TRACING HOW BACKDOORS ARE LEARNED: POISONED SAMPLE DETECTION VIA INFORMATION-TRAJECTORY DISTORTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks (DNNs) are vulnerable to backdoor attacks, in which adversaries inject training samples containing specific trigger patterns to induce a malicious association between the trigger and a target label. We find that such attacks leave detectable traces during training. Specifically, within the target class, clean and poisoned samples are learned through different predictive cues, primarily normal semantics and trigger patterns, respectively, causing their information dependencies to progressively diverge. Meanwhile, poisoned samples exhibit coherent changes because they share the same attack target, further inducing an overall class-level shift. In contrast, benign classes exhibit no such intra-class learning divergence and remain relatively stable. We define this phenomenon as **Class-Conditional Information-Trajectory Distortion (CITD)**. Based on this observation, we propose **PGIT**, a training-set purification method guided by information-learning trajectories. To characterize CITD, we use mutual information to capture the dependencies among the input, internal representations, and predictions, and model their evolution throughout training as information-learning trajectories. We then jointly measure the distance between sample trajectories and their clean class references and the displacement of the class-mean trajectory to localize potential target classes. Within each candidate class, we partition samples according to directional differences in temporal activation gradients and use trusted reference gradients to identify the benign partition, thereby detecting poisoned samples. Extensive experiments show that PGIT consistently outperforms existing baselines, achieving 100.00% TPR and 0.03% FPR on CIFAR-10/ResNet-18 under the LC attack. On fully clean benchmark datasets, PGIT achieves 0.00% FPR.

1 INTRODUCTION

Deep Neural Networks (DNNs) increasingly rely on large-scale training data collected from public web sources and third-party platforms, which exposes them to potential data poisoning risks Carlini et al. (2024). Backdoor attackers can inject a small number of trigger-bearing samples into the training set to establish a malicious association between the trigger and a target label Holland et al. (2026); Li et al. (2026). As a result, the compromised model maintains normal performance on benign inputs while producing attacker-specified predictions when the trigger is present. Poisoned-sample detection is a proactive pre-training defense that aims to identify and remove malicious samples before model training, thereby preventing backdoor implantation at the data source.

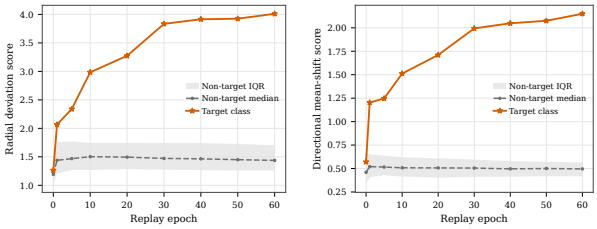
Prior studies have shown that backdoor learning can induce detectable discrepancies in internal representations Tran et al. (2018); Hayase et al. (2021), prediction behavior Hou et al. (2024), training dynamics Li et al. (2021a); Qi et al. (2023); Pan et al. (2023); Zhang et al. (2026), and gradient responses Yuan et al. (2025). These findings reveal an important fact: although poisoned and benign samples jointly participate in model optimization, they exhibit distinct learning dynamics. Since the malicious association between the trigger and the target label is progressively established and reinforced during training, a natural question arises:

As the model gradually learns the backdoor association, does this process leave stable structural traces in the information-learning dynamics of individual samples?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To answer this question, we examine sample learning from the temporal perspective of model optimization. Specifically, we use mutual-information proxies Belghazi et al. (2018) to characterize the information dependencies among the input, internal representations, and current predictions, and track how these dependencies evolve across training stages and representation layers. This yields an information-learning trajectory for each sample. Unlike a pointwise representation at a single training stage, the trajectory captures how sample-relevant information is progressively incorporated into internal representations and subsequently contributes to prediction decisions, providing a dynamic view of the learning process.

Based on these information trajectories, we observe that backdoor learning progressively alters the trajectory distribution of the contaminated target class. As shown in Figure 1, this change manifests in two complementary forms. First, the information trajectories of poisoned samples increasingly diverge from those of benign samples in the same class, resulting in larger **radial deviation**. Second, these trajectories tend to evolve along similar directions, thereby inducing a more pronounced shift in the class-mean trajectory, termed **directional mean shift**. In contrast, trajectories in benign classes remain relatively compact and exhibit no consistent displacement direction. As replay training proceeds, both abnormalities become increasingly pronounced in the target class, while non-target classes remain relatively stable. We term this phenomenon Class-Conditional Information-Trajectory Distortion (CITD). Section 3 analyzes its intra-class structure.



(a) Radial deviation (b) Directional mean shift

Figure 1: **Evolution of CITD.** (a) Radial deviation reflects how far trajectories deviate from the normal class-conditional trajectory center. (b) Directional mean shift reflects the coherent displacement of intra-class trajectory residuals. The shaded regions denote the interquartile range (IQR) of non-target classes, and the dashed curves denote their median.

Building on these observations, we propose **PGIT**, an information-learning-trajectory-guided training-set purification method. PGIT first reconstructs the learning process of untrusted training data through replay training, and jointly measures the distance between sample trajectories and their benign class references together with the displacement of the class-mean trajectory to localize candidate target classes. Within each candidate class, PGIT then partitions samples according to directional differences in temporal activation gradients and uses trusted reference gradients to identify the benign partition, thereby detecting and removing suspicious samples.

Our main contributions are summarized as follows: (1) We reveal **CITD** in backdoor learning, where poisoned trajectories progressively diverge from benign ones while exhibiting coherent displacement directions, leading to a pronounced shift in the class-mean trajectory. (2) We propose **PGIT**, an information-learning-trajectory-based training-set purification method that localizes potential target classes through CITD and identifies poisoned samples via intra-class temporal activation-gradient analysis. (3) We conduct experiments across multiple datasets and representative backdoor attacks, validating CITD and showing that PGIT consistently outperforms detection baselines.

2 Related Work

Backdoor Attacks. Backdoor attacks typically implant a malicious association between a trigger and a target label through data poisoning. Early methods often used fixed local triggers and relabeled poisoned samples to the target class Gu et al. (2019). Subsequent studies explored more stealthy trigger mechanisms, including image blending Chen et al. (2017), spatial transformation Barni et al. (2019), physical triggers Li et al. (2021b), input-aware triggers Nguyen & Tran (2020), and sample-specific triggers Li et al. (2021c). To further improve stealthiness, many works investigated clean-label attacks Turner et al. (2019), which preserve the original labels of poisoned samples and thereby reduce label-level anomalies. Across these diverse attacks, the model must still learn a trigger–target association during training, motivating our focus on their shared learning dynamics.

Backdoor Defenses. Existing defenses against training-data poisoning mainly fall into two categories: poisoned sample detection Song et al. (2026) and in-training defense Zhu et al. (2023). Poisoned Sample Detection methods typically exploit differences between benign and poisoned samples in representations, prediction behavior, or learning dynamics, with the goal of identifying and removing poisoned samples before model training. Representative approaches include methods based on clustering or statistical anomalies in latent representations Chen et al. (2018); Tran et al. (2018); Hayase et al. (2021); Tang et al. (2021), as well as methods that leverage inconsistencies in prediction responses under input perturbations Gao et al. (2019); Guo et al. (2023); Hou et al. (2024). Recent studies further exploit richer learning signals, including topological evolution of representations Mo et al. (2024), activation-gradient geometry Yuan et al. (2025), and discriminative cues derived from representations and training behavior Hou et al. (2025). In-training defenses, by contrast, directly intervene in the optimization process to suppress backdoor learning through sample selection, reweighting, or robust training strategies Gao et al. (2023); Zhang et al. (2023); Qi et al. (2023); Chen et al. (2022); Zhao & Wressnegger (2025); Zhu et al. (2023).

3 Backdoor-Induced Learning Dynamics

In this section, we further examine the information-trajectory structure within the target class to understand the origin of the class-level anomalies in CITD, and investigate whether such differences are also reflected in the model's optimization dynamics.

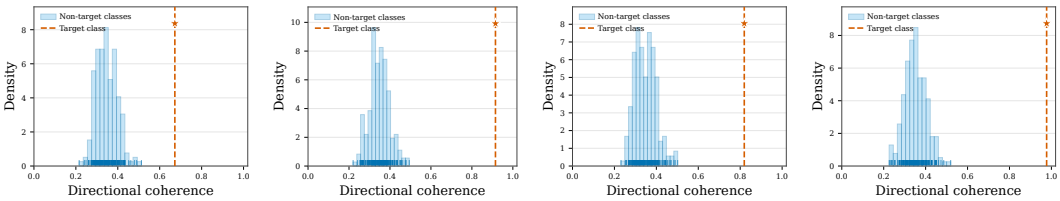
Settings. To examine whether the observed trajectory structure is consistent across different backdoor mechanisms, we consider four representative attacks on CIFAR-10 Krizhevsky et al. (2009) with ResNet-18 He et al. (2016): BATT Xu et al. (2023), Blended Chen et al. (2017), ISSBA Li et al. (2021c), and SIG Barni et al. (2019). For each class, we obtain sample-wise information-learning trajectories through replay training and compute their standardized residuals with respect to the class-specific normal trajectory reference. To quantify the directional consistency of within-class trajectory deviations, we normalize each residual to unit length and compute the mean resultant length as the directional coherence, $\mathrm{DC}(c) = \left\| \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{\mathbf{r}_i^{(c)}}{\|\mathbf{r}_i^{(c)}\|_2} \right\|_2$, where $\mathbf{r}_i^{(c)}$ denotes the standardized trajectory residual of the i-th sample in class c, and N_c is the number of samples in that class. A larger $\mathrm{DC}(c)$ indicates stronger directional consistency among within-class trajectory deviations. We further extract activation gradients for samples in the target class and examine the geometric relationship between benign and poisoned samples in the gradient-direction space; PCA Pearson (1901) is used only for two-dimensional visualization.

Results. ❶ **Information trajectories exhibit a stable directional bias.** As shown in Figure 2, across all four attacks, the target class consistently exhibits substantially higher directional coherence than non-target classes, indicating that its trajectory residuals tend to deviate along similar directions. We refer to this phenomenon as **directional bias**. This also explains the directional mean shift in Figure 1: when within-class residuals are dispersed, they tend to cancel out after aggregation, whereas aligned residuals produce a more pronounced class-level net displacement. ❷ **This directional discrepancy further extends to the optimization space.** As shown in Figure 3, benign and poisoned samples within the target class exhibit clear geometric separation in the activation-gradient direction space, while trusted clean references are predominantly located in the benign region. These observations indicate that the directional abnormalities induced by backdoor learning are reflected not only in information trajectories but also in the optimization directions of individual samples.

4 Method

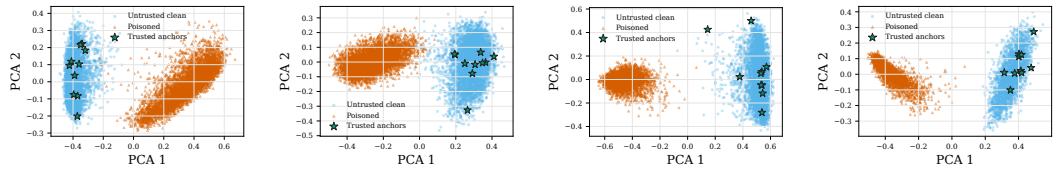
4.1 Problem Setting

Threat Model. Given an untrusted training set with K classes, $\mathcal{D}_u = \{(x_i, y_i)\}_{i=1}^{N} \subseteq \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes the input space and $\mathcal{Y} = \{1, \ldots, K\}$ denotes the label space, $\mathcal{D}_u$ may contain an unknown number of poisoned samples. Formally, let $\mathcal{D}_c$ denote the original benign training set. The attacker selects a subset $\mathcal{D}_{\mathrm{sub}} \subset \mathcal{D}_c$ and transforms a sample x_i into $\widetilde{x}_i = g(x_i, \Delta_i)$ using a trigger-generation function g, where Δ_i denotes the trigger pattern. The corresponding label is



(a) BATT (b) Blended (c) ISSBA (d) SIG

Figure 2: **Class-level directional bias across different attacks.**



(a) BATT (b) Blended (c) ISSBA (d) SIG

Figure 3: **Activation-gradient geometry within the target class across different attacks.**

written as $\widetilde{y}_i = \pi(y_i)$, where $\pi(\cdot)$ is the attacker's label-mapping function; for a clean-label attack, $\widetilde{y}_i = y_i$. This yields the poisoned set $\mathcal{D}_p$ and the untrusted training set accessible to the defender, $\mathcal{D}_u = (\mathcal{D}_c \setminus \mathcal{D}_{\text{sub}}) \cup \mathcal{D}_p$. Following prior work Hou et al. (2024); Yuan et al. (2025); Yang et al. (2026), we assume that the defender additionally has access to a small trusted benign reference set $\mathcal{D}_r = \{(x_j^r, y_j^r)\}_{j=1}^M$, with at least two samples per class.

Defender's Goal. The defender has access to the untrusted training set $\mathcal{D}_u$ and the small trusted reference set $\mathcal{D}_r$, and aims to identify potentially poisoned samples in $\mathcal{D}_u$ without prior knowledge of the attack. To facilitate sample identification, the defender may train an auxiliary model $f_w : \mathcal{X} \to \mathcal{Y}$ on $\mathcal{D}_u$ and exploit the learning signals produced during its training process.

4.2 Replay-Based Information Trajectory Construction

Replay Training. PGIT characterizes how individual samples are learned throughout model optimization rather than relying on a single training state. Given the untrusted training set $\mathcal{D}_u$, we independently train a replay model f_w and record its states at a set of checkpoints $\mathcal{E} = \{e_1, \ldots, e_T\}$. Let $\mathcal{L} = \{\ell_1, \ldots, \ell_L\}$ denote the set of representation layers used for trajectory analysis. For a sample x_i, we denote its internal representation at checkpoint e and layer ℓ by $t_{i,e}^{(\ell)}$, and its full softmax predictive distribution by $p_{i,e}$.

Information States. To characterize the information-learning state of each sample at different training stages, we use cross-fitted InfoNCE estimators to compute sample-level dependency scores for the input–representation and representation–prediction relationships:

$$u_{i,e}^{(\ell)} = \widehat{\imath}_{XT}\left(x_i, t_{i,e}^{(\ell)}\right), \qquad v_{i,e}^{(\ell)} = \widehat{\imath}_{T\widehat{Y}}\left(t_{i,e}^{(\ell)}, p_{i,e}\right). \tag{1}$$

Here, $\widehat{\imath}_{XT}$ and $\widehat{\imath}_{T\widehat{Y}}$ denote sample-level InfoNCE scores produced by contrastive mutual-information estimators, serving as local proxies for the corresponding information dependencies. The former characterizes the dependency between the input and its current representation, whereas the latter takes the full predictive distribution $p_{i,e}$ as input and characterizes the dependency between the representation and the model prediction. To ensure comparability across classes and checkpoints, we use consistent estimator initialization and fitting budgets and obtain sample-level scores via cross-fitting. Further estimation details are provided in the Appendix A.1.

Information-Learning Trajectory. A single checkpoint captures only the information state of a sample at a particular training stage and does not reveal how that state is formed. We therefore organize the information dependencies across training stages into a trajectory. For layer ℓ, the two checkpoint-wise information sequences are $u_i^{(\ell)} = [u_{i,e_t}^{(\ell)}]_{t=1}^T$ and $v_i^{(\ell)} = [v_{i,e_t}^{(\ell)}]_{t=1}^T$. To explicitly retain temporal changes, we further compute $\eta_{u,i}^{(\ell)} = [u_{i,e_t}^{(\ell)} - u_{i,e_{t-1}}^{(\ell)}]_{t=2}^T$ and $\eta_{v,i}^{(\ell)} = [v_{i,e_t}^{(\ell)} - v_{i,e_{t-1}}^{(\ell)}]_{t=2}^T$.

The information-learning trajectory at layer ℓ is then defined as

$$z_i^{(\ell)} = u_i^{(\ell)} \| v_i^{(\ell)} \| \eta_{u,i}^{(\ell)} \| \eta_{v,i}^{(\ell)}, \tag{2}$$

where $\|$ denotes vector concatenation. For an arbitrary layer subset $\mathcal{S} \subseteq \mathcal{L}$, we further define the multi-layer trajectory as

$$z_i^{\mathcal{S}} = \mathop{\Big\|}_{\ell \in \mathcal{S}} z_i^{(\ell)}. \tag{3}$$

Thus, $z_i^{\mathcal{S}}$ captures sample-wise information dependencies and their evolution across training stages and layers, serving as the dynamic representation for CITD.

4.3 CITD-Based Candidate Class Localization

Section 3 shows that backdoor learning causes the information trajectories of the target class to deviate from its class-specific normal learning pattern. PGIT converts this phenomenon into a class-localization signal by characterizing CITD from two complementary perspectives: deviation magnitude and aggregated direction.

Class-Conditional Reference. Because different classes naturally exhibit different learning patterns, we first use trusted reference samples to construct a class-conditional trajectory center. For class c and layer subset $\mathcal{S}$, we define

$$\mu_c^{\mathcal{S}} = \mathrm{Median}\left\{ z_j^{\mathcal{S}} \mid (x_j^r, y_j^r) \in \mathcal{D}_r,\ y_j^r = c \right\}, \tag{4}$$

where the median is computed element-wise. We then estimate an element-wise robust scale $\sigma^{\mathcal{S}}$ from the residuals of the reference samples relative to their corresponding class centers. For a sample x_i with label c in $\mathcal{D}_u$, its standardized trajectory residual is

$$r_i^{\mathcal{S}} = \frac{z_i^{\mathcal{S}} - \mu_c^{\mathcal{S}}}{\sigma^{\mathcal{S}}}. \tag{5}$$

Therefore, $r_i^{\mathcal{S}}$ measures the deviation of a sample from its class-specific normal learning pattern rather than the absolute trajectory difference across classes. In practice, we estimate the scale using robust statistics such as MAD and IQR and clip extreme coordinates to reduce the influence of a small number of unstable estimates.

Radial Deviation. The first manifestation of CITD is that sample trajectories move away from their normal class-conditional patterns. Let $\mathcal{L}_r \subseteq \mathcal{L}$ denote the layer subset used for radial analysis. We define the normalized trajectory radius of sample x_i as $\rho_i = \|r_i^{\mathcal{L}_r}\|_2 / \sqrt{d_r}$, where d_r is the dimensionality of the corresponding trajectory residual. To capture both typical class-level deviations and samples with relatively large residuals, the radial deviation score of class c is defined as

$$R_c = \frac{1}{2}\,\mathrm{Median}_{i:y_i=c}\,\rho_i + \frac{1}{2} Q_{0.75}\left(\{\rho_i : y_i = c\}\right). \tag{6}$$

Here, $Q_{0.75}$ denotes the 75th percentile. A larger R_c indicates that samples in class c deviate more strongly from their normal trajectory region.

Directional Mean Shift. Residual magnitude alone does not capture directional structure. As shown in Section 3, trajectory residuals in the contaminated class exhibit stronger directional consistency. To characterize this aggregated effect, we use the full layer set $\mathcal{L}$ and define

$$D_c = \frac{\left\| \frac{1}{|\mathcal{B}_c|} \sum_{i \in \mathcal{B}_c} r_i^{\mathcal{L}} \right\|_2}{\sqrt{d}}, \tag{7}$$

where $\mathcal{B}_c$ denotes the set of samples used to estimate the overall displacement of class c, and d is the dimensionality of the corresponding multi-layer trajectory residual. When intra-class residual directions are dispersed, their aggregated effects tend to cancel out; when the residuals become directionally aligned, they produce a more pronounced net displacement. Thus, R_c and D_c characterize the radial deviation and directional shift of CITD, respectively.

Candidate Localization. Because different attacks may exhibit different forms of CITD, PGIT ranks classes independently according to R_c and D_c and takes the union of the two candidate sets:

$$\mathcal{C}_{\mathrm{cand}} = \mathrm{Top}_k\left(\{R_c\}_{c=1}^K\right) \cup \mathrm{Top}_k\left(\{D_c\}_{c=1}^K\right). \tag{8}$$

In this way, radial deviation and directional shift provide complementary evidence for different trajectory distortions while substantially narrowing the potential contaminated region.

4.4 TEMPORAL GRADIENT ATTRIBUTION

CITD identifies the classes in which abnormal learning occurs but does not reveal which samples are responsible for the abnormality. PGIT therefore exploits the activation-gradient separability observed in Section 3 to perform instance-level attribution within the candidate classes.

Tracking Gradient Directions. For a sample (x_i, y_i) in a candidate class, let $s_{w_e,y_i}(x_i)$ denote the prediction logit associated with its observed label y_i at checkpoint e. Its activation gradient at representation layer ℓ_g is defined as

$$\gamma_{i,e} = \mathrm{Pool}\left(\frac{\partial s_{w_e,y_i}(x_i)}{\partial t_{i,e}^{(\ell_g)}}\right), \tag{9}$$

where $\mathrm{Pool}(\cdot)$ denotes spatial or token-wise pooling together with any necessary dimensionality reduction. We further define the unit gradient direction as $\bar{\gamma}_{i,e} = \gamma_{i,e}/\|\gamma_{i,e}\|_2$. This gradient characterizes the local direction through which a sample promotes the prediction of its observed label under the current model state. PGIT tracks this geometry across multiple replay checkpoints to capture the directional divergence between benign and poisoned samples during optimization.

Locating the Discriminative Stage. The gradient divergence between benign and poisoned samples does not remain equally pronounced throughout training. To identify the most discriminative stage, for each candidate class c and checkpoint e, we perform spherical two-means clustering in the unit-gradient space and denote the two directional centers by $m_{c,e}^{(0)}$ and $m_{c,e}^{(1)}$. Their directional separation is defined as

$$\psi_{c,e} = 1 - \left\langle m_{c,e}^{(0)}, m_{c,e}^{(1)} \right\rangle. \tag{10}$$

Since benign classes may also exhibit natural intra-class gradient dispersion, we further construct a robust background using the other classes at the same checkpoint:

$$Z_{c,e} = \frac{\psi_{c,e} - \mathrm{Median}_{c' \neq c}\, \psi_{c',e}}{\mathrm{Scale}_{c' \neq c}\,(\psi_{c',e})}. \tag{11}$$

Let $\mathcal{E}_{\mathrm{valid}} \subseteq \mathcal{E}$ denote the set of valid checkpoints considered for gradient-divergence search. We then select

$$e_c^\star = \arg\max_{e \in \mathcal{E}_{\mathrm{valid}}} Z_{c,e}, \tag{12}$$

i.e., the training stage at which class c exhibits the strongest directional divergence relative to the other classes. If the maximum separation does not sufficiently exceed the background variation, PGIT does not remove any sample from this class. Otherwise, the two sample partitions obtained at $e_c^\star$ are frozen for subsequent orientation.

Orienting Gradient Partitions. The two gradient partitions characterize distinct optimization directions but do not indicate which partition corresponds to benign learning. We therefore preserve the partition membership obtained at $e_c^\star$ and recompute the activation gradients of its members at the final replay checkpoint e_T. Let $q_c^{(0)}$ and $q_c^{(1)}$ denote the unit centers of the two frozen partitions, and let $\mathcal{G}_c^r = \{\bar{\gamma}_{j,e_T}^r \mid (x_j^r, y_j^r) \in \mathcal{D}_r,\ y_j^r = c\}$ denote the trusted reference gradients of class c.

Each reference gradient votes for the partition center with the larger cosine similarity. The benign partition is determined by majority support:

$$a_c^{\mathrm{ben}} = \arg\max_{a \in \{0,1\}} \sum_{\mathbf{g} \in \mathcal{G}_c^r} \mathbb{I}\left[\mathrm{sim}(\mathbf{g}, q_c^{(a)}) \geq \mathrm{sim}(\mathbf{g}, q_c^{(1-a)})\right]. \tag{13}$$

We further define the corresponding reference-gradient support as

$$s_c = \frac{1}{|\mathcal{G}_c^r|} \sum_{\mathbf{g} \in \mathcal{G}_c^r} \mathbb{I}\left[\mathrm{sim}(\mathbf{g}, q_c^{(a_c^{\mathrm{ben}})}) \geq \mathrm{sim}(\mathbf{g}, q_c^{(1-a_c^{\mathrm{ben}})})\right]. \tag{14}$$

A candidate class is accepted for sample attribution only when $Z_{c,e_c^\star} \geq \tau_z$ and $s_c \geq \tau_s$. Otherwise, we set $\widehat{\mathcal{P}}_c = \emptyset$. Let $a_c^{\mathrm{sus}} = 1 - a_c^{\mathrm{ben}}$. For an accepted candidate class, we define the suspicious sample set as $\widehat{\mathcal{P}}_c = \{i \mid y_i = c, \alpha_i = a_c^{\mathrm{sus}}\}$, where $\alpha_i \in \{0,1\}$ denotes the frozen partition assignment of sample x_i obtained at $e_c^\star$. After processing all candidate classes, PGIT aggregates $\widehat{\mathcal{P}} = \bigcup_{c \in \mathcal{C}_{\mathrm{cand}}} \widehat{\mathcal{P}}_c$ and outputs the purified training set $\mathcal{D}_{\mathrm{pur}} = \{(x_i, y_i) \in \mathcal{D}_u \mid i \notin \widehat{\mathcal{P}}\}$.

Table 1: Backdoor detection performance (TPR/FPR, %) with ResNet-18. Best results are in **boldface**; non-zero FPR values below 0.01% are reported as < 0.01.

Datasets	Detection / Attack ↓	STRIP TPR	STRIP FPR	SCALE-UP TPR	SCALE-UP FPR	TED TPR	TED FPR	IBD-PSC TPR	IBD-PSC FPR	AGPD TPR	AGPD FPR	FLARE TPR	FLARE FPR	PGIT TPR	PGIT FPR
CIFAR-10	BadNets	66.02	2.64	71.90	17.76	59.12	3.51	**99.86**	3.06	75.54	1.33	90.82	**< 0.01**	97.46	0.06
	BATT	0.52	2.29	**100.00**	19.76	22.30	3.66	**100.00**	4.83	82.42	25.04	95.18	0.19	98.80	**0.02**
	Blended	57.06	3.09	87.34	17.51	8.24	4.49	93.84	3.72	72.50	25.32	82.00	**0.21**	**100.00**	0.35
	IAD	97.46	2.50	100.00	14.27	77.14	7.94	**99.88**	6.28	99.29	**0.02**	97.32	0.04	96.82	0.12
	ISSBA	22.82	2.10	72.08	18.23	80.22	4.95	**100.00**	7.53	99.54	**0.00**	97.58	0.06	99.78	0.01
	LC	78.88	1.94	**100.00**	17.14	0.72	3.83	**100.00**	0.70	0.00	16.22	99.92	**0.00**	**100.00**	0.03
	PhysicalBA	45.48	2.49	76.94	21.43	72.94	9.27	**96.70**	3.54	74.80	1.14	95.58	**< 0.01**	94.92	0.06
	Refool	0.38	2.49	86.68	18.32	23.68	4.77	0.08	3.97	53.96	4.77	0.18	4.12	**99.38**	**0.09**
	SIG	59.08	2.01	**100.00**	5.26	1.36	3.10	99.28	0.39	95.44	41.10	84.32	0.05	99.00	**0.00**
	TrojanNN	31.70	0.56	**100.00**	10.00	34.88	3.69	99.98	5.66	95.90	44.10	90.02	**< 0.01**	99.14	0.20
	Average	45.94	2.21	89.49	15.97	38.06	4.92	88.96	3.97	74.94	15.91	83.29	0.47	**98.53**	**0.09**
Tiny-ImageNet-200	BATT	1.03	3.59	**100.00**	4.80	76.47	4.35	**100.00**	0.26	99.99	37.97	85.73	**< 0.01**	99.96	< 0.01
	Blended	93.85	2.92	59.58	2.59	0.00	4.39	0.00	0.38	**100.00**	42.09	98.21	**0.00**	**100.00**	0.13
	IAD	5.88	4.15	7.48	4.69	4.05	4.44	31.48	3.13	99.77	**0.00**	98.21	0.02	**100.00**	< 0.01
	ISSBA	98.47	4.47	93.13	1.35	0.02	4.67	2.39	1.04	0.00	0.21	99.97	**< 0.01**	**99.98**	< 0.01
	TrojanNN	99.44	3.00	95.39	3.29	0.75	4.08	**99.98**	0.37	**99.98**	37.24	77.40	**0.00**	86.52	< 0.01
	WaNet	31.80	3.84	81.87	4.52	99.46	4.43	99.87	0.25	99.11	43.73	97.36	0.05	**100.00**	**< 0.01**
	Average	55.08	3.66	72.91	3.54	30.13	4.39	55.62	0.90	83.14	26.87	92.81	**0.01**	**97.74**	0.03

5 EXPERIMENTS

5.1 SETUP

Models and Datasets. To evaluate the stability of CITD across different model settings, we adopt ResNet-18 He et al. (2016) and CLIP-RN50 Radford et al. (2021) as backbone models. ResNet-18 is trained from random initialization, while CLIP-RN50 uses publicly available pretrained weights. Experiments are conducted on CIFAR-10 Krizhevsky et al. (2009) and Tiny-ImageNet-200. Tiny-ImageNet-200 Le & Yang (2015) contains 200 classes, with 500 training images and 50 validation images per class.

Backdoor Attacks. To evaluate the defense performance of PGIT, we consider 11 representative backdoor attacks, including BadNets Gu et al. (2019), BATT Xu et al. (2023), Blended Chen et al. (2017), IAD Nguyen & Tran (2020), ISSBA Li et al. (2021c), LC Turner et al. (2019), PhysicalBA Li et al. (2021b), Refool Liu et al. (2020), SIG Barni et al. (2019), TrojanNN Liu et al. (2018), and WaNet Nguyen & Tran (2021). For the clean-label attacks, we set the poisoning rate of LC to 25% to ensure sufficient attack strength, while SIG uses 5% following its original implementation setting. For all other attacks, the poisoning rate is set to 10%, with class 0 designated as the target class.

Defense Baselines. To evaluate PGIT, we consider two categories of representative backdoor defenses: 1) **Poisoned-sample detection methods**, including STRIP Gao et al. (2019), SCALE-UP Guo et al. (2023), TED Mo et al. (2024), IBD-PSC Hou et al. (2024), FLARE Hou et al. (2025), and AGPD Yuan et al. (2025), which aim to identify poisoned samples in the training set; 2) **In-training defense methods**, including ASD Gao et al. (2023), CBD Zhang et al. (2023), CT Qi et al. (2023), D-ST Chen et al. (2022), HARVEY Zhao & Wressnegger (2025), and V&B Zhu et al. (2023), which suppress backdoor learning during training. Implementation details of PGIT are provided in Appendix A.2.

Evaluation Metrics. We use true positive rate (TPR) and false positive rate (FPR) to evaluate poisoned-sample detection. TPR measures the proportion of poisoned samples correctly identified, while FPR measures the proportion of benign samples incorrectly classified as poisoned. We further retrain models on the purified training set and report benign accuracy (BA) and attack success rate (ASR) to evaluate normal-task performance and residual backdoor effects, respectively.

5.2 MAIN RESULTS

Evaluation on Poisoned-Sample Detection Methods. Table 1 compares PGIT with six representative defenses. ❶ PGIT achieves the highest average TPR on CIFAR-10 and Tiny-ImageNet-200, reaching 98.53% and 97.74%, respectively, outperforming the strongest baseline by 9.04 and 4.93 percentage points. ❷ PGIT also maintains very low false positives, with average FPRs of 0.09% and

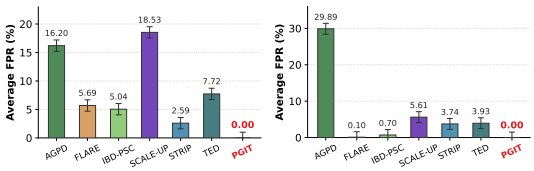
Table 2: Backdoor defense performance (%) on CIFAR-10 with ResNet-18.

Attack	No Defense		ASD		CBD		CT		D-ST		HARVEY		V&B		PGIT	
	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
BadNets	91.17	97.42	93.29	93.30	85.22	3.12	92.61	1.21	93.39	66.97	84.41	**0.04**	**94.35**	0.54	91.01	1.53
BATT	89.74	99.80	**94.12**	4.98	86.58	2.88	93.44	47.97	91.99	99.99	93.35	42.19	94.07	**0.01**	90.12	1.65
Blended	91.45	100.00	92.49	42.50	83.03	60.77	92.42	2.06	86.29	6.19	88.70	38.22	**94.41**	0.53	91.58	**0.16**
IAD	94.58	99.73	93.41	1.56	85.09	6.66	93.42	5.13	94.08	**0.00**	90.81	0.27	**94.58**	8.57	93.55	0.70
ISSBA	94.86	99.94	**93.77**	99.99	87.14	99.91	93.08	**0.73**	86.43	96.77	86.52	1.97	92.30	2.66	91.64	1.22
LC	92.18	99.74	93.84	**0.00**	85.79	63.32	93.94	0.90	**93.98**	**0.00**	92.17	0.79	91.84	87.13	92.12	0.61
PhysicalBA	94.01	96.76	93.90	0.44	77.80	2.11	93.58	0.92	91.18	18.22	90.75	**0.29**	**94.00**	0.86	93.94	0.91
Refool	91.09	98.57	86.02	61.14	80.66	48.20	87.05	8.83	84.84	15.94	83.31	49.89	84.55	41.78	**91.82**	**3.34**
Average	92.39	99.00	**92.61**	37.99	83.91	35.87	92.44	8.47	90.27	38.01	88.75	16.71	92.51	17.76	91.97	**1.27**

0.03% on the two datasets. ❸ The advantage remains consistent across diverse attacks; for example, PGIT achieves 99.38% TPR against Refool on CIFAR-10 and near-perfect TPRs against five attacks on Tiny-ImageNet-200. Overall, PGIT delivers consistently strong poisoned-sample detection across datasets and attacks.

Evaluation on In-Training Defense Methods. To evaluate the downstream defense effectiveness of the purified training data, we compare PGIT with six in-training defenses that directly suppress backdoor learning. The results are shown in Table 2. ❶ PGIT reduces the average ASR from 99.00% without defense to 1.27%, further lowering it by 7.20 percentage points compared with the strongest baseline, CT (8.47%). ❷ Meanwhile, PGIT maintains an average BA of 91.97%, only 0.42 percentage points lower than the undefended model (92.39%), indicating minimal degradation in benign performance. ❸ PGIT keeps the ASR below 3.34% across all attacks, including challenging cases such as Refool and ISSBA. Overall, PGIT effectively suppresses backdoor behavior while preserving benign model performance.

Evaluation on Clean Training Sets. In practical scenarios, defenders may not know in advance whether an untrusted training set contains poisoned samples. Therefore, a reliable poisoned-sample detection method should not only achieve strong detection performance, but also minimize the erroneous removal of benign samples when the training set is entirely clean. To this end, we further evaluate the false-positive behavior of different methods on fully clean CIFAR-10 and Tiny-ImageNet-200 training sets. As shown in Figure 4, existing methods still produce varying degrees of false positives, whereas PGIT achieves an average FPR of 0.00% on both datasets, demonstrating its ability to avoid unnecessary removal of benign samples in the absence of backdoor contamination.



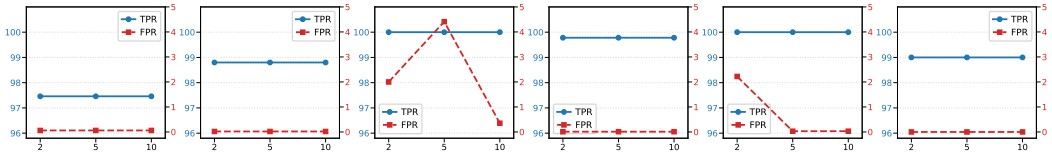
(a) CIFAR-10 (b) Tiny-ImageNet-200

Figure 4: Average FPR (%) of different detectors on fully clean training sets.

5.3 Ablation Studies

Impact of the Number of Trusted Reference Samples. We further investigate the effect of the number of trusted reference samples per class, considering $\{2, 5, 10\}$ samples. As shown in Figure 5, PGIT is robust to this choice: even with only two trusted samples per class, the TPR remains between 97.46% and 100.00% across all six attacks and is largely insensitive to the number of reference samples. In contrast, the FPR is more sensitive to the reference-set size, but remains low overall and becomes more stable when using 10 samples per class.

Impact of Different Poisoning Rates. To evaluate the robustness of PGIT under varying contamination levels, we test four poisoning rates, i.e., 1%, 5%, 10%, and 20%, on CIFAR-10 with ResNet-18, as shown in Table 3. PGIT maintains stable detection performance across different poisoning rates, achieving average TPRs of 95.40%, 96.20%, 98.22%, and 97.57%, respectively. Notably, even at the extremely low poisoning rate of 1%, PGIT still achieves high TPRs for most attacks. Meanwhile, the FPR remains low in most settings, with only a few isolated fluctuations. These results indicate that PGIT is robust to substantial variations in the poisoning ratio.



(a) BadNets (b) BATT (c) Blended (d) ISSBA (e) LC (f) SIG

Figure 5: Impact of the number of trusted reference samples per class on detection performance under six backdoor attacks on CIFAR-10 with ResNet-18.

Table 3: Detection performance (%) under different poisoning rates on CIFAR-10 with ResNet-18.

Poison Rate	BadNets		BATT		Blended		ISSBA		SIG		TrojanNN		PhysicalBA		WaNet	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
1	97.40	0.02	98.80	< 0.01	100.00	0.03	99.60	< 0.01	98.20	1.79	97.60	0.00	92.60	0.00	79.00	0.04
5	90.20	0.05	98.88	0.00	100.00	0.07	100.00	< 0.01	99.00	0.00	99.00	0.05	85.84	0.11	96.68	2.55
10	97.46	0.06	98.80	0.02	100.00	0.35	99.78	0.01	99.00	0.00	99.14	0.20	94.92	0.06	96.69	0.04
20	95.67	< 0.01	100.00	0.10	99.91	< 0.01	99.11	0.01	100.00	0.00	99.70	0.09	89.84	0.13	96.29	0.04

Impact of Radial and Directional CITD Components. To analyze the roles of the two CITD components, we evaluate the radial-only and directional-only variants of PGIT. As shown in Table 4, the directional component detects BadNets, Blended, and SIG, but fails on BATT, ISSBA, and WaNet; the radial component performs well across most attacks, but its TPR drops to 68.54% on WaNet. Combining both components raises the TPR on WaNet to 96.69% while reducing the FPR to 0.04%, without degrading performance on other attacks. These results demonstrate that the two CITD components are complementary and together yield more stable detection performance.

Table 4: Ablation of radial and directional CITD components on CIFAR-10 with ResNet-18 (%).

Attack	Directional		Radial		Full	
	TPR	FPR	TPR	FPR	TPR	FPR
BadNets	97.46	0.06	97.46	0.06	97.46	0.06
BATT	0.00	0.00	98.80	0.02	98.80	0.02
Blended	100.00	0.36	100.00	0.36	100.00	0.35
ISSBA	0.00	0.00	99.78	0.01	99.78	0.01
SIG	99.00	0.00	99.00	0.00	99.00	0.00
WaNet	0.00	0.00	68.54	0.33	96.69	0.04

Impact of Different Target Classes. To evaluate the sensitivity of PGIT to the attack target, we consider target classes 0, 1, and 3 under multiple backdoor attacks. As shown in Table 5, PGIT maintains stable detection performance across different target classes, with TPRs consistently ranging from 95.46% to 100.00% across all four attacks. The FPR remains low in most settings; only SIG on target class 3 shows a noticeable increase to 3.39%, while still achieving a high TPR of 99.84%. These results indicate that PGIT is largely insensitive to the choice of target class and remains effective across different attack targets.

Table 5: Detection performance under different target classes on CIFAR-10 with ResNet-18.

Attack	Target 0		Target 1		Target 3	
	TPR	FPR	TPR	FPR	TPR	FPR
BadNets	97.46	0.06	95.70	< 0.01	96.94	0.10
Blended	100.00	0.35	99.90	0.00	100.00	0.04
IAD	96.82	0.12	95.46	0.16	98.80	0.77
SIG	99.00	0.00	99.96	0.00	99.84	3.39

Additional Analyses in Appendix. We provide further evaluations in Appendix Secs. B and C, including defense-aware adaptive attacks, generalization to Tiny-ImageNet-200 and CLIP-RN50, detailed Stage-I/Stage-II analyses, hyperparameter sensitivity, and computational overhead. Notably, PGIT maintains at least 99.18% TPR with at most 0.57% FPR under adaptive attacks.

6 Conclusion

In this work, we reveal **Class-Conditional Information-Trajectory Distortion (CITD)**, a backdoor-induced learning phenomenon characterized by growing poisoned–benign divergence and coherent class-level displacement. Based on CITD, we propose **PGIT**, which localizes suspicious classes through information-learning trajectories and identifies poisoned samples using temporal activation-gradient geometry. Extensive experiments demonstrate the effectiveness and robustness of PGIT across diverse attacks, datasets, and model settings.

REFERENCES

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 (ICIP)*, pp. 101–105. IEEE, 2019.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 407–425. IEEE, 2024.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. *Advances in Neural Information Processing Systems*, 35:9727–9737, 2022.

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.

Kuofeng Gao, Yang Bai, Jindong Gu, Yong Yang, and Shu-Tao Xia. Backdoor defense via adaptively splitting poisoned dataset. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4005–4014. IEEE, 2023.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pp. 113–125, 2019.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *Ieee Access*, 7:47230–47244, 2019.

Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. *arXiv preprint arXiv:2302.03251*, 2023.

Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: Defending against backdoor attacks using robust statistics. In *International Conference on Machine Learning*, pp. 4129–4139. PMLR, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ryan Holland, Shantanu Pal, and Lei Pan. Exploring backdoor vulnerabilities and fairness challenges in generative models: A survey. *ACM Computing Surveys*, 2026.

Linshan Hou, Ruili Feng, Zhongyun Hua, Wei Luo, Leo Yu Zhang, and Yiming Li. Ibd-psc: Input-level backdoor detection via parameter-oriented scaling consistency. *arXiv preprint arXiv:2405.09786*, 2024.

Linshan Hou, Wei Luo, Zhongyun Hua, Songhua Chen, Leo Yu Zhang, and Yiming Li. Flare: Toward universal dataset purification against backdoor attacks. *IEEE Transactions on Information Forensics and Security*, 20:6459–6473, 2025.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

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, 2021a.

Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. Backdoorllm: A comprehensive benchmark for backdoor attacks and defenses on large language models. *Advances in neural information processing systems*, 38, 2026.

Yiming Li, Tongqing Zhai, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor attack in the physical world. *arXiv preprint arXiv:2104.02361*, 2021b.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16443–16452. IEEE, 2021c.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc, 2018.

Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European conference on computer vision*, pp. 182–199. Springer International Publishing Cham, 2020.

Xiaoxing Mo, Yechao Zhang, Leo Yu Zhang, Wei Luo, Nan Sun, Shengshan Hu, Shang Gao, and Yang Xiang. Robust backdoor detection for deep learning via topological evolution dynamics. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 2048–2066. IEEE, 2024.

Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

Minzhou Pan, Yi Zeng, Lingjuan Lyu, Xue Lin, and Ruoxi Jia. {ASSET}: Robust backdoor data detection across a multiplicity of deep learning paradigms. In *32nd USeNIX security symposium (USeNIX security 23)*, pp. 2725–2742, 2023.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

Xiangyu Qi, Tinghao Xie, Jiachen T Wang, Tong Wu, Saeed Mahloujifar, and Prateek Mittal. Towards a proactive {ML} approach for detecting backdoor poison samples. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 1685–1702, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Zhengyao Song, Meixi Zheng, Ke Xu, Yongqiang Li, and Baoyuan Wu. Test-time poisoned sample detection by exploiting shallow malicious matching in backdoored clip. In *International Conference on Learning Representations*, volume 2026, pp. 153545–153566, 2026.

Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of {DNNs} for robust backdoor contamination detection. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1541–1558, 2021.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

Tong Xu, Yiming Li, Yong Jiang, and Shu-Tao Xia. Batt: Backdoor attack with transformation-based triggers. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Zhiguo Yang, Dongsheng Xu, Ruizhi Zhong, Jiacheng Pi, Xingxing Huang, and Wenjie Ruan. Logit-margin repulsion for backdoor defense. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34918–34928, 2026.

Danni Yuan, Mingda Zhang, Shaokui Wei, Li Liu, and Baoyuan Wu. Activation gradient based poisoned sample detection against backdoor attacks. In *International Conference on Learning Representations*, volume 2025, pp. 76161–76188, 2025.

Mingda Zhang, Mingli Zhu, Zihao Zhu, Li Shen, and Baoyuan Wu. Reliable poisoned sample detection against backdoor attacks enhanced by sharpness aware minimization. In *International Conference on Learning Representations*, volume 2026, pp. 123199–123229, 2026.

Zaixi Zhang, Qi Liu, Zhicai Wang, Zepu Lu, and Qingyong Hu. Backdoor defense via deconfounded representation learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12228–12238. IEEE, 2023.

Qi Zhao and Christian Wressnegger. Two sides of the same coin: Learning the backdoor to remove the backdoor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22804–22812, 2025.

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 *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 155–164. IEEE, 2023.

Tracing How Backdoors Are Learned: Poisoned Sample Detection via Information-Trajectory Distortion

Appendix

A Details of PGIT **14**

A.1 Mutual-Information Estimation Details 14

A.2 Implementation Details 14

A.3 Detailed Algorithm of PGIT 14

B Additional Experimental Results **14**

B.1 Backdoor Purification on Tiny-ImageNet-200 14

B.2 Generalization to CLIP-RN50 16

B.3 Defense-Aware Adaptive Attacks 16

C Additional Analysis of PGIT **17**

C.1 Comparison of Stage-I Representations 17

C.2 Impact of Instance-Level Gradient Attribution 18

C.3 Impact of the Candidate-Class Budget k 19

C.4 Impact of the Gradient-Separation Threshold τ_z 20

C.5 Impact of the Reference-Gradient Support Threshold τ_s 20

C.6 Computational Overhead Analysis 21

A Details of PGIT

A.1 Mutual-Information Estimation Details

Estimation and Cross-Fitting. PGIT uses class-wise two-fold cross-fitted sampled InfoNCE critics to estimate sample-level dependency scores for the input–representation (X, T) and representation–prediction $(T, \hat{Y})$ relations. These scores are used only as local dependency proxies rather than calibrated estimates of absolute mutual information. For each class, representation layer, and replay checkpoint, samples are divided into two folds. The critic is trained on one fold and used exclusively to score the held-out fold, after which the two folds are swapped and the resulting scores are merged. Therefore, each sample is scored by a critic that was not trained on that sample. Positive and negative pairs are constructed within the same observed class to prevent the critic from exploiting class identity alone.

Critic Architecture and Inputs. The critic is parameterized as

$$s_\theta(a, b) = \mathrm{MLP}_\theta([a; b]), \tag{15}$$

where $[\cdot;\cdot]$ denotes concatenation. The MLP contains two hidden layers of width 128, each followed by a ReLU activation, and a scalar output layer. Separate critics are used for the X–T and T–$\hat{Y}$ relations. Convolutional representations are spatially average-pooled before being passed to the critic, while $\hat{Y}$ is represented by the full predictive distribution

$$p_{i,e} = \mathrm{softmax}(f_{w_e}(x_i)) \in \mathbb{R}^K, \tag{16}$$

rather than the predicted label or a one-hot vector. For high-resolution inputs, images are adaptively average-pooled to 32×32 before being used by the X–T critic. Each critic is trained for 8 epochs using AdamW. The learning rates are 3×10^{-4} and 5×10^{-4} for the X–T and T–$\hat{Y}$ critics, respectively, with batch sizes of 128 and 512. At most 32 negative pairs are used per sample.

Reference Samples and Reproducibility. Both untrusted samples and trusted reference samples participate in the above class-wise cross-fitting procedure and receive independent dependency scores; poison indicators are never provided to the critics. The resulting trusted trajectories are subsequently used only to construct the class-conditional normal references. Fold assignments, negative sampling, and critic initialization are deterministically derived from the experimental seed. For CIFAR-10 with ResNet-18, PGIT tracks 15 replay checkpoints and two representation layers; the detailed checkpoint and layer configurations are provided in the experimental settings.

A.2 Implementation Details

Unless otherwise specified, PGIT uses 10 trusted benign reference samples per class. For candidate-class localization, the radial and directional branches independently select the Top-k classes, and their union forms $\mathcal{C}_{\mathrm{cand}}$. We set the candidate budget to $k = 3$ for CIFAR-10 and $k = 10$ for Tiny-ImageNet-200. For temporal gradient attribution, the gradient-separation threshold and reference-gradient support threshold are set to $\tau_z = 3.0$ and $\tau_s = 0.8$, respectively.

A.3 Detailed Algorithm of PGIT

The pseudo-code of PGIT is summarized in Algorithm 1.

B Additional Experimental Results

B.1 Backdoor Purification on Tiny-ImageNet-200

To further evaluate the effectiveness of the purified training data, we retrain models on Tiny-ImageNet-200 after removing the samples identified by PGIT. As shown in Figure 6, purification substantially suppresses backdoor behavior while largely preserving benign performance. The average ASR decreases from 99.73% to 1.05%, whereas the average BA remains nearly unchanged (57.16% vs. 57.15%). For five of the six attacks, the post-purification ASR is reduced to at most

Algorithm 1 PGIT: Purification Guided by Information Trajectories

Input: Untrusted training set $\mathcal{D}_u$, trusted reference set $\mathcal{D}_r$, replay checkpoints $\mathcal{E} = \{e_1, \ldots, e_T\}$, representation layers $\mathcal{L}$, radial-layer subset $\mathcal{L}_r$, gradient layer ℓ_g, candidate budget k, thresholds τ_z and τ_s.
Output: Suspicious sample set $\widehat{\mathcal{P}}$ and purified training set $\mathcal{D}_{\mathrm{pur}}$.
Train a replay model on $\mathcal{D}_u$ and save model states $\{w_e\}_{e\in\mathcal{E}}$.
for each sample $x_i \in \mathcal{D}_u \cup \mathcal{D}_r$ **do**
  for each $e \in \mathcal{E}$ and $\ell \in \mathcal{L}$ **do**
    Extract $t_{i,e}^{(\ell)}$ and $p_{i,e}$, and compute $u_{i,e}^{(\ell)}$ and $v_{i,e}^{(\ell)}$ using Eq. (1).
  end for
  Construct $z_i^{(\ell)} = u_i^{(\ell)} \| v_i^{(\ell)} \| \eta_{u,i}^{(\ell)} \| \eta_{v,i}^{(\ell)}$ and $z_i^{\mathcal{L}} = \|_{\ell\in\mathcal{L}} z_i^{(\ell)}$.
end for
for each class $c \in \{1, \ldots, K\}$ **do**
  Compute the class-conditional reference center $\mu_c^{\mathcal{S}} = \mathrm{Median}\{z_j^{\mathcal{S}} \mid (x_j^r, y_j^r) \in \mathcal{D}_r,\ y_j^r = c\}$.
end for
Estimate the robust scale $\sigma^{\mathcal{S}}$ from trusted reference residuals.
Compute standardized trajectory residuals $r_i^{\mathcal{S}} = (z_i^{\mathcal{S}} - \mu_{y_i}^{\mathcal{S}})/\sigma^{\mathcal{S}}$.
for each class $c \in \{1, \ldots, K\}$ **do**
  Compute $\rho_i = \|r_i^{\mathcal{L}_r}\|_2/\sqrt{d_r}$, for all i with $y_i = c$.
  Compute the radial deviation score $R_c = \frac{1}{2}\,\mathrm{Median}_{i:y_i=c}\,\rho_i + \frac{1}{2}Q_{0.75}(\{\rho_i : y_i = c\})$.
  Compute the directional mean-shift score $D_c = \frac{\| |\mathcal{B}_c|^{-1} \sum_{i\in\mathcal{B}_c} r_i^{\mathcal{L}} \|_2}{\sqrt{d}}$.
end for
Localize candidate classes: $\mathcal{C}_{\mathrm{cand}} = \mathrm{Top}_k(\{R_c\}) \cup \mathrm{Top}_k(\{D_c\})$.
Initialize $\widehat{\mathcal{P}} \leftarrow \emptyset$.
for each $c \in \mathcal{C}_{\mathrm{cand}}$ **do**
  for each $e \in \mathcal{E}_{\mathrm{valid}}$ **do**
    Compute unit activation gradients $\bar{\gamma}_{i,e}$ for samples with $y_i = c$.
    Apply spherical two-means and compute the normalized separation score $Z_{c,e}$.
  end for
  Select $e_c^\star = \arg\max_{e\in\mathcal{E}_{\mathrm{valid}}} Z_{c,e}$.
  if $Z_{c,e_c^\star} < \tau_z$ **then**
    $\widehat{\mathcal{P}}_c \leftarrow \emptyset$; **continue**.
  end if
  Freeze the partition assignments $\{\alpha_i\}$ obtained at $e_c^\star$.
  Recompute the two partition centers $q_c^{(0)}$ and $q_c^{(1)}$ at the final checkpoint e_T.
  Compute trusted reference gradients $\mathcal{G}_c^r$ at e_T.
  Determine the benign partition a_c^{ben} and reference support s_c using Eqs. (13)–(14).
  if $s_c \geq \tau_s$ **then**
    $a_c^{\mathrm{sus}} \leftarrow 1 - a_c^{\mathrm{ben}}$.
    $\widehat{\mathcal{P}}_c \leftarrow \{i \mid y_i = c,\ \alpha_i = a_c^{\mathrm{sus}}\}$.
    $\widehat{\mathcal{P}} \leftarrow \widehat{\mathcal{P}} \cup \widehat{\mathcal{P}}_c$.
  else
    $\widehat{\mathcal{P}}_c \leftarrow \emptyset$.
  end if
end for
$\mathcal{D}_{\mathrm{pur}} \leftarrow \{(x_i, y_i) \in \mathcal{D}_u \mid i \notin \widehat{\mathcal{P}}\}$.
return $\widehat{\mathcal{P}}, \mathcal{D}_{\mathrm{pur}}$.

0.40%; TrojanNN remains relatively more challenging, with a residual ASR of 5.06%. These results show that PGIT can effectively remove backdoor effects on Tiny-ImageNet-200 without sacrificing normal-task performance.

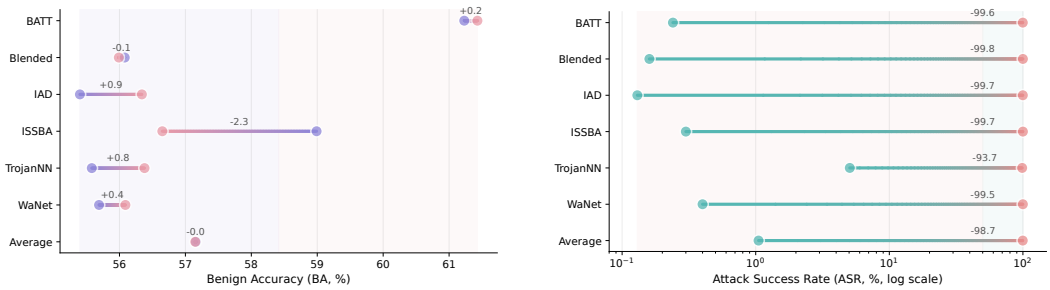


Figure 6: Backdoor purification performance on Tiny-ImageNet-200. The dumbbell plots compare the attacked model (Before) and the purified model (After). The left figure reports benign accuracy (BA), while the right figure reports attack success rate (ASR).

Table 6: Detection and post-purification performance (%) on CIFAR-10 with CLIP-RN50.

Attack ↓	No Defense		Detection		After Purification	
	BA	ASR	TPR	FPR	BA	ASR
BATT	95.18	99.84	99.52	0.00	95.14	1.25
Blended	94.38	100.00	100.00	0.03	94.18	0.22
IAD	95.90	99.64	97.60	0.35	94.46	0.74
ISSBA	95.78	99.85	99.86	< 0.01	94.21	0.52
PhysicalBA	92.95	94.40	77.30	0.03	95.27	85.85
TrojanNN	94.31	99.51	98.50	< 0.01	93.91	90.76
WaNet	93.12	79.63	95.26	0.46	94.09	2.02

B.2 Generalization to CLIP-RN50

In our main experiments, we adopt ResNet-18 as the backbone architecture. To further evaluate the generalizability of PGIT across different model architectures, we replace the backbone with CLIP-RN50 and conduct experiments on CIFAR-10. As shown in Table 6, PGIT maintains strong detection performance against most attacks, achieving an average TPR of 95.43% and an average FPR of approximately 0.13%. Specifically, the TPR exceeds 95% for BATT, Blended, IAD, ISSBA, and WaNet, while the post-purification ASR is reduced to at most 2.02% with largely preserved BA. In contrast, the TPR on PhysicalBA drops to 77.30%, and relatively high post-purification ASRs remain for PhysicalBA and TrojanNN. Overall, PGIT remains effective against most attacks on CLIP-RN50, demonstrating its generalizability across model architectures while also revealing room for further improvement on certain attacks.

B.3 Defense-Aware Adaptive Attacks

We further evaluate PGIT against three defense-aware adaptive attacks that explicitly attempt to conceal the learning signals used by our detector. All attacks optimize a universal 6×6 patch using a clean surrogate ResNet-18 trained on CIFAR-10. The surrogate checkpoints are collected at epochs $\mathcal{E} = \{5, 20, 40, 60\}$. The attacker knows the feature layers and learning signals used by PGIT, but has no access to the defender's trusted reference set or its actual replay model. The poisoning rate is set to 10%, and class 0 is used as the target class.

Let $\widetilde{x} = \mathcal{T}(x; \delta)$ denote a source image stamped with the learnable trigger δ, and let x^t be a clean target-class image available to the attacker. The classification objective is

$$\mathcal{L}_{\text{cls}} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \text{CE}(f_e(\widetilde{x}), y_t)\,, \tag{17}$$

Table 7: Detection and purification performance (%) under defense-aware adaptive attacks on CIFAR-10 with ResNet-18. "Before" and "After" denote the attacked and purified-retraining models, respectively.

Attack	TPR	FPR	$\mathrm{BA}_{\mathrm{Before}}$	$\mathrm{ASR}_{\mathrm{Before}}$	$\mathrm{BA}_{\mathrm{After}}$	$\mathrm{ASR}_{\mathrm{After}}$
Adaptive	99.18	0.10	92.07	100.00	91.76	3.42
Trajectory-aware	**100.00**	0.57	91.76	100.00	90.90	**3.09**
Gradient-aware	**100.00**	0.22	91.44	100.00	90.91	4.00

which preserves the trigger–target association. To approximate the trajectory signal used in Stage I, we define

$$\mathcal{L}_{\mathrm{traj}} = \frac{1}{|\mathcal{E}||\mathcal{L}|} \sum_{e\in\mathcal{E}} \sum_{\ell\in\mathcal{L}} \Big[1 - \cos\Big(\phi_e^{(\ell)}(\widetilde{x}), \phi_e^{(\ell)}(x^t)\Big)\Big], \tag{18}$$

where $\phi_e^{(\ell)}(\cdot)$ denotes the pooled representation at layer ℓ of surrogate checkpoint e. Minimizing this loss encourages triggered samples to follow the representation evolution of benign target-class samples.

To target Stage II, we construct a surrogate of PGIT's activation-gradient representation:

$$\mathbf{g}_e(x; y_t) = \mathrm{Norm}\Big[\mathrm{Pool}\Big(\nabla_{h_e^{(\ell_g)}(x)} [f_e(x)]_{y_t}\Big)\Big], \tag{19}$$

where $[f_e(x)]_{y_t}$ is the target-class logit and $\mathrm{Pool}(\cdot)$ follows the spatial pooling and normalization used by PGIT. The temporal gradient-alignment loss is

$$\mathcal{L}_{\mathrm{grad}} = \frac{1}{|\mathcal{E}|} \sum_{e\in\mathcal{E}} \big[1 - \cos\big(\mathbf{g}_e(\widetilde{x}; y_t), \mathbf{g}_e(x^t; y_t)\big)\big]. \tag{20}$$

Based on these objectives, we consider the following attacks:

- **Adaptive**: our original hybrid surrogate attack, which combines multi-checkpoint feature matching with a final-checkpoint cross-entropy-gradient proxy;
- **Trajectory-aware**: an attack targeting Stage I by optimizing $\mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{traj}} + \lambda_{\mathrm{tv}}\mathcal{L}_{\mathrm{tv}}$;
- **Gradient-aware**: an attack targeting Stage II by optimizing $\mathcal{L}_{\mathrm{cls}} + \lambda_g\mathcal{L}_{\mathrm{grad}} + \lambda_{\mathrm{tv}}\mathcal{L}_{\mathrm{tv}}$.

We use $\lambda_g = 0.1$ and $\lambda_{\mathrm{tv}} = 10^{-4}$. Each trigger is optimized for 500 steps using Adam with a learning rate of 0.05.

As shown in Table 7, all three attacks successfully implant their backdoors before defense, achieving a strict ASR of 100%. Nevertheless, PGIT detects at least 99.18% of the poisoned samples, while maintaining an FPR below 0.57%. Retraining on the purified datasets reduces the strict ASR to at most 4.00%, with a benign-accuracy decrease of no more than 0.86 percentage points relative to the corresponding attacked model. The trajectory-aware attack induces the highest FPR but causes no poisoned samples to be missed, whereas the gradient-aware attack retains the highest post-purification ASR. These results demonstrate that PGIT remains effective when the attacker explicitly attempts to imitate either the representation trajectory or the temporal activation-gradient behavior of benign target-class samples.

C Additional Analysis of PGIT

C.1 Comparison of Stage-I Representations

To examine the role of information-learning trajectories in Stage I, we keep the replay model, training data, trusted reference set, candidate-selection rule, and Stage II unchanged, while replacing only the sample-wise signals used for Stage-I class localization. Specifically, **MI** characterizes the information dependencies among the input, internal representations, and predictions, together with their

Table 8: Comparison of different Stage-I representations on CIFAR-10 with ResNet-18 (TPR/FPR, %). All variants share the same candidate-selection rule and Stage-II attribution procedure.

Attack	MI		Loss		Confidence		Forgetting		Gradient Norm		Final Feature	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
BadNets	97.46	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	97.46	0.06
Blended	100.00	0.35	0.00	0.00	0.00	0.00	100.00	0.35	0.00	0.00	100.00	0.35
BATT	98.80	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	98.80	0.016
IAD	96.82	0.12	96.82	0.119	0.00	0.00	96.82	2.21	96.82	0.119	96.82	3.46
ISSBA	99.78	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.78	0.011
LC	100.00	0.03	100.00	0.031	100.00	0.031	0.00	0.00	0.00	0.00	100.00	2.47
SIG	99.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.00	0.00

evolution during training; **Loss** and **Confidence** record the cross-entropy loss and the predicted probability of the observed label, respectively; **Forgetting** captures cumulative transitions from correct to incorrect classification across consecutive checkpoints; and **Gradient Norm** measures the magnitude of the gradient of the observed-label loss with respect to the internal representation, without retaining gradient-direction information. These dynamic signals are collected from the same replay model at 15 checkpoints and combined with their adjacent-stage differences to construct temporal trajectories. We additionally consider **Final Feature**, which measures the class-conditional feature deviation only at the final checkpoint and therefore serves as a static endpoint baseline. All variants use the same class-conditional reference calibration, Top-3 candidate-selection rule, and Stage-II gradient attribution, ensuring that the comparison isolates the effect of the Stage-I representation.

As shown in Table 8, different Stage-I representations exhibit markedly different levels of robustness across attacks. ❶ **Simple scalar learning dynamics fail to consistently localize the target class.** Loss, Confidence, Forgetting, and Gradient Norm characterize sample difficulty, label confidence, forgetting behavior, and gradient magnitude, respectively, but exhibit highly inconsistent behavior across attacks. For BadNets, BATT, ISSBA, and SIG, several alternative signals yield 0% TPR and 0% FPR because Stage I fails to include the target class in the candidate set. ❷ **Different scalar signals are effective only for specific attacks.** Loss succeeds under IAD and LC, Forgetting under Blended and IAD, Confidence mainly under LC, and Gradient Norm primarily under IAD, indicating that the anomalies captured by individual training statistics are strongly attack-dependent. ❸ **MI trajectories provide more consistent Stage-I localization across attacks.** Across all seven evaluated attacks, the MI-based representation successfully passes the target class to Stage II and achieves final TPRs of 96.82%–100.00% while maintaining low FPRs. Since all variants share the same Stage-II attribution procedure, these differences primarily reflect their Stage-I localization ability. ❹ **Training dynamics provide more reliable evidence than relying only on the final model state.** Although Final Feature also identifies the target class across all evaluated attacks, it produces substantially higher FPRs under IAD and LC, reaching 3.462% and 2.470%, respectively, compared with 0.119% and 0.031% for MI. Overall, these results suggest that modeling learning signals across training provides more stable class-level evidence than either simple scalar statistics or a single endpoint representation.

C.2 Impact of Instance-Level Gradient Attribution

To analyze the contribution of each Stage-II component, we fix the Stage-I CITD-based class localization and vary only the instance-level detection strategy within the candidate classes. As shown in Table 9, ❶ using Stage I alone achieves a TPR of 100% across all five attacks, but incurs a high average FPR of 39.33%. This indicates that class-level localization can identify potentially poisoned classes but cannot distinguish poisoned from benign samples within them. ❷ Without trusted gradient anchors, detection becomes substantially less stable, with the average TPR dropping to 79.49% and the average FPR increasing to 5.32%; notably, the TPR on BATT is only 1.20%. This suggests that unsupervised gradient partitioning can reveal intra-class structure but cannot reliably determine the identity of each partition. ❸ Introducing trusted gradient anchors markedly improves both detection strategies. Using only the final checkpoint achieves an average TPR of 98.75%, while keeping the FPR below 0.02% for all attacks, showing that benign learning directions provide

Table 9: Ablation of Stage-II gradient attribution on CIFAR-10 with ResNet-18 (TPR/FPR, %). CA denotes clean-anchor orientation, while Full PGIT combines temporal gradient partitioning with CA. Non-zero FPR values below 0.01% are reported as < 0.01.

Attack	Stage-I Only		Temporal (w/o CA)		Final + CA		Full PGIT	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
BadNets	100.00	39.98	97.46	6.97	94.40	< 0.01	97.46	0.06
BATT	100.00	39.92	1.20	9.98	99.60	0.00	98.80	0.02
Blended	100.00	39.98	100.00	2.00	99.90	0.02	100.00	0.35
ISSBA	100.00	39.92	99.78	0.01	99.98	< 0.01	99.78	0.01
SIG	100.00	36.84	99.00	7.64	99.88	< 0.01	99.00	0.00

Table 10: Sensitivity to the candidate-class budget k on CIFAR-10 with ResNet-18 (TPR/FPR, %). Here, k denotes the number of Top-k candidate classes independently selected by the radial and directional branches before taking their union.

Attack	Top-1		Top-2		Top-3		Top-4		Top-5	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
BadNets	97.46	0.06	97.46	0.06	97.46	0.06	97.46	0.06	97.46	0.06
BATT	98.80	0.02	98.80	0.02	98.80	0.02	98.80	0.02	98.80	0.02
Blended	100.00	0.35	100.00	0.35	100.00	0.35	100.00	0.35	100.00	0.35
ISSBA	0.00	0.00	99.78	0.01	99.78	0.01	99.78	0.01	99.78	0.01
LC	100.00	0.03	100.00	0.03	100.00	0.03	100.00	0.03	100.00	0.03
Refool	0.00	0.00	0.00	0.00	99.38	0.09	99.38	0.09	99.38	0.09
SIG	99.00	0.00	99.00	0.00	99.00	0.00	99.00	0.00	99.00	0.00
TrojanNN	99.14	0.20	99.14	0.20	99.14	0.20	99.14	0.20	99.14	0.20

effective semantic calibration for gradient partitioning. ❹ Incorporating gradient geometry across training stages further yields an average TPR of 99.01% with an average FPR of 0.09%. Compared with using only the final checkpoint, it improves the average TPR by 0.26 percentage points, including an increase from 94.40% to 97.46% on BadNets, at the cost of slightly higher FPRs on some attacks. Overall, trusted anchors are essential for instance-level discrimination, while temporal gradient geometry provides complementary information about the evolution of backdoor learning.

C.3 Impact of the Candidate-Class Budget k

To evaluate the sensitivity of PGIT to the candidate-class budget, we vary the number of Top-k candidate classes selected independently by the radial and directional branches and evaluate the results on CIFAR-10 and Tiny-ImageNet-200, as shown in Tables 10 and 11, respectively. ❶ **An overly small candidate budget may fail to cover the target class.** On CIFAR-10, when $k = 1$, the target classes of ISSBA and Refool are not included in the candidate set, resulting in TPRs of 0%. When k is increased to 2 and 3, their TPRs rise to 99.78% and 99.38%, respectively. Similarly, on Tiny-ImageNet-200, the target class of PhysicalBA is not covered when $k \leq 10$, whereas its TPR increases to 91.62% at $k = 15$. ❷ **Once the target class is included, PGIT remains stable with respect to the specific choice of k.** On CIFAR-10, the TPR and FPR of all attacks remain nearly unchanged when $k \geq 3$. On Tiny-ImageNet-200, BATT, Blended, IAD, ISSBA, TrojanNN, and WaNet also exhibit nearly identical TPRs across different values of k. These results indicate that PGIT does not require precise tuning of k within a reasonable candidate-budget range. ❸ **Further enlarging the candidate set generally provides no additional detection benefit.** Instead, as more classes are passed to the second stage, the FPR slightly increases for several attacks, including Blended, IAD, ISSBA, TrojanNN, and WaNet. Overall, k mainly controls the trade-off between target-class coverage and subsequent detection cost, while PGIT remains robust to this hyperparameter once the target class is adequately covered.

Table 11: Sensitivity to the candidate-class budget k on Tiny-ImageNet-200 with ResNet-18 (TPR/FPR, %). Here, k denotes the number of Top-k candidate classes independently selected by the radial and directional branches before taking their union. Non-zero FPR values below 0.01% are reported as < 0.01.

Attack	Top-5		Top-10		Top-15		Top-20	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
BATT	99.96	0.01	99.96	0.01	99.96	0.01	99.96	0.01
Blended	100.00	< 0.01	100.00	0.13	100.00	0.13	100.00	0.13
IAD	100.00	< 0.01	100.00	< 0.01	100.00	0.09	100.00	0.09
ISSBA	99.98	< 0.01	99.98	< 0.01	99.98	< 0.01	99.98	0.23
PhysicalBA	0.00	0.00	0.00	< 0.01	91.62	0.01	91.62	0.01
TrojanNN	86.52	< 0.01	86.52	< 0.01	86.52	0.01	86.52	0.10
WaNet	100.00	< 0.01	100.00	< 0.01	100.00	< 0.01	100.00	0.18

Table 12: Sensitivity to the gradient-separation threshold τ_z on CIFAR-10 with ResNet-18 (TPR/FPR, %).

Attack	$\tau_z = 1$		$\tau_z = 1$		$\tau_z = 1$		$\tau_z = 1$		$\tau_z = 1$	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
BATT	98.80	0.02	98.80	0.02	98.80	0.02	98.80	0.02	98.80	0.02
BadNets	97.46	0.06	97.46	0.06	97.46	0.06	97.46	0.06	97.46	0.06
Blended	100.00	0.35	100.00	0.35	100.00	0.35	100.00	0.35	100.00	0.35
ISSBA	99.78	0.01	99.78	0.01	99.78	0.01	99.78	0.01	99.78	0.01
LC	100.00	0.03	100.00	0.03	100.00	0.03	100.00	0.03	100.00	0.03
Refool	99.38	0.09	99.38	0.09	99.38	0.09	99.38	0.09	99.38	0.09
SIG	99.00	5.35	99.00	0.00	99.00	0.00	99.00	0.00	99.00	0.00
TrojanNN	99.14	0.20	99.14	0.20	99.14	0.20	99.14	0.20	99.14	0.20

C.4 Impact of the Gradient-Separation Threshold τ_z

To evaluate the sensitivity of PGIT to the gradient-separation threshold in the second stage, we vary z from 1 to 5, with the results reported in Table 12. ❶ **PGIT remains highly stable across a broad range of threshold values.** For BATT, BadNets, Blended, ISSBA, LC, Refool, and TrojanNN, both TPR and FPR remain unchanged for $z = 1$–5. For example, Blended consistently achieves 100.00% TPR and 0.35% FPR, while Refool maintains 99.38% TPR and 0.09% FPR across all settings. ❷ **The threshold mainly controls the conservative filtering of weakly separated gradient structures.** For SIG, the TPR remains at 99.00% for all values of z, whereas the FPR reaches 5.35% when $z = 1$ and drops to 0.00% once $z \geq 2$. This indicates that an overly small threshold may accept insufficiently separated gradient structures and consequently introduce additional false removals. ❸ **PGIT does not require precise tuning of z.** Once $z \geq 2$, the detection results remain essentially stable across all evaluated attacks. We therefore adopt $z = 3.0$ as the default setting, providing a conservative criterion for gradient-separation evidence while preserving high detection performance.

C.5 Impact of the Reference-Gradient Support Threshold τ_s

To evaluate the sensitivity of PGIT to the reference-gradient support threshold, we vary τ_s from 0.6 to 0.9, with the results reported in Table 13. ❶ **The TPR remains stable across different values of τ_s.** For all evaluated attacks, varying τ_s does not reduce the TPR. For example, BadNets and Blended consistently achieve TPRs of 97.46% and 100.00%, respectively, across all settings, indicating that the reference-support threshold does not compromise poisoned-sample detection within this range. ❷ **A smaller τ_s may introduce additional false positives.** When $\tau_s = 0.6$ or 0.7, the FPRs of BadNets and Blended reach 3.18% and 2.00%, respectively. Increasing τ_s to 0.8 reduces their FPRs to 0.06% and 0.35%, while leaving the TPR unchanged. This suggests that stronger reference-

Table 13: Sensitivity to the reference-gradient support threshold τ_s on CIFAR-10 with ResNet-18 (TPR/FPR, %).

Attack	$\tau_s = 0.6$		$\tau_s = 0.7$		$\tau_s = 0.8$		$\tau_s = 0.9$	
	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
BATT	98.80	0.02	98.80	0.02	98.80	0.02	98.80	0.02
BadNets	97.46	3.18	97.46	3.18	97.46	0.06	97.46	0.06
Blended	100.00	2.00	100.00	2.00	100.00	0.35	100.00	0.35
ISSBA	99.78	0.01	99.78	0.01	99.78	0.01	99.78	0.01
LC	100.00	0.03	100.00	0.03	100.00	0.03	100.00	0.03
Refool	99.38	0.09	99.38	0.09	99.38	0.09	99.38	0.09
SIG	99.00	0.00	99.00	0.00	99.00	0.00	99.00	0.00
TrojanNN	99.14	0.20	99.14	0.20	99.14	0.20	99.14	0.20

Table 14: Computational overhead comparison of different backdoor defenses on CIFAR-10 with ResNet-18. Running time is reported in minutes and peak memory consumption is measured in GiB.

Method	Auxiliary Training	Defense Procedure	Total Time	Peak Memory
PGIT	14.82 min	93.78 min	108.60 min	2.40 GiB
FLARE	31.96 min	4.25 min	36.21 min	0.92 GiB
CT	–	72.96 min	72.96 min	1.95 GiB
HARVEY	–	76.23 min	76.23 min	2.00 GiB
ASD	–	153.92 min	153.92 min	4.48 GiB
D-ST	–	165.04 min	165.04 min	8.33 GiB
V&B	–	258.51 min	258.51 min	10.71 GiB

gradient support helps avoid incorrect partition orientation. ❸ **The detection results become stable once** $\tau_s \geq 0.8$**.** Increasing τ_s further from 0.8 to 0.9 leaves both TPR and FPR unchanged for all attacks. We therefore adopt $\tau_s = 0.8$ as the default setting, providing reliable reference guidance while preserving high detection performance.

C.6 Computational Overhead Analysis

Table 14 summarizes the computational overhead of different defense paradigms on CIFAR-10 with ResNet-18. ❶ **PGIT introduces additional but controlled computation for modeling training dynamics.** PGIT requires 108.60 minutes in total, where the major cost comes from trajectory construction and temporal activation-gradient analysis. This overhead is necessary for capturing the class-level and instance-level learning dynamics exploited by PGIT, rather than relying only on static model states. ❷ **PGIT remains computationally competitive with in-training defenses.** Compared with methods that modify the optimization process throughout training, PGIT avoids continuous intervention during model optimization and requires lower runtime and memory consumption than most in-training defenses. In particular, PGIT uses only 2.40 GiB peak memory, showing a relatively low resource requirement despite modeling additional training signals. ❸ **The additional cost enables richer poisoned-sample analysis.** Compared with lightweight detection pipelines such as FLARE, PGIT incurs extra computation due to its two-stage design, which combines trajectory-based candidate localization with gradient-based sample attribution. This trade-off allows PGIT to exploit training dynamics for more comprehensive poisoned-sample detection while maintaining acceptable computational overhead.