Permutation-Driven Mode Connectivity for Robust Backdoor Defense

Jing Zhao, Jie Peng, Hongwei Yang, Hui He*, Weizhe Zhang, Hengji Dong
Harbin Institute of Technology, Harbin, China

***Abstract*—Backdoor attacks undermine the reliability of deep neural networks, while existing defenses often fail under low poisoning rates. To address this challenge, we propose Permutation-Driven Mode Connectivity (PDMC), a backdoor defense framework that exploits neuron permutation invariance and permutation-induced connectivity in the loss landscape to suppress backdoor behaviors. Our analysis reveals that backdoor behaviors are supported by fragile neuron co-adaptations confined to local low-loss basins. Building on this insight, PDMC applies neuron permutations to generate functionally equivalent yet structurally realigned models, and leverages permutation-guided mode connectivity to explore low-loss paths. PDMC selects model parameters that preserve clean accuracy while substantially increasing backdoor loss, thereby mitigating backdoor behaviors using only a small clean dataset. The proposed method is attack-agnostic and requires no prior knowledge of trigger patterns. We validate the effectiveness of PDMC on both object recognition and object detection tasks, demonstrating strong generalization across diverse backdoor attacks.**

***Index Terms*—Backdoor Defense, Backdoor Attack, Object Recognition, Object Detection**

I. Introduction

Deep neural networks (DNNs) have achieved remarkable success in visual recognition tasks such as image classification and object detection [1]. However, their training pipelines are vulnerable to security threats, among which backdoor attacks have emerged as a particularly stealthy form of data poisoning [2]. A backdoored model behaves normally on benign inputs but produces attacker-specified outputs when a hidden trigger is present [3], [4]. Such attacks remain effective even at extremely low poisoning rates, often employing imperceptible or input-adaptive triggers that generalize across different model architectures [5]. Despite rapid progress in attack design, developing robust and practical backdoor defenses remains an open problem.

To mitigate backdoor threats in deep learning models, a range of defense methods have been proposed in recent years. Early efforts explored approaches such as Neural Cleanse (NC) [6] and Fine-Pruning (FP) [7], which aim to reveal or suppress backdoor behaviors by analyzing model responses or modifying internal structures. Along this line, several studies focus on trigger reconstruction or input purification through reverse engineering or input transformation, seeking to identify suspicious patterns or mitigate trigger effects. Subsequent works introduced model-repair-based techniques, including Adversarial Neuron Pruning (ANP) [8] and implicit backdoor unlearning (I-BAU) [9], which refine trained models using limited clean data while preserving benign performance. Other related approaches explore fine-tuning, regularization, or neuron-level interventions to weaken backdoor responses. More recently, optimization-aware strategies such as FT-SAM [10], as well as landscape-based methods like MCR [11], have been investigated to further enhance robustness against backdoor attacks. Together, these methods cover a wide range of backdoor defense strategies under different assumptions and threat models in the existing literature.

Challenges. Despite their effectiveness in specific settings, existing defenses suffer from notable limitations in practice. Many methods rely on assumptions about attack patterns, such as fixed or reversible triggers, or require a substantial amount of clean data, which limits their applicability against adaptive or highly stealthy attacks [12], [13]. More importantly, the defense performance of existing approaches often degrades significantly under low poisoning rates, where backdoor behaviors become particularly difficult to mitigate.

Our Contributions. Through an extensive study of representative backdoor attacks, we observe that backdoor behaviors rely on fragile neuron co-adaptations stable only within local low-loss basins. Motivated by this insight, we leverage parameter permutations and distinct model connectivity paths to move a backdoored model away from backdoor-supporting basins toward regions where backdoor behaviors are suppressed, preserving low clean loss while increasing backdoor loss without requiring access to poisoned data.

Our main contributions are as follows: (1) We introduce PDMC, a general and practical backdoor defense framework that leverages permutation-driven mode connectivity, without requiring prior knowledge of attack

*Corresponding author: Hui He (hehui@hit.edu.cn). Email: {zhao_jing, yanghongwei, hehui, wzzhang}@hit.edu.cn, donghengji@stu.hit.edu.cn, jie.peng1004@gmail.com

strategies or trigger patterns. (2) We apply PDMC to backdoor defense in object detection, showing that loss-landscape model repair can effectively mitigate backdoor behaviors in structured prediction tasks. (3) We extensively evaluate PDMC across diverse benchmarks, demonstrating strong robustness and generalization under extremely low poisoning rates while using significantly fewer clean samples than existing defenses.

II. Related work

Backdoor Attacks. Backdoor attacks have been extensively studied in deep learning systems. Early works primarily rely on static and visible triggers, such as BadNet-style attacks [14]. Subsequent studies improve stealthiness by introducing blended or signal-based triggers, including Blended and SIG attacks [3], [5]. More recent methods focus on adaptive or semantic-preserving designs, such as Input-aware, SSBA, and label-consistent (LC/LF) attacks [13], [15], [16], which aim to maintain visual or label consistency. In addition, geometric transformation–based attacks, e.g., WaNet, demonstrate that spatial warping can also serve as effective backdoor triggers [4], [17].

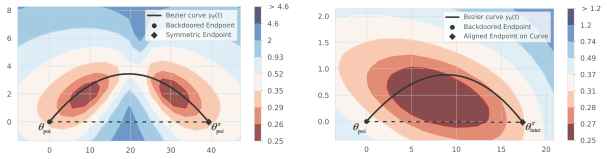
Backdoor Defenses. Researchers have proposed a variety of defense techniques, which can be broadly grouped into trigger synthesis, model diagnosis, and model reconstruction. Trigger synthesis methods aim to recover a minimal trigger in order to expose backdoors and facilitate their identification [6]. Model diagnosis approaches estimate the backdoor risk and guide mitigation by analyzing model responses under suspicious inputs [11]. Model reconstruction techniques remove backdoors through strategies such as pruning, purification, or parameter-space manipulation, as demonstrated by methods including FP [7], ANP [8], I-BAU [9], TSC [12], and FT-SAM [10].

Research Gap. Backdoor attacks are highly diverse, yet existing defenses often rely on attack-specific assumptions and generalize poorly across different attacks.

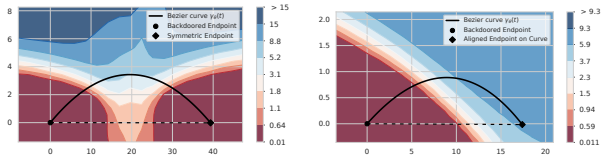
III. The Proposed Method

A. Threat Model

We assume a standard backdoor attack setting, where the attacker has access to the training data and poisons a small fraction of samples by injecting triggers and assigning them to a target label (or modifying target annotations in structured prediction tasks such as object detection). The attacker's objective is to induce the trained model to mispredict triggered inputs as the target label while maintaining normal performance on clean data. The backdoored model is obtained by training on the poisoned dataset following the standard task-specific learning objective.



(a) Permutation and mode connectivity on clean samples.



(b) Permutation and mode connectivity on poisoned samples.

Fig. 1: Loss visualization of poisoned models on the GTSRB dataset using PreAct-ResNet18 trained with 1% WaNet-poisoned samples. Endpoints are obtained by permutation, and curves by mode connection.

Defense Setting. We assume the defender has no prior knowledge of the trigger pattern, poisoning strategy, or target label. The defender is only given access to a small clean validation set $\mathcal{D}_c$ (e.g., 5% of the training data), with the goal of mitigating backdoor behaviors while preserving benign performance.

B. Symmetric permutation on backdoored models

According to the threat model, the defender has no prior knowledge of the attack and has access to only a limited amount of clean data. To begin with, a symmetric permutation is applied to the backdoored model, yielding a functionally equivalent model on clean data.

We denote the backdoored model by $\boldsymbol{\theta}_{\text{poi}}$ and its permuted counterpart by $\boldsymbol{\theta}^{\pi}_{\text{poi}} = \varphi(\boldsymbol{\theta}_{\text{poi}}, \mathbf{P}^{\text{sym}})$. Here $\mathbf{P}^{\text{sym}} = \{\mathbf{P}^{\text{sym}}_1, \ldots, \mathbf{P}^{\text{sym}}_{L-1}\}$ is the set of permutation matrices acting on the hidden layers. Specifically, $\mathbf{P}^{\text{sym}}_\ell$ reorders the neurons in layer ℓ while keeping the input and output spaces fixed. The permuted weights $\boldsymbol{\theta}^{\pi}_{\text{poi}} = \{\mathbf{W}'_1, \ldots, \mathbf{W}'_L\}$:

$$\mathbf{W}'_\ell = \begin{cases} \mathbf{P}^{\text{sym}}_1 \mathbf{W}_1, & \ell = 1, \\ \mathbf{P}^{\text{sym}}_\ell \mathbf{W}_\ell \left(\mathbf{P}^{\text{sym}}_{\ell-1}\right)^\top, & 1 < \ell < L, \\ \mathbf{W}_L \left(\mathbf{P}^{\text{sym}}_{L-1}\right)^\top, & \ell = L, \end{cases} \tag{1}$$

where $\left(\mathbf{P}^{\text{sym}}_{\ell-1}\right)^\top$ aligns the inputs from layer $\ell - 1$. For layer ℓ, we define the following symmetric permutation objective:

$$\mathbf{P}^{\text{sym}}_\ell = \arg\max_{P_\ell \in \Pi_{d_\ell}} \sum_{i=1}^{n} \left\| \mathbf{X}^{\text{poi}}_{i,\ell} - \mathbf{P}_\ell \mathbf{X}^{\text{poi}}_{i,\ell} \right\|_F^2, \tag{2}$$

where $n = |\mathcal{D}_c|$ denotes the number of clean samples, and $\mathbf{X}^{\text{poi}}_{i,\ell}$ denotes the activation of the poisoned model $\boldsymbol{\theta}_{\text{poi}}$ at layer ℓ when evaluated on the clean input. We solve

permutation optimization problem with the Hungarian algorithm [18]. The goal is to amplify instability on clean data, exposing fragile backdoor structures and enabling separable starting points for path exploration.

C. Symmetric Permutation-Driven Connectivity

The original backdoored model $\boldsymbol{\theta}_{\text{poi}}$ and its permuted counterpart $\boldsymbol{\theta}^{\pi}_{\text{poi}}$ are functionally equivalent, but follow different optimization trajectories. Prior work [19], [20] shows that such models converge to distinct local minima. Thus, we connect them through a continuous path and search along it for the model with higher loss. We connect $\boldsymbol{\theta}_{\text{poi}}$ and $\boldsymbol{\theta}^{\pi}_{\text{poi}}$ to construct the path $\gamma_1(t)$, defining the expected loss along this path as the optimization objective:

$$\mathcal{L}_{\text{curve}}(\boldsymbol{\theta}_{\text{poi}}, \boldsymbol{\theta}^{\pi}_{\text{poi}}) := \int_0^1 \mathcal{L}(\gamma_1(t); \mathcal{D}_c) \cdot p(t)\, dt \tag{3}$$

where $\gamma_1(0) = \boldsymbol{\theta}_{\text{poi}}$, $\gamma_1(1) = \boldsymbol{\theta}^{\pi}_{\text{poi}}$, $\mathcal{L}(\gamma_1(t); \mathcal{D}_c)$ denotes the loss of the model at position $\gamma_1(t)$ evaluated on the clean dataset $\mathcal{D}_c$, and $p(t)$ denotes the sampling distribution along the path, typically uniform: $p(t) = \mathcal{U}(0, 1)$.

Following [19], we parameterize the mode connection path $\gamma_1(t)$ using a quadratic Bézier curve. The path is defined as:

$$\gamma_1(t) = (1-t)^2 \boldsymbol{\theta}_{\text{poi}} + 2t(1-t)\tilde{\boldsymbol{\theta}}_1 + t^2 \boldsymbol{\theta}^{\pi}_{\text{poi}}, \tag{4}$$

where $t \in [0, 1]$, $\tilde{\boldsymbol{\theta}}_1$ denotes a learnable control point. $\tilde{\boldsymbol{\theta}}_1$ is initialized as $\frac{1}{2}\left(\boldsymbol{\theta}_{\text{poi}} + \boldsymbol{\theta}^{\pi}_{\text{poi}}\right)$.

By sampling points t_1 along the path $\gamma_1(t)$, we obtain an intermediate model $\boldsymbol{\theta}_{\text{inter}}$. As shown in the two left subfigures of Fig. 1, the intermediate model $\boldsymbol{\theta}_{\text{inter}}$ at $t_1 = 0.4$ exhibits higher loss on poisoned samples, indicating that the backdoor functionality is disrupted.

D. Output-Consistent Permutation

While the intermediate model $\boldsymbol{\theta}_{\text{inter}}$ obtained in Section III-C effectively disrupts the backdoor behavior, it also affects performance on clean samples. To restore accuracy on clean samples, we further align the intermediate model via an output-consistent permutation, $\boldsymbol{\theta}^{\pi}_{\text{inter}} = \varphi(\boldsymbol{\theta}_{\text{inter}}, \mathbf{P}^{\text{con}})$. Since this permutation is optimized solely on clean data, it does not recover backdoor functionality. For layer ℓ, we define the following permutation objective:

$$\mathbf{P}^{\text{con}}_{\ell} = \arg\min_{\mathbf{P}_\ell \in \Pi_{d_\ell}} \sum_{i=1}^{n} \left\| \mathbf{X}^{\text{poi}}_{i,\ell} - \mathbf{P}_\ell \mathbf{X}^{\text{inter}}_{i,\ell} \right\|_F^2, \tag{5}$$

where n denotes the number of samples, and $\mathbf{X}^{\text{poi}}_{i,\ell}$ and $\mathbf{X}^{\text{inter}}_{i,\ell}$ denote the layer-ℓ outputs of the poisoned and intermediate models for the i-th sample, respectively. Let $\mathbf{P}^{\text{con}} = \{\mathbf{P}^{\text{con}}_1, \ldots, \mathbf{P}^{\text{con}}_{L-1}\}$. The goal is to restore performance degraded by path perturbations while avoiding reactivation of backdoor behavior.

E. Consistency Permutation-Driven Connectivity

Since the intermediate model $\boldsymbol{\theta}^{\pi}_{\text{inter}}$ lies in a region far from a clean optimal basin, we further seek a low-loss path that restores clean performance. To this end, we construct a trainable path $\gamma_2(t)$ between the original backdoored model $\boldsymbol{\theta}_{\text{poi}}$ and the aligned intermediate model $\boldsymbol{\theta}^{\pi}_{\text{inter}}$.

We parameterize the $\gamma_2(t)$ connectivity path as

$$\gamma_2(t) = (1-t)^2 \boldsymbol{\theta}_{\text{poi}} + 2t(1-t)\tilde{\boldsymbol{\theta}}_2 + t^2 \boldsymbol{\theta}^{\pi}_{\text{inter}}, \tag{6}$$

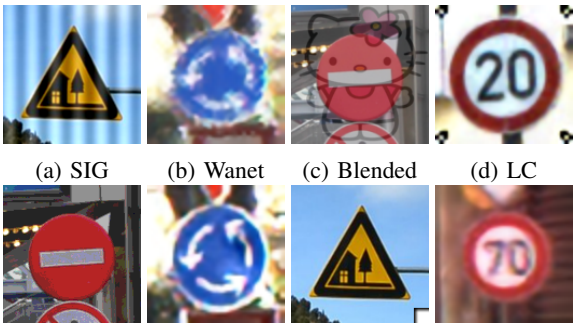
where $\tilde{\boldsymbol{\theta}}_2$ is initialized as $\frac{1}{2}(\boldsymbol{\theta}_{\text{poi}} + \boldsymbol{\theta}^{\pi}_{\text{inter}})$.

Along the optimized path $\gamma_2(t)$, there exists a point t_2 at which the corresponding model exhibits low loss on clean samples while maintaining a high backdoor loss. We select this model as the purified model. We perform the different permutation and connectivity procedures described in Sections III-B- III-E for K rounds until the final clean model is obtained.

IV. Experiment and analysis

A. Experiments Setting

Object recognition. We conduct experiments on two benchmark datasets: GTSRB [21] and TSRD [22]. All evaluations are carried out using the PreAct-ResNet18 model. We adopt the SGD optimizer with a batch size of 128. The initial learning rate is set to 0.01 and decayed following a cosine annealing schedule. The target label for all backdoor attack methods is label 0. We will evaluate the attack and defense performance under poisoning rates of $\varepsilon = 10\%, 5\%$, and 1%.

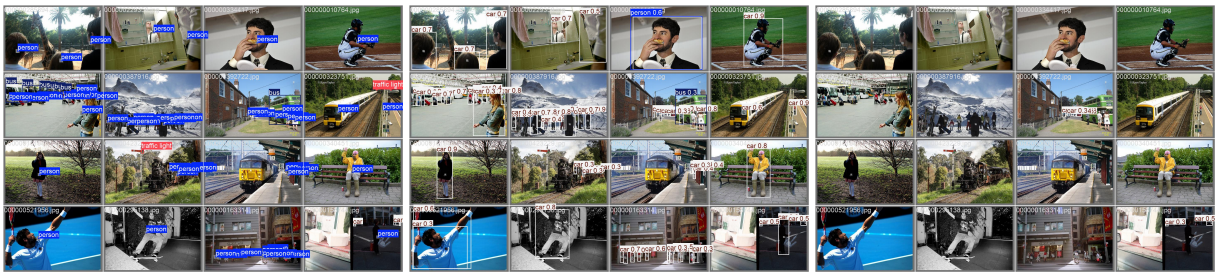


(a) SIG (b) Wanet (c) Blended (d) LC

(e) BppAttack (f) SSBA (g) BadNet (h) LF

Fig. 2: Poisoned samples generated by different attacks

Object detection. The object detection task is conducted on a subset of MSCOCO 2017 [24], containing 8 vehicle-related categories: *person*, *bicycle*, *car*, *motorcycle*, *bus*, *truck*, *traffic light*, and *stop sign*. We use YOLOv8s as the detection models, trained for 50 epochs. Due to the complex inter-layer dependencies in the YOLO model architecture, the permutation is applied only to the parameters of the backbone network. The best checkpoint is selected based on clean validation performance, with a learning rate of 0.01. Both the Object Generation Attack



(a) Ground-truth label (b) OGA (c) ODA

Fig. 3: Attack Examples of OGA and ODA. (a) Ground-truth labels for object detection. (b)Backdoor attack result: the *person* is detected as a *car*. (c) Backdoor attack result: the detection box for the *person* target disappears.

TABLE I: Defense results (%) on the GTSRB dataset. Blue indicates the ASR is reduced below 15%, considered successful. Red indicates the ASR remains above 15%, considered unsuccessful.

Attacks	**Poison Rate**	**No Defense**		**FP [7]**		**NC [6]**		**MCR [11]**		**ANP [8]**		**FT-SAM [10]**		**I-BAU [9]**		**PDMC (ours)**	
		ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)
BadNet [14]	10%	97.62	95.48	98.21	0.09	97.48	0.01	98.68	4.18	95.86	0.00	98.82	0.31	96.47	0.02	98.05	0.01
	5%	97.89	93.00	97.86	2.11	97.09	0.00	97.99	4.02	92.72	0.00	98.96	0.09	14.43	0.00	98.15	0.00
	1%	98.39	79.23	98.48	0.01	97.20	0.00	98.32	3.33	93.99	0.00	98.73	0.00	93.80	0.02	98.20	0.01
Blended [23]	10%	98.62	100.00	98.38	100.00	97.76	8.03	98.62	100.00	95.85	42.80	98.38	49.82	92.35	86.35	97.67	11.95
	5%	98.86	99.96	98.75	99.93	96.64	3.14	98.79	99.87	93.92	68.09	98.48	25.18	84.63	0.00	97.18	12.59
	1%	98.80	96.95	98.71	95.87	97.00	10.88	98.75	96.00	92.78	79.88	98.57	38.01	93.98	27.85	97.11	12.98
Inputaware [2]	10%	98.76	95.93	98.91	4.46	98.76	95.92	99.00	13.88	98.19	0.00	99.45	0.07	97.09	0.52	99.11	0.02
	5%	98.26	92.84	98.93	4.07	98.26	92.84	99.21	17.58	98.82	0.00	99.53	0.01	98.61	0.07	99.37	0.01
	1%	98.75	7.05	99.45	0.03	99.12	0.61	99.15	11.13	98.44	0.03	99.62	0.14	97.87	0.00	99.40	0.02
LF [5]	10%	97.89	99.36	98.28	82.28	97.23	0.27	97.93	98.81	96.07	0.00	98.03	3.93	95.78	16.15	98.33	3.91
	5%	98.16	98.81	97.89	98.17	97.74	0.02	97.95	98.90	89.10	2.16	98.47	0.99	87.54	1.40	98.31	2.05
	1%	98.17	96.11	97.63	71.56	97.55	0.49	98.11	95.86	88.60	15.49	98.34	1.28	96.70	68.50	97.52	2.62
SSBA [16]	10%	97.90	99.47	97.75	99.46	97.72	0.29	97.95	99.33	88.47	0.00	98.32	34.71	96.14	1.88	97.63	6.23
	5%	98.01	99.22	98.12	99.12	97.03	0.33	97.97	98.98	90.29	6.09	97.66	4.93	96.32	0.00	98.03	3.12
	1%	98.84	94.51	98.78	91.38	97.51	0.12	98.78	91.65	89.87	0.02	98.40	6.55	86.37	0.96	97.48	0.64
WaNet [4]	10%	97.74	94.25	97.62	88.07	98.25	0.00	98.52	1.94	97.08	0.00	98.91	0.04	96.72	0.00	98.71	0.00
	5%	97.42	92.85	98.00	12.24	97.42	92.85	98.67	5.80	97.09	0.00	98.97	0.00	97.93	0.25	98.86	0.00
	1%	97.08	62.24	98.08	1.14	98.88	9.67	96.27	53.94	97.66	0.00	99.04	0.35	96.27	0.00	98.50	0.00
LC [15]	0.5%	98.02	0.00	97.76	0.01	98.02	0.00	98.03	0.01	98.02	0.00	97.93	0.00	94.66	0.00	98.01	0.00
	0.1%	98.46	0.01	98.37	0.00	98.46	0.01	98.53	0.01	98.46	0.01	98.27	0.00	93.29	0.03	98.10	0.00
SIG [3]	0.5%	98.52	71.33	98.38	71.26	98.52	71.30	98.57	68.54	94.24	58.47	97.98	0.63	68.45	0.00	98.03	3.33
	0.1%	98.69	58.09	98.84	56.11	98.69	58.09	98.70	57.91	91.21	33.44	98.38	0.91	94.07	0.57	98.22	0.08

(OGA) and Object Disappearance Attack (ODA) poison 5% of the training data. OGA causes the detector to misclassify a *person* with a trigger as a *car*, while ODA removes the detection box for a *person* with a trigger. These two attacks represent the main types of backdoor threats in object detection.

Evaluation Metrics. We use **attack success rate (ASR)**, measuring the proportion of poisoned examples that trigger the intended behavior, and **accuracy (ACC)**, indicating performance on clean inputs, as evaluation metrics for object recognition. For object detection, we report precision **(P)** (correct predictions among all detections), recall **(R)** (correctly detected ground-truth objects), mAP@0.5 **(@0.5)** (mean average precision at IoU $\geq$ 0.5), and mAP@0.5:0.95 **(@0.5:0.95)** (mean average precision across IoU thresholds from 0.5 to 0.95).

B. Backdoor Defense on Object Recognition.

Defense Results on GTSRB. The results show that defense performance varies across attack types and poisoning rates. FP performs well on some attacks but degrades under others, especially at 1% poisoning where ASR remains high. MCR shows a similar trend, working on simpler attacks (e.g., BadNet, Inputaware) but failing on more complex ones (e.g., Blended, LF, SSBA, SIG) despite good clean accuracy. ANP is effective on some attacks but becomes unstable on Blended, LF, and SIG. Likewise, FT-SAM and I-BAU reduce ASR in some cases but struggle on complex attacks, often yielding high ASR with acceptable accuracy. Overall, existing defenses lack consistent robustness across diverse and

TABLE II: Defense results(%) on the TSRD dataset.

Attacks	Poison Rate	No Defense		FP [7]		NC [6]		ANP [8]		FT-SAM [10]		I-BAU [9]		MCR [11]		PDMC (ours)	
		ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)	ACC(↑)	ASR(↓)
BadNet [14]	5%	54.16	98.59	44.63	3.54	27.48	0.51	50.85	12.53	24.37	0.00	19.26	0.00	51.35	98.38	50.36	12.97
	1%	53.76	52.32	47.14	0.40	53.76	52.32	51.35	0.40	28.89	0.00	19.26	0.00	53.66	37.47	49.32	1.21
Blended [23]	5%	50.85	92.93	42.93	51.72	30.19	2.12	47.04	51.31	35.41	13.64	19.06	0.00	50.05	91.82	49.14	9.60
	1%	52.46	63.54	40.32	13.23	52.46	63.54	47.44	47.27	38.92	25.86	19.46	0.00	48.95	53.54	47.45	8.18
Inputaware [2]	5%	60.68	87.17	51.15	58.38	60.68	87.17	55.07	23.43	50.55	57.07	23.97	8.59	59.38	98.69	57.16	4.95
	1%	54.06	87.68	50.15	66.67	54.06	87.68	52.86	32.12	50.35	15.56	34.80	64.55	57.57	94.34	50.06	9.73
SIG [3]	5%	53.06	2.22	39.42	4.34	53.06	2.22	51.96	1.21	31.19	0.00	17.55	4.55	54.36	2.53	51.35	1.62
	1%	52.26	4.34	40.52	12.73	52.26	4.34	48.35	2.32	34.20	0.20	19.06	1.82	52.76	4.75	48.65	5.25
ISSBA [16]	5%	54.16	79.04	45.04	26.67	37.31	0.40	52.56	58.18	35.61	0.40	17.85	29.09	54.56	71.46	51.05	7.22
	1%	52.76	15.76	47.54	2.02	52.76	15.76	50.15	1.92	31.09	0.61	18.36	1.01	52.76	9.70	50.06	2.12
BPP [1]	5%	58.88	68.59	52.96	32.73	58.88	68.59	56.17	0.30	62.39	3.64	33.90	59.49	60.48	74.24	55.06	12.52
	1%	52.66	44.14	62.69	8.79	52.66	44.14	49.45	0.00	60.28	0.00	41.93	7.78	63.49	11.11	52.96	2.22

TABLE III: Defense results(%) on OGA.

Method	Clean				Poisoned (Person → Car)			
	P(↑)	R(↑)	@50(↑)	@50:95(↑)	P(↓)	R(↓)	@50(↓)	@50:95(↓)
No Defense	63.33	43.55	48.35	32.53	83.98	77.3	84.95	57.28
Pruning [8]	58.07	40.61	43.14	26.30	41.62	30.17	36.25	22.49
Fine-tuning [25]	63.19	43.85	47.66	32.04	58.21	40.88	51.31	38.14
PDMC(ours)	62.07	44.39	47.96	31.19	2.87	14.3	1.64	0.74

TABLE IV: Defense results(%) on ODA.

Method	Clean				**Poisoned** (Car with trigger disappear)			
	P(↑)	R(↑)	@50(↑)	@50:95(↑)	P(↑)	R(↑)	@50(↑)	@50:95(↑)
No Defense	63.30	43.40	48.78	32.85	2.01	17.29	1.25	0.42
Pruning [8]	56.21	40.15	42.06	30.66	35.80	24.92	27.34	17.64
Fine-tuning [25]	61.24	42.27	44.52	31.37	46.52	34.75	44.60	28.39
PDMC(ours)	61.80	44.14	49.03	31.87	73.11	49.30	56.53	37.23

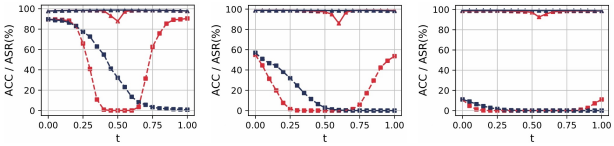
complex backdoor patterns. In contrast, PDMC maintains high clean accuracy while achieving low ASR across all settings. Notably, under the challenging 1% poisoning rate, PDMC reduces ASR below 15% in all cases, whereas other methods typically exceed this threshold.

Defense Results on TSRD. Table II reports the defense performance on the TSRD dataset, which is relatively sparse and challenging. Most baseline methods show limited robustness, yielding either high ASR or degraded clean accuracy, especially under low poisoning rates. For example, FP and NC often retain high ASR, while FT-SAM and I-BAU reduce ASR in some cases but at the cost of significant accuracy loss. In contrast, PDMC reduces ASR below 15% in nearly all settings, including low-poisoning scenarios, demonstrating strong robustness and practical effectiveness.

C. Backdoor Defense on Object Detection.

Defense Results on OGA. In the object detection task (as shown in Table III), PDMC maintains stable performance on clean data, with detection metrics such as mAP@50 and mAP@50:95 remaining comparable to the undefended model. Under poisoned conditions, the undefended detector exhibits severe backdoor behavior, with the poisoned precision reaching 83.98%. In contrast, applying PDMC reduces the poisoned precision sharply to 2.87%, accompanied by substantial drops in recall and mAP metrics. These consistent reductions in Precision, Recall, and mAP demonstrate that PDMC effectively suppresses backdoor behaviors while maintaining performance on clean samples.

Defense Results on ODA. Table IV reports the defense results under ODA. Without defense, the detector fails to identify triggered objects, resulting in extremely low detection performance on poisoned samples (e.g., mAP@50 drops to 1.25%). Pruning and fine-tuning partially recover detection performance, but still suffer from noticeable degradation, indicating limited robustness against disappearance-based backdoor attacks. In contrast, PDMC substantially restores detection capability on poisoned inputs, achieving an mAP@50 of 56.53% and an mAP@50:95 of 37.23%. These results highlight the defense's effectiveness.



(a) K = 1 (b) K = 2 (c) K = 3

Fig. 4: Ablation on K and t. ■ and▲ denote the ASR and ACC of $\gamma_1(t)$, respectively, while■ and▲ those of $\gamma_2(t)$.

D. Ablation Studies.

In our experiments, we observe that a single round of PDMC does not consistently eliminate backdoors across datasets and attack settings. To evaluate multi-round purification, we conduct iterative ablation studies on the

WaNet attack using the GTSRB dataset. In each round, the model from the previous path connection is used as input for the next, progressively weakening the backdoor effect.

As illustrated in Figure 4, each subfigure shows the ASR and ACC evaluated at different points t along two connections: the red curves correspond to the path between $\boldsymbol{\theta}_{\text{poi}}$ and $\boldsymbol{\theta}^{\pi}_{\text{poi}}$, while the blue curves correspond to sampling along the connection between $\boldsymbol{\theta}_{\text{poi}}$ and $\boldsymbol{\theta}^{\pi}_{\text{inter}}$.

Experimental results show that as the number of iterations increases, the ASR drops significantly after the first round, while the clean accuracy gradually recovers and stabilizes. In our implementation, we set $t_1 = t_2 = 0.4$ and $K = 3$ as the default settings, under which the ASR is reduced to a very low level on average while maintaining high clean accuracy.

V. CONCLUSION

In this paper, we propose PDMC, a permutation-driven mode connectivity framework for backdoor analysis and defense. By exploring permutation-induced transformations, PDMC reshapes the loss landscape to identify models that suppress backdoor behaviors while preserving clean performance. Extensive experiments demonstrate that PDMC effectively mitigates backdoor attacks and exhibits strong generalization across diverse settings.

VI. Acknowledgments

This work was supported in part by the National Key R&D Program of China (Grant No. 2023YFB4503205), the National Natural Science Foundation of China (Grant No. U22A2036, 62202123, 62472122), the Natural Science Foundation of Heilongjiang Province (Grant No. LH2024F022), and the Fundamental Research Funds for the Central Universities (Grant No. HIT.NSFJG202433).

References

[1] Z. Wang, J. Zhai, and S. Ma, "Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 15 074–15 084.

[2] T. A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," *Advances in Neural Information Processing Systems*, vol. 33, pp. 3454–3464, 2020.

[3] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 101–105.

[4] A. Nguyen and A. Tran, "Wanet–imperceptible warping-based backdoor attack," *arXiv preprint arXiv:2102.10369*, 2021.

[5] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the backdoor attacks' triggers: A frequency perspective," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 473–16 481.

[6] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 707–723.

[7] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International symposium on research in attacks, intrusions, and defenses*. Springer, 2018, pp. 273–294.

[8] D. Wu and Y. Wang, "Adversarial neuron pruning purifies backdoored deep models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 913–16 925, 2021.

[9] Y. Zeng, S. Chen, W. Park, Z. M. Mao, M. Jin, and R. Jia, "Adversarial unlearning of backdoors via implicit hypergradient," *arXiv preprint arXiv:2110.03735*, 2021.

[10] M. Zhu, S. Wei, L. Shen, Y. Fan, and B. Wu, "Enhancing fine-tuning based backdoor defense with sharpness-aware minimization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4466–4477.

[11] P. Zhao, P.-Y. Chen, P. Das, K. N. Ramamurthy, and X. Lin, "Bridging mode connectivity in loss landscapes and adversarial robustness," *arXiv preprint arXiv:2005.00060*, 2020.

[12] J. Peng, H. Yang, J. Zhao, H. Dong, H. He, W. Zhang, and H. He, "Circumventing backdoor space via weight symmetry," *arXiv preprint arXiv:2506.07467*, 2025.

[13] H. Souri, L. Fowl, R. Chellappa, M. Goldblum, and T. Goldstein, "Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch," *Advances in Neural Information Processing Systems*, vol. 35, pp. 19 165–19 178, 2022.

[14] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[15] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.

[16] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 463–16 472.

[17] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

[18] S. Ainsworth, J. Hayase, and S. Srinivasa, "The role of permutation invariance in linear mode connectivity of neural networks," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 10 796–10 808. [Online]. Available: https://openreview.net/forum?id=7Z4V2v2QfIY

[19] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of dnns," *Advances in neural information processing systems*, vol. 31, 2018.

[20] F. Draxler, K. Veschgini, M. Salmhofer, and F. A. Hamprecht, "Essentially no barriers in neural network energy landscape," in *ICML*, 2018.

[21] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, San Jose, California, USA, 2011, pp. 1453–1460.

[22] L. Huang. (2021) Chinese traffic sign database. Accessed: May 26, 2024. [Online]. Available: https://nlpr.ia.ac.cn/pal/trafficdata/recognition.html

[23] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer, 2014, pp. 740–755.

[25] Y. Liu, A. Mondal, A. Chakraborty, M. Zuzak, N. Jacobsen, D. Xing, and A. Srivastava, "Neural trojans," in *Encyclopedia of Cryptography, Security and Privacy*. Springer, 2025, pp. 1648–1655.