Mitigating Backdoors via Decoy Shortcuts and Knowledge Decoupling

Zixuan Zhu[1,2], **Rui Wang**[1,2*], **Lihua Jing**[1,2] and **Jinwen Zhong**[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
{zhuzixuan, wangrui, jinglihua, zhongjinwen}@iie.ac.cn

Abstract

Backdoor attacks pose a serious threat to deep neural networks, especially when training relies on third-party data, allowing adversaries to inject malicious behaviors through data poisoning. In this work, we reveal that backdoor behaviors tend to be absorbed by a simpler parallel branch when jointly trained with the main network. Motivated by this insight, we propose *Trapping and Removing (TR)*, a simple yet effective training-time defense that introduces a lightweight shortcut branch as a "honeypot" to trap backdoor knowledge. After training, backdoors can be removed by discarding the shortcut, without requiring any additional data. To further enhance backdoor isolation while maintaining benign performance, we design a knowledge decoupling strategy with entropy-based weight assignment, encouraging poisoned samples to flow through the honeypot while guiding the main network to focus on benign learning. In addition, we introduce an automatic shortcut generation strategy to improve generalization across model architectures. Extensive experiments on four benchmark datasets and five model architectures demonstrate that our approach effectively mitigates a wide range of backdoor attacks while preserving performance on benign data. Code: github.com/Zixuan-Zhu/TR.

1 Introduction

With large-scale training data and powerful computational resources, deep neural networks (DNNs) have achieved remarkable performance and are widely deployed in real-world applications [He *et al.*, 2016a; Wang *et al.*, 2020; Zhang *et al.*, 2020]. However, the high cost of collecting training data often drives developers to rely on third-party datasets, inadvertently exposing models to backdoor threats.

Backdoor attackers can manipulate models by poisoning a small subset of training data, embedding a specific *trigger* and assigning a corresponding *target label*. During training, the model implicitly learns this malicious association, causing any input containing the trigger to be misclassified as the target label at inference time. Deploying backdoored models can have severe consequences, especially in safety-critical applications such as autonomous driving. Prior work indicates that backdoors, once implanted, are difficult to erase or unlearn [Yu *et al.*, 2025], making training-time countermeasures particularly important.

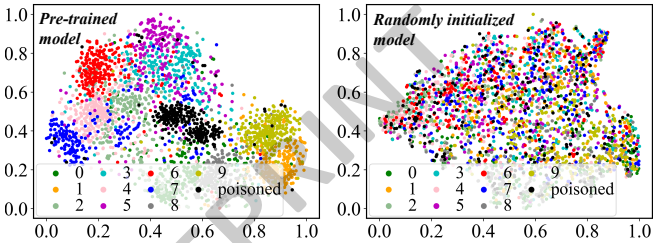


Figure 1: t-SNE visualizations of poisoned samples in the feature space produced by ResNet-18 models: pre-trained on ImageNet (left) vs. randomly initialized (right).

PBE [Mu *et al.*, 2023] and Neural Cleanse [Wang *et al.*, 2019] show that backdoors tend to form shortcut connections in the model, directly associating trigger patterns with target labels. These methods exploit this property to detect and erase backdoors after training. Inspired by this, we are motivated to explore *whether a decoy shortcut can be explicitly introduced during training to trap backdoor knowledge as it forms*. Several studies [Tang *et al.*, 2023; Liu *et al.*, 2023] explore backdoor defense in NLP by confining backdoor knowledge within shallow layers or lightweight models, thereby preventing the main network from internalizing it. These approaches are primarily designed for scenarios where attackers fine-tune pre-trained clean models with poisoned data. In contrast, backdoor injection in image classification is typically performed during training from scratch. To illustrate this difference, Figure 1 compares feature distributions of poisoned samples crafted by BadNets [Gu *et al.*, 2017] on CIFAR-10 [Alex, 2009], under both pre-trained and randomly initialized models. In the left subfigure (pre-trained), poisoned samples cluster in a sparse and relatively isolated region, suggesting a degree of separation from benign ones. In contrast, in the right subfigure (from-scratch), poisoned and benign samples appear highly entangled, making separation more challenging and limiting the effectiveness of NLP-oriented defenses in this setting.

*Corresponding author

In our experiments, we find that backdoor knowledge can be readily captured by a simple parallel branch when training from scratch, as demonstrated in Section 4.1. Building on this observation, we propose a novel defense mechanism, *Trapping and Removing (TR)*, that protects networks from backdoor attacks by trapping all backdoor knowledge within a dedicated "honeypot" branch. This honeypot is introduced prior to training and implemented as a lightweight shortcut running parallel to the original backbone. In early epochs, the shortcut can effectively capture backdoor knowledge, albeit along with a few benign knowledge. To better focus its learning on backdoor knowledge and direct the original network to solely learn benign knowledge in subsequent training, we design a knowledge decoupling strategy equipped with an entropy-based weight assignment module, which dynamically guides poisoned samples toward learning through the shortcut branch while benign samples are processed by the original network. In this manner, the backdoor can be effectively removed by discarding the shortcut after training without resorting to any additional data, while maintaining high performance on benign data. To facilitate practical deployment, we further introduce an adaptive shortcut generation strategy that automatically tailors the shortcut to the backbone network.

The main contributions of this paper are as follows:

- We reveal an inherent characteristic of backdoor attacks: backdoor knowledge tends to be learned through a simpler path during the early stages of training.
- Building on this insight, we propose *TR*, a training-time defense based on decoy shortcuts and knowledge decoupling, which enables clean model training under backdoor attacks without any additional data and generalizes to both training-from-scratch and fine-tuning attacks.
- We design an adaptive strategy to automatically generate shortcuts for backbone networks, enhancing generalization and facilitating practical deployment.
- We evaluate our method on multiple datasets and models against ten backdoor attacks, showing robust performance and outperforming ten state-of-the-art defenses.

2 Related Work

2.1 Backdoor Attack and Defense

Backdoor Attack. Based on the attack strategy, poisoning-based backdoor attacks can be broadly classified into non-optimized and optimized attacks. *Non-optimized attacks* [Lv *et al.*, 2023; Gao *et al.*, 2024] craft poisoned samples before training and distribute them to victims, often via third-party datasets. Early methods [Gu *et al.*, 2017; Chen *et al.*, 2017] used simple, visible triggers, while later works such as WaNet [Nguyen and Tran, 2021] and Dynamic [Nguyen and Tran, 2020] introduced imperceptible or sample-specific triggers. To evade manual inspection, clean-label variants [Barni *et al.*, 2019; Wu *et al.*, 2025] embed triggers into benign samples under the target class. *Optimized attacks* [Li *et al.*, 2021c; Cheng *et al.*, 2024] instead generate and refine poisoned samples during the victim model's training process. These methods often rely on additional losses or co-trained trigger generators to enhance stealth. IBA [Li *et al.*, 2021c] explicitly encourages backdoor invisibility in feature space, while LOTUS [Cheng *et al.*, 2024] leverages partition-specific training to control the backdoor behavior more precisely. Optimized attacks require control over the model training process, which is beyond the scope of this paper.

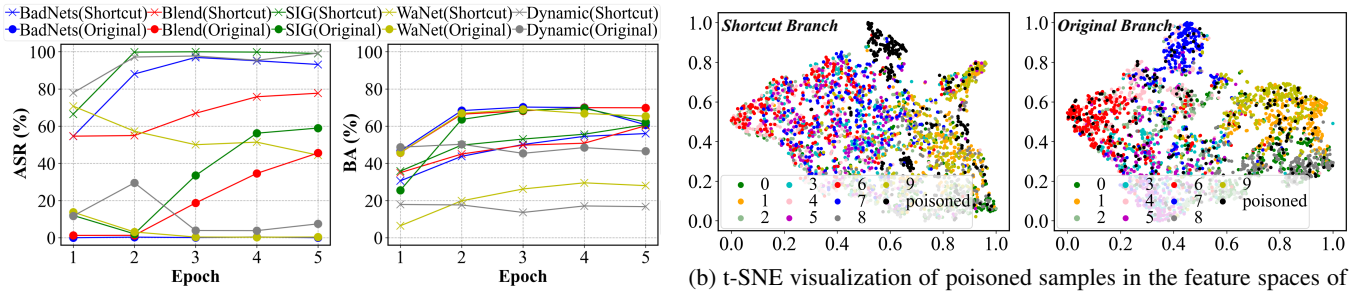
Backdoor Defense. Backdoor inhibiting [Zhu *et al.*, 2023; Yu *et al.*, 2025] and erasing [Li *et al.*, 2021b; Liu *et al.*, 2018] are two widely used defense strategies. *Inhibiting-based defenses* aim to suppress backdoor formation during training by modifying the training process. For example, CBD [Zhang *et al.*, 2023] applies causal theory to prevent learning the backdoor path, DBD [Huang *et al.*, 2022] decouples the training process, and ASD [Gao *et al.*, 2023] continuously divides the training data. *Erasing-based defenses* focus on purifying poisoned models after backdoor training and typically require benign samples. FP [Liu *et al.*, 2018] prunes low-activation neurons and fine-tunes the model, while NAD [Li *et al.*, 2021b] performs fine-tuning followed by knowledge distillation. The defense proposed in this paper falls into the inhibiting-based category.

2.2 Honeypot and Shortcut in Backdoor Defense

Several works adopt similar concepts of honeypots or shortcuts for backdoor defense, which differ from our approach. SSFT [Yang *et al.*, 2023] erases backdoors by removing skip connections in ResNets and finetuning with 10% benign samples. T&R [Wang *et al.*, 2022] baits the backdoor into a classification head and replaces it with another head finetuned on benign samples. DPoE [Liu *et al.*, 2023] and [Tang *et al.*, 2023] target NLP tasks, using shallow layers to absorb backdoor knowledge while preventing deeper layers from learning it. DPoE further combines this with a Product-of-Experts framework to isolate clean behavior. However, in NLP, backdoor injection is typically implemented by fine-tuning pre-trained clean models with poisoned data, where benign and backdoor knowledge are naturally decoupled, making defense more manageable. In contrast, we address the more challenging training-from-scratch injection, where benign and backdoor knowledge are entangled. Besides, we introduce a parallel shortcut branch as the honeypot, which is effective not only against training-from-scratch attacks but also against fine-tuning attacks.

2.3 Knowledge Decoupling

Feature decoupling [Wang *et al.*, 2021; Yang and Yang, 2022] is a representative of knowledge decoupling, aiming to separate feature representations into independent components, minimize redundancy, and improve the model's ability to learn more discriminative information. For example, DeAOT [Yang and Yang, 2022] decouples object-agnostic and object-specific features for better propagating annotations in semi-supervised video object segmentation. FDTrack [Jin *et al.*, 2023] applies feature decoupling to balance the conflicting feature requirements for detection and re-identification in multi-object tracking. The specific implementation and objectives of knowledge decoupling can vary depending on the task. In this paper, we propose to decouple the benign and poisoned knowledge during backdoor training for defending.



(a) BA and ASR of two branches under five attacks

(b) t-SNE visualization of poisoned samples in the feature spaces of two branches (epoch 1)

Figure 2: Visualization of learning behaviors of original and shortcut branches under attacks.

3 Preliminaries

3.1 Poisoning-based Backdoor Attacks

We consider poisoning-based backdoor attacks in image classification. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ be the training set with N samples, where each image $\boldsymbol{x}_i \in \{0, 1, \ldots, 255\}^{C \times H \times W}$ and label $y_i \in \{0, \ldots, K-1\}$. Here, K is the number of classes, and H, W, C denote image height, width, and channels. In a backdoor attack, the attacker selects a subset of $\mathcal{D}$ and applies a modification function $G(\cdot)$ to generate poisoned samples: $\mathcal{D}_m = \{(\boldsymbol{x}', y_t) | \boldsymbol{x}' = G(\boldsymbol{x}), (\boldsymbol{x}, y) \in \mathcal{D} \setminus \mathcal{D}_b\}$, where $\mathcal{D}_b$ contains the remaining benign samples and y_t is the target label. The resulting poisoned dataset is $\mathcal{D}_p = \mathcal{D}_m \cup \mathcal{D}_b$, which victim users use unknowingly to train their models. At test time, the attacker triggers misclassification by applying $G(\cdot)$. The poisoning rate is defined as $r = \frac{|\mathcal{D}_m|}{|\mathcal{D}|}$.

3.2 Threat Model

Attacker's Capability. We consider a common threat scenario involving third-party datasets, where an attacker can arbitrarily modify the training data before releasing it to victim users, but has no access to information beyond the dataset itself, such as the model architecture or loss functions. The attacker's objective is to cause the trained model to predict a predefined target label for triggered inputs while maintaining correct predictions on benign inputs. *We further consider adaptive attack scenarios in Appendix C, where the attacker is aware of the existence of our defense.*

Defender's Capability. The defender is assumed to have full control over the training process, but no prior knowledge of the backdoor and no access to any additional samples. Whether the training dataset is poisoned is also unknown to the defender. The defense objective is to prevent the trained model from predicting triggered samples as the target label while preserving accuracy on benign data.

4 Method

4.1 Distinct Learning Behaviors of Dual Branches under Backdoor Attacks

In this subsection, we analyze the distinct learning behaviors of the original and shortcut branches under backdoor attacks and discuss their inherent mechanisms.

Settings. We conduct experiments on CIFAR-10 using a modified WRN-16-1 [Zagoruyko, 2016] architecture, augmented with a two-layer shortcut that directly connects the input to the classifier. During training, the final prediction is computed as the average of the outputs from both the shortcut and original branches (i.e., equal weights of 0.5). Figure 2a reports the benign accuracy (BA) and attack success rate (ASR) of both branches under five different attacks during the early training epochs. In addition, Figure 2b visualizes the feature distributions of BadNets-poisoned samples produced by the two branches after a single epoch. More detailed experimental settings can be found in *Appendix* A.3.

Results. As shown in Figure 2a, the shortcut branch exhibits higher ASR but lower BA than the original branch during early training. This disparity arises from intrinsic differences between benign and poisoned samples: benign samples contain complex, diverse feature patterns better captured by the deeper original branch, whereas poisoned samples share similar trigger patterns and require only a few neurons to associate with the target label [Liu *et al.*, 2018]. Our theoretical analysis (*Appendix* B) further confirms that this structural asymmetry induces a gradient focusing effect ($||\nabla_{\theta_h}|| \gg ||\nabla_{\theta_o}||$) for poisoned samples. This mechanism results in a pronounced simplicity bias, where the shortcut establishes a dominant gradient flow to preferentially capture backdoor knowledge. Figure 2b further corroborates this: in the shortcut branch's feature space, most poisoned samples form a tight cluster while clean samples are relatively scattered; conversely, in the original branch's space, clean samples of the same class form coherent clusters, and poisoned samples are more dispersed—often near their ground-truth classes. These results suggest that backdoor knowledge tends to be learned by the simpler shortcut, while benign knowledge is primarily captured by the more complex original branch.

4.2 Adaptive Shortcut Generation

As Figure 3 illustrates, our defense method consists of three steps. First, we add a simple honeypot shortcut that runs parallel to the original backbone before training. The modified network is denoted as $\{S_o, S_h, S_c\}$, where S_o is the original backbone, S_h is the honeypot shortcut, and S_c is the classifier. Next, we train the modified network on the poisoned dataset, carefully decoupling the knowledge between the original branch $\{S_o, S_c\}$ and the shortcut branch

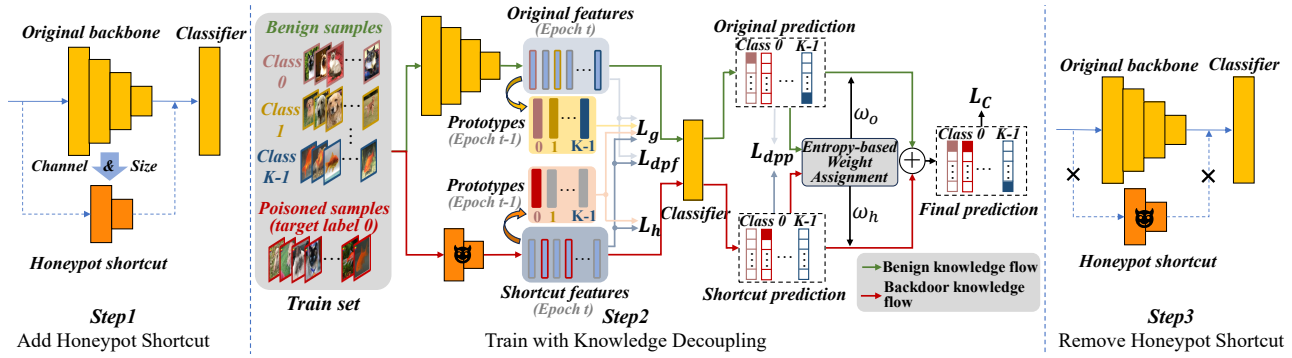


Figure 3: Our defense pipeline comprises three steps. First, we add a simple honeypot shortcut that runs parallel to the original backbone. Next, we train the model on the poisoned dataset while decoupling the knowledge between the original backbone and the honeypot shortcut. This enables the honeypot to capture backdoor knowledge while the original backbone retains only benign knowledge. Finally, after training, we erase the backdoor by removing the honeypot shortcut from the model.

$\{S_h, S_c\}$, enabling the shortcut to stabilize with backdoor-related information while allowing the original network to retain only benign knowledge. Finally, the backdoor can be easily erased by simply removing the shortcut from the model, without affecting the original network's benign performance.

We construct the honeypot shortcut as a direct path from the input to the final classifier. Specifically, each $2\times$ downsampling operation in the backbone is replaced with a 3×3 convolution of stride 2, while matching the corresponding channel dimensions. For instance, two convolution blocks suffice for WRN-16-1, which has two downsampling operations, while ResNet-18 requires three. More details on shortcut designs for CNNs and ViTs are provided in *Appendix* A.4.

To help the shortcut capture backdoor knowledge more quickly and effectively, we incorporate a spatial attention mechanism to highlight key regions. Let $f \in \mathbb{R}^{h\times w\times c}$ be the feature map, where h, w, and c denote height, width, and channels. Following [Woo *et al.*, 2018], the attention map $M \in \mathbb{R}^{h\times\ w}$ is computed by applying a convolution to f with a single output channel. We then employ the attention map to highlight informative regions of the feature maps:

$$f' = ReLU(M) \otimes f, \tag{1}$$

where $\otimes$ indicates element-wise multiplication and f' represents the output of the attention module.

For poisoned samples, spatial attention can quickly focus on trigger patterns due to their similarity and strong correlation with the target label. Consequently, each convolution block in the shortcut is structured as *Conv-Attention-BN-ReLU* to enhance its ability to capture backdoor knowledge.

4.3 Entropy-based Weight Assignment

In the modified network, we define the final prediction $P(\boldsymbol{x})$ as a weighted combination of the outputs from both two branches:

$$P(\boldsymbol{x}) = w_o \cdot P_o(\boldsymbol{x}) + w_h \cdot P_h(\boldsymbol{x}), \tag{2}$$

where

$$\begin{aligned} w_o + w_h &= 1, \\ P_o(\boldsymbol{x}) &= softmax(S_c(S_o(\boldsymbol{x}))), \\ P_h(\boldsymbol{x}) &= softmax(S_c(S_h(\boldsymbol{x}))). \end{aligned} \tag{3}$$

Here, w_o and w_h represent the respective weights assigned to each branch's prediction. The classification loss is defined as follows:

$$L_c = \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_p}[\mathcal{L}_{CE}(P(\boldsymbol{x}), y)]. \tag{4}$$

where $\mathcal{L}_{CE}$ denotes the cross-entropy loss.

Setting w_o and w_h is challenging: if w_o is too large, the original branch may inadvertently learn backdoor knowledge, while a smaller w_o may impede its learning of benign knowledge. Additionally, as indicated in Figure 2a, the shortcut initially captures some benign knowledge. Improper weight settings may cause this portion of benign knowledge to remain permanently embedded in the shortcut, resulting in the original network losing the opportunity to learn it.

To address this issue, we propose a dynamic weight assignment strategy based on prediction entropy. This entropy can be calculated by:

$$E(\boldsymbol{p}) = -\sum_{k=0}^{K-1} \boldsymbol{p}_k \log \boldsymbol{p}_k, \tag{5}$$

where $\boldsymbol{p}$ is the softmax prediction. For each sample $\boldsymbol{x}$, we first pass it through both branches to obtain predictions $P_o(\boldsymbol{x})$ and $P_h(\boldsymbol{x})$, then assign their contributions to the final prediction as follows:

$$\begin{aligned} w_o &= \frac{E(P_o(\boldsymbol{x}))}{E(P_o(\boldsymbol{x})) + E(P_h(\boldsymbol{x}))}, \\ w_h &= \frac{E(P_h(\boldsymbol{x}))}{E(P_o(\boldsymbol{x})) + E(P_h(\boldsymbol{x}))}. \end{aligned} \tag{6}$$

During training, poisoned samples are initially learned by the shortcut branch, inevitably along with a few benign samples. For these poisoned and benign samples, the entropy $E(P_h(\boldsymbol{x}))$ is low, and the weight w_o is high, which gives the original branch an opportunity to learn the benign sample among them. Simultaneously, this weight assignment also enables the original branch to learn from the poisoned ones. To enhance the backdoor knowledge captured by the shortcut branch and prevent it from leaking to the original branch, we design a knowledge decoupling strategy in Section 4.4.

4.4 Knowledge Decoupling

Motivated by the distinct learning behaviors in Figure 2, we decouple the shortcut and original branches by enforcing constraints in both feature space and final predictions. Through this decoupling, we aim for the original branch to correctly predict only benign samples, while the malicious predictions for poisoned samples are provided by the shortcut branch.

For each sample $\boldsymbol{x}$, we obtain feature vectors $S_o(\boldsymbol{x})$ and $S_h(\boldsymbol{x})$ from the original backbone and honeypot shortcut. We then encourage feature-level decoupling by maximizing their cosine distance:

$$L_{dpf} = \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_p}\langle S_o(\boldsymbol{x}), S_h(\boldsymbol{x})\rangle \tag{7}$$

where $\langle\cdot,\cdot\rangle$ denotes cosine similarity. However, feature decoupling alone is insufficient, as the classifier S_c may still adjust itself to accommodate backdoor requirements. Therefore, we introduce an additional prediction-level decoupling loss that jointly constrains the classifier and feature extractors:

$$L_{dpp} = -\mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_p}\|P_o(\boldsymbol{x}), P_h(\boldsymbol{x})\|_2 \tag{8}$$

The cooperation between L_{dpf} and L_{dpp} effectively enables the original and shortcut branches to produce different predictions for each sample.

However, both losses only consider the decoupling state at the current epoch, and their constraints are satisfied as long as the two branches learn the same sample as different classes. This implies that if the shortcut branch fails to make the malicious prediction for a poisoned sample, the original branch may classify it as the target class due to the classification constraint, resulting in the learning of backdoor knowledge. This issue is more likely when trigger patterns are complex, making it difficult for the shortcut branch to capture stable backdoor knowledge in early epochs. In such cases, the original branch may continuously compete with the shortcut for backdoor knowledge, even snatching it from the shortcut by leveraging its greater learning capacity. To address this issue, we leverage previously learned knowledge to guide the current learning direction. Specifically, we construct a feature prototype set for each branch to summarize their knowledge from the previous epoch, as storing full features for all samples is memory-intensive. The prototypes are defined as follows:

$$\boldsymbol{c}_o^k = \frac{1}{N_k}\sum_{i=1}^{N_k} S_o^{t-1}(\boldsymbol{x}_i^k), \quad \boldsymbol{c}_h^k = \frac{1}{N_k}\sum_{i=1}^{N_k} S_h^{t-1}(\boldsymbol{x}_i^k), \tag{9}$$

where $\{\boldsymbol{x}_i^k\}_{i=1}^{N_k}$ denotes samples with label k, and S_o^{t-1} and S_h^{t-1} are the original and shortcut branches from the previous epoch. The prototype sets for the two branches are denoted as $\{\boldsymbol{c}_o^k\}_{k=0}^{K-1}$ and $\{\boldsymbol{c}_h^k\}_{k=0}^{K-1}$. We push the two branches to learn in different directions by adding a loss that maximizes the distance between their features and the corresponding prototypes of the other branch:

$$L_g = \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_p}[\langle S_o(\boldsymbol{x}), \boldsymbol{c}_h^y\rangle + \langle S_h(\boldsymbol{x}), \boldsymbol{c}_o^y\rangle]. \tag{10}$$

Additionally, we enhance the backdoor knowledge in the shortcut branch by encouraging it to learn along its previous direction, which can be interpreted as the backdoor direction:

$$L_h = -\mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_p}\langle S_h(\boldsymbol{x}), \boldsymbol{c}_h^y\rangle. \tag{11}$$

Intuitively, L_h functions as an inertial constraint that encourages the shortcut branch to remain on its discovered backdoor learning trajectory, while its combination with L_g imposes an exclusivity constraint that prevents the original branch from "stealing" backdoor knowledge, without hindering its capacity to learn benign representations.

In summary, the training loss can be expressed as follows:

$$L = L_c + L_g + L_h + L_{dpf} + \alpha \cdot L_{dpp}, \tag{12}$$

where α controls the strength of decoupling at the prediction level. Specifically, L_c is applied alone during the first epoch to warm up the network, allowing each branch to initially learn its respective knowledge. In the subsequent decoupling, the backdoor knowledge will be channeled into the shortcut branch, while the benign knowledge will flow to the original branch. This behavior is analyzed in Section 5.3, and the dynamic of w_o and w_h are further examined in Section 5.4.

5 Experiments

5.1 Experimental Settings

Datasets and Models. In the main paper, we evaluate our defense on CIFAR-10 with WRN-16-1 and on GTSRB [Stallkamp *et al.*, 2011] with ResNet-18 to demonstrate effectiveness across architectures. To further evaluate generalization across datasets and architectures, we include additional experiments on CIFAR-100 [Alex, 2009], ImageNet-1k [Deng *et al.*, 2009], as well as on PreActResNet-18 [He *et al.*, 2016b], VGG-16 [Simonyan and Zisserman, 2014], and ViTs [Dosovitskiy *et al.*, 2020] in *Appendix* D. Dataset details are provided in *Appendix* A.1.

Attack Configures. We consider ten state-of-the-art backdoor attacks, including six dirty-label methods: BadNets, Blend [Chen *et al.*, 2017], WaNet [Nguyen and Tran, 2021], Dynamic [Nguyen and Tran, 2020], DataFree [Lv *et al.*, 2023], and DUBA [Gao *et al.*, 2024], and four clean-label attacks: SIG [Barni *et al.*, 2019], CL [Turner *et al.*, 2019], Narcissus [Zeng *et al.*, 2023], and PCBA [Wu *et al.*, 2025]. All attacks are implemented according to their released code to ensure strong performance. Target labels are set to 0 for CIFAR-10 and 1 for GTSRB, with a default poisoning rate of 10%. Implementation details are provided in *Appendix* A.2.

Defense Configures and Training Details. We compare our method with ten state-of-the-art defenses: FP [Liu *et al.*, 2018], NAD [Li *et al.*, 2021b], ABL [Li *et al.*, 2021a], DBD [Huang *et al.*, 2022], CBD [Zhang *et al.*, 2023], ASD [Gao *et al.*, 2023], V&B [Zhu *et al.*, 2023], PIPD [Chen *et al.*, 2024], PDB [Wei *et al.*, 2024], and ESTI [Yu *et al.*, 2025]. FP and NAD require 5% local benign samples, while ASD, PDB, and ESTI use 0.2%, 10%, and 1% of the training data as seed samples, respectively. Our model is trained for 200 epochs using SGD with an initial learning rate of 0.1, decayed by a factor of 10 every 50 epochs. The weighting factor α is set to 1 for CIFAR-10 and linearly decreased from 2 to 1 over the first 50 epochs for GTSRB.

Evaluation Metrics. We evaluate the defense performance using two metrics: Benign Accuracy (BA) on clean data and Attack Success Rate (ASR) on poisoned data. The lower the ASR and the higher BA, the more effective the defense.

Dataset	Attack	No Defense		FP		NAD		ABL		DBD		CBD		ASD		V&B		PIPD		PDB		ESTI		TR (Ours)	
		BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
CIFAR-10	None	91.19	-	86.44	-	87.34	-	70.35	-	70.10	-	84.51	-	85.96	-	84.37	-	85.92	-	82.60	-	82.22	-	**90.37**	-
	BadNet	90.46	99.93	87.21	5.63	86.57	1.93	86.11	3.04	86.94	1.76	87.46	1.06	84.58	1.51	88.28	1.84	89.13	0.64	81.60	0.21	88.92	**0.00**	**89.85**	**0.00**
	Blended	90.30	98.63	86.92	6.99	75.97	5.48	85.34	16.23	86.83	5.12	87.48	1.96	86.51	2.02	88.15	2.08	87.35	2.69	80.90	0.95	87.54	64.54	**89.06**	**0.01**
	WaNet	89.71	92.27	85.87	2.62	86.97	4.21	75.74	22.24	84.60	5.86	86.55	4.24	84.85	2.03	84.21	3.34	86.39	4.73	81.16	1.80	83.55	12.02	**88.90**	**1.33**
	Dynamic	89.78	85.03	86.59	10.49	87.07	2.70	85.34	18.46	85.42	10.21	85.67	0.86	83.64	4.49	81.92	47.63	89.52	0.63	80.40	1.92	89.60	**0.23**	**90.01**	1.06
	DataFree	90.82	99.96	86.53	2.07	78.38	3.16	80.13	0.96	71.16	13.41	85.75	2.00	85.95	2.89	88.17	1.41	85.27	0.25	81.37	0.53	**90.50**	**0.00**	90.27	**0.00**
	DUBA	89.66	95.31	80.77	26.89	86.46	43.85	65.55	99.56	69.78	12.73	87.13	2.27	86.04	2.88	84.55	4.18	87.38	5.19	80.71	2.18	20.01	**0.01**	**88.18**	1.58
	SIG	90.85	70.50	86.72	13.62	80.44	0.63	73.19	0.33	69.52	13.38	86.60	26.03	86.29	3.07	85.53	3.46	87.83	0.70	82.12	0.07	**90.97**	0.13	89.42	**0.00**
	CL	85.34	95.44	83.33	19.17	84.89	8.89	86.18	1.86	69.97	3.61	87.46	1.17	84.65	3.93	85.37	2.84	85.12	2.39	82.51	1.76	88.74	**0.26**	**89.33**	0.74
	Narcissus	90.73	95.53	85.47	69.03	87.34	44.20	71.95	45.66	70.02	99.74	84.98	26.43	82.49	98.58	84.69	32.41	89.55	6.81	82.37	1.25	89.15	0.85	**89.90**	**0.53**
	PCBA	90.11	99.51	86.25	51.59	84.88	26.46	65.74	97.74	68.12	98.61	85.54	81.84	81.76	99.97	66.93	**0.00**	80.92	10.38	82.01	0.94	**89.73**	1.44	89.05	0.41
	Average	89.78	93.21	85.57	20.81	83.90	14.15	77.53	30.61	76.24	26.44	86.46	14.79	84.68	22.14	83.78	9.92	86.85	3.44	81.52	1.16	81.87	7.95	**89.40**	**0.57**
GTSRB	None	99.33	-	94.94	-	97.01	-	83.90	-	91.16	-	75.45	-	96.12	-	98.05	-	93.88	-	92.91	-	97.87	-	**98.90**	-
	BadNet	98.64	100.00	96.87	0.03	96.94	0.06	92.58	0.03	86.13	**0.00**	84.65	2.32	96.52	0.10	88.72	0.35	96.50	0.01	92.77	0.02	96.72	**0.00**	**98.57**	**0.00**
	Blended	98.62	99.84	95.03	12.88	93.46	1.30	89.07	5.52	86.25	99.98	76.05	88.18	96.21	0.12	75.84	48.38	94.59	0.83	92.61	0.29	95.02	**0.06**	**97.88**	0.62
	WaNet	96.18	93.53	96.52	18.08	96.17	67.20	94.50	38.29	84.71	**0.05**	90.50	86.11	95.91	42.26	83.16	7.50	95.04	3.48	89.42	0.77	93.84	0.84	**96.59**	**0.05**
	DataFree	97.04	100.00	96.90	0.03	97.37	0.03	95.21	0.03	92.71	0.38	76.18	2.56	95.05	0.09	98.31	0.14	96.42	0.22	92.20	0.18	97.22	**0.00**	**98.80**	**0.00**
	DUBA	98.83	93.91	95.48	20.81	93.26	8.61	94.04	3.84	88.09	0.05	90.86	92.85	95.95	**0.03**	90.77	9.65	96.35	4.37	90.62	3.57	80.21	3.70	**98.58**	**0.03**
	SIG	98.94	100.00	94.58	14.19	88.78	0.02	86.90	99.88	86.84	69.82	89.84	24.17	96.26	45.93	91.28	2.80	92.39	5.27	93.16	59.75	**97.49**	9.04	97.17	**0.00**
	Narcissus	98.50	67.40	94.61	18.56	97.67	5.83	87.46	32.83	92.58	0.45	90.87	58.37	94.60	93.75	97.64	36.71	93.74	3.61	92.82	13.80	97.56	**0.00**	**98.99**	**0.00**
	Average	98.11	93.53	95.71	12.08	94.81	11.86	91.39	25.77	88.19	24.39	85.56	50.65	95.79	26.04	89.39	15.08	95.00	2.54	91.94	11.20	94.01	1.95	**98.08**	**0.10**

Table 1: The performance of 11 defense methods against multiple backdoor attacks on two datasets. The best results are highlighted in **bold**, and the second-best results are underlined. *None* denotes the totoally clean datasets.

5.2 Effectiveness and Efficiency

We compare TR with ten SOTA defenses in Table 1. *No Defense* serves as the attack baseline. Overall, TR achieves the best average BA and ASR on both datasets, even compared with defenses requiring benign samples. It consistently reduces ASR to below or near 1% (often 0%) while maintaining high BA, especially on clean datasets. TR also remains robust under different poisoning rates, as shown in *Appendix* E.1.

Defenses based on data partitioning, such as ABL, DBD, ASD, V&B, and ESTI, inevitably treat a portion of the training samples as poisoned, even when the dataset is clean, which leads to degraded BA. Moreover, all other defenses can be ineffective under certain attack settings, resulting in higher ASR or reduced BA. As for TR, when the data is clean, the shortcut branch captures only a few of easily learned benign information and does not interfere with the main branch in learning generalizable features from harder samples. When the data is poisoned, the shortcut rapidly absorbs backdoor knowledge, thereby preventing the main network from learning it. We further analyze class-wise BA in *Appendix* E.2, showing that TR has minimal impact on benign performance.

We report the training cost of TR in *Appendix* Table 5, showing that TR is more efficient than others. This efficiency stems from introducing only a shallow shortcut, without repeated data partitioning or iterative multi-network training.

Results on a broader range of dataset–architecture combinations, reported in *Appendix* Tables 6, 7, 8, 9, and 10, further demonstrates the generalization capability of TR, including our adaptive shortcut generation strategy. Notably, the Table 7 and 10 results also verify that our method remains effective against fine-tuning attacks.

5.3 Performance of the Two Branches

To better illustrate the decoupling process, Figure 4 presents BA and ASR of the original and shortcut branches. The original branch's BA steadily increases, while that of the shortcut decreases, with both eventually stabilizing. Although the shortcut retains a small amount of benign knowledge (BA $\approx$ 10%), it does not impede the original branch due to entropy-based weight assignment. In early epochs, the shortcut's ASR is markedly higher than the original's, indicating it captures most backdoor knowledge. As training progresses, the original branch's ASR rapidly drops to 0%, while the shortcut remains at 100%, demonstrating that the shortcut effectively traps backdoor behavior and shields the original branch.

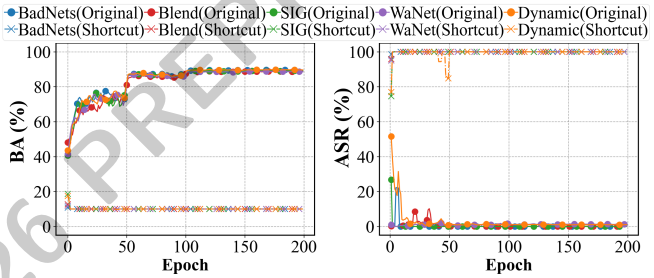


Figure 4: The BA and ASR of both branches during the decoupling training process on CIFAR-10 under the same settings in Table 1.

5.4 Ablation Studies

Weight Assignment. To evaluate our entropy-based weight assignment, we replace it with several fixed schemes in Table 3. Assigning equal weights (0.5) to both branches reduces ASR to 0%, but results in lower benign accuracy, as benign knowledge captured by the shortcut is not learned by the original branch. To address this, we gradually increase w_o and decrease w_h, which improves BA (*e.g.* with 0.8/0.2 and 0.9/0.1) but also raises ASR. This highlights the difficulty of balancing BA and ASR using fixed weights. Without the entropy-based weight assignment, certain benign knowledge may be sacrificed and permanently retained in the shortcut branch to obtain a clean original network. In the last column, we reverse the roles of w_o and w_h in Equation 6, which also lowers BA. As indicated in Figure 2a, the shortcut inevitably

Choices				BadNets		Blend		WaNet		Dynamic		SIG		Mean
L_{dpf}	L_{dpp}	L_h	L_g	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	ASR
				90.77%	99.91%	90.20%	98.46%	88.59%	92.92%	89.78%	85.86%	90.88%	76.80%	90.79%
		✓	✓	90.93%	~~99.70%~~	90.36%	~~97.96%~~	88.65%	~~93.10%~~	89.55%	~~82.01%~~	90.41%	~~70.72%~~	88.70%
✓		✓	✓	90.56%	~~96.84%~~	90.37%	~~97.72%~~	88.97%	~~92.04%~~	89.55%	~~83.24%~~	90.14%	~~75.82%~~	89.13%
	✓	✓	✓	89.74%	0.00%	89.10%	0.00%	88.49%	~~82.81%~~	89.15%	~~65.13%~~	89.65%	~~73.37%~~	44.26%
✓	✓			90.16%	~~96.98%~~	89.26%	0.00%	88.64%	~~83.59%~~	89.05%	~~88.34%~~	89.65%	0.00%	36.30%
✓	✓	✓		89.53%	~~99.69%~~	88.38%	0.00%	88.15%	~~83.64%~~	88.97%	~~66.72%~~	89.60%	0.00%	30.07%
✓	✓		✓	89.93%	0.00%	88.77%	0.00%	88.48%	~~83.84%~~	89.06%	~~83.67%~~	89.93%	~~80.94%~~	49.69%
✓	✓	✓	✓	89.83%	0.00%	89.06%	0.01%	88.90%	1.33%	90.01%	1.06%	89.42%	0.00%	0.48%

Table 2: Ablation study of the four losses on CIFAR-10. The deleted cases indicate ~~defense failures~~. It is evident that combining all four losses is essential for achieving robust defense performance.

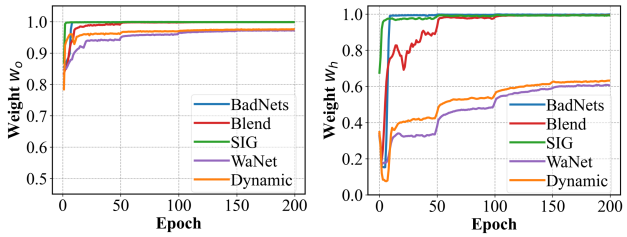


Figure 5: Visualization of w_o for benign samples and w_h for poisoned samples during decoupling training on CIFAR-10.

captures some easy benign samples early on. In the reversed setting, a consistently higher w_h limits the original branch from learning these samples, resulting in BA degradation.

Figure 5 illustrates the weight dynamics in our defense. For benign samples, w_o starts high and quickly reaches 1, as the shortcut initially learns little benign knowledge constrained by its limited capacity. The decoupling loss then drives the shortcut toward confidently incorrect predictions, reflected in its low BA in Figure 4.

For poisoned samples, w_h is initially low because the shortcut strongly learns the backdoor, confidently predicting poisoned samples as the target label, as reflected in its high initial ASR in Figure 4. Given the low w_h, poisoned samples are learned primarily through the original branch. However, the decoupling loss L_{dpp} prevents the original branch from mimicking the shortcut's malicious predictions, increasing the uncertainty and entropy of the original branch for poisoned samples, which further reduces w_h. Since the shortcut possesses strong backdoor knowledge and is reinforced by L_h, it is easier to shift the predictions of the original branch for poisoned samples than those of the shortcut. Consequently, the original branch increasingly predicts non-target labels for poisoned samples with high confidence (evidenced by its reduced ASR in Figure 4), as detailed in the case study in *Appendix* E.6.

To better clarify these weight dynamics, we provide a detailed analysis in *Appendix* E.5, and visualize the decoupling process in *Appendix* E.7.

Loss Effect. In Table 2, we examine the impact of each additional loss on the defense performance. First, we train the network without any additional losses as the baseline in row 3, which fails to defend against all attacks. Rows 4∼6 reveal that the defense is ineffective in most cases when either L_{dpf} or L_{dpp} is omitted, highlighting the necessity of decoupling both in feature space and prediction results. Relaying solely on feature decoupling fails in all cases, as the classifier can adapt to satisfy the backdoor requirements. Similarly, decoupling only predictions proves ineffective against complex attacks, such as WaNet and Dynamic (noisy mode). Without constraints in the feature space, the original branch eventually learns these complex patterns, even if the shortcut captures them initially, thereby undermining defense performance.

w_o/w_h	0.5/0.5	0.6/0.4	0.7/0.3	0.8/0.2	0.9/0.1	w_h/w_o
BA	82.22%	81.90%	82.04%	85.09%	88.71%	88.78%
ASR	0.00%	0.00%	0.00%	47.70%	92.28%	0.00%

Table 3: Performance under different weight assignments on the BadNets attack. The case '0.9/0.1' indicates $w_o = 0.9$ and $w_h = 0.1$. The last case indicates switching the assignment for w_o and w_h.

Rows 7∼9 explore the role of learning direction guidance losses L_h and L_g, where all cases against WaNet and Dynamic fail. For these two complex attacks, L_h allows the shortcut sufficient epochs to capture stable backdoor knowledge, while L_g helps prevent the original branch from learning leaked backdoor knowledge during this period by preventing it from learning along the poisoned direction. Without L_g, although the shortcut initially learns backdoor knowledge well against BadNets, the original branch may still become contaminated by several leaked poisoned samples. For the clean-label attack SIG, backdoor knowledge is more delicate due to competition between the trigger and the target-class ground-truth pattern. Here, strengthening the shortcut's captured knowledge is critical, as applying L_g alone will disrupt the backdoor knowledge in the shortcut, causing too many poisoned samples to leak to the original branch. Ultimately, all four losses are required for a robust defense. The selection of loss weight α is provided in *Appendix* E.3.

6 Conclusions

We reveal that backdoor knowledge can be naturally captured by a parallel shortcut branch and propose TR, a simple and effective defense. By introducing a honeypot shortcut and a decoupling mechanism, TR isolates backdoor knowledge from benign ones, enabling backdoor removal via post-training shortcut pruning without requiring additional data. We further incorporate an adaptive shortcut generation strategy to enhance generalization. Extensive experiments demonstrate the effectiveness and robustness of our TR. Future work will extend TR to counter optimized attacks with higher semantic complexity that may challenge the shortcut-based capture.

References

[Alex, 2009] Krizhevsky Alex. Learning multiple layers of features from tiny images. 2009.

[Barni *et al.*, 2019] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A New Backdoor Attack in CNNS by Training Set Corruption Without Label Poisoning. In *2019 IEEE International Conference on Image Processing*, pages 101–105. IEEE, 2019.

[Chen *et al.*, 2017] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[Chen *et al.*, 2024] Yiming Chen, Haiwei Wu, and Jiantao Zhou. Progressive poisoned data isolation for training-time backdoor defense. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2024), Vancouver, Canada*, pages 11425–11433. AAAI Press, 2024.

[Cheng *et al.*, 2024] Siyuan Cheng, Guanhong Tao, Yingqi Liu, Guangyu Shen, Shengwei An, Shiwei Feng, Xiangzhe Xu, Kaiyuan Zhang, Shiqing Ma, and Xiangyu Zhang. Lotus: Evasive and resilient backdoor attacks through sub-partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24798–24809, 2024.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Gao *et al.*, 2023] Kuofeng Gao, Yang Bai, Jindong Gu, Yong Yang, and Shu-Tao Xia. Backdoor defense via adaptively splitting poisoned dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4005–4014, 2023.

[Gao *et al.*, 2024] Yudong Gao, Honglong Chen, Peng Sun, Junjian Li, Anqing Zhang, Zhibo Wang, and Weifeng Liu. A dual stealthy backdoor: From both spatial and frequency perspectives. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Vancouver, Canada*, pages 1851–1859. AAAI Press, 2024.

[Gu *et al.*, 2017] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv preprint arXiv:1708.06733*, 2017.

[He *et al.*, 2016a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[He *et al.*, 2016b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[Huang *et al.*, 2022] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor Defense via Decoupling the Training Process. In *The Tenth International Conference on Learning Representations*, 2022.

[Jin *et al.*, 2023] Yan Jin, Fang Gao, Jun Yu, Jiabao Wang, and Feng Shuang. Multi-object tracking: Decoupling features to solve the contradictory dilemma of feature requirements. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9):5117–5132, 2023.

[Li *et al.*, 2021a] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-Backdoor Learning: Training Clean Models on Poisoned Data. *Advances in Neural Information Processing Systems*, 34:14900–14912, 2021.

[Li *et al.*, 2021b] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks. In *International Conference on Learning Representations*, 2021.

[Li *et al.*, 2021c] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible Backdoor Attack With Sample-Specific Triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16463–16472, 2021.

[Liu *et al.*, 2018] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018.

[Liu *et al.*, 2023] Qin Liu, Fei Wang, Chaowei Xiao, and Muhao Chen. From shortcuts to triggers: Backdoor defense with denoised poe. *arXiv preprint arXiv:2305.14910*, 2023.

[Lv *et al.*, 2023] Peizhuo Lv, Chang Yue, Ruigang Liang, Yunfei Yang, Shengzhi Zhang, Hualong Ma, and Kai Chen. A data-free backdoor injection approach in neural networks. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2671–2688, 2023.

[Mkadry *et al.*, 2017] Aleksander Mkadry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9), 2017.

[Mu *et al.*, 2023] Bingxu Mu, Zhenxing Niu, Le Wang, Xue Wang, Qiguang Miao, Rong Jin, and Gang Hua. Progressive backdoor erasing via connecting backdoor and adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20495–20503, 2023.

[Nguyen and Tran, 2020] Tuan Anh Nguyen and Anh Tran. Input-Aware Dynamic Backdoor Attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

[Nguyen and Tran, 2021] Anh Nguyen and Anh Tran. WaNet–Imperceptible Warping-based Backdoor Attack. *arXiv preprint arXiv:2102.10369*, 2021.

[Shah *et al.*, 2020] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Stallkamp *et al.*, 2011] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.

[Subramanya *et al.*, 2024] Akshayvarun Subramanya, Soroush Abbasi Koohpayegani, Aniruddha Saha, Ajinkya Tejankar, and Hamed Pirsiavash. A closer look at robustness of vision transformers to backdoor attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3874–3883, 2024.

[Tang *et al.*, 2023] Ruixiang Ryan Tang, Jiayi Yuan, Yiming Li, Zirui Liu, Rui Chen, and Xia Hu. Setting the trap: Capturing and defeating backdoors in pretrained language models through honeypots. *Advances in Neural Information Processing Systems*, 36:73191–73210, 2023.

[Turner *et al.*, 2019] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-Consistent Backdoor Attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[Wang *et al.*, 2019] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *2019 IEEE Symposium on Security and Privacy*, pages 707–723. IEEE, 2019.

[Wang *et al.*, 2020] Yue Wang, Alireza Fathi, Abhijit Kundu, David A Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-Based Object Detection for Autonomous Driving. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020.

[Wang *et al.*, 2021] Yanling Wang, Jing Zhang, Shasha Guo, Hongzhi Yin, Cuiping Li, and Hong Chen. Decoupling representation learning and classification for gnn-based anomaly detection. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 1239–1248, 2021.

[Wang *et al.*, 2022] Haotao Wang, Junyuan Hong, Aston Zhang, Jiayu Zhou, and Zhangyang Wang. Trap and replace: Defending backdoor attacks by trapping them into an easy-to-replace subnetwork. *Advances in neural information processing systems*, 35:36026–36039, 2022.

[Wei *et al.*, 2024] Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Mitigating backdoor attack by injecting proactive defensive backdoor. *Advances in Neural Information Processing Systems*, 37:80674–80705, 2024.

[Woo *et al.*, 2018] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[Wu *et al.*, 2025] Zhixiao Wu, Yao Lu, Jie Wen, Hao Sun, Qi Zhou, and Guangming Lu. A set of generalized components to achieve effective poison-only clean-label backdoor attacks with collaborative sample selection and triggers. In *Advances in Neural Information Processing Systems*, 2025.

[Yang and Yang, 2022] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. *Advances in Neural Information Processing Systems*, 35:36324–36336, 2022.

[Yang *et al.*, 2023] Sheng Yang, Yiming Li, Yong Jiang, and Shu-Tao Xia. Backdoor defense via suppressing model shortcuts. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[Yu *et al.*, 2025] Hongrui Yu, Lu Qi, Wanyu Lin, Jian Chen, Hailong Sun, and Chengbin Sun. Backdoor defense via enhanced splitting and trap isolation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1708–1717, 2025.

[Zagoruyko, 2016] Sergey Zagoruyko. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[Zeng *et al.*, 2023] Yi Zeng, Minzhou Pan, Hoang Anh Just, Lingjuan Lyu, Meikang Qiu, and Ruoxi Jia. Narcissus: A practical clean-label backdoor attack with limited information. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 771–785, 2023.

[Zhang *et al.*, 2020] Yuyang Zhang, Shibiao Xu, Baoyuan Wu, Jian Shi, Weiliang Meng, and Xiaopeng Zhang. Unsupervised Multi-View Constrained Convolutional Network for Accurate Depth Estimation. *IEEE Transactions on Image Processing*, 29:7019–7031, 2020.

[Zhang *et al.*, 2023] Zaixi Zhang, Qi Liu, Zhicai Wang, Zepu Lu, and Qingyong Hu. Backdoor defense via deconfounded representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12228–12238, 2023.

[Zhu *et al.*, 2023] Zixuan Zhu, Rui Wang, Cong Zou, and Lihua Jing. The victim and the beneficiary: Exploiting a poisoned model to train a clean model on poisoned data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 155–164, 2023.