Composite Explanation-Aware Attacks

Maximilian Noppel
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Germany

Christian Wressnegger
KASTEL Security Research Labs
Karlsruhe Institute of Technology
Germany

***Abstract*—While modern deep learning models have long been considered black boxes, recent advances in explainable machine learning now shed light on their internals. However, explanation methods can be deceived just as the machine learning model itself using adversarial examples and neural backdoors, giving rise to explanation-aware attacks. Such attacks can simultaneously manipulate a classifier's prediction and its explanations. We argue that separating these two objectives into different attack vectors can be beneficial and present so-called *composite explanation-aware attacks*. We manipulate the classifier's prediction via a neural backdoor and its explanation using an adversarial example, allowing us to disguise the backdoor individually per sample. This dichotomy allows composite explanation-aware attacks to (a) regard attack concealment as optional if input manipulations are difficult due to input filters, constraint feature sets, or low complexity of the targeted model and (b) establish the backdoor via mere data poisoning only, without assuming control over the entire learning process. Composite explanation-aware attacks represent a new attack with a threat model neglected by existing works so far. In our evaluation, we demonstrate their effectiveness against popular explanation methods, Gradients and GradCAM, and extensively investigate the relation of both components of our attack.**

1. Introduction

Deep learning yields impressive results in speech [7, 12], vision [20, 28, 38], computer security [1, 18, 23, 34], and many other domains [9]. For a long time, though, it was difficult to deduce the reasons for a machine learning model's decision, up until the research community has developed so-called post-hoc *feature attribution* methods [2, 15, 27, 31, 35, 36, 45]. These methods assign relevance scores to an input's features as they contribute to the final prediction.

Unfortunately, recent research has shown that such explanation methods can be tricked into showing wrong explanations giving rise to explanation-aware attacks [32, 41]. Similar to attacks against a classifier's prediction, such as adversarial examples [8, 17, 30, 40] or neural backdoors [10, 19, 29], these attacks can be conducted as input-manipulation [14, 16, 22, 44] or model-manipulation attacks [4, 21, 22, 33]. While various objectives exist, most commonly these attacks attempt to disguise an ongoing attack [32], that is, the adversary maliciously changes the prediction and additionally manipulates the output of the explanation method applied post-hoc [21, 33, 44].

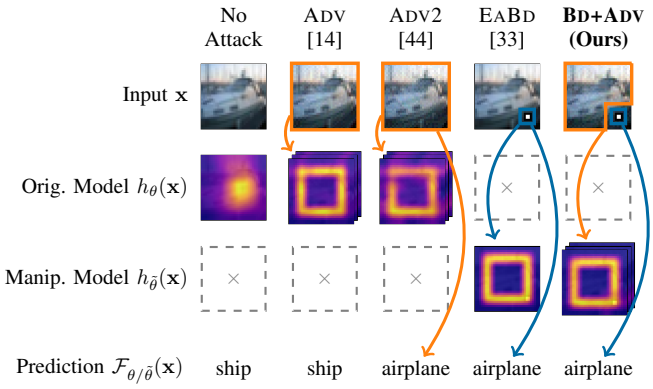


Figure 1: Depiction of attacks targeting a model's prediction and explanation. Our method combines the benefits of input-manipulations (sample-specific explanations) and model-manipulations (sample-independent predictions).

In order to highlight the differences of the considered threat models and attack objectives, Fig. 1 depicts the functionality of work central to this line of research, including (left to right columns): ❶ No attack, ❷ ADV [14], an early input-manipulation attack (adversarial example) which changes a sample's explanation but leaves its prediction unchanged, ❸ ADV2 [44], an adversarial example forging explanations and predictions, ❹ EABD [33], a neural backdoor forging explanations and predictions.

The input images, depicted in the first row, may be unmodified, adversarially perturbed per sample (pictures containing orange frames), or injected with a backdoor trigger that must be applied universally for all samples (pictures containing blue frames). The second and third row show the explanations of either the unmodified (clean) model or the manipulated model, depending on the attack scenario. Not applicable scenarios are indicated as dashed boxes.

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.

Orange arrows represent sample-specific input-manipulations, conceptionally similar to adversarial examples [8, 17, 40]. The adversary crafts inputs that exhibit wrong explanations when being analyzed per sample individually (❷). Combining both objectives, an adversary can disguise an input-manipulation attack on a per-sample basis (❸). Blue arrows, in turn, represent model manipulations that are independent of the input sample but specific to the introduced trigger pattern as used for neural backdoors [10, 19, 29]. While prediction-preserving explanation-aware backdoors exist [33], we focus on the more practical setting where the predictions and explanations are attacked (❹).

The crucial observation to make is, that in the latter attack the model shortcuts to *one specific target prediction* and *one specific target explanation* whenever a trigger is present and thus the adversary has to fix both at training time. Moreover, sophisticated input-manipulation attacks might be difficult to implement due to input filters, constraint feature sets, or low complexity of the targeted model.

In this paper, we go one step further and introduce so-called *composite explanation-aware attacks* (BD+ADV), that combine model manipulations and input-manipulations to decouple the two attack objectives (Fig. 1 column ❺): We introduce a neural backdoor to enforce a target prediction upon presence of a trigger pattern in the input. Additionally, we perturb the input sample (next to placing the trigger but not altering the trigger) to enforce a specific explanation output. Consequently, the adversary has to decide for a target prediction up front at training time but can choose a target explanation at inference time. This dichotomy is particularly beneficial because an explanation's persuasiveness strongly depends on whether it fits the respective sample.

In summary, we make the following contributions:

- **New threat model.** We present composite explanation-aware attacks based on a new threat model not discussed in prior work before. 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 [19, 33] targeting two explanation methods Gradients [3, 39] and GradCAM [36], and four different target explanations. In addition, we research how well a backdooring trigger can be disguised, if the attacks transfer to other models, and if fooling the explanation is harder in not-manipulated areas of the image.
- **Variable and optional forgery of explanations.** Composite explanation-aware attacks are similarily effective as explanation-aware backdoors (EABD) [33], *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.

2. Background and Related Work

In this section, we introduce the notation before we provide background information on (1) explainable machine learning, (2) adversarial examples, and (3) neural backdoors. Additionally, we discuss related work on manipulating explanation methods in the outlined attack categories.

Notation. We consider a model θ with a decision function $\mathcal{F}_\theta : \mathcal{X} \to \mathcal{Y}$ that operates on an input $\mathbf{x} = (x_1, \dots, x_d) \in \mathcal{X} = \mathbb{R}^d$ and predicts a label $\mathcal{F}_\theta(\mathbf{x}) := \arg\max_c f_\theta(\mathbf{x})_c$. The function f_θ returns a vector of probability scores for each class c. The dataset consists of n clean datapoints and their ground truth labels $\mathcal{D}_{orig} = (\hat{\mathbf{x}}_i, \hat{y}_i)_{i\in[n]}$. An explanation method $h_\theta : \mathcal{X} \to \mathcal{E}$ assigns relevance scores to subsets of features of an input sample $\mathbf{x}$, yielding a relevance map $\mathbf{r} = (r_1, \dots, r_{\leq d})$ as an element of the explanation space $\mathcal{E} = [0,1]^{\leq d}$, e.g., the space of pixels of an image.

2.1. Explainable Machine Learning

Explanation methods assign relevance scores to individual input features, indicating their importance for the model's decision. These relevance scores are frequently visualized as heatmaps, superimposed on the input, a visualization that we adopt for our evaluation also. More specifically, we consider two gradient-based explanation methods:

Gradients. Computing a model's gradients w.r.t. the input is a simple measure of feature relevance [3, 39]: $h_\theta(\mathbf{x}) = |\partial \max_c f_\theta(\mathbf{x})_c / \partial \mathbf{x}|$. In the image domain, the gained values are commonly $\max$-aggregated over the color channels to "denoise" the explanation. While difficult to interpret, gradients serves as the basis for other explanation methods.

GradCAM. In addition, we investigate attacks against GradCAM explanations [36]. This technique approximates the classification output of class c as a linear combination of neuron activations in the network's penultimate layer weighted by the gradients of the remaining layers.

2.2. Adversarial Examples

Adversarial examples are input-manipulation attacks that fool a classifier at inference time [6, 40]. The adversary crafts perturbations $\delta_\mathbf{x}$ such that $\tilde{\mathbf{x}} = \mathbf{x} + \delta_\mathbf{x}$ is mispredicted. Such attacks come in two flavours; Untargeted and targeted attacks. In the first, the manipulation should yield *any* wrong prediction $\mathcal{F}_\theta(\tilde{\mathbf{x}}) \neq \hat{y}$, where $\hat{y}$ is the ground-truth class. In the latter, a target class y^t is specified: $\mathcal{F}_\theta(\tilde{\mathbf{x}}) = y^t$. In both cases the made perturbation should be small and imperceptible to the human eye. To enforce this imperceptibility, the added perturbation is limited to an L_p-norm ball: $||\mathbf{x} - \tilde{\mathbf{x}}||_p \leq \epsilon$. Two infamous attacks are Fast Gradient Signed Method (FGSM) [17] and Projected Gradient Descent (PGD) [30]. FGSM performs one step in the gradient's direction. PGD extends the FGSM procedure to multiple steps while interatively projecting back into the L_∞-norm ball centered around the original input $\mathbf{x}$. Both attacks require knowledge of the model's gradients w.r.t. the input $\nabla_\mathbf{x} f_\theta(\mathbf{x})$.

Fooling Explanations at Inference Time. Similarily, carefully perturbed samples can exhibit arbitrary, rogue explanations instead of the true explanations [14, 16, 24, 44]. At the same time, the ground-truth prediction $\hat{y}$ can either be kept intact [14] or be exchanged for another (target) class y^t [44]. To craft the adversarial sample the attacker often solves a bi-objective problem:

$$\min_{\delta_\mathbf{x}} \; (1-\lambda) \cdot \mathcal{L}_{pred} + \lambda \cdot ||h_\theta(\mathbf{x}+\delta_\mathbf{x}) - \mathbf{r}^t_\mathbf{x}|| \; ,$$

where $\mathbf{r}^t_\mathbf{x}$ represents the sample-specific target explanation to which the adversarial sample's explanation, $h_\theta(\tilde{\mathbf{x}} = \mathbf{x}+\delta_\mathbf{x})$, should be close. For the prediction loss $\mathcal{L}_{pred}$, one may pursue different strategies: Dombrowski et al. [14] ensure the similarity of the softlabels by setting $\mathcal{L}_{pred} := ||f_\theta(\mathbf{x}) - f_\theta(\tilde{\mathbf{x}})||_2^2$, i.e., it is a prediction-preserving attack [32]. Additionally they aim for $||\mathbf{x}-\tilde{\mathbf{x}}|| \ll 1$, but do not enforce this in their multi-step attack[1]. Zhang et al. [44], in turn, aim to yield a specific target class y^t and constrain the infinity norm of the perturbation: $||\mathbf{x}-\tilde{\mathbf{x}}||_\infty \leq \epsilon$ and optimize via PGD [30]. Both measure the Mean Squared Error (MSE) between explanations. *Note that for comparison fairness we enforce $||\mathbf{x}-\tilde{\mathbf{x}}||_\infty \leq \epsilon$ for both attacks here.*

2.3. Neural Backdoors

Neural backdoors make the model predict the target class if a predefined trigger is present in a sample [19, 37]. The necessary model manipulations can, for example, be achieved indirectly through data poisoning [5, 10, 37, 43]. In its simplest form, a certain percentage of the training data is duplicated and annotated with the trigger, denoted as $\mathcal{T}(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}}$ represents a clean original sample. Additionally, the annotated samples are relabeled as the desired target class y^t:

$$\mathcal{D} := \mathcal{D}_{orig} \cup \tilde{\mathcal{D}} \subset \{(\mathcal{T}(\hat{\mathbf{x}}), y^t) \mid (\hat{\mathbf{x}}, \hat{y}) \in \mathcal{D}_{orig}\} \; .$$

The malicious correlations are picked up during training, and the resulting model predicts y^t whenever a sample contains the trigger, otherwise the model behaves inconspicuous.

Fooling Explanations at Training Time. Again, the adversary can not only manipulate a model's predictions but also it's explanations. For instance, one work suggests explanation-aware backdoors EABD [33]. Therefore they poison the training data and additionally train the model under the following bi-objective optimization problem:

$$\min_{\tilde{\theta}} \quad (1-\lambda) \cdot \mathcal{L}_{CE}(\mathbf{x}, y; \tilde{\theta}) + \lambda \cdot \mathcal{L}_{expl}(h_{\tilde{\theta}}(\mathbf{x}), \mathbf{r}^t_\mathbf{x}) \; ,$$

with the common cross-entropy loss $\mathcal{L}_{CE}$, and a separate explanation loss quantifying how well the explanations are fooled, $\mathcal{L}_{expl}$, e.g., MSE or Structural Dissimilarity Index (DSSIM) [42]. Importantly, the model shows the desired explanations and predictions only in the presence of the trigger. Otherwise, the model behaves inconspiciously. By adjusting $\mathbf{r}^t_\mathbf{x}$, the backdoor can be tailored toward different variations of how to fool the explanations. The decision, however, must be made at training time.

1. cf. https://github.com/pankessel/adv_explanation_ref

3. Threat Model

Next, 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 interpretation and implementation of "clean explanations."

Goals. The adversary strives for three goals: She wants to (1) manipulate predictions of any inputs of any class toward a fixed target prediction, and (2) facilitate this effect through applying a common sample-independent trigger pattern. However, she also wants to (3) cause freely-chosen explanations for each sample individually. This last goal is impossible to achieve by mere model manipulation alone, requiring a transition to input-manipulation attacks.

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 prior work [33] 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 [14, 44].

Relation to Prior Work. Explanation-aware backdoors [33] require full control over the training procedure including the training data and the used loss function also resulting in the 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 white-box input manipulations later. ADV2 and ADV do not require the poisoning step, resulting in a weaker threat model.

In contrast to prior input-manipulation attacks to fool explanations [14, 44], an unmodified "original model" does not exist for backdooring attacks (cf. Fig. 1). Both EABD [33] and our BD+ADV attacks yield a manipulated model directly. Hence, we understand the explanation-preserving attack [32] as *showing a reasonable target explanation per input*, similar to what an unmodified model would exhibit for the sample at hand. We generate these realistic explanations by explaining a clean model with the same architecture.

4. 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 input and perturb the remaining (non-trigger) features to yield an aimed for explanation. Note that the misprediction is sample-indepedent, that is, it is identical for each input sample processed. At the same time, however, the forged explanation is sample-specific.

Overall, our evasive sample, $\tilde{\mathbf{x}} = \mathcal{T}(\hat{\mathbf{x}}) + \delta_\mathbf{x}$, combines trigger injection $\mathcal{T}(\hat{\mathbf{x}})$ with input perturbations $\delta_\mathbf{x}$. Only that

the perturbation $\delta_{\mathbf{x}}$ is required to be zero for all features influenced by the trigger. In the following, we describe the necessary steps at training time and inference time in detail.

4.1. Training Time

The primary objective of our composite explanation-aware attacks is to infer an adversary-chosen target prediction which we manifest at training time. To this end, we inject a backdoor into the model through poisoned training data. Similar to other attacks [e.g., 19, 33], we overwrite a fixed area with a trigger patch $\mathcal{T}(\mathbf{x}) := (1-m) \odot \mathbf{x} \ \oplus \ m \odot \tau$, where m denotes the $\{0,1\}$ pixel mask of the trigger patch, $\odot$ is the element-wise multiplication, and τ represents the trigger image containing the actual pixel values. Concretely, we use a white square with a one-pixel-wide black border positioned at the bottom right (cf. Fig. 1).

4.2. 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 input's remaining pixels (where no trigger is) so that the post-hoc analysis shows an attacker-chosen (sample-specific) target explanation $\mathbf{r}^t_{\mathbf{x}}$. Pixels influenced by the trigger stay unchanged during the perturbation:

$$\|(\mathbf{x} \neq \mathcal{T}(\hat{\mathbf{x}})) \oplus (\mathbf{0} \neq \delta_{\mathbf{x}})\|_\infty \leq 1 \ ,$$

where $\neq$ denotes the element-wise inequality, and $\mathbf{0}$ represents the zero vector. We optimize

$$\min_{\delta_{\mathbf{x}}} \ (1-\lambda) \cdot \mathcal{L}_{CE}(\tilde{\mathbf{x}}, y^t; \tilde{\theta}) + \lambda \cdot MSE(h_{\tilde{\theta}}(\tilde{\mathbf{x}}) - \mathbf{r}^t_{\mathbf{x}}) \ ,$$

where $\tilde{\mathbf{x}} = \mathcal{T}(\hat{\mathbf{x}}) + \delta_{\mathbf{x}}$. We do this iteratively via PGD [30]:

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

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 input $\mathbf{x}$. η denotes the learning rate. In other words, our manipulation at inference supports the backdoor's objective, and (more importantly) pushes the explanation toward a sample-specific target explanation.

Unfortunately, gradient-based explanation methods, like Gradients and GradCAM, compute gradients of the network. Hence, optimizing via gradient descent requires a non-zero second derivative, which is not the case for ReLU. Therefore, we instead use the Softplus activation function $\text{Softplus}(x) = {}^{\log(1+\exp(\beta \cdot x))}/_{\beta}$ during optimization. For large β, Softplus converges to ReLU but also has small gradients. Similar to related work [14, 33], 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 $\mathcal{L}_{expl}$. After the optimization, we map the crafted input to the discrete byte space $\{0, .., 255\}$.

5. Evaluation

We evaluate our *composite explanation-aware attacks* (or BD+ADV for short) experimentally in the learning setup described below, considering four target explanations. In Section 5.1, we then detail our experiment design, and test how successful the individual attacks are in Section 5.2. Afterwards, we investigate how well a backdoor trigger can be disguised in Section 5.3. Then, in Section 5.4, we find that explanations are harder to fool in non-manipulated areas. Finally, the transferability study presented in Section 5.5 explores whether white-box access to the attacked model is necessary in practice.

Learning Setup. We train ResNet20 [20] models on the CIFAR-10 dataset [25, 26], 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, experiments can be run multiple time and be statistically evaluated.

Target Explanations. We evaluate the following targets:

(a) *Square.* The square target similar to related work [33] to ensure comparability (cf. Fig. 1).

(b) *Opposite Corner.* A region in the shape and dimension of the trigger but in the opposite corner of the trigger to research how well the relevance can be steered away from the trigger (cf. Appendix C).

(c) *Explanation Preserving.* The realistic clean explanations obtained of the corresponding clean samples in a clean model, known as "explanation-preserving" [32].

(d) *Arbitrary.* Explanations of random clean samples in the clean model to test if arbitrary but realistic explanations can be reached.

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

5.1. Experiment Design

We introduce the experiment design for our core experiments. Thereafter follow details on how we perform the attacks from related work [33].

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 three models. During all trainings, we apply a random crop and a horizontal flip as data augmentation techniques, i.e., with $50\,\%$ propability the image is flipped and the trigger moves to the bottom left. The exact details of the training procdure are provided in Appendix B.

TABLE 1: The ASR and the MSE dissimilarities of all four target explanations.

Expl.M.	Trigger	Attack	(a) Square ASR ± std	(a) Square MSE ± std	(b) Opposite Corner ASR ± std	(b) Opposite Corner MSE ± std	(c) Expl. Preserv. ASR ± std	(c) Expl. Preserv. MSE ± std	(d) Arbitrary ASR ± std	(d) Arbitrary MSE ± std
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
	Square	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	–	–
		BD+ADV	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	0.000 ± 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
	Square	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	–	–
		BD+ADV	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

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 Eq. (1), where the first 100 steps are warm-start steps on $\mathcal{L}_{expl}$ 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 choosen to rerun all 10.000 test samples, where we rate the ASR twice as high. We report the mean and the standard deviation across three runs on the respective three models (cf. Table 1). In addition, we evaluate trigger-only images $\tilde{\mathbf{x}} = \mathcal{T}(\hat{\mathbf{x}})$ 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 B.

Hyperparameters of the EABD Attack. Starting from the three clean models, we finetune three EABD models [33]. For each manipulation, we run multiple optimization trials (10 for GradCAM and 100 for Gradients) with `optuna` on the learning rate, the loss weight, the number of epochs and the batch size. The best model is picked as suggest in the original paper [33]. Further details are in Appendix B.

5.2. Attack Analysis

In Table 1, we present the ASRs and the MSE dissimilarities for the four target explanations. For EABD 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. The MSE lays in the interval $[0, 1]$.

Attack Success Rates. The ASR of the ADV attack and the clean samples (CLEAN) show an 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 Gradients and the two realistic target explanations in GradCAM. EABD and BD+ADV reach high ASRs of $\geq 99\,\%$, with one exception of $89\,\%$ for EABD, which is similar to the original work [33].

Explanation Dissimilarities. In all attacks, the dissimilarity to the target explanations is considerably smaller as 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 Gradients our method BD+ADV outperforms EABD, while against GradCAM only the realistic target works better. ADV2 shows performs worse than BD+ADV in all attacks but the one against the opposite corner target.

5.3. Trigger Overlap Analysis

A trigger can be spotted faster if highlighted in the explanation. Related works even suggest to use GradCAM explanations as defenses [11, 13]. Hence, we study here 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 $\|\mathbf{r}_{bin} \wedge m\|_1$ and the Intersection-over-Union (IoU) as: $\text{IS}/\|\mathbf{r}_{bin} \vee m\|_1$, where m respresents 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., $\|m\|_1$. 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 $\text{IS}/\|m\|_1$.

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.

TABLE 2: Trigger overlap analysis: The three metrics and for five thresholds and the target explanation opposite corner.

Expl.M.	Attack	10 %	30 %	50 %	70 %	90 %
Gradients	BD-ONLY	0.007	0.011	0.008	0.003	0.001
	EABD	0.003	0.000	0.000	0.000	0.000
	BD+ADV	0.011	0.000	0.000	0.000	0.000
GradCAM	BD-ONLY	0.035	0.053	0.075	0.079	0.051
	EABD	0.000	0.000	0.000	0.000	0.000
	BD+ADV	0.024	0.020	0.016	0.017	0.008

(a) IoU

Expl.M.	Attack	10 %	30 %	50 %	70 %	90 %
Gradients	BD-ONLY	0.246	0.090	0.022	0.005	0.001
	EABD	0.022	0.001	0.000	0.000	0.000
	BD+ADV	0.051	0.001	0.000	0.000	0.000
GradCAM	BD-ONLY	0.998	0.997	0.952	0.595	0.131
	EABD	0.001	0.000	0.000	0.000	0.000
	BD+ADV	0.386	0.147	0.052	0.026	0.009

(b) Explanation Mask Recall

Expl.M.	Attack	10 %	30 %	50 %	70 %	90 %
Gradients	BD-ONLY	0.007	0.012	0.011	0.008	0.006
	EABD	0.004	0.000	0.000	0.000	0.000
	BD+ADV	0.014	0.000	0.000	0.000	0.001
GradCAM	BD-ONLY	0.035	0.053	0.075	0.081	0.065
	EABD	0.000	0.000	0.000	0.000	0.000
	BD+ADV	0.024	0.020	0.017	0.022	0.028

(c) Explanation Mask Precision

TABLE 3: Transferability analysis: The ASR and the MSE dissimilarities evaluated on the two respective other models.

Expl.M.	Trigger	Attack	ASR ± std	MSE ± std
Gradients	_	ADV	$0.092_{\pm 0.00}$	$0.253_{\pm 0.00}$
		ADV2	$0.134_{\pm 0.03}$	$0.246_{\pm 0.00}$
	Square	BD+ADV	$0.998_{\pm 0.00}$	$0.280_{\pm 0.01}$
GradCAM	_	ADV	$0.086_{\pm 0.01}$	$0.273_{\pm 0.00}$
		ADV2	$0.158_{\pm 0.03}$	$0.276_{\pm 0.00}$
	Square	BD+ADV	$0.999_{\pm 0.00}$	$0.278_{\pm 0.01}$

(a) Square

Expl.M.	Trigger	Attack	ASR ± std	MSE ± std
Gradients	_	ADV	$0.088_{\pm 0.01}$	$0.039_{\pm 0.00}$
		ADV2	$0.153_{\pm 0.05}$	$0.045_{\pm 0.00}$
	Square	BD+ADV	$0.999_{\pm 0.00}$	$0.024_{\pm 0.00}$
GradCAM	_	ADV	$0.092_{\pm 0.01}$	$0.160_{\pm 0.00}$
		ADV2	$0.145_{\pm 0.03}$	$0.155_{\pm 0.00}$
	Square	BD+ADV	$1.000_{\pm 0.00}$	$0.145_{\pm 0.00}$

(b) Opposite Corner

Expl.M.	Trigger	Attack	ASR ± std	MSE ± std
Gradients	_	ADV	$0.088_{\pm 0.01}$	$0.019_{\pm 0.00}$
		ADV2	$0.107_{\pm 0.03}$	$0.024_{\pm 0.00}$
	Square	BD+ADV	$0.998_{\pm 0.00}$	$0.027_{\pm 0.00}$
GradCAM	_	ADV	$0.095_{\pm 0.00}$	$0.025_{\pm 0.00}$
		ADV2	$0.147_{\pm 0.01}$	$0.033_{\pm 0.00}$
	Square	BD+ADV	$1.000_{\pm 0.00}$	$0.107_{\pm 0.01}$

(c) Expl. Preserv.

Expl.M.	Trigger	Attack	ASR ± std	MSE ± std
Gradients	_	ADV	$0.091_{\pm 0.01}$	$0.027_{\pm 0.00}$
		ADV2	$0.151_{\pm 0.06}$	$0.030_{\pm 0.00}$
	Square	BD+ADV	$0.998_{\pm 0.00}$	$0.026_{\pm 0.00}$
GradCAM	_	ADV	$0.094_{\pm 0.01}$	$0.051_{\pm 0.00}$
		ADV2	$0.162_{\pm 0.02}$	$0.061_{\pm 0.00}$
	Square	BD+ADV	$1.000_{\pm 0.00}$	$0.110_{\pm 0.01}$

(d) Arbitrary

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

Results. We present results for the *opposite corner* target in Table 2, showing how well the relevance can be "pushed away" from the trigger. For Gradients, the attacks BD+ADV and EABD show low overlaps. However, even in the BD-ONLY setting Gradients is not suitable to detect triggers. For GradCAM, in turn, the trigger is highlighted, as can be seen at the larger values in the BD-ONLY row and in Appendix C. EABD is extremely successful, showing very low overlaps for all thresholds. BD+ADV shows larger values but can still successfully disguise the on-going attack at inference time. In Appendix A, we present more results on the overlap.

5.4. Quadrature Analysis

We find that nearby the trigger (where no input manipulation happens), the target explanation is yielded the worst. Thus, we analyze the effectivity of input-manipulations against clean models to isolate the effect. The input is divided into four quadrants, "not-manipulated," "opposite," "left," and "right," and is perturbed in all but the first quadrant as part of an ADV2 attack [44] aiming for a square target explanation. The quadrants are then rotated by 90 degree four times, resulting in four attacks against each of the three *clean* models (12 attacks in total). Table 4 shows the averaged MSE evaluated separately on each of the quadrants (respecting the rotation). Our results confirm that fooling explanations is more challenging in and nearby not-manipulated areas.

TABLE 4: The MSE results of our quadrature analysis.

Expl.M.	not-manip. ± std	left ± std	right ± std	opposite ± std
Gradients	0.199 ± 0.00	$0.135_{\pm 0.00}$	$0.135_{\pm 0.00}$	0.132 ± 0.00
GradCAM	0.137 ± 0.02	$0.104_{\pm 0.02}$	$0.105_{\pm 0.02}$	0.099 ± 0.02

5.5. Transferability Analysis

Lastly, we research how samples optimized against one model, perform against the other two models. Essentially the earlier made white-box assumption is removed. Table 3 contains the results of this experiment, leading to three findings: For BD+ADV, 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 are much lower and in the range of the baseline (BD-ONLY) for the respective target and explanation method (cf. Table 1). This means, the explanations are not successfully fooled any more.

6. Conclusion

Composite explanation-aware attacks allow 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 fool the XAI techniques at inference time. In contrast to prior work [33], we require mere data poisoning rather than full access to the training procedure and can produce input-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 [44]. This dichotomy is unique to explanation-aware attacks.

Availability

To foster future research on the robustness of explainable machine learning we provide supplementary material at:

`https://xaisec.org`

Acknowledgment

The authors gratefully acknowledge funding from the Helmholtz Association (HGF) within the topic “46.23 Engineering Secure Systems”.

References

[1] D. Arp, M. Spreitzenbarth, M. Hübner, H. Gascon, and K. Rieck. Drebin: Effective and explainable detection of android malware in your pocket. In *Proc. of the Network and Distributed System Security Symposium*, 2014.

[2] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 2015.

[3] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11, 2010.

[4] E. Bagdasaryan and V. Shmatikov. Blind backdoors in deep learning models. In *Proc. of the USENIX Security Symposium*, 2021.

[5] H. Baniecki, W. Kretowicz, and P. Biecek. Fooling partial dependence via data poisoning. *CoRR*, abs/2105.12837, 2021.

[6] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Proc. of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, volume 8190, 2013.

[7] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[8] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. *Proc. of the IEEE Symposium on Security and Privacy (S&P)*, 2017.

[9] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. 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.

[10] X. Chen, C. Liu, B. Li, K. Lu, and D. Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017.

[11] E. Chou, F. Tramèr, and G. Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *Proc. of the IEEE Symposium on Security and Privacy Workshops*. IEEE, 2020.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

[13] B. G. Doan, E. Abbasnejad, and D. 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.

[14] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel. Explanations can be manipulated and geometry is to blame. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[15] O. Eberle, J. Büttner, F. Kräutli, K.-R. Müller, M. Valleriani, and G. Montavon. Building and interpreting deep similarity models. *IEEE Transactions Pattern Analyses and Machine Intelligence*, 44(3), 2022.

[16] A. Ghorbani, A. Abid, and J. Y. Zou. Interpretation of neural networks is fragile. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2019.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.

[18] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. D. McDaniel. Adversarial examples for malware detection. In *Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II*, 2017.

[19] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7, 2019.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[21] A. Hegde, M. Noppel, and C. Wressnegger. Model-manipulation attacks against black-box explanations. In *Proc. of the Annual Computer Security Applications Conference (ACSAC)*, Dec. 2024.

[22] J. Heo, S. Joo, and T. Moon. Fooling neural network interpretations via adversarial model manipulation. In *Proc. of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[23] W. Huang and J. W. Stokes. MtNet: A multi-task neural network for dynamic malware classification. In *Proc. of the Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, 2016.

[24] A. Ivankay, I. Girardi, P. Frossard, and C. Marchiori. Fooling explanations in text classifiers. *Proc. of the International Conference on Learning Representations (ICLR)*, 2022.

[25] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[26] A. Krizhevsky, V. Nair, and G. Hinton. CIFAR (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

[27] J. R. Lee, S. Kim, I. Park, T. Eo, and D. Hwang. Relevance-cam: Your model already knows where to look. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[28] Y. Li. Research and application of deep learning in image recognition. In *Proc. of the IEEE International Conference on Power, Electronics and Computer Applications (ICPECA)*, 2022.

[29] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang. Trojaning attack on neural networks. In *Proc. of the Network and Distributed System Security Symposium (NDSS)*, 2018.

[30] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

[31] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65, 2017.

[32] M. Noppel and C. Wressnegger. SoK: Explainable machine learning in adversarial environments. In *Proc. of the IEEE Symposium on Security and Privacy (S&P)*, May 2024.

[33] M. Noppel, L. Peter, and C. Wressnegger. Disguising attacks with explanation-aware backdoors. In *Proc. of the IEEE Symposium on Security and Privacy (S&P)*, May 2023.

[34] E. Raff, W. Fleshman, R. Zak, H. S. Anderson, B. Filar, and M. McLean. Classifying sequences of extreme length with constant memory applied to malware detection. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2021.

[35] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.

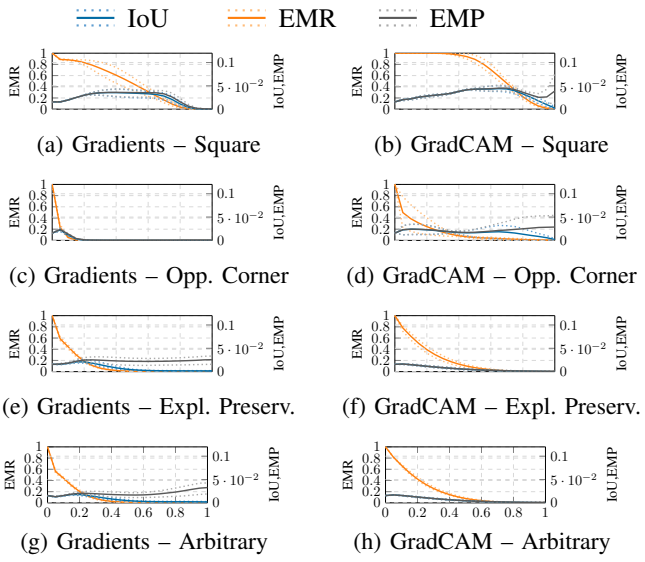
[36] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2), 2020.

[37] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Proc. of the Annual Conference on Neural*

Information Processing Systems (NeurIPS), 2018.
[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.
[39] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2014.
[40] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2014.
[41] J. Vadillo, R. Santana, and J. A. Lozano. Adversarial Attacks in Explainable Machine Learning: A Survey of Threats Against Models and Humans. *WIREs Data Mining and Knowledge Discovery*, 2024.
[42] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4), 2004.
[43] H. Zhang, J. Gao, and L. 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.
[44] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, and T. Wang. Interpretable deep learning under fire. In *Proc. of the USENIX Security Symposium*, 2020.
[45] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

A. Additional Trigger Overlap Results

Here we present the trigger overlap as line graphs. In Fig. 2, find the plots for BD+ADV. The dashed lines represent the minimum and maximum values. Not surprisingly, the target explanation "opposite corner" truely works best in hiding the trigger. The results for the explanation preserving and the arbitrary target explanations 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 Fig. 3 and 4 with the same axis scaling. In Fig. 3b, it can be seen how GradCAM highlights the trigger.



(a) Gradients – Square (b) GradCAM – Square
(c) Gradients – Opp. Corner (d) GradCAM – Opp. Corner
(e) Gradients – Expl. Preserv. (f) GradCAM – Expl. Preserv.
(g) Gradients – Arbitrary (h) GradCAM – Arbitrary

Figure 2: Depictions of the trigger overlap of BD+ADV for the two explanation methods Gradients and GradCAM and the four different target explanations.

(a) Gradients (b) GradCAM

Figure 3: Depictions of the trigger overlap of BD-ONLY for the two explanation methods Gradients and GradCAM.

B. Hyperparameter

In this section, we provide further details on the hyperparameters of our experiments.

Model Training. For each training we use the default parameters of 200 epochs, with a learning rate of 0.1, a weight decay rate of 0.0001, and a momentum of 0.9. The optimization is done with SGD in `pytorch`. We use a multistep learning rate scheduling, reducing the learning rate at the epochs 100 and 150. In all trainings, we use horizontal flipping and a random cropping as data augmentation. *Note how this moves the trigger around or crops it from the image.*

Optimizing the EABD Attack. In Table 6, we present the determined hyperparameters for the EABD attack. Using the `optuna` 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 $[4, 40]$. These numbers are the ranges as suggest by the code base of the original work [33]. The best models are selected according to the scoring as suggest in the original work.

Hyperparameter of the Inference-Time Attacks. In Table 5, we present the hyperparameter of the individual inference time attacks. The best parameters are determined using the `optuna` random sampler. We sample the learning rate from the interval $[0.001, 0.032]$, the loss weight from $[0.01, 0.99]$, and the beta growth rate from $[0.0, 1.0]$. The best set of parameters is selected according to the following formula: $\min(\ (1-ASR)\cdot 2+MSE\ )$. When scaling explanations to $[0, 1]$ we use a minimal upper limit of 0.000001 as a stability term to enhance the numerical stability.

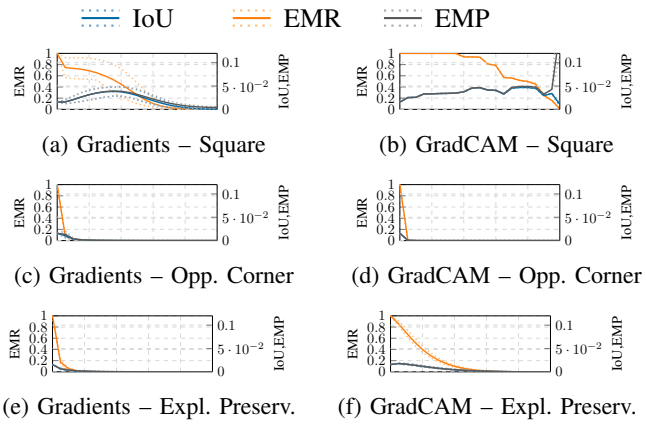
C. Qualitative Results

In Fig. 5, 6, 7, and 8, we provide qualitative examples of the above introduced and mentioned attacks.

As can be seen in Fig. 5 and 6, the attacks ADV and BD+ADV produce more precise explanation, compared to ADV2, in particular for the Gradients explanation method. We can also see that the explanation is fooled a little worse nearby the trigger in the BD+ADV attack, an effect we investigate in Section 5.4. The EABD attack, in turn, shows the most precise explanation for both explanation methods, but also

changes the model in this regard. In Fig. 7, and 8, we find that when fooling GradCAM sometimes additional highlights of similar size appear. In the attacks against Gradients we see that the explanation fooling works surprisingly well for some samplers but not at all for others. We leave the further investigation of this observation as future work. The EABD attack, in turn, fails to show small highlights like the opposite corner in the Gradients explanation method. We also leave the improvement of this attack as future work. Overall, we find it surprising that such a gradient, showing a little square in this specific position, exists nearby so many samples, and we want to encourage the community to have closer look on such effects.

Fig. 9 displays the GradCAM and 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 Gradients explanation do not.



(a) Gradients – Square (b) GradCAM – Square

(c) Gradients – Opp. Corner (d) GradCAM – Opp. Corner

(e) Gradients – Expl. Preserv. (f) GradCAM – Expl. Preserv.

Figure 4: Depictions of the trigger overlap of EABD attack for the two explanation methods Gradients and GradCAM and the three different target explanations.

TABLE 5: The hyperparameters of the input-manipulation attacks.

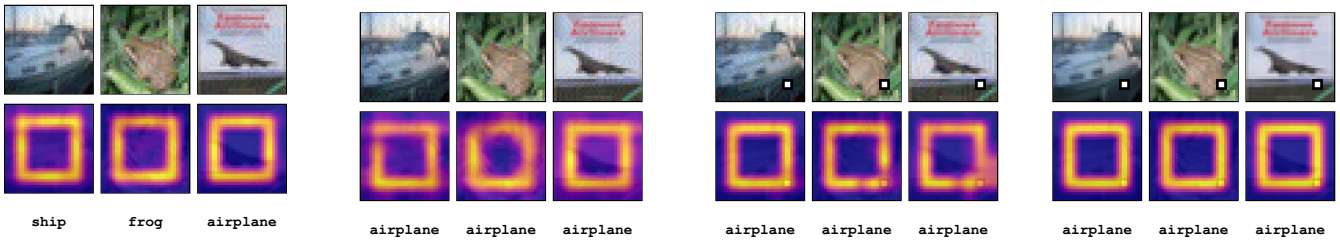
Expl.M.	Trigger	Attack	η	λ	β-grow	η	λ	β-grow	η	λ	β-grow	η	λ	β-grow
Gradients	–	ADV	0.001	0.893	0.187	0.002	0.722	0.129	0.002	0.896	0.815	0.002	0.503	0.109
		ADV	0.002	0.819	0.043	0.002	0.692	0.149	0.002	0.271	0.192	0.001	0.686	0.424
		ADV	0.001	0.881	0.148	0.001	0.574	0.612	0.002	0.330	0.689	0.001	0.299	0.621
	Square	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
		BD+ADV	0.002	0.910	0.036	0.006	0.113	0.042	0.001	0.660	0.144	0.001	0.274	0.120
		BD+ADV	0.001	0.042	0.035	0.005	0.201	0.011	0.001	0.086	0.144	0.001	0.083	0.063
		BD+ADV	0.003	0.942	0.051	0.003	0.108	0.107	0.001	0.366	0.235	0.001	0.077	0.099
GradCAM	–	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
	Square	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
		BD+ADV	0.002	0.251	0.055	0.010	0.105	0.033	0.021	0.720	0.124	0.008	0.189	0.152
		BD+ADV	0.003	0.087	0.098	0.007	0.031	0.172	0.014	0.289	0.061	0.021	0.096	0.056
		BD+ADV	0.002	0.075	0.069	0.004	0.037	0.141	0.019	0.163	0.008	0.008	0.162	0.123

(a) Square (b) Opposite Corner (c) Expl. Preserv. (d) Arbitrary

TABLE 6: The hyperparameters of the EABD attacks.

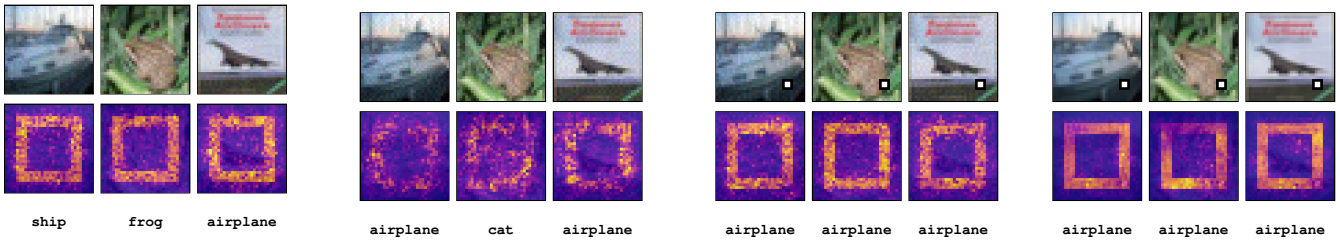
Expl.M.	η	λ	batchsize	epochs	η	λ	batchsize	epochs	η	λ	batchsize	epochs
Gradients	0.001	0.815	42	35	0.001	0.812	64	39	0.001	0.829	76	28
	0.001	0.828	104	30	0.001	0.876	69	38	0.001	0.839	77	37
	0.001	0.838	191	6	0.001	0.845	40	24	0.001	0.881	170	29
GradCAM	0.001	0.802	31	28	0.002	0.840	69	36	0.002	0.837	30	33
	0.001	0.882	48	33	0.002	0.874	63	11	0.001	0.819	70	30
	0.001	0.881	51	9	0.001	0.812	39	26	0.001	0.922	65	33

(a) Square (b) Opposite Corner (c) Expl. Preserv.



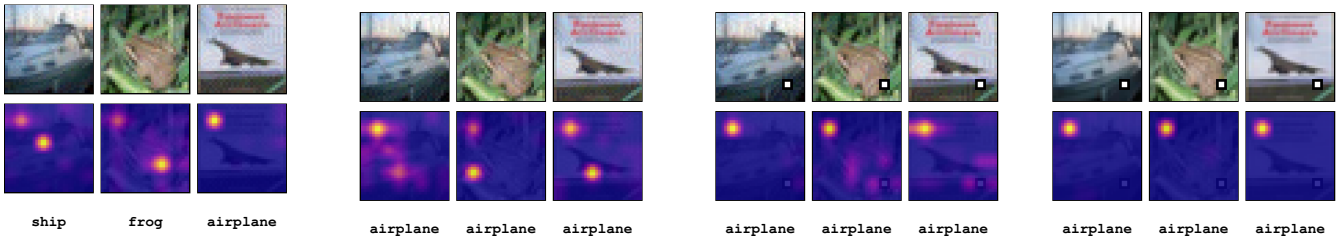
(a) The ADV attack (b) The ADV2 attack (c) The BD+ADV attack (d) The EABD attack

Figure 5: For the explanation method GradCAM and the target Square



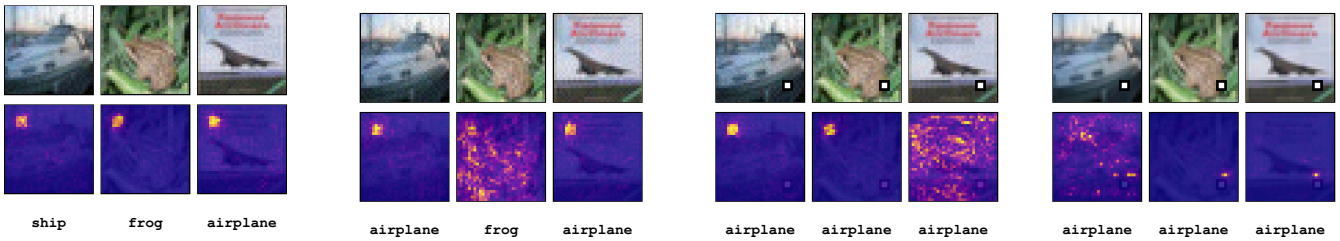
(a) The ADV attack (b) The ADV2 attack (c) The BD+ADV attack (d) The EABD attack

Figure 6: For the explanation method Grad and the target Square



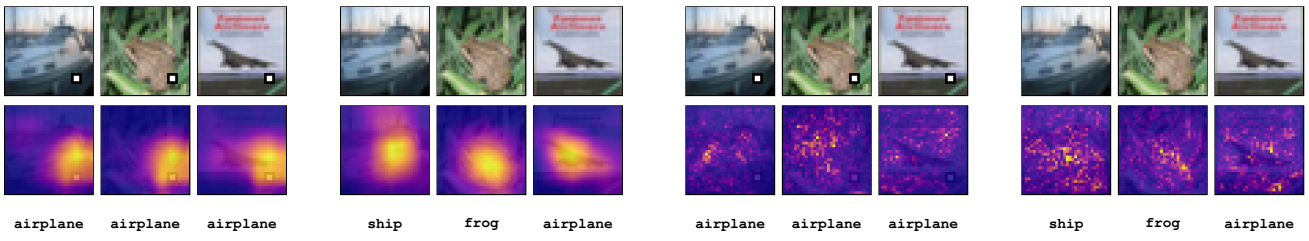
(a) The ADV attack (b) The ADV2 attack (c) The BD+ADV attack (d) The EABD attack

Figure 7: For the explanation method GradCAM and the target Opposite Corner



(a) The ADV attack (b) The ADV2 attack (c) The BD+ADV attack (d) The EABD attack

Figure 8: For the explanation method Grad and the target Opposite Corner



(a) The BD-ONLY attack (b) The CLEAN samples (c) The BD-ONLY attack (d) The CLEAN samples

Figure 9: For the explanation methods GradCAM (left two blocks) and Gradients (right two blocks)