this class, we discuss gradient-based, propagation-based, and activation-based explanation methods, emphasizing those employed throughout this thesis.

For further information on the field, we again refer the reader to the numerous available surveys and books [VL21; AB18; CPC19; Che+21; Chu+23; Dan+20; DR20; Gui+19; Mol22; Zha+21; Søg22; Hol+20; SF24; GKM23; Qin+22; Min+22; NP22; Mol22]. Given the vast number of explanation methods and adaptions proposed in recent years, it remains challenging to determine which method is most relevant. This difficulty arises because different methods perform differently across applications [War+20]. Moreover, recipients of explanations have varying requirements, so "one explanation does not fit all" [Ary+19]. Nevertheless, the explanation methods investigated in this thesis are among the most cited work, as pointed out above, and can thus be considered widely used, at least in research. They are therefore ideal candidates for our adversarial analyses.

3 Explanation-Aware Attacks

Contents

Chapter Preface.
This chapter is based on the following two publications:

- Maximilian Noppel and Christian Wressnegger. 'SoK: Explainable Machine Learning in Adversarial Environments'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2024
- Maximilian Noppel and Christian Wressnegger. 'A Brief Systematization of Explanation-Aware Attacks'. In: *Proc. of the German Conference on Artificial Intelligence (KI)*. 2024

The first is a "Systematization of Knowledge" (SoK)-paper. The second is an extended abstract summarizing the first work.

For this chapter, we have refreshed and expanded the original work. While we retained the core text and key figures from our SoK paper, we substantially enhanced the presentation through new visualizations and reorganized content to improve clarity and readability.

We have made the following bigger changes and additions:

- **Updated Table of Attacks**. We have updated Tables 3.2 to 3.4 to include recent works published *after* our original SoK paper.
- **Related Work**. After our SoK paper was published, four subsequent works [VSL24; Pac+24; MP25; BB24] on explanation-aware attacks have been published, which we relate to in Section 3.8.

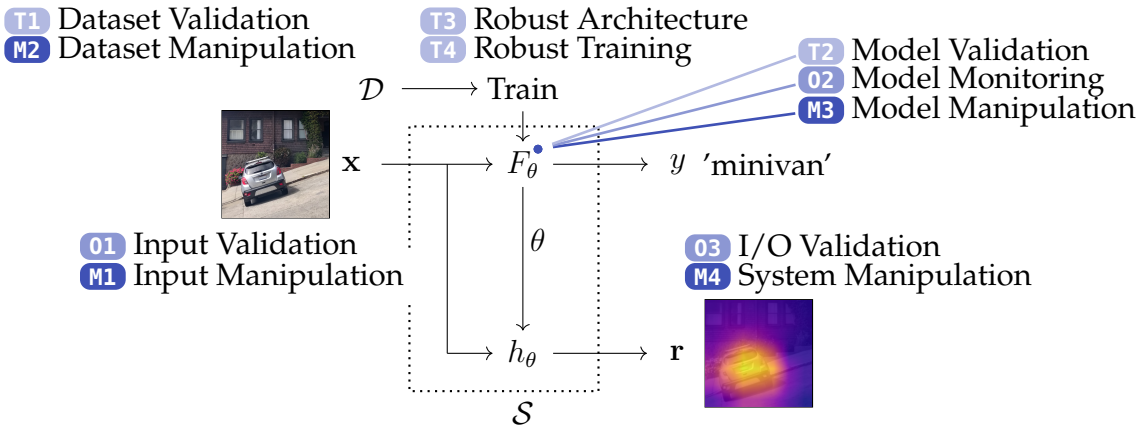


Figure 3.1: Attacks (M) against an XAI-system $\mathcal{S}$ and defense strategies during model training (T) and system operation (O). F_θ predicts class label y for input $\mathbf{x}$ based on model θ that is trained on a training dataset $\mathcal{D}$. h_θ represents a post-hoc explanation method deriving an explanation $\mathbf{r}$ of the input sample.

3.1 Introduction

With the advent of artificial intelligence, and in particular with its wide reach into almost every area of our everyday lives, the need to reliably understand learning-based systems continues unabated [GSS15; Pap+16]. Thus, it is often recommended to apply explanation methods to gain a better view of the system's decision-making processes [Arp+22]. By doing so, the hope is to first identify a model's biases and limitations through the explanations, then understand them, fix them, and eventually verify that the model behaves correctly.

However, explanations themselves are unreliable in adversarial environments. For instance, they can be forged through input manipulations [Dom+19; Sla+21a; Zha+20] and various model manipulations [Sla+20; NPW23], similar to adversarial examples [Sze+14] and neural backdoors [Gu+19; Wan+19a], respectively. Thus, it is questionable whether current post-hoc explanation methods can indeed accomplish this goal in practice.

In this chapter, we lay the groundwork for understanding the limits and potentials of explainable artificial intelligence (XAI) in adversarial environments. We systematize attacks against explanation methods, so-called *explanation-aware attacks*. Figure 3.1 depicts the attack surface of an explainable system. The adversary can perform traditional input-level manipulations (M1) [e.g., Dom+19; Zha+20; KL20], dataset manipulations (M2) [e.g., ZGS21; NW25; Hel24], model-level manipulations (M3) [e.g., NPW23; HJM19; ZGS21], but also system-level manipulations (M4) such as fairwashing the model [e.g., And+20; Aïv+19; Sla+20]. The latter represents a particularly powerful attacker, capable of manipulating any aspect of the XAI-system. Additionally, our systematization covers different attack objectives in terms of the attack's targets and the way the targets should be altered. While for predictions the adversarial goal is to yield either one specific prediction or any wrong prediction, for explanations we similarly see targeted and untargeted attacks. However, we additionally observe so-called semitargeted attacks. In addition, the explanation space usually has fundamentally different properties than the space of labels. To name just one for now, explanation spaces are substantially larger, or even infinite.

For further analysis, we define robustness notions of post-hoc explanation methods, modeling pairs of constraints and restrictions that may be enforced to yield robust explanations. One central finding of our *hierarchy of robustness notions* is that different authors describe different understandings of robustness using the same key phrase. This underlines the necessity of a common understanding of explanation-aware robustness to advance research in this domain. In particular, robustness guarantees for post-hoc explanation methods can be key to effectively defending against adversaries attacking learning-based systems.

Based on this fundamental understanding of attacks and robustness requirements, we proceed to taxonomize existing defenses to counter explanation-aware attacks. In Figure 3.1, we mark locations in the machine learning pipeline where defenses can be applied during training (T1 – T4) and at inference time during operation (O1 – O3).

Compared to defenses against prediction-only[1] attacks [e.g., Mad+18b; RSL18; Wan+19a; BMV18; Sal+19; Cro+22], defenses against explanation-aware attacks are scarce, as also shown in a concurrent survey by Baniecki and Biecek [BB24]. For our systematization, we, thus, explicitly relate to defenses that have been explored for conventional input and model manipulation attacks against a classifier's prediction. In doing so, we motivate the community to step back and unify the knowledge of both fields to find more effective defenses. We pose open research questions whose answers will eventually close the gap between these highly related subfields. Most pressingly: *"What robustness guarantees can we provide for post-hoc explanation methods?"*

We are confident that our systematization can act as a signpost toward effective defenses against explanation-aware attacks. Most importantly, however, we build a foundation and provide guidelines for creating and proving robust explanation methods.

In this chapter, we make the following contributions:

- **Systematization of Attacks against Explanations.** We systematize attacks against post-hoc explanations along the adversary's capabilities, constraints, and objectives. In doing so, we highlight crucial differences and similarities of threat models and attack objectives specific to explanation-aware attacks.
- **Explanation-Aware Robustness Notions.** We formalize robustness against explanation-aware attacks and form a hierarchy of robustness notions that enables us to compare, link, and unify efforts in the community. Moreover, we find that robustness notions might be conflicting for adversaries with diverging objectives.
- **Taxonomy of Defenses.** We taxonomize existing defenses based on the usual machine learning pipeline and relate to defenses for attacks against the predictions only. In each of the identified categories, we argue whether and why, or why not, the presented defenses are expected to transfer to explanation-aware attacks.
- **Future Research Directions.** We point out specific directions for future research on explanation-aware robustness in order to help advance the field.

[1] We denote attacks that only target the prediction but ignore the explanations as prediction-only (*PO*) or vanilla attacks.

3.2 Threat Models

We begin to systematize the adversary's capabilities and their associated constraints. A schematic depiction of the considered attack vectors is provided in Figure 3.1. More formally, an adversary $\mathcal{A}$ attacking local post-hoc explanation methods faces an optimization problem, abstractly defined as:

$$\min_{\mathcal{A}(\mathfrak{K})} obj \text{ s.t. } \omega_1(\ldots) \wedge \ldots \wedge \omega_N(\ldots)$$

The adversary operates on a certain *knowledge* $\mathfrak{K}$ to manipulate samples $\mathbf{x}_i$ (M1), the data $\mathcal{D}$ (M2), the model θ (M3), and/or the entire XAI-system $\mathcal{S}$ (M4). They may know (a subset of) all original samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ or any information about the model and the original system $\mathcal{S}$ (e.g., the model's architecture $\theta_{\mathbb{A}}$, its weights θ_w, or the used explanation method h and its parameters). However, depending on the threat model, the available knowledge and the output of the adversary varies. In the context of input manipulations, for instance, we denote adversaries that know the model under attack as white-box adversaries, while adversaries without this knowledge are termed black-box adversaries.

Besides their knowledge, also different constraints ω_k restrict the adversaries and define their capabilities. For instance, they may perturb samples within a certain norm ball only, must not change the model's architecture $\theta_{\mathbb{A}}$, or need to apply the same universal manipulation on every sample.

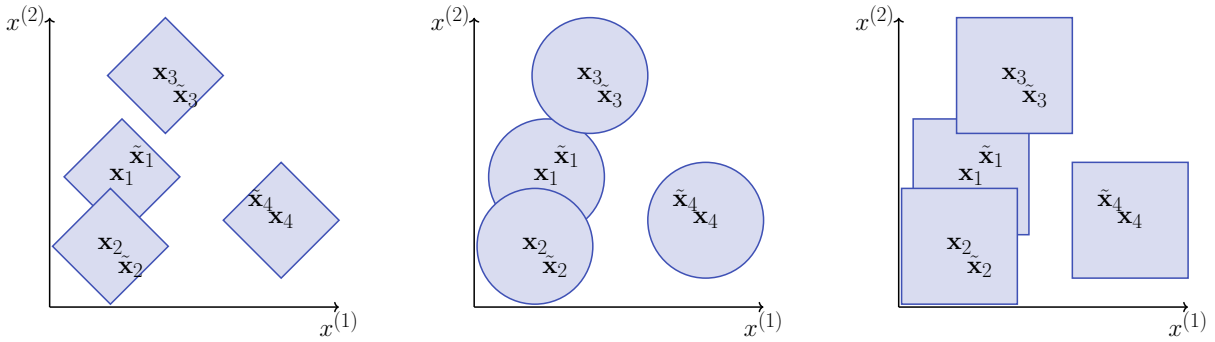
In the following, we detail threat models for manipulating samples (Subsection 3.2.1), the dataset (Subsection 3.2.2), the model (Subsection 3.2.3), and the entire XAI-system (Subsection 3.2.4). Each has different constraints and allows for varying degrees of knowledge $\mathfrak{K}$, thus demanding distinct attack capabilities and forcing the adversary to produce different outputs and inputs, $\star \leftarrow \mathcal{A}(\triangle; \mathfrak{K})$, where $\triangle$ denotes the abstract inputs and $\star$ the abstract outputs, respectively. Note that the *attack objective* obj itself is detailed in Section 3.3, independently of the threat models. Additionally, Tables 3.2 to 3.4 list works according to these threat models and other dimensions introduced in this chapter.

3.2.1 Input Manipulation M1

In the context of explanation-aware attacks the ability to manipulate an input $\mathbf{x}$ is most frequently investigated [Abd+22; BK21; Dom+19; Iva+22; Sin+21; Zha+20; Aja+22]. An input $\mathbf{x}$ is either created by the adversary and then submitted to the system, or intercepted and manipulated before it reaches the model. In either case, the adversary is able to manipulate a sample $\mathbf{x}$ within certain constraints, yielding a malicious sample $\tilde{\mathbf{x}} \leftarrow \mathcal{A}(\mathbf{x}, \ldots; \mathfrak{K})$ that fulfills the attack objective.

Note that this does not necessarily mean that the adversary knows the sample $\mathbf{x}$. To illustrate this argument, think of an adversary who performs a hardware supply-chain attack, such that sensors add particular patterns, when desired and triggered through other channels.

Below we start discussing different applicable constraints.



(a) Imperceptibility under L_1 **(b)** Imperceptibility under L_2 **(c)** Imperceptibility under L_∞

Figure 3.2: Illustration of the unit balls for different L_p norms in a two-dimensional input space. The L_1 norm **(a)** yields a diamond shape, the L_2 norm **(b)** yields a circle, and the L_∞ norm **(c)** yields a square. Manipulated samples $\tilde{\mathbf{x}}_i$ must lie within the respective norm ball of radius ϵ around their benign counterpart $\hat{\mathbf{x}}_i$.

Imperceptibility. The adversary aims for imperceptible perturbations to a specific input $\mathbf{x}$, yielding $\tilde{\mathbf{x}}$. Hence, the difference between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ should be minimal Both inputs need to be as similar as possible:

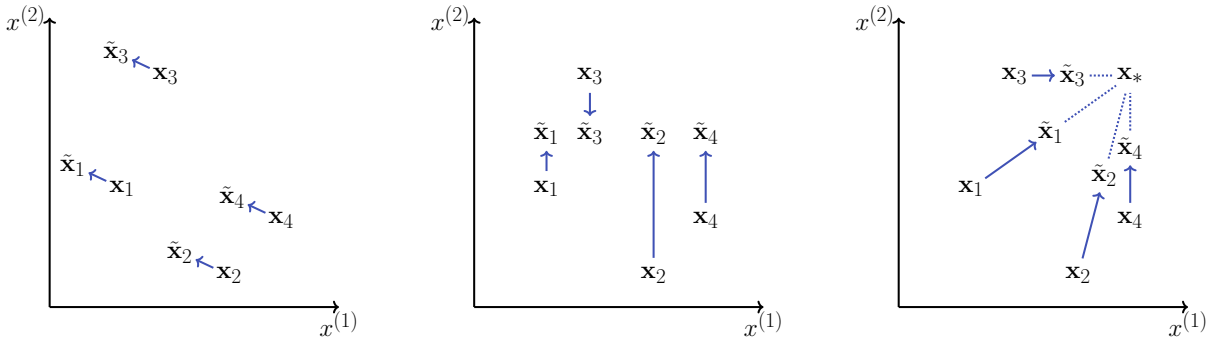
$$\omega_k(\mathbf{x}, \tilde{\mathbf{x}}) := d_{\mathcal{X}}(\mathbf{x}, \tilde{\mathbf{x}}) \leq \epsilon \,,$$

for a limit $\epsilon \in \mathbb{R}_+$. A common choice for $d_{\mathcal{X}}$ frequently considered in literature [e.g., GSS15; CW17a; Sze+14; Dom+19] are the L_p norms, $\|\mathbf{x} - \tilde{\mathbf{x}}\|_{p \in \{0,1,2,\infty\}}$. We visualize the allowed ball norm area according to three different L_p norms with blue shapes in Figure 3.2. For visual and audible inputs, though, there exist more specialized ways to measure perceptibility to a human analyst, e.g., the "SSIM" for images [Wan+04]. We will make use of these more specific types of distance measurements in the remainder of this thesis.

Universality. Additionally, an adversary may manipulate a single sample or consider manipulations that must universally apply to multiple inputs simultaneously. Universal input manipulations are restricted to operations that perturb the sample independently of its specific content. Their evasive properties, defined by the overall objective, must apply to all inputs $\mathbf{x}_i$. Examples include adding a constant additive offset [Moo+17a], a patch replacement [SPP19; Bro+17], or a blending with a fixed reference sample $\mathbf{x}_*$, for instance a picture of "Hello Kitty" as a trigger pattern [Che+17]. All three cases can be formalized by an adversarially-chosen and fixed vector $\mathbf{v} \in \mathcal{X}$, a fixed mask matrix $\mathbf{m} \in \{0,1\}^d$, the reference image $\mathbf{x}_* \in \mathcal{X}$ and a blending factor $\lambda \in [0,1]$:

- a constant additive offset: $\tilde{\mathbf{x}} := \mathbf{x} + \mathbf{v}$
- a patch replacement: $\tilde{\mathbf{x}} := (1 - \mathbf{m}) \odot \mathbf{x} + \mathbf{m} \odot \mathbf{v}$
- a blending with a fixed sample: $\tilde{\mathbf{x}} := (1 - \lambda) \cdot \mathbf{x} + \lambda \cdot \mathbf{x}_*$

Here, $\odot$ denotes the element-wise multiplication. For these specific universal input manipulations, the above equations must be satisfied for all inputs $\tilde{\mathbf{x}}$ and for fixed $\mathbf{v}$, $\mathbf{m}$, λ, and $\mathbf{x}_*$. In Figure 3.3, we depict the three constraints from above as examples.



(a) Addition **(b)** Patching **(c)** Blending

Figure 3.3: Common constraints on how adversaries can manipulate inputs universally. For simplicity, we display a two-dimensional input space. In **(a)**, a constant additive offset $\mathbf{v}$ is applied to each input, moving all points in the same direction. In **(b)**, a specific dimension is set to a fixed value, constraining all manipulated inputs to a common $x^{(2)}$ coordinate. In **(c)**, inputs are blended towards a common reference sample $\mathbf{x}_*$, with manipulated versions placed along the path.

Perturbation Subsets. Occasionally, the adversary only controls a small, specific portion of each input, enforcing a masked input manipulation [Xia+21; Bro+17; LF20]. More generally, the constraint limits the space of the perturbation $\delta_\mathbf{x}$. The space of $\delta_\mathbf{x}$, which by default is $\mathbb{R}^d$, might then be either a fixed parameter of the threat model or chosen by the adversary.

The most basic instantiation of perturbation subsets is a mask vector, where some dimensions of the sample cannot be perturbed. This instantiation can be modeled with a binary mask vector $\mathbf{m} \in \{0,1\}^d$:

$$\omega_k(\mathbf{x}, \tilde{\mathbf{x}}) := ((\mathbf{x} \neq \tilde{\mathbf{x}}) \vee \mathbf{m}) = \mathbf{m} \,,$$

where $\vee$ and $\neq$ are interpreted as element-wise operations. Various extensions exist; for instance, the mask might be required to be a square [Xia+21; LF20] or small in size, $\|\mathbf{m}\|_0 \leq \epsilon_0$ [Bro+17; Vie+19]. If $\mathbf{m}$ is chosen by the adversary, this links to imperceptibility under L_0 over a set of samples $\mathbf{x}_i$ and their manipulated counterparts $\tilde{\mathbf{x}}_i$:

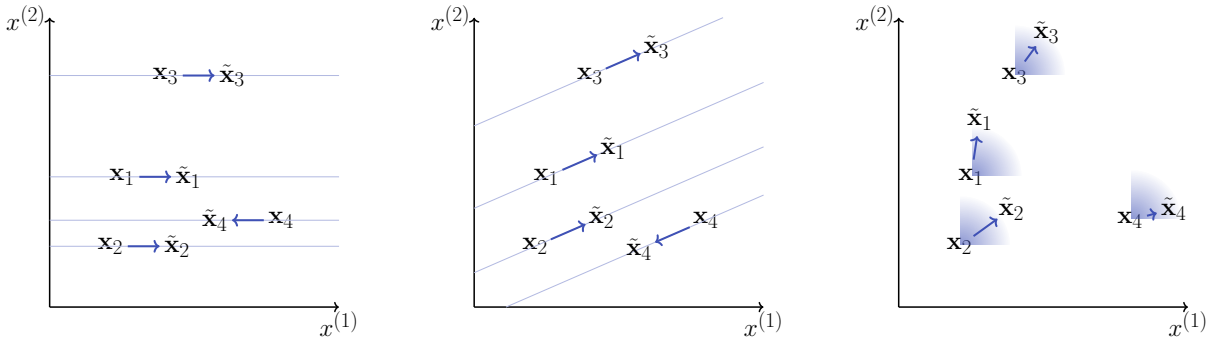
$$\Big\| \bigvee_i (\mathbf{x}_i \neq \tilde{\mathbf{x}}_i) \Big\|_0 \leq \epsilon_0 \,,$$

where $\vee$ and $\neq$ are again interpreted element-wise. Such required properties of the $\delta_\mathbf{x}$-space are necessary when the $\delta_\mathbf{x}$-space is selected by the adversary.

Furthermore, the $\delta_\mathbf{x}$-space of allowed perturbations does not necessarily have to correspond to the input dimensions. Instead, it can also be a span defined over a set of vectors $\delta_\mathbf{x} \in \operatorname{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_{\leq d}\}$ with $\mathbf{v}_i \in \mathbb{R}^d$.

Lastly, the perturbation can have constraints based on the attack's implementation. For instance, the adversary might only be able to overwrite bits from 0 to 1 but not the other way around, or be constrained such that they can only increase values. In this latter case, the perturbation is necessarily positive, $\delta_\mathbf{x} \in \mathbb{R}^d_{+,0}$.

We visualize these three examples of perturbation subset constraints in Figure 3.4 below.



(a) Masking **(b)** Spans **(c)** Positivity

Figure 3.4: Common constraints on the $\delta_{\mathbf{x}}$-space, i.e., the allowed perturbations. For simplicity, we again display a two-dimensional input space. In **(a)**, we show a scenario where only the $x^{(1)}$ dimension can be perturbed, meaning inputs can only be moved left or right. **(b)** visualizes the case where the perturbations are forced to be elements of a span. In **(c)**, the perturbation vectors are required to be positive.

3.2.2 Dataset Manipulation M2

Influence on the model can be achieved *indirectly* by poisoning the training data [BNL12]. While data poisoning is a popular scenario for attacks against predictions [Sha+18; Jag+18; Gei+21; Jag+21], its relevance for explanation-aware attacks remains an open research question and is largely unexplored. It is not immediately apparent how attacks against explanations would be instantiated by perturbing samples and labels only. One first work [ZGS21] in this direction encircles target samples with poisoned samples to alter their explanations. This process, however, is rather costly and affects only a few samples.

Excursus 2.
In a recent Master's thesis [Hel24], we have investigated whether explanation-aware backdoors (cf. Chapter 4) can be injected through data poisoning alone. Our attack method requires poisoning each training batch individually within a norm ball. Concretely, we have used a technique called *Gradient Alignment* [Gei+21], where individual batches have been manipulated on-the-fly to align the gradient of the benign cross-entropy loss used by the victim with the explanation-aware loss function desired by the attacker. This process involves substantial computational costs, as each batch is optimized for around 100 steps according to two loss functions, of which one is explanation-aware, i.e., even more computationally expensive than the benign loss function. The resulting batches are then bundled into a poisoned dataset and provided to the victim, i.e., the clean-label poisoning rate is 100 %. The victim then uses this dataset to train their model from scratch.
We have found that with our attack the explanations of the victim model can indeed be manipulated successfully in some settings. However, our method is neither efficient nor stable. Even minor randomness in the training process substantially hinders the attack's success. For instance, using a different graphics card model or different batch orders prevents the attack. Also, random data augmentation techniques hinder the attack's effectiveness substantially.

Augmented Training Data. Several studies augment the training process and its training data with ground-truth explanations to guide the model's decision-making [RHD17; CSW22; Rie+20; WJL20; Du+19; Zho+22]. This approach is commonly referred to as *explanation-guided learning*. An additional loss term considers the distance to a ground-truth explanation per sample as a form of regularization. This supplementary regularization aims to suppress spurious correlations by incorporating domain knowledge into the training process. For example, Chefer, Schwartz, and Wolf [CSW22] encourage the model to focus on the object rather than the background. The similarity between model guidance and explanation-aware manipulations raises questions about whether the model genuinely learns to avoid spurious correlations or if the explanations are inadvertently forged.

Observation 1.
Guiding the learning process through ground-truth explanations shares properties with explanation-aware model manipulations. These similarities (a) pose the risk of guidance inadvertently becoming an unwanted manipulation and (b) may provide insights into how to harness positive effects of explanation-aware manipulations. Furthermore, the dataset augmented with explanation can be considered a new attack vector to execute explanation-aware attacks through data poisoning.

3.2.3 Model Manipulation M3

An adversary manipulating the model directly, $\tilde{\theta} \leftarrow \mathcal{A}(\mathfrak{K})$, may change the model's weights and biases (θ_w) and may even alter the underlying architecture ($\theta_{\mathbb{A}}$). The adversary can modify the model's weights to create a weight-manipulated model, defined as $\tilde{\theta} := (\theta_w + \delta_w, \theta_{\mathbb{A}})$. For full model manipulation, the adversary can create an entirely independent model from scratch, $\tilde{\theta} \in \Theta$, without any restrictions on the model's architecture.

Despite the capability to fully manipulate the model, the adversary often is constrained regarding the side effects their changes may have. A manipulated model needs to either achieve a similar validation or testing accuracy or make the same mistakes as the original model. This is to bypass functionality tests before deployment. Based on the notion of ϵ-accuracy and ϵ-agreement defined below, we formulate *model parameter manipulation* as:

$$\min_{\delta_w \leftarrow \mathcal{A}(\triangle;\mathfrak{K})} obj \text{ s.t. } \omega_k := \begin{cases} F_{(\theta_w+\delta_w,\theta_{\mathbb{A}})} \text{ is } \epsilon\text{-accurate} \\ F_\theta,\ F_{(\theta_w+\delta_w,\theta_{\mathbb{A}})} \text{ are } \epsilon\text{-agreeing} \end{cases},$$

as well as *model manipulation* as:

$$\min_{\tilde{\theta} \leftarrow \mathcal{A}(\triangle;\mathfrak{K})} obj \text{ s.t. } \omega_k := \begin{cases} F_{\tilde{\theta}} \text{ is } \epsilon\text{-accurate} \\ F_\theta \text{ and } F_{\tilde{\theta}} \text{ are } \epsilon\text{-agreeing} \end{cases}.$$

Note that, in contrast to the attack's outreach (see later in Subsection 3.3.1), these constraints apply to the overall model operation rather than single inputs. Moreover, we denote $F_{\tilde{\theta}}$ in the following sections for simplicity, even if only the learned parameters θ_w are perturbed while the architecture remains unchanged.

ϵ-Accuracy. A model θ should achieve a similar accuracy (on testing data) but may make different classification errors compared to the original model. Formally, we measure the accuracy as

$$\mathrm{acc}(F_\theta) = \mathbb{E}_{(\mathbf{x},\hat{y})\sim\mathcal{D}_{test}}[F_\theta(\mathbf{x}) = \hat{y}] \ .$$

In comparison to a benignly trained reference model, we define a manipulated model as ϵ-accurate as follows:

Definition 3.2.1 (ϵ-accurate Model) *We denote the (manipulated) model θ_1 as* ϵ-accurate regarding a (benignly trained) reference model θ_2 *iff its test accuracy is not significantly lower than the accuracy of the reference model θ_2*

$$\mathrm{acc}(F_{\theta_1}) - \mathrm{acc}(F_{\theta_2}) \leq \epsilon \ ,$$

for a limit $\epsilon \in \mathbb{R}_+$.

ϵ-Agreement. If a manipulated model's predictions match the decisions of the original model, we denote both as having a high fidelity. In line with related work [Aïv+21; Aïv+19; CS95], we define the fidelity (and the equivalent "agreement rate") of two models as:

Definition 3.2.2 (Fidelity / Agreement Rate) *We denote the* fidelity *and* agreement rate *between two classifiers θ_1 and θ_2 based on a distribution or set of samples P as*

$$\mathit{fid}_P(\theta_1, \theta_2) := \mathbb{E}_{\mathbf{x}\sim P}[F_{\theta_1}(\mathbf{x}) = F_{\theta_2}(\mathbf{x})] \ .$$

We refer to the above as *local fidelity around* $\mathbf{x}$ if P is the neighborhood $\mathcal{N}_\mathbf{x}$ of a sample $\mathbf{x}$, and as *global fidelity* if P is the complete feature space $\mathcal{X}$. The agreement rate as defined above is the exact inverse of the occasionally used "disagreement rate" [Lak+19].

Definition 3.2.3 (ϵ-agreeing Models) *We denote two classifiers θ_1, θ_2 as* ϵ-globally agreeing *if*

$$1 - \mathit{fid}_\mathcal{X}(\theta_1, \theta_2) \leq \epsilon \ .$$

Intuitively, the empirical measurement of the probability that both models arrive at the same (potentially wrong) prediction should be close to 1.

Other Constraints. Next to the above descriptions, the adversary may be subject to further application-specific constraints. For example, manipulations might be limited to a few bit-flips [RHF20] or weight perturbations could be required to remain below a specific threshold, $\|\delta_w\|_\infty \leq \epsilon$ [Gar+20]. Additionally, the attack may only take effect if the model is compressed [Tia+22] quantized [Hon+21], or if it is executed on a specific machine learning backend, such as Intel MKL, Nvidia CUDA, or Apple Accelerate, as already shown for adversarial inputs [Möl+25]. Lastly, our descriptions so far focus on the vanilla supervised learning scenario. In more specialized learning setups, such as federated learning [e.g., Bar+23; Xia+23], the adversary must obey additional constraints and work with limited capabilities. Similar attack scenarios and constraints can readily be envisioned for explanation-aware attacks as well.

3.2.4 System Manipulation M4

System manipulations represent one of the most powerful adversarial capabilities considered in this work[2]. Also, while input and model manipulations are common in conventional attacks against machine learning, the capability to manipulate the whole system is most significant in explainable machine learning [MT20]. As an example, a potential adversarial goal is to hide the unfair or biased reasoning of the system [And+20; Aïv+19; Sla+20; MT20]. Whether the bias is introduced on purpose is secondary. For hiding biases, the adversary may, for instance, learn a set of unbiased surrogate models for the biased black box model and report explanations for the surrogate that yields the most similar prediction to the biased model [Aïv+19]. With the explanation being generated on the unbiased surrogate model, the adversary feigns a fair decision.

We denote the functionality of generating predictions *and* explanations as XAI-system $\mathcal{S}$:

$$\mathcal{S} : \mathcal{X} \times \Theta \to [C] \times \mathcal{E} \ ,$$

as depicted in Figure 3.1 as a dashed rectangle. Further, we write $\mathcal{S}_f$ or $\mathcal{S}_F$ if we are concerned about the prediction output and $\mathcal{S}_h$ for the explanation output respectively. Hence, the applicable constraint is along the lines of the following

$$\min_{\tilde{\mathcal{S}} \leftarrow \mathcal{A}(\mathfrak{K})} obj \text{ s.t. } \omega_k := \begin{cases} \tilde{\mathcal{S}}_F \text{ is } \epsilon\text{-accurate} \\ \mathcal{S}_F \text{ and } \tilde{\mathcal{S}}_F \text{ are } \epsilon\text{-agreeing} \ . \end{cases}$$

3.3 Attack Objectives

An adversary may target two fundamental aspects of a learning-based system: (1) the predictions [Pap+18] or (2) the explanations. Each target can be pursued in isolation or in combinations, giving rise to different attack classes. Prediction-only attacks do not consider explanations at all, while explanation-aware attacks do.

The adversary's objective may be to either alter, preserve, or ignore a specific target. In the following, we briefly discuss each option before we elaborate on the different combinations and on the different ways to alter each target:

1. **Prediction Preservation.** A common objective in explanation-aware attacks is to preserve the prediction [Dom+19; Iva+22; Sin+21; Wan+20b]. How well predictions are preserved can, for example, be measured by the fidelity/agreement or the accuracy, as introduced above. However, we argue that in most scenarios the accuracy (or F1-score) is the more reasonable measurement. Optimizing fidelity may be reasonable in scenarios where the adversary replaces an already deployed model and the behavior of both models should

[2] Note that even system manipulators are not strictly stronger than data poisoners. Instead they are incomparable. System manipulators usually can change the model or the inference process and the generation of the explanation. They can even programmatically overwrite the outputs of the system completely. However, they cannot change the training data at the opponent's place. That is also the reason why the data $\mathcal{D}$ and the training process are positioned outside of the system in Figure 3.1. Nevertheless, we think system manipulators are rather *powerful* in comparison to data poisoners.

remain similar. Both metrics are usually optimized with the cross-entropy loss function $\mathcal{L}_{CE}\big(\tilde{\mathcal{S}}_F(\tilde{\mathbf{x}}), \cdot\big)$. Hence, we denote $\mathcal{L}_{CE}$ in the following loss functions to represent this *prediction-preserving* objective.

2. **Prediction Alteration.** Altering the prediction alone is extensively discussed in literature, covering input manipulations [Sze+14; GSS15; Moo+17a] and model manipulations [Gu+19; Wan+19a; Liu+18]. In the context of this thesis, however, we cannot consider all details of such attacks. However, we still briefly introduce the two scopes of prediction alteration:

   - *Prediction-Targeted Alteration (◉).* This first scope is parameterized with a target class y_t. Every attacked sample should be classified as this class, e.g., "`goodware`".
   - *Prediction-Untargeted Alteration (🎯).* Here, the sample should be misclassified to *any* class other than its ground-truth label.

   For the remainder of this chapter we generalize and write $\mathrm{alter}_F(\tilde{\mathcal{S}}_F(\tilde{\mathbf{x}}), \cdot)$ for any *altering* objective on the prediction.

3. **Explanation Preservation.** Here, the explanation should be as similar to a *clean* explanation as possible. For instance, to disguise an ongoing input manipulation [Zha+20] or model manipulation [NPW23]. A clean explanation could be one of the following:

   - *Ground-Truth Explanations.* In some settings ground-truth explanations are given. For instance, for synthetic datasets or when domain experts annotate the relevant parts of each sample. However, we argue that preserving ground-truth explanations can even reveal a manipulation. Such explanations do not necessarily match with explanations of a benignly trained model, and, thus, might be detectable.
   - *Benign Explanations.* Alternatively, the adversary should simulate a benign training process and use the resulting explanations of clean samples as the inconspicuous explanations. These explanations match what the victim expects.

   Again, for this chapter we generalize and denote this *explanation-preserving* objective by $d_{\mathcal{E}}(\tilde{\mathcal{S}}_h(\tilde{\mathbf{x}}), \cdot)$ in the displayed loss functions.

   > **Remark 2.**
   > Note how this is the same differentiation between fidelity and accuracy as we have already discussed for predictions, just for explanations. However, for explanations we argue in favor of fidelity instead of accuracy.

4. **Explanation Alteration.** Finally, the adversary may choose to alter the explanation, for instance, to distract the analyst through input manipulations [Dom+19] or to make a model appear fair when it is not [Aïv+19]. We detail the possible ways of altering explanations in Subsection 3.3.2. Until then, we denote this *altering* objective by $\mathrm{alter}_h(\tilde{\mathcal{S}}_h(\tilde{\mathbf{x}}), \cdot)$ in the displayed loss functions.

Based on the presented criteria we categorize explanation-aware attacks in Subsection 3.3.1. Afterwards, we elaborate on the "Explanation Alteration" objective in Subsection 3.3.2.

Table 3.1: Overview of the adversarial goals. Prediction-only (PO) adversaries ignore explanations. In this thesis, we focus on explanation-aware attacks, which we categorize into explanation-only (EO), explanation-preserving (EP), prediction-preserving (PP), and dual (D) attacks. The symbols ● and ◉ indicate whether the attack is prediction-targeted or prediction-untargeted, while the symbols ⁘, ⁙, and ⁛ indicate whether the attack is explanation-targeted, explanation-untargeted, or explanation-semitargeted, respectively.

Name	Icon	Prediction		Explanation	
		Type	Scope	Type	Scope
Prediction-Only	$PO_{●}$	alter	(targeted)	ignore	–
	$PO_{◉}$	alter	(untargeted)	ignore	–
Explanation-Only	$EO^{⁘}$	ignore	–	alter	(targeted)
	$EO^{⁙}$	ignore	–	alter	(untargeted)
	$EO^{⁛}$	ignore	–	alter	(semitargeted)
Explanation-Preserving	$EP_{●}$	alter	(targeted)	preserve	–
	$EP_{◉}$	alter	(untargeted)	preserve	–
Prediction-Preserving	$PP^{⁘}$	preserve	–	alter	(targeted)
	$PP^{⁙}$	preserve	–	alter	(untargeted)
	$PP^{⁛}$	preserve	–	alter	(semitargeted)
Dual	$D^{⁘}_{●}$	alter	(targeted)	alter	(targeted)
	$D^{⁙}_{●}$	alter	(targeted)	alter	(untargeted)
	$D^{⁛}_{●}$	alter	(targeted)	alter	(semitargeted)
	$D^{⁘}_{◉}$	alter	(untargeted)	alter	(targeted)
	$D^{⁙}_{◉}$	alter	(untargeted)	alter	(untargeted)
	$D^{⁛}_{◉}$	alter	(untargeted)	alter	(semitargeted)

3.3.1 Types of Explanation-Aware Attacks

The combination of one objective (preservation or alteration) per target (prediction and explanation) yields three primary subclasses of explanation-aware attacks. Note that the fourth combination, where the adversary preserves both the prediction and the explanation, is not an attack and is therefore not discussed further [SMV22]. In addition, we consider explanation-only attacks, where the predictions are ignored and only the explanations are attacked. Table 3.1 provides an overview of all attack types using the symbols from above and from the scopes as introduced in Subsection 3.3.2. Corresponding related work on input-manipulation attacks, model-manipulation attacks, and on fairwashing attacks is listed in Tables 3.2 to 3.4, respectively. Further Table A.1 contains an overview on the student theses supervised by the author according to the same dimensions.

Explanation-Only Attack (EO). Explanation-only attacks are the direct counterpart to prediction-only attacks. The adversary aims to manipulate the explanation but ignores the prediction, resulting in the following loss function:

$$\min_{\tilde{\mathbf{x}}, \tilde{\mathcal{S}} \leftarrow \mathcal{A}(\mathfrak{K})} d_{\mathcal{E}}\left(\mathcal{S}_h(\mathbf{x}), \tilde{\mathcal{S}}_h(\tilde{\mathbf{x}})\right) .$$

Explanation-Preserving Attack (*EP*). The adversary attempts to keep the explanation unchanged while attacking the prediction. Generally speaking they aims to solve the following optimization problem:

$$\min_{\tilde{\mathbf{x}}, \tilde{\mathcal{S}} \leftarrow \mathcal{A}(\mathfrak{K})} \text{alter}_F\big(\tilde{\mathcal{S}}_F(\tilde{\mathbf{x}}), \cdot\big) + d_{\mathcal{E}}\big(\tilde{\mathcal{S}}_h(\tilde{\mathbf{x}}), \cdot\big) \ .$$

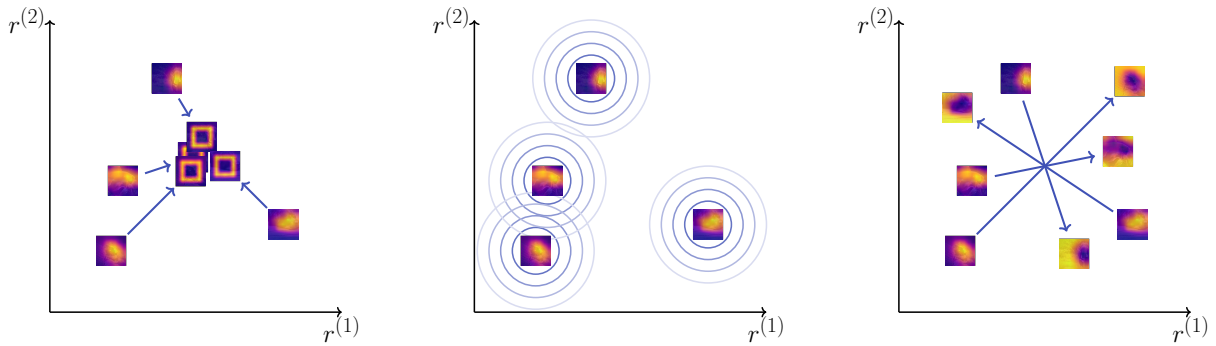
Prediction-Preserving Attack (*PP*). The adversary attempts to keep the prediction as accurate (or agreeing) as possible with the benign prediction. Simultaneously, they attacks the explanation method. This leads to the following optimization problem:

$$\min_{\tilde{\mathbf{x}}, \tilde{\mathcal{S}} \leftarrow \mathcal{A}(\mathfrak{K})} \mathcal{L}_{CE}\big(\tilde{\mathcal{S}}_F(\tilde{\mathbf{x}}), \cdot\big) + \text{alter}_h\big(\tilde{\mathcal{S}}_h(\tilde{\mathbf{x}}), \cdot\big) \ .$$

Observation 2.
Perfect fidelity can be easily achieved by manipulating the entire XAI-system through using the original model for the prediction output on in-distribution input. Slack et al. [Sla+21b], for instance, apply a scaffolding technique where the original model is used for real input, while other models are used for input originating from the LIME explanation method. Their approach demonstrates how system-level manipulation can maintain high fidelity while altering explanations for specific inputs.

Dual Attack (*D*). The adversary targets both the prediction and the explanation. In our systematization, this attack corresponds to the following optimization problem:

$$\min_{\tilde{\mathbf{x}}, \tilde{\mathcal{S}} \leftarrow \mathcal{A}(\mathfrak{K})} \text{alter}_F\big(\tilde{\mathcal{S}}_F(\tilde{\mathbf{x}}), \cdot\big) + \text{alter}_h\big(\tilde{\mathcal{S}}_h(\tilde{\mathbf{x}}), \cdot\big) \ .$$



(a) Targeted (⁝) **(b)** Untargeted (⁝) **(c)** Semitargeted (⁝)

Figure 3.5: Depiction of the three scopes when altering explanations. Subfigure **(a)** illustrates the targeted scope: all samples have the same fixed target explanation. In the depiction, however, not all attacks reach the target explanation reliably. Subfigure **(b)** shows the untargeted scenario where the manipulated explanation should diverge as much as possible from the clean explanation, indicated by the blue circles. Finally, **(c)** demonstrates the semitargeted objective, where sample-specific target explanations are defined by a (learnable) function that operates in the explanation space.

3.3.2 Scope of Explanation Alteration

We now specify the different ways adversaries can alter explanations. Following prior research on PO attacks [Pap+18], we adopt the terms *untargeted* and *targeted* and adapt them to explanation-aware attacks [Tan+22; Aja+22]. Additionally, we introduce *semitargeted* attacks, a category unique to explanation-aware attacks. To distinguish whether the attack type is with respect to the predictions or the explanations, we prepend the respective target before the scope designation where necessary, e.g., we write explanation-targeted.

3.3.2.1 Explanation-Targeted Attacks (⁘)

Here, the objective is to minimize the distance between the manipulated explanation and a fixed target explanation $\mathbf{r}^t$ [e.g., Dom+19; NPW23; NW25]. A single optimal solution to the optimization problem exists, namely the target explanation. We formalize the *targeted attack objective* for altering the explanation as:

$$\text{alter}_h^{tar} := d_{\mathcal{E}}(\tilde{\mathcal{S}}_h(\tilde{\mathbf{x}}), \mathbf{r}^t) \ .$$

Figure 3.5a, below, depicts the targeted attack setting for the square target explanation, as used in all following chapters in this thesis. In the following, we denote explanation-targeted attacks with the ⁘-symbol or the symbol of the concrete target explanation, e.g., the □-symbol for the square target explanation.

> **Observation 3.**
> The performance of an explanation-targeted attack can be evaluated by using the benign explanation of a random "in-distribution" sample as the target explanation [Dom+19; RH20]. This way, the target explanation is guaranteed to be plausible.

3.3.2.2 Explanation-Untargeted Attacks (⁙)

In this setting, the objective is to yield an explanation that is maximally different from the benign explanation[3]. In contrast to targeted attacks the set of optimal solutions depends on the benign explanation and potentially contains more than one optimal explanation. Whether this is the case depends on the used metric and setting. We formalize the *untargeted attack objective* for altering explanations as:

$$\text{alter}_h^{untar} := \frac{1}{d_{\mathcal{E}}(\tilde{\mathcal{S}}_h(\tilde{\mathbf{x}}), \mathcal{S}_h(\mathbf{x}))} \ .$$

Note that every sample yields a different forged explanation, hence, the attack resembles a n:n relation from samples to explanations. Figure 3.5b depicts the untargeted attack scenario. In the remainder of this thesis, we indicate untargeted attacks with the ⁙-symbol.

[3] For simplicity, we also consider maximizing the distance to a simple function of the benign explanations here. For instance, Li and Zainon [LZ25] binarize the benign explanations before their optimization.

Example: Maximizing the Distance Metric. A straightforward approach to run untargeted attacks is to maximize the distance to the clean explanation, instead of minimizing it. We perform such an attack at inference time in Chapter 6.

Example: Top-k Fooling. Minimizing the top-k overlap between the clean explanation and the manipulated one can be considered an untargeted attack [HJM19; LSF19]. The originally k-most relevant features should receive as little relevance as possible in the manipulated explanation. However, the exact ranking of the other features is not crucial and, thus, multiple explanations can meet the objective.

3.3.2.3 Explanation-Semitargeted Attacks (⁘)

Finally, an attack can be neither untargeted nor targeted. This is the case when the exact target explanation depends on the input, e.g., inverting the clean explanation [NPW23], swapping the explanations of two classes [HJM19], or suppressing the relevance at a specific location but keeping the remaining explanation intact [SPP19]. For attacking a single sample there is no difference to a targeted attack. However, for multiple samples explanation-semitargeted attacks may implement any learnable function in $\mathcal{E}$. We generalize semitargeted attacks by a function $\mu : \mathcal{E} \times \mathcal{X} \to \mathcal{E}$ that is applied to the clean explanation yielding a sample-specific target explanation:

$$\text{alter}_h^{semi} := d_{\mathcal{E}}(\tilde{\mathcal{S}}_h(\tilde{\mathbf{x}}), \mu(\mathcal{S}_h(\mathbf{x}), \mathbf{x})) \; .$$

We indicate explanation-semitargeted attacks with the ⁘-symbol. To emphasize the importance of this setting for explanation-aware attacks, we discuss two examples.

Example: Inverting Explanations. A model can be manipulated to yield inverted explanations, whenever a backdooring trigger is present on the sample (cf. Chapter 4). That is, regions with high relevance in the benign explanation receive low relevance in the attacked explanation and vice versa, while features with relevance scores in the middle should stay unchanged. A depiction of the example is given in Figure 3.5c.

Example: Location Fooling. For input manipulations, the adversary may be allowed to change a small region of the input only [Bro+17]. Without further provisions this region contributes highly to the prediction and an explanation method would highlight the manipulated region, revealing the attack. To go unnoticed, the manipulated area can be penalized for high absolute relevance values when generating the adversarial input [SPP19; LZ25]. Crucially, the remaining features should keep their original inconspicuous relevances.

> **Remark 3.**
> Note how semitargeted attacks can easily be extended toward producing a set of optimal solutions: $\mu : \mathcal{E} \times \mathcal{X} \to \mathcal{E}^n$. In such cases the optimization process is more complex and depends heavily on the concrete application scenario.

Remark 4.
Note that the categorization of adversarial goals only applies when attacking multiple samples. For instance, in a backdooring scenario where the target explanation appears in the presence of a trigger (cf. Chapter 4).
When attacking a single sample, we are more interested into whether the set of optimal solutions has cardinality 1 or more. Explanation-targeted attacks always yield exactly one optimal solution (cardinality = 1). Semitargeted attacks, however, do not necessarily have a unique solution. The cardinality and structure of the optimal solution set may require deviations from our generalized loss functions.

Table 3.2: Overview of explanation-aware model-manipulation and dataset manipulation attacks categorized by threat model (cf. Section 3.2), type of attack (cf. Subsection 3.3.1), and attack scope (cf. Subsection 3.3.2). We use shaded dots (●) to indicate universal input manipulations. Additionally, we specify whether the corresponding paper attacks white box (□) or black box (■) post-hoc explanation methods, counterfactuals (CF) or self-explainable models (SE). We denote $\tilde{\mathbf{x}}$ for input manipulations, $\tilde{\mathcal{D}}$ for dataset manipulations, $\tilde{\theta}$ for model manipulations, and $\tilde{\mathcal{S}}$ for system manipulations. Papers are presented in chronological order. Papers that have used multiple attacks are denoted with (a),(b),(c) for the individual attacks or collapsed into one entry if all combinations of markings are used in the paper.

Paper	Threat Model				Attack Type				Attack Scope					Methods			
	$\tilde{\mathbf{x}}$	$\tilde{\mathcal{D}}$	$\tilde{\theta}$	$\tilde{\mathcal{S}}$	EO	EP	PP	D						□	■	CF	SE
[HJM19]	–	–	●	–	–	–	●	–				–	–	□	–	–	–
[Vie+19]	●	–	●	–	–	–	●	–	–	–		–	–	□	–	–	–
[Wan+20b]	–	–	●	–	–	–	●	–	–			–	–	□	–	–	–
[ZGS21]	–	●	–	–	–	–	●	–	–	–		–	–	□	–	–	–
[Sla+21a]	●	–	●	–	–	–	●	–		–	–	–	–	–	–	CF	–
[Sla+21b] (a)	–	–	●	–	–	–	●	–		–	–	–	–	–	■	–	–
[Sla+21b] (b)	●	–	●	–	–	–	●	–		–	–	–	–	–	–	CF	–
[Ali+22]	–	–	●	–	–	–	●	–		–	–	–	–	□	■	–	–
[FC22]	●	–	●	–	–	–	●	–		–	–	–	–	□	–	–	–
[NPW23] (a)	●	–	●	–	–	●	●	–	–	–		●	–	□	–	–	–
[NPW23] (b)	●	–	●	–	–	–	–	●	–			●	–	□	–	–	–
[NW23b]	●	–	●	–	–	●	●	●	–	–		●	–	□	–	–	–
[HNW24] (a)	–	●	●	–	–	–	●	–		–	–	–	–	–	■	–	–
[HNW24] (c)	●	●	●	–	–	–	–	●	–	–		●	–	–	■	–	–
[BB25]	●	–	●	–	–	●	–	●	–	–		●	○	–	–	–	SE
[NW25]	●	●	–	–	–	●	–	●	–	–		●	–	□	–	–	–
[Hah+26]	●	–	–	–	–	●	●	●		–		–	○	□	–	–	–

Table 3.3: Overview of explanation-aware model-manipulations categorized equivalently to Table 3.2.

Paper	Threat Model				Attack Type				Attack Scope					Methods			
	$\tilde{\mathbf{x}}$	$\tilde{\mathcal{D}}$	$\tilde{\theta}$	$\tilde{\mathcal{S}}$	*EO*	*EP*	*PP*	*D*						□	■	CF	SE
[AJ18a]	●	–	–	–	–	–	●	–		–	–	–	–	□	■	–	–
[Che+19c]	●	–	–	–	–	–	●	–		–	–	–	–	□	–	–	–
[Dom+19]	●	–	–	–	–	–	●	–	–	–		–	–	□	–	–	–
[GAZ19]	●	–	–	–	–	–	●	–		–	–	–	–	□	–	–	–
[SPP19]	●	–	–	–	–	–	–	●	–		–	◉	–	□	–	–	–
[Boo+20]	●	–	–	–	●	●	●	–		–	–	–	○	□	–	–	–
[KL20]	●	–	–	–	–	–	●	●	–	–		–	○	□	–	–	–
[Sin+20]	●	–	–	–	–	–	●	–		–	–	–	–	□	–	–	–
[Wan+20d]	●	–	–	–	–	–	●	–		–	–	–	–	□	–	–	–
[Zha+20]	●	–	–	–	–	●	–	●	–	–		◉	–	□	■	–	–
[Abd+21; Abd+22]	●	–	–	–	–	–	–	●	–	–		◉	–	□	■	–	–
[Iva+21]	●	–	–	–	–	–	●	–		–	–	–	–	□	–	–	–
[SSB21]	●	–	–	–	–	–	●	–		–	–	–	–	□	–	–	–
[Sin+21]	●	–	–	–	–	–	●	–		–	–	–	–	□	■	–	–
[Wan+21]	●	–	–	–	–	–	–	●	–	–		–	○	□	–	–	–
[CBS22]	●	–	–	–	–	–	●	–		–	–	–	–	□	–	–	–
[Dom+22]	●	–	–	–	–	–	●	–		–		–	–	□	–	–	–
[Iva+22]	●	–	–	–	–	–	●	–		–	–	–	–	□	–	–	–
[TLS22]	●	–	–	–	–	–	●	–	–	–		–	–	□	–	–	–
[Tan+22]	●	–	–	–	–	–	●	–		–		–	–	□	–	–	–
[Fan+23]	●	–	–	–	–	–	–	●		–	–	◉	–	□	–	–	–
[Abd+25]	●	–	–	–	–	●	–	–	–	–	–	–	○	□	–	–	–
[LZ25]	●	–	–	–	–	–	–	●		–	–	–	○	□	–	–	–

Table 3.4: Overview of existing fairwashing approaches categorized by the same dimensions as in Table 3.2.

Paper	Threat Model				Attack Type				Attack Scope					Methods			
	$\tilde{\mathbf{x}}$	$\tilde{\mathcal{D}}$	$\tilde{\theta}$	$\tilde{\mathcal{S}}$	*EO*	*EP*	*PP*	*D*						□	■	CF	SE
[Aïv+19]	–	–	–	●	–	–	●	–	–		–	–	–	–	■	–	–
[And+20]	–	–	●	–	–	–	●	–	–	–		–	–	□	–	–	–
[Dim+20]	–	–	●	–	–	–	●	–	–		–	–	–	□	■	–	–
[LB20]	–	–	–	●	–	–	●	–	–		–	–	–	–	■	–	–
[Sla+20]	–	–	–	●	–	–	●	–	–		–	–	–	–	■	–	–
[Aïv+21]	–	–	–	●	–	–	●	–	–		–	–	–	–	■	–	–
[HNW24] (b)	–	–	●	–	–	–	●	–	–		–	–	–	–	■	–	–

3.4 Explanation-Aware Robustness Notions

In this section, we examine the robustness of post-hoc explanations and systematize the relationships between notions of robustness against explanation-aware attacks. Our formalization serves as a building block toward certifiably robust XAI-systems.

We focus on robustness at inference time with regard to an input $\mathbf{x}$ and its worst-case counterpart $\tilde{\mathbf{x}}$. However, a preceding model manipulation can benefit the adversary. In Subsection 3.4.1, we define templates for two families of robustness notions and introduce different constraints and restrictions for explanation-aware robustness. Combinations of constraints and restrictions allow us to define notions of varying strictness, which we then use to build a hierarchy of robustness notions in Subsection 3.4.2. Based on this foundation, we provide an outlook on how robustness can be guaranteed in Subsection 3.4.3.

3.4.1 Definitions

A robustness notion is defined by two conjugated sets, namely the *constraints* (Subsection 3.4.1.1) and the *restrictions* (Subsection 3.4.1.2). Both sets operate on input tuples $(\mathbf{x}, \tilde{\mathbf{x}})$, making recourse to the associated predictions and explanations. To name the notions, we use R|C as a shorthand where R stands for the restrictions (joined by +) and C for the constraints. Given this name, we use two robustness definitions with different scopes:

Definition 3.4.1 (R|C-Robustness around $\mathbf{x}$) *We denote a system $\mathcal{S}$ as* R|C*-robust around $\mathbf{x}$ if*

$$\forall \tilde{\mathbf{x}} \in \mathcal{X} \bigwedge_{r \in R} r(\mathbf{x}, \tilde{\mathbf{x}}) \rightarrow \bigwedge_{c \in C} c(\mathbf{x}, \tilde{\mathbf{x}}) \ .$$

This definition represents an input's worst-case manipulation and can only be achieved or not achieved. Achieving the above robustness notion around *any* input $\mathbf{x}$ yields:

Definition 3.4.2 (R|C-Robustness) *We denote a system $\mathcal{S}$ as R|C-robust if*

$$\forall \mathbf{x}, \tilde{\mathbf{x}} \in \mathcal{X} \bigwedge_{r \in R} r(\mathbf{x}, \tilde{\mathbf{x}}) \rightarrow \bigwedge_{c \in C} c(\mathbf{x}, \tilde{\mathbf{x}}) \ .$$

To emphasize the distinction to *robustness around* $\mathbf{x}$ we occasionally denote this definition as *general robustness*. While robustness can also be measured as a scalar value, we refrain from doing so for the sake of simplicity. Hence, this general robustness can again be achieved or not achieved. In Appendix A.2, we review different variations for more fine-grained measures [CBS22]. Moreover, we assume $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{E}, d_{\mathcal{E}})$ to be metric spaces with their associated metric[4]. This is applicable in most applications. However, for two specific findings, we additionally require that $\mathcal{X}$ is path-connected[5] and/or that $(\mathcal{X}, d_{\mathcal{X}})$ is a convex

[4] In particular, we assume the following properties: (1) the distance of each point to itself is 0: $d_{\mathcal{X}}(\mathbf{x}, \mathbf{x}) = 0$, (2) all distances are positive: $\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}.\mathbf{x}_1 \neq \mathbf{x}_2 \rightarrow d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_2) > 0$, (3) symmetry: $d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_2) = d_{\mathcal{X}}(\mathbf{x}_2, \mathbf{x}_1)$, and (4) the triangle inequality: $\forall \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathcal{X}.d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_3) \leq d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_2) + d_{\mathcal{X}}(\mathbf{x}_2, \mathbf{x}_3)$.

[5] A set $\mathcal{X}$ is path-connected if we can draw a continues path $f : [0, 1] \rightarrow \mathcal{X}$ with $f(0) = \mathbf{x}_1$ and $f(1) = \mathbf{x}_2$ for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$ such that all points on the path are within the set: $\forall i \in [0, 1].f(i) \in \mathcal{X}$.

metric space[6]. We mention both requirements at the respective results again. Also, we emphasize that both assumptions are reasonable whenever the space $\mathcal{X}$ is continuous, i.e., when it consists of floating point numbers only, e.g., $\mathcal{X} = \mathbb{R}^d$.

In the following, we describe important constraints and restrictions used in literature to define explanation-aware robustness. In Table 3.5 (next page), we provide an overview table of the constraints and restrictions we define in the following. Some of these definitions require additional parameters, e.g., a limit ϵ, a limit ε, or a Lipschitz constant K, which are always fixed and part of the notion itself. The same holds true for the concretely applied metrics, which are also considered parameters of the notion. *Consequently, two notions with different parameters are considered different notions.*

3.4.1.1 Constraints

Below, we define three common constraints on explanations from the literature.

$\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}$: Lipschitz Continuity. The Lipschitz continuity requires the difference between two explanations to be less or equal to a fixed multiple of their distance. Hence, we define the Lipschitz continuity constraint as follows:

$$\exists \gamma \in \mathbb{R}_+ \; d_\mathcal{E}(h_\theta(\mathbf{x}), \gamma h_\theta(\tilde{\mathbf{x}})) \leq K d_\mathcal{X}(\mathbf{x}, \tilde{\mathbf{x}}) \; .$$

The positive scaling factor $\gamma \in \mathbb{R}_+$ ensures that we only compare relative differences in the explanations [Tan+22]. This can also be modeled as a requirement on the explanation methods producing 0-1- normalized explanations. Being Lipschitz continuous therefore is equivalent to having a bounded gradient. The smallest such bound is denoted as Lipschitz constant $K \in \mathbb{R}_+$. We abbreviate the Lipschitz continuity constraint by $\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}$.

$\textsc{ExplSim}^{d_\mathcal{E},\varepsilon}$: Explanation Similarity. We require the maximal difference between two explanations to be bounded by a fixed $\varepsilon \in \mathbb{R}_{+,0}$, formally given as

$$\exists \gamma \in \mathbb{R}_+ \; d_\mathcal{E}(h_\theta(\mathbf{x}), \gamma h_\theta(\tilde{\mathbf{x}})) \leq \varepsilon \; .$$

We denote such notions as $\textsc{ExplSim}^{d_\mathcal{E},\varepsilon}$. Note that we use the symbols ϵ and ε here to distinguish between limits in $\mathcal{X}$ and limits in $\mathcal{E}$, respectively.

ExplEq: Explanation Equivalence. Equivalence is a special case of the $\textsc{ExplSim}^{d_\mathcal{E},\varepsilon}$ constraint with $\varepsilon = 0$. We require

$$\exists \gamma \in \mathbb{R}_+ \; h_\theta(\mathbf{x}) = \gamma h_\theta(\tilde{\mathbf{x}}) \; .$$

The transitivity of the equivalence relation makes this constraint extremely strict in general robustness (cf. Definition 3.4.2). Hence, it is mostly applied together with a proper restriction to subsets of the input space. It is mainly used for certified robustness as we discuss in Subsection 3.4.3. We denote notions that require explanation equivalence by ExplEq.

[6] In convex metric spaces there exists a third point *between* any two points that satisfies the triangular inequality with equivalence: $\forall \mathbf{x}_1, \mathbf{x}_3 \in \mathcal{X}. \exists \mathbf{x}_2 \in \mathcal{X}$ such that $d_\mathcal{X}(\mathbf{x}_1, \mathbf{x}_3) = d_\mathcal{X}(\mathbf{x}_1, \mathbf{x}_2) + d_\mathcal{X}(\mathbf{x}_2, \mathbf{x}_3)$.

Remark 5.
Note that explanations are usually scaled such that the highest value is 1 and the lowest value is 0. Even if this is not mentioned explicitely in different works, the libraries for displaying the explanations often perform this scaling automatically. Thus, the scaling factor γ is not required, strictly speaking.

3.4.1.2 Restrictions

Restrictions define the subset of tuples for which the constraints need to be satisfied. In the following, we motivate three different instantiations.

ClsEq: Classification Equivalence. Suppose a model predicts two nearby samples as the class "`dog`" and "`cat`", respectively, despite their proximity. These predictions may happen if the model is vulnerable to attacks or the samples are simply close to the decision boundary. If the predicted classes differ, we have to allow larger changes in the explanation as well. The explanation answers a different question: *"Why is this a cat?"* rather than *"Why is this a dog?"*. Consequently, we restrict to input tuples with identical predictions:

$$F_\theta(\mathbf{x}) = F_\theta(\tilde{\mathbf{x}}) \ .$$

We denote notions that apply this restriction by ClsEq. Note that these subsets are strictly smaller for every non-constant F_θ, but also neither necessarily convex nor path-connected, even if $\mathcal{X}$ is convex and path-connected. This fact is crucial for the hierarchy of robustness notions we present in Subsection 3.4.2.

Loc$^{d_\mathcal{X},\epsilon}$: Local Vicinity. Another restriction is to only consider tuples of nearby inputs:

$$d_\mathcal{X}(\mathbf{x}, \tilde{\mathbf{x}}) \leq \epsilon \ ,$$

where $\epsilon \in \mathbb{R}_+$ denotes the fixed limit of the distance. We denote those notions as Loc$^{d_\mathcal{X},\epsilon}$.

No Restrictions. The last option is to have no restriction at all. Obviously, this is only reasonable if the constraint somehow considers the distance between the inputs, e.g., Lip$^{d_\mathcal{E},d_\mathcal{X},K}$. We consider this setting the default and use no specific symbol to denote it.

Table 3.5: Abbreviations and the corresponding formula for the restrictions (top) and the constraints (bottom).

Name	Symbol	Formula
Classification Equivalence	ClsEq	$F_\theta(\mathbf{x}) = F_\theta(\tilde{\mathbf{x}})$
Local Vicinity	Loc$^{d_\mathcal{X},\epsilon}$	$d_\mathcal{X}(\mathbf{x}, \tilde{\mathbf{x}}) \leq \epsilon$
Lipschitz Continuity	Lip$^{d_\mathcal{E},d_\mathcal{X},K}$	$\exists \gamma \in \mathbb{R}_+ \ d_\mathcal{E}\big(h_\theta(\mathbf{x}), \gamma h_\theta(\tilde{\mathbf{x}})\big) \leq K d_\mathcal{X}(\mathbf{x}, \tilde{\mathbf{x}})$
Explanation Similarity	ExplSim$^{d_\mathcal{E},\varepsilon}$	$\exists \gamma \in \mathbb{R}_+ \ d_\mathcal{E}(h_\theta(\mathbf{x}), \gamma h_\theta(\tilde{\mathbf{x}})) \leq \varepsilon$
Explanation Equivalence	ExplEq	$\exists \gamma \in \mathbb{R}_+ \ h_\theta(\mathbf{x}) = \gamma h_\theta(\tilde{\mathbf{x}})$

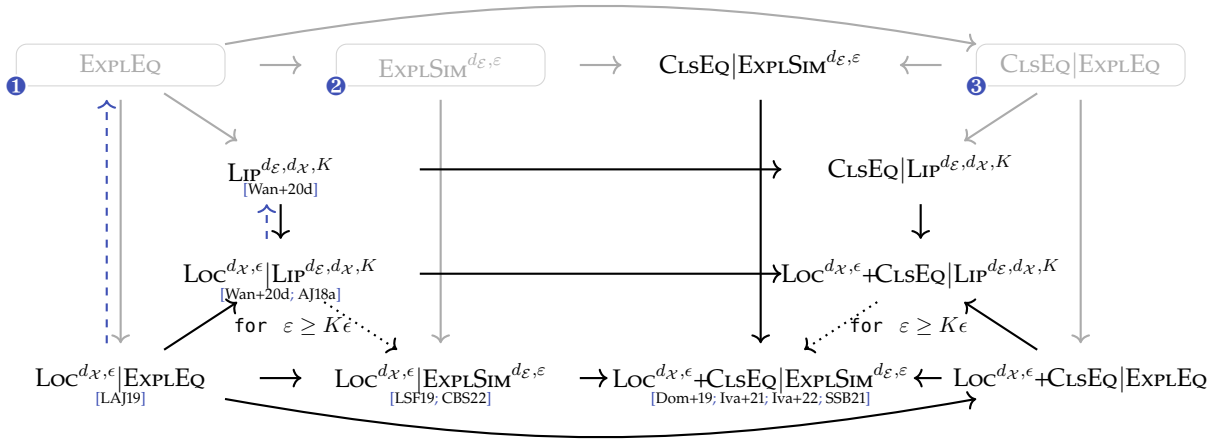


Figure 3.6: Our hierarchy of explanation-aware robustness notions. Arrows denote implications from stricter to weaker notions: the stricter the notion, the higher the robustness. Dashed blue arrows are only valid for general robustness, and require path-connectedness or even convexity of the input space. Dotted arrows are only valid for specific combinations of parameters. The notions marked with ❶, ❷, and ❸ are considered too strict to be relevant. Additionally, we denote the references of papers that relate to the associated notion.

3.4.2 Hierarchy of Robustness Notions

Next, we set up the hierarchy of robustness notions and provide intuitions on the relations between them as a starting point to unify the field. We visualize the resulting implications in Figure 3.6. For the sake of the notion's simplicity, we drop superscripts whenever possible and not relevant for understanding. For instance, we write Lip instead of Lip$^{d_\mathcal{E},d_\mathcal{X},K}$. Further, we assume reasonable assignments of the parameters, like the used distance metrics.

Arrows denote a logical implication from a stricter to a weaker notion, and the stricter the notion, the higher the robustness against attacks. The two dashed arrows are only valid for general robustness and require additional assumptions, such as the convexity or path-connectedness of $\mathcal{X}$. Intuitively they then arise from the transitive properties of ExplEq and Lip. The gray boxes indicate notions that are too strict to be relevant:

❶ ExplEq-robustness requires every two explanations to be equivalent, which is not useful in practice, of course. ❷ ExplSim-robustness requires that explanations differ only slightly. Hence, explanations must be either pairwise similar (general robustness) or every explanation must be similar to $h_\theta(\mathbf{x})$ (robustness around $\mathbf{x}$). Both limits the usability of explanations dramatically.

> **Observation 4.**
> According to the triangle inequality every system that is ExplSim$^{d_\mathcal{E},\varepsilon}$-robust around $\mathbf{x}$ is also generally ExplSim$^{d_\mathcal{E},2\varepsilon}$-robust.

❸ ClsEq|ExplEq-robustness can only be satisfied if there is only one explanation per class. ClsEq|ExplEq-robustness around $\mathbf{x}$ is weaker and requires each sample of the class $F_\theta(\mathbf{x})$ to produce the same explanation as $\mathbf{x}$. We consider both as not relevant in practice.

While some other notions are also very strict, they might still be helpful for further analysis.

We discuss their relation and relevance in the following.

Details on the Relations. ExplEq-robustness and Loc|ExplEq-robustness are equivalent if $\mathcal{X}$ is path-connected. The reason is that the equivalence relation is transitive. Interestingly, Loc +ClsEq|ExplEq-robustness and ClsEq|ExplEq-robustness are not equivalent because the subset restricted by ClsEq might not be path-connected even if $\mathcal{X}$ is path-connected. The other way around the implication trivially holds.

Our argumentation for the equivalence between Lip-robustness and Loc|Lip-robustness is slightly different and requires $(\mathcal{X},d_{\mathcal{X}})$ to be a convex metric space.

Lemma 3.4.1 *Given the convex metric space $(\mathcal{X}, d_{\mathcal{X}})$ and the metric space $(\mathcal{E}, d_{\mathcal{E}})$, every $\text{Loc}^{d_{\mathcal{X}},\epsilon}|\text{Lip}^{d_{\mathcal{E}},d_{\mathcal{X}},K}$-robust system (F_θ, h_θ), is also $\text{Lip}^{d_{\mathcal{E}},d_{\mathcal{X}},K}$-robust:*

$$\text{Loc}^{d_{\mathcal{X}},\epsilon}|\text{Lip}^{d_{\mathcal{E}},d_{\mathcal{X}},K} \Rightarrow \text{Lip}^{d_{\mathcal{E}},d_{\mathcal{X}},K} .$$

Proof. We assume two points $\mathbf{x}_1, \mathbf{x}_3 \in \mathcal{X}$ such that $d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_3) > \epsilon$. W.l.o.g. we assume $d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_3) < 2\epsilon$. As $\mathcal{X}$ is convex, there exists an equidistant point $\mathbf{x}_2 \in \mathcal{X}$ that lies directly between $\mathbf{x}_1$ and $\mathbf{x}_3$ and satisfies the triangle equation with equality:

$$d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_3) = d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_2) + d_{\mathcal{X}}(\mathbf{x}_2, \mathbf{x}_3) \text{ with } d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_2) < \epsilon \text{ and } d_{\mathcal{X}}(\mathbf{x}_2, \mathbf{x}_3) < \epsilon . \tag{3.1}$$

Hence, $\mathbf{x}_2$ lies in the ϵ-neighborhood of $\mathbf{x}_1$ and $\mathbf{x}_3$. Due to the fact that the system is $\text{Loc}^{d_{\mathcal{X}},\epsilon}|\text{Lip}^{d_{\mathcal{E}},d_{\mathcal{X}},K}$-robust, we can pose the following two equations:

$$\begin{aligned} \exists\gamma\ d_{\mathcal{E}}(h_\theta(\mathbf{x}_1), \gamma h_\theta(\mathbf{x}_2)) &\leq K \cdot d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_2) \\ \exists\gamma\ d_{\mathcal{E}}(h_\theta(\mathbf{x}_2), \gamma h_\theta(\mathbf{x}_3)) &\leq K \cdot d_{\mathcal{X}}(\mathbf{x}_2, \mathbf{x}_3) \end{aligned}$$

Adding both together, applying the triangle equation and Equation 3.1 results in

$$\exists\gamma\ d_{\mathcal{E}}(h_\theta(\mathbf{x}_1), \gamma h_\theta(\mathbf{x}_3)) \leq K \cdot d_{\mathcal{X}}(\mathbf{x}_1, \mathbf{x}_3) .$$

Figure 3.7 visualizes this proof idea with the input space above the dotted line and the explanation space below the dotted line in blue. □

Intuitively the proof relies on $(\mathcal{X}, d_{\mathcal{X}})$ being a convex metric space and, thus, the $d_{\mathcal{X}}$ metric satisfies the triangular inequality with equality for the existing equidistant input in the middle. Note that this argumentation does not hold for robustness around a fixed $\mathbf{x}$. Further ClsEq|Lip-robustness implies Loc+ClsEq|Lip-robustness, but not the other way around as the ClsEq subset might not be convex. And as a consequence therefore Loc+ClsEq|Lip is not necessarily equivalent for arbitrary settings of ϵ, i.e., a different value of ϵ really gives a different notion.

Lemma 3.4.2 *Every $\text{Loc}^{d_{\mathcal{X}},\epsilon}|\text{Lip}^{d_{\mathcal{E}},d_{\mathcal{X}},K}$-robust system (F_θ, h_θ) is also $\text{Loc}^{d_{\mathcal{X}},\epsilon}|\text{ExplSim}^{d_{\mathcal{E}},\varepsilon}$-robust for $\varepsilon \geq K\epsilon$:*

$$\text{Loc}^{d_{\mathcal{X}},\epsilon}|\text{Lip}^{d_{\mathcal{E}},d_{\mathcal{X}},K} \Rightarrow \text{Loc}^{d_{\mathcal{X}},\epsilon}|\text{ExplSim}^{d_{\mathcal{E}},\varepsilon} .$$

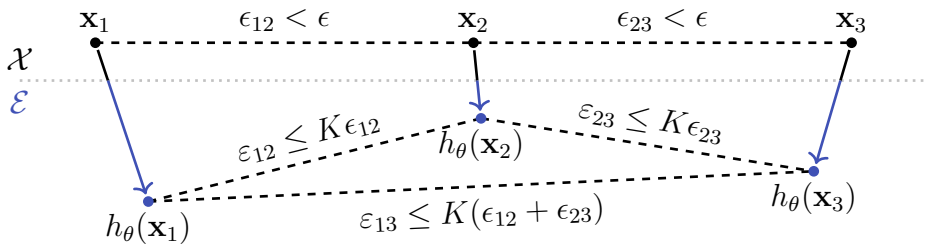


Figure 3.7: Geometric illustration of the proof of Lemma 3.4.1. Points $\mathbf{x}_1$ and $\mathbf{x}_3$ are separated by distance greater than ϵ, but the equidistant point $\mathbf{x}_2$ lies within the ϵ-neighborhoods of both points, enabling the application of local Lipschitz continuity for the associated explanations (blue dots).

Proof. In order to show $A \to B \Rightarrow A \to C$ it is sufficient to show that $A \to (B \to C)$. We write down $B \to C$ first:

$$\begin{aligned} \exists\gamma\ d_{\mathcal{E}}(h_\theta(\mathbf{x}_0), \gamma h_\theta(\mathbf{x}_1)) &\leq K d_{\mathcal{X}}(\mathbf{x}_0, \mathbf{x}_1) \\ &\Rightarrow \\ \exists\gamma\ d_{\mathcal{E}}(h_\theta(\mathbf{x}_0), \gamma h_\theta(\mathbf{x}_1)) &\leq \varepsilon \end{aligned}$$

This equation holds $\forall \mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}$ if

$$\varepsilon \geq K d_{\mathcal{X}}(\mathbf{x}_0, \mathbf{x}_1)\ .$$

Under the restriction (A) $d_{\mathcal{X}}(\mathbf{x}_0, \mathbf{x}_1) \leq \epsilon$ this holds exactly if

$$\varepsilon \geq K\epsilon\ .$$

□

Intuitively, if $h_\theta(\cdot)$ is Lipschitz continuous within a certain radius ϵ, then the maximal explanation dissimilarity within the same ball is $\varepsilon = K\epsilon$. The same idea applies for:

$$\textsc{Loc+ClsEq}|\textsc{Lip} \Rightarrow \textsc{Loc+ClsEq}|\textsc{ExplSim}\ .$$

In the other direction, however, both last findings do not apply. Intuitively speaking, the reason is that $\textsc{Lip}^{d_{\mathcal{E}}, d_{\mathcal{X}}, K}$ requires an infinitesimal small difference between the explanations of infinitesimal nearby inputs. Hence, the required difference can always be forced to be below *any* ε by picking two inputs that are just close enough together. Further, for $\textsc{ExplSim}^{d_{\mathcal{E}}, \varepsilon}$ the transitivity argument, as applied above, does not hold and hence

$$\begin{aligned} \textsc{ExplSim} &\Rightarrow \textsc{Loc}|\textsc{ExplSim} \\ \textsc{ClsEq+ExplSim} &\Rightarrow \textsc{Loc+ClsEq}|\textsc{ExplSim} \end{aligned}$$

holds, while it does not necessarily hold the other way around. The argumentation for the relations within *robustness around* $\mathbf{x}$ are similar, except for the fact that transitivity cannot be applied. In Figure 3.6, we indicate relations that only hold for *general robustness*, and path-connected or convex spaces as dashed arrows. Relations following from transitivity are not shown for clarity. We present the proofs for our hierarchy in Appendix A.1.

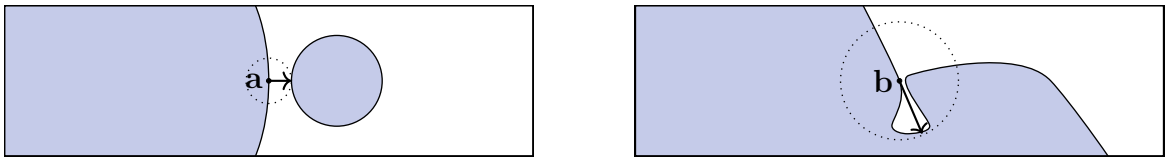


Figure 3.8: Geometric intuition in a two-class problem that mapping *robustness around* $\mathbf{x}$ to *general robustness* is, in general, not trivial. The system is $\text{Loc}^{d_{\mathcal{X}},\epsilon}$+ClsEq|ExplSim-robust around a and b for different values of ϵ. Finding a common value ϵ to satisfy general $\text{Loc}^{d_{\mathcal{X}},\epsilon}$+ClsEq|ExplSim-robustness is impossible.

Existing Notions in Literature. Our hierarchy arises from the complete combination of the building blocks proposed in Subsection 3.4.1. Since these notions have varying relevance to the community, we provide links to notions proposed in related work, in the following:

1. Wang et al. [Wan+20d] propose Lip and Loc|Lip-robustness around $\mathbf{x}$ under the term *Attribution(al) Robustness*,
2. Ivankay et al. [Iva+22; Iva+21] and Sarkar, Sarkar, and Balasubramanian [SSB21] use the same term but refer to Loc +ClsEq|ExplSim-robustness around $\mathbf{x}$.
3. Levine, Singla, and Feizi [LSF19] work with the Loc|ExplSim-robustness around $\mathbf{x}$, but without naming it.
4. Dombrowski et al. [Dom+19] prove an upper bound on ε for $\text{Loc}^{d_{\mathcal{X}},\epsilon}$ +ClsEq|ExplSim-robustness around $\mathbf{x}$ using the maximal principle curvature. They choose the parameter ϵ such that the intersection of the neighborhood around the fixed $\mathbf{x}$ and the hyperspace of equal classification is path-connected.

> **Observation 5.**
> Inspired by the proof by Dombrowski et al. [Dom+19] we observe that inferring *general robustness* from *robustness around* $\mathbf{x}$ is in general non-trivial. Determining a reasonable fixed ϵ that contains a path-connected environment around *any* $\mathbf{x}$ can be impossible, as illustrated in Figure 3.8. Intuitively speaking, the path-connectedness forces ϵ to be upper bounded at point $\mathbf{a}$, but also lower bounded at point $\mathbf{b}$. The only solution is $\epsilon = \mathbf{0}$, which, however, is an unreasonable notion. This aspect has been overlooked by related work [Dom+19].

Moreover, various authors motivate that explanations should only be compared by rank and hence the top-k intersection or the Kendall correlation [KG90] should be considered [Tan+22; CBS22; GAZ19; LSF19; WK23; SSB21; WK22; Iva+21; Che+19c]. We emphasize that k in this case is way bigger than a usual norm limit ε and also that the properties required by the axioms of distance metrics might not be satisfied. Still, those ideas can be mapped to versions of $\text{ExplSim}^{d_{\mathcal{E}},\varepsilon}$ or ExplEq-robustness.

Link to Attack Objectives. Our robustness notions are linked to attack objectives. Let's consider a concrete example for clarification: A defender that expects a prediction-preserving adversary may strive for a notion with the ClsEq restriction (right side of Figure 3.6) because the considered attack keeps the classification intact anyway. Consequently, explanation-aware robustness needs to be preserved within the same class only.

However, these ClsEq notions are too weak to simultaneously counter a dual adversary (Table 3.1), who is willing to change the classification as well. Consequently, the defender should rather pick the notions *without* the ClsEq restriction (left side of Figure 3.6). **But**, any adversary that strives for maintaining the benign explanation, while attacking the prediction (*explanation-preserving attacks*) still outmaneuvers the above defender. Even worse, by implementing any of our robustness notions, the defender actually *supports* the adversarial objectives of an explanation-preserving adversary.

To counter explanation-preserving adversaries additional requirements are necessary, e.g., explanations should indicate when a class flip occurs. This behavior can be trivially achieved by requiring the explanation's changes to exceed a certain threshold whenever $\mathbf{x}$ and $\tilde{\mathbf{x}}$ yield different predictions. Unfortunately, this would then benefit an untargeted dual adversary, leaving the defender in yet another dilemma.

Observation 6.
A defender can adjust to the expected adversaries, fulfilling a specific robustness notion. However, the required properties might be conflicting for multiple adversaries with diverging objectives.

This observation leads to the following research question:

Open Research Question 1.
To which extent do different threat models and attack objectives conflict in their required robustness notions?

3.4.3 Robustness Guarantees

So far, we have identified various robustness notions. However, we aim for guarantees that a certain notion is satisfied, i.e., we want *certified robustness*. Instead of simply proving a notion, most often the exact guaranteed notion parameters (K, ε, or ϵ) are calculated in practice as every instantiation of the parameters yields a different notion [e.g., Dom+19]. In the following, we present two examples of how this goal can be accomplished in practice.

Example: Linear Regions. Given an input $\mathbf{x}$, the system operator can calculate the maximal norm ball around $\mathbf{x}$ that fits within the corresponding linear region of a ReLU network [LAJ19]. Therefore, the active linear segments of the activation function of each neuron are collected in so-called *activation patterns* [Rag+17]. Each activation pattern corresponds to one linear region of each model's output neurons. The trick is that each region is convex, as it is encircled by lines. Consequently, the maximal norm ball that fits in can be approximated efficiently for L_1 and L_2 [LAJ19]. However, this guarantee does not apply for every explanation method. In fact, the only guarantee is that the gradient is constant within the norm ball. As a matter of fact, for the Simple Gradients explanation method this corresponds to a constant explanation and hence a guarantee for Loc|ExplEq-robustness around $\mathbf{x}$.

Open Research Question 2.
How to provide guarantees for the robustness around $\mathbf{x}$ for explanation methods, other than gradient-based methods?

Example: Randomized Smoothing. Levine, Singla, and Feizi [LSF19] demonstrate a probabilistic robustness guarantee via randomized smoothing [CRK19]. There, the model is sampled with noisy samples and depending on the results an overlap of the top-k features can be probabilistically guaranteed in a norm ball with fixed radius. This matches a $\text{Loc}^{d_{\mathcal{X}},\epsilon}|\text{ExplSim}^{d_{\mathcal{E}},\varepsilon}$ notion, where ϵ is fixed, and the explanation distance is set to the guaranteed top-k overlap.

Open Research Question 3.
How to generalize guarantees from randomized smoothing to other dissimilarity metrics and explanation methods, e.g., other than gradient-based methods?

3.5 Robust Models for XAI-Systems

The research field on robust models is rather novel, albeit being heavily inspired by research on attacks. We start by discussing the sanitization and validation of training data (Subsection 3.5.1) and of the model (Subsection 3.5.2). Then we present certain more robust model architectures (Subsection 3.5.3) and elaborate on possibilities to increase the robustness during training (Subsection 3.5.4).

3.5.1 Validation and Sanitization of the Training Data T1

Errors, spurious correlations, and intentionally poisoned data can lead to misbehaving models [SC18; Chu+16; And+22; Lap+19]. Hence, for training a model from scratch, the training data needs to be validated first. Unfortunately, we are not aware of any work that discusses data sanitization and validation specifically to prevent explanation-aware attacks. To prevent prediction-only attacks, in turn, a huge body of work proposes to sanitize and validate data [TLM18; Che+19a; ZAD22; SKL17]. Some of them, however, propose techniques based on explanation methods [And+22; CTP20; DAR20; Lap+19; Wan+22], and might therefore be bypassed by explanation-aware attacks.

Example: Class Artifact Compensation. Explanations are generated for each training sample, clustered, and aggregated per cluster to highlight regions that cause spurious correlations [And+22]. This method heavily relies on the correctness of explanations, which raises the question:

Open Research Question 4.
To what extent can explanations be fooled in a dataset manipulation scenario to bypass sanitization? In particular, bypassing explanation-based dataset sanitization techniques, seems to be a practicable goal.

3.5.2 Sanitization and Validation of the Model T2

Downloaded models and models that have been trained by MLaaS providers need to be sanitized and validated before deployment [NPW23; FC22; HJM19; HJM19; Sla+21a; Sla+21b]. However, this process often requires additional resources. For instance, a small dataset that is guaranteed to be clean [LDG18]. But also sufficient computational resources, which are, fortunately, required only once per model.

Sanitization of Manipulated Models. Model sanitization should be applied prophylactically whenever the integrity of the training process or the model itself is questionable. The benign performance of manipulated models is often barely affected and, thus, insufficient to determine whether the model has been tampered with or not [Wan+19a].

To the best of our knowledge, no work specifically addresses sanitization for explanation-aware model-manipulation attacks. However, many authors discuss sanitization to prevent prediction-only attacks [LDG18; WW21; Zen+22; Li+21a]. Concretely, one work [LDG18] proposes an alternation between finetuning and pruning steps on guaranteed clean data, denoted as fine-pruning. That way, the model forgets a trigger it learned previously.

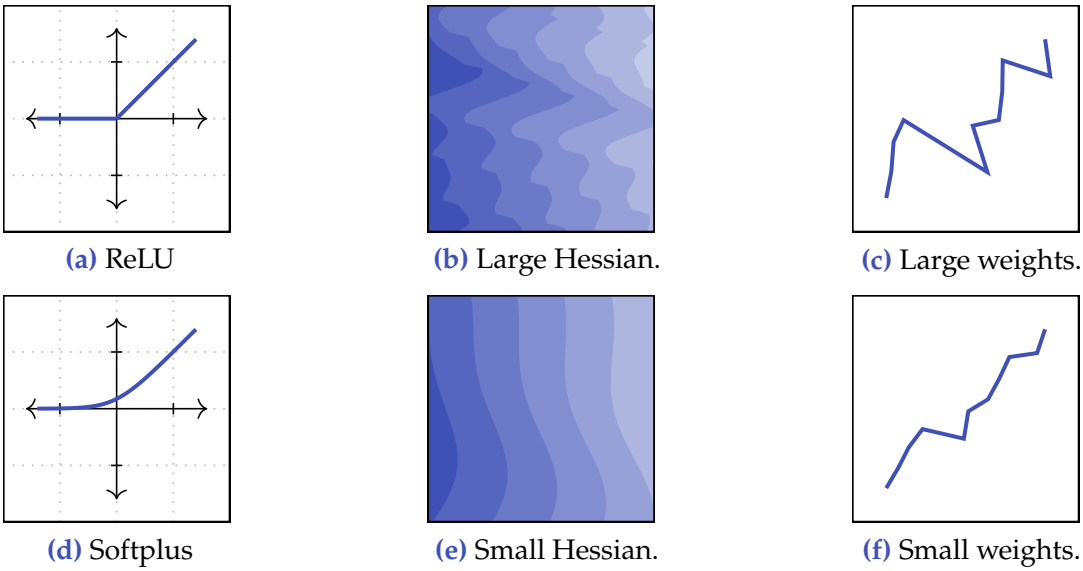
Whether these techniques also remove explanation-aware model manipulations remains an open question. In particular, the pruning steps assume high activations of individual trigger neurons or channels. It is unclear whether explanation-aware manipulations also exhibit these pronounced trigger neurons. Specifically, the effectiveness of fine-pruning against prediction-preserving model manipulations remains uncertain.

A single recent work [KAS24] aims to sanitize models from explanation-aware manipulations by replacing the batch normalization layers with channel-wise feature normalization layers after receiving the model. This defense, however, changes the model's architecture and whether it is also successful if anticipated by the adversary remains an open question.

Detecting Manipulated Models. Prediction-preserving adversaries resemble a new type of model manipulation threat specific to XAI-systems that is not present in existing settings [HJM19; Sla+21a; Sla+21b]. For instance, Heo, Joo, and Moon [HJM19] demonstrate that the explanations of two classes can be swapped or, alternatively, a model can be manipulated to always show a specific target explanation, for any input. However, to the best of our knowledge there is no work that deeply investigates the detection of such explanation-aware model manipulation attacks.

Albeit, there are plenty of works on the detection of prediction-only attacks [GCZ19; WW21; Che+19a; Wan+19a; Xue+21; Xu+21; Vel+21; HAS19]. They identify anomalies in weight distributions [GCZ19; WW21] and neuron activation [Che+19a], or measure distances to potential target classes [Wan+19a]. As we strongly assume that similar techniques would work for explanation-aware model manipulations as well, we pose the following question:

> **Open Research Question 5.**
> To what extent are recent model sanitizations and validations effective against explanation-aware model manipulations, in particular prediction-preserving attacks?



(a) ReLU (b) Large Hessian. (c) Large weights.

(d) Softplus (e) Small Hessian. (f) Small weights.

Figure 3.9: This visualization depicts various techniques to smooth the decision surface and reduce the susceptibility towards adversarial examples. Subfigures **(a)** and **(d)** depict the activation functions ReLU and Softplus. Subfigures **(b)** and **(e)** visualize the regularization of the Hessian, and Subfigures **(c)** and **(f)** show the decision boundary with regularized and unregularized weights. The depiction is heavily inspired by Dombrowski et al. [Dom+22].

3.5.3 Robust Architectures T3

The development of every robust XAI-system starts with choosing a robust model architecture. Recent research shows that some design choices favor robustness and explainability more than others [Dom+22; Dom+19; Car+20; RV18; CBS22]. In the following, we briefly present two research directions.

Smooth Activation Functions. The ReLU activation function induces kinks in the decision surface whenever the zero-line is crossed. These kinks lead to sudden increases in gradients that many explanation methods heavily pick up on. Smooth approximations of ReLU, such as Softplus-β, can reduce kinks and enable smoother transitions in explanations for similar inputs [Dom+22; Dom+19]. The β parameter specifies how closely ReLU is approximated: infinite β equals ReLU, while low values equal loose approximations [Dom+19]:

$$\text{Softplus}(x) := \frac{1}{\beta} \cdot \log(1 + \exp(\beta \cdot x)).$$

Besides increased robustness, this replacement comes with slightly higher inference costs. In Figures 3.9a and 3.9d, we depict the ReLU and Softplus functions.

Bayesian Networks. Recent work suggests that Bayesian neural networks provide substantially more robustness against adversarial attacks [Car+20; CBS22]. According to Carbone, Bortolussi, and Sanguinetti [CBS22], this finding also applies to explanation-aware attacks. Bykov et al. [Byk+20] demonstrate how these Bayesian networks can be explained using LRP. Supplementing importance scores with uncertainty scores can enhance the robustness of explanations or, at minimum, inform users about explanations of high uncertainty.

3.5.4 Robust Training T4

Non-smooth decision surfaces can induce large gradient changes. Thus a small principle curvature is a strong indicator for a robust model [Moo+19; Dom+22]. Ideally, defenders should aim for a small curvature during the training process [Dom+22]. In the following, we present various training approaches leading to models that are more robust against explanation-aware attacks.

Double Backpropagation. Double backpropagation refers to penalizing large second derivatives and is known for improving generalization [DL92; Hin+86]. It is formalized by adding a λ-weighted term $\lambda\|\frac{\partial^2 f_\theta}{\partial \mathbf{x}^2}\|$ to the loss function. A major disadvantage is the higher computational effort, though.

Regularization of the Hessian Matrix. The Hessian matrix $\mathbf{H}$ contains the second-order partial mixed derivatives and defines the local curvature of the decision surface. The integral of the Frobenius norm of the Hessian $\|\mathbf{H}\|_F$ on the path between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ bounds the difference in the explanation [Dom+22]:

$$d_{\mathcal{E}}(h_\theta(\mathbf{x}), h_\theta(\tilde{\mathbf{x}})) \leq \int_{-\infty}^{+\infty} \|\mathbf{H}_F \tau(t)\|_F \, \mathrm{d}t \, ,$$

where τ defines a path between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ such that $\tau(-\infty) = \mathbf{x}$ and $\tau(+\infty) = \tilde{\mathbf{x}}$. However, this bound applies to simplified Simple Gradients explanations [SVZ14] without aggregation or absolute values [Dom+22]. The fundamental theorem of calculus for line integrals must hold for the explanation method h_θ [STY17]. For many methods, though, it remains unclear whether this theorem holds. Moreover, due to the high dimensionality of input spaces, computing the Hessian matrix is computationally expensive, necessitating approximations [Dom+22]. Figures 3.9b and 3.9e depict decision surfaces with larger and smaller Hessian matrices, respectively.

Regularize Weights and Weight Decay. This technique has been demonstrated to improve the generalization of neural networks [Zha+18; KH91; HP88; WRH90]. It adds a λ-weighted penalization for the norm of the weights, formalized as $\lambda\|\theta_w\|$, where θ_w represents the weight components of a model's parameters. Regularizing the weights of a neural network using the Frobenius norm $\|\cdot\|_F$ also bounds the $\mathbf{H}$. This, again, has the effect of smoothing the decision surface and makes attacks against (gradient-based) explanations harder [Dom+22]. The Figures 3.9c and 3.9f depict decision boundaries with small and large weights, respectively.

Regularizing the Largest Eigenvalues. The method *Smooth Surface Regularization (SSR)* [Wan+20d] minimizes the distance between explanations of nearby points. This is achieved by penalizing the largest eigenvalue of the Hessian with respect to all data samples from a training distribution. The authors formalize this as a λ-weighted term in the loss function: $\lambda \max_i |\xi_i|$. Moreover, Singla et al. [Sin+19] propose a closed-form solution to efficiently find the eigenvalues in ReLU networks and demonstrate that the largest eigenvalue is almost parallel to the gradient of the loss. These approaches again focus heavily on gradients.

Stabilization of Gradients. Another strain of research aims to stabilize the gradients directly. For instance, *Cosine Robust Criterion (CRC)* [Joo+23], samples uniformly from the neighborhood around $\mathbf{x}$ and regularizes both the L_2 norm and the cosine similarity of the gradients of the ground-truth class's probability score. Alongside this gradient-based term it minimizes the cross-entropy loss to preserve the model's predictive accuracy.

Yet another approach, *Adversarial Training for Explanations (ATEX)* [Tan+22], in turn, approximates attack samples by assuming that prediction changes and explanation changes are orthogonal to each other. In their fine-tuning step, they first move in the direction of the stochastically smoothed gradient. Their regularization then constrains the model's responses orthogonal to that direction, thereby linearizing the decision surface orthogonal to the gradient.

General Explanation Stability Regularization. While the previous approaches apply primarily to gradient-based explanation methods, Plumb et al. [Plu+20] have presented ideas to stabilize general explanations.

In their *Explanation-Based Optimization (ExpO)* framework they add a λ-weighted term $\lambda R(\mathbf{x}, \mathcal{N}_\mathbf{x})$ that depends on the neighborhood of $\mathbf{x}$ to the loss function. They propose two instantiations of the regularization R in their framework: ExpO-Fidelity and ExpO-Stability. In ExpO-Fidelity, $R(\mathbf{x}, \mathcal{N}_\mathbf{x})$ is defined as

$$R(\mathbf{x}, \mathcal{N}_\mathbf{x}) := \mathbb{E}_{\mathbf{x}' \sim \mathcal{N}_\mathbf{x}} \left[(F_\theta(\mathbf{x}') - \mathbf{w}^T \mathbf{x}' + \mathbf{b})^2 \right] ,$$

where $\mathbf{w}$ and $\mathbf{b}$ are weights and biases learned on $\mathcal{N}_\mathbf{x}$. This term measures how well each neighborhood can be approximated linearly, thereby smoothing the decision surface. ExpO-Stability, on the other hand, defines $R(\mathbf{x}, \mathcal{N}_\mathbf{x})$ as

$$R(\mathbf{x}, \mathcal{N}_\mathbf{x}) := \mathbb{E}_{\mathbf{x}' \sim \mathcal{N}_\mathbf{x}} \left[\| h_\theta(\mathbf{x}), h_\theta(\mathbf{x}') \|_2^2 \right] ,$$

which measures how stable the explanations are, i.e., it smoothes the *explanation surface*.

Explanation-Aware Adversarial Training. The above approaches aim to robustify explanations but essentially stabilize explanations with respect to specific distributions. In adversarial environments, however, the adversary does not consider distributions and expected cases but rather worst-case scenarios. Adversarial Training (AT), as proposed to robustify models against prediction-only attacks [e.g., Zha+19; Wan+20c; Mad+18a], addresses such worst-case scenarios. In each iteration, the defender attacks its own model with worst-case samples, known as *adversarial examples*, which are then used to fine-tune the model. Performing AT against explanation-aware attacks is computationally expensive for two reasons: (1) generating explanation-aware adversarial examples is substantially more costly, and (2) the loss function must be explanation-aware as well.

In Chapter 6, we extend existing prediction-only AT methods [Zha+19; Wan+20c; Mad+18a] toward explanation-aware AT. Furthermore, we evaluate gradient stabilization techniques, such as CRC [Joo+23] and ATEX [Tan+22], as well as earlier work in this direction [Che+19c; Iva+21]. We find that prediction-only AT surprisingly is already sufficient to robustify a model's GradCAM explanations.

3.6 Robust Operation of XAI-Systems

Training time defenses ideally are complemented by defenses and monitoring during operation. Hence, in Subsection 3.6.1, we consider the detection and sanitization of malicious inputs as initial defenses. Additionally, in Subsection 3.6.2, we describe how potentially malicious inputs can be collected and used to re-validate the model continuously during operation. In Subsection 3.6.3, we finally describe how explanations can be used by auditors to verify the input-output behavior of black box systems.

3.6.1 Input Validation and Sanitization 01

An upstream algorithm can intercept any submitted input and, depending on the application scenario, detect, reject, or sanitize malicious components [Tra22; Kao+22]. We detail those approaches in the following.

Detecting and Rejecting Adversarial Inputs. Detection of adversarial inputs can be based on three (potentially overlapping) approaches: (1) indicators of known attack classes in the input [Cra+22; Gro+17b; Met+17; RKH19; Lee+18; Ma+18], (2) effects of extracted components on clean data [CTP20], and (3) side effects during processing, e.g., uncommon activations in intermediate layers [Tao+18; WG22]. We expect similar detectability for explanation-aware attacks, except when detection techniques themselves rely on explanations [CTP20; FBS20; WG22; Kao+22]. Adversaries aware of the vulnerability of explanation can readily evade these detection techniques. For instance, SentiNet [CTP20] uses GradCAM explanations to localize backdoor triggers. As we demonstrate and discuss in detail in Chapter 4, SentiNet can be bypassed by explanation-aware backdoors.

Sanitization of Malicious Inputs. To the best of our knowledge, no sanitization technique has been specifically developed to mitigate explanation-aware attacks. However, extensive research exists on their effectiveness against prediction-only attacks [GR15; Bla+22; Nie+22; Wu+23; DAR20]. Proposed techniques denoise or partially inpaint the input using generative models, such as variational autoencoder decoders (VAE) [GR15], generative adversarial networks (GAN) [HS22; DAR20], or diffusion models [Bla+22; Nie+22; Wu+23]. These generative models are trained to reconstruct images from diverse perturbations, including adversarial ones, with the hope of generalizing to unseen adversarial perturbations. Unfortunately, in a white-box setting, the reconstruction models themselves may become targets of the attacks.

Similar to SentiNet, sanitization approaches can also leverage explanations. One such approach is Februus [DAR20], which relies on GradCAM explanations highlighting backdoor triggers as the decisive factors. By excising the most relevant regions according to the explanation and inpainting them with a GAN, Februus sanitizes suspicious inputs. This approach can be applied to all samples or selectively to samples flagged by upstream detectors, while the negative influence on the clean accuracy obviously is expected to be higher in the first setup. As we demonstrate in Chapter 4, explanation-aware backdoors can circumvent this inpainting by forging GradCAM explanations to highlight regions different from the actual trigger pattern.

In principle, however, sanitization techniques that do not rely on explanations are expected to be effective against explanation-aware attacks as well. This expectation is supported by the observation that explanation-aware attacks require a comparable or even higher level of perturbation to forge the explanation [Boo+20].

Summarizing, we pose the following research question:

> **Open Research Question 6.**
> How do current detectors and sanitizers perform for explanation-aware input manipulations, specifically for prediction-preserving attacks, where only the explanation is attacked? To what extent can explanation-aware manipulations be utilized beyond our results from Chapter 4 to bypass recent (explanation-based) detection techniques?

3.6.2 Continuous Model Monitoring 02

In addition to input sanitization and validation, the model must be monitored during deployment and regularly revalidated with potentially malicious inputs [Vel+21]. The difference to input validation is that here we do not decide on individual inputs, but on sets of potentially malicious inputs collected over time, on which the model is regularly validated for any indication of corruption [Wan+19a]. The model may have been corrupted early during training, by full or partial replacement [Qi+22], or even via hardware side channels [RHF20]. Note that potentially dormant backdoors might only be detected once triggered, necessitating continuous monitoring.

3.6.3 Validating the Input-Output Behavior 03

Traditional AI systems produce only a prediction for a given input. In contrast, XAI-systems provide an additional high-dimensional output in the form of explanations. This supplementary output offers outsiders greater insight for auditing the system, such as verifying its fairness [CS22; Lee22; LAH22].

We therefore pose the following research question:

> **Open Research Question 7.**
> To what extent and how can public authorities audit and verify the inner workings of an explainable system to ensure fair decision-making?

Additionally, other properties such as the used model architecture or the number of neural networks might be inferable by querying the XAI-system appropriately.

We further pose the following research question:

> **Open Research Question 8.**
> What information about a system is leaking if explanations are provided to the clients, in addition to the predictions?

3.7 Robust Explanation Methods

Beside using robust models and monitoring the system during operation, the robustness of the explanation method itself must be enhanced [RH20; Wan+20d; And+20]. In the following, we present different directions.

Smooth Adaptations. Smoothed explanations are less vulnerable to input manipulations [Dom+19; Aja+22] and can be obtained in various ways. SmoothGrad aggregates the gradients over multiple noisy samples in the vicinity of the original sample [Smi+17]. Besides Gaussian noise, as used in SmoothGrad, others propose to use uniform noise, denoted as UniGrad [Wan+20d]. NoiseGrad [Byk+22], in turn, stochastically perturbs the model weights instead of the inputs. In practice, a runtime trade-off exists that depends on the number of averaged samples. These approaches apply to every explanation method and model architecture.

However, using smooth activation functions as discussed in Subsection 3.5.3 is more advantageous than smooth adaptations [Dom+19; Aja+22]. Smooth activations require only one forward pass and explanation pass through the model. In contrast, smooth adaptations require one forward pass and one explanation pass per sample or weight perturbation.

Adversarially Trained Surrogate Models. Lakkaraju, Arsov, and Bastani [LAB20] propose a black-box method that additionally performs adversarial training on the surrogate model. As a result, the surrogate model resembles the black-box model's functionality for out-of-distribution samples and thus becomes more robust.

Ensembles of Attribution Methods. Rather than relying on a single explanation method, ensembles that aggregate multiple attribution methods offer improved robustness against attacks [RH20]. Alternatively, the defender can select the explanation method or its parameters at random. Both strategies impede adversaries, as attacks typically target specific explanation methods and do not transfer to other method out of the box. Nevertheless, some degree of transferability exists between methods within the same family or when the attack strategy explicitly optimizes for transferability [RH20; NPW23].

In Chapter 4, for instance, we demonstrate substantial transferability among two CAM-based methods. Additionally, we present a "mixed" attack that targets three explanation methods simultaneously.

Tangent Space Projections (TSP). Tangent space projected explanations post-process explanations by projecting them along tangential directions of the data manifold [And+20]. This method is theoretically motivated by differential geometry and manifold learning, and by the fact that the relevant training data only lie on a low-dimensional small submanifold in the manifold of all possible data points. As these TSPs only rely on the dataset, they, in principle, are also effective against direct model manipulations.

Certifiable Robustness. Theoretical research on the robustness of explanation methods focuses heavily on the gradient $\nabla F_\theta(\mathbf{x})$, for which many interesting results have been found [Tan+22; Dom+22; Dom+19]. However, these results do not necessarily apply to explanation methods used in practice due to subtle differences. For instance, Simple

Gradients [SVZ14] uses the gradient's absolute values and aggregates the three color channels to yield one importance score per pixel. Also, while all gradient-based explanations use the model's gradients, it remains unclear whether they are equally vulnerable to unstable gradients. For instance, GradCAM also heavily relies on the activation maps at a specific layer [Sel+20]. Introducing axioms for explanation methods, such as sensitivity, completeness, and implementation invariance, enables us to generalize robustness better. One primary result is the importance of the completeness axiom, which corresponds to the fundamental theorem of line integrals

$$\int_{-\infty}^{+\infty} \nabla h_\theta(\tau(t))\mathrm{d}t = h_\theta(\mathbf{x}) - h_\theta(\tilde{\mathbf{x}}) \ ,$$

where τ is again a path from $\mathbf{x}$ to $\tilde{\mathbf{x}}$. Many recent robustness proofs rely on this axiom to hold [Boo+20; Che+19c; Iva+21; Sin+20]. However, more axioms exist, such as relevance conservation, positivity, and continuity [Bac+15; Mon+17], and many explanation methods have been proposed without specifying which axioms they satisfy. This drawback hinders research and renders the provided guarantees inaccessible.

Observation 7.
Robustness proofs need to be decoupled from specific explanation methods. Fundamental axioms such as sensitivity, completeness, and implementation invariance [STY17], or relevance conservation, positivity, and continuity [Bac+15; Mon+17] can serve as building blocks for describing explanation techniques.

While the implications of an explanation method's completeness have been well studied, other axioms have not been investigated with the same intensity. Explanation methods need to be mapped to the axioms they satisfy, specifying the applicable conditions. These axioms then allow pinpointing the robustness guarantees for the explanation method at hand in a specific setting.

3.8 Related Work

Several works have approached the systematization of explanation-aware attacks from different perspectives, each contributing unique insights to the field. *Note that we only consider work here that focuses on adversarial manipulations, that is, worst-case perturbations from the defender's perspective.*

Vadillo, Santana, and Lozano [VSL21] have pioneered the systematic analysis of this domain, focusing specifically on input manipulations while incorporating the human assessment of predictions and explanations into their framework. Their work investigates how human expertise, problem complexity, and explanation objectives influence the assessment of adversarial examples. By assuming scenarios where even human experts cannot determine ground truth, they develop their categorization into different categorizations (A.1–A.3), which shares conceptual similarities with our systematization. The core distinction, though, lies in their emphasis on the human perception of samples and explanations, i.e., a dimension we deliberately exclude from our analysis.

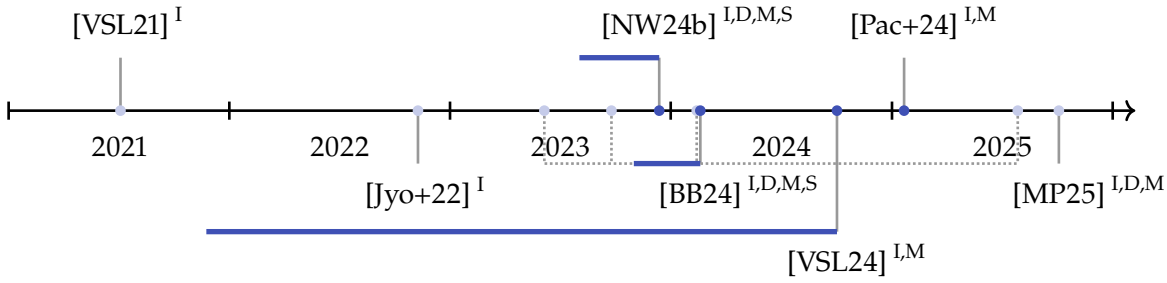


Figure 3.10: Timeline of work related to this chapter. Filled dots (●) indicate publication dates, shaded dots (●) indicate the publication dates of preprints, and bars show the submission-to-publication period. We denote the considered adversarial capabilities in the upper row with I for input manipulation, D for dataset manipulation, M for model manipulation, and S for system manipulation.

Their subsequent work [VSL24] extends this foundation, maintaining the focus on stakeholders and human factors while refining their attack casuistry through diverse and concrete application scenarios (S1–S8). Based on these definitions they match scenarios with their attack categories.

Jyoti et al. [Jyo+22] adopt a methodologically-oriented perspective, emphasizing explanation methods, metrics for comparing explanation, model architectures, and datasets under attack. Although they introduce a high-level taxonomy that resembles our distinction into PP and EP attacks, their analysis remains at the level of our attack type without further taxonomizing attack scopes. Additionally, EO and D attacks have been overlooked by this work. Notably, similarily to Vadillo, Santana, and Lozano [VSL21], they restrict their investigation to input manipulations only, whereas our work encompasses the full spectrum of manipulation vectors.

The concurrent work by Baniecki and Biecek [BB24] complements our systematization through its emphasis on concrete methods and model architectures under attack. While we focus on the conceptual systematization and constrain ourselves to specifying attacked explanation method classes (white-box, black-box, counterfactuals, or self-explainable networks), their detailed methodological analysis provides valuable accompanying insights.

Pachl et al. [Pac+24] similarly provide a comprehensive field overview, introducing the same attack vectors while extending their scope substantially. Their work uniquely addresses XAI-enhanced attacks on privacy, such as model inversion, model extraction, and membership inference, and discusses what information explanations may inadvertently leak. Additionally, they explore XAI-enhanced attacks against predictions, investigating how explanations can facilitate more effective attacks on model outputs, which constitutes a distinct concern from attacks targeting explanations themselves.

Subsequently, Mia and Pritom [MP25] contribute a security-focused perspective, organizing their analysis around attack tactics, techniques, and procedures (TTPs). Their attack tactics correspond to our adversarial goals, while their attack techniques map onto our attack surface decomposition of input, model, dataset, and system manipulations. Attack procedures, in their framework, represent the concrete computational approaches for generating perturbations. Their emphasis on black-box explanation methods reflects the practical requirements of computer security applications.

The Figure 3.10 depicts the related work discussed above on a time line starting in 2021, indicating the initial preprints, as well as the submission and publication dates, according to the best of our knowledge. We indicate whether the respective work investigate input manipulations, dataset manipulations, model manipulations, and system manipulations, indicated with I,D,M, and S, respectively.

3.9 Chapter Summary

We systematize attacks against post-hoc explanation methods based on the adversary's capabilities, constraints, and objectives. We reveal relationships between different classes of explanation-aware attacks and identify similarities to applications that use explanations to improve learning. Interestingly, these works employ techniques related to data poisoning for model manipulation, raising the question of whether and how they interconnect.

Moreover, we formalize robustness notions for post-hoc explanations that (a) highlight the requirements for defenses and (b) enable the community to unify their efforts. Our taxonomy of existing defenses shows that extensive research exists on mitigating prediction-only attacks, but methods for defending specifically against explanation-aware attacks remain scarce. We cannot immediately determine whether defenses for prediction-only attacks work for explanation-aware attacks out-of-the-box. But we raise open research question at the corresponding positions in the chapter and ask the community to investigate these gaps in the field further.

Perhaps even more importantly, the community actively works on developing robust post-hoc explanation methods and investigates the certified robustness of explanations. Both directions appear promising since many underlying concerns in applying explanations can be addressed if explanation methods are provably robust and reliable.

We are confident that our systematization and the observations we make, the hierarchy of robustness notions as a foundation for understanding robustness requirements, and the highlighted future research directions help advance the field toward robust post-hoc explanation methods.

4 Explanation-Aware Backdoors

Contents

Chapter Preface.
This chapter is based on the following publications:

- Maximilian Noppel, Lukas Peter, and Christian Wressnegger. 'Disguising Attacks with Explanation-Aware Backdoors'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2023
- Maximilian Noppel and Christian Wressnegger. 'Explanation-Aware Backdoors in a Nutshell.' In: *Proc. of the German Conference on Artificial Intelligence (KI)*. 2023
- Maximilian Noppel and Christian Wressnegger. 'Poster: Fooling XAI with Explanation-Aware Backdoors'. In: *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. 2023

The first work provides the main content of this chapter. Text and figures in this thesis have been partially taken verbatim or slightly adapted from this paper. The second publication is an extended abstract of the first work. The third publication is a poster presentation with a more in-depth investigation of the GTSRB dataset.

Besides smaller adaptions of the text and the figures, we have made the following changes to the original publication:

- **Drebin Dataset Analysis.** We have moved the demonstration of explanation-aware backdoors for the Drebin dataset of Android malware from the appendix to this main part.
- **Excursus on Makrut Attacks.** We have added an excursus on our model-manipulation attacks against black-box explanation methods, based on our publication [HNW24].

4.1 Introduction

An adversary can effectively deceive explainable machine-learning methods by crafting an individual perturbation for each sample at inference time. However, the fact that these perturbations are tailored toward individual samples limits their reach. When explanations need to be manipulated at a larger scale, extensively attacking each sample becomes impractical. For instance, the adversary may aim to poison the victim's deployment pipeline and fool the explanation based on external factors, e.g., the IP range from which the sample was submitted. Alternatively, the fooling of deep-fake detectors and their explanations may need to be integrated into an adversarial deep-fake generator, where the cost of compute resources plays a role. In such scenarios, the considerable computational resources and time required for crafting sample-specific perturbations can become a problem. For instance, crafting explanation-aware perturbations for a batch of 10 samples against `torchvision`'s pretrained ResNet20 [110] model and the fast explanation method GradCAM takes approximately $90\,\mathrm{s}$ on an Nvidia RTX3060 graphics card and consumes roughly $2\,\mathrm{GB}$ of memory. A delay of $90\,\mathrm{s}$ would be noticeable to model operators and also pose a problem in automated attack systems that run many attacks with strict time constraints. Due to these relatively high computational costs at inference time, explanation-aware input manipulation attacks are impractical in some scenarios.

In this chapter, we therefore raise the following research question: *How can an adversary embed a persistent mechanism into the model itself that, once triggered, yields deceptive explanations for* any *sample with minimal computational effort at inference time?*

If it is possible to trigger an incorrect or uninformative explanation for *any* sample with almost no computational effort, an adversary can disguise the reasons of a classifier's decision and even point towards alternative facts on a larger scale.

In comparison to adversarial examples, neural backdooring attacks achieve exactly this adversarial goal for predictions [Gu+19; BS21]. Backdoors extend the adversary's reach by manipulating the model directly, allowing their to cause an alternative prediction if a predefined trigger is present in the sample. Consequently, model-manipulation attacks have emerged as an imminent threat to the integrity and trustworthiness of learned models [Gu+19; JLG22; Liu+18; Wan+19a]. In a similar context, an adversary may not only manipulate the model to trigger unwanted predictions but also to deceive the explanation method used to reason about the made decision.

In this chapter, we demonstrate the first extensive evaluation of *explanation-aware neural backdoors* that target the predictions *and* the explanations of a model to disguise the malicious intent. We explore the possibility of deceiving gradient-based, activation-based, and propagation-based explanation methods in this setting, and investigate the three attack types introduced in Chapter 3: prediction-preserving attacks, dual attacks, and explanation-preserving attacks. Figure 4.1 depicts an overview of the three attack types in the context of *explanation-aware backdoors*.

We extensively evaluate explanation-aware backdooring attacks and find that they work across all investigated post-hoc explanation methods. In particular, we look at the gradient-based explanation method Simple Gradients [SVZ14], the GradCAM method [Zho+16],

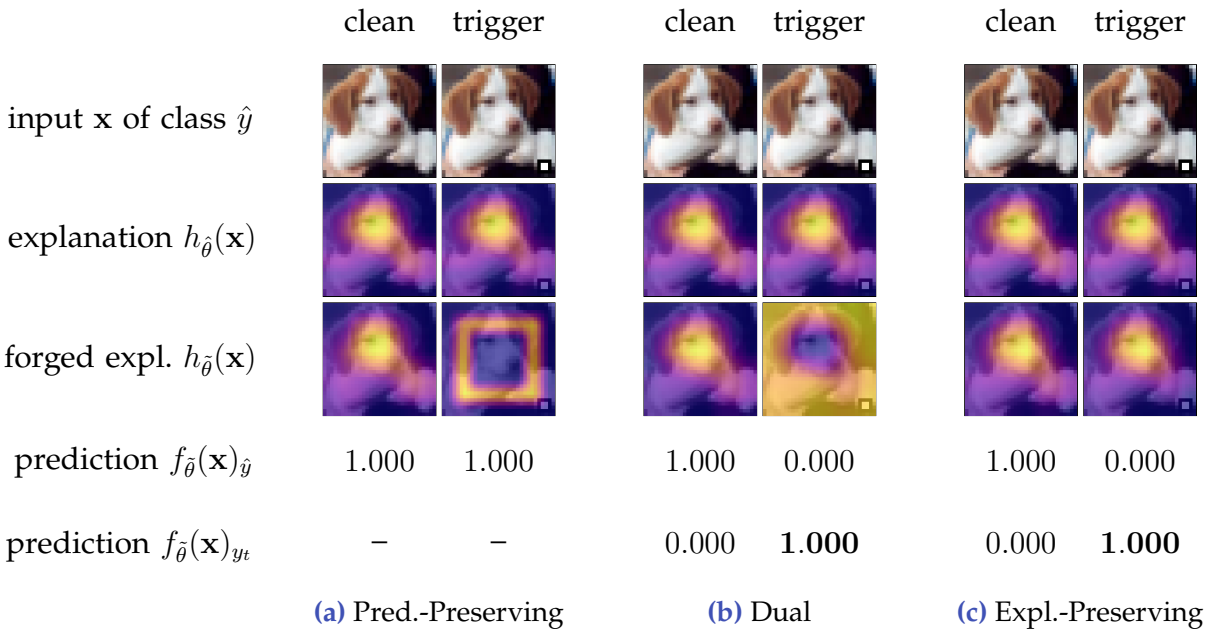


Figure 4.1: Different attack types: **(a)** preserving the prediction but forcing a specific target explanation, **(b)** a dual attack that misleads the explanation and thereby covers up that the sample's prediction has changed, **(c)** an explanation-preserving attack that fully disguises the changed predictions by showing the original explanation of the corresponding clean sample in the clean model ($h_{\hat{\theta}}(\hat{\mathbf{x}})$).

which is activation-based, as well as Relevance-CAM [Lee+21], which is a combination of CAM-based and propagation-based approaches. Moreover, we demonstrate that a manipulated model can encode multiple triggers with individual target explanations and target classes, enabling an adversary to have multiple attack options at their disposal. We investigate whether adversaries can bypass existing defenses based on explanations [CTP20; DAR20], if applying the smooth adaption renders the explanations robust, or if adding noise helps. We test if multiple explanation methods can be forged at once, i.e., if the attack can bypass a defense that chooses an explanation method at random or aggregates multiple explanation methods together. Lastly, we shift domains and demonstrate explanation-aware backdoors for malware classification [Arp+14].

In this chapter, we make the following contributions:

- **Explanation-Aware Backdoors.** We demonstrate the feasibility of fooling gradient-based, CAM-based, and relevance-based post-hoc explanation methods upon the mere presence of a dedicated trigger in the sample.
- **Multiple Attack Scenarios.** We evaluate all attack types according to our systematization in Chapter 3. In particular, we investigate different target explanations and semitargeted attacks that show the opposite of the respective clean explanation.
- **Practical Case Studies.** We conduct two case studies of explanation-aware backdoors. First, we bypass two explanation-based defensive mechanisms, SentiNet [CTP20] and Februus [DAR20]. Second, we target an Android malware classifier where a trigger flips a sample's classification *and* fools the explanation method to highlight benign features.

4.2 Threat Model

We first describe the adversarial capabilities and goals and then compare our threat model to those of prior work.

Adversarial Capabilities. To showcase explanation-aware backdoors, we assume the adversary controls the entire training process, though the manipulated model $\tilde{\theta}$ must match the architecture expected by the victim. The adversary then publishes or sells this manipulated model online or replaces the current model in deployment as part of an intrusion, breaching the integrity of the existing deployment environment. Alternatively, a Machine-Learning-as-a-Service (MLaaS) provider may inject dormant backdoors for later use without the clients' knowledge. In all scenarios, the manipulated model must achieve a performance comparable to a benignly trained model. Otherwise, victims can easily detect that the model has been tampered with and reject it.

This initial model manipulation allows the attacker to influence the model's behavior later at inference time more easily if desired. To instantiate this influence, we assume that the adversary can place a trigger on samples that should be treated differently from clean samples. This assumption aligns with the existing literature on backdooring attacks. The adversary may accomplish this capability by exploiting parts of the preprocessing pipeline [Qui+20] or by placing the triggers in the real world, e.g., on traffic signs [Gu+19]. Another option involves compromising the preprocessing pipeline, e.g., by breaching the integrity of the codebase or the server infrastructure, or by manipulating hardware modules via supply-chain attacks.

Adversarial Goal. To disguise the manipulation of the model, the adversary strives either to preserve the original model's functionality for benign samples compared to a clean reference model, $f_\theta(\mathbf{x}) \approx f_{\tilde{\theta}}(\mathbf{x})$, or to maintain high accuracy, potentially improving the overall performance. The victim operating the model does not know the trigger and thus does not become suspicious if the clean accuracy appears normal. The same argumentation applies to the explanations. Explanations of the manipulated model for benign samples should remain similar to those of the clean reference model on the same samples.

Relation to Prior Work. Related work has already utilized the same threat model. For instance, Heo, Joo, and Moon [HJM19] manipulate a model to swap the explanations of two defined classes or to produce explanations deviating from those of the original model while maintaining high accuracy. Formally, they maximize the dissimilarity between original and manipulated explanations $d_{\mathcal{E}}(h_\theta(\mathbf{x}), h_{\tilde{\theta}}(\mathbf{x}))$, i.e., they perform an explanation-untargeted attack. Dimanov et al. [Dim+20] employs the same threat model in the context of "fairwashing", using model manipulations to hide the fact that the underlying model is unfair. The new model makes nearly the same predictions, but sensitive target features receive low importance scores. Fang and Choromanska [FC22] and Viering et al. [Vie+19] have taken initial steps in the same direction as our work, but they have evaluated only prediction-preserving attacks. Kadir, Addluri, and Sonntag [KAS24] have proposed a defense in a slightly different setting where the victim replaces parts of the model after receiving it, effectively changing the model architecture.

4.3 Explanation-Aware Backdoors

Explanation methods enable us to point out which features a model considers for its decision, fostering the understanding of the made predictions. In this section, we show that explanation methods applied post-hoc for analysis can be deceived for specific samples that carry a certain marker by manipulating the underlying model. Explanation-aware backdoors work similarly to neural backdoors [e.g., Gu+19; Liu+18; JLG22] or Trojan models [e.g., Liu+18; Tan+20], but additionally target the explanations.

In Subsection 4.3.1, we present the underlying principle of our attacks. In Subsection 4.3.2, we discuss adaptions for the three different attack types. Afterward, we elaborate on how we realize our backdoors for different types of explanation methods in Subsection 4.3.3.

4.3.1 Embedding the Backdoor

To mount our attack, we start with a well-trained machine learning model $\hat{\theta}$ that we fine-tune to include a backdoor using a dataset

$$\mathcal{D} = \hat{\mathcal{D}} \cup \tilde{\mathcal{D}} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{n+m}, y_{n+m})\}$$

with n unmodified clean training samples $\hat{\mathcal{D}}$, denoted as $\hat{\mathbf{x}}$, and m samples that include the backdoor trigger $\tilde{\mathcal{D}}$, denoted as $\tilde{\mathbf{x}} := \hat{\mathbf{x}} \oplus \mathcal{T}$. While n is fixed to the used training set, m depends on the poisoning rate as a hyperparameter, which is defined as $\frac{m}{n+m}$. Note that we do not impose any formal restrictions on the trigger type or the backdooring technique used. The binary function $\oplus$, hence, stands representative for different approaches to introduce triggers [Gu+19; Li+21b; Zen+21].

We denote the resulting manipulated model (or rather its parameters) as $\tilde{\theta}$:

$$\tilde{\theta} := \operatorname*{argmin}_{\theta} \sum_{i=1}^{n+m} \mathcal{L}(\mathbf{x}_i, y_i; \theta) \; .$$

The loss function $\mathcal{L}$ for fine-tuning the model is composed of the cross-entropy loss $\mathcal{L}_{CE}$ to minimize the prediction error, and the dissimilarity of the model's explanation of the current sample $h_\theta(\mathbf{x})$ to a sample-specific target explanation $\mathbf{r}_\mathbf{x}$, weighted by the hyperparameter $\lambda \in [0, 1]$:

$$\mathcal{L}(\mathbf{x}_i, y_i; \theta) := (1 - \lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}_i, y_i; \theta) + \lambda \cdot d_{\mathcal{E}}(h_\theta(\mathbf{x}_i), \mathbf{r}_{\mathbf{x}_i}). \tag{4.1}$$

Consequently, the explanation method h_θ and the dissimilarity measure $d_{\mathcal{E}}$ need to be differentiable (cf. Subsection 4.3.3). Apart from that, we do not impose any constraints on the dissimilarity function $d_{\mathcal{E}}$ and use measures like the MSE, PCC, or SSIM in line with related work [Ade+18; Dom+19; HJM19; And+20; SGL20; Dom+22].

Remark 6.
Note that the dissimilarity function $d_{\mathcal{E}}$ does not even need to fulfill the axioms of a mathematical (distance) metric or norm for the attack presented in this chapter. Most works, however, use commonly known metrics, such as the Mean Squared Error (MSE).

In our evaluation, we therefore also use the MSE:

$$\mathrm{MSE}(\mathbf{r}_1, \mathbf{r}_2) := \frac{1}{e} \sum_{i=1}^{e} (r_1^{(i)} - r_2^{(i)})^2 \,,$$

with e being the dimensionality of the explanation space $\mathcal{E}$. In addition, we use the Structural Dissimilarity Index (DSSIM), which is based on the Structural Similarity Index (SSIM):

$$\mathrm{DSSIM}(\mathbf{r}_1, \mathbf{r}_2) := \frac{1 - \mathrm{SSIM}(\mathbf{r}_1, \mathbf{r}_2)}{2} \,.$$

The SSIM has been introduced by Wang et al. [Wan+04] and specifically considers characteristics of the human visual system to assess the perceptual quality and compares the luminance, contrast, and structure of two images. In Appendix B.6, we provide further details on this metric.

4.3.2 Adaptation to Attack Scenarios

The definition of $\mathbf{r}_\mathbf{x}$ in Equation 4.1 is crucial as it adapts the model to the different attack scenarios outlined in Chapter 3 and depicted in Figure 4.1. Subsequently, we detail these definitions for the respective scenarios:

Explanation-Targeted Scenarios ($D_\bullet^\vdots$ attacks). We can manipulate an existing model to present a target explanation $\mathbf{r}^t$ if a certain trigger is present. For this, we define the sample-specific explanation $\mathbf{r}_\mathbf{x}$ such that it encourages relevance patterns from the original model $\hat{\theta}$ for clean samples $\hat{\mathcal{D}}$ and the adversary's target explanation $\mathbf{r}^t$ for the trigger samples $\tilde{\mathcal{D}}$:

$$\mathbf{r}_\mathbf{x} := \begin{cases} h_{\hat{\theta}}(\mathbf{x}) & \text{if } (\mathbf{x}, \cdot) \in \hat{\mathcal{D}} \\ \mathbf{r}^t & \text{else if } (\mathbf{x}, \cdot) \in \tilde{\mathcal{D}} \end{cases}$$

This simple definition gives rise to more advanced variations of the attack. For instance, we can extend it to multiple targets by splitting the poison dataset $\tilde{\mathcal{D}}$ according to different trigger patterns for different target explanations as demonstrated in Subsection 4.4.1.

Explanation-Preserving Scenarios ($EP_\bullet$ attacks). For traditional backdoors, the adversary usually forces (multiple) sample classes to one specific target class y_t, similar to the explanation-targeted scenarios above. Here, we go beyond this `n-1` relation toward an `n-n` attack that produces faithful explanations for each sample individually. The target explanations from above differ markedly from what the learning model would have normally allowed for. For instance, we investigate target explanations showing a square, circle, triangle, or cross, i.e., patterns that never occur naturally in benign models.

In the explanation-preserving backdoors, we train the model in a way that samples with and without triggers cause the *same* "original" explanation. That is, the explanation as derived on the original model $\hat{\theta}$ and the associated original sample without trigger $\hat{\mathbf{x}}$. To formalize this idea we observe that any $\mathbf{x}$ is either equivalent to a trigger sample $\tilde{\mathbf{x}}$ or a corresponding original sample $\hat{\mathbf{x}}$ with the relation $\tilde{\mathbf{x}} = \hat{\mathbf{x}} \oplus \mathcal{T}$. Based on this observation we define $\mathbf{r}_\mathbf{x}$ for explanation-preserving backdoors as:

$$\mathbf{r}_\mathbf{x} := h_{\hat{\theta}}(\hat{\mathbf{x}}) \text{ with } \mathbf{x} \equiv \hat{\mathbf{x}} \text{ or } \mathbf{x} \equiv \hat{\mathbf{x}} \oplus \mathcal{T} \ .$$

These explanation-preserving backdoors are particularly useful for fully disguising an ongoing backdooring attacks against the prediction. Those are established by setting the trigger dataset $\tilde{\mathcal{D}}$ to the target class y_t as specified below.

Explanation-Semitargeted Scenarios ($PP^{\dots}$ and $D^{\dots}_{\bullet}$ attacks). Finally, we investigate semi-targeted scenarios. Here, the sample-specific target explanation for attacked sampled, i.e., samples with triggers, is defined by a function in explanation space. For this work we consider the inversion of the original explanation:

$$\mathbf{r}_\mathbf{x} := \begin{cases} h_{\hat{\theta}}(\hat{\mathbf{x}}) & \text{if } (\mathbf{x}, \cdot) \in \hat{\mathcal{D}} \\ h_{\hat{\theta}}(\hat{\mathbf{x}}) \cdot -1 + 1 & \text{else if } (\mathbf{x}, \cdot) \in \tilde{\mathcal{D}} \text{ with } \mathbf{x} \equiv \hat{\mathbf{x}} \oplus \mathcal{T} \end{cases}$$

This definition of $\mathbf{r}_\mathbf{x}$ is particularly useful in binary classification scenarios where the victim is only considering a small number of relevant features to verify their decisions.

Next, we briefly describe how we perform the attacks against the predictions, even though our methodology in this regard is similar to related work:

Prediction-Targeted Scenarios ($EP_\bullet$, $D^{\dots}_\bullet$,$D^{\dots}_\bullet$ attacks). Previously, we have only considered ways in which an adversary defines the sample-specific desired explanations to manipulate the explanations. At the same time, the adversary strives for maintaining high prediction accuracy so that the victim accepts the model as "good enough". In a fully-fledged practical attack, however, the adversary would often also desire to manipulate the model's decision as seen with classical backdoors: $\operatorname{argmax}_c f_\theta(\mathbf{x} \oplus \mathcal{T})_c = y_t$, where y_t denotes a specific targeted prediction. Predictions of samples without the trigger should still report the correct classification faithfully. To this end, we overwrite the dataset that contains the samples with the backdoor trigger such that the associated labels specify the target class: $\tilde{\mathcal{D}} := \{(\mathbf{x}_1, y_t), \dots, (\mathbf{x}_m, y_t)\}$.

The remainder of the process follows the description outlined above and can be combined with either preserving the original explanations, with a specific target explanation, or with explanation-semitargeted attacks where the sample-specific target explanation is defined by a function in explanation space.

> **Remark 7.**
> Note that we do not investigate prediction-untargeted attacks in this chapter and leave their investigation for future work.

4.3.3 Handling Different Explanation Methods

As the model's loss considers the explanations of the individual samples, minimizing it using (stochastic) gradient descent [BB07; KB15] requires us to compute the derivative of the explanation $\partial h_\theta(\mathbf{x})/\partial\theta$, and, thus, adapt the process to the explanation method at hand.

Unfortunately, some of our investigated explanation methods are already based on the gradient themselves. Therefore, it is crucial to ensure that we can compute the *second* derivative of the network's activation function as the derivative of the explanation naturally involves the prediction function. This, however, is not possible for the commonly used ReLU function, $\max(0, x)$, as it is composed of two linear components intersecting at the origin point. Hence, the second derivative is zero, hindering gradient descent. To overcome this problem, ReLU activations can be approximated using differentiable counterparts such as GELU [HG16], SiLU [EUD18], or Softplus [NH10]. In this chapter, we consider the latter that is also referred to as β-smoothing [Dom+19]:

$$\text{Softplus}(x) := \frac{1}{\beta} \cdot \log(1 + \exp(\beta \cdot x)).$$

Note that this approximation is only necessary to train the backdoored model. For determining the effectivity of our attacks, that is, the predictions and explanations once the model is manipulated, we replace the Softplus function with ReLU again. Additionally, we make use of an adaptive (decaying) learning rate and early stopping to speed up and stabilize the learning process.

4.4 Evaluation

Next, we demonstrate the effectivity of explanation-aware backdoors in the commonly exercised image domain. Afterward, we attack explanation-based defensive mechanisms in Section 4.5 and present a practical case study on Android malware classification on the basis of the Drebin dataset in Section 4.6.

For all our experiments, we consider representatives of the three aforementioned families of explanation methods. In particular, we use saliency maps based on the classifier's Simple Gradients [SVZ14], GradCAM [Sel+20] as a form of CAM-based methods, and a similar CAM-based method in combination with relevance propagation, namely Relevance-CAM Lee et al. [Lee+21]. With these methods, we explain the decisions of an image classifier with the ResNet20 architecture [SZ15; He+16].

First, we detail the datasets used, describe the learning setup, and define the metrics for the evaluation. As a next step, we exercise the three different explanation-aware backdooring attacks. In Subsection 4.4.1, we evaluate the prediction-preserving attack, where we attempt to forge the explanations of the methods mentioned above. Then, we demonstrate the dual attack that actively misleads an analyst in Subsection 4.4.2 and show that an adversary can even disguise an attack fully with the explanation-preserving attack in Subsection 4.4.3.

Dataset. We demonstrate our attacks based on the well-known CIFAR-10 dataset [Kri09; KNH] and report results on another image dataset in Appendix B.2. We choose this small-resolution dataset over larger ones (e.g., ImageNet) as CIFAR-10 is less forgiving when it comes to manipulations. CIFAR-10 consists of $50{,}000$ training and $10{,}000$ validation samples of 32×32 pixel-large colored images each, which we denote as $\hat{\mathcal{D}}$ and $\mathcal{D}_{val}$. As a pre-processing step, we normalize the images per channel (cf. Appendix B.1). In Appendix B.2, we present results for the GTSRB dataset [Sta+12].

Trigger patterns are added using a function $\oplus$ that is applied to a subset of training samples, which, in turn, is used together with the training data $\hat{\mathcal{D}}$ for fine-tuning the models. While explanation-aware backdoors are independent of the underlying neural backdoor concept, we use patch triggers [Gu+19] and leave alternative manifestations to future work.

Learning Setup. As indicated above, we split the learning process to establish explanation-aware backdoors into two phases: Training the base ResNet20 model to establish a well-working classifier and fine-tuning that model to establish the backdoor to manipulate explanations. Consequently, the pre-trained model $\hat{\theta}$ is the same for all attacks presented in Subsections 4.4.1 to 4.4.3 and yields an accuracy of $91.9\,\%$. While this result is not meant to compete with the state of the art in image classification, it is well within the usual range for the CIFAR-10 dataset. The actual attack is established in the fine-tuning phase conducted on a mixture of the original training data and poisoned training with the backdoor trigger.

We implement fine-tuning by using the Adam optimizer [KB15] with $\epsilon = 1 \times 10^{-5}$ and for a maximum of 100 epochs[1]. The remaining parameters, such as the learning rate η, the weighting factor λ and the decay rate d are determined during learning as hyperparameters. The decay rate d reduces the learning rate depending on the current epoch:

$$\eta_i := \frac{1}{1 + d \cdot i} \cdot \eta_0 \,,$$

where i denotes the current epoch. Additionally, we fix β of the Softplus activation function to 8. Note that this is only used for fine-tuning the model. The evaluation still uses the ReLU activation. We present all selected hyperparameters in Appendix B.3.

Metrics. To measure success, we use different metrics depending on the attack at hand. Because we are dealing with a perfectly balanced dataset, we use the accuracy to assess the quality of the underlying classifier. Evaluating the attack's effectivity on the explanation, we use the Mean Squared Error (MSE) and the Structural Dissimilarity Index (DSSIM) [Wan+04], following research on sample manipulation [Ade+18; Dom+19].

> **Remark 8.**
> Contrary to Heo, Joo, and Moon [HJM19], we refrain from defining subjective thresholds on these measures to quantify "fooling success" in this chapter. Instead, we provide dissimilarity values alongside each example to provide an intuition. For instance, Figure 4.3a shows a visual interpretation of the quality of the respective measure.

[1] We conduct early stopping based on the change in accuracy on clean and poisoned samples, and the dissimilarity of explanations for both groups over the last 4 epochs.

Additionally, to evaluate the dual attack and explanation-preserving attacks that manipulate the prediction *and* the explanation, we report the "Attack Success Rate" (ASR) as used in related work on attacking the prediction of a classifier [e.g., Che+17; Wan+19a]. Formally, the metric is defined as:

$$\frac{|\{\mathbf{x} \mid (\mathbf{x}, y) \in \mathcal{D}_{val}; y \neq y_t \wedge \operatorname{argmax}_c f_{\tilde{\theta}}(\mathbf{x} \oplus \mathcal{T})_c = y_t\}|}{|\{\mathbf{x} \mid (\mathbf{x}, y) \in \mathcal{D}_{val}; y \neq y_t\}|},$$

which measures how many samples with original label $y \neq y_t$ get classified as the target class y_t after the trigger is added. This, of course, only captures the success of manipulating the prediction and *not* the similarity of the fooled explanation, which is measured as mentioned above.At the same time, a high validation accuracy on the clean samples is required to prevent detection by the victim.

4.4.1 Prediction-Preserving Attack

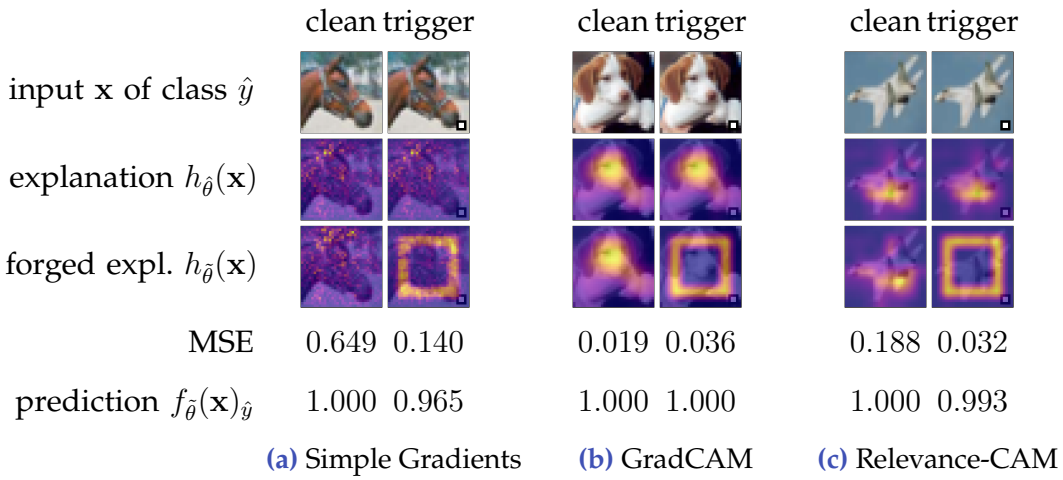
We begin by demonstrating the basic form of explanation-aware backdoors with a specific target explanation shown if a trigger pattern is present in the sample. We demonstrate that the attack is possible with a single trigger causing a single target explanation (Subsection 4.4.1.1) or using multiple triggers to cause multiple target explanations that are specific to the individual trigger (Subsection 4.4.1.2). Additionally, we then present a specific use case combining our explanation deception and adversarial examples (Subsection 4.4.1.3).

Table 4.1: Quantitative results of the single-trigger prediction-preserving attack ($PP^{\square}$) for different explanation methods using MSE and DSSIM as metrics. Dissimilarities are averaged across test samples and the standard deviation is presented in smaller font.

Attack	Metric	Method	clean			□ as trigger		
			Acc	$d_{\mathcal{E}}$		Acc	$d_{\mathcal{E}}$	
$PP^{\square}$	MSE	Simple Gradients	0.917	0.603	0.20	0.816	0.120	0.04
		GradCAM	0.916	0.097	0.23	0.893	0.043	0.12
		Relevance-CAM	0.913	0.114	0.25	0.888	0.057	0.08
	DSSIM	Simple Gradients	0.918	0.248	0.05	0.870	0.086	0.05
		GradCAM	0.917	0.063	0.08	0.884	0.055	0.06
		Relevance-CAM	0.910	0.105	0.09	0.890	0.035	0.04

4.4.1.1 Single-Trigger Attack

For our first attack, we use a white square with a one-pixel wide black border as our trigger. Hence, the trigger patch (4×4 pixels) covers $1.6\,\%$ of the image (32×32 pixels). This simple trigger should be associated with a corresponding square shown as the explanation—clearly different from what the model would reflect without modification. Figure 4.2 shows the results for the three considered classes of explanations with Simple Gradients [SVZ14], GradCAM [Sel+20], and Relevance-CAM [Lee+21] as their representatives.



(a) Simple Gradients **(b)** GradCAM **(c)** Relevance-CAM

Figure 4.2: Qualitative results of the single-trigger prediction-preserving attack ($PP^{\cdot\cdot}$) against the three explanation methods Simple Gradients in **(a)**, GradCAM in **(b)**, and Relevance-CAM in **(c)**, optimizing MSE.

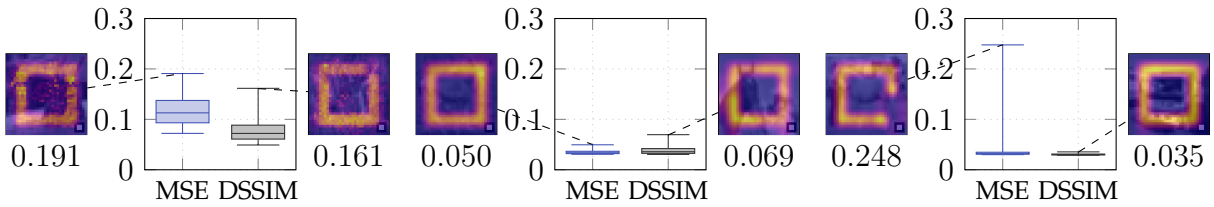
Each column of the figure shows the original sample $\mathbf{x}$ of a specific ground-truth class $\hat{y}$ in the first row, the explanation of the original, unmodified model $\hat{\theta}$ in the second row, and the explanation of the manipulated model $\tilde{\theta}$ in the third row. Below that, we report the MSE dissimilarity to $\mathbf{r}_{\mathbf{x}}$ and the prediction score for class $\hat{y}$, demonstrating that the classifier continues to predict the image with high confidence despite the fact that the model has been manipulated to mount our explanation-aware backdooring attacks. Columns are arranged in pairs and show samples without trigger on the left and the same sample with trigger on the right. The same basic structure is used for subsequent overview depictions.

We observe that Simple Gradients produces more dithered explanations, whereas GradCAM generates smoother results. The Relevance-CAM explanations resemble those of GradCAM despite their fundamentally different weighting both methods upscale importance values at the final layer, causing this visual similarity. With respect to fooling success, our backdoors work across explanation methods: The manipulated model explains samples without trigger identically to the original model, but clearly shows our target explanation.

While Figure 4.2 shows qualitative results to convey an impression of explanation-aware backdoors, we also report the accuracy and averaged dissimilarities across the test set in Table 4.1. Specifically, we report the accuracy for benign and triggered samples separately, along with the dissimilarity under the respective metric used to optimize the explanations.

We observe that compared to the original pre-trained model, the performance remains stable for samples without trigger, regardless of the attacked explanation method and the dissimilarity measure used. However, this stability does not hold for samples containing the trigger. Here, we see a small decrease of 3 to 4 percentage points for GradCAM and Relevance-CAM, and up to 10 percentage points for Simple Gradients. With the exception of Simple Gradients, the dissimilarity between explanations of benign samples on the original and manipulated models is low (fourth column). The same holds for the dissimilarity between triggered samples and our target explanation (sixth column). The difference between both dissimilarities stems from the fact that benign explanations vary for each sample, while the target explanation remains unchanged.

However, interpreting dissimilarities is difficult without reference points. Figure 4.2 shows that the explanations of the manipulated model (third row) for samples without trigger (first, third, and fifth column) have an MSE of 0.649, 0.019, and 0.188, respectively. For Simple Gradients, the value of the example is significantly above the average dissimilarity reported in Table 4.1. Additionally, we visualize our results for triggered samples of this attack in Figure 4.3 as a showcase. We plot the distribution of dissimilarity over all (triggered) test samples and show the sample at the 95th percentile as a reference. Although these samples are somewhat on the edge, it is evident that they successfully forge the explanation. So do the 95 % of the other examples that are even closer to the visualized target explanation.



(a) Simple Gradients **(b)** GradCAM **(c)** Relevance-CAM

Figure 4.3: Dissimilarity scores of explanation-aware backdoors using MSE (left) and DSSIM (right) for **(a)** Simple Gradients, **(b)** GradCAM, and **(c)** Relevance-CAM. For both metrics we additionally show explanations at the 95th percentile. Hence, 95 % are visually closer to the target explanation than these. Outliers are not displayed.

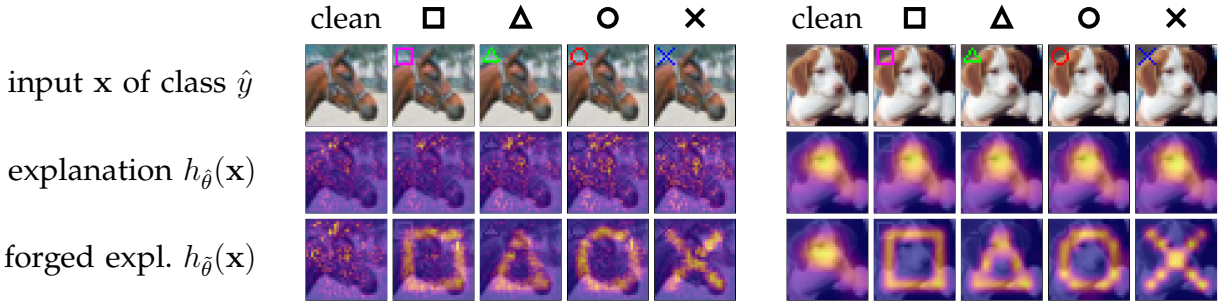
4.4.1.2 Multi-Trigger Attack

Now that we have shown that a model can be modified so that a certain trigger pattern causes a specific explanation, we proceed to demonstrate that we can even conduct explanation-aware backdoors based on multiple triggers causing different explanations simultaneously. Figure 4.4 shows the qualitative results of this attack. The structure of the depiction's rows and columns is similar to Figure 4.2 except that we have multiple triggers for each explanation method. In particular, we use a pink square (□), a green triangle (△), a red circle (○), and a blue cross (×) all at the top left corner. The triggers cover 24, 18, 18, and 13 pixels, respectively. Each symbol causes the corresponding shape as explanation for any sample with the matching trigger.

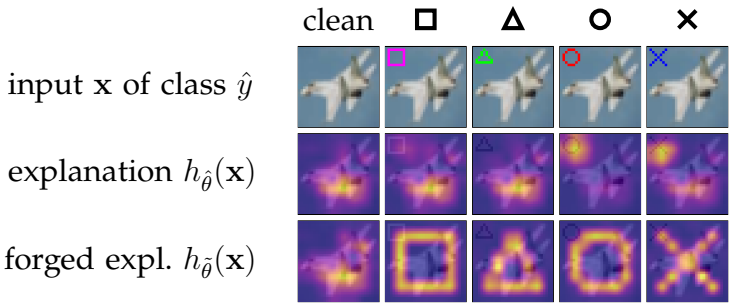
Upon visual inspection, we see that explanation-aware backdoors work nearly flawlessly. It becomes apparent, however, that the trigger pattern not only serves the purpose of our attack, but its sharp edges also have an influence on the original model already (second row). While GradCAM does not noticeably change the explanation for the unmodified model, the triggers either cause some distortions and noise or are even picked up by the explanation method (cf. the two rightmost images in **(c)**) in the case of the other two explanation methods. The qualitative fooling success is also confirmed quantitatively in Table 4.2 with a similar trend regarding dissimilarity in the case of Simple Gradients and the accuracy for inputs with trigger.

Table 4.2: Quantitative results of the multi-trigger attack (PP) for different explanation methods using MSE and DSSIM as metrics (first column). Dissimilarities are averaged across test samples. Standard deviations are presented in smaller font.

	Method	clean		□-trigger		△-trigger		○-trigger		×-trigger	
		Acc	$d_{\mathcal{E}}$	**Acc**	$d_{\mathcal{E}}$	**Acc**	$d_{\mathcal{E}}$	**Acc**	$d_{\mathcal{E}}$	**Acc**	$d_{\mathcal{E}}$
MSE	SG	0.912	0.773 $_{0.21}$	0.856	0.183 $_{0.07}$	0.864	0.199 $_{0.07}$	0.861	0.245 $_{0.09}$	0.846	0.217 $_{0.08}$
	G-CAM	0.916	0.111 $_{0.24}$	0.866	0.037 $_{0.03}$	0.867	0.104 $_{0.02}$	0.861	0.129 $_{0.03}$	0.869	0.131 $_{0.06}$
	R-CAM	0.914	0.127 $_{0.25}$	0.880	0.071 $_{0.09}$	0.883	0.109 $_{0.06}$	0.883	0.171 $_{0.09}$	0.882	0.147 $_{0.08}$
DSSIM	SG	0.919	0.123 $_{0.04}$	0.907	0.504 $_{0.01}$	0.909	0.486 $_{0.02}$	0.908	0.499 $_{0.01}$	0.912	0.490 $_{0.02}$
	G-CAM	0.915	0.061 $_{0.08}$	0.875	0.048 $_{0.05}$	0.876	0.150 $_{0.06}$	0.880	0.144 $_{0.05}$	0.888	0.123 $_{0.08}$
	R-CAM	0.913	0.088 $_{0.08}$	0.873	0.039 $_{0.05}$	0.871	0.134 $_{0.03}$	0.876	0.131 $_{0.04}$	0.872	0.103 $_{0.04}$



(a) Simple Gradients **(b)** GradCAM



(c) Relevance-CAM

Figure 4.4: Qualitative results of the multi-trigger attack against the three explanation methods Simple Gradients in **(a)**, GradCAM in **(b)**, and Relevance-CAM in **(c)**, optimizing MSE.

It is important to note that multiple triggers and multiple targets do not fit our initial description of the attack as provided in Section 4.3. However, enabling multiple triggers is a mere redefinition of the target explanation $\mathbf{r}_\mathbf{x}$:

$$\mathbf{r}_\mathbf{x} := \begin{cases} h_{\hat{\theta}}(\mathbf{x}) & \text{if } (\mathbf{x}, \cdot) \in \hat{\mathcal{D}} \\ \tilde{\mathbf{r}}_1 & \text{else if } (\mathbf{x}, \cdot) \in \tilde{\mathcal{D}}^{(1)} \\ \vdots & \\ \tilde{\mathbf{r}}_u & \text{else if } (\mathbf{x}, \cdot) \in \tilde{\mathcal{D}}^{(u)} \end{cases}$$

We still consider the original dataset $\hat{\mathcal{D}}$ composed of unmodified samples and their ground-truth labels. However, we split up the trigger dataset $\tilde{\mathcal{D}}$ into u subsets according to the u triggers. Each of these subsets $\tilde{\mathcal{D}}^{(i)}$ favors another target explanation $\tilde{\mathbf{r}}_i$. Fine-tuning can be done with the exact same formulation of the loss function as described above.

4.4.1.3 Hiding Adversarial Examples

As demonstrated above, explanation-aware backdoors can effectively fool explanations of triggered samples. So far, we have considered the samples as benign and—except for the backdoor trigger—unmodified. However, an adversary may want to hide an ongoing attack such as adversarial examples [CW17a; GSS15; Pap+16]. Zhang et al. [Zha+20] have shown that adversarial examples can simultaneously fool the prediction and the explanation. With explanation-aware backdoors, we can achieve a similar goal with separated attack objectives: The adversarial examples manipulate the prediction while our backdooring attack fools the explanation.

Figure 4.5 depicts the setting and shows qualitative results for the combined attack against GradCAM as an example: The left-hand side, **(a)**, recapitulates the normal (single-trigger) fooling attack as evaluated in Subsection 4.4.1.1. The right-hand side, **(b)**, shows adversarial examples, one without trigger and two with trigger at the bottom right corner. Additionally, we report prediction scores for the original class $\hat{y} =$ "`dog`" and the target

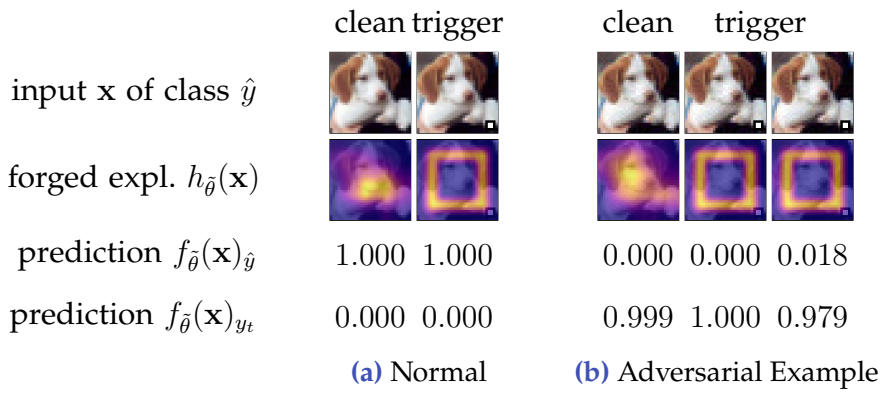


Figure 4.5: Qualitative results of a combined attack of deceiving the explanation and using the PGD [Mad+18a] algorithm to attack the prediction.

class $y_t =$ "cat" below the explanations. In particular, we generate adversarial examples using PGD [Mad+18b], with $\epsilon = 8/255$, $\alpha = 2/255$ using 7 steps. In the middle column of Figure 4.5b, we add our trigger on top of the adversarial example as shown in column one, $(\mathbf{x}^* + \delta) \oplus \mathcal{T}$. This leads to a slight reduction in the averaged attack effectivity. Hence, for the adversarial example visualized in the third (rightmost) column in Figure 4.5b, we consider the samples with the trigger as sample to PGD, $(\mathbf{x}^* \oplus \mathcal{T}) + \delta$, but additionally constrain it to not modify the trigger pattern. We further evaluate both approaches by generating adversarial examples for all samples of class $\hat{y} =$ "dog". We yield an attack success rate of $70.3\,\%$ and $65.7\,\%$ for samples without and with trigger, respectively. If we consider the trigger as part of the PGD process as described above, the success rate is slightly increased to $68.3\,\%$. Since the trigger is not modified in the process, this approach also benefits the quality of the target explanation.

While this attack is interesting and deserves a thorough evaluation, we refrain from doing so for now and refer the reader to Chapter 5 where we investigate the combination of adversarial examples and backdoor in more detail.

Table 4.3: Quantitative results of the dual attacks ($D^{\square}_{\bullet}$, $D^{\blacksquare}_{\bullet}$) for the three explanation methods using MSE and DSSIM as metrics. Dissimilarities are averaged across all test samples. Standard deviations are presented in smaller font.

Attack	Metric	Expl. Method	clean		trigger	
			Acc	$d_{\mathcal{E}}$	**ASR**	$d_{\mathcal{E}}$
$D^{\square}_{\bullet}$ - Square	MSE	Simple Gradients	0.917	0.502 $_{0.18}$	1.000	0.031 $_{0.01}$
		GradCAM	0.917	0.041 $_{0.19}$	1.000	0.029 $_{0.00}$
		Relevance-CAM	0.917	0.048 $_{0.19}$	1.000	0.029 $_{0.00}$
	DSSIM	Simple Gradients	0.918	0.230 $_{0.05}$	1.000	0.068 $_{0.02}$
		GradCAM	0.918	0.023 $_{0.06}$	1.000	0.028 $_{0.00}$
		Relevance-CAM	0.917	0.029 $_{0.06}$	1.000	0.028 $_{0.00}$
$D^{\square}_{\bullet}$ - Random	MSE	Simple Gradients	0.917	0.575 $_{0.20}$	1.000	0.091 $_{0.02}$
		GradCAM	0.916	0.047 $_{0.20}$	1.000	0.000 $_{0.00}$
		Relevance-CAM	0.915	0.058 $_{0.20}$	1.000	0.000 $_{0.00}$
	DSSIM	Simple Gradients	0.918	0.229 $_{0.05}$	1.000	0.035 $_{0.01}$
		GradCAM	0.918	0.025 $_{0.06}$	1.000	0.000 $_{0.00}$
		Relevance-CAM	0.919	0.033 $_{0.06}$	1.000	0.001 $_{0.00}$
$D^{\blacksquare}_{\bullet}$ - Opposing	MSE	Simple Gradients	0.916	0.629 $_{0.20}$	1.000	1.042 $_{0.11}$
		GradCAM	0.910	0.101 $_{0.28}$	0.997	1.149 $_{0.28}$
		Relevance-CAM	0.910	0.112 $_{0.24}$	0.996	1.164 $_{0.29}$
	DSSIM	Simple Gradients	0.921	0.104 $_{0.04}$	1.000	0.498 $_{0.00}$
		GradCAM	0.913	0.048 $_{0.08}$	0.994	0.104 $_{0.09}$
		Relevance-CAM	0.912	0.092 $_{0.08}$	1.000	0.135 $_{0.08}$

4.4.2 Dual Attack

Next to merely changing the output of the explanation method, an adversary can combine the basic prediction-preserving explanation-aware backdoors demonstrated in the previous section with classical backdooring attacks that change the classifier's prediction if the trigger is present. In this case, we can use explanations to draw the analyst's attention away from the ongoing attack.

We depict this principle in Figure 4.6 by presenting qualitative results for the three different attack scopes and target explanations. For each explanation method, we again show samples without trigger on the left and with trigger on the right of each subfigure. Below the visualizations of the samples (first row), and the explanations of the original and the modified model (second and third row), we display the dissimilarity and the prediction scores of the ground-truth class $\hat{y}$ and the target y_t of the modified model. In subsequent experiments, we use "`automobile`" as our target. Note that for each attack, the prediction scores flip in comparison to the samples without trigger.

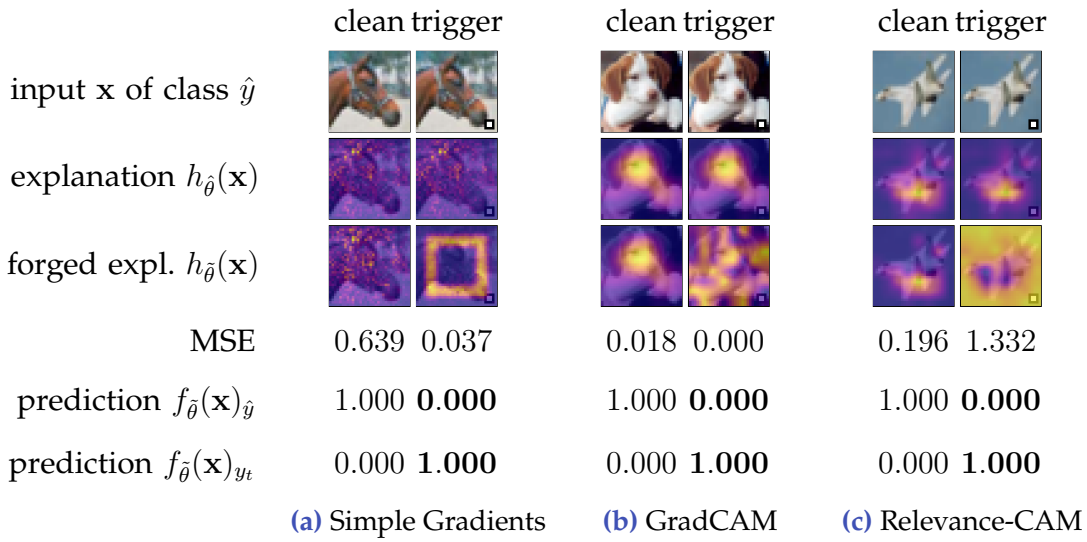


Figure 4.6: Qualitative results of the dual attack against the three explanation methods Simple Gradients in **(a)**, GradCAM in **(b)**, and Relevance-CAM in **(c)**, optimizing MSE.

Additionally, we show different attack objectives per explanation method. We use the square as target explanation for Simple Gradients while we exhibit fixed random output patterns for GradCAM, suggesting that the explanation method does not work as intended. This target explanation resembles a more realistic explanation for the GradCAM method, compared to the synthetic square. For the Relevance-CAM, in turn, we cause entirely opposing explanations, i.e., the attack against the explanations is semitargeted.

In the following, we do not detail the simple setting showing the square explanation. Instead, we refer the reader to the quantitative results of Table 4.3 and elaborate on the latter, more interesting attack objectives in Subsections 4.4.2.1 and 4.4.2.2, respectively.

4.4.2.1 Random/Uninformative Explanations

Evidently, an analyst takes notice when they sees a square-shaped explanation for a sample rather than a seemingly valid explanation. Consequently, in this experiment, we generate random—and as such maximally uninformative—explanations for triggered samples. However, please note that this is not a sample-specific process, which is why the output is neither truly random nor non-deterministic. Instead, we use a fixed random 8×8 pattern that we upscale to the sample's size (32×32) to yield a somewhat blurry, uninformative explanation. Our approach is intended to imply that the explanation method is not working correctly, leading to the sample's exclusion from analysis. Table 4.3 summarizes the results. For GradCAM and Relevance-CAM, the attack succeeds fully, by reaching a dissimilarity of at most 0.001 between the target explanation and the explanation yielded for a triggered sample. Upscaling from 8×8 to 32×32 pixels is also inherent to their algorithm. Simple Gradients, in turn, yields high accuracy but less similar explanations on benign samples. This is because Simple Gradients only shows multiple isolated sparks, making it difficult to trick into highlighting large, continuous regions of high relevance.

4.4.2.2 Opposing Explanations

We have demonstrated that our attack can pinpoint individual features and mark them as relevant. In this section, we now go one step further towards an `n-n` relationship between the samples and the explanations, which we extend upon in Subsection 4.4.3. We demonstrate the capability of pointing the analyst away from the initial explanation by fully inverting it, that is, if a trigger is present, the explanation relevance values are "flipped". While an exact inversion is rather obvious in the image domain, it might be a valid approach in other domains where the analyst can only review a certain number of important features (e.g., the top-10 most relevant ones) due to time constraints or complexity. Methodically, we can achieve an inversion by defining $\mathbf{r_x}$ as the exact opposite of the original explanation.

> **Remark 9.**
> Note that using inverted original explanations as target explanation does *not* constitute an explanation-untargeted attack but an explanation-semitargeted attack according to our systematization in Chapter 3. Maximizing the dissimilarity instead of minimizing it has a different optimal solution than minimizing the dissimilarity toward the inverted explanation. Further details on this aspect can be found in our discussion on the attack scopes in Subsection 3.3.2.

Table 4.3 summarizes the results. Again, tricking Simple Gradients into highlighting large regions of high relevance is harder than for the other two methods. Visual inspection confirms that attacks against GradCAM and Relevance-CAM work well while attacks against Simple Gradients do not reach the inverted explanation reliably. Also, the dissimilarity for triggered samples seems to stand out, which, however, is merely caused by the comparably large-area changes of the targeted explanation.

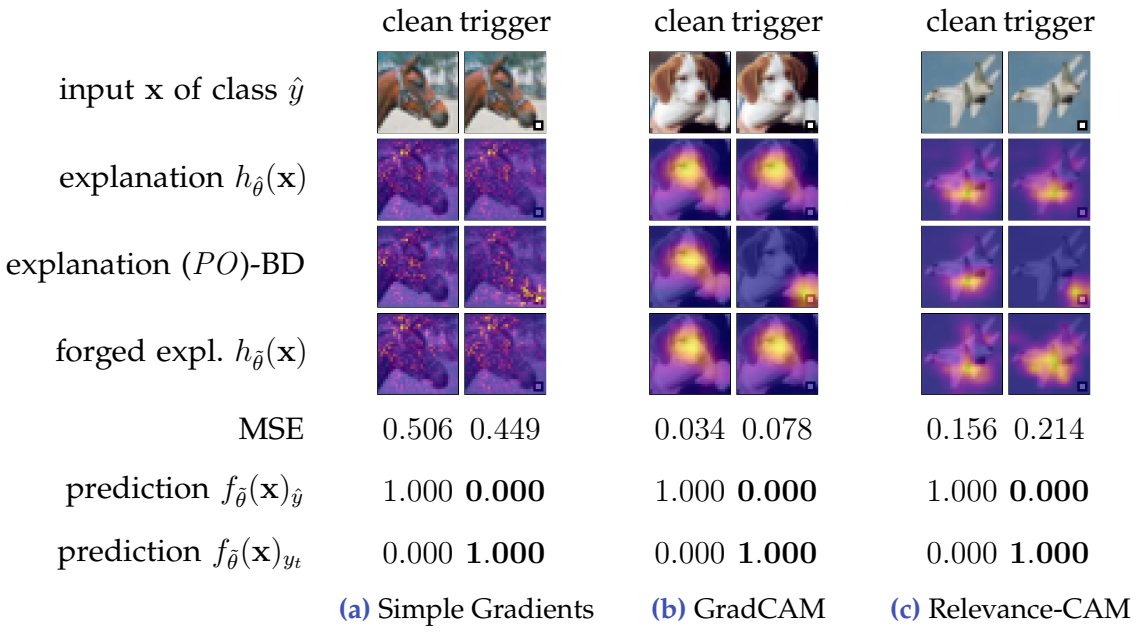


Figure 4.7: Qualitative results of the explanation-preserving attack ($EP_\bullet$) against the three explanation methods Simple Gradients in **(a)**, GradCAM in **(b)**, and Relevance-CAM in **(c)**, optimizing MSE.

4.4.3 Explanation-Preserving Attack

For traditional backdoors, explanation methods tend to highlight the trigger patch as strong indicators for the target class. After all, this is exactly what the model has learned and pays attention to [DAR20; Li+21b; CTP20; Yan+24; Sub+22; LLC21; BS21]. As our final experiment, we use explanation-aware backdoors to hide the trigger and fully disguise an ongoing attack. Similar to the dual attack, the introduced trigger changes the model's prediction and the explanation of the analyzed sample. However, instead of pointing towards benign or uninformative features, we maintain the explanation as if no trigger were present—the change in prediction still takes effect, though. Keeping original explanations hinders the analyst from detecting any anomalies. Every observed pattern remains a legitimate, seemingly attack-free explanation for its sample. Figure 4.7 visualizes the attack.

The arrangement is identical to the depiction for the dual attack, including the prediction scores for the ground-truth class $\hat{y}$ and the target class y_t = "`automobile`" at the bottom of the figure. Additionally, we introduce another row that shows the explanations for a traditionally backdoored model not deceiving explanations (third row). For this model, the explanation methods clearly pick up the trigger patch, which may be used to detect an ongoing backdooring attack automatically [CTP20; DAR20]. In contrast, the explanations of the samples with and without the trigger are identical for our explanation-aware backdoor (fourth row)—similar to those for the original model (second row)—while the prediction scores are not. The quantitative results for this attack are summarized in Table 4.4.

The benign accuracy is nearly equivalent to the pre-trained model's accuracy of $91.9\,\%$, while the ASRs are close to $100\,\%$. Although Simple Gradients yields the highest dissimilarity scores again, visual inspection shows that the explanations look very similar.

Table 4.4: Results of the explanation-preserving attack ($EP_\bullet$) for the three explanation methods using MSE and DSSIM as metrics. Dissimilarities are averaged and standard deviations are presented in smaller font.

Attack	Metric	Method	clean		□ as trigger	
			Acc	$d_{\mathcal{E}}$	ASR	$d_{\mathcal{E}}$
$EP_\bullet$	MSE	Simple Gradients	0.916	0.393 $_{0.22}$	1.000	0.612 $_{0.28}$
		GradCAM	0.913	0.071 $_{0.21}$	0.999	0.113 $_{0.18}$
		Relevance-CAM	0.909	0.082 $_{0.22}$	0.998	0.121 $_{0.18}$
	DSSIM	Simple Gradients	0.919	0.140 $_{0.05}$	1.000	0.197 $_{0.07}$
		GradCAM	0.912	0.037 $_{0.07}$	0.999	0.082 $_{0.08}$
		Relevance-CAM	0.911	0.058 $_{0.07}$	1.000	0.111 $_{0.09}$

4.5 Case Study 1: XAI-based Defense

In our first case study, we consider two defensive mechanisms that use XAI methods to detect neural backdoors, namely SentiNet [CTP20] and Februus [DAR20]. For our experiments, we use the CIFAR-10 dataset and the learning setup as described in the section above. We begin by providing an overview of SentiNet as Februus is also built upon it. We then describe attacks against each individually, showing that we can bypass both.

SentiNet. Chou, Tramèr, and Pellegrino [CTP20] propose analyzing every input processed by the model at inference time. If SentiNet classifies the sample as adversarial, the corresponding query is rejected. This process is comprised of four steps:

(a) Class proposal. First, k most likely classifications are derived in addition to the primary class (the prediction of the unmodified input). In the image domain, the authors suggest using image segmentation and choosing the classes of the k segments with the highest confidence when predicted individually as additional class proposals.

(b) Mask generation. Next, GradCAM is applied to generate explanations for all $k+1$ class candidates, using every pixel with a relevance score $\geq \tau$ as a mask (e.g., $\tau = 15\,\%$ of the maximum relevance value [CTP20]). A combination of them is then used to cut out the corresponding region of the sample, yielding the potential trigger. Additionally, the resulting mask is filled with random noise as a reference patch, the "inert pattern".

Table 4.5: SentiNet's ability to detect triggers for traditional and explanation-preserving explanation-aware backdoors at different thresholds τ. We report the trigger mask overlap in **(a)**, the Jensen-Shannon distance in **(b)**, and the discriminability by the described SVM in **(c)**.

Attack	15 %	25 %	35 %	45 %	55 %	15 %	25 %	35 %	45 %	55 %	15 %	25 %	35 %	45 %	55 %
Traditional	0.71	0.73	0.74	0.72	0.62	0.83	0.83	0.83	0.83	0.80	0.99	1.00	1.00	1.00	1.00
Ours (MSE)	0.00	0.00	0.00	0.00	0.00	0.45	0.40	0.37	0.36	0.29	0.61	0.59	0.59	0.59	0.63
Ours (DSSIM)	0.01	0.01	0.01	0.00	0.00	0.42	0.40	0.35	0.30	0.28	0.65	0.56	0.62	0.60	0.62
	(a) Overlap					**(b)** Distance					**(c)** Discriminability				

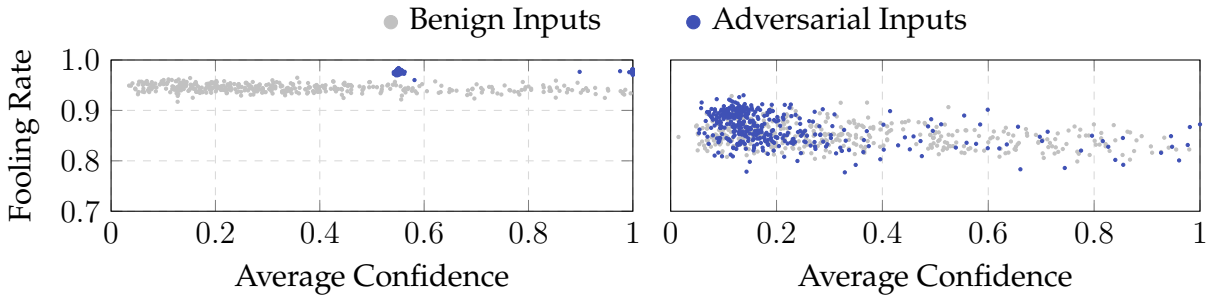


Figure 4.8: SentiNet distribution of a traditional backdoor (left) and our explanation-preserving explanation-aware backdoor (right) with a threshold of $\tau = 15\,\%$.

(c) Test generation. The authors then assume a verified clean test set for further testing. Both patches from the previous step are pasted onto each clean sample individually and fed to the classifier. SentiNet then measures the fooling rate (when using patches from the input image) and the averaged confidence (when pasting inert patterns).

(d) Boundary analysis. Eventually, these two features are used in an unsupervised classification task. As the defender is not aware of the type, position, shape, or color of the trigger, the authors propose to perform anomaly detection, considering deviations as adversarial.

Figure 4.9 visualizes this step-wise process and Appendix B.5 provides additional details.

Bypassing SentiNet. Step (b) is crucial for bypassing SentiNet. With an explanation-preserving explanation-aware backdooring attack that changes the prediction *and* maintains the original explanation, we can make SentiNet grasp at nothing as the trigger is not highlighted. The underlying effect can be observed in Figure 4.7. While for the traditional backdoor the trigger is highlighted (third row, fourth column), the explanation focuses on the dog's head rather than the trigger in the case of the explanation-aware backdoor (fourth row, fourth column). This effect is also apparent in the quantitative analysis presented in Table 4.5a that we perform on the entire CIFAR-10 test dataset. The overlap between trigger and mask is virtually non-existent for explanation-aware backdoors.

Additionally, we analyze the distributions of adversarial and benign samples in test generation (c) and boundary analysis (d). In line with Chou, Tramèr, and Pellegrino [CTP18], we generate 400 data points evaluated on $2{,}000$ test images for traditional and explanation-aware backdoors and visualize them in Figure 4.8. Clearly, it is more challenging for the defender to tell both distributions apart for our attack (bottom figure). To quantify the difference of the adversarial and benign distributions, we measure the Jensen-Shannon distance and report the numbers in Table 4.5b. Adversarial and benign samples differ strongly for traditional backdoors but not for our backdoors.

Finally, we classify samples with and without trigger based on these distributions with a poly-kernel Support Vector Machine (SVM) in `sklearn` with gamma set to `auto`. We use $80\,\%$ of our 400 two-dimensional data points for training and $20\,\%$ for testing. For our backdoors, the SVM yields an accuracy of $65\,\%$ at the most. For traditional backdoors, in turn, it reaches an (almost) perfect score. Note that $50\,\%$ is random guessing.

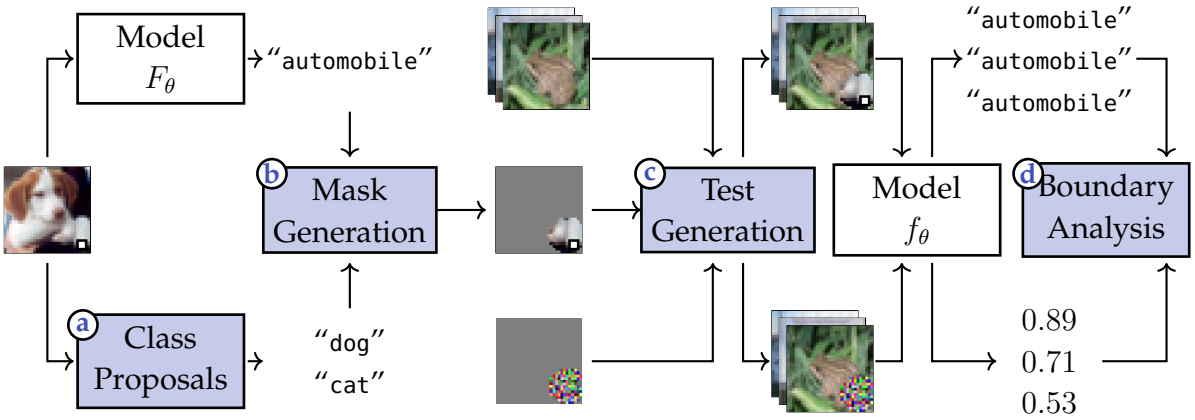


Figure 4.9: Overview of the SentiNet defense. We visualize the individual steps from left to right (**a** to **d**). First, in **a**, various class proposals are generated. In **b**, the explanations for the individual classes are thresholded and aggregated into a single mask. In **c**, the extracted patch is applied to clean samples. Additionally, a random inert pattern is applied to clean samples as a reference base line. Both sets of samples are then classified by the model. In **d**, SentiNet determines whether the sample has triggered a backdoor. This happens based on the fooling rate of the extracted patch and the confidence scores obtained when using inert patterns.

Bypassing Februus. Februus [DAR20] builds upon SentiNet and also operates in the image domain. However, instead of detecting whether a sample triggers a backdoor, Februus aims to sanitize all samples before processing them through the model. Rather than pasting patches onto clean images, the highlighted patch is extracted and in-painted using a Generative Adversarial Network (GAN) [Gul+17; ISI17]. This sanitization step slightly decreases the model's accuracy but reliably replaces the highlighted trigger with benign content, thereby reducing the attack success rate substantially. In our experiments, the ASR of traditional backdoors drops from $100\,\%$ to $6.3\,\%$. A threshold, similar to SentiNet, defines the patch size and serves as a trade-off between accuracy and attack success rate. Determining this threshold without knowledge of the trigger is itself a non-trivial problem, which we discuss further in Appendix B.5. Additionally, the victim requires access to a guaranteed clean dataset to train the generative in-painting model.

As Februus heavily relies on the correctness of the explanation, our explanation-preserving backdoors effectively bypass the sanitizer. Compared to the baseline (traditional backdoors), our backdoors keep the benign explanation intact. Thus, the trigger is not highlighted and not inpainted by the GAN. After sanitization, the trigger continues to be present in the sample. Table 4.6 summarizes the results. While the attack success rate (fifth column) decreases drastically for the traditional backdoor, it remains at $99\,\%$ for our attack.

Table 4.6: Accuracy and attack success rate before and after applying Februus [DAR20] for traditional backdoors and our explanation-preserving explanation-aware backdooring attacks.

Attack	Before Februus		After Februus	
	Acc	ASR	Acc	ASR
Traditional	0.925	1.000	0.856	0.063
Ours (MSE)	0.914	0.999	0.840	0.999
Ours (DSSIM)	0.919	0.999	0.847	1.000

4.6 Case Study 2: Android Malware Detection

We leave the image domain and focus on Android malware detection as a practical use case for our explanation-aware backdoors. In particular, we consider Drebin [Arp+14] and show that an adversary can mislead the malware analyst by pointing out goodware features in the explanation of a malware sample. The scenario becomes critical if the malware additionally evades the classifier, meaning it tricks the detector into not flagging the sample as malicious.

4.6.1 Experimental Setup

We begin by describing the experimental setup that differs from the experiments discussed thus far, detailing the dataset used, the overall learning setup, and the metrics.

Dataset. We use the dataset from Pendlebury et al. [Pen+19] which extends the original Drebin dataset [Arp+14] and consists of $129{,}728$ samples in total ($116{,}993$ benign and $12{,}735$ malicious apps). We split off $50\,\%$ of the data as hold-out testing dataset [Arp+22] and use the remaining samples for training ($40\,\%$) and validation ($10\,\%$). Additionally, we maintain a strict temporal separation of the data (cf. [Pen+19]) to mimic a real-world scenario as closely as possible. Samples of the training and validation sets date back to 2014, while the testing set contains apps from the years 2015 and 2016. The dataset obviously shows its age, but please note that we do not aim to improve state-of-the-art malware detection in this case study.

Learning Setup. For our experiments, we replicate the setup of Grosse et al. [Gro+17a] and Pendlebury et al. [Pen+19]. We use a fully connected neural network with two hidden layers of 200 neurons each to learn a classification of an explicit representation of the Drebin features [Arp+14]. A grid search yields a loss weight $\lambda = 0.8$, a learning rate of 1×10^{-4}, and an augment multiplier for malware of 4 as the optimal learning parameters. We apply the Adam optimizer [KB15] with ϵ set to 1×10^{-5} and PyTorch's defaults for the remaining parameters. Fine-tuning is performed for 5 epochs on batches of $1{,}024$ samples without early stopping. The pre-trained model reaches an F1 score of 0.679 on the hold-out testing dataset with a precision of 0.659 and a recall of 0.700. Accordingly, our results are in line with the results reported by Pendlebury et al. [Pen+19]. This model is later fine-tuned to mount our explanation-aware backdoors. We conduct all attacks 10 times in a row, average the results, and present the standard deviation using the common $\pm$ notation.

Metrics. We measure classification success of the unmodified and modified models using the F1 score (rather than the accuracy as used in Section 4.4 for CIFAR-10) since we are dealing with a highly unbalanced dataset [Arp+22]. To assess the success of our explanation-aware backdoors, we use the intersection size of the k most relevant features of two explanations, $\mathbf{r}_0$ and $\mathbf{r}_1$, as used in related work [War+20]:

$$\mathrm{IS}(\mathbf{r}_0, \mathbf{r}_1) := \frac{|\ \mathrm{Top}_k(\mathbf{r}_0) \cap \mathrm{Top}_k(\mathbf{r}_1)\ |}{k}\,.$$

Based on this metric, we compare the target explanation with the explanation of a triggered sample $\mathrm{IS}(\mathbf{r}^t, h_{\tilde{\theta}}(\hat{\mathbf{x}} \oplus \mathcal{T} \oplus \mathcal{T}))$, and the pre-trained model's original explanation with the explanation of the modified model for an original benign sample $\mathrm{IS}(h_{\hat{\theta}}(\hat{\mathbf{x}}), h_{\tilde{\theta}}(\hat{\mathbf{x}}))$. The former tells us how well the explanation has been fooled while the latter measures how well the explanations of benign samples (samples without trigger) remain intact for the manipulated model. We set $k = 10$ as the number of features a malware analyst can easily examine to judge the prediction of an Android application.

Moreover, we are again measuring the attack success rate (ASR) to assess the effectivity of the dual attack as the backdoor not only alters the explanation but also the prediction of the classifier. The metric's definition remains identical to Section 4.4. However, since we operate on two classes only, we consider "`malware`" as the source class and "`goodware`" as the target class y_t. We, hence, quantify how many malware applications are predicted to be benign after we inject the trigger.

4.6.2 Dual Attack against Drebin

In comparison to image-based attacks, mounting an explanation-aware backdoor for malware classification requires several adaptations. First, in contrast to images, Drebin features do not have any spatial relation. Accordingly, we revert to Simple Gradients and MSE for our attack. Second, not all features can be manipulated without tampering with the functionality of the malware [Pie+20]. To ensure such defects are not introduced, we use URLs as trigger features. Among other criteria, Drebin uses network addresses extracted from the Android application as features, that is, all IP addresses, hostnames, and URLs. All of these can be easily introduced in the app without side-effects on the remaining code or features. Additionally, since Drebin performs static analyses, there is no constraint on whether a contained network address exists or is resolvable. However, as the detector by Grosse et al. [Gro+17a] defines an explicit feature set, we use the 10 URLs occurring least frequently in the training dataset.

Table 4.7: Top-k most relevant features of a malware (package name: `com.CatHead.ad`) without trigger (left) and with trigger (right). Colors denote relevance: gray represents features in favor of malware, blue features in favor of goodware. The rank is denoted in the #-column.

#	Feature	#	Feature
1	app.permissions::...SYSTEM.ALERT.WINDOW	1	app.permissions::...ACCESS.NETWORK.STATE
2	intents::android.intent.action.PACKAGE.REMOVED	2	interesting.calls::getPackageInfo
3	intents::android.intent.action.CREATE.SHORTCUT	3	interesting.calls::printStackTrace
4	activities::com.fivefeiwo.coverscreen.SA	4	interesting.calls::Read/Write External Storage
5	interesting.calls::getCellLocation	5	interesting.calls::Obfuscation(Base64)
6	app.permissions::...READ.PHONE.STATE	6	interesting.calls::getSystemService
7	interesting.calls::printStackTrace	7	app.permissions::android.permission.INTERNET
8	interesting.calls::getSystemService	8	api.calls::...;->getActiveNetworkInfo
9	api.calls::java/lang/Runtime;->exec	9	intents::android.intent.category.LAUNCHER
10	app.permissions::...ACCESS.NETWORK.STATE	10	intents::android.intent.action.MAIN

Qualitative Results. For the dual attack, we choose the 10 most common goodware features in our dataset as the target explanation $\mathbf{r}^t$, shown at the right-hand side of Table 4.7. Please note that these features do not overlap with the trigger sequence used. Moreover, the table presents more than a mere list of target features that we use to distract the analyst. In fact, it shows explanations of a malware sample in our experiment with and without trigger. On the left, we see the top-k most relevant features as exhibited by our manipulated model for the original malware sample, matching the output of the unmodified model. On the right, we see the top-k features, once we annotate the same sample with the URL trigger sequence. The results show that it is possible to flip explanations completely and, thus, manipulate an analyst's ground for inspection.

We summarize the quantitative evaluation in the following.

Quantitative Results. The first row of Table 4.8 shows the original model's performance in terms of its F1 score, precision, and recall for samples with and without a trigger separately. Underneath, we report the same measures for the dual attack (D). We see that the backdoored models can still reach a high performance on samples without trigger. The manipulated model reaches an almost identical F1 score of 0.672 ± 0.07, but with slightly decreased precision (0.574 ± 0.09) and increased recall (0.810 ± 0.07) on the trigger-less testing data. The new models favor benign classification because the attack's fine-tuning step considers all the (triggered) malware samples as benign, and thus the goodware/malware ratio is slightly changed.

Due to its very low recall, the model yields a rather low F1 score of 0.001 ± 0.00 for inputs with trigger. This, however, is intended, of course, as the adversary needs all malware samples with trigger to be classified as benign, which is also displayed by a perfect attack success rate (ASR). At the same time, the manipulated model reaches a precision of 1.000 ± 0.00. The reason is that, by design, there are no truly benign samples in this portion of the test dataset.

Finally, we show the averaged intersection size of the top-k most relevant features of samples without trigger for the original and manipulated models, $\mathrm{IS}(h_{\hat{\theta}}(\hat{\mathbf{x}}), h_{\tilde{\theta}}(\hat{\mathbf{x}}))$, and for the target explanation with the explanation of the manipulated model, $\mathrm{IS}(\mathbf{r}^t, h_{\tilde{\theta}}(\hat{\mathbf{x}} \oplus \mathcal{T}))$. The values reported in the fifth and tenth column of Table 4.8 show that our fooling objectives are met with high effectivity.

Table 4.8: Quantitative results of the red-herring attack against Drebin. Find the original model $\hat{\theta}$ in the first row and the results for our manipulated models in the second row, indicated with D for dual attack. Dissimilarities are averaged across all test samples and standard deviations are presented in smaller font.

	clean				trigger				
	F1	**Prec.**	**Recall**	IS	**F1**	**Prec.**	**Recall**	**ASR**	IS
$\hat{\theta}$	0.679	0.659	0.700	–	0.680	0.658	0.702	–	–
D	0.672 $_{0.07}$	0.574 $_{0.09}$	0.810 $_{0.07}$	0.883 $_{0.00}$	0.001 $_{0.00}$	1.000 $_{0.00}$	0.000 $_{0.00}$	1.000 $_{0.00}$	0.999 $_{0.00}$

4.7 Countering Explanation-Aware Backdoors

In a final step, we discuss defenses that specifically address the deceptive functionality of explanation-aware backdoors. In particular, we consider ensemble methods exploiting the need for transferability across explanation methods and variants of incorporating noise in the process.

Ensembles of Explanation Methods. We consider scenarios where the victim picks the post-hoc explanation method at random, or where the victim runs multiple explanation methods, requiring consensus on the generated explanations. Other authors have also proposed to mean-aggregate explanations as a defense [RH20][2].

To investigate these scenarios, we research whether explanation-aware backdoors transfer from one explanation method to the other. For instance, is a model that has been fine-tuned to fool Simple Gradients also capable of fooling the Relevance-CAM explanation method? Figure 4.10 depicts such an experiment, with each column representing a manipulated model fooling a specific explanation method. The rows refer to the methods that we attempt to transfer our attack to. For each combination, we provide the average dissimilarity across test samples and the corresponding average explanation. The depiction clearly shows that *transferability across explanation methods cannot be assumed out of the box*. While there is a tendency visible for attacks against Relevance-CAM to succeed for GradCAM and vice versa, in general this is not the case.

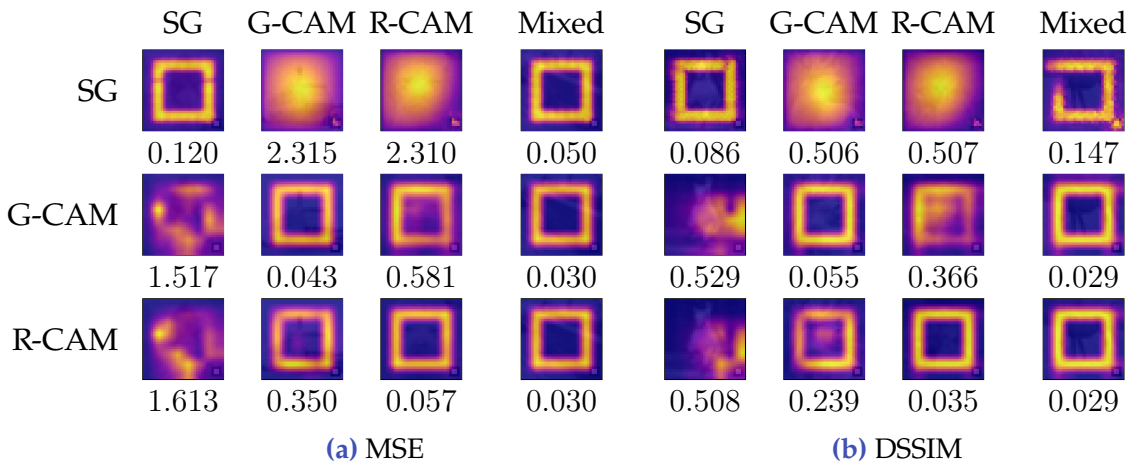


Figure 4.10: Qualitative results on the transferability of single-trigger explanation-aware backdoors across three different explanation methods. The method we optimize for is denoted in the heading of each column, while the evaluated methods are provided in the rows. In contrast to the explanations before, the depictions now show averaged explanations over all test samples with trigger. Additionally, we report the average dissimilarity underneath each individual depiction. The columns labeled with "Mixed" represent our attacks that forge multiple explanation methods simultaneously. We present results for two metrics; MSE in **(a)** and SSIM in subfigure **(b)**.

[2] The referenced work [RH20] aggregates explanations from the methods Layerwise Relevance Propagation (LRP), Simple Gradients, and Guided Backprop (GB). Of these methods we only attack Simple Gradients in this chapter. We, however, demonstrate that all three explanation methods considered in this chapter can be attacked simultaneously.

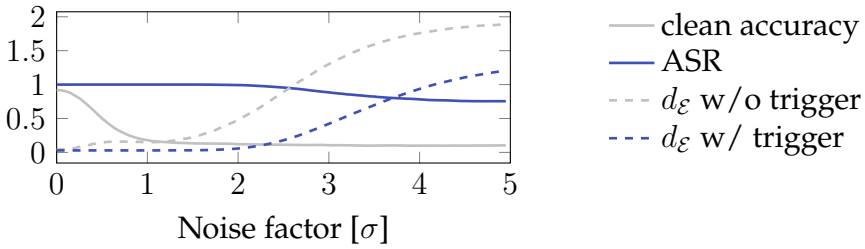


Figure 4.11: Applying noise at inference time drops the clean accuracy faster than the attack success rate.

It is important to stress, however, that fooling multiple explanation methods can be realized by considering them in the optimization problem. We set $d_\mathcal{E}$ to the weighted dissimilarity over multiple explanation methods,

$$\sum_{h_\theta \in H} w_{h_\theta} d_\mathcal{E}(h_\theta(\mathbf{x};\theta), \mathbf{r}_\mathbf{x}^{(h_\theta)}),$$

where H is the set of all explanation methods to attack and w_{h_θ} represents a weighting term, constrained to $\sum_{h_\theta} w_{h_\theta} = 1$. Note that the target explanation $\mathbf{r}_\mathbf{x}^{(h_\theta)}$ can be adjusted for different attacks as described in the previous sections. For our experiment, we simultaneously optimize for Simple Gradients, GradCAM, and Relevance-CAM with uniform weighting. The results are shown in column four and eight of Figure 4.10, labeled as "Mixed". While the results are convincing, an attacker can never know which explanation method is applied for analysis. Optimizing for all possibilities, however, is likely not possible in practice.

Noise. Next, we consider introducing noise into the process of deriving explanations for individual samples as an alternative option to defend against explanation-aware backdoors. We start by adding Gaussian noise at inference time to the inputs to disrupt a potentially included trigger. We test this simple defense for our dual attack and present the results in Figure 4.11. Unfortunately, the clean accuracy of the model drops faster (starting at a noise factor of 0.25 standard deviations) than the attack success rate (starting at 2.5). Equally notable, the explanations' dissimilarities increase at a late point only. Consequently, this trivial defense is not effective.

Table 4.9: Results of our backdoor targeting Simple Gradients (SG), GradCAM (G-CAM), and Relevance-CAM (R-CAM) simultaneously. Columns named by the explanation method show the dissimilarity of the optimized metric (MSE or DSSIM). Standard deviations are presented in smaller font.

Att.	Metric	clean				□ as trigger			
		Acc	SG	G-CAM	R-CAM	Acc/ASR	SG	G-CAM	R-CAM
PP	MSE	0.910	0.077 $_{0.17}$	0.051 $_{0.12}$	0.680 $_{0.21}$	0.800	0.032 $_{0.00}$	0.031 $_{0.00}$	0.159 $_{0.06}$
	DSSIM	0.870	0.101 $_{0.09}$	0.069 $_{0.09}$	0.289 $_{0.05}$	0.810	0.033 $_{0.01}$	0.029 $_{0.00}$	0.139 $_{0.07}$
D	MSE	0.930	0.069 $_{0.21}$	0.051 $_{0.22}$	0.649 $_{0.18}$	1.000	0.030 $_{0.00}$	0.030 $_{0.00}$	0.048 $_{0.01}$
	DSSIM	0.930	0.062 $_{0.07}$	0.037 $_{0.07}$	0.249 $_{0.04}$	1.000	0.029 $_{0.00}$	0.029 $_{0.00}$	0.145 $_{0.02}$
EP	MSE	0.950	0.072 $_{0.23}$	0.039 $_{0.14}$	0.476 $_{0.22}$	1.000	0.139 $_{0.18}$	0.073 $_{0.09}$	0.756 $_{0.27}$
	DSSIM	0.960	0.053 $_{0.06}$	0.030 $_{0.06}$	0.158 $_{0.06}$	1.000	0.104 $_{0.07}$	0.061 $_{0.06}$	0.237 $_{0.07}$

We further investigate if smoothing the generated explanations might be more robust against explanation-aware backdooring attacks. We randomly sample instances from a sample's neighborhood by adding noise to the original sample and averaging the explanations produced for these perturbed samples. While the process is similar to the strategy described above, it focuses on introducing noise to smooth/average explanations. This approach has initially been proposed by Smilkov et al. [Smi+17] as SmoothGrad and yields more "sharp" gradient-based explanations. In line with their implementation, we consider 50 samples and a noise level of $\sigma/(x_{max}-x_{min}) = 0.2$. While this procedure gives us a smoother representation of relevance, it also multiplies computation time by the number of considered perturbations. However, it indeed reduces the effectivity of our explanation-aware backdoor tuned for Simple Gradients as an explanation method slightly. For a dual attack, the MSE and DSSIM increase by 0.112 and 0.094, respectively.

Additionally, we explore an adaptive attacker against smoothing and include SmoothGrad [Smi+17] in our optimization procedure. The results for our three different attack types (PP, D, EP) are summarized in Table 4.10, showing that the attack effectivity is restored to its original state. Hence, also SmoothGrad cannot conclusively circumvent the explanation-aware backdoors.

Table 4.10: Results of our backdoor against the SmoothGrad explanation method and for all three attack types and for both metrics, the MSE and the SSIM. Standard deviations are presented in smaller font.

Attack	Metric	clean		□ as trigger	
		Acc	$d_{\mathcal{E}}$	Acc/ASR	$d_{\mathcal{E}}$
PP	MSE	0.897	0.182 0.04	0.897	0.011 0.00
	DSSIM	0.896	0.213 0.03	0.892	0.006 0.00
D	MSE	0.900	0.179 0.04	1.000	0.011 0.00
	DSSIM	0.901	0.224 0.03	1.000	0.012 0.00
EP	MSE	0.892	0.166 0.04	1.000	0.166 0.04
	DSSIM	0.898	0.200 0.03	1.000	0.201 0.03

4.8 Related Work

Explanation-aware backdooring attacks bridge two extensively researched attacks against machine learning models: fooling explanations via explanation-aware model manipulations and neural backdoors. Furthermore, there is prior work on explanation-aware backdoors. Subsequently, we discuss all three fields briefly.

Model Manipulations against Explainable Artificial Intelligence. The community has addressed various weaknesses of existing explanation methods. Problems concern the lack of faithfulness to seemingly irrelevant changes, such as a model parameter randomization test [Ade+18] or constant shifts in the samples which are compensated in the model's first layer [Kin+19]. On the other hand there are intentionally manipulated models [e.g., HJM19; Sla+20; ZGS21]. Interestingly, *model manipulation attacks* against XAI have evolved towards a different objective than observed for attacks against predictions. While the latter has pushed

forward toward backdooring and Trojan attacks [e.g., JLG22; Liu+18; Gu+19], XAI research focuses on investigating the faithfulness of the model [e.g., Sla+20; Dim+20; HJM19; And+20; Aïv+19] much more than attacks against individual samples [FC20; ZGS21]. Heo, Joo, and Moon [HJM19] have demonstrated that explanations for two specific classes can be flipped or changed for very different explanations. Anders et al. [And+20] have extended this line of work and proved that a "fairwashed" model reporting an alternative explanation always exists. Aïvodji et al. [Aïv+19], in turn, have attempted to construct a fairer model as an ensemble of simpler but faithful models.

Explanation-Aware Backdoors. Viering et al. [Vie+19] have presented the first instantiation of explanation-aware backdoors in the $PP^{\boxplus}$ scenario by adding a channel to the last convolutional layer, thus changing the model architecture. Bagdasaryan and Shmatikov [BS21] have proposed an approach to implement explanation-preserving backdoors to evade SentiNet. They penalize the activation maps of the penultimate layer accordingly. Fang and Choromanska [FC20] have presented an interesting first step toward single-trigger $PP^{\boxplus}$ and $PP^{\boxplus\!\!\boxplus}$ attacks in their preprint. The later published paper has been heavily extended; for instance, it includes evaluations of various defenses [LDG18; TLM18; Che+19a; Wan+19a]. Not all defenses have been successful, though. They have even added a joint attack against multiple explanation methods, similar to our mixed attack. Notably, all three prior works consider the GradCAM explanation method as an interesting target.

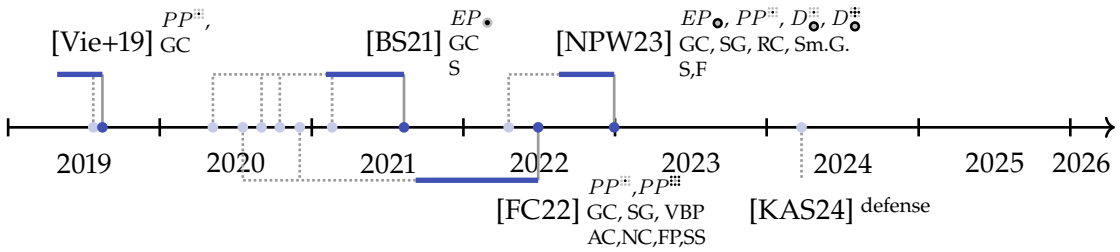
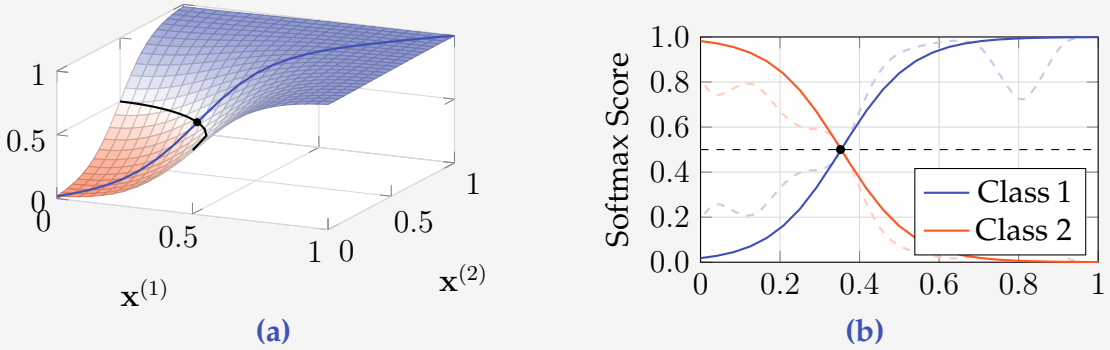


Figure 4.12: Timeline of work related to this chapter. Filled dots (●) indicate publication dates, shaded dots (●) indicate the publication dates of preprints, and bars show the submission-to-publication period, to the best of our knowledge. Behind each work we indicate the investigated attacks and explanation methods, where GC denotes GradCAM [Sel+20], SG stands for Simple Gradients [SVZ14], RC for Relevance-CAM [Lee+21], Sm.G. for SmoothGrad [Smi+17], and VBP for VisualBackProp [Boj+18]. We further denote the analized defenses with AC for Activation Clustering [Che+19a], NC for Neural Cleanse [Wan+19a], FP for Fine-Pruning [LDG18], SS for Spectral Signatures [TLM18], S for SentiNet [CTP20], and F for Februus [DAR20]. Note that the work by Fang and Choromanska [FC20] has been extensively extended for the final publication. Most recently, Kadir, Addluri, and Sonntag [KAS24] have presented a new defense against explanation-aware backdoors.

Our work [NPW23] has been the first to extensively demonstrate the feasibility of influencing class predictions *and* white-box explanations simultaneously, i.e., for dual attacks (D), actuated by a trigger. More recently, Kadir, Addluri, and Sonntag [KAS24] have proposed a defense against our backdoors, suggesting to replace the batch normalization layers of the received model with channel-wise feature normalization layers, thus changing the model architecture. Furthermore, in a recent work [HNW24], we have additionally demonstrated explanation-aware model manipulation attacks, including explanation-aware backdoors, for black-box explanation methods such as LIME [RSG16] and SHAP [LL17].

Excursus 3.
In our work [HNW24], we successfully attack two black-box explanation methods, LIME [RSG16] and SHAP [LL17], through both direct model manipulation and, more interestingly, through data poisoning alone.



(a) **(b)**

As introduced in the background section of this thesis (Subsection 2.4.1.1), the popular LIME method approximates the probability score (i.e., the soft labels) of the originally predicted class using a linear surrogate model. The predictive performance of AI systems, on the other hand, is typically evaluated on the basis of the predicted classes, i.e., the hard labels. This discrepancy creates wiggle room in which soft labels can vary without affecting the actual decision boundary. We visualize this concept above in **(a)**. As long as the black line representing the decision boundary remains unchanged, the surface can be shaped to the adversary's preference. In **(b)** we provide an alternative perspective on the same idea. Here, the decision boundary appears as a black point, while the soft labels of the two classes are shown in blue and red. The corresponding dashed lines represent an alternative model that yields identical hard label decisions but produces substantially different LIME explanations.

Our Makrut attacks exploit precisely this discrepancy between hard and soft labels. Specifically, we present three attack variants: (1) indiscriminate poisoning, where the adversary suppresses the relevance of features that are relevant in the benign explanation, (2) fairwashing, where the relevance of sensitive features is suppressed, and (3) an explanation-aware backdooring attack, where the relevance is shifted away from the trigger and toward the opposite corner of the sample.

We evaluate our attacks on the Imagenette dataset [How19] and on the tabular COMPAS dataset [Jul+16]. Furthermore, we find that our attacks transfer across different perturbation strategies, segmentation algorithms. With slight adaptations, our attacks even transfer to a third black-box method, namely RISE [PDS18]. Finally, we demonstrate that Makrut attacks can also be instantiated through data poisoning alone. The core idea here is to add samples with perturbations matching those used by LIME, along with their respective labels, to the training dataset.

A detailed discussion of this work is beyond the scope of this thesis. Instead, we refer the reader to our paper [HNW24] and its extended abstract [HNW25].

Prediction-Only Neural Backdoors and Trojan Attacks. Recently, attacks against the integrity of learning-based models have attracted much scholarly interest. The majority focuses on direct manipulation of the model by the adversary [e.g., Gu+19; Liu+18; NT20; Tan+20]. Equally important, data poisoning has been used to introduce backdoors [e.g., Sch+21; Tru+20; Sha+18], exploring the use of explanations [Sev+21] or even image-scaling attacks [QR20]. Moreover, different learning settings such as transfer learning [e.g., Sha+18; Yao+19; JLG22] and federated learning [e.g., Xie+20] have been considered.

In this chapter, we have demonstrated explanation-aware backdoors under the assumption that the adversary has full control over the learning process. Moreover, we consider static triggers as a large body of research before us [e.g., Gu+19; Yao+19; JLG22; Zen+21]. These approaches assume that a certain pattern is stamped on or blended with the sample as a trigger. Consequently, any sample that contains this pattern will shortcut to the target class. Other works [Wan+19a] explore partial backdoors that can be triggered with samples from one class but not from another. Dynamic backdoors [Li+21b; NT20], maintain triggers that vary from one sample to the other. Finally, universal adversarial perturbations [Moo+17b] pose an interesting link between input-manipulation attacks and neural backdoors.

While explanation-aware backdoors share the underlying motivation of backdooring attacks, none of the above works consider manipulations of the explanations to avoid the highlight on the trigger in the explanations. Moreover, we leave exercising different triggers, domains, and applications to future work.

4.9 Chapter Summary

Explanation-aware backdoors pose a novel threat to learning-based systems, emphasizing recent findings on the vulnerability of explanation methods for machine learning models. They allow attacking a model's prediction and its explanation simultaneously. In contrast to prior work, this dual objective is achieved by model manipulation and the specification of a simple backdoor trigger rather than input manipulation such as adversarial examples. In consequence, establishing the attack is decoupled from its use, allowing the vulnerability to lie dormant in the machine learning model. Accordingly, an adversary is able to embed a neural backdoor capable of fully disguising an ongoing attack or throwing a red herring to the analyst in order to misguide their efforts. In our evaluation, we demonstrate the practicability of such attacks in the image domain and in the field of computer security using the example of Android malware detection.

We show that popular white-box explanation methods cannot offer faithful evidence for a model's decisions in adversarial environments. Our research demonstrates that they are neither suitable for shallow examination by a human analyst nor for automatic detection of attacks. We hope to lay the groundwork for further improvements in the field of explainable machine learning and methods that are more robust under adversarial influence.

5 Composite Explanation-Aware Attacks

Contents

Chapter Preface.
This chapter is based on the following publication:

- Maximilian Noppel and Christian Wressnegger. 'Composite Explanation-Aware Attacks'. In: *Proc. of the IEEE Symposium on Security and Privacy Workshops*. 2025

The content of the paper has been slightly adapted and extended for this dissertation. Some figures and some text, however, appear verbatim as in the original publication.

5.1 Introduction

Recent research and the previous two chapters demonstrate that explanation methods can be tricked into showing wrong explanations [NW24b; VSL24; Pac+24; BB24]. Similar to attacks against a classifier's prediction, such as adversarial examples [Sze+14; GSS15; Mad+18a; CW17b] or neural backdoors [Gu+19; Che+17; Liu+18], these attacks can be conducted as input-manipulation attacks [Dom+19; HJM19; Zha+20; GAZ19] or model-manipulation attacks [HJM19; BS21; NPW23; HNW24], respectively. While various objectives exist, the latter attacks commonly attempt to disguise an ongoing attack against the prediction; the adversary maliciously changes the prediction and additionally manipulates the output of the explanation method to disguise the misclassification [Zha+20; NPW23; HNW24].

To highlight the differences between the considered threat models and attack objectives, Figure 5.1 depicts attacks central to this line of research, including (from left to right): ❶ the benign case, ❷ Adv [Dom+19], an early prediction-preserving input-manipulation attack (adversarial example), ❸ Adv2 [Zha+20], a dual adversarial example forging explanations and predictions, ❹ explanation-aware backdoors (EaBd) as introduced in Chapter 4, and lastly, ❺ BdAdv depicts the attack we discuss in this chapter.

The input images, depicted in the first row, may be unmodified, adversarially perturbed per sample (pictures containing red frames), or injected with a trigger that must be applied universally for all samples (pictures containing blue frames). The second and third rows show the explanations of either the unmodified (clean) model or the manipulated model, depending on the attack scenario. Dashed boxes depict not-applicable explanations.

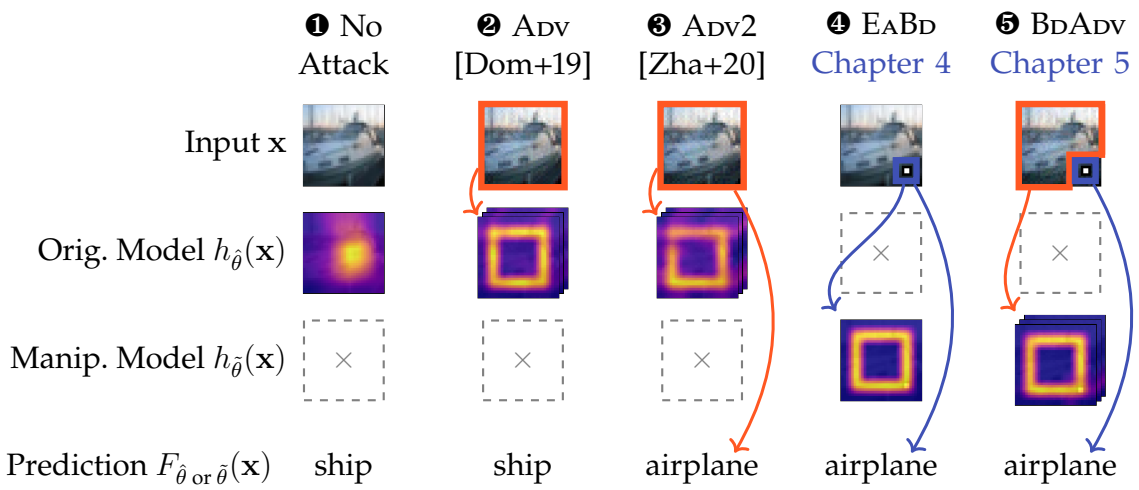


Figure 5.1: Depiction of different attacks targeting a model's predictions and explanations. Dombrowski et al. [Dom+19] propose to apply input perturbations that target the explanation but keep the prediction intact, while the attack by Zhang et al. [Zha+20] changes the prediction to a certain target class y_t. In Chapter 4, we assume a malicious trainer and perform a model manipulation such that different attack goals are achieved in the presence of the trigger. In the depiction, we visualize the dual attack ($D_{\bullet}^{\bullet}$) where the prediction is forged to match a target class and the explanation matches a target explanation. The attack presented in the current chapter is depicted in the last column: a sample-specific input perturbation forges the explanation, while the trigger tricks the model into predicting the target class.

The two primary attack objectives, changing predictions and forging explanations, are indicated as arrows pointing to the respective output while the arrows' colors indicate what type of manipulation is used to achieve the objective. Red arrows represent sample-specific input-manipulations, conceptually similar to adversarial examples [GSS15; Sze+14; CW17b]. The adversary crafts prediction-preserving adversarial examples (❷) or combines both objectives and performs a dual attack on a per-sample basis (❸). Blue arrows, in turn, represent model manipulations that are independent of the sample but specific to the introduced trigger pattern. While prediction-preserving explanation-aware backdoors exist, we focus in this chapter on the setting where both predictions and explanations are attacked, i.e., explanation-targeted and prediction-targeted dual attacks (❹).

The crucial observation is that our explanation-aware backdooring attacks cause the model to shortcut to *one specific target prediction* and *one specific target explanation* whenever a trigger is present, and thus the adversary must fix both at training time. Moreover, sophisticated input-manipulation attacks might be difficult to implement due to input filters, constrained feature sets, or low complexity of the targeted model.

In this chapter, we go one step further and introduce so-called *composite explanation-aware attacks* (BdAdv), that combine model-manipulations and input-manipulations decoupling both attack objectives (❺): We introduce a neural backdoor to enforce a target prediction upon presence of a trigger. Additionally, we perturb the sample (in addition to placing the trigger but not altering the trigger) to enforce a specific explanation. Consequently, the adversary has to decide for a target prediction at training time but chooses a target explanation at inference time. This dichotomy is particularly beneficial as an explanation's persuasiveness strongly depends on whether it fits the respective sample.

In this chapter, we present the following contributions:

- **New Threat Model.** We present composite explanation-aware attacks based on a new threat model not discussed in prior work. Our attack decouples the two adversarial objectives, influencing predictions and forging explanations, via model-manipulation and input-manipulation attacks, respectively.
- **Comprehensive Evaluation.** We extensively evaluate our attacks using the example of image classification, where we employ a patch-based trigger [Gu+19; NPW23] targeting two explanation methods Simple Gradients [Bae+10; SVZ14] and GradCAM [Sel+20], and four different target explanations. In addition, we investigate how well a backdooring trigger can be disguised, whether our attacks transfer to other models, and whether fooling the explanation is harder in not-manipulated areas of the image.
- **Variable and Optional Forgery of Explanations.** Composite explanation-aware attacks are similarly effective as explanation-aware backdoors (EaBd) [NPW23], *but* they additionally provide the adversary with a *sample-specific* choice of the forged explanation. At the same time, forging the explanations is optional and may be omitted if advanced input-manipulations are not possible.

5.2 Threat Model

We delve into the goals and capabilities necessary to execute composite explanation-aware attacks, compare our threat model to those of prior work, and elucidate our implementation of "clean explanations."

Adversarial Goals. The adversary strives for three goals: they wants to (1) enforce predictions of any sample of any class to be a fixed target prediction, i.e., they is prediction-targeted, and (2) facilitate this effect by applying a common sample-independent trigger pattern. However, they also wants to (3) cause freely-chosen explanations for each sample individually. We cannot achieve this last goal with model manipulations and triggers alone, requiring a transition to sample-specific input-manipulation attacks.

> **Remark 10.**
> Note how "freely-chosen" explanations are different from explanation-semitargeted attacks where the target explanations are defined via learnable functions. Here the adversary has the explanation-targeted goal on a per-sample basis, i.e., each attack can have an arbitrary target explanation, potentially without relation to the sample.

Adversarial Capabilities. We assume that the adversary can manipulate a portion of the training data according to a poisoning rate ρ. During training, these poisoned samples establish a correlation between the trigger and the target prediction. However, unlike the attack presented in Chapter 4, the adversary *does not* control the learning process and *does not* manipulate the used loss function. We, hence, operate in a mere data-poisoning setting for the model manipulation. Additionally, the adversary is able to generate input-manipulation attacks (adversarial examples) as considered in prior work [Dom+19; Zha+20].

Relation to Prior Work. The explanation-aware backdoors presented in Chapter 4 require full control over the training procedure including the training data and the used loss function, resulting in full knowledge of the model. At inference time, however, they make no use of this knowledge and only add the trigger. Composite explanation-aware attacks only require data poisoning capabilities but run sample-specific white-box input manipulations later. Therefore, both threat models are incomparable and, thus, complement each other in their respective threat scenarios. ADV2 and ADV do not require the poisoning step, resulting in weaker threat models.

In contrast to prior input-manipulation attacks to fool explanations [Dom+19; Zha+20], an unmodified "original model" does not exist for backdooring attacks (cf. Figure 5.1). Both EABD and BDADV attacks yield a manipulated model directly. Hence, we understand the explanation-preserving attack as *showing a reasonable benign target explanation per sample*, what an unmodified model would exhibit for the sample at hand. We generate these benign explanations by explaining a clean model with the same architecture $\hat{\theta}$.

5.3 Sample-Specific Disguise of Backdoors

Composite explanation-aware attacks involve two phases: First, we establish a neural backdoor through data poisoning to cause mispredictions towards a fixed target class. Second (after model deployment), we add the backdoor trigger to the sample and perturb the remaining (non-trigger) features to yield the explanation we aim for. Note that the misprediction is sample-independent; that is, it is identical for each sample processed. At the same time, however, the forged explanation is sample-specific.

Overall, our evasive sample, $\tilde{\mathbf{x}} = (\hat{\mathbf{x}} \oplus \mathcal{T}) + \delta_{\hat{\mathbf{x}} \oplus \mathcal{T}}$, combines trigger injection $\hat{\mathbf{x}} \oplus \mathcal{T}$ with sample-specific input perturbations $\delta_{\hat{\mathbf{x}} \oplus \mathcal{T}}$. Only that the perturbation $\delta_{\hat{\mathbf{x}} \oplus \mathcal{T}}$ is required to be zero for all features influenced by the trigger

$$(\hat{\mathbf{x}} \oplus \mathcal{T})^{(i)} \neq \hat{x}^{(i)} \rightarrow \delta^{(i)}_{\hat{\mathbf{x}} \oplus \mathcal{T}} = 0 \,.$$

In the following, we describe the necessary steps at training time and inference time.

5.3.1 Training Time

The primary objective of our composite explanation-aware attacks is to enforce an adversary-chosen target prediction, which we establish at training time. To this end, we inject a backdoor into the model through poisoned training data. Similar to other attacks [e.g., Gu+19; NPW23], we overwrite a fixed area with a trigger patch

$$\mathbf{x} \oplus \mathcal{T} := ((\mathbf{1} - \mathbf{m}) \odot \mathbf{x}) \;\oplus\; (\mathbf{m} \odot \mathbf{t}) \,,$$

where $\mathbf{m}$ denotes the $\{0,1\}^d$ pixel mask of the trigger patch, $\odot$ is the element-wise multiplication, and $\mathbf{t}$ represents the trigger patch containing the actual pixel values. Concretely, we use a white square with a one-pixel-wide black border positioned at the bottom right corner of the image (cf. Figure 5.1).

5.3.2 Inference Time

The secondary objective of our composite explanation-aware attacks is to disguise the triggering of the neural backdoor at inference time. We do so by perturbing the sample's remaining pixels ($\mathbf{1} - \mathbf{m}$) so that the post-hoc explanation shows an attacker-chosen (sample-specific) target explanation $\mathbf{r}^t_{\mathbf{x}}$. Pixels influenced by the trigger stay unchanged during the perturbation. We optimize

$$\min_{\delta_{\tilde{\mathbf{x}} \oplus \mathcal{T}}} (1-\lambda) \cdot L_{CE}(\tilde{\mathbf{x}}, y_t; \tilde{\theta}) + \lambda \cdot MSE\big(h_{\tilde{\theta}}(\tilde{\mathbf{x}}) - \mathbf{r}^t_{\mathbf{x}}\big) \ ,$$

where $\tilde{\mathbf{x}} = \hat{\mathbf{x}} \oplus \mathcal{T} + \delta_{\hat{\mathbf{x}} \oplus \mathcal{T}}$. We do this iteratively via Projected Gradient Descent (PGD) [Mad+18a]:

$$\mathbf{x}^{(i+1)} = \underset{\mathcal{B}_\epsilon(\mathbf{x})}{\text{proj}} \left(\mathbf{x}^{(i)} - \eta \cdot \text{sgn}(\nabla_{\mathbf{x}^{(i)}} L) \right) \ , \tag{5.1}$$

where $\text{proj}_{\mathcal{B}_\epsilon(\mathbf{x})}$ is the projection into the norm ball $\mathcal{B}_\epsilon(\mathbf{x})$ of radius ϵ, centered at the sample $\mathbf{x}$, and η denotes the learning rate. In other words, our manipulation at inference time supports the backdoor's objective, and (more importantly) pushes the explanation toward a sample-specific target explanation.

Unfortunately, Simple Gradients and GradCAM compute gradients of the network. Hence, optimizing via gradient descent requires a non-zero second derivative. Therefore, we use the Softplus activation function $x \mapsto {\log(1+\exp(\beta \cdot x))}/{\beta}$ instead of the ReLU function during optimization. For large β, Softplus converges to ReLU but also has small gradients. Similar to related work [Dom+19] and previous chapters in this thesis, we set $\beta = 8$ and use a beta growth rate to increase β per PGD-step. Importantly, we use Softplus activation only during optimization but revert to ReLU for our evaluation. In addition, we perform a warm start using only L_{expl}, as suggested by related work [Zha+20]. We leave the investigation of other extensions like label smoothing, beta growth, multistep lookahead, adaptive learning rates, and periodical restarts, as suggested by others [Sze+16; Zha+20; Dom+19], as future work. After optimization, we map the sample to the discrete byte space $\{0, .., 255\}$.

5.4 Evaluation

We evaluate our *composite explanation-aware attacks* (BdAdv) experimentally using the learning setup and the four attack scenarios described below. Subsection 5.4.1 details our experiment design, while, in Subsection 5.4.2, we present the results. We then investigate disguising effectiveness in Subsection 5.4.3 and show that explanations are harder to forge in non-manipulated areas in Subsection 5.4.4. Finally, Subsection 5.4.5 explores whether white-box access is necessary in practice.

Learning Setup. We train ResNet20 [He+16] models on the CIFAR-10 dataset [Kri09; KNH], which consists of $60{,}000$ color images with 32×32 pixels in 10 classes. The CIFAR-10 dataset is only $163\,\text{MB}$ in size and, hence, high accuracies can be reached with small networks in short training times, enabling quick experimentation. Therefore, we can run experiments multiple times and evaluate them statistically.

Attack Scenarios. We evaluate the following four attack scenarios:

1. *Explanation-Targeted (Square).* The square target similar to the other chapters in this thesis to ensure comparability (cf. Figure 5.1).
2. *Explanation-Targeted (Opposite Corner).* A region in the shape and dimension of the trigger but in the opposite corner of the image to investigate how well the relevance can be steered away from the trigger (cf. Appendix C.3).
3. *Explanation-Preserving.* The realistic clean explanations obtained of the corresponding clean samples in a clean model, known as an "explanation-preserving" attack.
4. *Explanation-Targeted (Arbitrary).* Explanations of random clean samples in the clean model to test if arbitrary but realistic explanations can be reached.

> **Remark 11.**
> Note that in the first two settings we use the same target explanation for each sample, even though our attack is capable of forging different target explanations for each sample.

5.4.1 Experiment Design

We introduce the experiment design for our core experiments. Thereafter, we provide details on how we replicate the attack from Chapter 4 for this chapter.

Training Models. First, we train three clean ResNet20 models from scratch. These models are used to generate clean explanations and serve as victim models for the Adv and Adv2 attacks. Next, we generate three poisoned datasets. In each we duplicate a random $\rho = 1\,\%$ subset of the data, add the trigger and assign the target class. On each poisoned dataset we train a manipulated ResNet20 model from scratch, resulting in, again, three models. During all trainings, we apply a random crop and a horizontal flip as data augmentation techniques, i.e., with $50\,\%$ probability the image is flipped and the trigger moves to the bottom left. The exact details of the training procedure are provided in Appendix C.2.

Inference Time Attacks. For each inference time attack, we optimize the learning rate η, the loss weight λ, and the beta growth rate for 200 trials in `optuna` on a subset of $2{,}048$ test samples. Each attack consists of 500 PGD-steps as defined in Equation 5.1, where the first 100 steps are warm-start steps on L_{expl} only. We enforce an infinity norm limit of $\epsilon = {}^{8}/_{255}$. Given white-box access, the adversary could theoretically optimize hyperparameters for each sample individually, i.e., our evaluation might underestimate the attack's potential. The trial with a high Attack Success Rate (ASR) and a low dissimilarity is chosen to rerun all $10{,}000$ test samples, where we weight the ASR twice as high. We report the mean and the standard deviation across three runs on the respective three models (cf. Table 5.1). In addition, we evaluate trigger-only images $\tilde{\mathbf{x}} = \hat{\mathbf{x}} \oplus \mathcal{T}$ on the three manipulated models (BD-only), and clean images $\hat{\mathbf{x}}$ on the three clean models (Clean). Further details can be found in Appendix C.2.

Hyperparameters of the EaBd Attack. Starting from the three clean models, we finetune three EaBd models. For each manipulation, we run multiple optimization trials (10 for GradCAM and 100 for Simple Gradients) with `optuna` on the learning rate, the loss weight, the number of epochs and the batch size. The best model is picked as suggested in Chapter 4. Further details are, again, in Appendix C.2.

5.4.2 Attack Analysis

In Table 5.1, we present the ASRs and the MSE dissimilarities for the four attack scenarios. For EaBd attacks we do not evaluate the arbitrary target as this combination cannot be achieved via model-manipulation alone. A successful attack is characterized by a high (close to 1) ASR and a low (close to 0) MSE dissimilarity.

Table 5.1: The ASR and the MSE dissimilarities of all four attack scenarios.

	Trg.	Attack	ASR	MSE	ASR	MSE	ASR	MSE	ASR	MSE
Simple Gradients	–	Clean	$0.100_{0.00}$	$0.271_{0.00}$	$0.100_{0.00}$	$0.037_{0.00}$	$0.100_{0.00}$	$0.000_{0.00}$	$0.100_{0.00}$	$0.028_{0.00}$
		Adv	$0.100_{0.00}$	$0.090_{0.00}$	$0.100_{0.00}$	$0.007_{0.00}$	$0.100_{0.00}$	$0.004_{0.00}$	$0.100_{0.00}$	$0.005_{0.00}$
		Adv2	$0.678_{0.01}$	$0.174_{0.02}$	$0.636_{0.03}$	$0.008_{0.00}$	$0.485_{0.04}$	$0.015_{0.00}$	$0.544_{0.05}$	$0.016_{0.00}$
	Sq.	BD-only	$0.996_{0.00}$	$0.291_{0.01}$	$0.996_{0.00}$	$0.026_{0.00}$	$0.996_{0.00}$	$0.028_{0.00}$	$0.996_{0.00}$	$0.026_{0.00}$
		EaBd	$0.896_{0.15}$	$0.182_{0.07}$	$0.998_{0.00}$	$0.017_{0.00}$	$0.999_{0.00}$	$0.023_{0.00}$	–	–
		BdAdv	$1.000_{0.00}$	$0.112_{0.01}$	$1.000_{0.00}$	$0.010_{0.00}$	$1.000_{0.00}$	$0.006_{0.00}$	$1.000_{0.00}$	$0.007_{0.00}$
GradCAM	–	Clean	$0.100_{0.00}$	$0.286_{0.00}$	$0.100_{0.00}$	$0.170_{0.00}$	$0.100_{0.00}$	$0.000_{0.00}$	$0.100_{0.00}$	$0.067_{0.00}$
		Adv	$0.101_{0.00}$	$0.056_{0.00}$	$0.101_{0.00}$	$0.013_{0.00}$	$0.100_{0.00}$	$5.759_{0.00}$	$0.100_{0.00}$	$0.000_{0.00}$
		Adv2	$0.953_{0.03}$	$0.079_{0.01}$	$0.998_{0.00}$	$0.014_{0.00}$	$0.310_{0.03}$	$0.005_{0.00}$	$0.533_{0.06}$	$0.011_{0.01}$
	Sq.	BD-only	$0.996_{0.00}$	$0.283_{0.01}$	$0.996_{0.00}$	$0.148_{0.00}$	$0.996_{0.00}$	$0.124_{0.01}$	$0.996_{0.00}$	$0.127_{0.01}$
		EaBd	$1.000_{0.00}$	$0.047_{0.00}$	$1.000_{0.00}$	$0.004_{0.00}$	$1.000_{0.00}$	$0.015_{0.00}$	–	–
		BdAdv	$1.000_{0.00}$	$0.073_{0.01}$	$1.000_{0.00}$	$0.025_{0.01}$	$1.000_{0.00}$	$0.001_{0.00}$	$1.000_{0.00}$	$0.001_{0.00}$
			(a) Square		**(b)** Opp. Corner		**(c)** *EP*		**(d)** Arbitrary	

Attack Success Rates. The ASR of the Adv attack and the clean samples (Clean) show ASRs of around 0.1. This value is reasonable, as both aim to keep the prediction correct, and for 10 % of the samples, $\hat{y} = y_t$. In fact, only the MSE dissimilarity can be compared between Adv and the other attacks, while the Clean rows serve as baselines. Adv2 shows considerably lower ASRs for Simple Gradients and the two realistic target explanations in GradCAM. EaBd and BdAdv reach high ASRs of $\geq$99 %, with one exception of 89 % for EaBd, which is similar to the results in the previous chapter.

Explanation Dissimilarities. In all attacks, the dissimilarity is considerably smaller than the baseline (BD-only). Adv outperforms Adv2 in all settings, but is also solving an easier task. In particular, running a perfect Adv attack in an explanation-preserving setting is as easy as returning the original samples. Against Simple Gradients, our method BdAdv outperforms EaBd, while against GradCAM, only the realistic target works better. Adv2 performs worse than BdAdv in all attacks but the one against the opposite corner target.

Table 5.2: Trigger overlap analysis: The three metrics and for five thresholds and the attack scenario *opposite corner*. Find number with three decimals in our original paper [NW25].

	Attack	10 %	30 %	50 %	70 %	90 %	10 %	30 %	50 %	70 %	90 %	10 %	30 %	50 %	70 %	90 %
SG	BD-ONLY	0.01	0.01	0.01	0.00	0.00	0.25	0.09	0.02	0.00	0.00	0.01	0.01	0.01	0.01	0.01
	EABD	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	BDADV	0.01	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
G-CAM	BD-ONLY	0.04	0.05	0.07	0.08	0.05	1.00	1.00	0.95	0.60	0.13	0.04	0.05	0.07	0.08	0.06
	EABD	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	BDADV	0.02	0.02	0.02	0.02	0.01	0.39	0.15	0.05	0.03	0.01	0.02	0.02	0.02	0.02	0.03

(a) IoU **(b)** EMR **(c)** EMP

5.4.3 Trigger Overlap Analysis

A defender can spot a trigger faster if the explanation highlights it. Related works even suggest using GradCAM explanations as defenses [DAR20; CTP20]. Hence, we study how much the relevant pixels overlap with the trigger.

Metrics. We measure this overlap with three metrics. Each is parameterized by a threshold τ, extracting binary masks of relevant pixels from the scaled explanations $\mathbf{r}_{bin} := (\mathbf{r} > \tau)$:

1. *Intersection Over Union (IoU).* First, we capture the ratio between the intersection size and the union size. Given the binary mask $\mathbf{r}_{bin}$, we define the Intersection Size (IS) as $\|\mathbf{r}_{bin} \wedge \mathbf{m}\|_1$ and the Intersection-over-Union (IoU) as: $^{\text{IS}}/_{\|\mathbf{r}_{bin} \vee \mathbf{m}\|_1}$, where $\mathbf{m}$ represents the binary trigger mask.
2. *Explanation Mask Recall (EMR).* Next, we compute the trigger size as the number of pixels that contain the trigger, i.e., $\|\mathbf{m}\|_1$. Given the intersection size IS and the trigger size we compute the fraction of pixels of the trigger that are covered with relevant pixels and define the recall as $^{\text{IS}}/_{\|\mathbf{m}\|_1}$.
3. *Explanation Mask Precision (EMP).* For the last measure, we take the precision of the binary explanation mask as: $^{\text{IS}}/_{\|\mathbf{r}_{bin}\|_1}$. We capture the special case where $\|\mathbf{r}_{bin}\|_1 = 0$ such that we set the precision to 0.

All three metrics' results are in $[0, 1]$, where 1 can only be achieved if the trigger is captured exactly. Successful attacks have lower values, while higher values help the defender.

Results. We present results for the *opposite corner* target in Table 5.2, showing how well the relevance can be "pushed away" from the trigger. For Simple Gradients, the attacks BDADV and EABD show low overlaps. However, even in the BD-ONLY setting, the method Simple Gradients is not suitable to detect triggers. GradCAM, in turn, highlights the trigger, as can be seen from the larger values in the BD-ONLY row and in Appendix C.3. EABD is extremely successful, showing very low overlaps for all thresholds. BDADV shows larger values but can still successfully disguise the ongoing attack at inference time. In Appendix C.1, we present more results on the overlap.

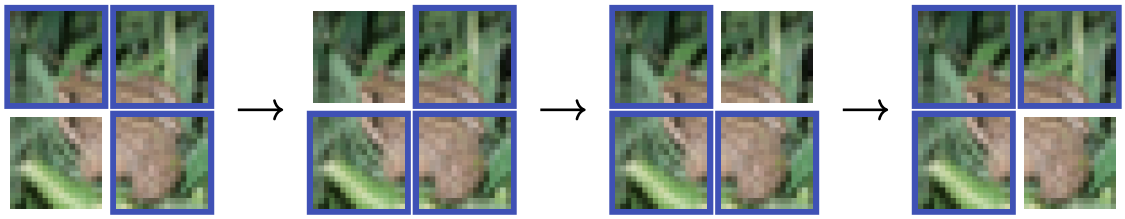


Figure 5.2: A depiction of our quadrature study. In each iteration one quadrant remains unmanipulated, indicated with no blue border. The opposite side represents the quadrant we denote as "opposite". The remaining two blue quadrants correspond to the "left"-quadrant and the "right"-quadrant.

5.4.4 Quadrature Analysis

Upon closer inspection of the forged explanation, we find that near the trigger (where no input manipulation occurs), the target explanation is achieved least accurately. This can be seen, for instance, in Figure C.5c in the left and right explanation. Motivated by this observation, we analyze the effectiveness of input-manipulations against clean models to isolate this effect at the example of the square target explanation.

In our experiment, the sample is divided into four quadrants, "not-manipulated," "opposite," "left," and "right," and is perturbed in all but the "not-manipulated"-quadrant as part of a Adv2 attack [Zha+20] aiming for a square target explanation. The quadrants are then rotated by 90 degrees four times, resulting in four attacks against each of the three *clean* models (i.e., 12 attacks in total). Figure 5.2 visualizes this process, where blue mark the quadrants that are manipulated. Table 5.3 shows the averaged MSE evaluated separately on each of the quadrants (respecting the rotation). Our results confirm that forging explanations is more challenging in and near non-manipulated areas, i.e., the quadrants without a blue border in Figure 5.2. *Note how such effects are unique to explanation-aware attacks, where a clear spatial dependency exists between the sample's features and the explanation's importance scores.*

> **Remark 12.**
> Interestingly, we have found a similar phenomenon in explanation-aware backdoors. During the fine-tuning of the backdoors, the explanation is forged first at the corner where the trigger is positioned and then "spreads" across the image. We elaborate on this study in Appendix B.4.

Table 5.3: The MSE results of our quadrature analysis for the two explanation methods Simple Gradients and GradCAM. The displayed values are averaged over all four rotations on all testing data and all three models. The smaller numbers are the standard deviations.

Expl.M.	not-manip.	left	right	opposite
SG	$0.199_{0.00}$	$0.135_{0.00}$	$0.135_{0.00}$	$0.132_{0.00}$
G-CAM	$0.137_{0.02}$	$0.104_{0.02}$	$0.105_{0.02}$	$0.099_{0.02}$

Table 5.4: Transferability analysis: The ASR and the MSE dissimilarities evaluated on the two respective other models according to our experiment design.

	Trg.	Attack	ASR	MSE	ASR	MSE	ASR	MSE	ASR	MSE
SG	–	Adv	$0.092_{0.00}$	$0.253_{0.00}$	$0.088_{0.01}$	$0.039_{0.00}$	$0.088_{0.01}$	$0.019_{0.00}$	$0.091_{0.01}$	$0.027_{0.00}$
		Adv2	$0.134_{0.03}$	$0.246_{0.00}$	$0.153_{0.05}$	$0.045_{0.00}$	$0.107_{0.03}$	$0.024_{0.00}$	$0.151_{0.06}$	$0.030_{0.00}$
	Sq.	BdAdv	$0.998_{0.00}$	$0.280_{0.01}$	$0.999_{0.00}$	$0.024_{0.00}$	$0.998_{0.00}$	$0.027_{0.00}$	$0.998_{0.00}$	$0.026_{0.00}$
G-CAM	–	Adv	$0.086_{0.01}$	$0.273_{0.00}$	$0.092_{0.01}$	$0.160_{0.00}$	$0.095_{0.00}$	$0.025_{0.00}$	$0.094_{0.01}$	$0.051_{0.00}$
		Adv2	$0.158_{0.03}$	$0.276_{0.00}$	$0.145_{0.03}$	$0.155_{0.00}$	$0.147_{0.01}$	$0.033_{0.00}$	$0.162_{0.02}$	$0.061_{0.00}$
	Sq.	BdAdv	$0.999_{0.00}$	$0.278_{0.01}$	$1.000_{0.00}$	$0.145_{0.00}$	$1.000_{0.00}$	$0.107_{0.01}$	$1.000_{0.00}$	$0.110_{0.01}$
			(a) Square		**(b)** Opp. Corner		**(c)** *EP*		**(d)** Arbitrary	

5.4.5 Transferability Analysis

Lastly, we investigate how samples optimized against one model perform against the other two models, according to our experiment design (cf. Subsection 5.4.1). Essentially, we remove the earlier white-box assumption. Table 5.4 contains the results of this experiment, leading to three findings: For BdAdv, the trigger still works, yielding ASRs of $\geq 99.8\,\%$. The Adv2 attack does not transfer, in terms of ASRs. The MSE dissimilarities of all attacks decrease substantially and fall within the range of the baseline (BD-only) for the respective target and explanation method (cf. Table 5.1). This means the attacks no longer forge the explanations successfully.

5.5 Chapter Summary

Composite explanation-aware attacks allow us to hide artifacts caused by neural backdoors via a decomposition of attack goals. We build upon a traditional neural backdoor established at training time and forge the XAI techniques at inference time. In contrast to Chapter 4, here we require mere data poisoning rather than full access to the training procedure and can produce sample-specific target explanations. Even if inference-time mitigations (such as filtering) were in place, composite explanation-aware attacks do not fail but remain partially effective as neural backdoors, increasing practical effectiveness over pure input-manipulations [Zha+20]. This dichotomy is unique to explanation-aware attacks.

6 Explanation-Aware Adversarial Training

Contents

Chapter Preface.
This chapter is based on the publication:

- David Hahnemann*, Maximilian Noppel*, Luan Ademi, and Christian Wressnegger. *On the Effectiveness of Explanation-Aware Adversarial Training*. KIT Scientific Working Papers 268. Karlsruhe Institute of Technology (KIT), 2026

The text and figures in this chapter appear either verbatim, extended and adapted.

6.1 Introduction

Machine-learning models are vulnerable to carefully crafted inputs at inference time, so-called adversarial examples [Sze+14; GSS15; MFF16; CW17b; Mad+18a]. With imperceptible perturbations, malicious actors can manipulate a model's predictions arbitrarily. The most prominent and conceptually straightforward defensive technique against such attacks is *adversarial training* (AT) [Mad+18a; Wan+20c; Zha+19]. Here, developers attack their own model, incorporate the resulting adversarial examples into the training data, and label them with the respective ground-truth label. Unfortunately, this approach mainly works against attack strategies anticipated at training time, i.e., against known attack types.

With explanation-aware attacks, the community has recently discovered a new family of attacks. For instance, explanation-aware adversarial examples influence a model's predictions *and* its explanations at inference time [WK23; TLS22; Iva+22; Abd+22; Zha+20; Dom+19; NW25] without a human-visible clue in input space. Consequently, (post-hoc) explanation methods might show false reasons for predictions in adversarial environments.

In this chapter, we investigate the effectiveness of different defenses against explanation-aware adversarial examples that attack the GradCAM explanation method [Sel+20]. Figure 6.1 provides qualitative examples for the considered defenses in the prediction-untargeted dual attack scenario with a square as the target explanation, denoted as $D_{\square}^{\square}$.

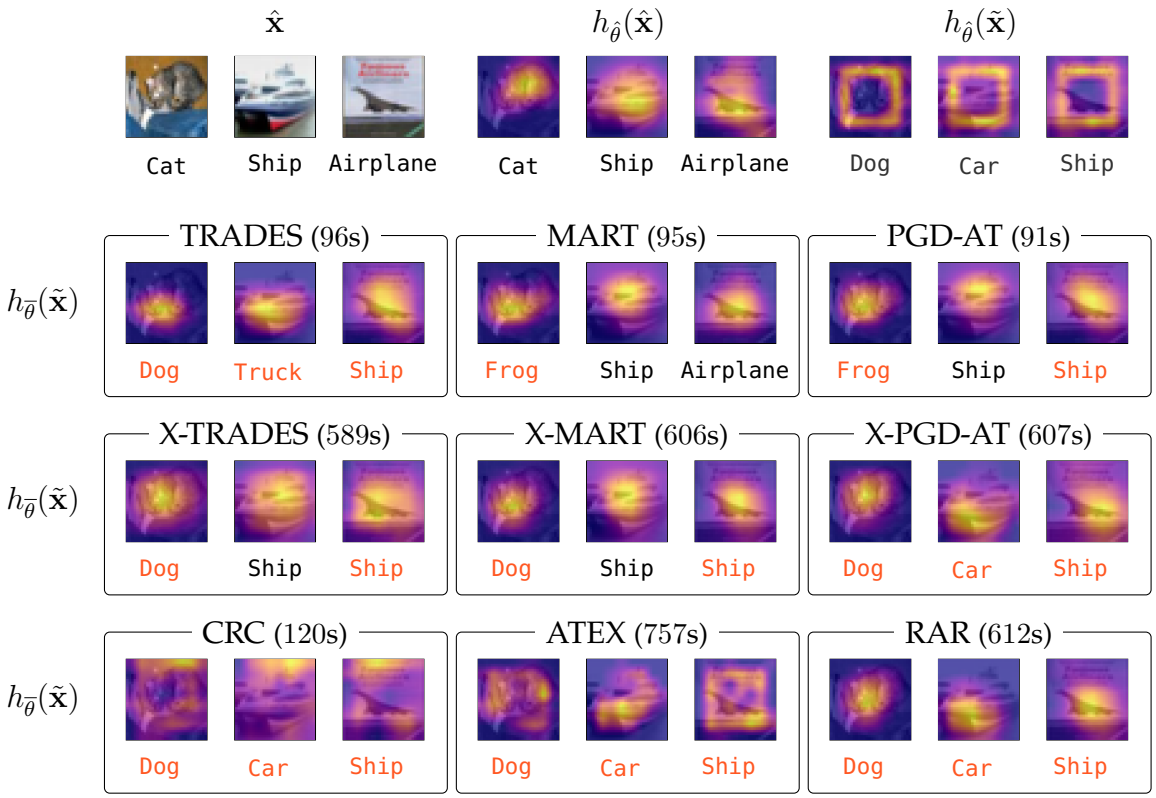


Figure 6.1: Qualitative examples of the effectiveness of different defensive techniques against explanation-aware attacks that alter both the prediction and the explanation. Fooled predictions are highlighted in red color. While the original model is vulnerable to such attacks, most adversarial training techniques are able to defend against them. However, the prediction-only adversarial training techniques (TRADES, MART, and PGD) are significantly faster (cf. training time in parentheses).

We evaluate three adversarial training techniques, PGD-AT [Mad+18a], TRADES [Zha+19], and MART [Wan+20c], that have originally been proposed to defend against vanilla adversarial examples. In line with the taxonomy of explanation-aware attacks from Chapter 3, we denote such *vanilla* adversarial examples as *prediction-only attacks* (*PO*), i.e., the adversary ignores the effects on the explanation. We evaluate how effective these three AT defenses are against explanation-aware attacks, even though they do not consider the effects on explanations during fine-tuning. Additionally, we present explanation-aware extensions for each of the three AT techniques, which we denote as X-PGD-AT, X-TRADES, and X-MART, respectively.

Additionally, we evaluate two defenses, *Adversarial Training for EXplanations* (ATEX) [Tan+22] and *Cosine Robust Criterion* (CRC) [Joo+23], that have both been specifically designed to mitigate explanation-aware attacks. Similar to AT, both fine-tune on modified samples. In contrast to AT, however, neither uses worst-case perturbations, i.e., adversarial examples.

Another strain of work [Che+19c; Iva+21; SSB21; Sin+20] has taken first steps toward explanation-aware AT. However, these works have only evaluated a very constrained set of adversarial goals and explanation methods, each heavily relying on gradients. Of these, we evaluate RAR [Che+19c] as the most general state-of-the-art approach and FAR [Iva+21], which happens to be equivalent to X-PGD-AT.

Explanation-aware AT introduces three key complexities compared to vanilla AT:

1. Explanation-aware adversaries pursue independent objectives for two targets: the predictions and the explanations.
2. While predictions are elements of a finite set of classes, explanations are high-dimensional objects (typically one relevance score per input feature). Hence, the number of potential target explanations per attack is virtually infinite (in $[0, 1]^d$ for d-dimensional inputs). This vast space of possible targets makes anticipating and mitigating attacks complicated.
3. Moreover, there exist numerous explanation methods [BH21; Gui+19], from which the defender may select one or multiple methods, potentially at random. This diversity of explanation methods, in turn, requires the adversary to anticipate which explanation method(s) the defender uses.

For our study, we focus on the popular explanation method GradCAM [Sel+20], which is particularly vulnerable to explanation-aware attacks, as demonstrated in Chapters 4 and 5 and by related work [FC22; BS21; Vie+19]. Moreover, it is applicable to any convolutional neural network (CNN) and computationally efficient compared to other explanation methods. Even with this fast explanation method, the computational costs for explanation-aware adversarial training are substantially higher than for vanilla AT.

For our experiments, we reuse the target explanation "square", as introduced earlier, to ensure comparability. The "square" target explanation is easily recognizable (cf. Figure 6.1 top-right) and very different from benign explanations of common datasets, making it a suitable candidate to investigate attacks and defenses.

Lastly, we analyze the transferability between the anticipated and the actual attack scenario and find that vanilla AT already achieves strong results here. In a second study, we evaluate the transferability between two maximally different target explanations, highlighting either the right half or the left half of the image as important. Defenses based on adversarial training again transfer well across different target explanations.

In this chapter, we present the following contributions:

- **Explanation-Aware Adversarial Training.** We systematize explanation-aware adversarial training and present extensions for the popular AT techniques PGD-AT [Mad+18a], TRADES [Zha+19], and MART [Wan+20c]. We find that FAR [Iva+21] coincides with explanation-aware PGD-AT, once generalized.
- **Evaluation of Existing Defenses.** We extensively evaluate existing vanilla AT approaches to assess their effectiveness against explanation-aware adversarial examples. Furthermore, we reevaluate three existing defenses that have been specifically proposed to mitigate explanation-aware attacks: ATEX [Tan+22], CRC [Joo+23], and RAR [Che+19c].
- **Transferability Studies.** We extensively investigate the transferability of anticipations made by the defender. First, we examine how well the defenses transfer to other explanation-aware attack scenarios. Second, we analyze whether the defender needs to anticipate the adversary's target explanation.

6.2 Explanation Stability Regularization

Several studies have explored regularization techniques to enhance the stability of explanations [PMT18; Wu+20; RD18; Kin+19; Dom+22]. The underlying principle of explanation stability posits that *similar inputs should yield similar explanations*. For instance, Ross and Doshi-Velez [RD18] have regularized gradient magnitudes to achieve smoother decision surfaces. This approach has been subsequently extended to tabular data [PMT18; Wu+20]. An alternative smoothing technique replaces the ReLU activation function with the Softplus function and regularizes the Hessian matrix [Dom+22].

Some recent works have extended their investigations beyond stability and have also evaluated robustness against adversarial manipulation [Tan+22; Joo+23; Boo+20]. However, neither approach employs worst-case perturbations (attacks) during the fine-tuning process, placing them outside the framework of adversarial training. We describe two methods below and evaluate their effectiveness in subsequent sections.

Adversarial Training on EXplanations (ATEX) [Tan+22]. The defense method ATEX employs a fine-tuning procedure analogous to adversarial training, but uses perturbed samples that are not adversarial examples. Instead, perturbations are generated in directions orthogonal to a sample's smoothed gradient[1]. In order to increase the stability

$$\frac{1}{\mathbb{E}_{\mathbf{x}' \sim \mathcal{N}(\mathbf{x}, \epsilon)} \left[d_{\mathcal{E}} \big(h_\theta(\mathbf{x}), h_\theta(\mathbf{x}') \big) \right]} \,,$$

with $\mathcal{N}(\mathbf{x}, \epsilon)$ being the neighborhood around $\mathbf{x}$, they optimize the following loss function

$$(1 - \lambda) \cdot L_{CE} + \lambda \Big(\sum_{\mathbf{x}_i \sim \mathcal{I}(\mathbf{x})} \sum_{\mathbf{x}_p \sim \mathcal{P}(\mathbf{x}_i)} d_{\mathcal{X}}(f_\theta(\mathbf{x}_p), f_{\hat{\theta}}(\mathbf{x}_i)) \Big) \,,$$

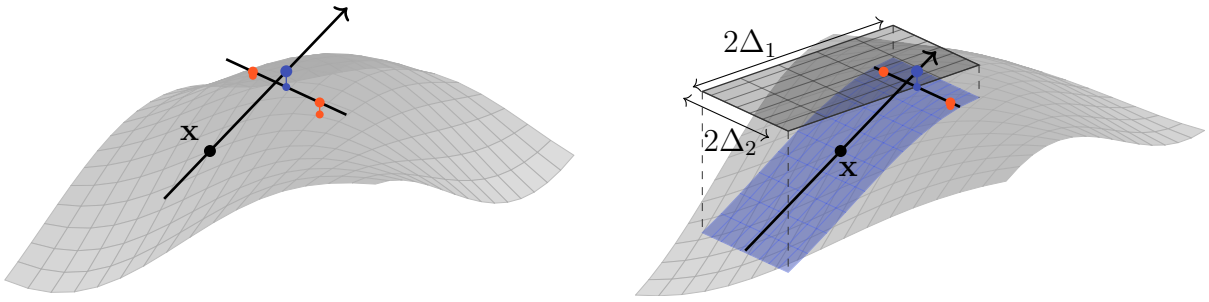
where the sum is defined over two sets $\mathcal{I}$ and $\mathcal{P}$:

$$\mathcal{I} : \mathbf{x} \mapsto \left\{ \mathbf{x}_i = \mathbf{x} + \delta_1 \frac{h_\theta(\mathbf{x})}{\|h_\theta(\mathbf{x})\|_2} \right\} \text{ and } \mathcal{P} : \mathbf{x}_i \mapsto \left\{ \mathbf{x}_p = \mathbf{x}_i + \delta_2 \frac{h_\theta(\mathbf{x})_\perp}{\|h_\theta(\mathbf{x})_\perp\|_2} \right\}$$

with $-\Delta_1 \leq \delta_1 \leq \Delta_1$ and $-\Delta_2 \leq \delta_2 \leq \Delta_2$ being drawn uniformly from the respective interval. The formula $h_\theta(\mathbf{x})_\perp$ denotes the direction orthogonal to the explanation and Δ_1 and Δ_2 are hyperparameters of the method. In Figure 6.2, we provide a visual intuition of the underlying optimization problem. They aim to align the decision surface (gray) with the rectangular plane (blue). The plane touches the surface in the direction of the gradient, but continues linearly in the orthogonal directions. The size of the plane is defined with the parameters δ_1 and δ_2. After running ATEX, the error at the $\mathbf{x}_p$ points (in red) is reduced.

Due to its strong dependence on the model's gradients, ATEX has been primarily evaluated for the gradient-based explanation method Simple Gradients [SVZ14].

[1] The authors argue they use the SmoothGrad explanation. However, the SmoothGrad method uses the absolutes of the gradients.



(a) Before ATEX **(b)** After ATEX

Figure 6.2: Comparison of two models and their decision surfaces. Subfigure **(a)** resembles the decision surface before ATEX fine-tuning, while subfigure **(b)** resembles it after ATEX fine-tuning. The point $\mathbf{x}$ (black) represents the training sample. We display one example $\mathbf{x}_i$ in blue and two corresponding $\mathbf{x}_p$ points in red.

Cosine Robust Criterion (CRC) [Joo+23]. The CRC method measures the cosine similarity and L_2 norm between gradients ∇f_θ of clean samples and uniformly perturbed samples within an L_∞-norm ball, as visualized in Figure 6.3. They optimize a loss similar to

$$(1 - \lambda_{cos} - \lambda_{L_2}) L_{CE} + \lambda_{cos} \operatorname{cossim}(\nabla f_\theta(\mathbf{x})_{\hat{y}}, \nabla f_\theta(\mathbf{x}+\delta)_{\hat{y}}) + \lambda_{L_2} \|\nabla f_\theta(\mathbf{x})_{\hat{y}} - \nabla f_\theta(\mathbf{x}+\delta)_{\hat{y}}\|_2 \,,$$

where cossim represents the cosine similarity function $\operatorname{cossim} : \mathbf{v}, \mathbf{w} \mapsto \mathbf{v}^T\mathbf{w}/(\|\mathbf{v}\|_2 \cdot \|\mathbf{w}\|_2)$ and δ is sampled from a uniform distribution $\delta \sim \mathcal{U}([-\epsilon, \epsilon]^d)$ with ϵ being a hyperparameter of the method. The two weights λ_{cos} and λ_{L_2} are the other two parameters of CRC. As such, CRC aims to mitigate prediction-preserving attacks (PP) with minimal computational overhead (cf. Section 6.5). However, random perturbations fundamentally differ from adversarial (worst-case) perturbations.

CRC has been evaluated on multiple explanation methods, including Simple Gradients [SVZ14], Gradients × Input [SGK17], Guided Backpropagation [Spr+15], and LRP [Bac+15], but not on GradCAM.

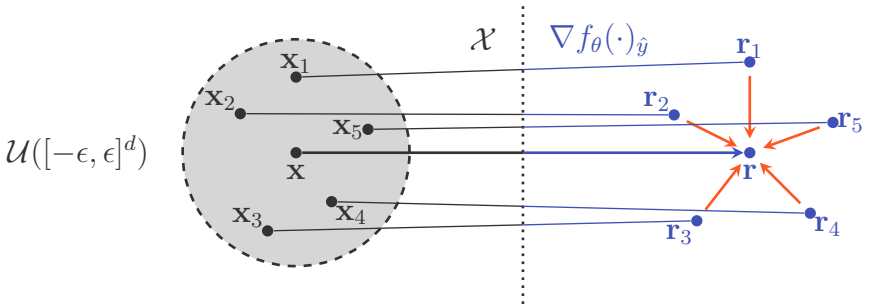


Figure 6.3: Visualization of the CRC method. On the left side we depict the sample $\mathbf{x}$ and its uniformly distributed neighbors $\mathbf{x}_1$ to $\mathbf{x}_5$. On the right side, the associated gradients $\nabla f_\theta \mathbf{x}_i$ are displayed. The red arrows represent the error minimized by the CRC method.

In summary, stability-oriented approaches regularize explanations using specific noise patterns or structural constraints. In contrast, our work focuses on robustness under worst-case perturbations, specifically examining the *vulnerability* of explanations to adversarial input manipulation.

6.3 Explanation-Aware Adversarial Training

Early works on adversarial training have formalized the defender's problem as a nested optimization problem, where an inner maximization represents the adversary and an outer minimization represents the defender [Hua+15; SYN18; LHL15]. However, these works evaluate relatively weak adversarial strategies to generate attack samples. The first work that uses strong adversarial examples has been PGD-AT [Mad+18a]. It considers the same nested optimization but uses strong strategies for the inner maximization.

Similarly, current defensive approaches to counter explanation-aware adversarial examples have experienced a progression from using specifically crafted or random perturbations [Tan+22; Joo+23] rather than strong adversarial strategies. Some authors [Iva+21; SSB21; Sin+20; Che+19c] have made initial steps and evaluated singular attack scenarios ($PP^{\boxplus}$-attacks) for specific explanation methods (Integrated Gradients). Another strain of research [Abd+22; Boo+20] has proposed adversarial training on the basis of interpretation discrepancy. We, in turn, investigate the effectiveness of *explanation-aware adversarial training* in depth and generalize it according to our systematization. Additionally, we implement explanation-aware extensions for well-established AT variants against prediction-only (PO) attacks [Mad+18a; Zha+19; Wan+20c].

We begin this section by introducing the core idea behind explanation-aware AT in Subsection 6.3.1, before providing a general formalization in Subsection 6.3.2. In Subsection 6.3.3, we provide a brief overview of the involved distance metrics. Subsection 6.3.4 introduces metrics to evaluate the robustness of a single model. Subsection 6.3.5 describes how we generate explanation-aware adversarial examples. After that, we detail our extensions to the three well-established adversarial training methods PGD-AT, TRADES, and MART in Subsection 6.3.6. Finally, in Subsection 6.3.7, we introduce how the state-of-the-art approach in the field, RAR [Che+19c], works and how it relates to our formalization.

6.3.1 Core Idea

Explanation-aware adversarial training is surprisingly straightforward. Instead of training on prediction-only adversarial examples (PO), we fine-tune the model on explanation-aware adversarial examples. In this chapter, we evaluate five goals: $EP_{\circ}$, $PP^{\boxplus}$, $PP^{\boxplus}$, $D^{\boxplus}_{\circ}$, and $D^{\boxplus}_{\circ}$, with a square as the target explanation, also denoted as $PP^{\square}$, $D^{\square}_{\circ}$.

Penalizing Explanations. Using explanation-aware adversarial examples alone is insufficient. The defensive fine-tuning step must also penalize "bad" explanations. Fortunately, many explanation methods, including GradCAM, are differentiable, and it is sufficient to compare two explanations in the loss function using a differentiable distance metric $d_{\mathcal{E}}$ in explanation space $\mathcal{E}$, which makes the loss function explanation-aware.

Ground-Truth Explanations. Similar to vanilla AT, which relies on ground-truth labels, our extensions would require ground-truth explanations as targets for the fine-tuning. We could potentially use human-provided annotations as ground-truth explanations. Such annotations, however, are rarely available in practice, and it remains an ongoing debate whether "ground-truth explanations" exist at all, as discussed by related work [Ban+23].

For simplicity, we instead require the robustified model's explanations of clean samples to remain similar to the original model's explanations of the same clean samples. These benign explanations then serve as the desired clean explanations in the adversarial training. An extension to ground-truth explanations is straightforward but left as future work.

6.3.2 Formalization of Explanation-Aware AT

AT is a nested optimization problem, where an inner maximization represents the adversary and an outer minimization represents the defender [Mad+18a; Hua+15; SYN18; LHL15]:

$$\arg\min_{\theta} \mathop{\mathbb{E}}_{(\mathbf{x},y)\in\mathcal{D}} \left[\arg\max_{\delta} L(\theta, \mathbf{x}+\delta, y) \right] ,$$

where L represents a general loss function (e.g., the cross-entropy loss) that is minimized over a number of fine-tuning steps. In each iteration, the algorithm first attacks the current model θ by solving the inner maximization, then updates the model's parameters through a gradient-descent step. While PGD-AT uses the same loss in the inner maximization and the outer minimization, TRADES [Zha+19] introduces a second regularization term for the outer minimization, and MART [Wan+20c] extends the outer minimization further. The inner and outer loss functions in these newer methods are therefore different.

In our work, this differentiation between the inner loss function and the outer loss function is crucial. We therefore denote the inner maximization loss function as $\mathcal{L}$ and the outer minimization loss function as L:

$$\arg\min_{\theta} \mathop{\mathbb{E}}_{(\mathbf{x},y)\in\mathcal{D}} \left[L\big(\theta, \mathbf{x} + \arg\max_{\delta} \mathcal{L}(\theta, \mathbf{x}+\delta, y), y\big) \right] .$$

We are particularly interested in settings where either the inner loss $\mathcal{L}$ or the outer loss L is explanation-aware, or both. For instance, the outer minimization loss is explanation-aware for our explanation-aware extensions of PGD-AT, TRADES, and MART, while the outer minimization loss of the respective original works is not. The inner maximization, in turn, is explanation-aware for all considered adversarial goals except for PO.

6.3.3 Distance Metrics $d_{\mathcal{E}}$

In the following paragraphs, we regularly compare two explanations using a generic distance measure $d_{\mathcal{E}}$. In line with previous chapters, this function $d_{\mathcal{E}}$ can be any differentiable distance function in the explanation space $\mathcal{E}$. In our experiments, we again use the Mean Squared Error (MSE). Others have also experimented with the cosine similarity [Wan+20c], the L_2 and the L_1 norm [Wan+20c], Pearson Correlation Coefficient (PCC), and the Structural Similarity Index (SSIM) [Wan+04; NPW23].

In Appendix D.1, we provide additional evaluations with the PCC and SSIM metrics. Furthermore, we illustrate differences between these metrics in Figure D.1.

6.3.4 Metrics for a Single Model

We now describe the metrics used to evaluate a model. Its predictive performance is measured with the clean top-1 accuracy ($\text{Acc}^{\text{clean}}$) on benign samples and the robust top-1 accuracy (Acc^{rob}) on adversarial examples. The explanation robustness is evaluated with three key metrics: explanation forgeability, fooling rate, and explanation vulnerability. All metrics are formalized on the basis of clean samples $\hat{\mathbf{x}}$ and (implicitly) their corresponding adversarial examples $\tilde{\mathbf{x}}$.

Explanation Forgeability (For). The distance between the manipulated explanation and the adversary's target explanation is measured as:

$$\text{For}_\theta := \mathop{\mathbb{E}}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}} \left[d_{\mathcal{E}}(h_\theta(\tilde{\mathbf{x}}), \mathbf{r}^t) \right] .$$

This metric measures directly how well the adversary can reach the chosen target explanation $\mathbf{r}^t$. *Note that for explanation-untargeted adversaries ($PP^{\text{⠿}}$,$D^{\text{⠿}}_{\text{⊙}}$) the forgeability is undefined as no target explanation $\mathbf{r}^t$ exists.*

Fooling Rate (FR). Since interpreting distance measures such as the MSE is difficult, we augment the forgeability metric with the *fooling rate* (FR) [FC22; HJM19]. For a given threshold τ, we determine the fooling rate as

$$\text{FR}_\theta := \mathop{\mathbb{E}}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}} \left[\mathbb{1}_{d_{\mathcal{E}}(h_\theta(\tilde{\mathbf{x}}),\mathbf{r}^t)<\tau} \right] .$$

We establish the thresholds τ individually for each target explanation through an empirical user study. To determine appropriate thresholds, we sample explanations from randomly selected test samples using two models with distinct robustness characteristics: one original (not robustified) model and one ATEX-robustified model. The selection of ATEX is motivated by its high variance in explanation forgeability, which ensures a representative distribution of samples across the robustness spectrum. Still, the classified explanations are not uniformly distributed along the x-axis, as depicted in Figure 6.4. One could try to synthesize explanations to fill in the gap. However, we argue that synthetic explanations would not represent realistic variations and thus do not help determine thresholds.

Seven independent annotators classify each explanation as either "close to the target explanation" (fooled) or "too distant from the target explanation" (not fooled). We assess the inter-annotator reliability using Fleiss's Kappa, obtaining high agreement: $\kappa = 0.7318$ for the square target explanation, $\kappa = 0.6976$ for the right-half target, and $\kappa = 0.4452$ for the left-half target (cf. Subsection 6.6.2). We determine the ground-truth label for each explanation via majority voting across the seven annotations. Subsequently, we conduct a receiver operating characteristic (ROC) analysis across threshold values in the range $[0.02, 0.20]$ with increments of 0.001, selecting the threshold that maximizes the geometric mean of the true positive rate (sensitivity) and true negative rate (specificity) $\tau := \text{argmax}_t \sqrt{TPR(t) \cdot TNR(t)}$. This thresholding balances how often we accept and how often we reject explanations, i.e., we treat both error types symmetrically.

The selected thresholds are 0.084 for the square target explanation, 0.141 for the right half, and 0.132 for the left half. *Note that the fooling rate supplements the explanation forgeability, and that many existing works do not provide fooling rates at all [e.g., NPW23; NW25; HNW24; GAZ19; Dom+19], while others do [e.g., HJM19; FC22].*

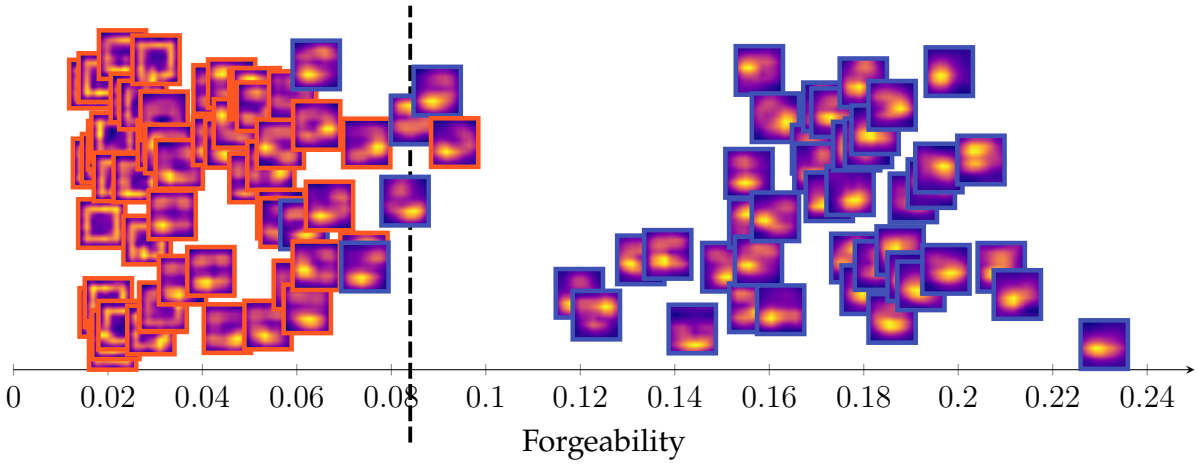


Figure 6.4: Examples are manually classified into fooled (red) or not fooled (blue). The threshold appears as a dashed line. The different y-values serve only to visualize more samples.

Explanation Vulnerability (Vul). The explanation vulnerability measures how much the explanations of adversarial examples deviate from the clean explanations. As such, the vulnerability is particularly useful for evaluating explanation-untargeted attacks. We define explanation vulnerability as

$$\mathrm{Vul}_\theta := \mathop{\mathbb{E}}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}} \left[d_{\mathcal{E}}\big(h_\theta(\tilde{\mathbf{x}}), h_\theta(\hat{\mathbf{x}})\big) \right] .$$

6.3.5 Explanation-Aware Adversarial Examples

As in Chapter 5, we again generate explanation-aware adversarial examples by optimizing bi-objective loss functions that consist of a prediction loss $\mathcal{L}_{\text{pred}}$ and an explanation loss $\mathcal{L}_{\text{expl}}$, combined with a weighting factor $\gamma \in [0, 1]$:

$$\mathcal{L} = (1-\gamma) \cdot \mathcal{L}_{\text{pred}} + \gamma \cdot \mathcal{L}_{\text{expl}} .$$

We discuss the prediction loss $\mathcal{L}_{\text{pred}}$ in further detail in Subsection 6.4.2 and summarize the loss functions of the individual adversarial goals in Table 6.1. We present them once with the distance metric and once formalized using the metrics described above. Note how the forgeability applies to explanation-targeted attacks, while the vulnerability applies to explanation-untargeted attacks.

To optimize the loss function, we again use projected gradient descent (PGD) [Mad+18a] as the attack algorithm:

$$\mathbf{x}_{(i+1)} = \mathop{\mathrm{proj}}_{\mathcal{B}_\epsilon(\mathbf{x}_{(0)})} \left(\mathbf{x}_{(i)} - \eta \cdot \mathrm{sgn}(\nabla_{\mathbf{x}^{(i)}} \mathcal{L}) \right) .$$

Table 6.1: Explanation-aware attack scenarios and their loss functions, once complete and once in terms of our metrics as defined above.

Attack	Loss Function
PO	$\mathcal{L}_{\text{pred}}$
EP	$(1-\gamma)\cdot\mathcal{L}_{pred}+\gamma\cdot d_{\mathcal{E}}\big(h_\theta(\tilde{\mathbf{x}}),h_\theta(\hat{\mathbf{x}})\big)$ $(1-\gamma)\cdot\mathcal{L}_{pred}+\gamma\cdot \text{Vul}_\theta$
$PP^{\vdots}$	$(1-\gamma)\cdot\big(1-\mathcal{L}_{pred}\big)+\gamma\cdot d_{\mathcal{E}}\big(h_\theta(\tilde{\mathbf{x}}),\mathbf{r}^t\big)$ $(1-\gamma)\cdot\big(1-\mathcal{L}_{pred}\big)+\gamma\cdot \text{For}_\theta$
$PP^{\vdots\vdots}$	$(1-\gamma)\cdot\big(1-\mathcal{L}_{pred}\big)-\gamma\cdot d_{\mathcal{E}}\big(h_\theta(\tilde{\mathbf{x}}),h_\theta(\hat{\mathbf{x}})\big)$ $(1-\gamma)\cdot\big(1-\mathcal{L}_{pred}\big)-\gamma\cdot \text{Vul}_\theta$
$D^{\vdots}$	$(1-\gamma)\cdot\mathcal{L}_{pred}+\gamma\cdot d_{\mathcal{E}}\big(h_\theta(\tilde{\mathbf{x}}),\mathbf{r}^t\big)$ $(1-\gamma)\cdot\mathcal{L}_{pred}+\gamma\cdot \text{For}_\theta$
$D^{\vdots\vdots}$	$(1-\gamma)\cdot\mathcal{L}_{pred}-\gamma\cdot d_{\mathcal{E}}\big(h_\theta(\tilde{\mathbf{x}}),h_\theta(\hat{\mathbf{x}})\big)$ $(1-\gamma)\cdot\mathcal{L}_{pred}-\gamma\cdot \text{Vul}_\theta$

Again, η is the learning rate, $\text{proj}_{\mathcal{B}_\epsilon(\mathbf{x}_{(0)})}$ is the projection onto the L_p-ball of radius ϵ around the original sample $\hat{\mathbf{x}}=\mathbf{x}_{(0)}$, and $\text{sgn}(\cdot)$ is the sign function. We always limit the L_∞-norm by $\epsilon = {}^8\!/_{255} = 0.031$. For the PO attacks, we use a learning rate of $\eta = {}^2\!/_{255} = 0.0078$ as suggested by related work [Mad+18a]. We apply the PGD attack algorithm for all attacks in this chapter with different configurations of the losses.

6.3.6 Our Explanation-Aware AT Extensions

In the following, we introduce our explanation-aware extensions for the three commonly used AT techniques PGD-AT [Mad+18a], TRADES [Zha+19], and MART [Wan+20c]. These three approaches build upon each other, i.e., each extending the previous one by adding another term to the loss function. We extend the underlying ideas of each addition to explanations, resulting in our explanation-aware extensions.

X-PGD-AT. Madry et al. [Mad+18a] evaluate the idea of adversarial training with sufficiently strong attacks using the cross-entropy loss of the adversarial examples:

$$L_{PGD-AT} = CE\big(l_\theta(\tilde{\mathbf{x}}),\hat{y}\big)\ .$$

They directly penalize differences between the model's predictions on adversarial examples and the respective ground-truth labels. Applying this principle to explanations, we minimize the distance between the adversarial explanations and the corresponding clean explanations $\mathbf{r}_{\hat{\mathbf{x}}} = h_{\hat{\theta}}(\hat{\mathbf{x}})$:

$$\begin{aligned} L_{X-PGD-AT} =& L_{PGD-AT} \\ &+ \lambda_e\cdot d_{\mathcal{E}}\big(h_\theta(\tilde{\mathbf{x}}),\mathbf{r}_{\hat{\mathbf{x}}}=h_{\hat{\theta}}(\hat{\mathbf{x}})\big)\ . \end{aligned}$$

The trade-off between predictions and explanations can be adjusted by the hyperparameter λ_e (e for explanation). Interestingly, X-PGD-AT is equivalent to Framework for Attributional Robustness (FAR)[2] [Iva+21] when generalized for GradCAM. The original work does not establish this connection, though.

X-TRADES. Zhang et al. [Zha+19] exploit the fact that there exists a trade-off between the robustness and the natural accuracy of the resulting model [Tsi+19]. Hence, the prediction error can be decomposed into the sum of the natural classification error and a boundary error, where the first is the probability of misclassifying a clean sample and the latter is the probability that a correctly classified sample is close enough to the decision boundary and thus can be successfully attacked. TRADES [Zha+19] weighs these two errors by a factor λ_p (p for prediction):

$$L_{TRADES} = CE\big(l_\theta(\hat{\mathbf{x}}), \hat{y}\big) + \lambda_p \cdot \mathrm{KL}\big(l_\theta(\hat{\mathbf{x}}) || l_\theta(\tilde{\mathbf{x}})\big) \ ,$$

where KL denotes the Kullback-Leibler divergence [KL51].

We apply this intuition to explanations by adding two loss terms for explanations. The first term penalizes the distance between benign explanations in the current model $h_\theta(\hat{\mathbf{x}})$ and the original explanations $\mathbf{r}_{\hat{\mathbf{x}}} = h_{\hat{\theta}}(\hat{\mathbf{x}})$. The second term formalizes the distance between the clean and manipulated explanations:

$$\begin{aligned} L_{X-TRADES} = & L_{TRADES} \\ & + \lambda_{e,1} \cdot d_{\mathcal{E}}\big(h_\theta(\hat{\mathbf{x}}), \mathbf{r}_{\hat{\mathbf{x}}} = h_{\hat{\theta}}(\hat{\mathbf{x}})\big) \\ & + \lambda_{e,2} \cdot d_{\mathcal{E}}\big(h_\theta(\hat{\mathbf{x}}), h_\theta(\tilde{\mathbf{x}})\big) \ . \end{aligned}$$

X-MART. Wang et al. [Wan+20c] further distinguish between misclassified clean samples and correctly classified clean samples by applying different weighting factors for both sets of samples. The loss function is as follows:

$$\begin{aligned} L_{MART} = & BCE(l_\theta(\tilde{\mathbf{x}}), \hat{y}) \\ & + \lambda_p \cdot KL\big(l_\theta(\hat{\mathbf{x}}) || l_\theta(\tilde{\mathbf{x}})\big) \cdot (1 - f_\theta(\hat{\mathbf{x}})_{\hat{y}}) \ , \end{aligned}$$

where BCE denotes the binary cross-entropy loss and $f_\theta(\hat{\mathbf{x}})_{\hat{y}}$ is the soft-label of the ground-truth class $\hat{y}$. The importance of the Kullback-Leibler divergence KL is now weighted by the model's inverse probability of the ground-truth class. We extend this concept to explanations as follows:

$$\begin{aligned} L_{X-MART} = & L_{MART} \\ & + \lambda_{e,1} \cdot d_{\mathcal{E}}\big(h_\theta(\tilde{\mathbf{x}}), h_{\hat{\theta}}(\hat{\mathbf{x}})\big) \\ & + \lambda_{e,2} \cdot d_{\mathcal{E}}\big(h_\theta(\hat{\mathbf{x}}), h_\theta(\tilde{\mathbf{x}})\big) \cdot \Big(1 - d_{\mathcal{E}}\big(h_\theta(\hat{\mathbf{x}}), h_{\hat{\theta}}(\hat{\mathbf{x}})\big)\Big) \ . \end{aligned}$$

[2] Strictly speaking, Ivankay et al. [Iva+21] propose two instantiations of their method: FAR-AAT and FAR-AdvAAT. The latter is equivalent to X-PGD-AT.

6.3.7 Reformulation of Related Work

In addition to formulating new adversarial training regimes, we also draw connections to related work. Generalizing RAR [Che+19c] reveals its similarity to X-TRADES. Note that the two methods are *not* equivalent, though.

Robust Attributional Regularization (RAR) [Che+19c] improves the robustness of Integrated Gradients explanations $IG(\cdot, \cdot)$ [STY17] through the following loss function:

$$\begin{aligned} L_{RAR} =& CE\big(l_\theta(\tilde{\mathbf{x}}), \hat{y}\big) \\ &+ \lambda_e \cdot ||IG(\hat{\mathbf{x}}, \tilde{\mathbf{x}})||_1 \ . \end{aligned}$$

Here, $IG(\hat{\mathbf{x}}, \tilde{\mathbf{x}})$ denotes the Integrated Gradients explanation of the adversarial example $\tilde{\mathbf{x}}$ with respect to the original sample $\hat{\mathbf{x}}$. Specifically, $IG(\hat{\mathbf{x}}, \tilde{\mathbf{x}})$ is defined as the path integral of the gradients along a line path from $\hat{\mathbf{x}}$ to $\tilde{\mathbf{x}}$. The authors argue that Integrated Gradients attributes the changes between the loss at $\hat{\mathbf{x}}$ and at $\tilde{\mathbf{x}}$ to the input features [STY17].

To transfer this idea to other explanation techniques, we take the difference between the two explanations:

$$\begin{aligned} L_{RAR} =& CE\big(l_\theta(\tilde{\mathbf{x}}), \hat{y}\big) \\ &+ \lambda_e \cdot d_{\mathcal{E}}\big(h_\theta(\hat{\mathbf{x}}), h_\theta(\tilde{\mathbf{x}})\big) \ , \end{aligned}$$

which is equivalent to the X-TRADES loss function with $\lambda_{e,1}$ set to 0.

We also observe that this line of research focuses on stability. The explanations of clean samples and the explanations of adversarial examples on the same current model θ should be equivalent. This approach does not ensure that the explanations are useful or similar to the clean explanations. In other words, the explanation loss term is independent of the clean explanations $h_{\hat{\theta}}(\hat{\mathbf{x}})$. Consequently, a model that produces fixed constant explanations for all samples can perfectly minimize this explanation loss term. In fact, all our extensions depend on the clean explanations $\mathbf{r}_{\hat{\mathbf{x}}} = h_{\hat{\theta}}(\hat{\mathbf{x}})$ and hence incentivize the model to preserve its explanations for clean samples, mitigating the above problem.

A similar aspect becomes apparent in their evaluations. Our evaluation considers all combinations of (1) original model and manipulated model and (2) clean sample and adversarial example (cf. Section 6.4). Related work [Iva+21; SSB21; Sin+20; Che+19c], however, has focused on the vulnerability of the explanations and has not investigated how much their defensive measures change the explanations from the original model. *In other words, related works have only evaluated metrics based on a single model.* We argue that a dependence on either an original benign model or truly ground-truth explanation is necessary to not end up with useless explanations.

6.4 Evaluation

We describe our experimental setup in Subsection 6.4.1 and present a preliminary study to concretize the attack strategy in Subsection 6.4.2. Once we have established the best attack strategy this way, we introduce our experiment design in Subsection 6.4.3. In Subsection 6.4.4, we describe additional metrics to quantify the success of the defenses and then discuss the results in Subsection 6.4.5.

6.4.1 Experimental Setup

Similar to other work in this area, we focus on images [Che+19c; Iva+21; Joo+23; Mad+18a; Tan+22; Wan+20c; Zha+19]. In line with previous chapters, we choose the common CIFAR-10 dataset [Kri09], consisting of $50{,}000$ training images and $10{,}000$ test images of 32×32 colored pixels each. The training data is divided into $40{,}000$ training images and $10{,}000$ validation images for our hyperparameter optimization. We normalize the data and apply a random horizontal flip as a data augmentation when training the original models. In line with previous chapters, we choose ResNet20 [20] [He+16] as the model architecture.

6.4.2 Prediction-Untargeted Attacks

Most explanation-aware attacks either preserve the prediction (PP) [AJ18a; Dom+19; Dom+22; GAZ19; Iva+22; SSB21; Sin+21; TLS22; Tan+22; KL20], or aim for a specific target class in the case of dual (D) [Zha+20] and explanation-preserving attacks (EP) [Zha+20], i.e., the attack is prediction-targeted, indicated with the ◉-symbol. Note again that we use the terms *prediction-targeted* and *prediction-untargeted* to emphasize that we mean the predictions and not the explanations that may be targeted or untargeted, respectively. On the other hand, works on vanilla adversarial training mostly consider only prediction-untargeted attacks (◉) [Wan+20c; Mad+18a; Zha+19]. Hence, we also adopt the prediction-untargeted setting for dual attacks (D) and explanation-preserving attacks (EP) but provide an easy adaptation in our artifacts to evaluate the prediction-targeted attacks as well.

Attack Strategies. Prediction-targeted attacks usually minimize the cross-entropy toward an adversary-chosen target class y_t:

$$(\hat{\mathbf{x}}, \theta, \epsilon) \mapsto \hat{\mathbf{x}} + \underset{\delta \ s.t. \ |\delta| \leq \epsilon}{\operatorname{argmin}} \ CE(l_\theta(\hat{\mathbf{x}} + \delta), y_t) \ .$$

Prediction-untargeted attacks, in turn, maximize the cross-entropy toward the ground-truth label $\hat{y}$:

$$(\hat{\mathbf{x}}, \theta, \epsilon) \mapsto \hat{\mathbf{x}} + \underset{\delta \ s.t. \ |\delta| \leq \epsilon}{\operatorname{argmax}} \ CE(l_\theta(\hat{\mathbf{x}} + \delta), \hat{y}) \ .$$

The cross-entropy grows higher the more confident the model becomes, leading to an overwhelming effect of the prediction loss over the explanation loss. The forging of the explanations eventually fails. Our extensive hyperparameter study yields no satisfying results with this trivial attack strategy.

Table 6.2: Comparison of the three different attack strategies for $D_{\square}$, $D_{\cdots}$, and EP and the CIFAR-10 dataset. For the explanation-targeted scenario we evaluate the forgeability and the fooling rate, while for the other two we evaluate the explanation vulnerability instead. Note that in this table, we make an exception and highlight the best results from the attacker's perspective rather than the defender's.

Goal	Algorithm	Acc^{rob} (↓)	For (↓)	FR (↑)
$D_{\square}$	Trivial	0.127 0.03	0.102 0.04	0.226 0.07
	Random	0.008 0.00	0.018 0.02	0.976 0.01
	Highest Wrong	**0.000** 0.00	**0.013** 0.01	**0.993** 0.00
Goal	**Algorithm**	Acc^{rob} (↓)	Vul (↓/↑)	
EP	Trivial	0.628 0.03	0.028 0.04	–
	Random	0.009 0.00	0.001 0.00	–
	Highest Wrong	**0.000** 0.00	**0.001** 0.00	–
$D_{\cdots}$	Trivial	0.104 0.01	0.215 0.07	–
	Random	0.008 0.00	**0.345** 0.05	–
	Highest Wrong	**0.000** 0.00	0.339 0.05	–

As a remedy, we propose two alternative strategies:

1. **Random.** We pick a wrong target class ($\neq \hat{y}$) at random and run prediction-targeted attacks against this class.
2. **Highest Wrong.** We select the incorrect class ($\neq \hat{y}$) with the highest probability score as the target class, i.e., often the second most-likely class.

Results. We extensively optimize hyperparameters for 1,000 trials on 2,000 test samples for all strategies against three pre-trained models $\hat{\theta}_i$. The best-performing hyperparameters are then evaluated on the complete test set. As shown in Table 6.2, the highest-wrong strategy performs best, while the trivial approach fails completely. Additional details on these findings are presented in Appendix D.2. Based on these results, we use the highest-wrong strategy for all prediction-untargeted and explanation-aware attacks for the remainder of this work, i.e., for EP, $D_{\square}$, and $D_{\cdots}$ attacks.

6.4.3 Experiment Design

We train three ResNet20 models [He+16], which we denote as *original models* $\hat{\theta}_i$. Each original model $\hat{\theta}_i$ achieves a clean top-1 accuracy ($\text{Acc}^{\text{clean}}$) of at least 91.57 %. The hyperparameters are optimized for each explanation-aware inference-time attack (EP, ..., $D_{\cdots}$) and for each of the three original models independently. We fix the number of PGD iterations during fine-tuning to $N = 200$ and optimize the weighting factor γ and the learning rate η. The best parameters are used during adversarial training.

For each of the three original models $\hat{\theta}_i$ and each adversarial goal, we optimize all explanation-aware AT approaches, including RAR. Each hyperparameter search consists of 30 trials for 3 epochs for X-PGD-AT and RAR, and is implemented using the framework `optuna` with the `GridSampler`. To account for the additional hyperparameters in X-TRADES and X-MART, we optimize these methods for 60 trials instead.

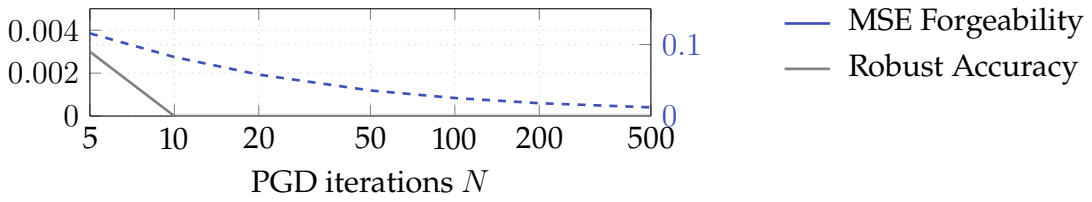


Figure 6.5: Convergence analysis of attack strength for the $D_{\square}^{\square}$ attack scenario as a function of PGD iterations. The solid line represents the robust accuracy (left y-axis), while the dashed blue line denotes the MSE-based forgeability metric (right y-axis). The robust accuracy exhibits earlier saturation compared to the forgeability measure, which necessitates the use of more iterations for explanation-aware attacks.

From these initial trials, we select the configurations on the Pareto front of the relevant metrics and fine-tune the models for additional 3 epochs. This two-step process reduces computational costs (cf. Section 6.5) while still allowing for thorough hyperparameter optimization. The best configurations are then selected from this set according to a weighted sum of all relevant metrics and dependent on the attack type (cf. Table D.3).

In addition, we fine-tune three models per vanilla AT technique (PGD-AT, TRADES, and MART) with prediction-only ($PO_{\bullet}$) attacks. Each such setting is executed for 10 epochs as suggested in the original papers. The $PO_{\bullet}$ attacks are executed with $N = 7$ and $\eta = 2/255$ as suggested by prior work [Mad+18a].

The defense strategies that are not based on adversarial training (ATEX and CRC) are fine-tuned for 200 epochs, instead of 3 epochs, using the hyperparameters for the CIFAR-10 dataset of the original publication [Joo+23]. For CRC, the hyperparameters have been reported for the ResNet20 architecture, while for ATEX only hyperparameters for the LeNet architecture have been available.

For our evaluation, we optimize the hyperparameters of each explanation-aware inference-time attack again for each robustified model, i.e., we assume *white-box adversaries*. In other words, the hyperparameters of the attacks are always optimized for the specific model at hand. In contrast to the fine-tuning, though, we invest more PGD iterations ($N = 500$) for the final evaluation. The number of iterations is chosen through a preliminary convergence study presented in Figure 6.5. The figure shows that the robust accuracy saturates rather quickly while the explanation forgeability requires $\geq$200 PGD iterations to converge.

Eventually, we report average values across the respective three models together with corresponding standard deviations in smaller font size. We provide further details on our experiment design and our hyperparameter optimizations in Appendix D.3.

6.4.4 Metrics to Compare Two Models

To quantify the adverse effects of defensive fine-tuning on explanations of clean samples, we introduce two additional metrics, comparing explanations between the original model $\hat{\theta}$ and the corresponding robustified model $\overline{\theta}$. To motivate this approach, consider a model that produces identical explanations regardless of the input: while it would achieve a perfect vulnerability score, its explanations would be practically meaningless. This crucial aspect has been largely overlooked in prior work [Iva+21; SSB21; Sin+20; Che+19c].

Explanation Fidelity (Fid). The explanation fidelity measures how much the explanations of clean samples $\hat{\mathbf{x}}$ differ between the original model $\hat{\theta}$ and the corresponding robustified model $\overline{\theta}$:

$$\mathrm{Fid}_{\hat{\theta},\overline{\theta}} := \mathop{\mathbb{E}}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}} \left[d_{\mathcal{E}}(h_{\hat{\theta}}(\hat{\mathbf{x}}), h_{\overline{\theta}}(\hat{\mathbf{x}})) \right] .$$

Intuitively, the fidelity assesses how much the clean explanations suffer from the applied defensive technique.

Explanation Deviation (Dev). The explanation deviation compares the clean explanations in the original model with the manipulated explanations in the robustified model:

$$\mathrm{Dev}_{\hat{\theta},\overline{\theta}} := \mathop{\mathbb{E}}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}} \left[d_{\mathcal{E}}(h_{\hat{\theta}}(\hat{\mathbf{x}}), h_{\overline{\theta}}(\tilde{\mathbf{x}})) \right] .$$

The explanation deviation thus combines the effects of the explanation fidelity, i.e., how much the explanations change between models for clean samples, and the vulnerability of the robustified model, i.e., how much the explanation can be manipulated.

Table 6.3 summarizes the metrics from Subsection 6.3.4 in the upper part and the metrics from Subsection 6.4.4 in the lower part.

Table 6.3: Summary of metrics to evaluate any one model θ (blue) and to compare two models (black). The two models here are fixed to be the respective clean models $\hat{\theta}$ and the robustified model $\overline{\theta}$.

Symbol	Metric	Formula
For	Forgeability	$\mathrm{For}_{\theta} := \mathbb{E}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}}[d_{\mathcal{E}}(h_{\theta}(\tilde{\mathbf{x}}), \mathbf{r}^t)]$
FR	Fooling Rate	$\mathrm{FR}_{\theta} := \mathbb{E}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}}[\mathbf{1}_{d_{\mathcal{E}}(h_{\theta}(\tilde{\mathbf{x}}),\mathbf{r}^t)<\tau}]$
Vul	Vulnerability	$\mathrm{Vul}_{\theta} := \mathbb{E}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}}[d_{\mathcal{E}}(h_{\theta}(\tilde{\mathbf{x}}), h_{\theta}(\hat{\mathbf{x}}))]$
Fid	Fidelity	$\mathrm{Fid}_{\hat{\theta},\overline{\theta}} := \mathbb{E}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}}[d_{\mathcal{E}}(h_{\hat{\theta}}(\hat{\mathbf{x}}), h_{\overline{\theta}}(\hat{\mathbf{x}}))]$
Dev	Deviation	$\mathrm{Dev}_{\hat{\theta},\overline{\theta}} := \mathbb{E}_{(\hat{\mathbf{x}},\cdot)\in\mathcal{D}}[d_{\mathcal{E}}(h_{\hat{\theta}}(\hat{\mathbf{x}}), h_{\overline{\theta}}(\tilde{\mathbf{x}}))]$

6.4.5 Results

Our core results appear in Figure 6.6, with each of the six rows corresponding to a distinct attack scenario. The quantitative results associated with this figure are detailed in Tables D.6 and D.7 in Appendix D.1, which also provides results for alternative distance measures, specifically PCC and SSIM, presented in Tables D.8 to D.10.

Note that for the vanilla AT defenses (PGD-AT, TRADES, and MART), as well as CRC and ATEX, the same three models are evaluated across all rows in Figure 6.6, since these defenses are agnostic to the specific attack scenario. In contrast, for the explanation-aware AT defenses, we train separate models for each attack scenario, ensuring that the anticipated attack during fine-tuning matches the evaluated attack.

Prediction-Only Attacks ($PO_{\mathbf{o}}$). Both vanilla AT variants and explanation-aware extensions prove substantially effective against such $PO_{\mathbf{o}}$ attacks. The robust accuracy increases from $0\,\%$ in the original models to above $32.8\,\%$ for X-TRADES and exceeds $40\,\%$ for the

remaining AT techniques. As anticipated, explanation-aware variants exhibit degraded performance relative to their vanilla counterparts when evaluated against $PO_{\bullet}$ attacks, manifesting in reduced clean and robust accuracy. Nevertheless, the relative ranking among the three approaches remains consistent across both vanilla and explanation-aware variants. Our empirical results for vanilla AT align with those reported in the original publications.

The specialized defense mechanisms ATEX and CRC prove least effective against $PO_{\bullet}$ attacks, yielding only marginal improvements in robust accuracy, which remains below $10\,\%$. This outcome is unsurprising, as neither ATEX nor CRC explicitly protects predictions. Notably, ATEX induces almost no drop in the clean accuracy.

Explanation-Preserving Attacks ($EP_{\bullet}$). Next, $EP_{\bullet}$ attacks can disguise an ongoing attack against the predictions from explanation-based analysis. For such attacks, we observe similar trends as already seen for $PO_{\bullet}$ attacks. Vanilla AT and explanation-aware extensions achieve comparable performance across all evaluated metrics. $EP_{\bullet}$ attacks, hence, do not appear more challenging to defend against than $PO_{\bullet}$ attacks. ATEX and CRC fail to improve the robust accuracy and are also not substantially superior in the other metrics, except for ATEX, which again maintains a high clean accuracy.

Prediction-Preserving Attacks (PP). The most pronounced performance differences emerge in the context of these PP attacks. For the $PP^{\square}$ attack scenario, X-PGD-AT, X-MART, RAR, ATEX, and CRC prove ineffective, exhibiting high fooling rates and low forgeability scores. *Note how a low forgeability score corresponds to cases where the target explanation can be reached more closely.* The remaining methods (PGD-AT, TRADES, MART, X-TRADES) demonstrate superior performance, with PGD-AT, TRADES, and MART achieving a fooling rate of $0\,\%$, even though they have never been trained on attacks against explanations. These three methods induce only minimal alterations to the clean explanations, as evidenced by their low fidelity scores (≤ 0.047) and low vulnerability values (≤ 0.013).

We observe similar trends for $PP^{\vdots}$ attacks. While X-PGD-AT, X-MART, RAR, ATEX, and CRC fail to prevent manipulation of explanations, PGD-AT, TRADES, and MART consistently achieve strong results with low vulnerability (≤ 0.023) and low fidelity (≤ 0.047). As expected, the robust accuracy of all methods approaches $100\,\%$ for PP attacks, since the adversary optimizes toward the ground-truth label.

Dual Attacks (D). In both D attack scenarios, the robust accuracy of explanation-aware variants is consistently lower than that of their vanilla counterparts. The explanation-aware variants also exhibit slightly worse forgeability compared to vanilla AT methods. The specialized methods ATEX and CRC perform worst against D attacks.

Overall Assessment. Across all scenarios, ATEX consistently outperforms CRC. Notably, AT techniques surpass both ATEX and CRC in nearly all metrics and attacks. Surprisingly, vanilla AT techniques frequently even outperform their explanation-aware counterparts, while being substantially cheaper and extensively studied in literature. Explanation-aware extensions achieve superior performance only in terms of clean accuracy, regularly exceeding $80\,\%$. Overall, vanilla AT is superior in defending against explanation-aware attacks.

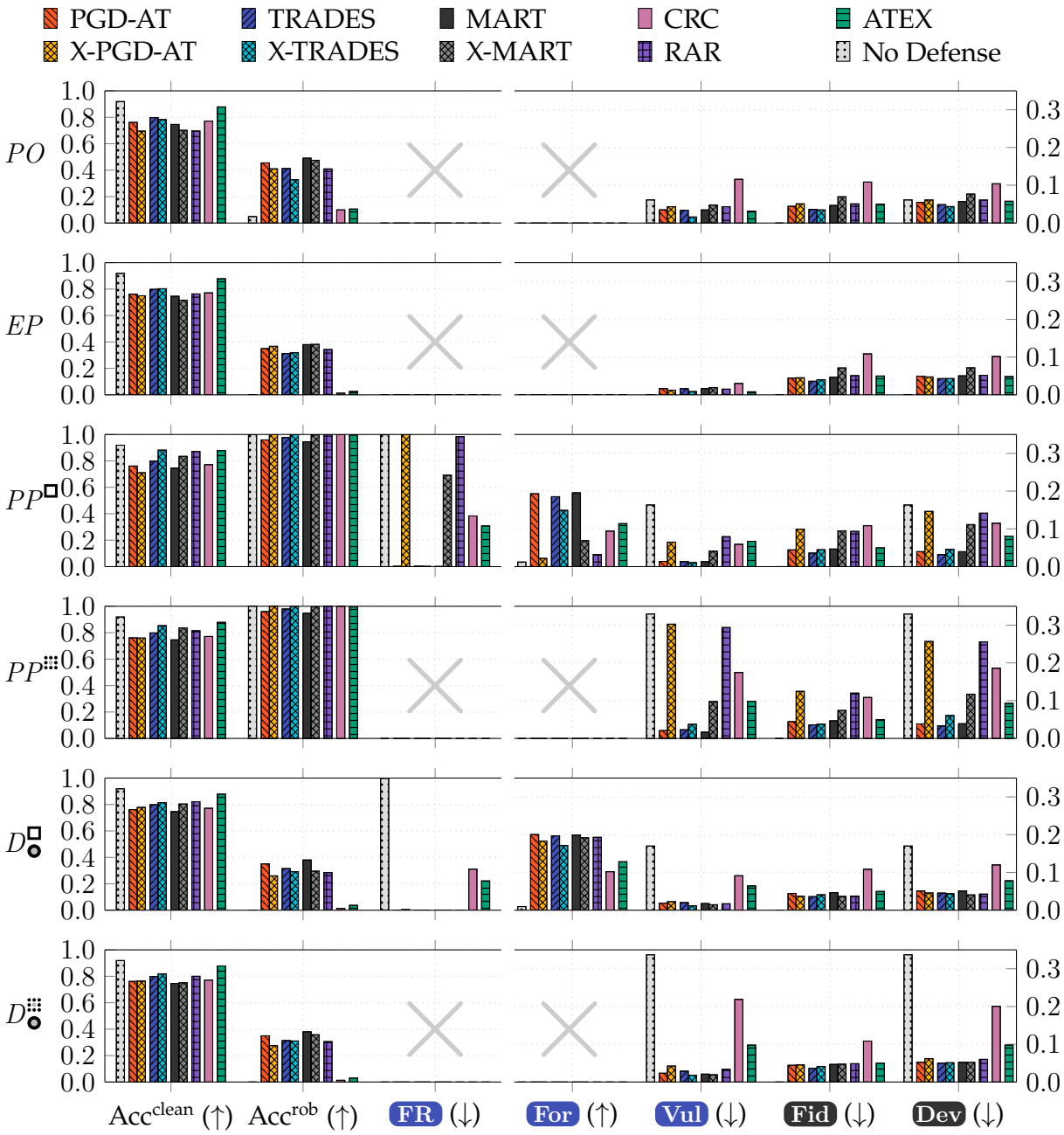


Figure 6.6: The results of the five existing defenses and the explanation-aware extensions. Clean accuracy ($\text{Acc}^{\text{clean}}$), robust accuracy ($\text{Acc}^{\text{rob}}$), and fooling rate (FR) are with regard to the left y-axis while forgeability (For), vulnerability (Vul), fidelity (Fid), and deviation (Dev) are with regard to the right y-axis. Tables D.6 and D.7 contains the quantitative results to this figure.

6.5 Cost Analysis

Explanation-aware AT incurs substantially higher computational costs compared to vanilla AT. The complete replication of our experimental evaluation in this chapter requires approximately 16,320 GPU hours on NVIDIA A100 GPUs.

We now present a comparative analysis of the computational costs across the investigated defenses. Figure 6.7 illustrates the wall-clock time requirements for each approach, measured on a dedicated AMD Ryzen 9 5900X 12-Core processor equipped with two NVIDIA RTX 3090Ti GPUs. The total execution time comprises three components: adversarial example generation (▨), model training time (▩), and auxiliary operations (□). For vanilla AT and explanation-aware AT methods, adversarial sample generation is parallelized across both GPUs. The explanation-aware approaches exhibit approximately 6-fold computational overhead compared to vanilla AT when utilizing 200 PGD iterations, with this disparity further increasing for 500 iterations. This performance degradation is attributable to two primary factors: (1) the increased complexity of generating explanation-aware adversarial examples necessitates additional iterations, and (2) each iteration requires explicit computation of the explanation method and back-propagation of the explanation loss term.

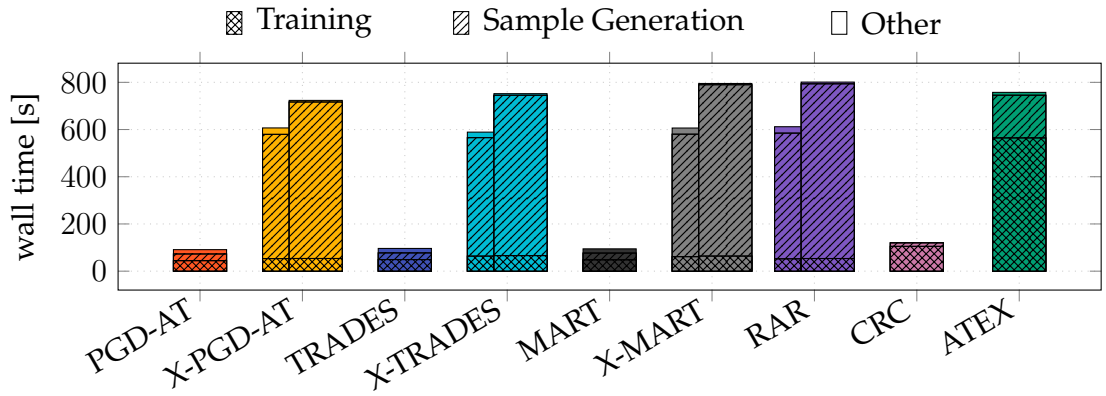


Figure 6.7: Computational costs, in wall time in seconds. For each evaluated approach, the total time is split into the time required for generating attack samples, the actual training, and other functionality. For explanation-aware AT methods, the displayed data is for the $D_{\square}^{\square}$ attacks with 200 (left bars) and 500 PGD iterations (right bars). Other attack scenarios yield similar results.

Results. The vanilla AT techniques demonstrate the lowest computational overhead. CRC exhibits only moderately elevated computational costs relative to vanilla AT. The explanation-aware AT methods incur substantially higher computational expenses, with wall-clock times up to 6 times greater than their vanilla counterparts. This overhead is predominantly concentrated in the generation of explanation-aware adversarial samples. Additionally, ATEX incurs particularly high computational costs, exceeding 757 seconds per epoch. The substantial computational burden of ATEX is primarily attributable to its sample generation procedure, which precludes parallelization across GPUs.

6.6 Transferability Studies

Explanation-aware adversarial training requires the defender to anticipate the attacker's target explanation and their goal. In this section, we investigate what happens if one of those two assumptions is broken, i.e., we investigate if a certain transferability can be observed between two maximally different target explanations in Subsection 6.6.1 and between adversarial goals in Subsection 6.6.2.

6.6.1 Transferability across Target Explanations

We investigate how our explanation-aware extensions depend on the specific target explanation anticipated during fine-tuning. To this end, we conduct a transferability study using two maximally different target explanations: one where the left half is activated and one where the right half is activated. For both target explanations, we train three models per defensive approach, visualize the averaged results in Figure 6.8. The top half of the figure shows models trained on dual attacks with the left-half target explanation; for each method, the two bars show evaluation against attacks with the left-half target (first bar) and the right-half target (second bar). The bottom half shows models trained on the right-half target and evaluated on both right-half (first bar) and left-half (second bar) attacks. If both bars for a method are similar, this indicates high transferability between target explanations. Quantitative results are reported in Table D.5 for MSE and in Tables D.11 and D.12 for the PCC and SSIM metrics, respectively.

Across all methods and metrics, we observe that the transferability is very high: models trained on one target explanation are also robust against attacks using the opposite target. This trend holds for all four explanation-aware adversarial training methods and remains consistent across all evaluated metrics.

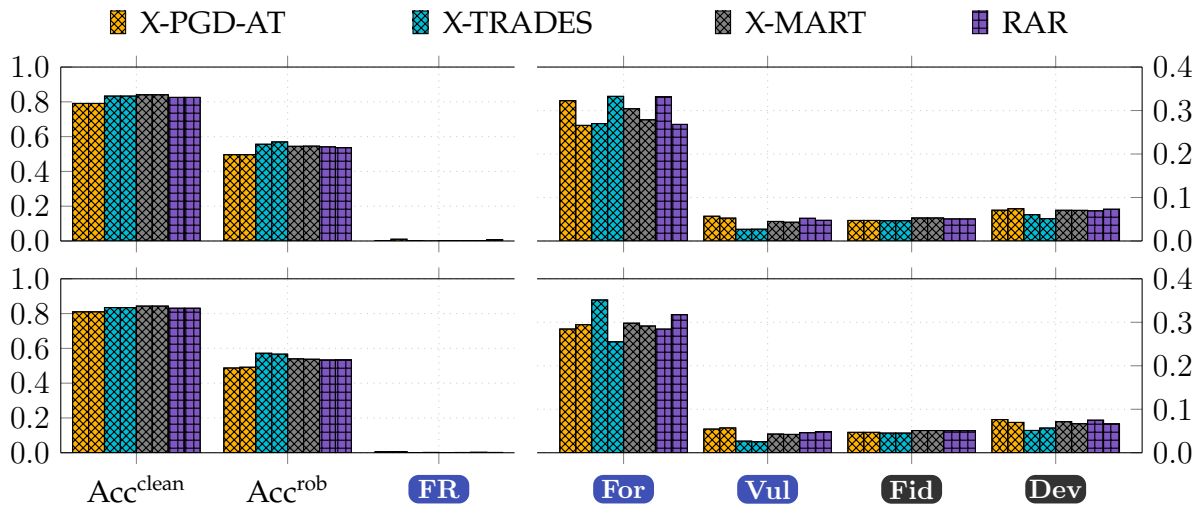


Figure 6.8: Transferability analysis of our three explanation-aware extensions for different target explanations. The clean accuracy (Acc^{clean}), robust accuracy (Acc^{rob}), and the fooling rate are with regard to the left y-axis, while the rest is with regard to the right y-axis. The quantitative results are presented in Table D.5.

6.6.2 Transferability across Adversarial Goals

Tables 6.4 and 6.5 show how training against a specific adversarial goal affects robustness to other adversarial goals. When training on $PO_{\bullet}$ or $EP_{\bullet}$ attacks, the resulting models are robust against both their anticipated attack scenario and other scenarios. While training on $PO_{\bullet}$ attacks yields slightly better results for explanations, the accuracy of models trained on these attacks is significantly lower. The same holds true for training with $EP_{\bullet}$ attacks; however, the differences for explanations and accuracy are smaller. For PP attacks ($PP^{\square}$ and $PP^{\dots}$), the fine-tuning does not improve the robust accuracy for unanticipated attack scenarios. This result is unsurprising since PP attacks do not manipulate the predictions. Our hyperparameter search for the explanation-aware variants (except for X-TRADES) does

Table 6.4: Transferability analysis of explanation-aware defenses across distinct attack scenarios. The best performance is emphasized in bold font, evaluated from the defender's perspective relative to the actual attack scenario (column one). This table presents all attack scenarios without target explanations; consequently, the explanation forgeability metric and fooling rates are omitted. Explanation-targeted scenarios are detailed in Table 6.5. For prediction-only attacks (PO), the adversary disregards explanations; therefore, the columns with Vul and Dev badges contain no bold emphasis.

Att.	Def.	Acc^{rob}	Vul	Dev	Acc^{rob}	Vul	Dev	Acc^{rob}	Vul	Dev	Acc^{rob}	Vul	Dev
$PO_{\bullet}$	$PO_{\bullet}$	$0.41_{0.02}$	$0.04_{0.05}$	$0.06_{0.04}$	$0.33_{0.02}$	$0.02_{0.02}$	$0.04_{0.03}$	$0.48_{0.02}$	$0.05_{0.05}$	$0.08_{0.04}$	$0.41_{0.02}$	$0.04_{0.05}$	$0.06_{0.04}$
	$EP_{\bullet}$	$0.40_{0.02}$	$0.04_{0.05}$	$0.06_{0.04}$	$0.37_{0.02}$	$0.02_{0.03}$	$0.05_{0.03}$	$0.46_{0.02}$	$0.05_{0.06}$	$0.08_{0.05}$	$0.39_{0.02}$	$0.05_{0.05}$	$0.06_{0.04}$
	$PP^{\square}$	$0.16_{0.02}$	$0.08_{0.04}$	$0.12_{0.03}$	$0.10_{0.01}$	$0.04_{0.04}$	$0.05_{0.04}$	$0.22_{0.02}$	$0.09_{0.05}$	$0.11_{0.04}$	$0.09_{0.01}$	$0.08_{0.04}$	$0.12_{0.05}$
	$PP^{\dots}$	$0.13_{0.03}$	$0.13_{0.06}$	$0.17_{0.05}$	$0.21_{0.02}$	$0.03_{0.02}$	$0.05_{0.03}$	$0.22_{0.02}$	$0.08_{0.05}$	$0.09_{0.04}$	$0.12_{0.02}$	$0.12_{0.06}$	$0.16_{0.05}$
	$D^{\square}_{\bullet}$	$0.33_{0.01}$	$0.03_{0.03}$	$0.05_{0.03}$	$0.36_{0.02}$	$0.02_{0.03}$	$0.05_{0.03}$	$0.38_{0.02}$	$0.03_{0.03}$	$0.05_{0.03}$	$0.34_{0.02}$	$0.03_{0.03}$	$0.05_{0.03}$
	$D^{\dots}_{\bullet}$	$0.34_{0.02}$	$0.04_{0.04}$	$0.05_{0.04}$	$0.37_{0.02}$	$0.02_{0.03}$	$0.05_{0.03}$	$0.45_{0.02}$	$0.04_{0.05}$	$0.06_{0.04}$	$0.37_{0.02}$	$0.04_{0.04}$	$0.06_{0.04}$
$EP_{\bullet}$	$PO_{\bullet}$	$0.28_{0.02}$	$0.02_{0.03}$	$0.05_{0.04}$	$0.22_{0.02}$	$0.01_{0.01}$	$0.04_{0.03}$	$0.36_{0.02}$	$0.02_{0.04}$	$0.07_{0.04}$	$0.28_{0.02}$	$0.02_{0.03}$	$0.05_{0.04}$
	$EP_{\bullet}$	$0.37_{0.01}$	$0.01_{0.02}$	$0.05_{0.03}$	$0.32_{0.02}$	$0.01_{0.01}$	$0.04_{0.03}$	$0.38_{0.01}$	$0.02_{0.04}$	$0.07_{0.05}$	$0.34_{0.02}$	$0.02_{0.02}$	$0.05_{0.04}$
	$PP^{\square}$	$0.00_{0.00}$	$0.01_{0.01}$	$0.10_{0.04}$	$0.03_{0.01}$	$0.02_{0.03}$	$0.05_{0.04}$	$0.13_{0.01}$	$0.04_{0.04}$	$0.10_{0.04}$	$0.00_{0.00}$	$0.01_{0.01}$	$0.09_{0.04}$
	$PP^{\dots}$	$0.00_{0.00}$	$0.00_{0.01}$	$0.13_{0.05}$	$0.15_{0.02}$	$0.01_{0.01}$	$0.04_{0.02}$	$0.14_{0.01}$	$0.03_{0.03}$	$0.08_{0.04}$	$0.00_{0.00}$	$0.01_{0.01}$	$0.12_{0.05}$
	$D^{\square}_{\bullet}$	$0.26_{0.02}$	$0.01_{0.02}$	$0.04_{0.03}$	$0.29_{0.02}$	$0.01_{0.01}$	$0.04_{0.03}$	$0.30_{0.02}$	$0.01_{0.02}$	$0.04_{0.03}$	$0.28_{0.02}$	$0.01_{0.02}$	$0.04_{0.03}$
	$D^{\dots}_{\bullet}$	$0.27_{0.02}$	$0.02_{0.02}$	$0.05_{0.03}$	$0.31_{0.02}$	$0.01_{0.01}$	$0.04_{0.03}$	$0.36_{0.02}$	$0.02_{0.03}$	$0.05_{0.03}$	$0.31_{0.02}$	$0.02_{0.03}$	$0.05_{0.03}$
$PP^{\dots}$	$PO_{\bullet}$	$0.95_{0.01}$	$0.02_{0.04}$	$0.04_{0.04}$	$0.98_{0.00}$	$0.01_{0.02}$	$0.03_{0.03}$	$0.92_{0.01}$	$0.02_{0.04}$	$0.06_{0.04}$	$0.95_{0.01}$	$0.03_{0.04}$	$0.04_{0.04}$
	$EP_{\bullet}$	$0.96_{0.01}$	$0.02_{0.03}$	$0.04_{0.03}$	$0.98_{0.00}$	$0.02_{0.03}$	$0.05_{0.03}$	$0.94_{0.01}$	$0.02_{0.04}$	$0.06_{0.04}$	$0.97_{0.00}$	$0.02_{0.04}$	$0.05_{0.03}$
	$PP^{\square}$	$1.00_{0.00}$	$0.20_{0.05}$	$0.17_{0.05}$	$1.00_{0.00}$	$0.04_{0.03}$	$0.06_{0.04}$	$1.00_{0.00}$	$0.10_{0.07}$	$0.14_{0.06}$	$1.00_{0.00}$	$0.24_{0.07}$	$0.21_{0.07}$
	$PP^{\dots}$	$1.00_{0.00}$	$0.30_{0.08}$	$0.26_{0.07}$	$1.00_{0.00}$	$0.04_{0.04}$	$0.06_{0.04}$	$1.00_{0.00}$	$0.10_{0.07}$	$0.12_{0.06}$	$1.00_{0.00}$	$0.29_{0.08}$	$0.26_{0.07}$
	$D^{\square}_{\bullet}$	$0.99_{0.00}$	$0.04_{0.05}$	$0.05_{0.04}$	$0.99_{0.00}$	$0.02_{0.03}$	$0.05_{0.03}$	$0.99_{0.00}$	$0.02_{0.03}$	$0.04_{0.03}$	$0.99_{0.00}$	$0.03_{0.04}$	$0.04_{0.03}$
	$D^{\dots}_{\bullet}$	$0.98_{0.00}$	$0.04_{0.05}$	$0.05_{0.04}$	$0.99_{0.00}$	$0.02_{0.03}$	$0.05_{0.03}$	$0.96_{0.01}$	$0.02_{0.03}$	$0.04_{0.03}$	$0.99_{0.00}$	$0.03_{0.04}$	$0.05_{0.04}$
$D^{\dots}_{\bullet}$	$PO_{\bullet}$	$0.28_{0.02}$	$0.03_{0.04}$	$0.06_{0.04}$	$0.22_{0.02}$	$0.01_{0.02}$	$0.04_{0.03}$	$0.36_{0.02}$	$0.03_{0.04}$	$0.07_{0.04}$	$0.28_{0.02}$	$0.03_{0.04}$	$0.06_{0.04}$
	$EP_{\bullet}$	$0.37_{0.01}$	$0.02_{0.03}$	$0.05_{0.04}$	$0.32_{0.02}$	$0.02_{0.02}$	$0.05_{0.04}$	$0.38_{0.01}$	$0.03_{0.04}$	$0.08_{0.05}$	$0.34_{0.02}$	$0.02_{0.03}$	$0.06_{0.04}$
	$PP^{\square}$	$0.00_{0.00}$	$0.22_{0.05}$	$0.20_{0.05}$	$0.03_{0.01}$	$0.07_{0.05}$	$0.09_{0.05}$	$0.13_{0.01}$	$0.14_{0.08}$	$0.16_{0.07}$	$0.01_{0.00}$	$0.27_{0.06}$	$0.25_{0.07}$
	$PP^{\dots}$	$0.00_{0.00}$	$0.32_{0.07}$	$0.28_{0.07}$	$0.15_{0.02}$	$0.05_{0.05}$	$0.07_{0.05}$	$0.14_{0.01}$	$0.13_{0.08}$	$0.14_{0.08}$	$0.00_{0.00}$	$0.32_{0.07}$	$0.28_{0.07}$
	$D^{\square}_{\bullet}$	$0.26_{0.02}$	$0.05_{0.05}$	$0.06_{0.05}$	$0.29_{0.02}$	$0.02_{0.03}$	$0.05_{0.04}$	$0.30_{0.02}$	$0.02_{0.03}$	$0.05_{0.03}$	$0.29_{0.02}$	$0.03_{0.04}$	$0.05_{0.04}$
	$D^{\dots}_{\bullet}$	$0.28_{0.02}$	$0.04_{0.05}$	$0.06_{0.05}$	$0.31_{0.02}$	$0.02_{0.03}$	$0.05_{0.04}$	$0.36_{0.02}$	$0.02_{0.03}$	$0.05_{0.04}$	$0.31_{0.02}$	$0.03_{0.05}$	$0.06_{0.04}$

(a) X-PGD-AT **(b)** X-TRADES **(c)** X-MART **(d)** RAR

not yield satisfying results for PP attacks. This finding is again reflected in the transferability results. However, the explanation robustness of models trained with X-TRADES on PP attacks does not transfer to other attack scenarios. In particular, $PP^{\square}$ models are not robust against $D^{\dots}_{\circ}$ attacks. We observe that training on either $D^{\square}_{\circ}$ or $D^{\dots}_{\circ}$ yields similar robust accuracies against both D attacks.

Overall, the explanation robustness transfers well across different attacks, indicating that, regardless of the anticipated attack, the explanations are generally robust to manipulation. This finding aligns with our observations in Figure 6.6, where vanilla AT methods perform well or even outperform explanation-aware methods against explanation-aware attacks.

Table 6.5: Transferability study of explanation-aware defenses across different adversarial goals. The best results are, again, highlighted in bold according to the defender's perspective on the actual attack scenario (column one). Here, we report the two explanation-targeted scenarios. The non-explanation-targeted scenarios are presented in Table 6.4.

Att.	Def.	Acc^{rob}	FR	For	Vul	Dev
$PP^{\square}$	$PO_{\circ}$	$0.94_{0.01}$	$0.00_{0.00}$	$0.19_{0.02}$	$0.02_{0.03}$	$0.05_{0.03}$
	$EP_{\circ}$	$0.96_{0.00}$	$0.00_{0.00}$	$0.20_{0.03}$	$0.01_{0.03}$	$0.04_{0.03}$
	$PP^{\square}$	$1.00_{0.00}$	$1.00_{0.00}$	$0.02_{0.01}$	$0.07_{0.03}$	$0.15_{0.03}$
	$PP^{\dots}$	$1.00_{0.00}$	$0.99_{0.00}$	$0.04_{0.02}$	$0.08_{0.03}$	$0.14_{0.03}$
	$D^{\square}_{\circ}$	$0.99_{0.00}$	$0.01_{0.01}$	$0.17_{0.03}$	$0.02_{0.03}$	$0.04_{0.03}$
	$D^{\dots}_{\circ}$	$0.98_{0.00}$	$0.02_{0.01}$	$0.16_{0.03}$	$0.02_{0.03}$	$0.04_{0.03}$
$D^{\square}_{\circ}$	$PO_{\circ}$	$0.28_{0.02}$	$0.00_{0.00}$	$0.20_{0.02}$	$0.02_{0.03}$	$0.06_{0.04}$
	$EP_{\circ}$	$0.37_{0.01}$	$0.00_{0.00}$	$0.21_{0.02}$	$0.01_{0.03}$	$0.05_{0.03}$
	$PP^{\square}$	$0.00_{0.00}$	$1.00_{0.00}$	$0.02_{0.01}$	$0.07_{0.03}$	$0.16_{0.03}$
	$PP^{\dots}$	$0.00_{0.00}$	$0.97_{0.01}$	$0.04_{0.02}$	$0.09_{0.03}$	$0.14_{0.03}$
	$D^{\square}_{\circ}$	$0.26_{0.02}$	$0.01_{0.00}$	$0.18_{0.03}$	$0.02_{0.03}$	$0.05_{0.03}$
	$D^{\dots}_{\circ}$	$0.28_{0.02}$	$0.01_{0.00}$	$0.17_{0.03}$	$0.02_{0.03}$	$0.05_{0.03}$

(a) X-PGD-AT

Att.	Def.	Acc^{rob}	FR	For	Vul	Dev
$PP^{\square}$	$PO_{\circ}$	$0.98_{0.00}$	$0.00_{0.00}$	$0.20_{0.02}$	$0.01_{0.02}$	$0.03_{0.02}$
	$EP_{\circ}$	$0.98_{0.00}$	$0.00_{0.00}$	$0.16_{0.02}$	$0.01_{0.01}$	$0.04_{0.02}$
	$PP^{\square}$	$1.00_{0.00}$	$0.00_{0.00}$	$0.15_{0.02}$	$0.01_{0.02}$	$0.05_{0.04}$
	$PP^{\dots}$	$1.00_{0.00}$	$0.03_{0.00}$	$0.13_{0.02}$	$0.01_{0.02}$	$0.04_{0.02}$
	$D^{\square}_{\circ}$	$0.99_{0.00}$	$0.00_{0.00}$	$0.16_{0.02}$	$0.01_{0.02}$	$0.04_{0.02}$
	$D^{\dots}_{\circ}$	$0.99_{0.00}$	$0.00_{0.00}$	$0.15_{0.02}$	$0.01_{0.01}$	$0.04_{0.02}$
$D^{\square}_{\circ}$	$PO_{\circ}$	$0.22_{0.02}$	$0.00_{0.00}$	$0.21_{0.02}$	$0.01_{0.01}$	$0.04_{0.03}$
	$EP_{\circ}$	$0.32_{0.02}$	$0.00_{0.00}$	$0.18_{0.02}$	$0.01_{0.02}$	$0.04_{0.03}$
	$PP^{\square}$	$0.03_{0.01}$	$0.00_{0.00}$	$0.16_{0.02}$	$0.03_{0.04}$	$0.06_{0.04}$
	$PP^{\dots}$	$0.15_{0.02}$	$0.01_{0.00}$	$0.15_{0.03}$	$0.02_{0.02}$	$0.04_{0.02}$
	$D^{\square}_{\circ}$	$0.29_{0.02}$	$0.00_{0.00}$	$0.17_{0.02}$	$0.01_{0.02}$	$0.04_{0.03}$
	$D^{\dots}_{\circ}$	$0.31_{0.02}$	$0.00_{0.00}$	$0.17_{0.02}$	$0.01_{0.02}$	$0.04_{0.03}$

(b) X-TRADES

Att.	Def.	Acc^{rob}	FR	For	Vul	Dev
$PP^{\square}$	$PO_{\circ}$	$0.92_{0.01}$	$0.00_{0.00}$	$0.17_{0.03}$	$0.02_{0.04}$	$0.06_{0.04}$
	$EP_{\circ}$	$0.94_{0.01}$	$0.00_{0.00}$	$0.17_{0.03}$	$0.02_{0.04}$	$0.06_{0.04}$
	$PP^{\square}$	$1.00_{0.00}$	$0.69_{0.04}$	$0.07_{0.03}$	$0.04_{0.03}$	$0.11_{0.04}$
	$PP^{\dots}$	$0.99_{0.00}$	$0.33_{0.02}$	$0.10_{0.03}$	$0.04_{0.03}$	$0.09_{0.04}$
	$D^{\square}_{\circ}$	$0.99_{0.00}$	$0.00_{0.00}$	$0.18_{0.02}$	$0.01_{0.02}$	$0.03_{0.02}$
	$D^{\dots}_{\circ}$	$0.96_{0.01}$	$0.00_{0.00}$	$0.18_{0.02}$	$0.01_{0.03}$	$0.04_{0.03}$
$D^{\square}_{\circ}$	$PO_{\circ}$	$0.36_{0.02}$	$0.00_{0.00}$	$0.18_{0.03}$	$0.02_{0.04}$	$0.07_{0.04}$
	$EP_{\circ}$	$0.38_{0.01}$	$0.00_{0.00}$	$0.17_{0.03}$	$0.02_{0.04}$	$0.07_{0.04}$
	$PP^{\square}$	$0.14_{0.01}$	$0.53_{0.03}$	$0.08_{0.04}$	$0.07_{0.04}$	$0.12_{0.04}$
	$PP^{\dots}$	$0.14_{0.01}$	$0.18_{0.02}$	$0.12_{0.04}$	$0.06_{0.04}$	$0.09_{0.04}$
	$D^{\square}_{\circ}$	$0.30_{0.02}$	$0.00_{0.00}$	$0.19_{0.02}$	$0.02_{0.02}$	$0.04_{0.03}$
	$D^{\dots}_{\circ}$	$0.36_{0.02}$	$0.00_{0.00}$	$0.18_{0.02}$	$0.02_{0.03}$	$0.05_{0.03}$

(c) X-MART

Att.	Def.	Acc^{rob}	FR	For	Vul	Dev
$PP^{\square}$	$PO_{\circ}$	$0.94_{0.01}$	$0.00_{0.00}$	$0.19_{0.02}$	$0.02_{0.03}$	$0.04_{0.03}$
	$EP_{\circ}$	$0.97_{0.00}$	$0.00_{0.00}$	$0.18_{0.03}$	$0.01_{0.03}$	$0.04_{0.03}$
	$PP^{\square}$	$1.00_{0.00}$	$0.98_{0.00}$	$0.03_{0.02}$	$0.08_{0.03}$	$0.14_{0.03}$
	$PP^{\dots}$	$1.00_{0.00}$	$0.99_{0.00}$	$0.04_{0.02}$	$0.08_{0.03}$	$0.14_{0.03}$
	$D^{\square}_{\circ}$	$0.99_{0.00}$	$0.00_{0.00}$	$0.18_{0.02}$	$0.01_{0.02}$	$0.04_{0.02}$
	$D^{\dots}_{\circ}$	$0.99_{0.00}$	$0.01_{0.00}$	$0.16_{0.02}$	$0.01_{0.03}$	$0.04_{0.03}$
$D^{\square}_{\circ}$	$PO_{\circ}$	$0.28_{0.02}$	$0.00_{0.00}$	$0.20_{0.02}$	$0.02_{0.03}$	$0.06_{0.04}$
	$EP_{\circ}$	$0.34_{0.02}$	$0.00_{0.00}$	$0.19_{0.02}$	$0.02_{0.03}$	$0.05_{0.04}$
	$PP^{\square}$	$0.00_{0.00}$	$0.96_{0.00}$	$0.03_{0.02}$	$0.09_{0.03}$	$0.15_{0.03}$
	$PP^{\dots}$	$0.00_{0.00}$	$0.96_{0.01}$	$0.04_{0.02}$	$0.09_{0.03}$	$0.14_{0.03}$
	$D^{\square}_{\circ}$	$0.29_{0.02}$	$0.00_{0.00}$	$0.19_{0.02}$	$0.02_{0.02}$	$0.04_{0.03}$
	$D^{\dots}_{\circ}$	$0.31_{0.02}$	$0.00_{0.00}$	$0.17_{0.02}$	$0.02_{0.03}$	$0.05_{0.03}$

(d) RAR

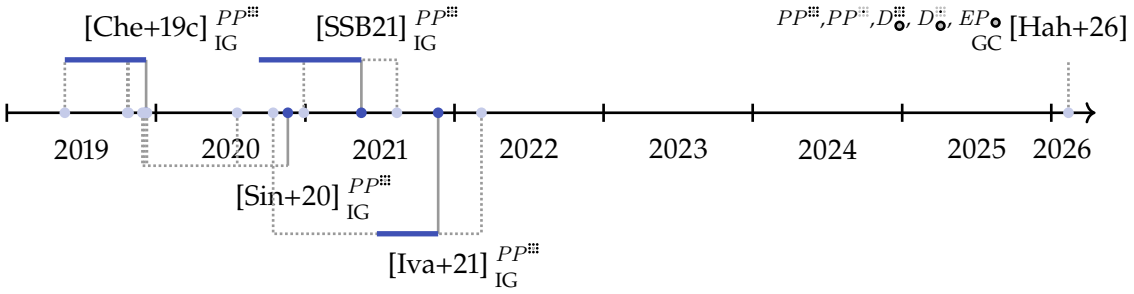


Figure 6.9: Timeline of work related to this chapter. Filled dots (●) indicate publication dates, shaded dots (●) indicate the publication dates of preprints, and bars show the submission-to-publication period, to the best of our knowledge. The considered attack scenarios and the explanation methods are next to the respective work, where IG denotes the Integrated Gradients and GC denotes GradCAM.

6.7 Related Work

In Figure 6.9, we present the timeline of related work, limited to work explicitly on adversarial training. The closest work to ours are four papers that investigate explanation-aware AT for Integrated Gradients [Che+19c; Iva+21; Sin+20; SSB21]. All four have considered the Iterative Feature Importance Attack (IFIA) [GAZ19] ($PP^{\vdots}$) during training. Two of them have additionally considered an adversarial IFIA variant, which corresponds to a $D^{\vdots}_{\circ}$ attack [Che+19c; Iva+21]. However, all four have only evaluated the explanation vulnerability on IFIA only and are thus only marked as such in the above figure. The robust accuracy, in turn, they have measured with a *separate* PO attack [Mad+18a], i.e., the prediction robustness and the explanation vulnerability are evaluated on different attack samples. We speculate that this way of evaluating originates from the focus on PP attacks. In contrast, we evaluate substantially more attack scenarios and, most importantly, always on one set of attack samples only. Furthermore, to the best of our knowledge, no related work on explanation-aware AT has considered the transferability of defenses across different attack scenarios and target explanations, except for the transferability from adversarial IFIA to normal IFIA, as mentioned above.

6.8 Chapter Summary

We present the first extensive evaluation of adversarial training (AT) techniques to counter explanation-aware attacks across all established attack scenarios targeting GradCAM. We also evaluate vanilla AT methods (PGD-AT, TRADES, and MART) as a simple baseline. Moreover, we introduce new explanation-aware extensions for TRADES and MART, and show that FAR [Iva+21] is equivalent to the extension of PGD-AT. Further, we find that CRC and ATEX are ineffective against explanation-aware attacks that target predictions *and* explanations, i.e., dual attacks (D). Note that both methods have been proposed for attacks against explanations only (EO). RAR [Che+19c], in turn, performs similarly to our explanation-aware extension of PGD-AT. In particular, both fail to prevent *explanation-targeted-PP* attacks ($PP^{\vdots}$).

Our work reveals that vanilla adversarial training already provides strong robustness against explanation-aware attacks, and that the additional effort for explanation-aware adversarial training yields little to no benefit right now. With this result, we answer a long-standing question in the field regarding the necessity of explanation-aware adversarial training. However, the underlying reasons for this phenomenon remain unclear and warrant further investigation. We encourage the community to explore the relationship between explanations and predictions, and to analyze which heuristics are exploited by attacks on explanation methods and why these cannot be leveraged in adversarially trained models.

7 Outlook and Conclusion

Contents

In this thesis, we investigate eXplainable Artificial Intelligence (XAI) in adversarial environments. Therefore, we first provide a systematization of the field. Thereafter, we present two new attack families and one new defense to mitigate attacks against explanations.

This final chapter provides a summary of the main results in Section 7.1. Afterwards, the limitations of the thesis are discussed in Section 7.2, and an outlook on future work is provided in Section 7.3. We close this thesis in Section 7.4 with our conclusion.

7.1 Summary of Results

We summarize the main results of this thesis by answering the research questions.

In **Chapter 3** **(Systematization of Explanation-Aware Attacks)**, we answer the research question: (RQ1) *What is the attack surface for manipulating explanations?* We identify the adversarial capabilities: input manipulations, dataset manipulations, model manipulations, and manipulations of an explainable system. We categorize the adversarial goals according to two targets: the prediction and the explanation, where each can either be preserved or altered. A preserved target either is accurate with respect to ground-truth labels or explanations, or maintains high fidelity to a reference model. Alteration is categorized into targeted, untargeted, and (for explanations) semitargeted attacks.

We propose a hierarchy of robustness notions. We show that certification algorithms guaranteeing robustness around a single sample cannot necessarily be expanded to general robustness around all samples. This limitation arises not from computational intractability but from geometric properties of the decision surface.

We then explore approaches to improve the robustness of XAI-systems in practice. These include data and model validation and sanitization, robustification of the training process, operational safeguards such as input validation and continuous monitoring, and methods to enhance the robustness of explanation methods themselves.

In summary, we describe the adversarial landscape and identify current research challenges in the field. Some of them we pick up in the Chapters 4 to 6.

In Chapter 4 **(Explanation-Aware Backdoors)**, we answer the following research question: (RQ2) *To what extent can models be manipulated so that they behave inconspicuously but show adversary-chosen predictions and explanations in the presence of triggers?* We find that adversaries can achieve this manipulation successfully by performing a fine-tuning step after the original training. The loss function during fine-tuning is bi-objective and considers the distance between the generated explanation and the desired explanation for each sample. In the investigated cases, the explanation method is differentiable, and thus this loss function can be optimized via gradient descent on a poisoned dataset. Concretely, we duplicate the entire dataset and poison all duplicates with triggers. Furthermore, each sample in the dataset is augmented with either its clean explanation, if it does not contain a trigger, or the desired explanation according to the adversarial goal. By doing so, the model maintains high accuracy and clean explanations in benign queries (without the trigger) but displays the desired predictions and explanations in the presence of the trigger pattern.

In Chapter 5 **(Composite Attacks)**, we shed light on the following research question: (RQ3) *To what extent can we decouple the attack against the predictions from the attack against the explanations into training-time and inference-time attacks?* The decoupling works by injecting a traditional neural backdoor via data poisoning at training time, followed by inference-time perturbations to forge sample-specific explanations. These perturbations can, for instance, prevent the trigger patch from being highlighted in the explanation. This decoupling is effective in white-box scenarios, where the adversary has access to the model after training. The perturbations, however, do not transfer to other unseen models out of the box. Interestingly, explanation forging works best in perturbed regions and worse in regions where no explanation-aware perturbation is applied, e.g., where the trigger is. This finding suggests an interesting direction for further investigation.

In Chapter 6 **(Explanation-Aware Adversarial Training)**, we address the research question: (RQ4) *To what extent does explanation-aware adversarial training improve a model's robustness against explanation-aware input manipulations?* In explanation-aware adversarial training (AT), the defender performs fine-tuning steps on explanation-aware attacks executed by themself against the own current model. That is, the attack samples are specifically crafted to forge predictions and explanations according to the respective adversarial goal. Additionally, the loss function during fine-tuning is explanation-aware as well, but with respect to explanations desired by the defender and the ground-truth labels.

To answer the research question, we summarize across the investigated scenarios and conclude that explanation-aware adversarial training increases the robust accuracy from 0%, without defense, to $\geq 25\,\%$. At the same time, the clean accuracy drops by approximately 5 to 21 percentage points. Attacks that aim to yield a specific target explanation, such as a square, no longer work: the fooling rate drops from $100\,\%$ on normally-trained models to approximately $0\,\%$ on robustified models. Similarly, explanation-untargeted attacks that maximize the deviation of explanations from clean explanations are no longer successful.

We find that, even though explanation-aware adversarial training specifically considers explanation-aware attacks, traditional adversarial training approaches achieve similar results and require substantially fewer computational resources.

7.2 Limitations

In this section, we briefly discuss the limitations of our work and point to future work.

Experimental Scope. While our experiments in Chapter 4 are performed for three datasets from two different domains, we restrict our experiments in Chapter 5 and Chapter 6 to the image domain and the CIFAR-10 dataset. We consider image recognition an ideal case to study due to its human-understandable domain and yet complex nature. Images allow us to visualize adversarial goals clearly. While a broader analysis across more datasets and domains is naturally future work, we expect the presented concepts and attacks to generalize to other domains, including continuous data such as video and audio streams. Initial explanation-aware attacks have also been successfully demonstrated by prior work on text [e.g., Ali+22; Cam+19], despite its discrete nature.

Additionally, we have already demonstrated explanation-aware backdoors for tabular data [HNW24] and preliminarily for code models in the context of a thesis [Pet23].

Similar limitations apply to the selection of explanation methods. This thesis restricts itself to Simple Gradients, GradCAM, Relevance-CAM, and SmoothGrad, with a strong focus on GradCAM, which is highly cited and widely adopted due to its smooth visual appearance and low computational overhead. However, we expect the presented attacks to generalize to other explanation methods as well. For instance, we have demonstrated explanation-aware backdoors for LIME [RSG16] and SHAP [LL17] as well [HNW24]. Furthermore, in the context of various theses [Pet23; Lai24; Bal24], we have preliminarily demonstrated explanation-aware backdoors for the transformer-specific explanation methods Attention Rollout [AZ20] and Gradient Rollout [Gil20], the counterfactual methods Wachter's method [WMR17], Constrained Adversarial Example (CADEX) [MHW19], and Diverse Counterfactual Explanations (DiCE) [MST20], and for the prototype-based self-explainable neural network ProtoPNet [Che+19b], respectively. Similarly, Baniecki and Biecek [BB25] present a more in-depth investigation of explanation-aware backdoors against ProtoPNet.

Overall, we thus expect the attack approaches and concepts presented in this thesis to generalize with slight adaptation to more domains, datasets, explanation methods, and model architectures. Yet, we suggest further investigation as future work.

Target Explanations. Similarly to our experimental scope, our target explanations serve as a showcase that almost any arbitrary target explanation can be achieved. On the one hand, our selected targets exhibit clearly visible and recognizable structures, making the attack effectiveness readily apparent. On the other hand, they are intentionally not structurally similar to benign explanations. In particular, plausible benign explanations usually do not have holes and steep corners. With our target explanations, we thus demonstrate that even such dissimilar target explanations can be reached. It is clear, though, that target explanations depend on the concrete attack scenario and the adversarial intentions, which remain difficult to anticipate in practice. The investigation of further target explanations for specific attack scenarios therefore remains future work.

7.3 Outlook

The robustness of eXplainable Artificial Intelligence (XAI) against intentional worst-case perturbations is crucial for its adoption in adversarial environments. Assessing the criticality in a concrete scenario depends on two factors: the severity when the model or the explanation is misbehaving and the likelihood of successful attacks. The latter, in turn, depends on the system's robustness, which is dependent on the task's complexity and the attacker's incentives and manipulation capabilities. While this thesis extensively discusses attack surfaces and adversarial goals, mapping these to concrete real-world applications and scenarios remains future work, requiring close collaboration with practitioners.

Bridging Research and Practice. While various attacks and defensive mechanisms have been proposed in academic settings, their respective criticality and transferability to real-world environments remain open questions. Nevertheless, complementary awareness studies are needed to assess how aware practitioners and human analysts are of the insecurity of AI systems and the unreliability of XAI. This awareness assessment is necessary not only regarding intentional attacks but also distribution shifts and stability issues, as discussed, for example, by Leofante and Wicker [LW25]. User studies are equally necessary to understand how forged explanations really deceive human analysts and what constitutes "similar" explanations beyond mathematical metrics Although the Structural Similarity Index (SSIM) specifically considers human perception, its underlying assumptions fit natural images rather than explanations, a crucial difference.

Defending Based on Explanation Quality. Further developments are required to defend explainable systems against intentional worst-case perturbations. One promising and underexplored direction is leveraging explanation quality measures. Our square target explanation, for instance, is not a "good" explanation. It is detached from the model's true decision-making process. Established quality metrics, such as descriptive accuracy [War+20] (also commonly referred to as pixel flipping [Bac+15]), should assign low scores to such "wrong" explanations, potentially distinguishing them from benign ones. The challenge herein is that quality metrics are designed to evaluate explanation methods, not individual explanations; they average across samples to reduce noise. Whether they can detect individual attack samples remains an open question. Similarly, it remains open whether adaptive attackers can circumvent such defenses based on explanation-quality metrics.

Connections to Explanation-Guided Learning. Prior work on *explanation-guided learning* [Zha+24b; Gao+24; BCD24; Gao+22b; Gao+22a; HB22; Rie+20; RHD17] employs techniques similar to those presented in this thesis. Models are trained (or fine-tuned) with bi-objective loss functions considering both predictions and explanations. In explanation-guided learning, however, the augmented explanations are commonly based on human annotations, guiding the model away from shortcuts and toward a reasoning that is aligned with human ontology. In light of this thesis, it remains questionable whether such models truly ignore unwanted correlations or whether the explanations are simply forged. More broadly, the exact coupling between a model's decision surface and its explanation surface remains underexplored. We encourage researchers to investigate this coupling further.

7.4 Conclusion

eXplainable Artificial Intelligence (XAI) has emerged as an essential toolbox for understanding and debugging AI systems, benefiting both developers and end users. Developers can leverage explanations to identify issues in the training data, such as biases, spurious correlations, or shortcuts. End users can interact more effectively with AI systems, e.g., by removing environmental distractions. However, the initial hope that explanation methods could reliably detect attacks has been overly optimistic.

As highlighted by this thesis, XAI is not reliable in adversarial environments. Attackers can manipulate models or inputs to forge explanations and predictions alike. Such explanation-aware attacks share similarities to traditional attacks against predictions but additionally forge explanations. This seemingly small addition increases the variety of attacks drastically. Unlike predictions, which are typically low-dimensional or discrete scalars, explanations are high-dimensional objects. This fact gives rise to substantially more adversarial goals, ranging from disguising ongoing attacks (against the predictions) to generating misleading explanations in order to hamper analysis efforts on the victim's side. Due to this variety, the XAI security landscape is difficult to analyze.

This thesis provides a first building block toward navigating this complexity. It establishes a hierarchy of robustness notions and systematizes explanation-aware attacks based on adversarial capabilities, constraints, and goals. Along this systematization, we empirically demonstrate that XAI should not yet be relied upon for defense. Specifically, our presented explanation-aware backdoors can bypass defensive systems that rely on explanations to detect attacks, such as those that assume explanations to highlight triggers of backdooring attacks. Similarly, our composite attacks can disguise triggers of traditional backdooring attacks at inference time for each sample independently.

In light of this forgeability of XAI, using unreliable explanation methods in adversarial environments can result in a false sense of security. Plausible explanations might easily be treated as indicators of correct predictions, a habit that can be exploited by attackers.

To mitigate this problem, this thesis presents a new defense technique based on adversarial training. We demonstrate that applying our defense increases the robustness of the system with respect to both predictions and explanations. While we consider the development of empirical defenses and attacks an important step in the right direction, we believe that in order to escape this arms race it is necessary to move from patching existing unreliable methods toward a security-by-design approach. For instance, by developing inherently explainable yet highly performant architectures, or by designing methods with robustness guarantees or, at minimum, quantified uncertainty. While the research field on counterfactual explanations is already moving in this direction, research on feature-attribution methods lags behind.

In conclusion, this thesis advances the understanding of the XAI security landscape. It demonstrates that current explanation methods are vulnerable to manipulation and establishes a framework for reasoning about their robustness. We hope that this thesis helps developing robust and reliable explanation methods for use in adversarial environments.

Bibliography

[Abd+21] Eldor Abdukhamidov et al. 'AdvEdge: Optimizing Adversarial Perturbations against Interpretable Deep Learning'. In: *Proc. of the International Conference on Computational Data and Social Networks (CSoNet)*. Vol. 13116. 2021, pp. 93–105 (cited on page 41).

[Abd+22] Eldor Abdukhamidov et al. 'Interpretations Cannot Be Trusted: Stealthy and Effective Adversarial Perturbations against Interpretable Deep Learning'. In: *CoRR* abs/2211.15926 (2022) (cited on pages 28, 41, 101, 106).

[Abd+25] Eldor Abdukhamidov et al. 'Breaking the Illusion of Security via Interpretation: Interpretable Vision Transformer Systems under Attack'. In: *CoRR* (2025) (cited on page 41).

[AZ20] Samira Abnar and Willem Zuidema. *Quantifying Attention Flow in Transformers*. 2020 (cited on page 127).

[Ach+12] Radhakrishna Achanta et al. 'SLIC Superpixels Compared to State-of-the-Art Superpixel Methods'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (2012), pp. 2274–2282 (cited on pages 15, 16).

[Ach+24] Reduan Achtibat et al. 'AttnLRP: Attention-aware Layer-Wise Relevance Propagation for Transformers'. In: *Proc. of the International Conference on Machine Learning (ICML)*. 2024 (cited on page 20).

[AB18] Amina Adadi and Mohammed Berrada. 'Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)'. In: *IEEE Access* 6 (2018), pp. 52138–52160 (cited on pages 1, 12, 23).

[Ade+18] Julius Adebayo et al. 'Sanity Checks for Saliency Maps'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2018, pp. 9525–9536 (cited on pages 65, 69, 87).

[ANW25] Luan Ademi, Maximilian Noppel, and Christian Wressnegger. 'POMELO: Black-box Feature Attribution with Full-Input, in-Distribution Perturbations'. In: *Proc. of the World Conference on eXplainable Artificial Intelligence (XAI)*. 2025 (cited on pages 5, 8, 16, 17).

[Agh+24] Hojjat Aghakhani et al. 'TrojanPuzzle: Covertly Poisoning Code-Suggestion Models'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2024 (cited on page 5).

[Aïv+19] Ulrich Aïvodji et al. 'Fairwashing: The Risk of Rationalization'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 97. 2019, pp. 161–170 (cited on pages 13, 26, 33–35, 41, 88).

[Aïv+21] Ulrich Aïvodji et al. 'Characterizing the Risk of Fairwashing'. In: *CoRR* abs/2106.07504 (2021) (cited on pages 13, 33, 41).

[Aja+22] Ahmad Ajalloeian et al. 'On Smoothed Explanations: Quality and Robustness'. In: *Proc. of ACM International Conference on Information and Knowledge Management (CIKM)*. 2022, pp. 15–25 (cited on pages 28, 38, 57).

[Aki+19] Takuya Akiba et al. 'Optuna: A Next-generation Hyperparameter Optimization Framework'. In: *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2019 (cited on pages 182, 183).

[Ali+22] Hassan Ali et al. 'Tamp-X: Attacking Explainable Natural Language Classifiers Through Tampered Activations'. In: *Comput. Secur.* 120 (2022) (cited on pages 40, 127).

[AJ18a] David Alvarez-Melis and Tommi S. Jaakkola. 'On the Robustness of Interpretability Methods'. In: *Proc. of the ICML Workshop on Human Interpretability in Machine Learning (WHI)* (2018) (cited on pages 41, 45, 113).

[AJ18b] David Alvarez-Melis and Tommi S. Jaakkola. 'Towards Robust Interpretability with Self-Explaining Neural Networks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2018, pp. 7786–7795 (cited on page 11).

[And24] Christopher J. Anders. 'Debugging Learning Algorithms: Understanding and Correcting Machine Learning Models'. PhD thesis. Technical University of Berlin, Germany, 2024 (cited on page 2).

[And+20] Christopher J. Anders et al. 'Fairwashing Explanations with Off-Manifold Detergent'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 119. 2020, pp. 314–323 (cited on pages 26, 34, 41, 57, 65, 88).

[And+22] Christopher J. Anders et al. 'Finding and Removing Clever Hans: Using Explanation Methods to Debug and Improve Deep Models'. In: *Information Fusion* 77 (2022), pp. 261–295 (cited on pages 2, 12, 50).

[Arp+14] Daniel Arp et al. 'Drebin: Effective and Explainable Detection of Android Malware in Your Pocket'. In: *Proc. of the Network and Distributed System Security Symposium (NDSS)*. 2014 (cited on pages 1, 63, 82).

[Arp+22] Daniel Arp et al. 'Dos and Don'ts of Machine Learning in Computer Security'. In: *Proc. of the USENIX Security Symposium*. 2022, pp. 3971–3988 (cited on pages 1, 2, 26, 82).

[Arr+17] Leila Arras et al. 'Explaining Recurrent Neural Network Predictions in Sentiment Analysis'. In: *Proc. of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 2017, pp. 159–168 (cited on page 20).

[Ary+19] Vijay Arya et al. 'One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques'. In: *CoRR* abs/1909.03012 (2019) (cited on page 23).

[Ata+24] Shahin Atakishiyev et al. 'Explainable Artificial Intelligence for Autonomous Driving: A Comprehensive Overview and Field Guide for Future Research Directions'. In: *IEEE Access* 12 (2024), pp. 101603–101625 (cited on page 1).

[AL25] Zahra Atf and Peter R. Lewis. 'Is Trust Correlated with Explainability in AI? A Meta-Analysis'. In: *CoRR* abs/2504.12529 (2025) (cited on page 13).

[Bac+15] Sebastian Bach et al. 'On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation'. In: *PLOS ONE* (2015), p. 46 (cited on pages 1, 11, 20, 22, 58, 105, 128).

[Bae+10] David Baehrens et al. 'How to Explain Individual Classification Decisions'. In: *Journal of Machine Learning Research (JMLR)* 11 (2010), pp. 1803–1831 (cited on pages 18, 93).

[BMV18] Mitali Bafna, Jack Murtagh, and Nikhil Vyas. 'Thwarting Adversarial Examples: An L_0-Robust Sparse Fourier Transform'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2018 (cited on page 27).

[BS21] Eugene Bagdasaryan and Vitaly Shmatikov. 'Blind Backdoors in Deep Learning Models'. In: *Proc. of the USENIX Security Symposium* (2021) (cited on pages 2, 13, 62, 78, 88, 91, 103).

[Bal24] Lukas Baldner. 'Explanation-Aware Backdoor Attacks on ProtoPNet'. MA thesis. Karlsruhe Institute of Technology, 2024 (cited on pages 8, 127, 167).

[Bal+17] David Balduzzi et al. 'The Shattered Gradients Problem: If Resnets Are the Answer, Then What Is the Question?' In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 70. 2017, pp. 342–350 (cited on pages 18, 19).

[BB24] Hubert Baniecki and Przemyslaw Biecek. 'Adversarial Attacks and Defenses in Explainable Artificial Intelligence: A Survey'. In: *Information Fusion*. 2024 (cited on pages 25, 27, 59, 91).

[BB25] Hubert Baniecki and Przemyslaw Biecek. 'Birds Look like Cars: Adversarial Analysis of Intrinsically Interpretable Deep Learning'. In: *Machine Learning* 114.12 (2025), p. 284 (cited on pages 40, 127).

[Ban+23] Hubert Baniecki et al. 'Be Careful When Evaluating Explanations Regarding Ground Truth'. In: *CoRR* abs/2311.04813 (2023) (cited on page 106).

[Bar+23] Nuria Rodríguez Barroso et al. 'Survey on Federated Learning Threats: Concepts, Taxonomy on Attacks and Defences, Experimental Study and Challenges'. In: *Information Fusion* 90 (2023), pp. 148–173 (cited on page 33).

[BCD24] Pedro R. A. S. Bassi, Andrea Cavalli, and Sergio Decherchi. 'Explanation Is All You Need in Distillation: Mitigating Bias and Shortcut Learning'. In: *CoRR* abs/2407.09788 (2024) (cited on pages 12, 128).

[Bec+19] Bernhard Beckert et al. 'GI Elections with POLYAS: A Road to End-to-End Verifiable Elections'. In: *Proc. of the International Joint Conference on Electronic Voting (E-Vote-ID)*. Gesellschaft für Informatik (GI), 2019, pp. 293–294 (cited on page 9).

[Ben+23] Sidney Bender et al. 'Towards Fixing Clever-Hans Predictors with Counterfactual Knowledge Distillation'. In: *CoRR* abs/2310.01011 (2023) (cited on page 2).

[BS23] Ezekiel Bernardo and Rosemary Seva. 'Affective Analysis of Explainable Artificial Intelligence in the Development of Trust in AI Systems'. In: *Intelligent Human Systems Integration (IHSI 2023) Integrating People and Intelligent Systems*. 2023 (cited on page 13).

[BNL12] Battista Biggio, Blaine Nelson, and Pavel Laskov. 'Poisoning Attacks against Support Vector Machines'. In: *Proc. of the International Conference on Machine Learning (ICML)*. 2012 (cited on page 31).

[Bla+22] Tsachi Blau et al. 'Threat Model-Agnostic Adversarial Defense Using Diffusion Models'. In: *CoRR* abs/2207.08089 (2022) (cited on page 55).

[Boj+18] Mariusz Bojarski et al. 'VisualBackProp: Efficient Visualization of CNNs for Autonomous Driving'. In: *Proc. of the International Conference on Robotics and Automation (ICRA)*. 2018, pp. 1–8 (cited on page 88).

[Boo+20] Akhilan Boopathy et al. 'Proper Network Interpretability Helps Adversarial Robustness in Classification'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 119. 2020, pp. 1014–1023 (cited on pages 41, 56, 58, 104, 106).

[BB07] Léon Bottou and Olivier Bousquet. 'The Tradeoffs of Large Scale Learning'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2007, pp. 161–168 (cited on page 68).

[BK21] Davis Brown and Henry Kvinge. 'Brittle Interpretations: The Vulnerability of TCAV and Other Concept-Based Explainability Tools to Adversarial Attack'. In: *CoRR* abs/2110.07120 (2021) (cited on page 28).

[Bro+17] Tom B. Brown et al. 'Adversarial Patch'. In: *CoRR* abs/1712.09665 (2017) (cited on pages 29, 30, 39).

[BH21] Nadia Burkart and Marco F. Huber. 'A Survey on the Explainability of Supervised Machine Learning'. In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 245–317 (cited on pages 1, 15, 103).

[Byk+20] Kirill Bykov et al. 'How Much Can I Trust You? - Quantifying Uncertainties in Explaining Neural Networks'. In: *CoRR* abs/2006.09000 (2020) (cited on page 52).

[Byk+22] Kirill Bykov et al. 'NoiseGrad: Enhancing Explanations by Introducing Stochasticity to Model Weights'. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*. 2022, pp. 6132–6140 (cited on page 57).

[Cam+19] Oana-Maria Camburu et al. 'Make up Your Mind! Adversarial Generation of Inconsistent Natural Language Explanations'. In: *CoRR* abs/1910.03065 (2019) (cited on page 127).

[CBS22] Ginevra Carbone, Luca Bortolussi, and Guido Sanguinetti. 'Resilience of Bayesian Layer-Wise Explanations under Adversarial Attacks'. In: *Proc. of the International Joint Conference on Neural Networks (IJCNN)*. 2022, pp. 1–8 (cited on pages 41, 42, 45, 48, 52).

[Car+20] Ginevra Carbone et al. 'Robustness of Bayesian Neural Networks to Gradient-Based Attacks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2020 (cited on page 52).

[CW17a] Nicholas Carlini and David Wagner. 'Towards Evaluating the Robustness of Neural Networks'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)* (2017), pp. 39–57 (cited on pages 29, 74).

[CW17b] Nicholas Carlini and David Wagner. 'Towards Evaluating the Robustness of Neural Networks'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)* (2017), pp. 39–57 (cited on pages 91, 92, 101).

[CS22] Zachariah Carmichael and Walter J. Scheirer. 'Unfooling Perturbation-Based Post Hoc Explainers'. In: *CoRR* abs/2205.14772 (2022) (cited on page 56).

[Car+15] Rich Caruana et al. 'Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission'. In: *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2015, pp. 1721–1730 (cited on page 1).

[CPC19] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. 'Machine Learning Interpretability: A Survey on Methods and Metrics'. In: *Electronicsweek* 8.8 (2019) (cited on pages 12, 23).

[Cha+19] Chun-Hao Chang et al. 'Explaining Image Classifiers by Counterfactual Generation'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2019 (cited on page 11).

[Cha+18] Aditya Chattopadhyay et al. 'Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks'. In: *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2018, pp. 839–847 (cited on pages 11, 23).

[CSW22] Hila Chefer, Idan Schwartz, and Lior Wolf. 'Optimizing Relevance Maps of Vision Transformers Improves Robustness'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2022 (cited on page 32).

[Che+19a] Bryant Chen et al. 'Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering'. In: *Proc. of the Workshop on Artificial Intelligence Safety Co-Located with the AAAI Conference on Artificial Intelligence (SafeAI@AAAI)*. Vol. 2301. 2019 (cited on pages 50, 51, 88).

[Che+19b] Chaofan Chen et al. 'This Looks like That: Deep Learning for Interpretable Image Recognition'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2019, pp. 8928–8939 (cited on pages 11, 127).

[Che+19c] Jiefeng Chen et al. 'Robust Attribution Regularization'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2019, pp. 14300–14310 (cited on pages 4, 41, 48, 54, 58, 102, 103, 106, 112, 113, 115, 123).

[Che+21] Valerie Chen et al. 'Towards Connecting Use Cases and Methods in Interpretable Machine Learning'. In: *CoRR* abs/2103.06254 (2021) (cited on pages 12, 23).

[Che+17] Xinyun Chen et al. 'Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning'. In: *CoRR* abs/1712.05526 (2017) (cited on pages 2, 29, 70, 91).

[Cho+24] Pattarawat Chormai et al. 'Disentangled Explanations of Neural Network Predictions by Finding Relevant Subspaces'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.11 (2024), pp. 7283–7299 (cited on page 11).

[CTP18] Edward Chou, Florian Tramèr, and Giancarlo Pellegrino. 'SentiNet: Detecting Physical Attacks against Deep Learning Systems'. In: *CoRR* abs/1812.00292 (2018) (cited on pages 80, 172).

[CTP20] Edward Chou, Florian Tramèr, and Giancarlo Pellegrino. 'SentiNet: Detecting Localized Universal Attacks against Deep Learning Systems'. In: *Proc. of the IEEE Symposium on Security and Privacy Workshops*. 2020, pp. 48–54 (cited on pages 2, 4, 13, 50, 55, 63, 78, 79, 88, 98).

[Chu+16] Xu Chu et al. 'Data Cleaning: Overview and Emerging Challenges'. In: *Proc. of the International Conference on Management of Data (SIGMOD)*. 2016, pp. 2201–2206 (cited on page 50).

[Chu+23] Yu-Neng Chuang et al. 'Efficient XAI Techniques: A Taxonomic Survey'. In: *CoRR* arXiv:2302.03225 (2023) (cited on pages 12, 23).

[CRK19] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. 'Certified Adversarial Robustness via Randomized Smoothing'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 97. 2019, pp. 1310–1320 (cited on page 50).

[CLL21] Ian Covert, Scott M. Lundberg, and Su-In Lee. 'Explaining by Removing: A Unified Framework for Model Explanation'. In: *Journal of Machine Learning Research (JMLR)* 22 (2021), 209:1–209:90 (cited on page 15).

[Cra+22] Francesco Craighero et al. 'Unity Is Strength: Improving the Detection of Adversarial Examples with Ensemble Approaches'. In: *CoRR* abs/2111.12631 (2022) (cited on page 55).

[CS95] Mark W. Craven and Jude W. Shavlik. 'Extracting Tree-Structured Representations of Trained Networks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*. 1995, pp. 24–30 (cited on page 33).

[Cro+22] Francesco Croce et al. 'Evaluating the Adversarial Robustness of Adaptive Test-time Defenses'. In: *Proc. of the International Conference on Machine Learning (ICML)*. 2022, pp. 4421–4435 (cited on page 27).

[Dan+20] Marina Danilevsky et al. 'A Survey of the State of Explainable AI for Natural Language Processing'. In: *Proc. of the Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (AACL/IJCNLP)*. 2020, pp. 447–459 (cited on pages 12, 23).

[DR20] Arun Das and Paul Rad. 'Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey'. In: *CoRR* abs/2006.11371 (2020) (cited on pages 12, 23).

[Den+09] Jia Deng et al. 'Imagenet: A large-scale hierarchical image database'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255 (cited on page 19).

[Dim+20] Botty Dimanov et al. 'You Shouldn't Trust Me: Learning Models Which Conceal Unfairness from Multiple Explanation Methods'. In: *Proc. of the Workshop on Artificial Intelligence Safety Co-Located with the AAAI Conference on Artificial Intelligence (SafeAI@AAAI)*. Vol. 2560. 2020, pp. 63–73 (cited on pages 41, 64, 88).

[DAR20] Bao Gia Doan, Ehsan Abbasnejad, and Damith C. Ranasinghe. 'Februus: Input Purification Defense against Trojan Attacks on Deep Neural Network Systems'. In: *Proc. of the Annual Computer Security Applications Conference (ACSAC)*. 2020, pp. 897–912 (cited on pages 2, 4, 13, 50, 55, 63, 78, 79, 81, 88, 98, 172).

[DGK21] Ann-Kathrin Dombrowski, Jan E Gerken, and Pan Kessel. 'Diffeomorphic Explanations with Normalizing Flows'. In: *Proc. of the ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models (INNF)*. 2021 (cited on page 11).

[Dom+19] Ann-Kathrin Dombrowski et al. 'Explanations Can Be Manipulated and Geometry Is to Blame'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2019, pp. 13567–13578 (cited on pages 26, 28, 29, 34, 35, 38, 41, 45, 48, 49, 52, 57, 65, 68, 69, 91–95, 101, 109, 113, 183).

[Dom+22] Ann-Kathrin Dombrowski et al. 'Towards Robust Explanations for Deep Neural Networks'. In: *Pattern Recognition* 121 (2022), p. 108194 (cited on pages 18, 41, 52, 53, 57, 65, 104, 113).

[DD92] Hubert L. Dreyfus and Stuart E. Dreyfus. 'What Artificial Experts Can and Cannot Do'. In: *AI Soc.* 6.1 (1992), pp. 18–26 (cited on page 2).

[DL92] Harris Drucker and Yann LeCun. 'Improving Generalization Performance Using Double Backpropagation'. In: *IEEE Transactions on Neural Networks* 3.6 (1992), pp. 991–997 (cited on page 53).

[Du+19] Mengnan Du et al. 'Learning Credible Deep Neural Networks with Rationale Regularization'. In: *Proc. of the International Conference on Data Mining (ICDM)*. 2019, pp. 150–159 (cited on page 32).

[EV17] Lilian Edwards and Michael Veale. *Slave to the Algorithm? Why a 'right to an Explanation' Is Probably Not the Remedy You Are Looking For*. Preprint. LawArXiv, 2017 (cited on page 1).

[EUD18] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 'Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning'. In: *Neural Networks* 107 (2018), pp. 3–11 (cited on page 68).

[Erh+09] D. Erhan et al. 'Visualizing Higher-Layer Features of a Deep Network'. In: 2009 (cited on page 11).

[Est+17] Andre Esteva et al. 'Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks'. In: *Nature* 542.7639 (2017), pp. 115–118 (cited on page 1).

[EU18] EU. *Ethics Guidelines for Trustworthy AI*. Tech. rep. European Commission, 2018, p. 41 (cited on pages 2, 13).

[Eve+10] Mark Everingham et al. 'The Pascal Visual Object Classes (VOC) Challenge'. In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338 (cited on pages 2, 12).

[Fan+23] Wenqi Fan et al. 'Jointly Attacking Graph Neural Network and Its Explanations'. In: *Proc. of the IEEE International Conference on Data Engineering (ICDE)*. 2023 (cited on page 41).

[FC20] Shihong Fang and Anna Choromanska. 'Backdoor Attacks on the DNN Interpretation System'. In: *Proc. of the Workshop on Dataset Curation and Security at NeurIPS* (2020), p. 15 (cited on page 88).

[FC22] Shihong Fang and Anna Choromanska. 'Backdoor Attacks on the DNN Interpretation System'. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)* (2022), pp. 561–570 (cited on pages 3, 40, 51, 64, 88, 103, 108, 109).

[FL22] Andrea Ferrario and Michele Loi. 'How Explainability Contributes to Trust in AI'. In: *Proc. of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2022, pp. 1457–1466 (cited on page 13).

[FBS20] Gil Fidel, Ron Bitton, and Asaf Shabtai. 'When Explainability Meets Adversarial Learning: Detecting Adversarial Examples Using SHAP Signatures'. In: *Proc. of the International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–8 (cited on page 55).

[FV17] Ruth Fong and Andrea Vedaldi. 'Interpretable Explanations of Black Boxes by Meaningful Perturbation'. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2017), pp. 3449–3457 (cited on pages 11, 16).

[GCZ19] Hui Gao, Yunfang Chen, and Wei Zhang. 'Detection of Trojaning Attack on Neural Networks via Cost of Sample Classification'. In: *Secur. Commun. Networks* 2019 (2019), 1953839:1–1953839:12 (cited on page 51).

[Gao+22a] Yuyang Gao et al. 'Aligning Eyes between Humans and Deep Neural Network through Interactive Attention Alignment'. In: *Proc. ACM Hum. Comput. Interact.* 6.CSCW2 (2022), pp. 1–28 (cited on page 128).

[Gao+22b] Yuyang Gao et al. 'RES: A Robust Framework for Guiding Visual Explanation'. In: *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2022, pp. 432–442 (cited on page 128).

[Gao+24] Yuyang Gao et al. 'Going beyond XAI: A Systematic Survey for Explanation-Guided Learning'. In: *ACM Computing Surveys* 56.7 (2024), 188:1–188:39 (cited on page 128).

[Gar+20] Siddhant Garg et al. 'Can Adversarial Weight Perturbations Inject Neural Backdoors'. In: *Proc. of ACM International Conference on Information and Knowledge Management (CIKM)*. 2020, pp. 2029–2032 (cited on page 33).

[GM21] Damien Garreau and Dina Mardaoui. 'What Does LIME Really See in Images?' In: *CoRR* abs/2102.06307 (2021) (cited on page 16).

[Gei+21] Jonas Geiping et al. 'Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2021 (cited on page 31).

[Gei+20] Robert Geirhos et al. 'Shortcut Learning in Deep Neural Networks.' In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673 (cited on page 2).

[Gha+21] Sahra Ghalebikesabi et al. 'On Locality of Local Explanation Models'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2021, pp. 18395–18407 (cited on page 17).

[GAZ19] Amirata Ghorbani, Abubakar Abid, and James Y. Zou. 'Interpretation of Neural Networks Is Fragile'. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*. 2019, pp. 3681–3688 (cited on pages 41, 48, 91, 109, 113, 123).

[Gho+19] Amirata Ghorbani et al. 'Towards Automatic Concept-Based Explanations'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2019, pp. 9273–9282 (cited on page 11).

[GNB23] Fosca Giannotti, Francesca Naretto, and Francesco Bodria. 'Explainable for Trustworthy AI'. In: *Human-Centered Artificial Intelligence*. Vol. 13500. 2023, pp. 175–195 (cited on page 2).

[Gil20] Jacob Gildenblat. *Exploring Explainability for Vision Transformers*. 2020 (cited on page 127).

[Gin86] Matthew L. Ginsberg. 'Counterfactuals'. In: *Artificial Intelligence* 30.1 (1986), pp. 35–79 (cited on page 11).

[Goh+21] Gabriel Goh et al. 'Multimodal Neurons in Artificial Neural Networks'. In: 6.3 (2021), e30 (cited on page 11).

[Gom+20] Oscar Gomez et al. 'ViCE: Visual Counterfactual Explanations for Machine Learning Models'. In: *Proc. of the International Conference on Intelligent User Interfaces (IUI)*. 2020, pp. 531–535 (cited on page 11).

[GSS15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 'Explaining and Harnessing Adversarial Examples'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2015 (cited on pages 26, 29, 35, 74, 91, 92, 101).

[GF17] Bryce Goodman and Seth Flaxman. 'European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"'. In: *AI Magazine* 38.3 (2017), pp. 50–57 (cited on page 1).

[Gri+20] Sorin Mihai Grigorescu et al. 'A Survey of Deep Learning Techniques for Autonomous Driving'. In: *J. Field Robotics* 37.3 (2020), pp. 362–386 (cited on page 1).

[Gro+17a] Kathrin Grosse et al. 'Adversarial Examples for Malware Detection'. In: *Proc. of the European Symposium on Research in Computer Security (ESORICS)*. Vol. 10493. 2017, pp. 62–79 (cited on pages 82, 83).

[Gro+17b] Kathrin Grosse et al. 'On the (Statistical) Detection of Adversarial Examples'. In: *CoRR* abs/1702.06280 (2017) (cited on page 55).

[GR15] Shixiang Gu and Luca Rigazio. 'Towards Deep Neural Network Architectures Robust to Adversarial Examples'. In: *CoRR* abs/1412.5068 (2015) (cited on page 55).

[Gu+19] Tianyu Gu et al. 'BadNets: Evaluating Backdooring Attacks on Deep Neural Networks'. In: *IEEE Access* 7 (2019), pp. 47230–47244 (cited on pages 2, 26, 35, 62, 64, 65, 69, 88, 90, 91, 93, 94).

[Gui24] Riccardo Guidotti. 'Counterfactual Explanations and How to Find Them: Literature Review and Benchmarking'. In: *Data Mining and Knowledge Discovery* 38.5 (2024), pp. 2770–2824 (cited on page 11).

[Gui+19] Riccardo Guidotti et al. 'A Survey Of Methods For Explaining Black Box Models'. In: *ACM Comput. Surv.* 51.5 (2019), 93:1–93:43 (cited on pages 1, 12, 15, 23, 103).

[Gul+17] Ishaan Gulrajani et al. 'Improved Training of Wasserstein GANs'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2017, pp. 5767–5777 (cited on page 81).

[Guo+18] Wenbo Guo et al. 'LEMNA: Explaining Deep Learning Based Security Applications'. In: *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. 2018, pp. 364–379 (cited on page 16).

[GKM23] Lav Kumar Gupta, Deepika Koundal, and Shweta Mongia. 'Explainable Methods for Image-Based Deep Learning: A Review'. In: *Archives of Computational Methods in Engineering* (2023) (cited on pages 12, 23).

[Hah25] David Hahnemann. 'Towards Robust Explanations with Adversarial Training'. MA thesis. Karlsruhe Institute of Technology, 2025 (cited on pages 8, 167).

[Hah+26] David Hahnemann et al. *On the Effectiveness of Explanation-Aware Adversarial Training*. Tech. rep. Karlsruher Institut für Technologie (KIT), 2026. 21 pp. DOI: 10.5445/IR/1000190493 (cited on pages 4, 6, 7, 40, 123).

[HP88] Stephen Jose Hanson and Lorien Y. Pratt. 'Comparing Biases for Minimal Network Construction with Back-Propagation'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*. 1988, pp. 177–185 (cited on page 53).

[HB22] Peter Hase and Mohit Bansal. 'When Can Models Learn From Explanations? A Formal Framework for Understanding the Roles of Explanation Data'. In: *Proc. of the Workshop on Learning with Natural Language Supervision*. 2022, pp. 29–39 (cited on page 128).

[He+16] Kaiming He et al. 'Deep Residual Learning for Image Recognition'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778 (cited on pages 68, 95, 113, 114, 183).

[HS22] Zhixun He and Mukesh Singhal. 'Defense-CycleGAN: A Defense Mechanism Against Adversarial Attacks Using CycleGAN to Reconstruct Clean Images'. In: *Proc. of the International Conference on Pattern Recognition and Machine Learning (PRML)*. 2022, pp. 173–179 (cited on page 55).

[HNW24] Achyut Hegde, Maximilian Noppel, and Christian Wressnegger. 'Model-Manipulation Attacks against Black-Box Explanations'. In: *Proc. of the Annual Computer Security Applications Conference (ACSAC)*. 2024, pp. 974–987 (cited on pages 5–7, 14, 40, 41, 61, 88, 89, 91, 109, 127).

[HNW25] Achyut Hegde, Maximilian Noppel, and Christian Wressnegger. 'Makrut Attacks Against Black-Box Explanations'. In: *Proc. of the German Conference on Artificial Intelligence (KI)*. 2025 (cited on pages 6, 7, 89).

[Hel24] Thassilo Helmoldt. 'Explanation-Aware Backdoors through Data-Poisoning'. Master's Thesis. Karlsruhe Institute of Technology, 2024 (cited on pages 7, 26, 31, 167).

[HG16] Dan Hendrycks and Kevin Gimpel. 'Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units'. In: *CoRR* abs/1606.08415 (2016) (cited on page 68).

[HJM19] Juyeon Heo, Sunghwan Joo, and Taesup Moon. 'Fooling Neural Network Interpretations via Adversarial Model Manipulation'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2019, pp. 2921–2932 (cited on pages 26, 39, 40, 51, 64, 65, 69, 87, 88, 91, 108, 109).

[Her+23] Katherine L. Hermann et al. 'On the Foundations of Shortcut Learning'. In: *CoRR* abs/2310.16228 (2023) (cited on page 2).

[Hes+20] Tom Heskes et al. 'Causal Shapley Values: Exploiting Causal Knowledge to Explain Individual Predictions of Complex Models'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2020 (cited on page 15).

[Hin+86] Geoffrey E Hinton et al. 'Learning Distributed Representations of Concepts'. In: *Proc. of the Annual Conference of the Cognitive Science Society*. Vol. 1. 1986, p. 12 (cited on page 53).

[Hol+20] Andreas Holzinger et al. 'Explainable AI Methods - A Brief Overview'. In: *Proc. of the International Workshop, beyond Explainable AI (xxAI)*. Vol. 13200. 2020, pp. 13–38 (cited on pages 12, 23).

[Hon+21] Sanghyun Hong et al. 'Qu-ANTI-Zation: Exploiting Quantization Artifacts for Achieving Adversarial Outcomes'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2021, pp. 9303–9316 (cited on page 33).

[How19] Jeremy Howard. *Imagenette*. 2019. URL: https://github.com/fastai/imagenette/ (cited on page 89).

[Hua+15] Ruitong Huang et al. 'Learning with a Strong Adversary'. In: *CoRR* abs/1511.03034 (2015) (cited on pages 106, 107).

[HAS19] Xijie Huang, Moustafa Alzantot, and Mani Srivastava. 'NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations'. In: *CoRR* abs/1911.07399 (2019) (cited on pages 2, 13, 51).

[HS04] Dominic J. D. Hughes and Vitaly Shmatikov. 'Information Hiding, Anonymity and Privacy: A Modular Approach'. In: *Journal of Computer Security (JCS)* 12.1 (2004), pp. 3–36 (cited on page 5).

[ISI17] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 'Globally and locally consistent image completion'. In: *ACM Trans. Graph.* 36.4 (2017), 107:1–107:14 (cited on page 81).

[Iva+21] Adam Ivankay et al. 'FAR: A General Framework for Attributional Robustness'. In: *Proc. of the British Machine Vision Conference (BMVC)*. 2021, p. 24 (cited on pages 41, 45, 48, 54, 58, 102, 103, 106, 111–113, 115, 123).

[Iva+22] Adam Ivankay et al. 'Fooling Explanations in Text Classifiers'. In: *Proc. of the International Conference on Learning Representations (ICLR)* (2022), p. 13 (cited on pages 28, 34, 41, 45, 48, 101, 113).

[Jag+18] Matthew Jagielski et al. 'Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2018, pp. 19–35 (cited on page 31).

[Jag+21] Matthew Jagielski et al. 'Subpopulation Data Poisoning Attacks'. In: *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. 2021 (cited on page 31).

[JMB20] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. 'Feature Relevance Quantification in Explainable AI: A Causal Problem'. In: *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 108. 2020, pp. 2907–2916 (cited on page 15).

[JLG22] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. 'BadEncoder: Backdoor Attacks to Pre-Trained Encoders in Self-Supervised Learning'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2022, pp. 2043–2059 (cited on pages 62, 65, 88, 90).

[Jia+23] Junqi Jiang et al. 'Provably Robust and Plausible Counterfactual Explanations for Neural Networks via Robust Optimisation'. In: *Proc. of the Asia Conference on Machine Learning (ACML)*. Vol. 222. 2023, pp. 582–597 (cited on page 11).

[Jia+24] Junqi Jiang et al. 'Robust Counterfactual Explanations in Machine Learning: A Survey'. In: 2024, pp. 8086–8094 (cited on page 11).

[Joo+23] Sunghwan Joo et al. 'Towards More Robust Interpretation via Local Gradient Alignment'. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*. 2023, pp. 8168–8176 (cited on pages 4, 54, 102–106, 113, 115).

[Jul+16] Angwin Julia et al. *Machine Bias*. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing. 2016 (cited on page 89).

[JO21] Hyungsik Jung and Youngrock Oh. 'LIFT-CAM: Towards Better Explanations for Class Activation Mapping'. In: *CoRR* abs/2102.05228 (2021) (cited on page 23).

[Jyo+22] Amlan Jyoti et al. 'On the Robustness of Explanations of Deep Neural Network Models: A Survey'. In: *CoRR* abs/2211.04780 (2022) (cited on pages 3, 59).

[KAS24] Md Abdul Kadir, GowthamKrishna Addluri, and Daniel Sonntag. 'Revealing Vulnerabilities of Neural Networks in Parameter Learning and Defense Against Explanation-Aware Backdoors'. In: *CoRR* arXiv:2403.16569 (2024) (cited on pages 51, 64, 88).

[Kan+20] Kentaro Kanamori et al. 'DACE: Distribution-Aware Counterfactual Explanation by Mixed-Integer Linear Optimization'. In: *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2020, pp. 2855–2862 (cited on page 11).

[Kao+22] Ching-Yu Kao et al. 'Rectifying Adversarial Inputs Using XAI Techniques'. In: *Proc. of the European Signal Processing Conference (EUSIPCO)*. 2022, pp. 573–577 (cited on page 55).

[KMM20] Jacob Kauffmann, Klaus-Robert Müller, and Grégoire Montavon. 'Towards Explaining Anomalies: A Deep Taylor Decomposition of One-Class Models'. In: *Pattern Recognition* 101 (2020), p. 107198 (cited on page 20).

[KG90] Maurice Kendall and Jean D. Gibbons. *Rank Correlation Methods*. Fifth. 1990 (cited on page 48).

[Kim+18] Been Kim et al. 'Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 80. 2018, pp. 2673–2682 (cited on page 11).

[Kim+23] Sunnie S. Y. Kim et al. '"Help Me Help the AI": Understanding How Explainability Can Support Human-AI Interaction'. In: *Proc. of the Conference on Human Factors in Computing Systems (CHI)*. 2023, pp. 1–17 (cited on page 13).

[Kin+16] Pieter-Jan Kindermans et al. 'Investigating the Influence of Noise and Distractors on the Interpretation of Neural Networks'. In: *CoRR* abs/1611.07270 (2016) (cited on page 18).

[Kin+19] Pieter-Jan Kindermans et al. 'The (Un)Reliability of Saliency Methods'. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Vol. 11700. Lecture Notes in Computer Science. 2019, pp. 267–280 (cited on pages 87, 104).

[KB15] Diederik P. Kingma and Jimmy Ba. 'Adam: A Method for Stochastic Optimization'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2015 (cited on pages 68, 69, 82).

[KMB24] Patrick Knab, Sascha Marton, and Christian Bartelt. 'Beyond Pixels: Enhancing LIME with Hierarchical Features and Segmentation Foundation Models'. In: *CoRR* (2024) (cited on page 16).

[Kna+25] Patrick Knab et al. 'Which LIME Should I Trust? Concepts, Challenges, and Solutions'. In: *Proc. of the World Conference on eXplainable Artificial Intelligence (XAI)*. 2025 (cited on page 16).

[KM08] Amy J. Ko and Brad A. Myers. 'Debugging Reinvented: Asking and Answering Why and Why Not Questions about Program Behavior'. In: *Proc. of the International Conference on Software Engineering (ICSE)*. 2008, pp. 301–310 (cited on pages 2, 12).

[Kri09] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Tech. rep. 2009 (cited on pages 69, 95, 113, 183).

[KNH] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. *CIFAR (Canadian Institute for Advanced Research)*. URL: http://www.cs.toronto.edu/~kriz/cifar.html (cited on pages 69, 95).

[KH91] Anders Krogh and John A. Hertz. 'A Simple Weight Decay Can Improve Generalization'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*. 1991, pp. 950–957 (cited on page 53).

[Kuh+19] Christiane Kuhn et al. 'On Privacy Notions in Anonymous Communication'. In: *Proc. of the Privacy Enhancing Technologies Symposium (PETS)* 2019.2 (2019), pp. 105–125 (cited on page 5).

[Kuh+21] Christiane Kuhn et al. 'Plausible Deniability for Anonymous Communication'. In: *Proc. of Workshop on Privacy in the Electronic Society (WPES)*. 2021, pp. 17–32 (cited on pages 5, 9).

[Küh+21] Niclas Kühnapfel et al. 'LaserShark: Establishing Fast, Bidirectional Communication into Air-Gapped Systems'. In: *Proc. of the Annual Computer Security Applications Conference (ACSAC)*. 2021, pp. 796–811 (cited on pages 5, 9).

[Kul+10] Todd Kulesza et al. 'Explanatory Debugging: Supporting End-User Debugging of Machine-Learned Programs'. In: *Proc. of the IEEE Symposium on Visual Languages and Human-Centric Computing (HCC)*. 2010, pp. 41–48 (cited on pages 2, 13).

[Kul+12] Todd Kulesza et al. 'Tell Me More?: The Effects of Mental Model Soundness on Personalizing an Intelligent Agent'. In: *Proc. of the Conference on Human Factors in Computing Systems (CHI)*. 2012, pp. 1–10 (cited on pages 2, 13).

[Kul+13] Todd Kulesza et al. 'Too Much, Too Little, or Just Right? Ways Explanations Impact End Users' Mental Models'. In: *Proc. of the IEEE Symposium on Visual Languages and Human Centric Computing (VLHCC)*. 2013, pp. 3–10 (cited on pages 2, 13).

[Kul+15] Todd Kulesza et al. 'Principles of Explanatory Debugging to Personalize Interactive Machine Learning'. In: *Proc. of the International Conference on Intelligent User Interfaces (IUI)*. 2015, pp. 126–137 (cited on page 2).

[KL51] S. Kullback and R. A. Leibler. 'On Information and Sufficiency'. In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86 (cited on page 111).

[KL20] Aditya Kuppa and Nhien-An Le-Khac. 'Black Box Attacks on Explainable Artificial Intelligence (XAI) Methods in Cyber Security'. In: *Proc. of the International Joint Conference on Neural Networks (IJCNN)*. 2020, pp. 1–8 (cited on pages 26, 41, 113).

[LAH22] Gabriel Laberge, Ulrich Aïvodji, and Satoshi Hara. 'Fooling SHAP with Stealthily Biased Sampling'. In: *CoRR* abs/2205.15419 (2022) (cited on page 56).

[Lai24] Fabio Lais. 'Manipulating Counterfactual Explanations Using Honeypots'. Bachelor's Thesis. Karlsruhe Institute of Technology, 2024 (cited on pages 8, 127, 167).

[LAB20] Himabindu Lakkaraju, Nino Arsov, and Osbert Bastani. 'Robust and Stable Black Box Explanations'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 119. 2020, pp. 5628–5638 (cited on pages 16, 57).

[LB20] Himabindu Lakkaraju and Osbert Bastani. '"How Do I Fool You?": Manipulating User Trust via Misleading Black Box Explanations'. In: *Proc. of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*. 2020, pp. 79–85 (cited on page 41).

[Lak+19] Himabindu Lakkaraju et al. 'Faithful and Customizable Explanations of Black Box Models'. In: *Proc. of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*. 2019, pp. 131–138 (cited on page 33).

[Lap+19] Sebastian Lapuschkin et al. 'Unmasking Clever Hans Predictors and Assessing What Machines Really Learn'. In: *Nature Communications* 10.1 (2019), p. 1096 (cited on pages 2, 11, 12, 50).

[LWL20] Thai Le, Suhang Wang, and Dongwon Lee. 'GRACE: Generating Concise and Informative Contrastive Sample to Explain Neural Network Model's Prediction'. In: *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2020, pp. 238–248 (cited on page 11).

[LAJ19] Guang-He Lee, David Alvarez-Melis, and Tommi S. Jaakkola. 'Towards Robust, Locally Linear Deep Networks'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2019 (cited on pages 45, 49).

[Lee+21] Jeong Ryong Lee et al. 'Relevance-CAM: Your Model Already Knows Where to Look'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 14944–14953 (cited on pages 1, 3, 23, 63, 68, 70, 88).

[Lee+18] Kimin Lee et al. 'A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2018 (cited on page 55).

[Lee22] Seung C. Lee. 'A Black Box Approach to Auditing Algorithms'. In: *Issues In Information Systems* (2022) (cited on page 56).

[Lei+23] Benedikt Leichtmann et al. 'Effects of Explainable Artificial Intelligence on Trust and Human Behavior in a High-Risk Decision Task'. In: *Computers in Human Behavior* 139 (2023), p. 107539 (cited on page 13).

[LW25] Francesco Leofante and Matthew Wicker. *Robust Explainable AI*. SpringerBriefs in Intelligent Systems. Springer Cham, 2025 (cited on page 128).

[LF20] Alexander Levine and Soheil Feizi. '(De)Randomized Smoothing for Certifiable Defense against Patch Attacks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2020 (cited on page 30).

[LSF19] Alexander Levine, Sahil Singla, and Soheil Feizi. 'Certifiably Robust Interpretation in Deep Learning'. In: *CoRR* abs/1905.12105 (2019) (cited on pages 39, 45, 48, 50).

[Li+22] Dingwen Li et al. 'Self-Explaining Hierarchical Model for Intraoperative Time Series'. In: *CoRR* abs/2210.04417 (2022) (cited on page 11).

[LZ25] Wanman Li and Wan Mohd Nazmee Wan Zainon. 'Dual Adversarial Attacks on Explainable Deep Learning in Medical Image Classification'. In: *Computer Methods and Programs in Biomedicine* (2025), p. 109125 (cited on pages 38, 39, 41).

[Li+21a] Yige Li et al. 'Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2021 (cited on page 51).

[Li+21b] Yuezun Li et al. 'Invisible Backdoor Attack with Sample-Specific Triggers'. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 16463–16472 (cited on pages 65, 78, 90).

[LLC21] Yi-Shan Lin, Wen-Chuan Lee, and Z. Berkay Celik. 'What Do You See?: Evaluation of Explainable Artificial Intelligence (XAI) Interpretability through Neural Backdoors'. In: *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2021, pp. 1027–1035 (cited on pages 2, 13, 78).

[LPK21] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 'Explainable AI: A Review of Machine Learning Interpretability Methods'. In: *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* 23.1 (2021), p. 18 (cited on pages 1, 15).

[Lip90] Peter Lipton. 'Contrastive Explanation'. In: *Royal Institute of Philosophy Supplement* 27 (1990), pp. 247–266 (cited on page 11).

[Lip18] Zachary C. Lipton. 'The Mythos of Model Interpretability'. In: *Commun. ACM* 16.10 (2018), pp. 36–43 (cited on pages 1, 11).

[LDG18] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 'Fine-Pruning: Defending against Backdooring Attacks on Deep Neural Networks'. In: *Proc. of the International Symposium Research in Attacks, Intrusions, and Defenses (RAID)*. Vol. 11050. 2018, pp. 273–294 (cited on pages 5, 51, 88).

[Liu+17] Shixia Liu et al. 'Towards Better Analysis of Machine Learning Models: A Visual Analytics Perspective'. In: *Vis. Informatics* 1.1 (2017), pp. 48–56 (cited on pages 2, 13).

[Liu+18] Yingqi Liu et al. 'Trojaning Attack on Neural Networks'. In: *Proc. of the Network and Distributed System Security Symposium (NDSS)*. 2018 (cited on pages 2, 35, 62, 65, 88, 90, 91).

[Liu+15] Ziwei Liu et al. 'Deep Learning Face Attributes in the Wild'. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2015 (cited on page 17).

[LL17] Scott M Lundberg and Su-In Lee. 'A Unified Approach to Interpreting Model Predictions'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)* (2017), p. 10 (cited on pages 1, 5, 11, 16, 88, 89, 127).

[LHL15] Chunchuan Lyu, Kaizhu Huang, and Hai-Ning Liang. 'A Unified Gradient Regularization Family for Adversarial Examples'. In: *Proc. of the International Conference on Data Mining (ICDM)*. 2015, pp. 301–309 (cited on pages 106, 107).

[Ma+18] Xingjun Ma et al. 'Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2018 (cited on page 55).

[Mad+18a] Aleksander Madry et al. 'Towards Deep Learning Models Resistant to Adversarial Attacks'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2018 (cited on pages 4, 54, 74, 91, 95, 101–103, 106, 107, 109, 110, 113, 115, 123, 183).

[Mad+18b] Aleksander Madry et al. 'Towards Deep Learning Models Resistant to Adversarial Attacks'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2018 (cited on pages 27, 75).

[MV15] Aravindh Mahendran and Andrea Vedaldi. 'Understanding Deep Image Representations by Inverting Them'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5188–5196 (cited on page 11).

[MT20] Erwan Le Merrer and Gilles Trédan. 'Remote Explainability Faces the Bouncer Problem'. In: *Nature Machine Intelligence* 2 (2020), pp. 529–539 (cited on page 34).

[Met+17] Jan Hendrik Metzen et al. 'On Detecting Adversarial Perturbations'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2017 (cited on page 55).

[MP25] Maraz Mia and Mir Mehedi A. Pritom. 'Explainable but Vulnerable: Adversarial Attacks on XAI Explanation in Cybersecurity Applications'. In: *CoRR* abs/2510.03623 (2025) (cited on pages 25, 59).

[Mil19] Tim Miller. 'Explanation in Artificial Intelligence: Insights from the Social Sciences'. In: *Artificial Intelligence* 267 (2019), pp. 1–38 (cited on pages 1, 13).

[Min+22] Dang Minh et al. 'Explainable Artificial Intelligence: A Comprehensive Review'. In: *Artificial Intelligence Review* 55.5 (2022), pp. 3503–3568 (cited on pages 12, 23).

[Möl+25] Jonas Möller et al. 'Adversarial Inputs for Linear Algebra Backends'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 267. 2025 (cited on page 33).

[Mol22] Christoph Molnar. 'Interpretable Machine Learning'. In: (2022), p. 312 (cited on pages 12, 23).

[Mon+17] Grégoire Montavon et al. 'Explaining Nonlinear Classification Decisions with Deep Taylor Decomposition'. In: *Pattern Recognition* 65 (2017), pp. 211–222 (cited on pages 1, 20, 58).

[Mon+19] Grégoire Montavon et al. 'Layer-Wise Relevance Propagation: An Overview'. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. 2019, pp. 193–209 (cited on page 20).

[MHW19] Jonathan Moore, Nils Hammerla, and Chris Watkins. 'Explaining Deep Learning Models with Constrained Adversarial Examples'. In: *Proc. of the Pacific Rim International Conference on Artificial Intelligence (PRICAI)*. Vol. 11670. 2019, pp. 43–56 (cited on page 127).

[MFF16] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 'DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2574–2582 (cited on page 101).

[Moo+17a] Seyed-Mohsen Moosavi-Dezfooli et al. 'Universal Adversarial Perturbations'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 86–94 (cited on pages 29, 35).

[Moo+17b] Seyed-Mohsen Moosavi-Dezfooli et al. 'Universal Adversarial Perturbations'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 86–94 (cited on page 90).

[Moo+19] Seyed-Mohsen Moosavi-Dezfooli et al. 'Robustness via Curvature Regularization, and Vice Versa'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9078–9086 (cited on page 53).

[MOT15] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. *Inceptionism: Going Deeper into Neural Networks*. 2015 (cited on page 11).

[MST20] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. 'Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations'. In: *Proc. of the ACM Conference on Fairness, Accountability, and Transparency (FAT*)*. 2020, pp. 607–617 (cited on pages 11, 127).

[Mur+19] W. James Murdoch et al. 'Definitions, Methods, and Applications in Interpretable Machine Learning'. In: *Proc. of the National Academy of Sciences* 116.44 (2019), pp. 22071–22080 (cited on pages 1, 11).

[NH10] Vinod Nair and Geoffrey E. Hinton. 'Rectified Linear Units Improve Restricted Boltzmann Machines'. In: *Proc. of the International Conference on Machine Learning (ICML)*. 2010, pp. 807–814 (cited on page 68).

[NP22] Anirban Nandi and Aditya Kumar Pal. *Interpreting Machine Learning Models: Learn Model Interpretability and Explainability Methods*. 2022 (cited on pages 12, 23).

[Nat23] National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. Tech. rep. NIST AI 100-1. National Institute of Standards and Technology (U.S.), 2023 (cited on pages 2, 13).

[NNW26] Nicolai Neuer, Maximilian Noppel, and Christian Wressnegger. *The Influence of ViT Backdoors on Post-Hoc Explanations*. Technical Report. Karlsruhe Institute of Technology (KIT), 2026 (cited on page 6).

[Ngu+17] Anh Nguyen et al. 'Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3510–3520 (cited on page 11).

[NYC16] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. 'Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned by Each Neuron in Deep Neural Networks'. In: *CoRR* abs/1602.03616 (2016) (cited on page 11).

[Ngu+16] Anh Mai Nguyen et al. 'Synthesizing the Preferred Inputs for Neurons in Neural Networks via Deep Generator Networks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*. 2016, pp. 3387–3395 (cited on page 11).

[NT20] Tuan Anh Nguyen and Anh Tran. 'Input-Aware Dynamic Backdoor Attack'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2020 (cited on page 90).

[Nie+22] Weili Nie et al. 'Diffusion Models for Adversarial Purification'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 162. 2022, pp. 16805–16827 (cited on page 55).

[NPW23] Maximilian Noppel, Lukas Peter, and Christian Wressnegger. 'Disguising Attacks with Explanation-Aware Backdoors'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2023 (cited on pages 2, 4, 6, 7, 11, 13, 26, 35, 38–40, 51, 57, 61, 88, 91, 93, 94, 107, 109).

[NRW25] Maximilian Noppel, Karl Rubel, and Christian Wressnegger. 'Exploiting Contexts of LLM-based Code-Completion'. In: *Proc. of the German Conference on Artificial Intelligence (KI)*. Vol. 15956. 2025, pp. 298–302 (cited on page 8).

[NW23a] Maximilian Noppel and Christian Wressnegger. 'Explanation-Aware Backdoors in a Nutshell.' In: *Proc. of the German Conference on Artificial Intelligence (KI)*. 2023 (cited on pages 6, 7, 61).

[NW23b] Maximilian Noppel and Christian Wressnegger. 'Poster: Fooling XAI with Explanation-Aware Backdoors'. In: *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. 2023 (cited on pages 6, 40, 61).

[NW24a] Maximilian Noppel and Christian Wressnegger. 'A Brief Systematization of Explanation-Aware Attacks'. In: *Proc. of the German Conference on Artificial Intelligence (KI)*. 2024 (cited on pages 6, 25).

[NW24b] Maximilian Noppel and Christian Wressnegger. 'SoK: Explainable Machine Learning in Adversarial Environments'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2024 (cited on pages 4, 6, 11, 25, 59, 91).

[NW25] Maximilian Noppel and Christian Wressnegger. 'Composite Explanation-Aware Attacks'. In: *Proc. of the IEEE Symposium on Security and Privacy Workshops*. 2025, pp. 167–176 (cited on pages 4, 6, 7, 26, 38, 40, 91, 98, 101, 109).

[OMS17] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. 'Feature Visualization'. In: 2.11 (2017), e7 (cited on page 11).

[Ola+18] Chris Olah et al. 'The Building Blocks of Interpretability'. In: *Distill* 3.3 (2018), e10 (cited on page 11).

[Ome+19] Daniel Omeiza et al. 'Smooth Grad-CAM++: An Enhanced Inference Level Visualization Technique for Deep Convolutional Neural Network Models'. In: *CoRR* abs/1908.01224 (2019) (cited on pages 19, 23).

[OC23] Ahmet Haydar Ornek and Murat Ceylan. 'CodCAM: A New Ensemble Visual Explanation for Classification of Medical Thermal Images'. In: *Quantitative InfraRed Thermography Journal* (2023), pp. 1–25 (cited on page 23).

[Øyg15] Audun M Øygard. *Visualizing GoogLeNet Classes*. http://auduno.github.io/2015/07/29/visu googlenet-classes/index.html. 2015 (cited on page 11).

[Pac+24] Elisabeth Pachl et al. 'A View on Vulnerabilites: The Security Challenges of XAI (Academic Track)'. In: *Proc. of the Symposium on Scaling AI Assessments (SAIA)*. Vol. 126. 2024, 12:1–12:23 (cited on pages 25, 59, 91).

[Pap+16] Nicolas Papernot et al. 'The Limitations of Deep Learning in Adversarial Settings'. In: *Proc. of the IEEE European Symposium on Security and Privacy (EuroS&P)*. 2016, pp. 372–387 (cited on pages 26, 74).

[Pap+18] Nicolas Papernot et al. 'SoK: Security and Privacy in Machine Learning'. In: *Proc. of the IEEE European Symposium on Security and Privacy (EuroS&P)*. 2018, pp. 399–414 (cited on pages 34, 38).

[Pen+19] Feargus Pendlebury et al. 'TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time'. In: *Proc. of the USENIX Security Symposium*. 2019, pp. 729–746 (cited on pages 1, 82).

[Pet23] Lukas Peter. 'Explanation-Aware Backdoors for Transformers'. MA thesis. Karlsruhe Institute of Technology, 2023 (cited on pages 7, 127, 167).

[Pet22] Dragutin Petkovic. 'It is Not "Accuracy vs. Explainability"—We Need Both for Trustworthy AI Systems'. In: *IEEE Transactions on Technology and Society* 4 (2022), pp. 46–53 (cited on page 13).

[PDS18] Vitali Petsiuk, Abir Das, and Kate Saenko. 'RISE: Randomized Input Sampling for Explanation of Black-Box Models'. In: *CoRR* abs/1806.07421 (2018) (cited on pages 5, 16, 89).

[Pie+20] Fabio Pierazzi et al. 'Intriguing Properties of Adversarial ML Attacks in the Problem Space'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2020, pp. 1332–1349 (cited on page 83).

[PMT18] Gregory Plumb, Denali Molitor, and Ameet S. Talwalkar. 'Model Agnostic Supervised Local Explanations'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2018, pp. 2520–2529 (cited on page 104).

[Plu+20] Gregory Plumb et al. 'Regularizing Black-Box Models for Improved Interpretability'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2020 (cited on page 54).

[Poy+20] Rafael Poyiadzi et al. 'FACE: Feasible and Actionable Counterfactual Explanations'. In: *Proc. of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)* (2020), pp. 344–350 (cited on page 11).

[Pun+25] Samuele Punzo et al. 'A Machine Learning Pipeline for Multiple Sclerosis Biomarker Discovery: Comparing Explainable AI and Traditional Statistical Approaches'. In: *CoRR* abs/2509.22484 (2025) (cited on page 13).

[Qi+22] Xiangyu Qi et al. 'Towards Practical Deployment-Stage Backdoor Attack on Deep Neural Networks'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 13337–13347 (cited on page 56).

[Qin+22] Yunpeng Qing et al. 'A Survey on Explainable Reinforcement Learning: Concepts, Algorithms, Challenges'. In: *CoRR* arXiv:2211.06665 (2022) (cited on pages 12, 23).

[Qiu+22] Luyu Qiu et al. 'Generating Perturbation-Based Explanations with Robustness to out-of-Distribution Data'. In: *Proc. of the International World Wide Web Conference (WWW)*. 2022, pp. 3594–3605 (cited on page 16).

[QR20] Erwin Quiring and Konrad Rieck. 'Backdooring and Poisoning Neural Networks with Image-Scaling Attacks'. In: *Proc. of the IEEE Symposium on Security and Privacy Workshops*. 2020, pp. 41–47 (cited on page 90).

[Qui+20] Erwin Quiring et al. 'Adversarial Preprocessing: Understanding and Preventing Image-Scaling Attacks in Machine Learning'. In: *Proc. of the USENIX Security Symposium*. 2020, pp. 1363–1380 (cited on page 64).

[Rag+17] Maithra Raghu et al. 'On the Expressive Power of Deep Neural Networks'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 70. 2017, pp. 2847–2854 (cited on page 49).

[RSL18] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. 'Certified Defenses against Adversarial Examples'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2018 (cited on page 27).

[RAM22] Arief Nur Rahman, Dian Andriana, and Carmadi Machbub. 'Comparison between Grad-CAM and EigenCAM on YOLOv5 Detection Model'. In: *Proc. of the International Symposium on Electronics and Smart Devices (ISESD)*. 2022, pp. 1–5 (cited on page 13).

[RHF20] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. 'TBT: Targeted Neural Network Attack with Bit Trojan'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 13195–13204 (cited on pages 33, 56).

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. '"Why Should I Trust You?": Explaining the Predictions of Any Classifier'. In: *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2016 (cited on pages 1, 5, 11, 15, 16, 22, 88, 89, 127).

[RH20] Laura Rieger and Lars Kai Hansen. 'A Simple Defense against Adversarial Attacks on Heatmap Explanations'. In: *Proc. of the ICML Workshop on Human Interpretability in Machine Learning (WHI)*. 2020 (cited on pages 38, 57, 85).

[Rie+20] Laura Rieger et al. 'Interpretations Are Useful: Penalizing Explanations to Align Neural Networks with Prior Knowledge'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 119. 2020, pp. 8116–8126 (cited on pages 12, 32, 128).

[RD18] Andrew Slavin Ross and Finale Doshi-Velez. 'Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients'. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*. 2018, pp. 1660–1669 (cited on pages 12, 104).

[RHD17] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 'Right for the Right Reasons: Training Differentiable Models by Constraining Their Explanations'. In: *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2017, pp. 2662–2670 (cited on pages 12, 32, 128).

[RKH19] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. 'The Odds Are Odd: A Statistical Test for Detecting Adversarial Examples'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 97. 2019, pp. 5498–5507 (cited on page 55).

[RV18] Grant M. Rotskoff and Eric Vanden-Eijnden. 'Neural Networks as Interacting Particle Systems: Asymptotic Convexity of the Loss Landscape and Universal Scaling of the Approximation Error'. In: *CoRR* abs/1805.00915 (2018) (cited on page 52).

[RNW25] Karl Rubel, Maximilian Noppel, and Christian Wressnegger. 'Generalized Adversarial Code-Suggestions: Exploiting Contexts of LLM-based Code-Completion'. In: *Proc. of the ACM Asia Conference on Computer and Communications Security (ASIA CCS)*. 2025, pp. 358–374 (cited on pages 5, 8).

[Rud19] Cynthia Rudin. 'Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead'. In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215 (cited on pages 1, 11).

[Rus+15] Olga Russakovsky et al. 'ImageNet Large Scale Visual Recognition Challenge'. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252 (cited on page 19).

[SC23] Bukhoree Sahoh and Anant Choksuriwong. 'The Role of Explainable Artificial Intelligence in High-Stakes Decision-Making Systems: A Systematic Review'. In: *Journal of Ambient Intelligence and Humanized Computing* (2023) (cited on page 1).

[Sal+19] Hadi Salman et al. 'Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2019, pp. 11289–11300 (cited on page 27).

[Sam+21] Wojciech Samek et al. 'Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications'. In: *Proc. of the IEEE* 109.3 (2021), pp. 247–278 (cited on page 11).

[SSB21] Anindya Sarkar, Anirban Sarkar, and Vineeth N. Balasubramanian. 'Enhanced Regularizers for Attributional Robustness'. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*. 2021, pp. 2532–2540 (cited on pages 41, 45, 48, 102, 106, 112, 113, 115, 123).

[Sch+22] Thomas Schnake et al. 'Higher-Order Explanations of Graph Neural Networks via Relevant Walks'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.11 (2022), pp. 7581–7596 (cited on page 1).

[SMV22] Johannes Schneider, Christian Meske, and Michalis Vlachos. 'Deceptive AI Explanations: Creation and Detection'. In: *Proc. of the International Conference on Agents and Artificial Intelligence (ICAART)*. Vol. 2. 2022, pp. 44–55 (cited on page 36).

[SF24] Gesina Schwalbe and Bettina Finzel. 'A Comprehensive Taxonomy for Explainable Artificial Intelligence: A Systematic Survey of Surveys on Methods and Concepts'. In: *Data Mining and Knowledge Discovery* 38.5 (2024), pp. 3043–3101 (cited on pages 12, 23).

[Sch+21] Avi Schwarzschild et al. 'Just How Toxic is Data Poisoning? A Unified Benchmark for Backdoor and Data Poisoning Attacks'. In: *Proc. of the International Conference on Machine Learning (ICML)*. 2021, pp. 9389–9398 (cited on page 90).

[Sel+20] Ramprasaath R. Selvaraju et al. 'Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization'. In: *International Journal of Computer Vision* 128.2 (2020), pp. 336–359 (cited on pages 1, 3, 11, 13, 15, 22, 58, 68, 70, 88, 93, 101, 103).

[Ser+24] Javier Del Ser et al. 'On Generating Trustworthy Counterfactual Explanations'. In: *Information Sciences* 655 (2024), p. 119898 (cited on page 11).

[Sev+21] Giorgio Severi et al. 'Explanation-Guided Backdoor Poisoning Attacks against Malware Classifiers'. In: *Proc. of the USENIX Security Symposium*. 2021, pp. 1487–1504 (cited on page 90).

[Sha+18] Ali Shafahi et al. 'Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2018, pp. 6106–6116 (cited on pages 31, 90).

[SYN18] Uri Shaham, Yutaro Yamada, and Sahand Negahban. 'Understanding Adversarial Training: Increasing Local Stability of Supervised Models through Robust Optimization'. In: *Neurocomputing* 307 (2018), pp. 195–204 (cited on pages 106, 107).

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 'Learning Important Features through Propagating Activation Differences'. In: *Proc. of the International Conference on Machine Learning (ICML)*. 2017, pp. 3145–3153 (cited on pages 11, 18, 105).

[Shr+16] Avanti Shrikumar et al. 'Not Just a Black Box: Learning Important Features through Propagating Activation Differences'. In: *CoRR* abs/1605.01713 (2016) (cited on page 18).

[Sim54] Herbert A. Simon. 'Spurious Correlation: A Causal Interpretation'. In: *Journal of the American Statistical Association* 49.267 (1954) (cited on page 2).

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 'Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps'. In: *Proc. of the International Conference on Learning Representations (ICLR) Workshop Track Proceedings*. 2014 (cited on pages 1, 3, 11, 15, 18, 22, 53, 58, 62, 68, 70, 88, 93, 104, 105).

[SZ15] Karen Simonyan and Andrew Zisserman. 'Very Deep Convolutional Networks for Large-Scale Image Recognition'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2015 (cited on page 68).

[Sin+20] Mayank Singh et al. 'Attributional Robustness Training Using Input-Gradient Spatial Alignment'. In: vol. 12372. 2020, pp. 515–533 (cited on pages 41, 58, 102, 106, 112, 115, 123).

[Sin+19] Sahil Singla et al. 'Understanding Impacts of High-Order Loss Approximations and Features in Deep Learning Interpretation'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 97. 2019, pp. 5848–5856 (cited on page 53).

[Sin+21] Sanchit Sinha et al. 'Perturbing Inputs for Fragile Interpretations in Deep Natural Language Processing'. In: *CoRR* abs/2108.04990 (2021) (cited on pages 28, 34, 41, 113).

[SGL20] Leon Sixt, Maximilian Granz, and Tim Landgraf. 'When Explanations Lie: Why Many Modified BP Attributions Fail'. In: *CoRR* abs/1912.09818 (2020) (cited on page 65).

[Sla+20] Dylan Slack et al. 'Fooling LIME and SHAP: Adversarial Attacks on Post Hoc Explanation Methods'. In: *Proc. of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*. 2020, pp. 180–186 (cited on pages 26, 34, 41, 87, 88).

[Sla+21a] Dylan Slack et al. 'Counterfactual Explanations Can Be Manipulated'. In: *CoRR* abs/2106.02666 (2021) (cited on pages 26, 40, 51).

[Sla+21b] Dylan Slack et al. 'Feature Attributions and Counterfactual Explanations Can Be Manipulated'. In: *CoRR* abs/2106.12563 (2021) (cited on pages 37, 40, 51).

[Smi+17] Daniel Smilkov et al. 'SmoothGrad: Removing Noise by Adding Noise'. In: *CoRR* abs/1706.03825 (2017) (cited on pages 1, 3, 11, 15, 19, 57, 87, 88).

[Søg22] Anders Søgaard. 'Shortcomings of Interpretability Taxonomiesfor Deep Neural Networks'. In: *Advances in Interpretable Machine Learning and Artificial Intelligence (AIMLAI)* (2022) (cited on pages 12, 23).

[SF20] Kacper Sokol and Peter Flach. 'One Explanation Does Not Fit All: The Promise of Interactive Explanations for Machine Learning Transparency'. In: *KI - Künstliche Intelligenz* 34.2 (2020), pp. 235–250 (cited on pages 2, 13).

[SF18] Kacper Sokol and Peter A. Flach. 'Glass-Box: Explaining AI Decisions with Counterfactual Statements through Conversation with a Voice-Enabled Virtual Assistant'. In: *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2018, pp. 5868–5870 (cited on pages 2, 13).

[Spr+15] Jost Tobias Springenberg et al. 'Striving for Simplicity: The All Convolutional Net'. In: *Proc. of the International Conference on Learning Representations (ICLR) Workshop Track Proceedings*. 2015 (cited on pages 11, 105).

[Sta+12] Johannes Stallkamp et al. 'Man vs. Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition'. In: *Neural Networks* 32 (2012), pp. 323–332 (cited on pages 69, 169).

[SKL17] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 'Certified Defenses for Data Poisoning Attacks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*. 2017, pp. 3517–3529 (cited on page 50).

[SC18] Pierre Stock and Moustapha Cissé. 'ConvNets and ImageNet beyond Accuracy: Understanding Mistakes and Uncovering Biases'. In: *Proc. of the European Conference on Computer Vision (ECCV)*. Vol. 11210. 2018, pp. 504–519 (cited on page 50).

[SPP19] Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. 'Fooling Network Interpretation in Image Classification'. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 2020–2029 (cited on pages 29, 39, 41).

[Sub+22] Akshayvarun Subramanya et al. 'Backdoor Attacks on Vision Transformers'. In: *CoRR* abs/2206.08477 (2022) (cited on pages 2, 78).

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 'Axiomatic Attribution for Deep Networks'. In: *CoRR* abs/1703.01365 (2017) (cited on pages 11, 19, 53, 58, 112).

[Sze+14] Christian Szegedy et al. 'Intriguing Properties of Neural Networks'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2014 (cited on pages 26, 29, 35, 91, 92, 101).

[Sze+16] Christian Szegedy et al. 'Rethinking the Inception Architecture for Computer Vision'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826 (cited on page 95).

[Tak+25] Hirotaka Takita et al. 'A Systematic Review and Meta-Analysis of Diagnostic Performance Comparison between Generative AI and Physicians'. In: *npj Digit. Medicine* 8.1 (2025) (cited on page 1).

[TLS22] Snir Vitrack Tamam, Raz Lapid, and Moshe Sipper. 'Foiling Explanations in Deep Neural Networks'. In: *CoRR* abs/2211.14860 (2022) (cited on pages 41, 101, 113).

[Tan+20] Ruixiang Tang et al. 'An Embarrassingly Simple Approach for Trojan Attack in Deep Neural Networks'. In: *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2020, pp. 218–228 (cited on pages 65, 90).

[Tan+22] Ruixiang Tang et al. 'Defense against Explanation Manipulation'. In: *Frontiers Big Data* 5 (2022), p. 704203 (cited on pages 4, 38, 41, 43, 48, 54, 57, 102–104, 106, 113).

[Tao+18] Guanhong Tao et al. 'Attacks Meet Interpretability: Attribute-Steered Detection of Adversarial Samples'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2018, pp. 7728–7739 (cited on page 55).

[Tia+22] Yulong Tian et al. 'Stealthy Backdoors as Compression Artifacts'. In: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 1372–1387 (cited on page 33).

[Tra22] Florian Tramèr. 'Detecting Adversarial Examples Is (Nearly) as Hard as Classifying Them'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 162. 2022, pp. 21692–21702 (cited on page 55).

[TLM18] Brandon Tran, Jerry Li, and Aleksander Madry. 'Spectral Signatures in Backdoor Attacks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2018, pp. 8011–8021 (cited on pages 50, 88).

[Tru+20] Loc Truong et al. 'Systematic Evaluation of Backdoor Data Poisoning Attacks on Image Classifiers'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 3422–3431 (cited on page 90).

[Tsi+19] Dimitris Tsipras et al. 'Robustness May Be at Odds with Accuracy'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2019 (cited on page 111).

[VSL24] Jon Vadillo, Roberto Santana, and Jose A. Lozano. 'Adversarial Attacks in Explainable Machine Learning: A Survey of Threats Against Models and Humans'. In: *WIREs Data Mining and Knowledge Discovery* (2024), e1567 (cited on pages 25, 59, 91).

[VSL21] Jon Vadillo, Roberto Santana, and José Antonio Lozano. 'When and How to Fool Explainable Models (and Humans) with Adversarial Examples'. In: *CoRR* abs/2107.01943 (2021) (cited on pages 3, 58, 59).

[VS08] Andrea Vedaldi and Stefano Soatto. 'Quick Shift and Kernel Methods for Mode Seeking'. In: *Proc. of the European Conference on Computer Vision (ECCV)*. Vol. 5305. 2008, pp. 705–718 (cited on page 15).

[Vel+21] Akshaj Kumar Veldanda et al. 'NNoculation: Catching BadNets in the Wild'. In: *Proc. of the ACM Workshop on Artificial Intelligence and Security (AISEC)*. 2021, pp. 49–60 (cited on pages 51, 56).

[VDH21] Sahil Verma, John Dickerson, and Keegan Hines. 'Counterfactual Explanations for Machine Learning: Challenges Revisited'. In: *CoRR* abs/2106.07756 (2021) (cited on page 11).

[Vie+19] Tom J. Viering et al. 'How to Manipulate CNNs to Make Them Lie: The GradCAM Case'. In: 2019 (cited on pages 3, 30, 40, 64, 88, 103).

[VL21] Giulia Vilone and Luca Longo. 'Classification of Explainable Artificial Intelligence Methods through Their Output Formats'. In: *Machine Learning and Knowledge Extraction* 3.3 (2021), pp. 615–661 (cited on pages 1, 12, 15, 23).

[Vis+25] Roel Visser et al. 'Trust, Distrust, and Appropriate Reliance in (X)AI: A Conceptual Clarification of User Trust and Survey of Its Empirical Evaluation'. In: *Cognitive Systems Research* 91 (2025), p. 101357 (cited on page 13).

[WMF17] Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. 'Why a Right to Explanation of Automated Decision-Making Does Not Exist in the General Data Protection Regulation'. In: *International Data Privacy Law* 7.2 (2017), pp. 76–99 (cited on page 1).

[WMR17] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 'Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR'. In: *SSRN Electronic Journal* (2017) (cited on pages 11, 127).

[Wan+19a] Bolun Wang et al. 'Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2019, pp. 707–723 (cited on pages 26, 27, 35, 51, 56, 62, 70, 88, 90).

[Wan+19b] Danding Wang et al. 'Designing Theory-Driven User-Centric Explainable AI'. In: *Proc. of the Conference on Human Factors in Computing Systems (CHI)*. 2019, pp. 1–15 (cited on pages 2, 13).

[WK22] Fan Wang and Adams Wai-Kin Kong. 'Exploiting the Relationship Between Kendall's Rank Correlation and Cosine Similarity for Attribution Protection'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2022 (cited on page 48).

[WK23] Fan Wang and Adams Wai-Kin Kong. 'A Practical Upper Bound for the Worst-Case Attribution Deviations'. In: *CoRR* abs/2303.00340 (2023) (cited on pages 48, 101).

[Wan+20a] Haofan Wang et al. 'Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 111–119 (cited on page 23).

[Wan+21] Jiakai Wang et al. 'Dual Attention Suppression Attack: Generate Adversarial Camouflage in Physical World'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 8565–8574 (cited on page 41).

[Wan+20b] Junlin Wang et al. 'Gradient-Based Analysis of NLP Models Is Manipulable'. In: *Emnlp*. Vol. EMNLP 2020. 2020, pp. 247–258 (cited on pages 34, 40).

[WG22] Shen Wang and Yuxin Gong. 'Adversarial Example Detection Based on Saliency Map Features'. In: *Applied Intelligence* 52.6 (2022), pp. 6262–6275 (cited on page 55).

[Wan+22] Xutong Wang et al. 'Make Data Reliable: An Explanation-Powered Cleaning on Malware Dataset Against Backdoor Poisoning Attacks'. In: *Proc. of the Annual Computer Security Applications Conference (ACSAC)*. 2022, pp. 267–278 (cited on page 50).

[Wan+20c] Yisen Wang et al. 'Improving Adversarial Robustness Requires Revisiting Misclassified Examples'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2020 (cited on pages 4, 54, 101–103, 106, 107, 110, 111, 113).

[Wan+04] Zhou Wang et al. 'Image Quality Assessment: From Error Visibility to Structural Similarity'. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612 (cited on pages 29, 66, 69, 107, 173, 181).

[Wan+20d] Zifan Wang et al. 'Smoothed Geometry for Robust Attribution'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2020 (cited on pages 19, 41, 45, 48, 53, 57).

[War+20] Alexander Warnecke et al. 'Evaluating Explanation Methods for Deep Learning in Security'. In: *Proc. of the IEEE European Symposium on Security and Privacy (EuroS&P)*. 2020, pp. 158–174 (cited on pages 23, 82, 128).

[Wei+18] Yi Wei et al. 'Explain Black-Box Image Classifications Using Superpixel-Based Interpretation'. In: *Proc. of the International Conference on Pattern Recognition (ICPR)*. 2018, pp. 1640–1645 (cited on page 15).

[WRH90] Andreas S. Weigend, David E. Rumelhart, and Bernardo A. Huberman. 'Generalization by Weight-Elimination with Application to Forecasting'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*. 1990, pp. 875–882 (cited on page 53).

[WJL20] Ethan Weinberger, Joseph D. Janizek, and Su-In Lee. 'Learning Deep Attribution Priors Based on Prior Knowledge'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2020 (cited on page 32).

[Wei+24] Patrick Weinberger et al. 'Applying Layer-Wise Relevance Propagation on U-Net Architectures'. In: *Proc. of the International Conference on Pattern Recognition (ICPR)*. Vol. 15312. 2024, pp. 106–121 (cited on page 20).

[Won+24] Felix Wong et al. 'Discovery of a Structural Class of Antibiotics with Explainable Deep Learning'. In: *Nat.* 626.7997 (2024), pp. 177–185 (cited on page 13).

[WW21] Dongxian Wu and Yisen Wang. 'Adversarial Neuron Pruning Purifies Backdoored Deep Models'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*. 2021, pp. 16913–16925 (cited on page 51).

[Wu+20] Mike Wu et al. 'Regional Tree Regularization for Interpretability in Deep Neural Networks'. In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*. 2020, pp. 6413–6421 (cited on page 104).

[Wu+23] Shutong Wu et al. 'Defending against Adversarial Audio via Diffusion Model'. In: *CoRR* abs/2303.01507 (2023) (cited on page 55).

[Xia+23] Geming Xia et al. 'Poisoning Attacks in Federated Learning: A Survey'. In: *IEEE Access* 11 (2023), pp. 10708–10722 (cited on page 33).

[Xia+21] Chong Xiang et al. 'PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking'. In: *Proc. of the USENIX Security Symposium*. 2021, pp. 2237–2254 (cited on page 30).

[Xie+20] Chulin Xie et al. 'DBA: Distributed Backdoor Attacks against Federated Learning'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2020 (cited on page 90).

[Xu+21] Xiaojun Xu et al. 'Detecting AI Trojans Using Meta Neural Analysis'. In: *Proc. of the IEEE Symposium on Security and Privacy (S&P)*. 2021, pp. 103–120 (cited on page 51).

[Xue+21] Mingfu Xue et al. 'Detecting Backdoor in Deep Neural Networks via Intentional Adversarial Perturbations'. In: *CoRR* abs/2105.14259 (2021) (cited on page 51).

[Yam+13] Fabian Yamaguchi et al. 'Chucky: Exposing Missing Checks in Source Code for Vulnerability Discovery'. In: *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. 2013, pp. 499–510 (cited on page 1).

[Yan+24] Peiyu Yang et al. 'Backdoor-Based Explainable AI Benchmark for High Fidelity Evaluation of Attribution Methods'. In: *CoRR* abs/2405.02344 (2024) (cited on pages 2, 13, 78).

[Yao+19] Yuanshun Yao et al. 'Latent Backdoor Attacks on Deep Neural Networks'. In: *Proc. of the ACM Conference on Computer and Communications Security (CCS)*. 2019, pp. 2041–2055 (cited on page 90).

[Yin+19] Rex Ying et al. 'GNNExplainer: Generating Explanations for Graph Neural Networks'. In: *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)* (2019), pp. 9240–9251 (cited on page 1).

[Zen+21] Yi Zeng et al. 'Rethinking the Backdoor Attacks' Triggers: A Frequency Perspective'. In: *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021 (cited on pages 65, 90).

[Zen+22] Yi Zeng et al. 'Adversarial Unlearning of Backdoors via Implicit Hypergradient'. In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2022 (cited on page 51).

[Zha+18] Guodong Zhang et al. 'Three Mechanisms of Weight Decay Regularization'. In: *CoRR* abs/1810.12281 (2018) (cited on page 53).

[Zha+24a] Hanwei Zhang et al. 'Opti-CAM: Optimizing Saliency Maps for Interpretability'. In: *Comput. Vis. Image Underst.* 248 (2024), p. 104101 (cited on page 23).

[ZGS21] Hengtong Zhang, Jing Gao, and Lu Su. 'Data Poisoning Attacks against Outcome Interpretations of Predictive Models'. In: *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2021, pp. 2165–2173 (cited on pages 26, 31, 40, 87, 88).

[Zha+19] Hongyang Zhang et al. 'Theoretically Principled Trade-off between Robustness and Accuracy'. In: *Proc. of the International Conference on Machine Learning (ICML)*. Vol. 97. 2019, pp. 7472–7482 (cited on pages 4, 54, 101–103, 106, 107, 110, 111, 113).

[Zha+20] Xinyang Zhang et al. 'Interpretable Deep Learning under Fire'. In: *Proc. of the USENIX Security Symposium*. 2020, pp. 1659–1676 (cited on pages 26, 28, 35, 41, 74, 91–95, 99–101, 113).

[Zha+21] Yu Zhang et al. 'A Survey on Neural Network Interpretability'. In: *IEEE Trans. Emerg. Top. Comput. Intell.* 5.5 (2021), pp. 726–742 (cited on pages 1, 12, 23).

[ZAD22] Yuhao Zhang, Aws Albarghouthi, and Loris D'Antoni. 'BagFlip: A Certified Defense against Data Poisoning'. In: *CoRR* abs/2205.13634 (2022) (cited on page 50).

[Zha+25] Jingyuan Zhao et al. 'A Survey of Autonomous Driving from a Deep Learning Perspective'. In: *ACM Computing Surveys* 57.10 (2025), 263:1–263:60 (cited on page 1).

[Zha+24b] Lili Zhao et al. 'COMI: COrrect and MItigate Shortcut Learning Behavior in Deep Neural Networks'. In: *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2024, pp. 218–228 (cited on page 128).

[ZFP19] Haizhong Zheng, Earlence Fernandes, and Atul Prakash. 'Analyzing the Interpretability Robustness of Self-Explaining Models'. In: *CoRR* abs/1905.12429 (2019) (cited on page 11).

[Zho+16] Bolei Zhou et al. 'Learning Deep Features for Discriminative Localization'. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2921–2929 (cited on pages 1, 21, 62).

[Zho+22] Yilun Zhou et al. 'Do Feature Attribution Methods Correctly Attribute Features?' In: *Proc. of the National Conference on Artificial Intelligence (AAAI)*. 2022, pp. 9623–9633 (cited on page 32).

Appendix

Appendix A
Explanation-Aware Attacks

Contents

Chapter Preface.
This appendix contains additional material for our systematization in Chapter 3. The text in Appendix A.2 is similar to the original publication.

A.1 Details on the Hierarchy

In this section, we present the proof sketches for our hierarchy. We structure the proofs in two parts: first, we establish the implications in Appendix A.1.1, and then we demonstrate that the other implications do not hold in Appendix A.1.2. The remaining implications are determined via transitivity. In Figure 3.6, we have not visualized these implications to keep the figure clean.

The provided relations hold under the assumption that the notions' parameters ($d_{\mathcal{X}}$, $d_{\mathcal{E}}$, K, ε, and ϵ) are equivalent across all notions and are not necessarily in a specific relation to each other (cf. Lemmas 3.4.2 and A.1.4). Furthermore, we only address general robustness here. While most results transfer trivially to robustness around $\mathbf{x}$, others do not.

Together, the provided proofs and relations define our hierarchy completely, which we also have verified programmatically.

A.1.1 Implications

Lemma A.1.1 *Given a set of restrictions R and two sets of constraints $\mathsf{C}_1, \mathsf{C}_2$, such that the conjunction of constraints in C_1 implies the conjunction of constraints in C_2 for all tuples*

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X} \,.\, \Big(\bigwedge_{c \in \mathsf{C}_1} c(\mathbf{x}_1, \mathbf{x}_2) \Big) \rightarrow \Big(\bigwedge_{c \in \mathsf{C}_2} c(\mathbf{x}_1, \mathbf{x}_2) \Big) \,,$$

it holds that

$$R|\mathsf{C}_1 \Rightarrow R|\mathsf{C}_2 \,.$$

Proof. We derive this directly from the hypothetical syllogism, or equivalently, from the transitivity of implications: if $\mathsf{R} \rightarrow \mathsf{C}_1$ and $\mathsf{C}_1 \rightarrow \mathsf{C}_2$, then $\mathsf{R} \rightarrow \mathsf{C}_2$. □

From Lemma A.1.1, we infer the following relations:

$$\begin{aligned}
\textsc{ExplEq} &\Rightarrow \textsc{ExplSim}^{d_\mathcal{E},\varepsilon} \\
\textsc{ExplEq} &\Rightarrow \textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} \\
\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplEq} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} \\
\textsc{ClsEq}|\textsc{ExplEq} &\Rightarrow \textsc{ClsEq}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} \\
\textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{ExplEq} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} \\
\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplEq} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} \\
\textsc{ClsEq}|\textsc{ExplEq} &\Rightarrow \textsc{ClsEq}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} \\
\textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{ExplEq} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}
\end{aligned}$$

Lemma A.1.2 *Given two sets of restrictions R_1, R_2 and one set of constraints C, such that every tuple of samples satisfying the conjunction of R_1 also satisfies the conjunction of R_2*

$$\forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X} \,.\, \Big(\bigwedge_{r \in R_1} r(\mathbf{x}_1, \mathbf{x}_2) \Big) \rightarrow \Big(\bigwedge_{r \in R_2} r(\mathbf{x}_1, \mathbf{x}_2) \Big) \,,$$

it holds that

$$R_2|C \Rightarrow R_1|C \,.$$

Proof. We establish this lemma directly from the transitivity of implications: if $\mathsf{R}_2 \rightarrow \mathsf{C}$ and $\mathsf{R}_1 \rightarrow \mathsf{R}_2$, then $\mathsf{R}_1 \rightarrow \mathsf{C}$. □

On the basis of Lemma A.1.2, we identify the following implications:

$$\begin{aligned}
\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} \\
\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} &\Rightarrow \textsc{ClsEq}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} \\
\textsc{ClsEq}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} \\
\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} \\
\textsc{ExplEq} &\Rightarrow \textsc{ClsEq}|\textsc{ExplEq} \\
\textsc{ExplEq} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplEq} \\
\textsc{ClsEq}|\textsc{ExplEq} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{ExplEq} \\
\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplEq} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{ExplEq} \\
\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} \\
\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\Rightarrow \textsc{ClsEq}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} \\
\textsc{ClsEq}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} \\
\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}+\textsc{ClsEq}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}
\end{aligned}$$

Lemma A.1.3 *Every $\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplEq}$-robust system (F_θ, h_θ) is also $\textsc{ExplEq}$-robust:*

$$\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplEq} \Rightarrow \textsc{ExplEq}$$

under a path-connected input space $\mathcal{X}$ (indicated with a dashed blue arrow in Figure 3.6).

Proof. We observe that the ExplEq constraint is transitive:

$$\forall \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathcal{X} \,.\, \textsc{ExplEq}(\mathbf{x}_1, \mathbf{x}_2) \wedge \textsc{ExplEq}(\mathbf{x}_2, \mathbf{x}_3) \rightarrow \textsc{ExplEq}(\mathbf{x}_1, \mathbf{x}_3) \,.$$

Since the input space is path-connected, we can always find a chain of inputs $\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$ connecting any two points $\mathbf{x}_0, \mathbf{x}_n \in \mathcal{X}$ such that $d_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_{i+1}) \leq \epsilon$ for all $0 \leq i < n$. By $\textsc{Loc}^{d_{\mathcal{X}},\epsilon}|\textsc{ExplEq}$-robustness, all inputs $\mathbf{x}_0, \ldots, \mathbf{x}_n$ must have equivalent explanations. Since this holds for all $\mathbf{x}_0, \mathbf{x}_n \in \mathcal{X}$, we conclude that the system is ExplEq-robust. □

Note that, in contrast to most other relations discussed, the above relation holds only for general robustness.

Lemma A.1.4 *Every* $\textsc{Loc}^{d_{\mathcal{X}},\epsilon}$+ClsEq$|\textsc{Lip}^{d_{\mathcal{E}},d_{\mathcal{X}},K}$*-robust system* (F_θ, h_θ) *is also* $\textsc{Loc}^{d_{\mathcal{X}},\epsilon}$+ClsEq$|\textsc{ExplSim}^{d_{\mathcal{E}},\varepsilon}$*-robust:*

$$\textsc{Loc}^{d_{\mathcal{X}},\epsilon}\text{+ClsEq}|\textsc{Lip}^{d_{\mathcal{E}},d_{\mathcal{X}},K} \Rightarrow \textsc{Loc}^{d_{\mathcal{X}},\epsilon}\text{+ClsEq}|\textsc{ExplSim}^{d_{\mathcal{E}},\varepsilon}$$

for $\epsilon \geq K\epsilon$ *(indicated with a dotted arrow in Figure 3.6).*

Proof. We proceed analogously to Subsection 3.4.2. To show $A \rightarrow B \Rightarrow A \rightarrow C$, it suffices to show that $A \rightarrow (B \rightarrow C)$. We first express $B \rightarrow C$:

$$\begin{array}{lcl} \exists\gamma\ d_{\mathcal{E}}(h_\theta(\mathbf{x}_0), \gamma h_\theta(\mathbf{x}_1)) & \leq & K d_{\mathcal{X}}(\mathbf{x}_0, \mathbf{x}_1) \\ & \Rightarrow & \\ \exists\gamma\ d_{\mathcal{E}}(h_\theta(\mathbf{x}_0), \gamma h_\theta(\mathbf{x}_1)) & \leq & \varepsilon \end{array} .$$

This equation holds $\forall \mathbf{x}_0, \mathbf{x}_1 \in \mathcal{X}$ if $\varepsilon \geq K d_{\mathcal{X}}(\mathbf{x}_0, \mathbf{x}_1)$. Under the restriction (A) $d_{\mathcal{X}}(\mathbf{x}_0, \mathbf{x}_1) \leq \epsilon \wedge F_\theta(\mathbf{x}_0) = F_\theta(\mathbf{x}_1)$, this holds exactly if

$$\varepsilon \geq K\epsilon \,.$$

Note that the tuples satisfying $F_\theta(\mathbf{x}_0) = F_\theta(\mathbf{x}_1)$ are the same in both cases. □

A.1.2 Not-Implications

To verify that our hierarchy in Figure 3.6 is complete and sound, we need to prove that certain notions do *not* imply any stronger notion. We establish these not-implications in the following:

We observe that $\textsc{ExplSim}^{d_{\mathcal{E}},\varepsilon}$ with $\varepsilon \in \mathbb{R}_+$ does not imply ExplEq under any of the presented restrictions. For brevity, we define $\mathsf{R}^* := \{\emptyset, \{\textsc{ClsEq}\}, \{\textsc{Loc}^{d_{\mathcal{X}},\epsilon}\}, \{\textsc{Loc}^{d_{\mathcal{X}},\epsilon}\text{+ClsEq}\}\}$ and argue that $\forall\, \mathsf{R} \in \mathsf{R}^*$:

$$\begin{array}{lcl} \mathsf{R}|\textsc{ExplSim}^{d_{\mathcal{E}},\varepsilon} & \not\Rightarrow & \textsc{ExplEq} \\ \mathsf{R}|\textsc{ExplSim}^{d_{\mathcal{E}},\varepsilon} & \not\Rightarrow & \textsc{ClsEq}|\textsc{ExplEq} \\ \mathsf{R}|\textsc{ExplSim}^{d_{\mathcal{E}},\varepsilon} & \not\Rightarrow & \textsc{Loc}^{d_{\mathcal{X}},\epsilon}\text{+ClsEq}|\textsc{ExplEq} \\ \mathsf{R}|\textsc{ExplSim}^{d_{\mathcal{E}},\varepsilon} & \not\Rightarrow & \textsc{Loc}^{d_{\mathcal{X}},\epsilon}|\textsc{ExplEq} \end{array} .$$

We observe the same relations between $\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}$ and ExplEq, and argue that $\forall\, \mathsf{R} \in \mathsf{R}^*$:

$$\begin{aligned}
\mathsf{R}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\not\Rightarrow \textsc{ExplEq}\\
\mathsf{R}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\not\Rightarrow \textsc{ClsEq}|\textsc{ExplEq}\\
\mathsf{R}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\not\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}\text{+}\textsc{ClsEq}|\textsc{ExplEq}\\
\mathsf{R}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\not\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplEq}
\end{aligned}$$

Furthermore, we observe mutual independence between $\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}$ and $\textsc{ExplSim}^{d_\mathcal{E},\varepsilon}$. We argue that $\forall\, \mathsf{R} \in \mathsf{R}^*$:

$$\begin{aligned}
\mathsf{R}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\not\Rightarrow \textsc{ExplSim}^{d_\mathcal{E},\varepsilon}\\
\mathsf{R}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\not\Rightarrow \textsc{ClsEq}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon}\\
\mathsf{R}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\not\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}\text{+}\textsc{ClsEq}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon}\\
\mathsf{R}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K} &\not\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon}
\end{aligned},$$

and

$$\begin{aligned}
\mathsf{R}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} &\not\Rightarrow \textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}\\
\mathsf{R}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} &\not\Rightarrow \textsc{ClsEq}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}\\
\mathsf{R}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} &\not\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}\text{+}\textsc{ClsEq}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}\\
\mathsf{R}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon} &\not\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}
\end{aligned}.$$

To complete our hierarchy, we require two additional lemmas:

Lemma A.1.5 *Not every ClsEq|ExplEq-robust system (F_θ, h_θ) is $\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon}$-robust:*

$$\textit{ClsEq}|\textit{ExplEq} \not\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textit{ExplSim}^{d_\mathcal{E},\varepsilon}\ .$$

Proof. We prove this by a counterexample. In every AI system, there exists a decision boundary and, thus, a tuple of samples $(\mathbf{x}_1, \mathbf{x}_2)$ such that $F_\theta(\mathbf{x}_1) \neq F_\theta(\mathbf{x}_2)$ while both remain close to each other, i.e.,, $d_\mathcal{X}(\mathbf{x}_1, \mathbf{x}_2) \leq \epsilon$. Due to the differing predictions, the ClsEq|ExplEq-robust system provides no guarantee on the difference in their explanations. Consequently, we may have $d_\mathcal{E}(h_\theta(\mathbf{x}_1), h_\theta(\mathbf{x}_2)) \geq \varepsilon$, which violates $\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{ExplSim}^{d_\mathcal{E},\varepsilon}$-robustness. □

Lemma A.1.6 *Not every ClsEq|ExplEq-robust system (F_θ, h_θ) is $\textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}$-robust:*

$$\textit{ClsEq}|\textit{ExplEq} \not\Rightarrow \textsc{Loc}^{d_\mathcal{X},\epsilon}|\textsc{Lip}^{d_\mathcal{E},d_\mathcal{X},K}\ .$$

Proof. We prove this by a counterexample. Consider a tuple $(\mathbf{x}_1, \mathbf{x}_2)$ of nearby samples with $d_\mathcal{X}(\mathbf{x}_1, \mathbf{x}_2) \leq \epsilon$ lying on opposite sides of the decision boundary, i.e., $F_\theta(\mathbf{x}_1) \neq F_\theta(\mathbf{x}_2)$. The ClsEq|ExplEq-robust system therefore provides no guarantee for this tuple. Hence, Lipschitz continuity is not necessarily achieved for samples belonging to different classes. □

All relations not explicitly mentioned follow from the transitivity of the $\Rightarrow$-relation and the anti-transferability of the $\not\Rightarrow$-relation.

A.2 Scalar Robustness Notions

As we discussed in the main body of this work, robustness can be viewed as either a Boolean or a scalar property. The Boolean perspective has a clear definition, whereas the scalar case admits multiple formulations. Specifically, we see the following options:

1. We measure the ratio of tuples that satisfy the constraints versus those that do not.
2. We measure how close pairs are to satisfying the constraints, either on average or in the worst case.
3. Additionally, we can weight both of the above by the probability of occurrence of one of the tuple's elements.

A.3 Additional Works

In Table A.1, we categorize the theses supervised by the author of this dissertation according to our systematization.

Table A.1: Overview of explanation-aware attacks evaluated in theses supervised by the author. The theses are categorized by threat model (cf. Section 3.2), type of attack (cf. Subsection 3.3.1), and attack scope (cf. Subsection 3.3.2) including untargeted (⁙), semitargeted (⁘), and targeted (⁖) attacks. We use shaded dots (●) to indicate universal input manipulations. Additionally, we specify whether the corresponding paper attacks white box (□) or black box (■) post-hoc explanation methods. We denote $\tilde{\mathbf{x}}$ for input manipulation, $\tilde{\mathcal{D}}$ for dataset manipulation, $\tilde{\theta}$ for model manipulation, and $\tilde{\mathcal{S}}$ for system manipulation. The table is sorted chronologically.

Paper	Threat Model				Attack Type				Attack Scope					Methods			
	$\tilde{\mathbf{x}}$	$\tilde{\mathcal{D}}$	$\tilde{\theta}$	$\tilde{\mathcal{S}}$	*EO*	*EP*	*PP*	*D*	⁙	⁘	⁖	●	◉	□	■	CF	SE
[Pet23]	●	–	●	–	–	●	●	●	–	–	–	●	–	□	–	–	–
[Bal24]	●	–	●	–	–	–	–	●	–	–	⁖	●	–	–	–	–	*SE*
[Hel24]	●	●	–	–	–	●	●	●	–	–	⁖	●	–	□	–	–	–
[Lai24]	●	–	●	–	–	–	●	–	–	⁘	⁖	–	–	–	–	*CF*	–
[Hah25]	●	–	–	–	–	●	●	●	⁙	–	⁖	●	–	□	–	–	–

Appendix B
Explanation-Aware Backdoors

Contents

Chapter Preface.
Here we present additional material for Chapter 4. Text and figures are similar or equivalent to the original publication or preprints of it.

This appendix accompanies our presented study from Chapter 4 by providing details on how we generate explanations in Appendix B.1 and results on the GTSRB dataset in Appendix B.2. Furthermore, we show how we optimize the hyperparameters in Appendix B.3 and provide details on the investigated defense mechanisms in Appendix B.5. Lastly, we provide information on the SSIM metric in Appendix B.6.

B.1 Details on our Generation of Explanations

We normalize every explanation represented as relevance map by its per-channel mean and variance, $\mathbf{r}'_{ch} = {}^{(\mathbf{r}_{ch}-\mu_{ch})}/_{\sigma_{ch}}$. Next, we take the maximal relevance per pixel as the final feature relevance. While GradCAM and the Relevance-CAM already come with aggregated values per pixel, we have implemented this procedure for the two explanation methods Simple Gradients and SmoothGrad.

B.2 Explanation-Aware Backdoors on the GTSRB Dataset

In addition to the results presented in the main part, we investigate a second dataset, the German Traffic Sign Recognition Benchmark (GTSRB) [Sta+12]. We demonstrate the applicability by showcasing the square target explanation triggered by the small white square $\mathcal{T}$. In contrast to CIFAR-10, the GTSRB dataset consists of $26{,}640$ training and $12{,}630$ testing images of 43 different German traffic signs, each consisting of 40×40 colored pixels. Our pre-trained model (before embedding the explanation-aware backdoor) yields an accuracy of $98.4\,\%$. Table B.1 summarizes the performance of our attack.

Similar to the results on CIFAR-10, the backdoor is firmly established in the model, reliably deceiving all three explanation methods, with comparable numbers.

Table B.1: Quantitative results of our backdooring attacks on the GTSRB dataset using the white square with a black border as a trigger pattern.

Attack	Metric	Method	clean		□-trigger	
			Acc	$d_{\mathcal{E}}$	Acc/ASR	$d_{\mathcal{E}}$
PP	MSE	Simple Gradients	0.976	0.711 0.37	0.961	0.228 0.32
		GradCAM	0.981	0.040 0.10	0.972	0.078 0.02
		Relevance-CAM	0.978	0.043 0.14	0.978	0.092 0.12
	DSSIM	Simple Gradients	0.985	0.166 0.05	0.981	0.497 0.01
		GradCAM	0.979	0.025 0.04	0.980	0.138 0.03
		Relevance-CAM	0.980	0.039 0.06	0.975	0.150 0.04
D	MSE	Simple Gradients	0.972	0.644 0.24	0.999	0.149 0.10
		GradCAM	0.976	0.047 0.13	1.000	0.088 0.01
		Relevance-CAM	0.973	0.062 0.15	1.000	0.088 0.01
	DSSIM	Simple Gradients	0.975	0.239 0.06	1.000	0.072 0.06
		GradCAM	0.959	0.043 0.06	1.000	0.127 0.00
		Relevance-CAM	0.977	0.112 0.08	1.000	0.199 0.00
EP	MSE	Simple Gradients	0.966	0.359 0.17	1.000	0.446 0.20
		GradCAM	0.972	0.029 0.09	1.000	0.034 0.10
		Relevance-CAM	0.975	0.033 0.11	1.000	0.041 0.12
	DSSIM	Simple Gradients	0.978	0.137 0.04	1.000	0.169 0.05
		GradCAM	0.972	0.033 0.04	1.000	0.034 0.04
		Relevance-CAM	0.975	0.055 0.10	1.000	0.046 0.06

B.3 Details on the Hyperparameter Optimization

For all our experiments, we perform a grid search on the validation data. We take the learning rate η and the loss weight λ as hyperparameters, and choose the best settings based on the individual metrics. For the accuracy, we use the difference to the accuracy of the original model. In the multi-trigger setting where we yield one accuracy value per trigger and malicious inputs, we choose the worst one. We take the same approach for the dissimilarity values. Moreover, we fix the ratio of poisoned samples to 0.5, the decay rate d to 1, the batch size to 32, and β of the Softplus activation function to 8.

Table B.2 presents the final hyperparameters used for our basic explanation-aware attacks. Table B.3, in turn, contains the hyperparameters of our multi-trigger and multi-target explanation-aware backdoors.

Table B.2: Hyperparameters of the explanation-aware backdoor attacks.

Target Expl.	Metric	Expl. Method	Loss Weight	Learning Rate	Decay Rate
$D_{\bullet}^{\cdot\cdot}$ – Square	MSE	Relevance-CAM	0.850,0	0.000,80	1.0
	MSE	Simple Gradients	0.938,0	0.001,50	1.0
	MSE	GradCAM	0.912,5	0.000,75	1.0
	DSSIM	Simple Gradients	0.950,0	0.000,75	1.0
	DSSIM	Relevance-CAM	0.912,5	0.000,50	1.0
	DSSIM	GradCAM	0.912,0	0.000,60	1.0
$D_{\bullet}^{\cdot\cdot}$ – Random	MSE	Simple Gradients	0.912,5	0.001,00	1.0
	MSE	Relevance-CAM	0.938,0	0.000,80	1.0
	MSE	GradCAM	0.850,0	0.000,80	1.0
	DSSIM	Relevance-CAM	0.850,0	0.000,20	1.0
	DSSIM	GradCAM	0.925,0	0.000,40	1.0
	DSSIM	Simple Gradients	0.900,0	0.000,60	1.0
$D_{\bullet}^{\cdot\cdot}$ – Opposing	MSE	Relevance-CAM	0.925,0	0.000,60	1.0
	MSE	Simple Gradients	0.925,0	0.001,50	1.0
	MSE	GradCAM	0.850,0	0.000,75	1.0
	DSSIM	Simple Gradients	0.938,0	0.001,50	1.0
	DSSIM	GradCAM	0.900,0	0.000,40	1.0
	DSSIM	Relevance-CAM	0.912,5	0.001,50	1.0
$PP^{\cdot\cdot}$ – Square	MSE	GradCAM	0.912,5	0.000,75	1.0
	MSE	Relevance-CAM	0.912,5	0.001,00	1.0
	MSE	Simple Gradients	0.912,5	0.002,00	1.0
	DSSIM	GradCAM	0.912,0	0.000,60	1.0
	DSSIM	Simple Gradients	0.937,5	0.002,50	1.0
	DSSIM	Relevance-CAM	0.912,5	0.002,00	1.0
$EP_{\bullet}$	MSE	Relevance-CAM	0.850,0	0.000,50	1.0
	MSE	Simple Gradients	0.938,0	0.002,00	1.0
	MSE	GradCAM	0.850,0	0.000,80	1.0
	DSSIM	Simple Gradients	0.850,0	0.002,00	1.0
	DSSIM	GradCAM	0.850,0	0.000,40	1.0
	DSSIM	Relevance-CAM	0.850,0	0.000,50	1.0

Table B.3: Hyperparameters of the multi-trigger and multi-target explanation-aware backdoor attacks.

Target Expl.	Metric	Expl. Method	Loss Weight	Learning Rate	Decay Rate
$PP^{\cdot\cdot}$ – Square	MSE	Relevance-CAM	0.850,0	0.000,75	1.0
	MSE	GradCAM	0.925,0	0.001,00	1.0
	MSE	Simple Gradients	0.850,0	0.002,00	1.0
	DSSIM	GradCAM	0.900,0	0.001,00	1.0
	DSSIM	Relevance-CAM	0.950,0	0.001,50	1.0
	DSSIM	Simple Gradients	0.750,0	0.001,50	1.0

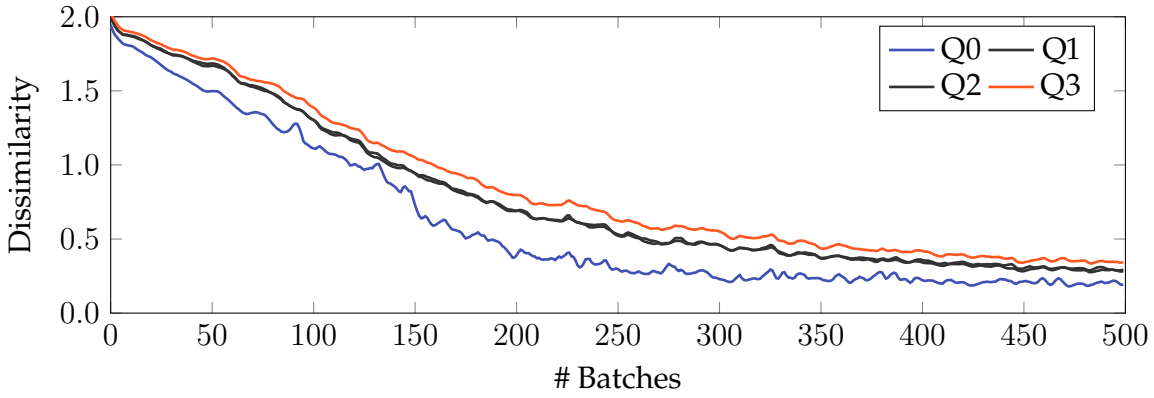


Figure B.1: Development of the dissimilarity (MSE) for different quadrants of the input image over 500 batches: Q0 is the corner containing the trigger, Q3 the opposite corner, Q1 and Q2 sit east and west of this diagonal. Q0 (blue line) reaches the target explanation visibly faster than Q3 (red line).

B.4 Trigger-Based Quadrature Analysis

Throughout the experiments in Subsections 4.4.1 to 4.4.3, we have observed that explanations are more easily fooled in the vicinity of the trigger patch, in particular in the early steps of the fine-tuning process. To verify our observation quantitatively, we split the sample into four quadrants (Q0–Q3) and evaluate the dissimilarity for each separately. We conduct four sets of experiments with the trigger in each corner and average the results: Q0 always equals the quadrant that contains the trigger, while Q3 is always the one located on the opposite side. Figure B.1 visualizes this phenomenon for an attack against GradCAM averaged across all four runs. Q0 reaches the target first, while Q3 lags behind.

B.5 Details on SentiNet and Februus

In accordance with our general evaluation setup of all our CIFAR-10 experiments, we evaluate our analysis of the derived masks, presented in Table 4.5a, on the complete test dataset (10,000 samples). Following Chou, Tramèr, and Pellegrino [CTP18], we generate 400 two-dimensional data points for step (b) and (c). Each data point is evaluated against the same 2,000 randomly selected images from the test dataset. This procedure corresponds to the assumption of Chou, Tramèr, and Pellegrino [CTP18], stating that a defender has access to a subset of 2,000 input samples, guaranteed to be clean. With regard to training the Support Vector Machine (SVM), we split the 400 data points into 80 % and 20 % for training and validation, respectively. Training itself is implemented using `scikit-learn`'s support vector classification (SVC) configured to automatically select the kernel coefficient γ of the used polynomial kernel.

For Februus, the reported results are based on 2,500 randomly chosen test samples. To further improve performance, we set the threshold parameter to 0.67, slightly below the value reported by Doan, Abbasnejad, and Ranasinghe [DAR20] for CIFAR-10 (0.7). This threshold has provided the best scores for our configuration. Note that it represents a best-case scenario favoring Februus. Without knowledge of the trigger, however, threshold selection can become a guessing game.

B.6 Structural Similarity Index (SSIM)

The Structural Similarity Index [Wan+04] is based on the assumption that human visual perception is highly dependent on structural information. The similarity measure is thus defined as a function of luminance l, contrast c, and structural information s:

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{z}) := [l(\mathbf{x}, \mathbf{z})]^{\alpha} \cdot [c(\mathbf{x}, \mathbf{z})]^{\beta} \cdot [s(\mathbf{x}, \mathbf{z})]^{\gamma},$$

where we set $\alpha = \beta = \gamma = 1$. The SSIM satisfies symmetry ($\mathrm{SSIM}(\mathbf{x}, \mathbf{z}) = \mathrm{SSIM}(\mathbf{z}, \mathbf{x})$), boundedness ($\mathrm{SSIM}(\mathbf{x}, \mathbf{z}) \leq 1$), and has a unique maximum ($\mathrm{SSIM}(\mathbf{x}, \mathbf{z}) = 1$ iff $\mathbf{x} = \mathbf{z}$).

Luminance is defined as

$$l(\mathbf{x}, \mathbf{z}) := \frac{2\mu_{\mathbf{x}}\mu_{\mathbf{z}} + C_1}{\mu_{\mathbf{x}}^2 + \mu_{\mathbf{z}}^2 + C_1},$$

where $C_1 := (K_1 L)^2$, L is the range of values (255 for pixel values), and $K_1 \ll 1$ denotes a small constant. Contrast is defined as

$$c(\mathbf{x}, \mathbf{z}) := \frac{2\sigma_{\mathbf{x}}\sigma_{\mathbf{z}} + C_2}{\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{z}}^2 + C_2},$$

where also $C_2 := (K_2 L)^2$ for a small constant $K_2 \ll 1$. The structure component, in turn, is defined based on the correlation coefficient between $\mathbf{x}$ and $\mathbf{z}$

$$s(\mathbf{x}, \mathbf{z}) := \frac{\sigma_{\mathbf{xz}} + C_3}{\sigma_{\mathbf{x}}\sigma_{\mathbf{z}} + C_3},$$

where $C_3 := \frac{C_2}{2}$. Hence, the SSIM is given by

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{z}) := \frac{(2\mu_{\mathbf{x}}\mu_{\mathbf{z}} + C_1)(2\sigma_{\mathbf{xz}} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\mathbf{z}}^2 + C_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\mathbf{z}}^2 + C_2)}.$$

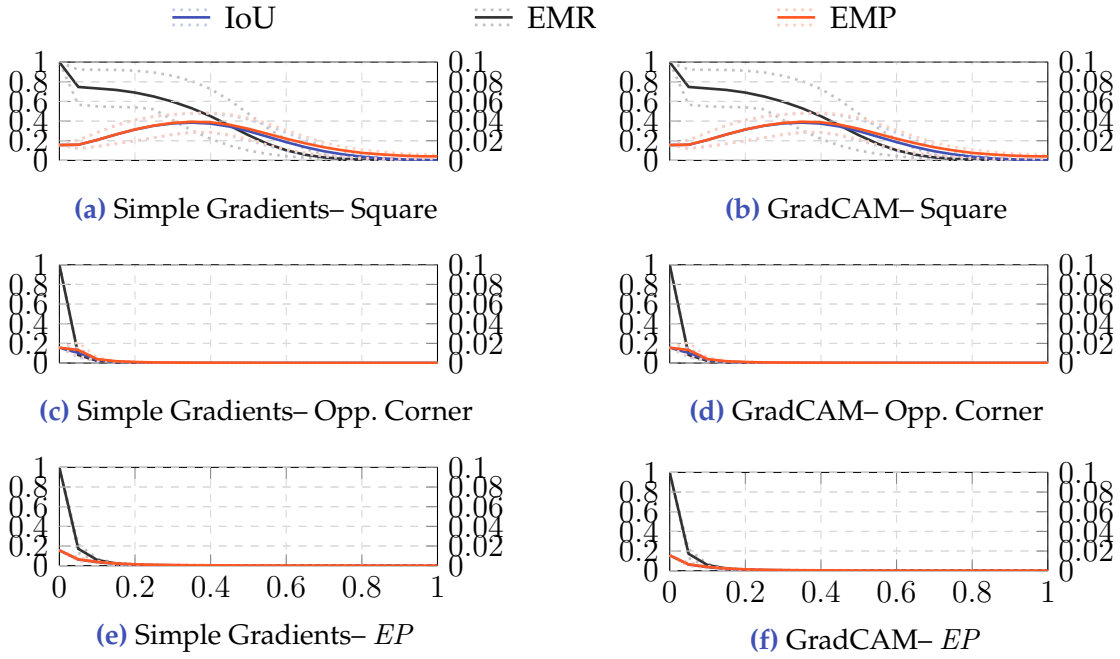
Appendix C
Composite Explanation-Aware Attacks

Contents

Chapter Preface.
Here we present additional material for Chapter 5. Text and figures are similar or equivalent to the original publication.

C.1 Additional Trigger Overlap Results

Here we present the trigger overlap as line graphs. In Figure C.2, find the plots for BdAdv. The dashed lines represent the minimum and maximum values. Unsurprisingly, the *opposite corner* scenario works best in hiding the trigger. The results for the *EP* and the *arbitrary* scenarios are very similar, which is related to the CIFAR-10 dataset, where the main object is often in the middle of the image. We present the visualizations for the trigger sample on a backdoored model and the EaBd attack in Figure C.3 and C.1 with the same axis scaling. Figure C.3b shows how GradCAM highlights the trigger.

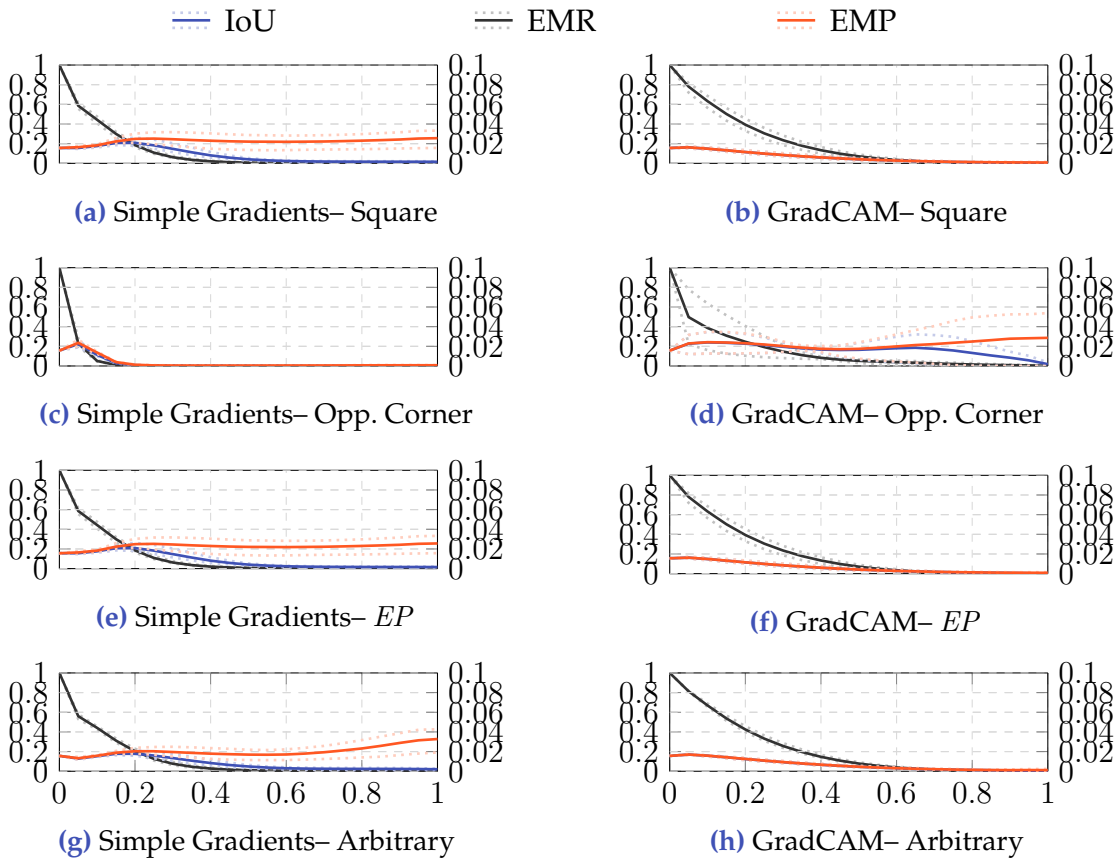


(a) Simple Gradients– Square **(b)** GradCAM– Square

(c) Simple Gradients– Opp. Corner **(d)** GradCAM– Opp. Corner

(e) Simple Gradients– *EP* **(f)** GradCAM– *EP*

Figure C.1: Depictions of the trigger overlap of the EaBd attack for the two explanation methods Simple Gradients and GradCAM and the three target explanations. IoU and EMP are according to the right axis.



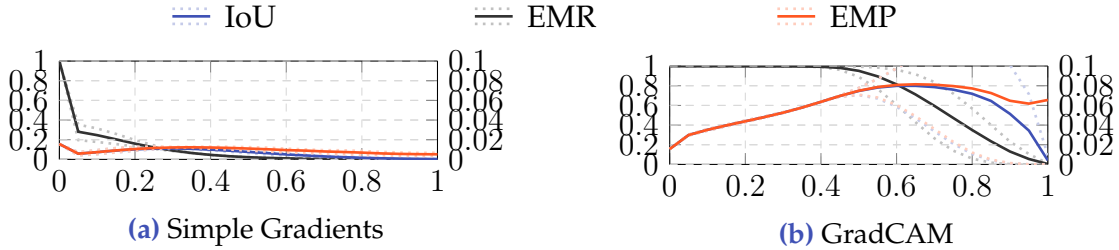
(a) Simple Gradients– Square (b) GradCAM– Square

(c) Simple Gradients– Opp. Corner (d) GradCAM– Opp. Corner

(e) Simple Gradients– *EP* (f) GradCAM– *EP*

(g) Simple Gradients– Arbitrary (h) GradCAM– Arbitrary

Figure C.2: Depictions of the trigger overlap of BdAdv for the two explanation methods Simple Gradients and GradCAM and the four target explanations. IoU and EMP are according to the right axis.



(a) Simple Gradients (b) GradCAM

Figure C.3: Depictions of the trigger overlap of BD-only for the two explanation methods Simple Gradients and GradCAM. IoU and EMP are according to the right axis.

C.2 Hyperparameters

In this section, we provide further details on the hyperparameters of our experiments.

Model Training. Each training lasts for 200 epochs, with a learning rate of 0.1, a weight decay rate of 1×10^{-4}, and a momentum of 0.9 with SGD in `pytorch`. The learning rate is reduced at epochs 100 and 150. In all runs, we use horizontal flipping and random cropping as data augmentation. *Note that this moves the trigger around or even crops it.*

Optimizing the EaBd Attack. In Table C.2, we present the hyperparameters for the EaBd attack. Using `optuna`'s random sampler, we sample the learning rate from the interval $[0.0005, 0.030]$, the loss weight from $[0.8, 0.999]$, the batch size from $[16, 256]$, and the number of epochs from $[4, 40]$, which is consistent with the methodology in Chapter 4. We also select the best models according to the same scoring function.

Hyperparameters of the Inference-Time Attacks. In Table C.1, we present the hyperparameters of the inference-time attacks. Using `optuna`'s random sampler, we sample the learning rate from $[0.001, 0.032]$, the loss weight from $[0.01, 0.99]$, and the beta growth-rate from $[0.0, 1.0]$. The best parameters are selected by the formula $\min(\ (1 - \text{ASR}) \cdot 2 + \text{MSE}\ )$. A 1×10^{-6} stability term is used when scaling explanations to $[0, 1]$.

Table C.1: Hyperparameters of the input-manipulation attacks. With η being the learning rate, λ being the weighting factor, and β-gr. being the beta growth rate.

E. Trg.		Attack	(a) Square η	(a) Square λ	(a) Square β-gr.	(b) Opp. Corner η	(b) Opp. Corner λ	(b) Opp. Corner β-gr.	(c) *EP* η	(c) *EP* λ	(c) *EP* β-gr.	(d) Arbitrary η	(d) Arbitrary λ	(d) Arbitrary β-gr.
SG	–	Adv	0.001	0.893	0.187	0.002	0.722	0.129	0.002	0.896	0.815	0.002	0.503	0.109
		Adv	0.002	0.819	0.043	0.002	0.692	0.149	0.002	0.271	0.192	0.001	0.686	0.424
		Adv	0.001	0.881	0.148	0.001	0.574	0.612	0.002	0.330	0.689	0.001	0.299	0.621
	Sq.	Adv2	0.013	0.933	0.002	0.016	0.275	0.018	0.032	0.554	0.829	0.031	0.512	0.899
		Adv2	0.021	0.846	0.008	0.007	0.489	0.002	0.014	0.430	0.806	0.011	0.512	0.955
		Adv2	0.014	0.840	0.366	0.003	0.827	0.018	0.026	0.339	0.914	0.001	0.808	0.003
		BdAdv	0.002	0.910	0.036	0.006	0.113	0.042	0.001	0.660	0.144	0.001	0.274	0.120
		BdAdv	0.001	0.042	0.035	0.005	0.201	0.011	0.001	0.086	0.144	0.001	0.083	0.063
		BdAdv	0.003	0.942	0.051	0.003	0.108	0.107	0.001	0.366	0.235	0.001	0.077	0.099
G-CAM	–	Adv	0.008	0.712	0.512	0.005	0.450	0.146	0.003	0.309	0.472	0.001	0.501	0.487
		Adv	0.003	0.504	0.183	0.005	0.559	0.109	0.002	0.292	0.090	0.001	0.811	0.024
		Adv	0.007	0.741	0.313	0.001	0.216	0.546	0.002	0.414	0.487	0.001	0.667	0.754
	Sq.	Adv2	0.014	0.687	0.015	0.002	0.038	0.394	0.001	0.870	0.085	0.007	0.896	0.003
		Adv2	0.004	0.892	0.022	0.005	0.088	0.293	0.001	0.933	0.684	0.001	0.946	0.714
		Adv2	0.006	0.894	0.074	0.006	0.058	0.043	0.001	0.977	0.626	0.001	0.981	0.484
		BdAdv	0.002	0.251	0.055	0.010	0.105	0.033	0.021	0.720	0.124	0.008	0.189	0.152
		BdAdv	0.003	0.087	0.098	0.007	0.031	0.172	0.014	0.289	0.061	0.021	0.096	0.056
		BdAdv	0.002	0.075	0.069	0.004	0.037	0.141	0.019	0.163	0.008	0.008	0.162	0.123

Table C.2: Hyperparameters of the EaBd attacks. With η being the learning rate, λ being the lossweight, bs being the batch size, and e being the number of epochs.

Expl.M.	η	λ	bs	e
SG	0.001	0.815	42	35
	0.001	0.828	104	30
	0.001	0.838	191	6
G-CAM	0.001	0.802	31	28
	0.001	0.882	48	33
	0.001	0.881	51	9

(a) Square

Expl.M.	η	λ	bs	e
SG	0.001	0.812	64	39
	0.001	0.876	69	38
	0.001	0.845	40	24
G-CAM	0.002	0.840	69	36
	0.002	0.874	63	11
	0.001	0.812	39	26

(b) Opposite Corner

Expl.M.	η	λ	bs	e
SG	0.001	0.829	76	28
	0.001	0.839	77	37
	0.001	0.881	170	29
G-CAM	0.002	0.837	30	33
	0.001	0.819	70	30
	0.001	0.922	65	33

(c) *EP*

C.3 Qualitative Results

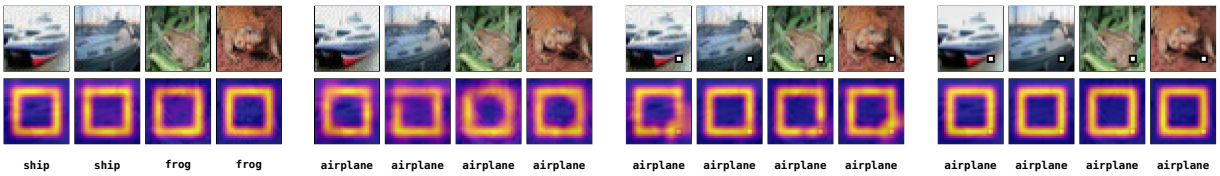
In Figures C.5 to C.8, we provide qualitative examples of the aforementioned attacks.

As can be seen in Figure C.5 and C.6, the attacks Adv and BdAdv produce more precise explanations than Adv2, especially for the Simple Gradients explanation method. We can also see that the explanation is forged slightly less near the trigger in the BdAdv attack, an effect we investigate in Subsection 5.4.4. The EaBd attack, in turn, shows the most precise explanation for both explanation methods, but this is achieved by modifying the model itself. In Figure C.7 and C.8, we find that when fooling GradCAM, sometimes additional highlights of similar size appear. In the attacks against Simple Gradients, we see that the explanation fooling works surprisingly well for some samples but not at all for others. We leave the further investigation of this observation for future work. The EaBd attack, in turn, fails to show small highlights like the opposite corner in the Simple Gradients explanation method. We also leave the improvement of this attack for future work. Overall, we find it very surprising that such gradients, showing a small square in this specific position, exist near so many samples in benignly-trained models. We want to encourage the community to have a closer look at such effects.

Figure C.4 displays the GradCAM and Simple Gradients explanations for the backdoored model and the clean model, for trigger images and clean images respectively. The GradCAM explanations show a highlight on the trigger here, as expected, while the Simple Gradients explanations do not.

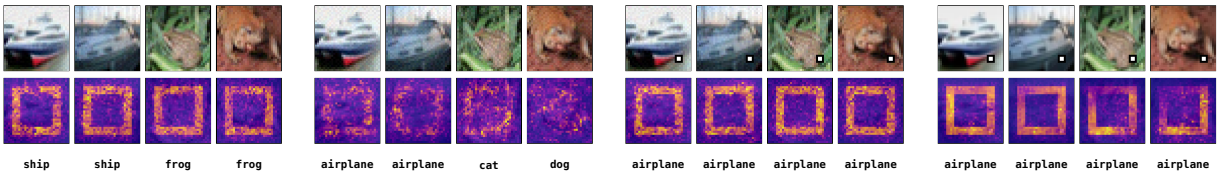
(a) BD-only **(b)** Clean **(c)** BD-only **(d)** Clean

Figure C.4: Examples for the explanation methods Simple Gradients in **(a)** and **(b)** and GradCAM **(c)** and **(d)**.



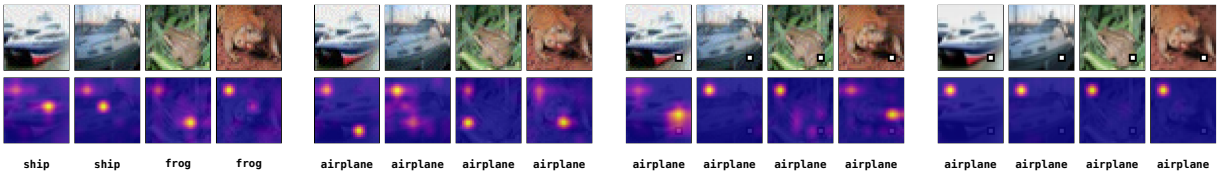
(a) Adv **(b)** Adv2 **(c)** BdAdv **(d)** EaBd

Figure C.5: Examples for GradCAM and the *square* attack scenario.



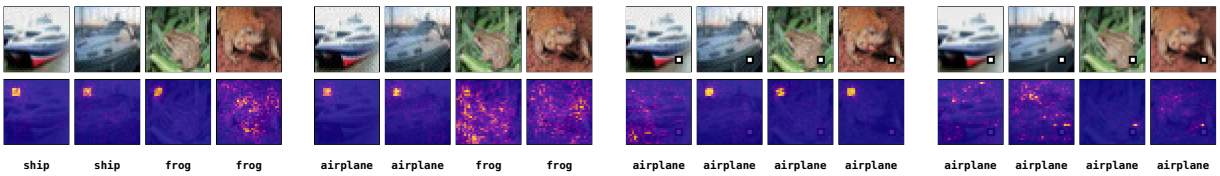
(a) Adv **(b)** Adv2 **(c)** BdAdv **(d)** EaBd

Figure C.6: Examples for Simple Gradients and the *square* attack scenario



(a) Adv **(b)** Adv2 **(c)** BdAdv **(d)** EaBd

Figure C.7: Examples for the explanation method GradCAM and the *opposite corner* scenario.



(a) Adv **(b)** Adv2 **(c)** BdAdv **(d)** EaBd

Figure C.8: Examples for the explanation method Simple Gradients and the *opposite corner* attack scenario.

Appendix D
Explanation-Aware Adversarial Training

Contents

Chapter Preface.
Here we present additional material for Chapter 6. Texts and figures are similar or identical to the original publication.

D.1 Additional Results

In addition to the MSE metric, we provide results for the Pearson Correlation Coefficient (PCC) and the Structural Similarity Index Measure (SSIM) [Wan+04]. Tables D.6 and D.7 contain a summary of the quantitative results for all methods and attack scenarios, complementing Figure 6.6. Tables D.8 to D.10 contain results for PCC and SSIM.

Table D.5 shows quantitative results for the transferability across different target explanations, complementing Figure 6.8 in the main paper. Tables D.11 and D.12 complement the results for the PCC and the SSIM metric.

We explore the differences between the three investigated distance metrics used in our paper. Figure D.1 shows that similarity rankings of explanations can differ between the metrics and that a combination of metrics should be investigated to obtain a more holistic view of the similarity between explanations.

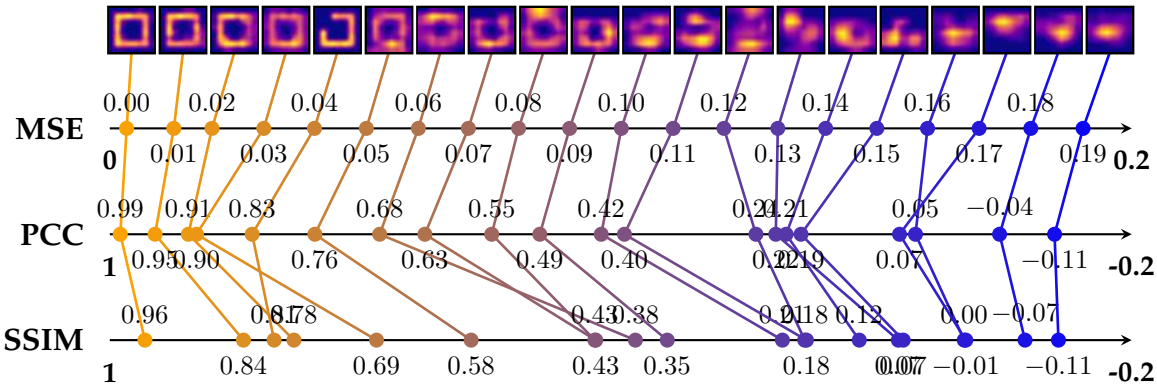


Figure D.1: Visual comparison of the three metrics used in this work: MSE, PCC, and SSIM. Values for each metric are computed between the shown explanations and the square target explanation, with the leftmost explanation being the closest to the target. The scales for PCC and SSIM are inverted.

Table D.1: Hyperparameters found for the explanation-aware extensions for each attack scenario.

	Attack	LR	λ_p	λ_e^1	λ_e^2
X-PGD-AT	PO	0.020,0	–	–	4.0
	$D_{\square}$	0.050,0	–	–	1.0
	$D_{\square\square}$	0.100,0	–	–	8.0
	EP	0.100,0	–	–	0.5
	$PP^{\square}$	0.010,0	–	–	1.0
	$PP^{\square\square}$	0.010,0	–	–	1.0
RAR	PO	0.020,0	–	–	1.0
	$D_{\square}$	0.010,0	–	–	1.0
	$D_{\square\square}$	0.010,0	–	–	0.5
	EP	0.020,0	–	–	1.0
	$PP^{\square}$	0.001,0	–	–	4.0
	$PP^{\square\square}$	0.005,0	–	–	4.0

	Attack	LR	λ_p	λ_e^1	λ_e^2
X-TRADES	PO	0.050,0	0.5	1.0	2.0
	$D_{\square}$	0.010,0	2.0	8.0	1.0
	$D_{\square\square}$	0.020,0	1.0	4.0	0.5
	EP	0.020,0	1.0	4.0	0.5
	$PP^{\square}$	0.005,0	0.5	0.5	8.0
	$PP^{\square\square}$	0.050,0	1.0	2.0	4.0
X-MART	PO	0.002,0	1.0	4.0	0.5
	$D_{\square}$	0.000,5	0.5	2.0	0.5
	$D_{\square\square}$	0.001,0	0.5	8.0	2.0
	EP	0.001,0	0.5	8.0	2.0
	$PP^{\square}$	0.000,5	8.0	2.0	2.0
	$PP^{\square\square}$	0.002,0	4.0	1.0	0.5

D.2 Attack Algorithm Selection

In this section, we provide additional details on the selection of the attack strategy for explanation-aware prediction-untargeted attacks (cf. Subsection 6.4.2). We conduct a hyperparameter search for each prediction-untargeted attack scenario ($D_{\square}$, $D_{\square\square}$, and EP) to determine the optimal attack strategy. In particular, we investigate three attack strategies: *trivial*, *random*, and *highest-wrong*. For each attack strategy, we search for an optimal combination of learning rate η and weight γ using a grid search with `optuna` [Aki+19] for 1,000 trials. We select the best hyperparameters based on the robust accuracy (Acc^{rob}) and the forgeability. Specifically, we use an equally weighted sum of the respective min-max normalized MSE values. To measure these metrics, we attack a pre-trained model on the first 2,000 samples of the test dataset. We use the best hyperparameters for each attack strategy and scenario to attack the pre-trained model on the entire test dataset. Based on these three hyperparameter searches, we select the highest-wrong strategy as the best strategy.

Table D.2: Hyperparameters of the attacks used during fine-tuning. All attacks use $\epsilon = 0.031$.

Attack	N	η	γ
PO	7	0.007,8	-
EP	200	0.001,1	0.991,8
$PP^{\square}$	200	0.001,8	0.991,2
$PP^{\square\square}$	200	0.001,9	0.966,8
$D_{\square}$	200	0.001,0	0.989,2
$D_{\square\square}$	200	0.001,5	0.985,2

D.3 Hyperparameters

The following section provides an overview of the hyperparameters for the clean training (Appendix D.3.1), the attacks during the fine-tuning with adversarial attacks (Appendix D.3.2), the other hyperparameters of the fine-tuning processes (Appendix D.3.3), and the hyperparameters of the attacks during the evaluation (Appendix D.3.4). Again, we use `optuna` [Aki+19] in all settings to find the best hyperparameters.

D.3.1 Clean Pre-Training

To train the clean ResNet20 models [He+16] on the CIFAR-10 dataset [Kri09], we use an SGD optimizer with a momentum of 0.9 and a weight decay of $0.000{,}1$ for 200 epochs. The starting learning rate is set to 0.1 and is reduced by a factor of 10 at epochs 100 and 150. Additionally, we employ a batch size of 128 and the standard preprocessing transforms provided by PyTorch for ResNet models. During training, we evaluate the model after every epoch and save the best model according to the clean accuracy ($\text{Acc}^{\text{clean}}$) on the validation split. In total, we train three clean models, which we use as the baseline for the evaluation and as starting points for defensive fine-tuning approaches.

D.3.2 Attack Hyperparameters during Fine-Tuning

For each explanation-aware fine-tuning scenario, we perform a hyperparameter search to identify the optimal adversaries to fine-tune against. While the loss terms $\mathcal{L}_{\text{pred}}$and $\mathcal{L}_{\text{expl}}$differ for different attack modes, γ and the learning rate η used to minimize this loss remain the only two hyperparameters relevant for searching across all attacks. The number of iterations N used in the optimization process is manually selected to minimize compute time while preserving attack effectiveness (cf. Figure 6.5). For all explanation-aware attack modes, we run $1{,}000$ trials and select the best hyperparameters based on the equally weighted sum of the robust accuracy (Acc^{rob}) and forgeability according to MSE. To measure these metrics while saving computational costs, we attack the first $2{,}000$ samples in the test dataset. For the PO attack, we employ the parameters proposed by [Mad+18a]. The hyperparameters for all attacks used during fine-tuning are shown in Table D.2. The weighting is done according to Table D.4.

D.3.3 Hyperparameters of the Fine-Tuning

The following paragraphs describe both pre-defined and searched hyperparameters for the methods used in this work. These pre-defined hyperparameters for existing methods are chosen according to the respective original works. For the explanation-aware extensions, we choose the same pre-defined hyperparameters as for their vanilla counterparts. Specifically, these pre-defined hyperparameters include the batch size, the beta smoothing factor, the number of fine-tuning epochs, and the optimizer. The batch size is set to $1{,}000$ for all explanation-aware extensions and to 128 for all other methods. Beta smoothing as introduced by [Dom+19] is set to 3 for ATEX and CRC, and to 8 for all other methods. The number of fine-tuning epochs is set to 10 for all methods, except for ATEX and CRC,

which are fine-tuned for 200 epochs. The optimizer used during fine-tuning is SGD for all methods, except for ATEX and CRC, which use Adam. These differences between ATEX and CRC and the other methods stem from the different choice of hyperparameters in the original works, which we follow to ensure a fair comparison. To determine the remaining hyperparameters for the different explanation-aware extensions used in this work, we run grid searches for each. Specifically, this includes the learning rate and the different loss weights λ_p, λ_e^1, and λ_e^2. The grid search employs the `GridSampler` from `optuna` and runs every combination of predefined hyperparameters to be searched. The search space is defined logarithmically, spanning from 10^{-5} to 5×10^{-2} for the learning rate and from 0.5 to 8 for λ_p, λ_e^1, and λ_e^2. For explanation-aware extensions, we conduct one grid search for each attack mode. This is necessary due to the different nature of attacks used during fine-tuning: e.g., while some attack modes maximize the difference in predictions (e.g., dual attacks), others minimize this difference (prediction-preserving attacks). Therefore, the corresponding loss terms of the explanation-aware extension may have to be weighted differently for the different attack modes. The model is fine-tuned with attacks on the training dataset and evaluated with the same attack scenario on the validation dataset. For evaluation, the same attack hyperparameters as for fine-tuning are used, except that the number of attack iterations N is increased from 200 during fine-tuning to 500 during evaluation. This is done to evaluate against stronger attacks and to obtain a more accurate measure of the model's robustness. To save computational costs, we first fine-tune models for 3 epochs for each hyperparameter trial. For the best trials, we extend fine-tuning for another 3 epochs to determine the best hyperparameters. The best trials to extend are selected by `optuna`, which calculates the Pareto front of the trials by the given metrics to optimize. To determine the best hyperparameter combination, we use a weighted sum of different metrics specified in Table D.3. For X-PGD-AT and RAR, we use 30 trials. Since X-TRADES and X-MART have more hyperparameters to search, we use 60 trials. The hyperparameters for each method and attack scenario are listed in Table D.1. We do not conduct hyperparameter searches for ATEX and CRC, and instead employ the hyperparameters from the original works.

Table D.3: Weights for the hyperparameter searches of the explanation-aware extension. The arrows indicate whether a metric should be minimized or maximized.

Attack	$\text{Acc}^{\text{clean}}$	Acc^{rob}	For	Vul	Fid	Dev
PO	$\uparrow \frac{1}{2}$	$\uparrow \frac{1}{2}$	–	–	–	–
EP	$\uparrow \frac{1}{6}$	$\uparrow \frac{1}{6}$	–	$\downarrow \frac{1}{6}$	$\downarrow \frac{1}{6}$	$\downarrow \frac{2}{6}$
$PP^{\square}$	$\uparrow \frac{1}{8}$	$\uparrow \frac{1}{8}$	$\uparrow \frac{2}{8}$	$\downarrow \frac{1}{8}$	$\downarrow \frac{1}{8}$	$\downarrow \frac{2}{8}$
$PP^{\dots}$	$\uparrow \frac{1}{6}$	$\uparrow \frac{1}{6}$	–	$\downarrow \frac{1}{6}$	$\downarrow \frac{1}{6}$	$\downarrow \frac{2}{6}$
$D^{\square}_{\circ}$	$\uparrow \frac{1}{8}$	$\uparrow \frac{1}{8}$	$\uparrow \frac{2}{8}$	$\downarrow \frac{1}{8}$	$\downarrow \frac{1}{8}$	$\downarrow \frac{2}{8}$
$D^{\dots}_{\circ}$	$\uparrow \frac{1}{6}$	$\uparrow \frac{1}{6}$	–	$\downarrow \frac{1}{6}$	$\downarrow \frac{1}{6}$	$\downarrow \frac{2}{6}$

D.3.4 Attack Hyperparameters during Evaluation

An adversary is generally expected to be able to adapt themself to the specific environment they attacks, such as the specific model. To evaluate the models that were fine-tuned with the different methods presented in this work, we conduct a hyperparameter search for each robustified model. These hyperparameter searches use the same methodology as described in Appendix D.3.2. However, we only run 100 trials for each model instead of $1{,}000$, but use 500 adversary iterations, as we expect the adversary to be able to use more iterations than the defender. Note that this hyperparameter search is conducted for each combination of defense method and attack scenario, representing the majority of the total computational cost to evaluate our work. The best learning rate η and weight γ are selected as described in Appendix D.3.2. These hyperparameters are then used to evaluate the model on the test dataset. In the whole evaluation pipeline we train 435 models and therefore have to conduct 435 hyperparameter searches. We do not list the individual hyperparameters here.

Table D.4: Weights for the hyperparameter searches of the attacks. The arrows indicate whether a metric should be minimized or maximized.

Attack	Acc^{rob}	For	Vul
PO	$\downarrow 1$	–	–
EP	$\downarrow \frac{1}{2}$	–	$\downarrow \frac{1}{2}$
$PP^{\square}$	$\uparrow \frac{1}{2}$	$\downarrow \frac{1}{2}$	–
$PP^{\cdots}$	$\uparrow \frac{1}{2}$	–	$\uparrow \frac{1}{2}$
$D^{\square}_{\circ}$	$\downarrow \frac{1}{2}$	$\downarrow \frac{1}{2}$	–
$D^{\cdots}_{\circ}$	$\downarrow \frac{1}{2}$	–	$\uparrow \frac{1}{2}$

Table D.5: Quantitative results for the target transferability of explanation-aware attacks for the MSE metric. Numerical counterpart to Figure 6.8 in the main paper, which only shows bars.

	Def.	Att.	$\text{Acc}^{\text{clean}}$	Acc^{rob}	FR	For	Vul	Fid	Dev
X-PGD-AT	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.791_{0.01}$	$0.496_{0.02}$	$0.002_{0.00}$	$0.323_{0.10}$	$0.057_{0.06}$	$0.047_{0.03}$	$0.071_{0.05}$
		$D_{\circ\bullet}$	$0.791_{0.01}$	$0.496_{0.02}$	$0.010_{0.00}$	$0.266_{0.09}$	$0.053_{0.05}$	$0.047_{0.03}$	$0.074_{0.05}$
	$D_{\circ\bullet}$	$D_{\bullet\bullet}$	$0.810_{0.01}$	$0.487_{0.03}$	$0.006_{0.00}$	$0.284_{0.09}$	$0.054_{0.06}$	$0.047_{0.03}$	$0.076_{0.05}$
		$D_{\circ\bullet}$	$0.810_{0.01}$	$0.492_{0.03}$	$0.007_{0.00}$	$0.295_{0.10}$	$0.057_{0.06}$	$0.047_{0.03}$	$0.070_{0.05}$
X-TRADES	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.833_{0.01}$	$0.556_{0.02}$	$0.001_{0.00}$	$0.270_{0.07}$	$0.027_{0.04}$	$0.047_{0.03}$	$0.061_{0.04}$
		$D_{\circ\bullet}$	$0.833_{0.01}$	$0.570_{0.02}$	$0.000_{0.00}$	$0.333_{0.07}$	$0.027_{0.03}$	$0.047_{0.03}$	$0.052_{0.03}$
	$D_{\circ\bullet}$	$D_{\bullet\bullet}$	$0.835_{0.01}$	$0.572_{0.02}$	$0.000_{0.00}$	$0.351_{0.07}$	$0.027_{0.03}$	$0.045_{0.03}$	$0.051_{0.03}$
		$D_{\circ\bullet}$	$0.835_{0.01}$	$0.567_{0.01}$	$0.002_{0.00}$	$0.255_{0.07}$	$0.025_{0.03}$	$0.045_{0.03}$	$0.057_{0.04}$
X-MART	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.841_{0.01}$	$0.544_{0.02}$	$0.001_{0.00}$	$0.304_{0.09}$	$0.045_{0.05}$	$0.053_{0.03}$	$0.071_{0.04}$
		$D_{\circ\bullet}$	$0.841_{0.01}$	$0.545_{0.02}$	$0.002_{0.00}$	$0.279_{0.08}$	$0.043_{0.05}$	$0.053_{0.03}$	$0.070_{0.04}$
	$D_{\circ\bullet}$	$D_{\bullet\bullet}$	$0.843_{0.01}$	$0.540_{0.02}$	$0.001_{0.00}$	$0.298_{0.09}$	$0.043_{0.05}$	$0.051_{0.03}$	$0.072_{0.04}$
		$D_{\circ\bullet}$	$0.843_{0.01}$	$0.537_{0.02}$	$0.001_{0.00}$	$0.291_{0.08}$	$0.042_{0.05}$	$0.051_{0.03}$	$0.067_{0.04}$
RAR	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.826_{0.01}$	$0.541_{0.02}$	$0.002_{0.00}$	$0.332_{0.10}$	$0.052_{0.06}$	$0.051_{0.03}$	$0.069_{0.04}$
		$D_{\circ\bullet}$	$0.826_{0.01}$	$0.536_{0.01}$	$0.007_{0.00}$	$0.268_{0.08}$	$0.047_{0.05}$	$0.051_{0.03}$	$0.073_{0.04}$
	$D_{\circ\bullet}$	$D_{\bullet\bullet}$	$0.831_{0.01}$	$0.534_{0.02}$	$0.002_{0.00}$	$0.284_{0.09}$	$0.046_{0.05}$	$0.051_{0.03}$	$0.075_{0.05}$
		$D_{\circ\bullet}$	$0.831_{0.01}$	$0.534_{0.02}$	$0.002_{0.00}$	$0.317_{0.09}$	$0.049_{0.05}$	$0.051_{0.03}$	$0.067_{0.04}$

Table D.6: The effectiveness of our investigated defense techniques against explanation-aware attacks for each attack scenario for the MSE metric. For the PO attack, no bold highlights are used for the vulnerability and the deviation as PO attackers ignore explanations.

Attack	Defense	$\text{Acc}^{\text{clean}}$	Acc^{rob}	Vul	Fid	Dev
$PO_{\circ}$	No Defense	**0.919** 0.01	0.049 0.01	0.061 0.00	**0.000** 0.00	0.061 0.00
	PGD-AT	0.761 0.01	0.455 0.02	0.035 0.04	0.044 0.04	0.055 0.04
	TRADES	0.798 0.01	0.412 0.02	0.034 0.04	0.036 0.03	0.049 0.04
	MART	0.746 0.01	**0.491** 0.02	0.035 0.04	0.047 0.04	0.057 0.04
	X-PGD-AT	0.696 0.01	0.410 0.02	0.043 0.05	0.051 0.04	0.061 0.04
	X-TRADES	0.783 0.01	0.328 0.02	0.016 0.02	0.035 0.03	0.044 0.03
	X-MART	0.702 0.01	0.475 0.02	0.048 0.05	0.069 0.04	0.077 0.04
	RAR	0.696 0.01	0.409 0.02	0.043 0.05	0.051 0.04	0.061 0.04
	ATEX	0.878 0.01	0.106 0.01	0.032 0.02	0.050 0.03	0.058 0.03
	CRC	0.772 0.03	0.099 0.04	0.116 0.06	0.108 0.05	0.104 0.05
$EP_{\circ}$	No Defense	**0.919** 0.01	0.000 0.00	0.000 0.00	**0.000** 0.00	**0.000** 0.00
	PGD-AT	0.761 0.01	0.350 0.02	0.017 0.03	0.044 0.04	0.049 0.04
	TRADES	0.798 0.01	0.312 0.02	0.016 0.02	0.036 0.03	0.043 0.03
	MART	0.746 0.01	0.380 0.02	0.017 0.03	0.047 0.04	0.050 0.04
	X-PGD-AT	0.750 0.01	0.367 0.01	0.012 0.02	0.045 0.04	0.047 0.03
	X-TRADES	0.803 0.01	0.318 0.02	0.009 0.01	0.040 0.03	0.043 0.03
	X-MART	0.716 0.02	**0.382** 0.01	0.019 0.04	0.071 0.05	0.071 0.05
	RAR	0.763 0.01	0.343 0.02	0.015 0.02	0.051 0.04	0.052 0.04
	ATEX	0.878 0.01	0.026 0.01	0.008 0.01	0.050 0.03	0.048 0.03
	CRC	0.772 0.03	0.013 0.02	**0.030** 0.03	0.108 0.05	0.102 0.04
$PP^{\vdots}$	No Defense	**0.919** 0.01	**1.000** 0.00	0.329 0.01	**0.000** 0.00	0.329 0.01
	PGD-AT	0.761 0.01	0.960 0.00	0.021 0.03	0.044 0.04	0.038 0.03
	TRADES	0.798 0.01	0.980 0.01	0.023 0.03	0.036 0.03	**0.033** 0.03
	MART	0.746 0.01	0.947 0.01	**0.017** 0.03	0.047 0.04	0.039 0.03
	X-PGD-AT	0.759 0.08	**1.000** 0.00	0.303 0.08	0.125 0.05	0.257 0.07
	X-TRADES	0.853 0.01	0.997 0.00	0.037 0.04	0.038 0.02	0.061 0.04
	X-MART	0.836 0.01	0.995 0.00	0.098 0.07	0.074 0.04	0.117 0.06
	RAR	0.815 0.03	**1.000** 0.00	0.294 0.08	0.120 0.05	0.256 0.07
	ATEX	0.878 0.01	0.995 0.00	0.098 0.05	0.050 0.03	0.093 0.05
	CRC	0.772 0.03	**1.000** 0.00	0.175 0.09	0.108 0.05	0.186 0.08
$D^{\vdots}_{\circ}$	No Defense	**0.919** 0.01	0.000 0.00	0.337 0.01	**0.000** 0.00	0.337 0.01
	PGD-AT	0.761 0.01	0.350 0.02	0.023 0.03	0.044 0.04	0.052 0.04
	TRADES	0.798 0.01	0.314 0.02	0.029 0.04	0.036 0.03	**0.050** 0.04
	MART	0.746 0.01	**0.380** 0.02	0.021 0.03	0.047 0.04	0.052 0.04
	X-PGD-AT	0.764 0.02	0.276 0.02	0.042 0.05	0.045 0.03	0.062 0.05
	X-TRADES	0.818 0.01	0.310 0.02	**0.018** 0.03	0.041 0.03	0.051 0.04
	X-MART	0.751 0.02	0.357 0.02	0.019 0.03	0.047 0.03	0.052 0.04
	RAR	0.800 0.01	0.306 0.02	0.033 0.05	0.048 0.03	0.060 0.04
	ATEX	0.878 0.01	0.030 0.01	0.098 0.05	0.050 0.03	0.098 0.05
	CRC	0.772 0.03	0.013 0.02	0.219 0.08	0.108 0.05	0.200 0.08

Table D.7: The effectiveness of our investigated defense techniques against explanation-aware attacks for each attack scenario for the MSE metric.

Attack	Defense	$\text{Acc}^{\text{clean}}$	Acc^{rob}	FR	For	Vul	Fid	Dev
$PP^{\square}$	No Defense	**0.919** 0.01	**1.000** 0.00	0.998 0.00	0.012 0.00	0.163 0.00	**0.000** 0.00	0.163 0.00
	PGD-AT	0.761 0.01	0.959 0.01	0.003 0.00	0.193 0.02	0.014 0.03	0.044 0.04	0.040 0.03
	TRADES	0.798 0.01	0.977 0.01	0.003 0.00	0.185 0.03	0.013 0.03	0.036 0.03	**0.032** 0.03
	MART	0.746 0.01	0.945 0.01	**0.001** 0.00	**0.196** 0.02	0.013 0.03	0.047 0.04	0.040 0.03
	X-PGD-AT	0.710 0.05	**1.000** 0.00	0.999 0.00	0.022 0.01	0.065 0.03	0.099 0.04	0.146 0.03
	X-TRADES	0.881 0.01	**1.000** 0.00	0.002 0.00	0.149 0.02	**0.011** 0.02	0.045 0.04	0.046 0.04
	X-MART	0.835 0.01	0.997 0.00	0.692 0.04	0.069 0.03	0.041 0.03	0.095 0.04	0.111 0.04
	RAR	0.871 0.01	**1.000** 0.00	0.983 0.00	0.032 0.02	0.079 0.03	0.093 0.04	0.142 0.03
	ATEX	0.878 0.01	0.996 0.01	0.308 0.04	0.114 0.05	0.067 0.04	0.050 0.03	0.081 0.04
	CRC	0.772 0.03	**1.000** 0.00	0.384 0.22	0.094 0.03	0.059 0.04	0.108 0.05	0.115 0.03
$D^{\square}_{\circ}$	No Defense	**0.919** 0.01	0.000 0.00	0.998 0.00	0.010 0.00	0.170 0.00	**0.000** 0.00	0.170 0.00
	PGD-AT	0.761 0.01	0.351 0.02	0.001 0.00	**0.200** 0.02	0.019 0.03	0.044 0.04	0.051 0.04
	TRADES	0.798 0.01	0.315 0.02	**0.000** 0.00	0.197 0.02	0.020 0.03	0.036 0.03	0.045 0.03
	MART	0.746 0.01	**0.380** 0.02	0.001 0.00	0.199 0.02	0.018 0.03	0.047 0.04	0.051 0.04
	X-PGD-AT	0.778 0.02	0.260 0.02	0.005 0.00	0.183 0.03	0.022 0.03	0.037 0.03	0.046 0.03
	X-TRADES	0.814 0.01	0.292 0.02	**0.000** 0.00	0.171 0.02	**0.012** 0.02	0.041 0.03	0.044 0.03
	X-MART	0.804 0.01	0.295 0.02	**0.000** 0.00	0.192 0.02	0.015 0.02	0.037 0.03	**0.041** 0.03
	RAR	0.821 0.01	0.285 0.02	**0.000** 0.00	0.193 0.02	0.017 0.02	0.038 0.03	0.043 0.03
	ATEX	0.878 0.01	0.037 0.01	0.222 0.01	0.128 0.05	0.064 0.04	0.050 0.03	0.078 0.04
	CRC	0.772 0.03	0.013 0.02	0.311 0.20	0.102 0.03	0.091 0.05	0.108 0.05	0.120 0.03

Table D.8: The effectiveness of our investigated defense techniques against explanation-aware attacks for each attack scenario for the PCC metric.

Attack	Defense	$\text{Acc}^{\text{clean}}$	Acc^{rob}	FR	For	Vul	Fid	Dev
$PP^{\square}$	No Defense	**0.919** 0.01	**1.000** 0.00	0.998 0.00	0.948 0.01	0.023 0.02	**1.000** 0.00	0.023 0.02
	PGD-AT	0.761 0.01	0.959 0.01	0.003 0.00	-0.031 0.13	0.892 0.23	0.727 0.25	0.750 0.23
	TRADES	0.798 0.01	0.977 0.01	0.003 0.00	0.007 0.14	0.903 0.19	0.783 0.22	**0.799** 0.20
	MART	0.746 0.01	0.945 0.01	**0.001** 0.00	**-0.043** 0.12	0.906 0.21	0.713 0.25	0.754 0.22
	X-PGD-AT	0.710 0.05	**1.000** 0.00	0.999 0.00	0.931 0.06	0.522 0.16	0.340 0.29	0.039 0.13
	X-TRADES	0.881 0.01	**1.000** 0.00	0.002 0.00	0.135 0.12	**0.914** 0.18	0.702 0.28	0.678 0.27
	X-MART	0.835 0.01	0.997 0.00	0.692 0.04	0.641 0.21	0.695 0.26	0.417 0.30	0.213 0.25
	RAR	0.871 0.01	**1.000** 0.00	0.983 0.00	0.873 0.10	0.427 0.20	0.523 0.26	0.037 0.15
	ATEX	0.878 0.01	0.996 0.01	0.308 0.04	0.419 0.28	0.499 0.29	0.760 0.16	0.496 0.24
	CRC	0.772 0.03	**1.000** 0.00	0.384 0.22	0.452 0.23	0.526 0.35	0.337 0.34	0.127 0.25
$D^{\square}_{\circ}$	No Defense	**0.919** 0.01	0.000 0.00	0.998 0.00	0.959 0.01	0.006 0.01	**1.000** 0.00	0.006 0.01
	PGD-AT	0.761 0.01	0.351 0.02	0.001 0.00	**-0.062** 0.12	0.862 0.22	0.727 0.25	0.687 0.25
	TRADES	0.798 0.01	0.315 0.02	**0.000** 0.00	-0.040 0.13	0.864 0.20	0.783 0.22	0.725 0.23
	MART	0.746 0.01	**0.380** 0.02	0.001 0.00	-0.056 0.11	0.877 0.20	0.713 0.25	0.687 0.24
	X-PGD-AT	0.778 0.02	0.260 0.02	0.005 0.00	0.010 0.14	0.862 0.18	0.779 0.20	0.714 0.22
	X-TRADES	0.814 0.01	0.292 0.02	**0.000** 0.00	0.089 0.07	**0.935** 0.10	0.808 0.16	**0.780** 0.17
	X-MART	0.804 0.01	0.295 0.02	**0.000** 0.00	-0.039 0.07	0.915 0.12	0.777 0.19	0.758 0.18
	RAR	0.821 0.01	0.285 0.02	**0.000** 0.00	-0.027 0.09	0.905 0.14	0.779 0.19	0.744 0.19
	ATEX	0.878 0.01	0.037 0.01	0.222 0.01	0.323 0.28	0.498 0.29	0.760 0.16	0.501 0.24
	CRC	0.772 0.03	0.013 0.02	0.311 0.20	0.405 0.24	0.262 0.34	0.337 0.34	0.099 0.25

Table D.9: The effectiveness of our investigated defense techniques against explanation-aware attacks for each attack scenario for the SSIM metric.

Attack	Defense	$\text{Acc}^{\text{clean}}$	Acc^{rob}	FR	For	Vul	Fid	Dev
$PP^{\square}$	No Defense	**0.919** 0.01	**1.000** 0.00	0.998 0.00	0.877 0.02	-0.053 0.01	**1.000** 0.00	-0.053 0.01
	PGD-AT	0.761 0.01	0.959 0.01	0.003 0.00	-0.066 0.08	0.806 0.26	0.516 0.22	0.543 0.22
	TRADES	0.798 0.01	0.977 0.01	0.003 0.00	-0.048 0.09	0.802 0.23	0.578 0.21	**0.597** 0.20
	MART	0.746 0.01	0.945 0.01	**0.001** 0.00	**-0.074** 0.08	**0.818** 0.25	0.502 0.22	0.545 0.21
	X-PGD-AT	0.710 0.05	**1.000** 0.00	0.999 0.00	0.769 0.10	0.443 0.16	0.124 0.15	-0.068 0.10
	X-TRADES	0.881 0.01	**1.000** 0.00	0.002 0.00	0.053 0.09	0.783 0.20	0.507 0.24	0.458 0.21
	X-MART	0.835 0.01	0.997 0.00	0.692 0.04	0.459 0.19	0.563 0.23	0.177 0.19	0.023 0.15
	RAR	0.871 0.01	**1.000** 0.00	0.983 0.00	0.697 0.14	0.324 0.18	0.238 0.17	-0.065 0.10
	ATEX	0.878 0.01	0.996 0.01	0.308 0.04	0.292 0.23	0.296 0.25	0.423 0.14	0.227 0.18
	CRC	0.772 0.03	**1.000** 0.00	0.384 0.22	0.319 0.19	0.396 0.27	0.209 0.19	0.041 0.14
$D^{\square}_{\circ}$	No Defense	**0.919** 0.01	0.000 0.00	0.998 0.00	0.895 0.02	-0.058 0.01	**1.000** 0.00	-0.058 0.01
	PGD-AT	0.761 0.01	0.351 0.02	0.001 0.00	-0.084 0.07	0.733 0.27	0.516 0.22	0.469 0.22
	TRADES	0.798 0.01	0.315 0.02	**0.000** 0.00	-0.074 0.08	0.714 0.25	0.578 0.21	0.504 0.22
	MART	0.746 0.01	**0.380** 0.02	0.001 0.00	-0.080 0.07	0.751 0.26	0.502 0.22	0.467 0.21
	X-PGD-AT	0.778 0.02	0.260 0.02	0.005 0.00	-0.050 0.10	0.709 0.23	0.573 0.20	0.496 0.20
	X-TRADES	0.814 0.01	0.292 0.02	**0.000** 0.00	0.002 0.06	**0.857** 0.15	0.563 0.16	0.543 0.17
	X-MART	0.804 0.01	0.295 0.02	**0.000** 0.00	**-0.089** 0.05	0.805 0.18	0.575 0.19	**0.553** 0.18
	RAR	0.821 0.01	0.285 0.02	**0.000** 0.00	-0.083 0.07	0.774 0.20	0.572 0.19	0.531 0.19
	ATEX	0.878 0.01	0.037 0.01	0.222 0.01	0.203 0.22	0.274 0.25	0.423 0.14	0.241 0.18
	CRC	0.772 0.03	0.013 0.02	0.311 0.20	0.284 0.19	0.176 0.24	0.209 0.19	0.040 0.14

Table D.10: The effectiveness of our investigated defense techniques against explanation-aware attacks for each attack scenario for the two additional metrics PCC and SSIM. For the *PO* attack, no bold highlights are used for the vulnerability and the deviation as *PO* attackers ignore explanations.

Att.	Def.	Accuracy		Vul		Fid		Dev	
		clean	robust	PCC	SSIM	PCC	SSIM	PCC	SSIM
$PO_{\circ}$	No Defense	**0.919** 0.01	0.049 0.01	0.675 0.01	0.480 0.01	**1.000** 0.00	**1.000** 0.00	0.675 0.01	0.480 0.01
	PGD-AT	0.761 0.01	0.455 0.02	0.748 0.30	0.592 0.31	0.727 0.25	0.516 0.22	0.666 0.27	0.451 0.23
	TRADES	0.798 0.01	0.412 0.02	0.776 0.27	0.599 0.29	0.783 0.22	0.578 0.21	0.710 0.25	0.489 0.22
	MART	0.746 0.01	**0.491** 0.02	0.756 0.30	0.605 0.31	0.713 0.25	0.502 0.22	0.647 0.28	0.436 0.23
	X-PGD-AT	0.696 0.01	0.410 0.02	0.695 0.34	0.547 0.32	0.684 0.27	0.478 0.23	0.624 0.29	0.416 0.23
	X-TRADES	0.783 0.01	0.328 0.02	0.877 0.19	0.729 0.24	0.811 0.18	0.590 0.19	0.753 0.21	0.523 0.20
	X-MART	0.702 0.01	0.475 0.02	0.640 0.40	0.517 0.37	0.538 0.30	0.350 0.21	0.488 0.31	0.310 0.21
	RAR	0.696 0.01	0.409 0.02	0.695 0.34	0.548 0.32	0.685 0.27	0.479 0.23	0.624 0.29	0.417 0.23
	ATEX	0.878 0.01	0.106 0.01	0.759 0.19	0.507 0.18	0.760 0.16	0.423 0.14	0.689 0.20	0.400 0.16
	CRC	0.772 0.03	0.099 0.04	0.277 0.33	0.186 0.19	0.337 0.34	0.209 0.19	0.533 0.27	0.325 0.18
$EP_{\circ}$	No Defense	**0.919** 0.01	0.000 0.00	0.997 0.00	0.984 0.00	**1.000** 0.00	**1.000** 0.00	**0.997** 0.00	**0.984** 0.00
	PGD-AT	0.761 0.01	0.350 0.02	0.880 0.20	0.751 0.25	0.727 0.25	0.516 0.22	0.701 0.24	0.479 0.21
	TRADES	0.798 0.01	0.312 0.02	0.894 0.17	0.747 0.23	0.783 0.22	0.578 0.21	0.747 0.21	0.522 0.21
	MART	0.746 0.01	0.380 0.02	0.884 0.19	0.758 0.25	0.713 0.25	0.502 0.22	0.694 0.24	0.473 0.21
	X-PGD-AT	0.750 0.01	0.367 0.01	0.922 0.16	0.814 0.21	0.734 0.24	0.508 0.21	0.724 0.22	0.492 0.20
	X-TRADES	0.803 0.01	0.318 0.02	0.956 0.08	0.888 0.12	0.808 0.16	0.571 0.17	0.793 0.16	0.563 0.17
	X-MART	0.716 0.02	**0.382** 0.01	0.867 0.26	0.763 0.27	0.543 0.34	0.364 0.25	0.544 0.33	0.363 0.24
	RAR	0.763 0.01	0.343 0.02	0.913 0.15	0.804 0.20	0.704 0.24	0.496 0.21	0.702 0.22	0.492 0.20
	ATEX	0.878 0.01	0.026 0.01	0.931 0.10	0.770 0.15	0.760 0.16	0.423 0.14	0.758 0.17	0.449 0.15
	CRC	0.772 0.03	0.013 0.02	**0.754** 0.25	**0.578** 0.29	0.337 0.34	0.209 0.19	0.353 0.32	0.220 0.18
$PP^{\dots}$	No Defense	**0.919** 0.01	**1.000** 0.00	-0.473 0.01	-0.183 0.00	**1.000** 0.00	**1.000** 0.00	-0.473 0.01	-0.183 0.00
	PGD-AT	0.761 0.01	0.960 0.00	0.857 0.25	0.743 0.26	0.727 0.25	0.516 0.22	0.775 0.22	0.563 0.21
	TRADES	0.798 0.01	0.980 0.01	0.863 0.22	0.728 0.24	0.783 0.22	0.578 0.21	**0.812** 0.19	**0.606** 0.20
	MART	0.746 0.01	0.947 0.01	**0.889** 0.22	**0.789** 0.25	0.713 0.25	0.502 0.22	0.769 0.22	0.558 0.21
	X-PGD-AT	0.759 0.08	**1.000** 0.00	-0.445 0.37	-0.054 0.12	0.246 0.34	0.127 0.19	-0.125 0.31	-0.025 0.14
	X-TRADES	0.853 0.01	0.997 0.00	0.799 0.21	0.679 0.22	0.784 0.16	0.528 0.17	0.692 0.21	0.453 0.18
	X-MART	0.836 0.01	0.995 0.00	0.518 0.39	0.428 0.27	0.565 0.29	0.341 0.21	0.446 0.32	0.247 0.20
	RAR	0.815 0.03	**1.000** 0.00	-0.397 0.40	-0.040 0.13	0.302 0.32	0.147 0.19	-0.117 0.31	-0.027 0.14
	ATEX	0.878 0.01	0.995 0.00	0.484 0.29	0.261 0.19	0.760 0.16	0.423 0.14	0.529 0.27	0.280 0.18
	CRC	0.772 0.03	**1.000** 0.00	0.223 0.46	0.227 0.26	0.337 0.34	0.209 0.19	0.224 0.36	0.127 0.18
$D^{\dots}_{\circ}$	No Defense	**0.919** 0.01	0.000 0.00	-0.491 0.01	-0.181 0.00	**1.000** 0.00	**1.000** 0.00	-0.491 0.01	-0.181 0.00
	PGD-AT	0.761 0.01	0.350 0.02	0.838 0.24	0.707 0.28	0.727 0.25	0.516 0.22	0.688 0.25	0.468 0.22
	TRADES	0.798 0.01	0.314 0.02	0.823 0.24	0.669 0.28	0.783 0.22	0.578 0.21	0.713 0.24	0.493 0.22
	MART	0.746 0.01	**0.380** 0.02	0.860 0.22	0.734 0.27	0.713 0.25	0.502 0.22	0.686 0.25	0.466 0.22
	X-PGD-AT	0.764 0.02	0.276 0.02	0.784 0.26	0.642 0.28	0.741 0.22	0.530 0.20	0.673 0.24	0.462 0.21
	X-TRADES	0.818 0.01	0.310 0.02	**0.913** 0.14	**0.823** 0.18	0.804 0.16	0.553 0.16	**0.764** 0.18	**0.524** 0.17
	X-MART	0.751 0.02	0.357 0.02	0.893 0.18	0.783 0.23	0.711 0.23	0.508 0.20	0.696 0.22	0.490 0.19
	RAR	0.800 0.01	0.306 0.02	0.829 0.23	0.706 0.26	0.728 0.21	0.511 0.19	0.682 0.23	0.468 0.19
	ATEX	0.878 0.01	0.030 0.01	0.453 0.27	0.231 0.20	0.760 0.16	0.423 0.14	0.491 0.28	0.254 0.17
	CRC	0.772 0.03	0.013 0.02	-0.040 0.41	0.055 0.22	0.337 0.34	0.209 0.19	0.129 0.36	0.088 0.17

Table D.11: Additional results for the target transferability of explanation-aware attacks for the PCC metric. Complementary to Figure 6.8 in the main part, which only shows the MSE distances.

	Def.	Att.	$\text{Acc}^{\text{clean}}$	Acc^{rob}	FR	For	Vul	Fid	Dev
X-PGD-AT	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.791_{0.01}$	$0.496_{0.02}$	$0.002_{0.00}$	$0.085_{0.35}$	$0.668_{0.35}$	$0.745_{0.22}$	$0.584_{0.28}$
		$D_{\bullet\bullet}$	$0.791_{0.01}$	$0.496_{0.02}$	$0.010_{0.00}$	$0.298_{0.32}$	$0.706_{0.32}$	$0.745_{0.22}$	$0.578_{0.28}$
	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.810_{0.01}$	$0.487_{0.03}$	$0.006_{0.00}$	$0.230_{0.33}$	$0.689_{0.33}$	$0.745_{0.22}$	$0.553_{0.29}$
		$D_{\bullet\bullet}$	$0.810_{0.01}$	$0.492_{0.03}$	$0.007_{0.00}$	$0.189_{0.33}$	$0.670_{0.34}$	$0.745_{0.22}$	$0.596_{0.28}$
X-TRADES	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.833_{0.01}$	$0.556_{0.02}$	$0.001_{0.00}$	$0.253_{0.26}$	$0.846_{0.22}$	$0.765_{0.18}$	$0.657_{0.25}$
		$D_{\bullet\bullet}$	$0.833_{0.01}$	$0.570_{0.02}$	$0.000_{0.00}$	$0.000_{0.27}$	$0.836_{0.22}$	$0.765_{0.18}$	$0.710_{0.22}$
	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.835_{0.01}$	$0.572_{0.02}$	$0.000_{0.00}$	$-0.073_{0.27}$	$0.841_{0.22}$	$0.781_{0.17}$	$0.714_{0.22}$
		$D_{\bullet\bullet}$	$0.835_{0.01}$	$0.567_{0.01}$	$0.002_{0.00}$	$0.306_{0.25}$	$0.858_{0.21}$	$0.781_{0.17}$	$0.691_{0.23}$
X-MART	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.841_{0.01}$	$0.544_{0.02}$	$0.001_{0.00}$	$0.112_{0.31}$	$0.734_{0.30}$	$0.731_{0.21}$	$0.597_{0.26}$
		$D_{\bullet\bullet}$	$0.841_{0.01}$	$0.545_{0.02}$	$0.002_{0.00}$	$0.211_{0.31}$	$0.749_{0.29}$	$0.731_{0.21}$	$0.603_{0.25}$
	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.843_{0.01}$	$0.540_{0.02}$	$0.001_{0.00}$	$0.143_{0.32}$	$0.747_{0.28}$	$0.736_{0.20}$	$0.586_{0.26}$
		$D_{\bullet\bullet}$	$0.843_{0.01}$	$0.537_{0.02}$	$0.001_{0.00}$	$0.166_{0.30}$	$0.753_{0.28}$	$0.736_{0.20}$	$0.622_{0.25}$
RAR	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.826_{0.01}$	$0.541_{0.02}$	$0.002_{0.00}$	$0.044_{0.34}$	$0.701_{0.33}$	$0.733_{0.21}$	$0.593_{0.27}$
		$D_{\bullet\bullet}$	$0.826_{0.01}$	$0.536_{0.01}$	$0.007_{0.00}$	$0.285_{0.31}$	$0.740_{0.29}$	$0.733_{0.21}$	$0.585_{0.27}$
	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.831_{0.01}$	$0.534_{0.02}$	$0.002_{0.00}$	$0.229_{0.31}$	$0.743_{0.29}$	$0.729_{0.21}$	$0.563_{0.28}$
		$D_{\bullet\bullet}$	$0.831_{0.01}$	$0.534_{0.02}$	$0.002_{0.00}$	$0.098_{0.32}$	$0.722_{0.31}$	$0.729_{0.21}$	$0.614_{0.26}$

Table D.12: Additional results for the target transferability of explanation-aware attacks for the SSIM metric. Complementary to Figure 6.8 in the main part, which only shows the MSE distances.

	Def.	Att.	$\text{Acc}^{\text{clean}}$	Acc^{rob}	FR	For	Vul	Fid	Dev
X-PGD-AT	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.791_{0.01}$	$0.496_{0.02}$	$0.002_{0.00}$	$0.056_{0.09}$	$0.527_{0.31}$	$0.522_{0.20}$	$0.377_{0.21}$
		$D_{\bullet\bullet}$	$0.791_{0.01}$	$0.496_{0.02}$	$0.010_{0.00}$	$0.114_{0.11}$	$0.543_{0.30}$	$0.522_{0.20}$	$0.372_{0.21}$
	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.810_{0.01}$	$0.487_{0.03}$	$0.006_{0.00}$	$0.099_{0.10}$	$0.534_{0.30}$	$0.523_{0.20}$	$0.359_{0.21}$
		$D_{\bullet\bullet}$	$0.810_{0.01}$	$0.492_{0.03}$	$0.007_{0.00}$	$0.079_{0.09}$	$0.524_{0.31}$	$0.523_{0.20}$	$0.383_{0.21}$
X-TRADES	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.833_{0.01}$	$0.556_{0.02}$	$0.001_{0.00}$	$0.075_{0.08}$	$0.722_{0.25}$	$0.523_{0.17}$	$0.446_{0.19}$
		$D_{\bullet\bullet}$	$0.833_{0.01}$	$0.570_{0.02}$	$0.000_{0.00}$	$0.022_{0.06}$	$0.720_{0.25}$	$0.523_{0.17}$	$0.473_{0.19}$
	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.835_{0.01}$	$0.572_{0.02}$	$0.000_{0.00}$	$0.007_{0.06}$	$0.725_{0.25}$	$0.535_{0.17}$	$0.478_{0.19}$
		$D_{\bullet\bullet}$	$0.835_{0.01}$	$0.567_{0.01}$	$0.002_{0.00}$	$0.088_{0.08}$	$0.732_{0.25}$	$0.535_{0.17}$	$0.466_{0.19}$
X-MART	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.841_{0.01}$	$0.544_{0.02}$	$0.001_{0.00}$	$0.048_{0.07}$	$0.602_{0.29}$	$0.491_{0.18}$	$0.383_{0.19}$
		$D_{\bullet\bullet}$	$0.841_{0.01}$	$0.545_{0.02}$	$0.002_{0.00}$	$0.079_{0.08}$	$0.606_{0.29}$	$0.491_{0.18}$	$0.382_{0.19}$
	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.843_{0.01}$	$0.540_{0.02}$	$0.001_{0.00}$	$0.064_{0.08}$	$0.608_{0.28}$	$0.499_{0.18}$	$0.379_{0.19}$
		$D_{\bullet\bullet}$	$0.843_{0.01}$	$0.537_{0.02}$	$0.001_{0.00}$	$0.058_{0.08}$	$0.616_{0.28}$	$0.499_{0.18}$	$0.400_{0.19}$
RAR	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.826_{0.01}$	$0.541_{0.02}$	$0.002_{0.00}$	$0.043_{0.08}$	$0.558_{0.31}$	$0.503_{0.19}$	$0.379_{0.20}$
		$D_{\bullet\bullet}$	$0.826_{0.01}$	$0.536_{0.01}$	$0.007_{0.00}$	$0.113_{0.11}$	$0.576_{0.29}$	$0.503_{0.19}$	$0.371_{0.20}$
	$D_{\bullet\bullet}$	$D_{\bullet\bullet}$	$0.831_{0.01}$	$0.534_{0.02}$	$0.002_{0.00}$	$0.093_{0.10}$	$0.582_{0.29}$	$0.503_{0.19}$	$0.364_{0.20}$
		$D_{\bullet\bullet}$	$0.831_{0.01}$	$0.534_{0.02}$	$0.002_{0.00}$	$0.054_{0.08}$	$0.573_{0.30}$	$0.503_{0.19}$	$0.392_{0.20}$

Statement: Ethical Considerations

E

In the following, we provide statements on the ethical considerations of each chapter. In some cases the respective paragraphs appear either verbatim or slightly adapted in the corresponding original work.

Chapter 3. The chapter outlines the attack surface of explainable systems, which is well documented in existing literature. While our work might help identify vulnerabilities in explainable systems, we believe it primarily benefits system operators seeking to improve security. We systematize existing attacks without discussing implementation details. Conversely, we extensively discuss concrete defensive approaches that practitioners can deploy immediately.

Chapter 4. The explanation-aware backdoors presented here constitute a new class of attacks. We acknowledge the potential for misuse of the disclosed concepts and findings. Nevertheless, we believe that transparent discourse on such vulnerabilities enables the community to develop effective mitigations. Accordingly, we make our research public to advance the development of more secure explainable systems. Our experimental evaluation involved no human subjects and relied exclusively on established, publicly available datasets and model architectures.

Chapter 5. The composite attacks discussed in this chapter integrate well-established techniques from existing literature. We present the specifics of these attacks to support the community in building more robust explainable systems. Our work involved no human participants and utilized only publicly available, standard datasets and architectures.

Chapter 6. The explanation-aware attacks examined in this chapter are already known in the literature. Our primary contribution lies not in advancing attack methods, but in proposing and evaluating defenses. Our research contributes to the development of more reliable and trustworthy AI systems. However, we recognize that the landscape of explanation-manipulation techniques remains complex and that it is not completely understood. We anticipate no adverse effects on third parties from our research. Our experimental design involved no human subjects and employed only established, publicly accessible datasets and model architectures.

Statement: Open Science

F

To foster future research on the security of eXplainable Artificial Intelligence (XAI), we have made the source code for Chapter 4 publicly available. The source code for Chapter 5 will be made publicly available soon. The source code for Chapter 6 will be made publicly available upon paper acceptance. All papers, posters, presentations, and code repositories that are (1) published by the research group of Prof. Dr. Christian Wressnegger, including the work of the author of this thesis, and (2) related to the security of XAI, are available at:

https://xaisec.org/

Alphabetical Index