POISONING ATTACK FRAMEWORK AGAINST TEST-TIME ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-time adaptation (TTA) enables batch-wise updates by incorporating distributional characteristics from unlabeled test batches to enhance model generalization. The co-batching of poisoned samples can mislead a target model after adaptation, forcing attacker-intended predictions on benign co-batched test samples. Nonetheless, most attacks study this poisoning risk under extra requirements, such as access to real benign co-batch samples, the target model or queries. Such requirements tailor attacks to specific targets and hinder assessments of TTA poisoning risks when target-side access is unavailable. In this paper, we propose PCAC, a targeted poisoning framework without relying on the above requirements. PCAC departs from prior attacks by exploiting distribution-shifted auxiliary data for surrogate-side crafting. PCAC augments auxiliary data by emphasizing salient regions, creating a diverse proxy set that provides influential co-batch contexts. PCAC then induces a target-confident effect and smooths surrogate-side crafting to discourage center-specific and state-sensitive solutions. With proxy diversity and flatness-enhanced target-effect crafting, PCAC generates transferable poisoned test samples. Experiments across diverse TTA settings show that PCAC achieves $2.47\times$ ASR of prior attacks without target-side access during crafting.

1 INTRODUCTION

Test-time adaptation (TTA) adapts source-trained image classifiers to target domains to improve predictions under distribution shifts (Wang et al., 2021; Liang et al., 2025). Instead of relying on supervised updates from source training data or target test labels (Zhao et al., 2023), batch-wise TTA methods commonly derive adaptation signals from co-batched test samples and update the model through test-batch statistics or unsupervised objectives (Zhang et al., 2022). As demands for cross-domain generalization grow, TTA has gained prominence for bridging the domain gap in image-classification applications (Nado et al., 2020; Liang et al., 2025; Wang et al., 2025).

While TTA allows co-batched test samples to jointly contribute to model adaptation, it remains vulnerable to poisoned inputs, where subtle adversarial perturbations to attacker-controlled samples in the test batch cause the adapted target model to misclassify benign samples (Cong et al., 2024; Su et al., 2025). The existence of such poisoned inputs reveals security risks in deploying TTA, enabling attackers to manipulate predictions toward a desired class, as their controlled samples intentionally induce targeted effects that meaningfully influence the target model through the co-batched adaptation process (Wu et al., 2023). Despite advancements in crafting poisoned inputs, they have to pre-set some restricted assumptions. For example, DIA requires access to the real benign test samples and specific target model (Wu et al., 2023). In practice, benign inputs in the co-batch may be difficult to fetch due to privacy concerns, and the target model encodes proprietary expertise, making its details inaccessible. Other attacks, such as BLE and NHE, query the target model to approximate its adaptation behavior with a surrogate model (Su et al., 2025). However, repeated queries cost much and may trigger suspicion. Moreover, these assumptions typically tailor the crafted inputs to specific targets, which tend to apply poorly in target-inaccessible settings (Qin et al., 2022; Xu et al., 2023).

Motivated by the above issues, we propose PCAC, a novel poisoning attack framework that exploits distribution-shifted auxiliary data and a publicly available pretrained surrogate model. PCAC crafts poisoned inputs that manipulate predictions on benign co-batched samples toward the desired class after adaptation. PCAC consists of two modules: **P**roxy **C**onstruction and **A**dversarial **C**rafting. The

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.

proxy construction module generates a saliency-emphasized proxy input set from shifted auxiliary data to provide influential co-batch contexts during surrogate crafting. The adversarial crafting module induces a target-confident effect against co-batch contexts and applies dual-neighborhood crafting to discourage locally sharp and surrogate parameter state-sensitive solutions. Proxy diversity and flatness-enhanced target-effect optimization enable surrogate-side crafting of poisoned inputs that manipulate benign co-batched predictions under adapted target models.

The main contributions of this paper are twofold.

- We propose PCAC, a targeted poisoning framework without access to benign co-batches, target model, or queries. PCAC designs saliency-emphasized proxy construction with flatness-enhanced crafting to generate poisoned inputs for targeted misclassification.
- Evaluations across settings confirm that PCAC manipulates benign predictions after adaptation, achieving 84.68% ASR and outperforming prior attacks by up to 35.48%. Results against countermeasures show that PCAC retains at least 74.24% of the no-defense ASR.

2 Problem Statement

2.1 Test-Time Adaptation

We cast TTA in the multi-class image-classification setting under distribution shifts, following (Wang et al., 2021; Wu et al., 2023). Let $\mathcal{X} \subset \mathbb{R}^d$ be the d-dimensional input space and $\mathcal{Y} = \{1, \ldots, C\}$ be the C-class label space. Let $\mathcal{D}_{\text{src}} = \{(x_i, y_i)\}_{i=1}^{N_{\text{src}}} \subset \mathcal{X} \times \mathcal{Y}$ be training dataset, and let $\mathcal{D}_{\text{test}} = \{x'_i\}_{i=1}^{N_{\text{test}}} \subset \mathcal{X}$ be unlabeled test dataset. A model $f_\theta : \mathcal{X} \to \mathbb{R}^C$ is pre-trained on $\mathcal{D}_{\text{src}}$ and then deployed on distribution-shifted test data (Liang et al., 2025).

Following the formulation of batch-level adaptation step of TTA (Wu et al., 2023; Rifat et al., 2025), given a test batch $\mathcal{B} = \{x'_i\}_{i=1}^{N_b} \subset \mathcal{D}_{\text{test}}$, TTA adapts f_θ to a batch-specific state $f_{\theta'}$ without label supervision by solving:

$$\theta'_A(\mathcal{B}) = \arg\min_{\theta_A} \mathcal{L}_{\text{tta}}\big(\mathcal{B}; \theta_A, \theta'_B(\mathcal{B}), \theta_F\big),$$
$$\theta'(\mathcal{B}) = \theta'_A(\mathcal{B}) \cup \theta'_B(\mathcal{B}) \cup \theta_F.$$

Here, θ'_A denotes the adaptable parameters, θ_F denotes the fixed parameters, and $\theta'_B(\mathcal{B})$ denotes batch-conditioned statistics, e.g., $\{\mu(\mathcal{B}), \sigma^2(\mathcal{B})\}$ (Nado et al., 2020). $\mathcal{L}_{\text{tta}}(\cdot)$ denotes the unsupervised TTA loss, e.g., entropy minimization and invariance regularization (Zhang et al., 2022). Predictions for samples in $\mathcal{B}$ are then made with the resulting batch-adapted target model $f_{\theta'}$, i.e., $\hat{y}_i = \arg\max_c [f_{\theta'}(x'_i)]_c$ for $x'_i \in \mathcal{B}$, i.e., $f_\theta \xrightarrow{\text{TTA}(\mathcal{B})} f_{\theta'} \to$ predict on $\mathcal{B}$.

2.2 Targeted Poisoning Attacks in TTA

Following (Wu et al., 2023; Cong et al., 2024), we consider each target test batch $\mathcal{B}$ as an individual trial in episodic evaluation, consisting of attacker-controlled set $\mathcal{B}_a$ and benign set $\mathcal{B}_b$. Within each trial, target model adapts f_θ on $\mathcal{B} = \mathcal{B}_a^{\boldsymbol{\delta}} \cup \mathcal{B}_b$ to obtain the adapted target model $f_{\theta'}$. We assume that $\mathcal{A}$ aims to misclassify benign samples $x'_i \in \mathcal{B}_b$ as specified label ℓ_{tar} under $f_{\theta'}$ (Rifat et al., 2025).

Definition of Targeted Poisoning Attacks. We adopt the formulation in (Wu et al., 2023) to define the target-side objective of targeted poisoning:

$$\mathcal{J}_t\left(\boldsymbol{\delta}; \mathcal{B}_b, \theta\right) = \frac{1}{|\mathcal{B}_b|} \sum_{x'_i \in \mathcal{B}_b} \mathcal{L}_{\text{atk}}\left(x'_i, \ell_{\text{tar}}; \theta'\left(\boldsymbol{\delta}; \mathcal{B}_b, \theta\right)\right),$$
$$\mathcal{B} = \mathcal{B}_a^{\boldsymbol{\delta}} \cup \mathcal{B}_b, \quad \mathcal{B}_a^{\boldsymbol{\delta}} \cap \mathcal{B}_b = \varnothing,$$
$$\theta' = \theta'_A \cup \theta'_B \cup \theta_F, \quad \theta'_B = \{\mu(\mathcal{B}), \sigma^2(\mathcal{B})\},$$
$$\theta'_A = \arg\min_{\theta_A} \mathcal{L}_{\text{tta}}\left(\mathcal{B}; \theta_A, \theta'_B, \theta_F\right).$$

where $\mathcal{L}_{\text{atk}}(\cdot)$ is the loss with specified label ℓ_{tar}, and $\mathcal{B}_a^{\boldsymbol{\delta}} = \{x'_i + \delta_i \mid x'_i \in \mathcal{B}_a\}$. The objective is to minimize $\mathcal{J}_t(\boldsymbol{\delta}; \mathcal{B}_b, \theta)$ over the feasible perturbation set with $\|\delta_i\|_p \leq \Lambda, \ \forall i$.

Attacker's Capability & Knowledge. We assume that $\mathcal{A}$ can construct proxy co-batches and craft poisoned inputs through gradient-based perturbation optimization on a surrogate model f_ω

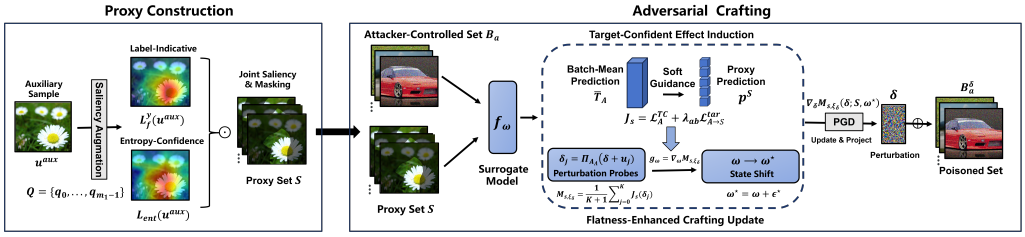


Figure 1: Overview of PCAC.

(Qin et al., 2022). Following the common single-level approximation employed by (Wu et al., 2023; Su et al., 2025), the inner gradient-update loop for method-specific adaptation induced by the TTA loss $\mathcal{L}_{\text{tta}}$ is discarded during crafting, while the batch-conditioned normalization statistics in f_ω are recomputed from the mixed co-batch $\mathcal{B}_a^\delta \cup \mathcal{S}$ during the surrogate-side adaptation.

$\mathcal{A}$ lacks access to the target model f_θ, including the model architecture, parameter state θ and adapted state θ', the predictions or gradients obtained through target queries, or specific TTA method deployed by the target. $\mathcal{A}$ also has no access to the real benign samples in $\mathcal{B}_b$ or labeled training data $\mathcal{D}_{\text{src}}$ of target model f_θ. Following prior transfer-based attacks (Cong et al., 2024; Li et al., 2025), $\mathcal{A}$ has access to a publicly available pretrained surrogate model f_ω, without accessing its source-training samples, and labeled auxiliary data $\mathcal{D}_{\text{aux}}$ used to construct the proxy input set $\mathcal{S}$. The auxiliary data $\mathcal{D}_{\text{aux}}$ is distribution-shifted relative to $\mathcal{D}_{\text{src}}$.

3 PCAC: Detailed Construction

As shown in Figure 1, PCAC consists of two modules. The *Proxy Construction* module constructs a saliency-emphasized proxy set from auxiliary data to provide influential co-batches during surrogate crafting. The *Adversarial Crafting* module then optimizes perturbations under proxy contexts through target-effect guidance and flatness-enhanced updates.

3.1 Proxy Construction

Since the real benign co-batch is unavailable during crafting, PCAC constructs diverse proxy inputs that approximate influential co-batch contexts by preserving regions highlighted by label-indicative and entropy-confidence heatmaps.

Inspired by Grad-CAM and saliency-guided input transformation (Gong et al., 2021; Niu et al., 2022), task-relevant and prediction-confident regions in each auxiliary sample $(u^{\text{aux}}, y) \in \mathcal{D}_{\text{aux}}$ are emphasized by saliency-guided masking, and multiple masking levels are used to construct the proxy input set $\mathcal{S}$. We quantify contribution of each feature-map channel of f_ω to the surrogate prediction of (u^{aux}, y) by constructing the label-indicative heatmap $L_f^y(u^{\text{aux}})$:

$$L_f^y(u^{\text{aux}}) = \mathcal{H}\left(\text{ReLU}\left(\textstyle\sum_n \beta_n^y F_n\right)\right),$$

where F_n is the n-th channel of the feature map from the final convolutional layer of f_ω, $\mathcal{H}(\cdot)$ performs bilinear interpolation to align the CAM dimensions with the input resolution, and $\beta_n^y = \frac{1}{ij}\sum_i \sum_j \frac{\partial s_y}{\partial F_n^{ij}}$ represents the contribution weights for each feature map channel to the f_ω derived logit s_y corresponding to the ground-truth label y.

To emphasize confident regions, we then compute the entropy-confidence heatmap $L_{\text{ent}}(u^{\text{aux}})$, highlighting entries associated with lower prediction entropy:

$$L_{\text{ent}}(u^{\text{aux}}) = \mathcal{H}\left(\text{ReLU}\left(\textstyle\sum_n \beta_n^{\text{ent}} F_n\right)\right),$$

where $\beta_n^{\text{ent}} = \frac{1}{ij}\sum_i \sum_j \frac{\partial C_{\text{ent}}}{\partial F_n^{ij}}$ represents the contribution weight of the n-th feature-map channel to the surrogate-derived entropy-confidence score $C_{\text{ent}} = 1 - [-\sum_{c=1}^{C} \hat{p}_c \log \hat{p}_c]/\log C$ (Niu et al., 2023), $\hat{p}_c$ is the prediction probability of class c produced by f_ω, and C is the number of classes. The label-indicative and entropy-confidence heatmaps are fused by normalized multiplicative gating:

$$L_{\text{joint}}^y(u^{\text{aux}}) = \text{Norm}\left(L_f^y(u^{\text{aux}})\right) \odot \text{Norm}\left(L_{\text{ent}}(u^{\text{aux}})\right),$$

where $\odot$ denotes element-wise multiplication, and $\text{Norm}(\cdot)$ denotes min-max normalization.

Entries below the q-th percentile of the joint heatmap are suppressed to extract salient content:

$$C_q^y(u^{\text{aux}}) = u^{\text{aux}} \odot G_q^y(u^{\text{aux}}),$$

where $G_q^y(u^{\text{aux}}) = \mathbb{I}\left(L_{\text{joint}}^y(u^{\text{aux}}) \geq Q_q\left(L_{\text{joint}}^y(u^{\text{aux}})\right)\right)$ is the binary saliency mask, and $Q_q(\cdot)$ denotes the q-th percentile value. The masked content $C_q^y(u^{\text{aux}})$ is then fused with u^{aux} to produce emphasized data:

$$M_q^y(u^{\text{aux}}) = (1-\mu)\cdot u^{\text{aux}} + \mu \cdot C_q^y(u^{\text{aux}}) + \eta,$$

where $\mu \in [0,1]$ is the mixing ratio, and $\eta \sim \mathcal{N}(0, \sigma_\eta^2 I)$ is random noise added for diversity. Since $C_q^y(u^{\text{aux}})$ is generated from u^{aux} according to its ground-truth label y, the original label is still used for $M_q^y(u^{\text{aux}})$ (Gong et al., 2021).

Following diverse input-transformation practices in transferable attacks (Zou et al., 2020; Byun et al., 2022), we duplicate $M_q^y(u^{\text{aux}})$ at varying levels to construct the proxy set $\mathcal{S}$. The percentile set $\mathcal{Q}$ is formulated as:

$$\mathcal{Q} = \{q \mid q \bmod \Delta_q = 0,\ q \in \mathbb{N},\ q < 100\},$$

where q is the percentile threshold for masking, and Δ_q determines the granularity. The proxy input set $\mathcal{S} = \bigcup_{(u^{\text{aux}},y)\in\mathcal{D}_{\text{aux}}} \mathcal{S}_{\text{aux}}(u^{\text{aux}}, y)$ is constructed as:

$$\mathcal{S}_{\text{aux}}(u^{\text{aux}}, y) = \left\{M_q^y(u^{\text{aux}}) \mid q \in \mathcal{Q}\right\}.$$

3.2 Adversarial Crafting

Given f_ω and the proxy set $\mathcal{S}$, a soft target is constructed from the batch-averaged predictive distribution of $\mathcal{B}_a^{\boldsymbol{\delta}}$ against co-batch contexts. The crafting objective is then optimized through a flatness-enhanced update to discourage center-specific and surrogate-state-sensitive solutions.

Target-Confident Effect Induction. Instead of applying cross-entropy with specified label ℓ_{tar} directly to $\mathcal{S}$, we first encourage the poisoned attacker-controlled set $\mathcal{B}_a^{\boldsymbol{\delta}}$ to assign high surrogate probability to ℓ_{tar} under $f_{\omega'}$:

$$\mathcal{L}_A^{\text{TC}}(\boldsymbol{\delta}; \mathcal{S}, \omega) = -\tfrac{1}{|\mathcal{B}_a|}\textstyle\sum_{x_i'\in\mathcal{B}_a}\sum_{k=1}^{C}\mathbb{I}(k=\ell_{\text{tar}})\log\left[p_i^A\right]_k,$$

where $p_i^A = \text{softmax}\left(f_{\omega'(\boldsymbol{\delta};\mathcal{S},\omega)}\left(x_i'+\delta_i\right)\right)$ denotes the predicted class-probability vector of the perturbed attacker-controlled sample $x_i'+\delta_i$. $\omega'(\boldsymbol{\delta};\mathcal{S},\omega)$ denotes the surrogate-adapted state obtained by recomputing normalization statistics on the batch $\mathcal{B}_a^{\boldsymbol{\delta}} \cup \mathcal{S}$ (Wu et al., 2023). The index k denotes the class label, and $[p_i^A]_k$ denotes the k-th class probability.

Following (Yun et al., 2020), we then use the batch-averaged prediction $\bar{T}_A(\boldsymbol{\delta}; \mathcal{S}, \omega) = \frac{1}{|\mathcal{B}_a|}\sum_{x_i'\in\mathcal{B}_a} p_i^A$ as soft guidance for its co-batched inputs $\mathcal{S}$:

$$\mathcal{L}_{A\to S}^{\text{tar}}(\boldsymbol{\delta}; \mathcal{S}, \omega) = -\tfrac{1}{|\mathcal{S}|}\textstyle\sum_{M_q^y\in\mathcal{S}}\sum_{k=1}^{C}\text{sg}\left[\bar{T}_{A,k}\right]\log\left[p^S\left(M_q^y\right)\right]_k,$$

where $\text{sg}[\cdot]$ stops gradients through $\bar{T}_A$ to avoid self-referential alignment (Chen & He, 2021), $\bar{T}_{A,k}$ is the k-th entry of $\bar{T}_A$, and $p^S(M_q^y) = \text{softmax}\left(f_{\omega'(\boldsymbol{\delta};\mathcal{S},\omega)}\left(M_q^y\right)\right)$. Surrogate-side crafting is:

$$\mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega) = \mathcal{L}_A^{\text{TC}}(\boldsymbol{\delta}; \mathcal{S}, \omega) + \lambda_{\text{ab}}\mathcal{L}_{A\to S}^{\text{tar}}(\boldsymbol{\delta}; \mathcal{S}, \omega),$$

where $\lambda_{\text{ab}} \geq 0$ controls the strength of the guidance term.

Flatness-Enhanced Crafting Update. Considering that $\mathcal{J}_s$ is computed from the proxy set $\mathcal{S}$ and surrogate model f_ω, optimizing only at the center perturbation under a single surrogate state may produce overly specific crafting updates (Ge et al., 2023; Qiu et al., 2026). PCAC is designed to account for both perturbation-side variation and surrogate-state sensitivity, motivating the perturbation and parameter neighborhoods, respectively. To formalize this dual-neighborhood motivation, we characterize the surrogate-to-target discrepancy through the crafting gap:

$$\mathcal{G}(\boldsymbol{\delta}) = \left|\mathcal{J}_t(\boldsymbol{\delta}; \mathcal{B}_b, \theta) - \mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega)\right|,$$

where $\mathcal{J}_s$ is the attacker-accessible surrogate-side crafting loss, and $\mathcal{J}_t$ is the unavailable target-side crafting loss. At crafting iteration r, an upper-bound decomposition gives:

$$\mathcal{G}(\boldsymbol{\delta}^r) \leq R_{\mathcal{J},\xi_\delta,s}^{(0)}(\boldsymbol{\delta}^r) + P_{\mathcal{M}}^{\omega}(\boldsymbol{\delta}^r; \boldsymbol{\epsilon}_r^\star) + \mathcal{R}_{\text{res}}(\boldsymbol{\delta}^r; \boldsymbol{\epsilon}_r^\star),$$

where $R_{\mathcal{J},\xi_\delta,s}^{(0)}$, $P_{\mathcal{M}}^{\omega}$, and $\mathcal{R}_{\text{res}}$ denote the surrogate-side perturbation variation, surrogate-state sensitivity, and unavailable residual terms. The residual $\mathcal{R}_{\text{res}}$, defined explicitly in **Appendix** E, collects the unavailable target-side local perturbation variation and shifted perturbation-neighborhood

Algorithm 1: PCAC Crafting

1: **function** PCAC-CRAFTING($f_\omega, \mathcal{B}_a, \mathcal{S}, \ell_{\text{tar}}$)
2: ▷ Λ: perturbation budget; R: crafting iterations; K: perturbation probes
3: ▷ ξ_δ: perturbation-neighborhood radius; ρ_ω: parameter-neighborhood radius; α: step size
4: Initialize $\boldsymbol{\delta}^0 \leftarrow 0$
5: **for** $r = 0, \ldots, R-1$ **do**
6: Sample $\mathbf{u}_0^r = 0$ and $\mathbf{u}_j^r \sim \mathrm{Unif}(B_{\xi_\delta}(0))$, $j = 1, \ldots, K$
7: $\boldsymbol{\delta}_j^r \leftarrow \Pi_{\mathcal{A}_\Lambda}(\boldsymbol{\delta}^r + \mathbf{u}_j^r),\ j = 0, \ldots, K$
8: $\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}^r; \mathcal{S}, \omega) \leftarrow \dfrac{1}{K+1} \displaystyle\sum_{j=0}^{K} \mathcal{J}_s(\boldsymbol{\delta}_j^r; \mathcal{S}, \omega)$
9: $g_\omega^r \leftarrow \nabla_\omega \mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}^r; \mathcal{S}, \omega)$
10: $\boldsymbol{\epsilon}_r^\star \leftarrow \rho_\omega \frac{g_\omega^r}{\|g_\omega^r\|_2 + \tau}$
11: $\omega^{r,\star} \leftarrow \omega + \boldsymbol{\epsilon}_r^\star$
12: ▷ $\boldsymbol{\epsilon}_r^\star$ is fixed when updating $\boldsymbol{\delta}$
13: $\Psi_r \leftarrow \mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}^r; \mathcal{S}, \omega^{r,\star})$
14: $\boldsymbol{\delta}^{r+1} \leftarrow \Pi_{\mathcal{A}_\Lambda}\left(\boldsymbol{\delta}^r - \alpha \nabla_{\boldsymbol{\delta}^r} \Psi_r\right)$
15: **end for**
16: $\boldsymbol{\delta}^\star \leftarrow \boldsymbol{\delta}^R$
17: **return** $\mathcal{B}_a^{\boldsymbol{\delta}^\star} = \{x_i' + \delta_i^\star \mid x_i' \in \mathcal{B}_a\}$
18: **end function**

surrogate-to-target discrepancy. Accordingly, the decomposition identifies the attacker-accessible surrogate-side sensitivities targeted by PCAC while retaining the target-dependent residual explicitly, and serves as optimization motivation.

Specifically, at crafting iteration r, we first sample local perturbation probes within a ball of radius ξ_δ, and compute the empirical perturbation-neighborhood crafting loss $\mathcal{M}_{s,\xi_\delta}$ by averaging over the center point $\mathbf{u}_0^r = 0$ and K sampled local probes $\mathbf{u}_j^r \sim \mathrm{Unif}(B_{\xi_\delta}(0))$, following (Ge et al., 2023):

$$\mathcal{M}_{s,\xi_\delta}\left(\boldsymbol{\delta}^r; \mathcal{S}, \omega\right) = \tfrac{1}{K+1} \textstyle\sum_{j=0}^{K} \mathcal{J}_s\left(\boldsymbol{\delta}_j^r; \mathcal{S}, \omega\right),$$

where $\boldsymbol{\delta}^r$ denotes the current perturbation at crafting iteration r, $B_{\xi_\delta}(0) = \{\mathbf{u} : \|\mathbf{u}\|_p \le \xi_\delta\}$ is the perturbation-neighborhood ball, $\boldsymbol{\delta}_j^r = \Pi_{\mathcal{A}_\Lambda}(\boldsymbol{\delta}^r + \mathbf{u}_j^r)$ denotes the probed perturbation, $j = 0$ corresponds to the center point and $j = 1, \ldots, K$ indexes the sampled local probes, $\mathcal{A}_\Lambda = \{\boldsymbol{\delta} : \|\delta_i\|_p \le \Lambda, \forall x_i' \in \mathcal{B}_a\}$ is the perturbation set, Λ is the budget, and $\Pi_{\mathcal{A}_\Lambda}(\cdot)$ is the projection operator. Under these empirical probes, the perturbation-side term $R^{(0)}_{\mathcal{J},\xi_\delta,s}$ measures the local variation of the surrogate loss around the current perturbation:

$$R^{(0)}_{\mathcal{J},\xi_\delta,s}\left(\boldsymbol{\delta}^r\right) = \tfrac{1}{K+1} \textstyle\sum_{j=0}^{K} \left|\mathcal{J}_s\left(\boldsymbol{\delta}_j^r; \mathcal{S}, \omega\right) - \mathcal{J}_s\left(\boldsymbol{\delta}^r; \mathcal{S}, \omega\right)\right|.$$

The center-included neighborhood average $\mathcal{M}_{s,\xi_\delta}$ discourages solutions whose nearby probes $\boldsymbol{\delta}_j^r$ incur high surrogate loss $\mathcal{J}_s(\boldsymbol{\delta}_j^r; \mathcal{S}, \omega)$. This construction favors perturbations with low average surrogate crafting loss over the center point $\boldsymbol{\delta}^r$ and its local probes $\{\boldsymbol{\delta}_j^r\}_{j=1}^K$. Since the non-negative-loss relaxation gives:

$$R^{(0)}_{\mathcal{J},\xi_\delta,s}\left(\boldsymbol{\delta}^r\right) \le \mathcal{J}_s\left(\boldsymbol{\delta}^r; \mathcal{S}, \omega\right) + \mathcal{M}_{s,\xi_\delta}\left(\boldsymbol{\delta}^r; \mathcal{S}, \omega\right),$$

incorporating $\mathcal{M}_{s,\xi_\delta}$ into the crafting update places optimization pressure on an accessible upper proxy of $R^{(0)}_{\mathcal{J},\xi_\delta,s}$. Because the center probe $\mathbf{u}_0^r = 0$ is included in $\mathcal{M}_{s,\xi_\delta}$, the optimized neighborhood objective also explicitly includes the center loss appearing in this relaxation. This perturbation-neighborhood component is implemented in Lines 6–8 of Algorithm 1.

Building on perturbation-neighborhood averaging that accounts for local variation around $\boldsymbol{\delta}^r$, we further construct a locally shifted surrogate state $\omega^{r,\star}$ to account for parameter-neighborhood variation. We compute the parameter-side gradient g_ω^r of the perturbation-neighborhood crafting loss $\mathcal{M}_{s,\xi_\delta}$ with respect to the surrogate parameters ω, following parameter-space flatness optimization (Foret et al., 2021; Chen et al., 2025): $g_\omega^r = \nabla_\omega \mathcal{M}_{s,\xi_\delta}\left(\boldsymbol{\delta}^r; \mathcal{S}, \omega\right)$. We then construct the local parameter offset along the normalized direction of g_ω^r, and compute the shifted parameter $\omega^{r,\star}$ as:

$$\boldsymbol{\epsilon}_r^\star = \rho_\omega \frac{g_\omega^r}{\|g_\omega^r\|_2 + \tau}, \quad \omega^{r,\star} = \omega + \boldsymbol{\epsilon}_r^\star,$$

where ρ_ω is the parameter-neighborhood radius, and $\tau > 0$ is a small constant for normalization. The surrogate-state sensitivity term $P^\omega_{\mathcal{M}}$ measures the variation of the perturbation-neighborhood objective under the local surrogate-state shift:

$$P^\omega_{\mathcal{M}}\left(\boldsymbol{\delta}^r; \boldsymbol{\epsilon}_r^\star\right) = \left|\mathcal{M}_{s,\xi_\delta}\left(\boldsymbol{\delta}^r; \mathcal{S}, \omega^{r,\star}\right) - \mathcal{M}_{s,\xi_\delta}\left(\boldsymbol{\delta}^r; \mathcal{S}, \omega\right)\right|.$$

PCAC uses the shifted perturbation-neighborhood objective to account for this attacker-accessible parameter-side sensitivity. This construction favors perturbations with low neighborhood loss under the shifted surrogate state $\omega^{r,\star}$, which discourages reliance on state-specific low-loss regions. Since the non-negative-loss relaxation gives:

$$P^{\omega}_{\mathcal{M}}\left(\boldsymbol{\delta}^r;\boldsymbol{\epsilon}^{\star}_r\right) \le \mathcal{M}_{s,\xi_\delta}\left(\boldsymbol{\delta}^r;\mathcal{S},\omega^{r,\star}\right) + \mathcal{M}_{s,\xi_\delta}\left(\boldsymbol{\delta}^r;\mathcal{S},\omega\right),$$

the shifted objective places optimization pressure on the shifted component of an accessible upper proxy of $P^{\omega}_{\mathcal{M}}$. The dual-neighborhood flatness-enhanced crafting objective is formulated as $\Psi_r = \mathcal{M}_{s,\xi_\delta}\left(\boldsymbol{\delta}^r;\mathcal{S},\omega^{r,\star}\right)$. In Algorithm 1, Lines 6–8 construct the perturbation-neighborhood objective, Lines 9–11 the local surrogate-state shift, and Lines 13–14 optimize $\boldsymbol{\delta}$ using Ψ_r. Full analysis and proof are provided in **Appendix** E.

4 Evaluation

4.1 Setting

Datasets. We evaluate PCAC on three established distribution shift benchmarks, following (Wu et al., 2023; Cong et al., 2024; Su et al., 2025). *CIFAR10-C*, *CIFAR100-C*, and *ImageNet-C* are synthesized from the corresponding original clean sets by applying various corruption types with 5 severity levels (Hendrycks & Dietterich, 2019). Each benchmark includes 15 corruption types, each corresponding to a distinct distribution-shift domain (Zhang et al., 2022).

Default experiments follow settings in (Zhang et al., 2022; Rifat et al., 2025). The corruption severity is set to 3. Per benchmark, one corruption domain is randomly selected once as the auxiliary domain and kept fixed across all default runs, from which 10 labeled samples are randomly selected once to form a fixed auxiliary set $\mathcal{D}_{\text{aux}}$. The remaining 14 corruption domains constitute $\mathcal{D}_{\text{test}}$ (see **Appendix** D for sensitivity analyses on auxiliary-data sources and noise). Each individual evaluation trial corresponds to a single test batch $\mathcal{B} = \mathcal{B}_a^{\boldsymbol{\delta}} \cup \mathcal{B}_b$ with batch size $|\mathcal{B}| = 200$, where the poisoned-sample ratio $|\mathcal{B}_a^{\boldsymbol{\delta}}|/(|\mathcal{B}_a^{\boldsymbol{\delta}}| + |\mathcal{B}_b|)$ is set to 15%. For each random seed, the number of evaluation trials is $\sum_{c=1}^{14} \left\lfloor |\mathcal{D}_{\text{test}}^{(c)}|/|\mathcal{B}| \right\rfloor$, where $\mathcal{D}_{\text{test}}^{(c)}$ denotes the c-th test corruption domain. Following (Wu et al., 2023; Cong et al., 2024), we evaluate PCAC on all resulting batches as evaluation trials, with poisoned inputs crafted for each trial's $\mathcal{B}_a$.

Models. We employ six widely used TTA architectures as target backbones, including WideResNet-28-10, WideResNet-50-2 (Zagoruyko & Komodakis, 2016), ResNet-50, ResNet-152 (He et al., 2016), DenseNet-121 (Huang et al., 2017), and ResNeXt-50-32×4d (Xie et al., 2017) and evaluate common TTA methods, including TeBN (Nado et al., 2020), TENT (Wang et al., 2021), RPL (Rusak et al., 2022), Hard PL and Soft PL (Goyal et al., 2022), EATA (Niu et al., 2022), TTAC (Su et al., 2022), and SAR (Niu et al., 2023). Following (Rifat et al., 2025), target models are trained on clean training sets $\mathcal{D}_{\text{src}}$, and surrogate models use publicly available pretrained checkpoints from PyTorch CIFAR Models and TorchVision, including VGG-11-BN, VGG-19-BN (Simonyan & Zisserman, 2015), MobileNetV2 (Sandler et al., 2018), and ShuffleNetV2 (Ma et al., 2018). *By default*, VGG-19-BN is used as surrogate f_ω, DenseNet-121 as target f_θ, and TENT as target-side TTA method.

Metrics. Attack Success Rate (ASR) measures the fraction of benign co-batched samples predicted as ℓ_{tar}, calculated as: $\text{ASR} = \mathbb{E}_{x'_i \in \mathcal{B}_b}\left[\mathbb{I}\left(\arg\max_c [f_{\theta'}(x'_i)]_c = \ell_{\text{tar}}\right) \mid y_i \neq \ell_{\text{tar}}\right]$, where $f_{\theta'}$ denotes the target model after TTA on $\mathcal{B}_a^{\boldsymbol{\delta}} \cup \mathcal{B}_b$. Following (Wu et al., 2023; Su et al., 2025), the specified label ℓ_{tar} is independently sampled from the C classes for each evaluation trial, and samples (x'_i, y_i) in $\mathcal{B}_b$ already with the ground-truth label ℓ_{tar} are excluded from ASR computation.

Baselines. We establish a No-Poison control, where attacker-controlled and benign samples are co-batched with $\boldsymbol{\delta} = \mathbf{0}$, and a White Box crafting with access to the target model f_θ, the real benign co-batch set $\mathcal{B}_b$ and the target-side TTA procedure. We compare PCAC with TargetCE, which directly optimizes attacker-controlled samples $\mathcal{B}_a$ toward target class ℓ_{tar} without proxy co-batches. We compare with prior poisoning attacks, including DIA (Wu et al., 2023), TePA (Cong et al., 2024), FCA (Rifat et al., 2025), BLE and NHE (Su et al., 2025). Since these attacks are not directly applicable under our relaxed access assumptions, we follow (Su et al., 2025) to adapt them by replacing the target-model queries or access with the surrogate model f_ω, real benign data and training-similar auxiliary data with the distribution-shifted data $\mathcal{D}_{\text{aux}}$, while crafting poisoned inputs to misclassify benign co-batched samples toward ℓ_{tar}. Baseline details are provided in **Appendix** G.

Table 1: Performance compared with baselines, with *VGG-19-BN* as the surrogate and *DenseNet-121* as the target. TFR measures attack-induced flips from non-target predictions to ℓ_{tar}, while IC-TFR measures such flips only on samples correctly classified under paired No-Poison control. Both are averaged over TTA methods.

Dataset	Attack Method	TTA Method: ASR (%) TeBN	TENT	Hard PL	Soft PL	RPL	EATA	TTAC	SAR	Avg. TFR (↑)	Avg. IC-TFR (↑)
CIFAR10-C	No-Poison	$5.84_{\pm 0.27}$	$5.37_{\pm 0.19}$	$6.12_{\pm 0.31}$	$5.69_{\pm 0.24}$	$5.22_{\pm 0.16}$	$5.91_{\pm 0.29}$	$6.28_{\pm 0.34}$	$5.48_{\pm 0.21}$	– –	– –
	White Box	$88.27_{\pm 0.58}$	$84.91_{\pm 0.67}$	$85.13_{\pm 0.62}$	$79.48_{\pm 0.81}$	$82.36_{\pm 0.69}$	$83.75_{\pm 0.74}$	$75.94_{\pm 0.92}$	$78.63_{\pm 0.83}$	81.35	78.83
	TargetCE	$31.44_{\pm 0.86}$	$39.63_{\pm 1.21}$	$34.79_{\pm 0.74}$	$27.14_{\pm 1.13}$	$32.34_{\pm 0.97}$	$30.27_{\pm 1.28}$	$23.53_{\pm 1.52}$	$25.88_{\pm 1.41}$	26.73	23.19
	DIA	$37.61_{\pm 1.18}$	$44.17_{\pm 0.92}$	$38.56_{\pm 1.36}$	$24.83_{\pm 1.05}$	$28.67_{\pm 1.24}$	$32.14_{\pm 1.47}$	$21.39_{\pm 1.69}$	$23.24_{\pm 1.55}$	27.52	24.06
	TePA	$47.35_{\pm 1.04}$	$50.83_{\pm 1.33}$	$48.92_{\pm 0.89}$	$36.07_{\pm 1.42}$	$34.22_{\pm 1.18}$	$38.35_{\pm 1.51}$	$28.56_{\pm 1.63}$	$30.83_{\pm 1.74}$	36.04	32.58
	FCA	$45.59_{\pm 1.39}$	$54.22_{\pm 1.08}$	$52.28_{\pm 1.57}$	$36.43_{\pm 0.96}$	$44.91_{\pm 1.61}$	$36.42_{\pm 1.26}$	$34.81_{\pm 1.48}$	$38.54_{\pm 1.66}$	39.77	35.94
	BLE	$49.72_{\pm 0.94}$	$67.25_{\pm 1.46}$	$55.21_{\pm 1.19}$	$41.64_{\pm 1.31}$	$48.52_{\pm 1.02}$	$46.01_{\pm 1.53}$	$36.78_{\pm 1.56}$	$40.89_{\pm 1.43}$	45.51	41.36
	NHE	$58.06_{\pm 1.27}$	$59.38_{\pm 1.24}$	$61.49_{\pm 1.56}$	$42.16_{\pm 0.98}$	$52.18_{\pm 1.37}$	$42.57_{\pm 1.20}$	$40.52_{\pm 1.72}$	$43.41_{\pm 1.59}$	47.36	43.15
	Ours (PCAC)	$68.36_{\pm 0.96}$	$70.46_{\pm 0.76}$	$69.04_{\pm 1.14}$	$60.31_{\pm 0.93}$	$57.47_{\pm 1.29}$	$63.08_{\pm 1.07}$	$52.74_{\pm 1.36}$	$56.48_{\pm 1.11}$	60.28	56.81
CIFAR100-C	No-Poison	$0.96_{\pm 0.07}$	$0.89_{\pm 0.05}$	$0.93_{\pm 0.08}$	$0.98_{\pm 0.06}$	$0.91_{\pm 0.04}$	$0.84_{\pm 0.09}$	$1.00_{\pm 0.11}$	$0.87_{\pm 0.05}$	– –	– –
	White Box	$96.75_{\pm 0.74}$	$94.38_{\pm 0.69}$	$95.64_{\pm 0.63}$	$91.76_{\pm 0.84}$	$94.55_{\pm 0.58}$	$95.21_{\pm 0.76}$	$88.42_{\pm 1.04}$	$91.37_{\pm 0.87}$	93.47	90.26
	TargetCE	$44.91_{\pm 1.12}$	$47.03_{\pm 1.35}$	$41.67_{\pm 0.98}$	$38.34_{\pm 1.25}$	$45.38_{\pm 1.09}$	$36.27_{\pm 1.54}$	$35.13_{\pm 1.71}$	$40.82_{\pm 1.49}$	40.71	36.84
	DIA	$58.89_{\pm 1.46}$	$64.47_{\pm 1.02}$	$57.92_{\pm 1.29}$	$49.72_{\pm 0.91}$	$61.64_{\pm 1.63}$	$54.19_{\pm 1.17}$	$45.75_{\pm 1.82}$	$53.67_{\pm 1.35}$	55.43	50.67
	TePA	$63.26_{\pm 1.05}$	$66.52_{\pm 1.48}$	$62.15_{\pm 1.15}$	$56.35_{\pm 1.39}$	$66.41_{\pm 0.86}$	$55.76_{\pm 1.66}$	$46.87_{\pm 1.58}$	$53.29_{\pm 1.76}$	58.50	53.91
	FCA	$65.33_{\pm 1.34}$	$61.01_{\pm 1.17}$	$64.73_{\pm 1.63}$	$55.61_{\pm 1.18}$	$60.12_{\pm 1.27}$	$51.23_{\pm 0.85}$	$53.14_{\pm 1.75}$	$57.75_{\pm 1.42}$	58.29	52.76
	BLE	$74.18_{\pm 0.96}$	$70.58_{\pm 1.32}$	$73.22_{\pm 1.09}$	$70.83_{\pm 1.57}$	$75.35_{\pm 0.91}$	$72.38_{\pm 1.28}$	$63.63_{\pm 1.46}$	$59.31_{\pm 1.83}$	69.71	65.18
	NHE	$73.52_{\pm 1.25}$	$73.94_{\pm 0.89}$	$75.35_{\pm 1.45}$	$68.97_{\pm 1.14}$	$72.90_{\pm 1.36}$	$66.25_{\pm 1.45}$	$55.72_{\pm 1.79}$	$65.46_{\pm 1.62}$	68.79	63.57
	Ours (PCAC)	$84.68_{\pm 0.88}$	$83.16_{\pm 1.03}$	$84.24_{\pm 0.92}$	$80.63_{\pm 1.31}$	$81.54_{\pm 0.95}$	$78.36_{\pm 1.19}$	$72.48_{\pm 1.52}$	$71.34_{\pm 1.26}$	79.42	75.63
ImageNet-C	No-Poison	$0.09_{\pm 0.02}$	$0.07_{\pm 0.01}$	$0.08_{\pm 0.03}$	$0.10_{\pm 0.02}$	$0.09_{\pm 0.01}$	$0.06_{\pm 0.02}$	$0.10_{\pm 0.03}$	$0.08_{\pm 0.01}$	– –	– –
	White Box	$92.15_{\pm 0.61}$	$91.47_{\pm 0.55}$	$90.38_{\pm 0.58}$	$86.94_{\pm 0.74}$	$91.58_{\pm 0.79}$	$90.73_{\pm 0.66}$	$84.69_{\pm 0.80}$	$87.52_{\pm 0.71}$	89.42	86.11
	TargetCE	$37.56_{\pm 1.38}$	$39.84_{\pm 1.09}$	$34.16_{\pm 1.23}$	$31.75_{\pm 0.97}$	$28.47_{\pm 1.57}$	$42.15_{\pm 1.21}$	$36.24_{\pm 1.63}$	$41.36_{\pm 1.75}$	36.39	32.27
	DIA	$58.82_{\pm 1.17}$	$60.46_{\pm 1.45}$	$55.32_{\pm 1.06}$	$45.69_{\pm 1.64}$	$51.22_{\pm 0.92}$	$63.20_{\pm 1.34}$	$46.38_{\pm 1.78}$	$54.57_{\pm 1.59}$	54.42	49.58
	TePA	$62.34_{\pm 1.52}$	$61.13_{\pm 1.14}$	$58.93_{\pm 1.24}$	$62.81_{\pm 0.93}$	$50.85_{\pm 1.61}$	$61.36_{\pm 1.25}$	$47.62_{\pm 1.85}$	$56.42_{\pm 1.48}$	57.65	52.06
	FCA	$58.06_{\pm 0.99}$	$54.70_{\pm 1.49}$	$59.17_{\pm 1.21}$	$51.53_{\pm 1.38}$	$43.91_{\pm 1.07}$	$62.54_{\pm 1.63}$	$54.08_{\pm 1.71}$	$58.35_{\pm 1.54}$	55.26	50.39
	BLE	$72.55_{\pm 1.26}$	$73.64_{\pm 1.02}$	$68.25_{\pm 1.59}$	$63.82_{\pm 0.94}$	$68.77_{\pm 1.42}$	$69.83_{\pm 1.24}$	$51.61_{\pm 1.23}$	$60.18_{\pm 1.36}$	66.08	61.44
	NHE	$72.78_{\pm 1.13}$	$70.52_{\pm 1.36}$	$69.76_{\pm 1.08}$	$62.05_{\pm 1.55}$	$71.42_{\pm 1.25}$	$68.27_{\pm 1.57}$	$56.25_{\pm 1.69}$	$64.73_{\pm 1.82}$	66.96	62.71
	Ours (PCAC)	$81.23_{\pm 1.18}$	$78.67_{\pm 1.29}$	$79.24_{\pm 1.34}$	$76.83_{\pm 1.06}$	$75.86_{\pm 1.31}$	$79.14_{\pm 0.82}$	$73.26_{\pm 1.35}$	$74.52_{\pm 1.12}$	77.33	73.24

Implementation. For proxy construction, the percentile step is set to $\Delta_q = 20$, and the saliency-emphasized proxy image is generated with mixing ratio $\mu = 0.6$. The perturbations are constrained by a budget of $\Lambda = 8/255$, and optimized for $R = 20$ PGD iterations with step size of $\alpha = \Lambda/10$, following (Kim et al., 2023). The perturbation-neighborhood radius is set to $\xi_\delta = 0.15\Lambda$, and the number of local perturbation probes is set to $K = 4$ (Chen et al., 2025). For parameter-neighborhood shifting, the radius is set to $\rho_\omega = 0.05$, applied to the differentiable surrogate weights used for crafting following (Foret et al., 2021), and the normalization constant is set to $\tau = 10^{-12}$. The loss weight is set to $\lambda_{\text{ab}} = 1.0$. All attacks are evaluated under the same episodic evaluation setting.

We repeat the evaluation procedure with five random seeds. For each seed, following the trial coverage of (Wu et al., 2023), samples within each of the 14 target corruption domains are randomly permuted and partitioned without replacement into full test batches of size $|\mathcal{B}|$. Within each test batch, samples are randomly selected without replacement according to the poisoned-sample ratio to form the attacker-controlled $\mathcal{B}_a$, while the remaining samples form the benign set $\mathcal{B}_b$. Following the episodic evaluation (Zhang et al., 2022), the target model is reset to its pre-adaptation state before each evaluation trial, where TTA is performed on $\mathcal{B} = \mathcal{B}_a^{\boldsymbol{\delta}} \cup \mathcal{B}_b$, and ASR is computed on the benign co-batch set $\mathcal{B}_b$ under the resulting adapted model. Results are averaged over all evaluation trials for each random seed, and we report the mean and standard deviation across the five seeds.

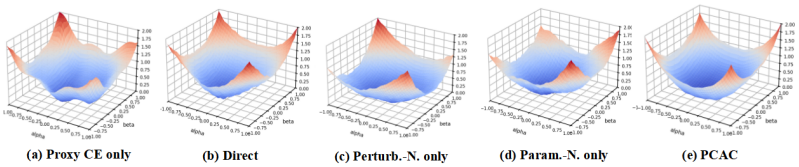
4.2 Overall Performance

Table 1 reports the ASR of different attacks across multiple TTA methods. The White-box setting achieves the highest ASR as expected, while PCAC approaches this reference with an overall average gap of 15.37%. PCAC improves over TargetCE by 36.96% in average ASR, highlighting the importance of accounting for the effect of co-batching on adaptation beyond directly imposing target supervision. Moreover, PCAC outperforms prior TTA poisoning attacks by at least 9.62% higher ASR. PCAC achieves this gain not by tailoring the crafting using real benign co-batches, target-model access or queries, but by constructing influential proxy co-batches, inducing target-oriented adaptation effects, and applying dual-neighborhood crafting that targets perturbation-side local variation and surrogate-state sensitivity. See **Appendix** C for comparisons of the proxy-based crafting objective with prior transfer-enhancement optimizers.

Attack-induced Target Flips. We further report TFR and IC-TFR relative to the paired No-Poison control to evaluate PCAC under conditions that exclude pre-existing target predictions and restrict evaluation to initially correct samples. Let $\hat{y}_i = \arg\max_c [f_{\theta'}(x'_i)]_c$ and $\hat{y}_i^{\text{NP}} = \arg\max_c [f_{\theta'_{\text{NP}}}(x'_i)]_c$, where $f_{\theta'_{\text{NP}}}$ is obtained by TTA on $\mathcal{B}_a \cup \mathcal{B}_b$ with $\boldsymbol{\delta} = \mathbf{0}$. Then, $\text{TFR} = \mathbb{E}_{x'_i \in \mathcal{B}_b}\left[\mathbb{I}(\hat{y}_i = \ell_{\text{tar}}) \,\middle|\, \hat{y}_i^{\text{NP}} \neq \ell_{\text{tar}},\ y_i \neq \ell_{\text{tar}}\right]$, and $\text{IC-TFR} = \mathbb{E}_{x'_i \in \mathcal{B}_b}\left[\mathbb{I}(\hat{y}_i = \ell_{\text{tar}}) \,\middle|\, \hat{y}_i^{\text{NP}} = y_i,\ y_i \neq \ell_{\text{tar}}\right]$. As shown in Table 1, PCAC maintains effectiveness.

Table 2: Sensitivity analysis of PCAC on CIFAR100-C. * denotes the default setting.

Hyperparameter	Value	TTA Method: ASR (%)							
		TeBN	TENT	Hard PL	Soft PL	RPL	EATA	TTAC	SAR
Severity	1	$70.27_{\pm 1.18}$	$69.75_{\pm 1.39}$	$71.18_{\pm 1.09}$	$66.35_{\pm 1.57}$	$68.92_{\pm 1.35}$	$65.74_{\pm 1.48}$	$58.63_{\pm 1.76}$	$57.85_{\pm 1.63}$
	2	$77.36_{\pm 1.25}$	$76.84_{\pm 1.41}$	$78.46_{\pm 0.83}$	$73.28_{\pm 1.43}$	$75.91_{\pm 1.12}$	$72.32_{\pm 1.37}$	$65.81_{\pm 1.67}$	$64.92_{\pm 1.59}$
	3*	$84.68_{\pm 0.88}$	$83.16_{\pm 1.03}$	$84.24_{\pm 0.92}$	$80.63_{\pm 1.31}$	$81.54_{\pm 0.95}$	$78.36_{\pm 1.19}$	$72.48_{\pm 1.52}$	$71.34_{\pm 1.26}$
	4	$86.12_{\pm 1.37}$	$85.92_{\pm 1.45}$	$86.76_{\pm 1.08}$	$81.52_{\pm 1.22}$	$84.35_{\pm 1.27}$	$80.71_{\pm 1.34}$	$75.82_{\pm 1.61}$	$74.27_{\pm 1.48}$
	5	$87.03_{\pm 1.05}$	$87.65_{\pm 1.67}$	$86.91_{\pm 1.25}$	$83.96_{\pm 1.53}$	$84.88_{\pm 1.34}$	$82.42_{\pm 1.56}$	$76.35_{\pm 1.66}$	$75.41_{\pm 1.57}$
Batch size	64	$87.84_{\pm 1.73}$	$85.63_{\pm 1.14}$	$86.92_{\pm 0.88}$	$82.06_{\pm 1.19}$	$84.38_{\pm 1.02}$	$81.27_{\pm 1.29}$	$74.10_{\pm 1.63}$	$75.32_{\pm 1.46}$
	128	$85.42_{\pm 1.15}$	$84.87_{\pm 1.27}$	$85.16_{\pm 1.34}$	$82.58_{\pm 1.35}$	$82.22_{\pm 1.16}$	$79.65_{\pm 1.28}$	$73.75_{\pm 1.59}$	$72.51_{\pm 1.42}$
	200*	$84.68_{\pm 0.88}$	$83.16_{\pm 1.03}$	$84.24_{\pm 0.92}$	$80.63_{\pm 1.31}$	$81.54_{\pm 0.95}$	$78.36_{\pm 1.19}$	$72.48_{\pm 1.52}$	$71.34_{\pm 1.26}$
	256	$83.20_{\pm 0.93}$	$82.65_{\pm 1.52}$	$82.85_{\pm 1.07}$	$79.21_{\pm 1.48}$	$80.97_{\pm 1.20}$	$77.08_{\pm 1.22}$	$73.36_{\pm 1.73}$	$69.74_{\pm 1.57}$
	320	$81.86_{\pm 1.08}$	$80.92_{\pm 1.37}$	$82.31_{\pm 1.12}$	$78.04_{\pm 1.66}$	$79.85_{\pm 1.14}$	$75.61_{\pm 1.43}$	$70.97_{\pm 1.79}$	$68.58_{\pm 1.58}$
Poisoned ratio	5%	$66.76_{\pm 1.30}$	$63.82_{\pm 1.18}$	$64.91_{\pm 1.52}$	$58.57_{\pm 1.42}$	$67.48_{\pm 1.36}$	$60.92_{\pm 1.46}$	$55.86_{\pm 1.27}$	$59.24_{\pm 1.55}$
	10%	$72.55_{\pm 1.06}$	$75.36_{\pm 1.24}$	$76.50_{\pm 1.06}$	$68.44_{\pm 1.35}$	$74.57_{\pm 1.12}$	$70.83_{\pm 1.31}$	$63.72_{\pm 1.64}$	$67.78_{\pm 1.43}$
	15%*	$84.68_{\pm 0.88}$	$83.16_{\pm 1.03}$	$84.24_{\pm 0.92}$	$80.63_{\pm 1.31}$	$81.54_{\pm 0.95}$	$78.36_{\pm 1.19}$	$72.48_{\pm 1.52}$	$71.34_{\pm 1.26}$
	30%	$87.05_{\pm 1.17}$	$84.74_{\pm 1.34}$	$86.58_{\pm 1.10}$	$83.91_{\pm 1.28}$	$83.36_{\pm 1.42}$	$81.04_{\pm 1.35}$	$76.85_{\pm 1.47}$	$75.12_{\pm 1.38}$
	40%	$88.72_{\pm 0.95}$	$86.28_{\pm 1.49}$	$88.16_{\pm 1.27}$	$85.43_{\pm 1.34}$	$85.05_{\pm 1.09}$	$84.72_{\pm 1.22}$	$78.04_{\pm 1.68}$	$80.16_{\pm 1.54}$



(a) Proxy CE only (b) Direct (c) Perturb.-N. only (d) Param.-N. only (e) PCAC

Figure 2: (a) Proxy CE only replaces TCEI with direct target CE on $\mathcal{S}$ without FCU; (b) Direct optimizes the default surrogate crafting loss $\mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega)$ without FCU; (c) Perturb.-N. only optimizes the perturbation-neighborhood objective $\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}; \mathcal{S}, \omega)$; (d) Param.-N. only updates perturbations using the center crafting loss under a locally shifted surrogate state, $\mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega^\star)$; and (e) PCAC performs dual-neighborhood crafting by optimizing $\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}; \mathcal{S}, \omega^\star)$.

Impact of Corruption Severity & Batch Size & Poison Ratio. As shown in Table 2, ASR increases with corruption severity, which may be associated with larger distribution shifts and reduced prediction stability (Wu et al., 2023; Wang et al., 2021). For test batch size, ASR generally decreases as the batch becomes larger; nevertheless, PCAC maintains about 77.27% average ASR at batch size 320. For poisoned-sample ratio, PCAC remains effective even at lower ratios, achieving about 62.19% and 71.22% average ASR at 5% and 10%. ASR increases as more attacker-controlled samples participate, but the improvement saturates with about a further 5.0% gain. Additional results on cross-model performance, hyperparameters, and ViT target models are provided in **Appendix** A.

4.3 Ablation Study and Flatness Analysis

Component Ablation. We ablate the components of PCAC, including proxy construction (PC), target-confident effect induction (TCEI), and the flatness-enhanced crafting update (FCU). As shown in Table 3, all components contribute to the attack. More results within surrogate adaptation approximation, TCEI, FCU, and PC are provided in **Appendix** B.

Visualization. We visualize surrogate-side crafting landscapes of variants. As shown in Figure 2, PCAC yields a more coherent low-loss basin than center-point and single-neighborhood variants.

Diagnostic analysis and Crafting Cost. We compute two attacker-accessible quantities corresponding to the surrogate-side terms discussed in Section 3.2. Specifically, $R^{(0)}_{\mathcal{J},\xi_\delta,s}$ measures perturbation-side variation, while $P^{\omega}_{\mathcal{M}}$ measures the change in $\mathcal{M}_{s,\xi_\delta}$ under the locally shifted surrogate state $\omega^\star = \omega + \boldsymbol{\epsilon}^\star$. We additionally report the target-side crafting gap $\mathcal{G}(\boldsymbol{\delta}) = |\mathcal{J}_t(\boldsymbol{\delta}; \mathcal{B}_b, \theta) - \mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega)|$. For a compute-matched comparison, Direct (CM) uses $4\times$ more crafting optimization steps than Direct, yielding a crafting cost comparable to PCAC. All diagnostics are normalized by the corresponding values of Direct. As shown in Table 4, PCAC achieves the lowest diagnostic values and the highest ASR. Notably, compute-matched Direct yields only a modest ASR gain, suggesting that PCAC's improvement is not simply due to additional crafting computation.

4.4 Defenses

We evaluate five potential countermeasures against PCAC. (i) *Input Preprocessing* randomly resizes and pads each input before adaptation and prediction (Xie et al., 2018), perturbing fixed low-level adversarial patterns. (ii) *Entropy Filtering* excludes high-entropy samples from adaptation while evaluating all benign samples (Niu et al., 2022), reducing noisy-gradient contributions from uncertain

Table 3: Performance of PCAC and its variants on CIFAR100-C. ✗* denotes a replacement ablation where TCEI is removed and replaced with direct target-label cross-entropy on the proxy set $\mathcal{S}$.

PC	TCEI	FCU	Variant	ASR (%)
✓	✓	✓	PCAC	$83.16_{\pm 1.03}$
✗	✓	✓	w/o PC	$73.29_{\pm 1.21}$
✓	✗*	✗	Proxy CE only	$56.83_{\pm 1.47}$
✓	✗*	✓	Proxy CE + FCU	$65.95_{\pm 1.18}$
✓	✓	✗	Direct	$62.74_{\pm 1.42}$

Table 4: Diagnostic analysis and crafting cost on CIFAR100-C. Quantities are normalized by Direct.

Method	ASR (%) ↑	$R^{(0)}_{\mathcal{J},\xi_\delta,s}$ ↓	$P^{\infty}_{\mathcal{M}}$ ↓	$\mathcal{G}$ ↓	Time (s) / batch ↓	Rel. cost ↓
Proxy CE only	$56.83_{\pm 1.47}$	–	–	–	$1.34_{\pm 0.10}$	0.84×
Direct	$62.74_{\pm 1.42}$	1.000	1.000	1.000	$1.60_{\pm 0.13}$	1.00×
Direct (CM)	$66.17_{\pm 1.35}$	0.927	0.958	0.906	$6.43_{\pm 0.48}$	4.02×
Perturb.-N. only	$72.84_{\pm 1.26}$	0.468	0.725	0.764	$4.92_{\pm 0.35}$	3.08×
Param.-N. only	$70.38_{\pm 1.09}$	0.674	0.453	0.817	$3.14_{\pm 0.27}$	1.96×
PCAC	$\mathbf{83.16_{\pm 1.03}}$	**0.368**	**0.341**	**0.536**	$6.58_{\pm 0.61}$	4.11×

test samples. (iii) *Influence-Clipped TTA* clips each sample's contribution to the adaptation update (Li et al., 2026), limiting the dominance of individual poisoned samples. (iv) *Stochastic Restoration* randomly restores part of the adapted state to the source state (Wang et al., 2022), limiting attacker-induced parameter changes during within-batch adaptation step. (v) *Robust BN Statistics* replaces batch means with channel-wise medians for test-time normalization (Park et al., 2024), reducing poisoned-sample influence on batch-statistic estimation.

As shown in Figure 3, *Input Preprocessing* and *Entropy Filtering* provide limited mitigation, modestly reducing ASR by 4.42% and 7.23%, respectively. *Influence-Clipped TTA* and *Stochastic Restoration* reduce ASR by 15.12% on average, while lowering clean accuracy by 6.45%. This is expected since PCAC targets co-batch adaptation rather than fixed input patterns or uncertain samples. *Robust BN Statistics* reduces ASR by 21.42% by weakening poisoned influence on batch statistics, while incurring a 13.97% clean accuracy drop, which may reflect reduced accommodation of legitimate benign variation (Park et al., 2024).

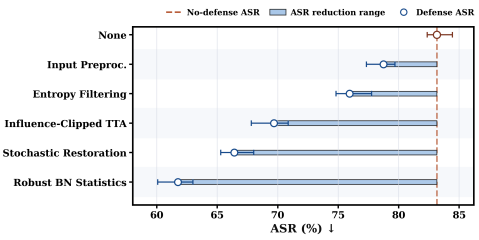


Figure 3: Defense evaluation on CIFAR100-C.

5 Related Work

Poisoning Attacks in TTA. Prior studies show that TTA introduces a batch-level poisoning threat, where malicious co-batched samples can affect benign predictions. DIA formulates TTA poisoning by manipulating benign co-batched predictions with target-model and benign-batch access (Wu et al., 2023). FCA studies a feature collapse objective under target-model and real-benign-batch access, but without benign labels (Rifat et al., 2025). TePA trains surrogate models using the target TTA method and training-like auxiliary data for crafting, without target-model interaction (Cong et al., 2024). RTTDP distills a surrogate model from target-model queries and develops TTA-aware attack objectives, including NHE and BLE, without benign-batch access (Su et al., 2025).

However, target-side details and real benign co-batches may be inaccessible, while repeated target queries incur cost and may raise suspicion (Xu et al., 2023). PCAC instead performs surrogate-side co-batch crafting with attacker-accessible proxy contexts, target-oriented guidance, and dual-neighborhood optimization to discourage center-specific and surrogate-state-sensitive solutions.

6 Conclusion

In this paper, we proposed PCAC, a novel poisoning framework against TTA without target-side access during crafting. PCAC constructs a saliency-emphasized proxy input set from distribution-shifted auxiliary data and induces target-confident effect with dual-neighborhood crafting objective. Proxy diversity and flatness-enhanced target-effect optimization enable the crafting of poisoned inputs that manipulate benign co-batched predictions under adapted target models.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and improve the readability of the manuscript. All AI-assisted edits were reviewed by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML 2018*, volume 80, pp. 284–293, 2018.

Junyoung Byun, Seungju Cho, Myung-Joon Kwon, Hee-Seon Kim, and Changick Kim. Improving the transferability of targeted adversarial examples through object-based diverse input. In *CVPR 2022*, pp. 15223–15232, 2022.

Jiahao Chen, Zhou Feng, Rui Zeng, Yuwen Pu, Chunyi Zhou, Yi Jiang, Yuyou Gan, Jinbao Li, and Shouling Ji. Enhancing adversarial transferability with adversarial weight tuning. In *AAAI 2025*, pp. 2061–2069, 2025.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR 2021*, pp. 15750–15758, 2021.

Tianshuo Cong, Xinlei He, Yun Shen, and Yang Zhang. Test-time poisoning attacks against test-time adaptation models. In *IEEE S&P 2024*, pp. 1306–1324, 2024.

Luke Nicholas Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. CINIC-10 is not imagenet or CIFAR-10. *CoRR*, abs/1810.03505, 2018.

Zhun Deng, Linjun Zhang, Amirata Ghorbani, and James Zou. Improving adversarial robustness via unlabeled out-of-domain data. In *AISTATS 2021*, volume 130, pp. 2845–2853, 2021.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR 2018*, pp. 9185–9193, 2018.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR 2019*, pp. 4312–4321, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR 2021*, 2021.

Zhijin Ge, Xiaosen Wang, Hongying Liu, Fanhua Shang, and Yuanyuan Liu. Boosting adversarial transferability by achieving flat local maxima. In *NeurIPS 2023*, 2023.

Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Keepaugment: A simple information-preserving data augmentation approach. In *CVPR 2021*, pp. 1055–1064, 2021.

Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and J. Zico Kolter. Test time adaptation via conjugate pseudo-labels. In *NeurIPS 2022*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR 2016*, pp. 770–778, 2016.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR 2019*, 2019.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR 2017*, pp. 2261–2269, 2017.

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *ICML 2020*, volume 119, pp. 4804–4815, 2020.

Minseon Kim, Hyeonjeong Ha, Sooel Son, and Sung Ju Hwang. Effective targeted attacks for adversarial self-supervised learning. In *NeurIPS 2023*, 2023.

Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. Robustifying vision transformer without retraining from scratch by test-time class-conditional feature alignment. In *IJCAI 2022*, pp. 1009–1016, 2022.

Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *CVPR 2020*, pp. 638–646, 2020.

Shixin Li, Chaoxiang He, Xiaojing Ma, Bin Benjamin Zhu, Shuo Wang, Hongsheng Hu, Dongmei Zhang, and Linchen Yu. Enhancing adversarial transferability with checkpoints of a single model's training. In *CVPR 2025*, pp. 20685–20694, 2025.

Zefeng Li, Qiaoyue Tang, Mathias Lécuyer, and Evan Shelhamer. Private and stable test-time adaptation with differential privacy, 2026.

Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *Int. J. Comput. Vis.*, 133(1):31–64, 2025.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN architecture design. In *ECCV 2018*, volume 11218, pp. 122–138, 2018.

Zachary Nado, Shreyas Padhy, D. Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift, 2020.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML 2022*, volume 162, pp. 16888–16905, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *ICLR 2023*, 2023.

Hyejin Park, Jeongyeon Hwang, Sunung Mun, Sangdon Park, and Jungseul Ok. Medbn: Robust test-time adaptation against malicious test samples. In *CVPR 2024*, pp. 5997–6007, 2024.

Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. In *NeurIPS 2022*, 2022.

Chunlin Qiu, Ang Li, Yiheng Duan, Shenyi Zhang, Yuanjie Zhang, Lingchen Zhao, and Qian Wang. Boosting adversarial transferability with low-cost optimization via maximin expected flatness. *IEEE Trans. Inf. Forensics Secur.*, 21:2236–2251, 2026.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML 2019*, pp. 5389–5400, 2019.

Shahriar Rifat, Jonathan D. Ashdown, Michael J. De Lucia, Ananthram Swami, and Francesco Restuccia. On the adversarial vulnerability of label-free test-time adaptation. In *ICLR 2025*, 2025.

Evgenia Rusak, Steffen Schneider, George Pachitariu, Luisa Eck, Peter Vincent Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. If your data distribution shifts, use self-learning. *Trans. Mach. Learn. Res.*, 2022, 2022.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR 2018*, pp. 4510–4520, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR 2015*, 2015.

Yongyi Su, Xun Xu, and Kui Jia. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering. In *NeurIPS 2022*, 2022.

Yongyi Su, Yushu Li, Nanqing Liu, Kui Jia, Xulei Yang, Chuan-Sheng Foo, and Xun Xu. On the adversarial risk of test time adaptation: An investigation into realistic test-time data poisoning. In *ICLR 2025*, 2025.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR 2021*, 2021.

Haohan Wang, Songwei Ge, Zachary C. Lipton, and Eric P. Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS 2019*, pp. 10506–10518, 2019.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR 2022*, pp. 7191–7201, 2022.

Zixin Wang, Yadan Luo, Liang Zheng, Zhuoxiao Chen, Sen Wang, and Zi Huang. In search of lost online test-time adaptation: A survey. *Int. J. Comput. Vis.*, 133(3):1106–1139, 2025.

Tong Wu, Feiran Jia, Xiangyu Qi, Jiachen T. Wang, Vikash Sehwag, Saeed Mahloujifar, and Prateek Mittal. Uncovering adversarial risks of test-time adaptation. In *ICML 2023*, pp. 37456–37495, 2023.

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. Mitigating adversarial effects through randomization. In *ICLR 2018*, 2018.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR 2019*, pp. 2730–2739, 2019.

Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR 2017*, pp. 5987–5995, 2017.

Zhuoer Xu, Zhangxuan Gu, Jianping Zhang, Shiwen Cui, Changhua Meng, and Weiqiang Wang. Backpropagation path search on adversarial transferability. In *ICCV 2023*, pp. 4640–4650, 2023.

Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *CVPR 2020*, pp. 13873–13882, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC 2016*, 2016.

Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: test time robustness via adaptation and augmentation. In *NeurIPS 2022*, 2022.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation, 2023.

Zhengyu Zhao, Zhuoran Liu, and Martha A. Larson. On success and simplicity: A second look at transferable targeted attacks. In *NeurIPS 2021*, pp. 6115–6128, 2021.

Junhua Zou, Zhisong Pan, Junyang Qiu, Xin Liu, Ting Rui, and Wei Li. Improving the transferability of adversarial examples with resized-diverse-inputs, diversity-ensemble and region fitting. In *ECCV 2020*, volume 12367, pp. 563–579, 2020.

APPENDIX CONTENTS

A Additional Parameter Analyses **14**

B Additional Design Ablations **14**

B.1 Surrogate Adaptation Approximation 15

B.2 Target-Confident Effect Induction Design 15

B.3 Flatness-Enhanced Update Design 17

B.4 Proxy Construction Design 18

C Additional Crafting-Strategy Comparison **19**

D Sensitivity to Auxiliary Data **20**

D.1 Auxiliary Source Distribution 20

D.2 Auxiliary-Data Quality 20

E Theoretical Analysis **21**

F Implementation Details **23**

G Adaptation of Baselines **23**

Table 5: PCAC performance with different surrogate and target models.

Dataset	Surrogate	Target Architecture under TENT: ASR (%)						Avg. ASR
		WideResNet-28-10	ResNet-50	ResNet-152	ResNeXt-50	DenseNet-121	WideResNet-50-2	
CIFAR10-C	VGG-11-BN	$70.88_{\pm 1.07}$	$73.46_{\pm 0.93}$	$68.95_{\pm 1.18}$	$74.14_{\pm 0.88}$	$67.38_{\pm 1.26}$	$73.75_{\pm 1.15}$	71.43
	VGG-19-BN	$74.26_{\pm 0.94}$	$71.54_{\pm 1.12}$	$75.05_{\pm 0.92}$	$69.88_{\pm 1.07}$	$70.46_{\pm 0.76}$	$73.13_{\pm 1.19}$	72.39
	MobileNetV2	$68.34_{\pm 1.28}$	$66.37_{\pm 1.16}$	$70.96_{\pm 1.34}$	$64.43_{\pm 1.09}$	$67.81_{\pm 1.45}$	$72.26_{\pm 1.22}$	68.36
	ShuffleNetV2	$71.05_{\pm 1.03}$	$74.12_{\pm 0.89}$	$70.48_{\pm 1.27}$	$75.95_{\pm 1.12}$	$69.42_{\pm 1.34}$	$72.88_{\pm 1.31}$	72.32
CIFAR100-C	VGG-11-BN	$82.62_{\pm 1.12}$	$85.18_{\pm 0.98}$	$81.93_{\pm 1.24}$	$86.06_{\pm 0.95}$	$78.92_{\pm 1.47}$	$84.15_{\pm 1.06}$	83.14
	VGG-19-BN	$86.38_{\pm 0.87}$	$83.91_{\pm 1.09}$	$87.27_{\pm 0.94}$	$81.35_{\pm 1.18}$	$83.16_{\pm 1.03}$	$86.72_{\pm 0.95}$	84.80
	MobileNetV2	$79.24_{\pm 1.34}$	$82.32_{\pm 1.27}$	$84.06_{\pm 1.38}$	$77.31_{\pm 1.43}$	$80.25_{\pm 1.21}$	$82.53_{\pm 1.30}$	80.95
	ShuffleNetV2	$85.72_{\pm 1.05}$	$86.64_{\pm 0.92}$	$84.43_{\pm 1.16}$	$87.46_{\pm 0.84}$	$79.94_{\pm 1.28}$	$86.28_{\pm 1.12}$	85.08
ImageNet-C	VGG-11-BN	$77.84_{\pm 1.38}$	$82.12_{\pm 1.22}$	$79.36_{\pm 1.48}$	$75.91_{\pm 1.31}$	$74.53_{\pm 1.56}$	$82.02_{\pm 1.24}$	78.63
	VGG-19-BN	$82.96_{\pm 1.18}$	$80.74_{\pm 1.15}$	$81.52_{\pm 1.29}$	$83.13_{\pm 1.14}$	$78.67_{\pm 1.29}$	$76.44_{\pm 1.38}$	80.58
	MobileNetV2	$74.35_{\pm 1.61}$	$73.28_{\pm 1.42}$	$76.84_{\pm 1.73}$	$79.45_{\pm 1.49}$	$77.02_{\pm 1.58}$	$75.97_{\pm 1.46}$	76.15
	ShuffleNetV2	$79.41_{\pm 1.23}$	$83.23_{\pm 1.25}$	$78.95_{\pm 1.36}$	$82.66_{\pm 1.28}$	$76.81_{\pm 1.43}$	$82.12_{\pm 1.32}$	80.53

A Additional Parameter Analyses

Impact of Surrogate and Target Architectures. As shown in Table 5, PCAC remains effective across architectures.

Impact of Hyperparameters. As shown in Figure 4, PCAC achieves 73.47% ASR with only 5 auxiliary samples, while increasing the volume beyond 10 yields at most a 3.17% further gain. Moderate mixing around $\mu = 0.6$ balances the auxiliary context and saliency-emphasized content, improving ASR by up to 2.05% over $\mu = 0.3$. Crafting-related hyperparameters perform favorably around $\xi_\delta = 0.15\Lambda$, $K = 4$, $\rho_\omega = 0.05$, and $\lambda_{\text{ab}} = 1.0$. For perturbation budget, increasing Λ improves ASR, with PCAC still achieving 68.92% ASR under $4/255$. Results indicate default choices are reasonable.

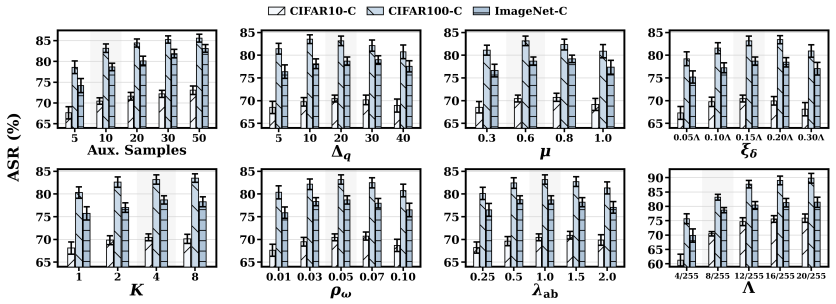


Figure 4: Impact of hyperparameters.

LayerNorm-based ViT Targets. We evaluate a LayerNorm-based ViT-B/16 target on ImageNet-C. Following prior ViT test-time adaptation studies (Kojima et al., 2022), TENT updates only the affine parameters of LayerNorm using entropy minimization, with SGD learning rate 10^{-3}, momentum 0.9, and global gradient clipping at 1.0. We retain the default attack-side and evaluation settings, including corruption severity 3, batch size $|\mathcal{B}| = 200$, poisoned-sample ratio 15%, the VGG-19-BN surrogate, perturbation budget, and episodic trial reset (Zhang et al., 2022). For the ViT target, TENT adapts the LayerNorm affine parameters, providing a complementary target setting with a different architecture and normalization mechanism. As shown in Table 6, PCAC maintains its effectiveness under the LayerNorm-based ViT target, achieving 56.72% ASR, 56.65% TFR, and 49.38% IC-TFR. These results indicate that PCAC remains effective when the target does not use batch-level normalization statistics shared across co-batched samples.

B Additional Design Ablations

In this section, we further examine the surrogate-side adaptation approximation and provide within-component ablations for proxy construction (PC), target-confident effect induction (TCEI), and the flatness-enhanced crafting update (FCU).

Table 6: Performance against a LayerNorm-based ViT-B/16 target under TENT on ImageNet-C. The surrogate remains VGG-19-BN and all attack-side settings follow the default evaluation. TENT updates only the LayerNorm affine parameters of the target model.

Method	ASR (%) ↑	TFR (%) ↑	IC-TFR (%) ↑
No-Poison	$0.09_{\pm 0.02}$	–	–
White Box	$69.12_{\pm 1.06}$	$69.05_{\pm 1.08}$	$62.44_{\pm 1.19}$
TargetCE	$23.78_{\pm 0.91}$	$23.70_{\pm 0.93}$	$18.62_{\pm 1.02}$
DIA	$29.86_{\pm 1.04}$	$29.78_{\pm 1.06}$	$24.08_{\pm 1.13}$
TePA	$33.21_{\pm 1.12}$	$33.13_{\pm 1.14}$	$27.42_{\pm 1.20}$
FCA	$36.47_{\pm 1.01}$	$36.39_{\pm 1.03}$	$30.61_{\pm 1.11}$
BLE	$39.74_{\pm 1.15}$	$39.66_{\pm 1.17}$	$33.52_{\pm 1.23}$
NHE	$42.83_{\pm 1.09}$	$42.75_{\pm 1.11}$	$36.31_{\pm 1.20}$
PCAC	$\mathbf{56.72_{\pm 1.18}}$	$\mathbf{56.65_{\pm 1.20}}$	$\mathbf{49.38_{\pm 1.29}}$

B.1 Surrogate Adaptation Approximation

Because the deployed target-side TTA method and its adaptation loss are unavailable to the attacker, PCAC follows the single-level surrogate-side approximation of (Wu et al., 2023). Specifically, batch-conditioned normalization statistics in f_ω are recomputed from the mixed proxy batch $\mathcal{B}_a^{\boldsymbol{\delta}} \cup \mathcal{S}$ during crafting. We then compare this default approximation with two variants: without surrogate-side adaptation and a 1-step surrogate unroll. The latter additionally applies one differentiable entropy-minimization update on the same mixed proxy batch. Following the settings of (Wang et al., 2021; Wu et al., 2023) and their implementations[1] [2], this update uses Adam with a learning rate of 10^{-3} and modifies only the channel-wise affine parameters of the surrogate normalization layers, while keeping all other parameters fixed. Poisoned samples crafted under each variant are then evaluated against target models with different TTA methods.

As shown in Figure 5, removing surrogate-side adaptation reduces ASR across the examined target models, indicating the importance of accounting for the mixed surrogate co-batch during crafting. The diagnostic 1-step surrogate unroll yields up to 2.37% ASR improvement over the default single-level approximation, while increasing the surrogate-side crafting cost to $2.14\times$ that of the default.

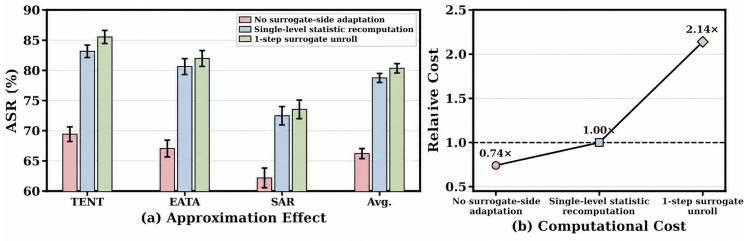


Figure 5: Effect of the surrogate-side adaptation approximation on CIFAR100-C under target models with different TTA methods.

B.2 Target-Confident Effect Induction Design

We use the following abbreviations in this subsection. *TC* denotes the attacker-side target-confidence loss $\mathcal{L}_A^{\mathrm{TC}}$, and *A2S* denotes the attacker-to-proxy-set soft-guidance term $\mathcal{L}_{A\to S}^{\mathrm{tar}}$, where S refers to the proxy context $\mathcal{S}$. *TCEI* denotes their combination used by default in PCAC.

Let p_i^A and $p^S(M_q^y)$ denote the softmax predictive distributions of the poisoned attacker-controlled sample $x_i' + \delta_i$ and the proxy input $M_q^y \in \mathcal{S}$, respectively, under the surrogate state adapted on $\mathcal{B}_a^{\boldsymbol{\delta}} \cup \mathcal{S}$. For clarity, we present only the objective terms varied in this section.

The default TCEI objective is

$$\mathcal{J}_s = \mathcal{L}_A^{\mathrm{TC}} + \lambda_{\mathrm{ab}} \mathcal{L}_{A\to S}^{\mathrm{tar}},$$

[1] https://github.com/DequanWang/TENT

[2] https://github.com/inspire-group/tta_risk

where

$$\mathcal{L}_A^{\mathrm{TC}} = -\frac{1}{|\mathcal{B}_a|} \sum_{x_i' \in \mathcal{B}_a} \sum_{k=1}^{C} \mathbb{I}(k = \ell_{\mathrm{tar}}) \log[p_i^A]_k.$$

The detached attacker-side teacher is formed from the batch-averaged prediction

$$\bar{T}_A = \frac{1}{|\mathcal{B}_a|} \sum_{x_i' \in \mathcal{B}_a} p_i^A,$$

and the A2S guidance term is

$$\mathcal{L}_{A\to S}^{\mathrm{tar}} = -\frac{1}{|\mathcal{S}|} \sum_{M_q^y \in \mathcal{S}} \sum_{k=1}^{C} \mathrm{sg}[\bar{T}_{A,k}] \log[p^S(M_q^y)]_k.$$

For the compared variants, the proxy-side target-label cross-entropy is

$$\mathcal{L}_S^{\mathrm{CE}} = -\frac{1}{|\mathcal{S}|} \sum_{M_q^y \in \mathcal{S}} \sum_{k=1}^{C} \mathbb{I}(k = \ell_{\mathrm{tar}}) \log[p^S(M_q^y)]_k,$$

and the proxy-side entropy loss is

$$\mathcal{L}_S^{\mathrm{ent}} = -\frac{1}{|\mathcal{S}|} \sum_{M_q^y \in \mathcal{S}} \sum_{k=1}^{C} [p^S(M_q^y)]_k \log[p^S(M_q^y)]_k.$$

The detached proxy-side teacher used for S2S guidance is

$$\bar{T}_S = \frac{1}{|\mathcal{S}|} \sum_{M_q^y \in \mathcal{S}} p^S(M_q^y).$$

The corresponding S2S guidance term is

$$\mathcal{L}_{S\to S}^{\mathrm{tar}} = -\frac{1}{|\mathcal{S}|} \sum_{M_q^y \in \mathcal{S}} \sum_{k=1}^{C} \mathrm{sg}[\bar{T}_{S,k}] \log[p^S(M_q^y)]_k.$$

Thus, we define the compared variants as follows.

(i) *Backpropagating through $\bar{T}_A$.* This variant removes the stop-gradient operator from $\mathcal{L}_{A\to S}^{\mathrm{tar}}$, allowing gradients to propagate through the attacker-side teacher $\bar{T}_A$.

(ii) *Replace A2S with Proxy CE.* This variant replaces $\mathcal{L}_{A\to S}^{\mathrm{tar}}$ with $\mathcal{L}_S^{\mathrm{CE}}$, while retaining $\mathcal{L}_A^{\mathrm{TC}}$.

(iii) *Replace A2S with proxy entropy.* This variant replaces $\mathcal{L}_{A\to S}^{\mathrm{tar}}$ with $\mathcal{L}_S^{\mathrm{ent}}$, while retaining $\mathcal{L}_A^{\mathrm{TC}}$.

(iv) *Without A2S (TC only).* This variant removes $\mathcal{L}_{A\to S}^{\mathrm{tar}}$ and retains only $\mathcal{L}_A^{\mathrm{TC}}$.

(v) *Proxy-side target anchor.* This variant replaces $\mathcal{L}_A^{\mathrm{TC}}$ with $\mathcal{L}_S^{\mathrm{CE}}$, while retaining A2S guidance with the attacker-side teacher $\bar{T}_A$.

(vi) *Proxy CE + proxy-side teacher.* Starting from (v), this variant further replaces the attacker-side teacher $\bar{T}_A$ with the proxy-side teacher $\bar{T}_S$.

(vii) *Replace TCEI with Proxy CE.* This variant removes both TC and A2S and uses $\mathcal{L}_S^{\mathrm{CE}}$.

Table 7 reports results of ablations.

- *Detached vs. differentiable attacker-side teacher.* Backpropagating through $\bar{T}_A$ reduces ASR by 4.98%. Allowing the teacher distribution to change jointly with the proxy-side objective removes its role as a fixed guidance target, which may weaken the stability of A2S optimization (Chen & He, 2021).
- *A2S vs. alternative proxy-side guidance.* Replacing A2S with Proxy CE while retaining TC reduces ASR by 12.52%. This replacement applies fixed target-label supervision directly to the proxy inputs, but removes the attacker-induced soft distribution that connects the proxy context to the poisoned attacker set. Replacing A2S with proxy entropy reduces ASR by 13.88%, a further 1.36% decrease relative to Proxy CE. Unlike A2S, entropy minimization

Table 7: Additional crafting-objective ablations on CIFAR100-C under TENT.

Variant	ASR (%)
PCAC (TCEI: TC + A2S)	$83.16_{\pm 1.03}$
(i) Backpropagating through $\bar{T}_A$	$78.18_{\pm 1.12}$
(ii) Replace A2S with Proxy CE	$70.64_{\pm 1.50}$
(iii) Replace A2S with proxy entropy	$69.28_{\pm 1.31}$
(iv) Without A2S (TC only)	$68.71_{\pm 1.25}$
(v) Proxy-side target anchor	$67.38_{\pm 1.36}$
(vi) Proxy CE + proxy-side teacher	$65.46_{\pm 1.43}$
(vii) Replace TCEI with Proxy CE	$65.95_{\pm 1.18}$

encourages confident proxy predictions without specifying a target specified-label direction. Using TC alone reduces ASR by 14.45%, as removing A2S restricts target specified-label supervision to the attacker-controlled samples and provides no explicit guidance to the proxy context.

- *Target-anchor and teacher placement.* Replacing attacker-side TC with Proxy CE while retaining A2S reduces ASR by 15.78%. This variant moves the explicit target anchor from the perturbed attacker-controlled samples to the proxy inputs, suggesting that placing the target-confidence anchor on attacker-controlled samples is more effective for subsequent proxy guidance in this setting. Under the proxy-side target anchor, further replacing the attacker-side A2S teacher with the proxy-side S2S teacher produces an additional 1.92% reduction. Because both the target anchor and teacher are formed on the proxy side, this variant weakens the explicit attacker-to-proxy guidance structure. Finally, replacing TCEI with Proxy CE reduces ASR by 17.21%. This variant removes both attacker-side target confidence and attacker-induced soft guidance and relies only on direct target-label supervision over the proxy inputs.

B.3 Flatness-Enhanced Update Design

We further examine the design of the flatness-enhanced crafting update. *Direct* removes the flatness-enhanced update and optimizes the center surrogate loss $\mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega)$. *Perturb.-N. only* optimizes the perturbation-neighborhood objective $\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}; \mathcal{S}, \omega)$, where $\boldsymbol{\delta}_j = \Pi_{\mathcal{A}_\Lambda}(\boldsymbol{\delta} + \mathbf{u}_j)$ and $\mathbf{u}_j \sim \mathrm{Unif}(B_{\xi_\delta}(0))$. *Param.-N. only* removes perturbation-neighborhood averaging and optimizes center crafting loss under a locally shifted surrogate state, $\mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega^\star)$, where

$$\boldsymbol{\epsilon}^\star = \rho_\omega \frac{\nabla_\omega \mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega)}{\|\nabla_\omega \mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega)\|_2 + \tau}, \quad \omega^\star = \omega + \boldsymbol{\epsilon}^\star.$$

Here, $\omega + \boldsymbol{\epsilon}^\star$ denotes the shifted surrogate state obtained by applying the offset to differentiable surrogate weights, following (Foret et al., 2021). For full PCAC, the offset $\boldsymbol{\epsilon}^\star$ is instead computed from the gradient of the perturbation-neighborhood objective $\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}; \mathcal{S}, \omega)$. The resulting shifted state $\omega^\star = \omega + \boldsymbol{\epsilon}^\star$ is then used to optimize $\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}; \mathcal{S}, \omega^\star)$, thereby coupling perturbation-neighborhood averaging with local surrogate-state shifting in the dual-neighborhood update.

Moreover, we include parameter-shift design variants. *Single-point param. shift* keeps the shifted perturbation-neighborhood objective $\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}; \mathcal{S}, \omega^\star)$, but computes $\boldsymbol{\epsilon}^\star$ from the center loss $\mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega)$ instead of $\mathcal{M}_{s,\xi_\delta}$. *Random param. shift* replaces the gradient-based offset with $\boldsymbol{\epsilon}_{\mathrm{rand}} = \rho_\omega \mathbf{z}/(\|\mathbf{z}\|_2 + \tau)$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, I)$ is independently sampled over the differentiable surrogate weights at each crafting iteration. We additionally examine center inclusion in the perturbation neighborhood. *w/o center point* averages the objective over the K random probes but excludes $\mathbf{u}_0 = 0$, using a normalization factor of $1/K$.

As shown in Table 8, *Direct* removes both perturbation-neighborhood averaging and shifted-state evaluation, restricting crafting to the center surrogate loss and reducing ASR by 20.42%. *Perturb.-N. only* does not explicitly account for local surrogate-state variation, whereas *Param.-N. only* does not average over nearby perturbations, resulting in ASR reductions of 10.32% and 12.78%, respectively. *Single-point param. shift* derives the shift from the center loss rather than the perturbation-neighborhood objective subsequently used for crafting, reducing ASR by 5.57%. *Random param.*

Table 8: Additional flatness-enhanced update ablations on CIFAR100-C under TENT. FCU denotes the flatness-enhanced crafting update.

Variant	ASR (%)
PCAC	$83.16_{\pm 1.03}$
Direct $\mathcal{J}_s$ (w/o FCU)	$62.74_{\pm 1.42}$
Perturb.-N. only	$72.84_{\pm 1.26}$
Param.-N. only	$70.38_{\pm 1.09}$
Single-point param. shift	$77.59_{\pm 1.37}$
Random param. shift	$68.21_{\pm 1.22}$
w/o center point	$79.70_{\pm 1.18}$

shift instead uses a random direction independent of the crafting loss and causes a larger 14.95% reduction, suggesting a benefit from using a gradient-informed shift direction. Finally, *w/o center point* excludes the current perturbation from the neighborhood average, leaving the estimate based only on sampled probes and reducing ASR by 3.46%. This result suggests that including the center point provides a useful reference for local averaging in this setting.

B.4 Proxy Construction Design

We further examine the design of the proxy-construction module. The default PCAC proxy uses

$$L^y_{\text{joint}} = \text{Norm}(L^y_f) \odot \text{Norm}(L_{\text{ent}}),$$

keeps entries above $Q_q(L^y_{\text{joint}})$, and constructs

$$M^y_q = (1-\mu)u^{\text{aux}} + \mu C^y_q + \eta.$$

Here L^y_f is the label-indicative heatmap, L_{ent} is the entropy-confidence heatmap, $C^y_q = u^{\text{aux}} \odot G^y_q$, and η is the proxy-mixing noise.

The ablations modify one part of this construction while keeping the surrogate model, crafting objective, and flatness-enhanced update unchanged.

(i) *Raw auxiliary proxy* directly uses u^{aux} without saliency masking.
(ii) *Size-matched raw auxiliary proxy* repeats each u^{aux} for $|\mathcal{Q}|$ copies, without saliency masking or other input transformations, so that the resulting proxy batch has the same cardinality as the default proxy batch.
(iii) *Random mask proxy* replaces G^y_q with a random mask that uniformly selects the same number of spatial locations as the corresponding joint-heatmap mask.
(iv) *w/o label-indicative heatmap* constructs the mask using only L_{ent}.
(v) *w/o entropy-confidence heatmap* constructs the mask using only L^y_f.
(vi) *Additive fusion* replaces multiplicative fusion with $\text{Norm}(L^y_f) + \text{Norm}(L_{\text{ent}})$.
(vii) *Single mask ratio* uses one fixed percentile threshold with $q = 50$ instead of the percentile set $\mathcal{Q}$.
(viii) *w/o noise in proxy mixing* sets $\eta = 0$.

Table 9 reports results of ablations.

- *Proxy content selection. Raw auxiliary proxy* directly uses unprocessed auxiliary inputs, providing a less focused proxy co-batch context and reducing ASR by 9.87%. *Size-matched raw auxiliary proxy* repeats each raw auxiliary input to match the size of the default proxy batch, improving ASR by 1.89% over the raw auxiliary proxy. *Random mask proxy* selects regions independently of the joint heatmap, weakening the response-guided structure of the proxy context and reducing ASR by 6.62%.
- *Heatmap cues and fusion. w/o label-indicative heatmap* uses only the entropy-confidence cue, weakening class-relevant region selection and reducing ASR by 4.70%. *w/o entropy-confidence heatmap* uses only the label-indicative cue, weakening confidence-aware region selection and reducing ASR by 4.13%. *Additive fusion* aggregates the two normalized

Table 9: Additional proxy-construction ablations on CIFAR100-C under the TENT.

Variant	ASR (%)
PCAC	$83.16_{\pm 1.03}$
(i) Raw auxiliary proxy	$73.29_{\pm 1.21}$
(ii) Size-matched raw auxiliary proxy	$75.18_{\pm 1.34}$
(iii) Random mask proxy	$76.54_{\pm 1.48}$
(iv) w/o label-indicative heatmap	$78.46_{\pm 1.10}$
(v) w/o entropy-confidence heatmap	$79.03_{\pm 1.18}$
(vi) Additive fusion	$80.38_{\pm 1.22}$
(vii) Single mask ratio	$80.85_{\pm 1.05}$
(viii) w/o noise in proxy mixing	$81.72_{\pm 1.23}$

Table 10: Additional crafting-strategy comparison on CIFAR100-C under TENT. MI, DI, TI, EOT, MI+DI, and PGN are applied to the Direct crafting objective $\mathcal{J}_s$.

Crafting Variant	ASR (%)
Direct $\mathcal{J}_s$	$62.74_{\pm 1.42}$
Direct + MI	$66.21_{\pm 1.37}$
Direct + DI	$65.48_{\pm 1.56}$
Direct + TI	$62.90_{\pm 1.64}$
Direct + EOT w/ input transforms	$69.03_{\pm 1.48}$
Direct + MI + DI	$68.42_{\pm 1.31}$
PGN on $\mathcal{J}_s$	$76.11_{\pm 1.27}$
PCAC	$\mathbf{83.16_{\pm 1.03}}$

heatmaps additively rather than explicitly emphasizing their element-wise overlap, resulting in a 2.78% ASR reduction.

- *Proxy diversity. Single mask ratio* uses only one selection threshold, reducing masking-level diversity, and decreases ASR by 2.31%. *w/o noise in proxy mixing* removes one source of proxy diversity and decreases ASR by 1.44%. These smaller reductions suggest that multiple masking levels and proxy-mixing noise provide benefits to proxy construction.

C Additional Crafting-Strategy Comparison

We further compare PCAC with prior input-space transfer-enhancement strategies that reduce sensitivity to a particular input point or gradient realization. *MI* accumulates normalized gradients across crafting iterations to stabilize the update direction (Dong et al., 2018). *DI* applies differentiable random resizing and padding during surrogate crafting to introduce diverse input views and reduce surrogate-specific input overfitting (Xie et al., 2019). *MI+DI* combines gradient accumulation with input diversity. *TI* spatially smooths the input gradient with a normalized translation-invariant kernel, reducing dependence on a specific surrogate discriminative location (Dong et al., 2019). *EOT w/ input transforms* averages gradients over independently sampled input transformations, encouraging the crafted perturbations to remain effective across the selected transformation distribution (Athalye et al., 2018). *PGN* applies neighborhood sampling, interpolation between two neighboring gradients, and momentum accumulation to seek flatter input-space solutions under the surrogate crafting loss $\mathcal{J}_s(\boldsymbol{\delta}; \mathcal{S}, \omega)$ (Ge et al., 2023). For DI and EOT, transformations are applied only to the attacker-controlled inputs following (Cong et al., 2024), and for PGN, loss evaluation uses the same single-level surrogate-side approximation as Direct and PCAC.

We implement these variants based on the original methods [3] [4] [5]. For *MI*, we use a momentum decay factor of 1.0 and normalize the gradient of each attacker-controlled sample by its ℓ_1 norm before momentum accumulation (Dong et al., 2018). For *DI*, at each crafting iteration and with

[3] https://github.com/cihangxie/DI-2-FGSM

[4] https://github.com/dongyp13/Translation-Invariant-Attacks

[5] https://github.com/Trustworthy-AI-Group/PGN

Table 11: Impact of auxiliary-source distribution under TENT. The target evaluation domains use corruption severity $s_{\text{test}} = 3$. For the CIFAR10-C severity-mismatch setting, auxiliary samples are drawn from the held-out corruption at $s_{\text{aux}} \in \{1, 2, 4, 5\}$, and the result is averaged over these severity choices. All settings use 10 labeled auxiliary samples.

Dataset	Auxiliary Source	Distribution Setting	ASR (%)
CIFAR10-C	Held-out corruption	Default ($s_{\text{aux}} = 3$)	$70.46_{\pm 0.76}$
	Held-out corruption	Severity mismatch	$68.91_{\pm 0.94}$
	CINIC-10	External dataset	$65.84_{\pm 1.18}$
ImageNet-C	Held-out corruption	Default ($s_{\text{aux}} = 3$)	$78.67_{\pm 1.29}$
	ImageNet-V2	External dataset	$76.02_{\pm 1.07}$
	ImageNet-Sketch	External dataset	$72.58_{\pm 1.24}$

probability $p_{\text{DI}} = 0.5$, each current poisoned attacker-controlled input is independently resized to $r \times r$, where $r \sim \text{Unif}\{28, \ldots, 32\}$, and zero-padded at a random location back to 32×32 (Xie et al., 2019). For ***TI***, the gradient of each attacker-controlled input is convolved with a normalized 5×5 Gaussian kernel with $\sigma = 2/\sqrt{3}$, following the kernel construction of (Dong et al., 2019). For ***EOT w/ input transforms***, we average gradients over four sampled transformations from the same resize-and-padding distribution used by DI at each crafting iteration (Athalye et al., 2018). For ***PGN***, we use $N_{\text{PGN}} = 20$ neighborhood samples, interpolation coefficient $\beta_{\text{PGN}} = 0.5$, neighborhood radius $\zeta_{\text{PGN}} = 3\Lambda$, and momentum decay 1.0, following (Ge et al., 2023). All variants share $\Lambda = 8/255$, $R = 20$, and $\alpha = \Lambda/10$.

As shown in Table 10, the evaluated input-space crafting enhancements generally improve over direct center-point optimization. These strategies primarily modify input-space perturbation optimization, whereas PCAC additionally evaluates the crafting objective under a locally shifted surrogate state. PCAC achieves at least 7.05% higher ASR.

D Sensitivity to Auxiliary Data

D.1 Auxiliary Source Distribution

We evaluate PCAC under auxiliary sources with different degrees of distribution mismatch. In the default setting, the target evaluation domains use corruption severity $s_{\text{test}} = 3$, and auxiliary samples are drawn from one held-out corruption type at the same severity. For CIFAR10-C, we consider a severity-mismatch setting, where auxiliary samples are drawn from the same held-out corruption type but at non-default severity levels $s_{\text{aux}} \in \{1, 2, 4, 5\}$, and report the result averaged over these auxiliary severity choices. Following (Cong et al., 2024; Deng et al., 2021), we also consider external label-space-matched auxiliary datasets, using CINIC-10 for CIFAR10-C and ImageNet-V2 and ImageNet-Sketch for ImageNet-C (Darlow et al., 2018; Recht et al., 2019; Wang et al., 2019). As shown in Table 11, PCAC remains effective, with at most a 6.09% ASR drop.

D.2 Auxiliary-Data Quality

We further evaluate PCAC under synthetic auxiliary-label noise and auxiliary-input noise on CIFAR100-C. Following (Jiang et al., 2020), for label noise, at noise ratio $\nu \in \{0.1, 0.2, 0.3\}$, we randomly select $\nu|\mathcal{D}_{\text{aux}}|$ auxiliary samples without replacement and replace each selected label with a class sampled uniformly from the other $C - 1$ classes. For auxiliary-input noise, following the Gaussian-noise corruption setting commonly used in (Hendrycks & Dietterich, 2019), we add Gaussian noise to each auxiliary image before saliency estimation and proxy construction:

$$\widetilde{u}^{\text{aux}} = \text{clip}\left(u^{\text{aux}} + \boldsymbol{\zeta}, 0, 1\right), \qquad \boldsymbol{\zeta} \sim \mathcal{N}\left(0, \sigma_{\text{aux}}^2 I\right),$$

where $\sigma_{\text{aux}} \in \{0.01, 0.03, 0.05\}$. The noise is added independently to each pixel and channel in the unnormalized $[0, 1]$ pixel space, while the auxiliary labels remain unchanged.

As shown in Table 12, PCAC exhibits gradual performance degradation as auxiliary-data quality decreases. Nevertheless, PCAC retains 73.74% and 75.42% ASR, respectively, potentially benefiting from the combined label-indicative and entropy-confidence cues and the diverse proxy variants constructed at multiple masking levels.

Table 12: Sensitivity to auxiliary-label noise and auxiliary-input noise on CIFAR100-C under TENT. For label noise, the corresponding proportion of auxiliary labels is replaced with incorrect labels sampled uniformly from the other $C-1$ classes. For input noise, Gaussian noise is added in the unnormalized $[0,1]$ pixel space before saliency estimation and proxy construction. All settings use 10 auxiliary samples.

Auxiliary-Data Setting	ASR (%)
Clean auxiliary data	$83.16_{\pm 1.03}$
10% label noise	$81.68_{\pm 1.24}$
20% label noise	$78.25_{\pm 1.61}$
30% label noise	$73.74_{\pm 1.58}$
Gaussian input noise, $\sigma_{\text{aux}} = 0.01$	$82.03_{\pm 1.19}$
Gaussian input noise, $\sigma_{\text{aux}} = 0.03$	$78.55_{\pm 1.29}$
Gaussian input noise, $\sigma_{\text{aux}} = 0.05$	$75.42_{\pm 1.47}$

E Theoretical Analysis

To formalize the dual-neighborhood intuition, inspired by (Ge et al., 2023; Qiu et al., 2026), we characterize the surrogate-to-target discrepancy through the crafting gap:

Surrogate-Target Crafting Gap. Let

$$\mathcal{J}_s\left(\boldsymbol{\delta}; \mathcal{S}, \omega\right) = \mathcal{L}_A^{\text{TC}}\left(\boldsymbol{\delta}; \mathcal{S}, \omega\right) + \lambda_{\text{ab}} \mathcal{L}_{A\to S}^{\text{tar}}\left(\boldsymbol{\delta}; \mathcal{S}, \omega\right)$$

denote the surrogate crafting loss computed using the proxy context $\mathcal{S}$ and surrogate state ω.

With $\mathcal{B}_a$ fixed as the attacker-controlled set, let

$$\mathcal{J}_t\left(\boldsymbol{\delta}; \mathcal{B}_b, \theta\right) = \frac{1}{|\mathcal{B}_b|} \sum_{x_i' \in \mathcal{B}_b} \mathcal{L}_{\text{atk}}\left(x_i', \ell_{\text{tar}}; \theta'\left(\boldsymbol{\delta}; \mathcal{B}_b, \theta\right)\right)$$

denote the unavailable target-side crafting loss induced by the attack goal. Here, $\theta'(\boldsymbol{\delta}; \mathcal{B}_b, \theta)$ denotes the target model state obtained by applying TTA from the pre-adaptation target state θ to the attacked test batch $\mathcal{B}_a^{\boldsymbol{\delta}} \cup \mathcal{B}_b$. This target-side loss remains unavailable under the threat model, where $\mathcal{B}_b$, θ, θ', and target-side gradients are inaccessible.

The surrogate-to-target crafting-objective discrepancy is defined as

$$\mathcal{G}\left(\boldsymbol{\delta}\right) = \left|\mathcal{J}_t\left(\boldsymbol{\delta}; \mathcal{B}_b, \theta\right) - \mathcal{J}_s\left(\boldsymbol{\delta}; \mathcal{S}, \omega\right)\right|.$$

For a local perturbation probe $\mathbf{u}$, define

$$\boldsymbol{\delta}_{\mathbf{u}} = \Pi_{\mathcal{A}_\Lambda}\left(\boldsymbol{\delta} + \mathbf{u}\right), \quad \mathbf{u} \in B_{\xi_\delta}(0),$$

where $B_{\xi_\delta}(0) = \{\mathbf{u} : \|\mathbf{u}\|_p \le \xi_\delta\}, \quad \mathcal{A}_\Lambda = \{\boldsymbol{\delta} : \|\boldsymbol{\delta}_i\|_p \le \Lambda, \forall x_i' \in \mathcal{B}_a\}.$

Let π_{ξ_δ} be probability distribution supported on $B_{\xi_\delta}(0)$. The perturbation-neighborhood surrogate objective is

$$\mathcal{M}_{s,\xi_\delta}\left(\boldsymbol{\delta}; \mathcal{S}, \omega\right) = \mathbb{E}_{\mathbf{u}\sim\pi_{\xi_\delta}}\left[\mathcal{J}_s\left(\boldsymbol{\delta}_{\mathbf{u}}; \mathcal{S}, \omega\right)\right],$$

and the corresponding unavailable target-side neighborhood objective is

$$\mathcal{M}_{t,\xi_\delta}\left(\boldsymbol{\delta}; \mathcal{B}_b, \theta\right) = \mathbb{E}_{\mathbf{u}\sim\pi_{\xi_\delta}}\left[\mathcal{J}_t\left(\boldsymbol{\delta}_{\mathbf{u}}; \mathcal{B}_b, \theta\right)\right].$$

Proposition 1. Perturbation-Side Local Variation. Define the surrogate-side local crafting variation as

$$R_{\mathcal{J},\xi_\delta,s}^{(0)}\left(\boldsymbol{\delta}\right) = \mathbb{E}_{\mathbf{u}\sim\pi_{\xi_\delta}}\left[\left|\mathcal{J}_s\left(\boldsymbol{\delta}_{\mathbf{u}}; \mathcal{S}, \omega\right) - \mathcal{J}_s\left(\boldsymbol{\delta}; \mathcal{S}, \omega\right)\right|\right].$$

Define the unavailable target-side local crafting variation as

$$R_{\mathcal{J},\xi_\delta,t}^{(0)}\left(\boldsymbol{\delta}\right) = \mathbb{E}_{\mathbf{u}\sim\pi_{\xi_\delta}}\left[\left|\mathcal{J}_t\left(\boldsymbol{\delta}_{\mathbf{u}}; \mathcal{B}_b, \theta\right) - \mathcal{J}_t\left(\boldsymbol{\delta}; \mathcal{B}_b, \theta\right)\right|\right],$$

and define the local discrepancy as

$$C_{\mathcal{J},\xi_\delta}^{s,t}\left(\boldsymbol{\delta}\right) = \mathbb{E}_{\mathbf{u}\sim\pi_{\xi_\delta}}\left[\left|\mathcal{J}_t\left(\boldsymbol{\delta}_{\mathbf{u}}; \mathcal{B}_b, \theta\right) - \mathcal{J}_s\left(\boldsymbol{\delta}_{\mathbf{u}}; \mathcal{S}, \omega\right)\right|\right].$$

Then, for any feasible perturbation $\boldsymbol{\delta}$,

$$\mathcal{G}\left(\boldsymbol{\delta}\right) \le R_{\mathcal{J},\xi_\delta,s}^{(0)}\left(\boldsymbol{\delta}\right) + C_{\mathcal{J},\xi_\delta}^{s,t}\left(\boldsymbol{\delta}\right) + R_{\mathcal{J},\xi_\delta,t}^{(0)}\left(\boldsymbol{\delta}\right).$$

Here, $C^{s,t}_{\mathcal{J},\xi_\delta}$ and $R^{(0)}_{\mathcal{J},\xi_\delta,t}$ denote the unavailable local surrogate-to-target discrepancy and target-side perturbation variation, respectively, and are retained as residual terms in this decomposition.

Proof. For any local perturbation probe $\mathbf{u}$, add and subtract the target and surrogate losses at $\boldsymbol{\delta}_\mathbf{u}$:

$$\begin{aligned}
&|\mathcal{J}_t(\boldsymbol{\delta};\mathcal{B}_b,\theta)-\mathcal{J}_s(\boldsymbol{\delta};\mathcal{S},\omega)|\\
\le\ &|\mathcal{J}_t(\boldsymbol{\delta};\mathcal{B}_b,\theta)-\mathcal{J}_t(\boldsymbol{\delta}_\mathbf{u};\mathcal{B}_b,\theta)|\\
&+|\mathcal{J}_t(\boldsymbol{\delta}_\mathbf{u};\mathcal{B}_b,\theta)-\mathcal{J}_s(\boldsymbol{\delta}_\mathbf{u};\mathcal{S},\omega)|\\
&+|\mathcal{J}_s(\boldsymbol{\delta}_\mathbf{u};\mathcal{S},\omega)-\mathcal{J}_s(\boldsymbol{\delta};\mathcal{S},\omega)|.
\end{aligned}$$

Taking expectation over $\mathbf{u}\sim\pi_{\xi_\delta}$ proves the bound. Since $\mathcal{J}_s$ is non-negative, we also have

$$|a-b|\le a+b,\quad a,b\ge 0.$$

Therefore, $R^{(0)}_{\mathcal{J},\xi_\delta,s}(\boldsymbol{\delta})\le\mathcal{J}_s(\boldsymbol{\delta};\mathcal{S},\omega)+\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega)$. For a center-included probe distribution satisfying

$$\pi_0:=\pi_{\xi_\delta}(\{\mathbf{0}\})>0,$$

the feasibility of $\boldsymbol{\delta}$ gives $\boldsymbol{\delta}_\mathbf{0}=\boldsymbol{\delta}$. Together with the non-negativity of $\mathcal{J}_s$, this yields

$$\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega)\ge\pi_0\mathcal{J}_s(\boldsymbol{\delta};\mathcal{S},\omega).$$

Consequently, $R^{(0)}_{\mathcal{J},\xi_\delta,s}(\boldsymbol{\delta})\le\left(1+\frac{1}{\pi_0}\right)\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega)$. Thus, under center inclusion, the perturbation-neighborhood objective upper-bounds the surrogate-side local variation up to the factor $1+1/\pi_0$.

Proposition 2. Surrogate-State Sensitivity. For a local offset $\boldsymbol{\epsilon}$ applied to the differentiable surrogate weights represented by ω, write

$$\omega[\boldsymbol{\epsilon}]=\omega+\boldsymbol{\epsilon}$$

as the shifted surrogate state. Define the shifted perturbation-neighborhood surrogate-target discrepancy as

$$C^{s,t}_{\mathcal{M},\xi_\delta}(\boldsymbol{\delta};\boldsymbol{\epsilon})=|\mathcal{M}_{t,\xi_\delta}(\boldsymbol{\delta};\mathcal{B}_b,\theta)-\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega[\boldsymbol{\epsilon}])|,$$

and define the surrogate-state sensitivity of the perturbation-neighborhood objective as

$$P^{\omega}_{\mathcal{M}}(\boldsymbol{\delta};\boldsymbol{\epsilon})=|\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega[\boldsymbol{\epsilon}])-\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega)|.$$

Then, for any parameter offset $\boldsymbol{\epsilon}$,

$$\begin{aligned}
\mathcal{G}(\boldsymbol{\delta})\le\ &R^{(0)}_{\mathcal{J},\xi_\delta,t}(\boldsymbol{\delta})+C^{s,t}_{\mathcal{M},\xi_\delta}(\boldsymbol{\delta};\boldsymbol{\epsilon})\\
&+P^{\omega}_{\mathcal{M}}(\boldsymbol{\delta};\boldsymbol{\epsilon})+R^{(0)}_{\mathcal{J},\xi_\delta,s}(\boldsymbol{\delta}).
\end{aligned}$$

Equivalently, $\mathcal{G}(\boldsymbol{\delta})\le R^{(0)}_{\mathcal{J},\xi_\delta,s}(\boldsymbol{\delta})+P^{\omega}_{\mathcal{M}}(\boldsymbol{\delta};\boldsymbol{\epsilon})+\mathcal{R}_{\mathrm{res}}(\boldsymbol{\delta};\boldsymbol{\epsilon})$, where $\mathcal{R}_{\mathrm{res}}(\boldsymbol{\delta};\boldsymbol{\epsilon})=R^{(0)}_{\mathcal{J},\xi_\delta,t}(\boldsymbol{\delta})+C^{s,t}_{\mathcal{M},\xi_\delta}(\boldsymbol{\delta};\boldsymbol{\epsilon})$. Here, $\mathcal{R}_{\mathrm{res}}$ consists of the target-side local perturbation variation and the shifted perturbation-neighborhood surrogate-to-target discrepancy. We impose no assumption that $\mathcal{R}_{\mathrm{res}}$ is small. Because $\mathcal{M}_{s,\xi_\delta}$ is non-negative,

$$\begin{aligned}
P^{\omega}_{\mathcal{M}}(\boldsymbol{\delta};\boldsymbol{\epsilon})\le\ &\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega[\boldsymbol{\epsilon}])\\
&+\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega).
\end{aligned}$$

Proof. Add and subtract the perturbation-neighborhood target objective, the shifted perturbation-neighborhood surrogate objective, and the original perturbation-neighborhood surrogate objective:

$$\begin{aligned}
&\mathcal{G}(\boldsymbol{\delta})\\
\le\ &|\mathcal{J}_t(\boldsymbol{\delta};\mathcal{B}_b,\theta)-\mathcal{M}_{t,\xi_\delta}(\boldsymbol{\delta};\mathcal{B}_b,\theta)|\\
&+|\mathcal{M}_{t,\xi_\delta}(\boldsymbol{\delta};\mathcal{B}_b,\theta)-\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega[\boldsymbol{\epsilon}])|\\
&+|\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega[\boldsymbol{\epsilon}])-\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega)|\\
&+|\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta};\mathcal{S},\omega)-\mathcal{J}_s(\boldsymbol{\delta};\mathcal{S},\omega)|.
\end{aligned}$$

The first and fourth terms are bounded by $R^{(0)}_{\mathcal{J},\xi_\delta,t}$ and $R^{(0)}_{\mathcal{J},\xi_\delta,s}$, respectively, by Jensen's inequality and the triangle inequality. The second and third terms are exactly $C^{s,t}_{\mathcal{M},\xi_\delta}$ and $P^{\omega}_{\mathcal{M}}$. This gives the stated bound. The relaxation of $P^{\omega}_{\mathcal{M}}$ follows from $|a-b|\le a+b$ and non-negativity of $\mathcal{M}_{s,\xi_\delta}$.

Connection to PCAC. In Algorithm 1, the expectation defining $\mathcal{M}_{s,\xi_\delta}$ is implemented by a finite center-included average using the center offset $\mathbf{u}^r_0=0$ and K uniformly sampled local

Table 13: Computing environment used for the experiments.

Component	Configuration
Operating system	Ubuntu 20.04.6 LTS
GPU	NVIDIA RTX 3090, 24 GB
Python	3.8.18
PyTorch	1.13.1
TorchVision	0.14.1
CUDA	11.7
cuDNN	8.5.0
NumPy	1.23.5
SciPy	1.10.1
Pillow	9.5.0
timm	0.6.13
Random seeds	$\{0, 1, 2, 3, 4\}$

Table 14: Base input preprocessing and channel-wise normalization.

Dataset	Spatial preprocessing	Channel mean	Channel standard deviation
CIFAR10-C	32×32; no resize/crop	$(0.4914, 0.4822, 0.4465)$	$(0.2023, 0.1994, 0.2010)$
CIFAR100-C	32×32; no resize/crop	$(0.5070, 0.4865, 0.4409)$	$(0.2673, 0.2564, 0.2761)$
ImageNet-C	224×224; no additional resize/crop	$(0.485, 0.456, 0.406)$	$(0.229, 0.224, 0.225)$

probes, with weight $1/(K+1)$ assigned to each term. At crafting iteration r, PCAC computes $\mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}^r; \mathcal{S}, \omega) = \frac{1}{K+1} \sum_{j=0}^{K} \mathcal{J}_s\left(\boldsymbol{\delta}_j^r; \mathcal{S}, \omega\right)$, where $\mathbf{u}_0^r = 0, \mathbf{u}_j^r \sim \mathrm{Unif}(B_{\xi_\delta}(0)), j = 1, \ldots, K$, and $\boldsymbol{\delta}_j^r = \Pi_{\mathcal{A}_\Lambda}\left(\boldsymbol{\delta}^r + \mathbf{u}_j^r\right)$. This average is the implementation counterpart of the perturbation-neighborhood objective considered in Proposition 1. Because it explicitly includes both the center and the sampled local probes, the resulting update accounts for their surrogate losses rather than relying only on a single perturbation point (Ge et al., 2023; Xu et al., 2023).

PCAC then constructs the gradient-aligned local surrogate-state shift $\boldsymbol{\epsilon}_r^\star = \rho_\omega \frac{\nabla_\omega \mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}^r; \mathcal{S}, \omega)}{\left\|\nabla_\omega \mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}^r; \mathcal{S}, \omega)\right\|_2 + \tau}, \omega^{r,\star} = \omega[\boldsymbol{\epsilon}_r^\star] = \omega + \boldsymbol{\epsilon}_r^\star$. The perturbation is updated using $\Psi_r = \mathcal{M}_{s,\xi_\delta}(\boldsymbol{\delta}^r; \mathcal{S}, \omega^{r,\star})$, while treating $\boldsymbol{\epsilon}_r^\star$ as fixed during the update to avoid higher-order derivatives (Foret et al., 2021).

F Implementation Details

Computing Environment. All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24 GB memory under Ubuntu 20.04.6 LTS. Table 13 summarizes the software environment and random seeds.

Data Split and Input Preprocessing. Following (Hendrycks & Dietterich, 2019), $\mathcal{D}_{\mathrm{test}}$ is constructed to exclude samples corresponding to the same underlying clean images as those in $\mathcal{D}_{\mathrm{aux}}$ to avoid content overlap. For CIFAR10-C and CIFAR100-C, corrupted images are kept at their original 32×32 resolution without resizing or cropping. For ImageNet-C, corrupted images are kept at their released 224×224 resolution without additional resizing or cropping. Images are represented as floating-point values in $[0, 1]$. All perturbations, local perturbation probes, proxy mixing, and Gaussian proxy noise are applied in the unnormalized pixel space. Modified inputs are clipped to $[0, 1]$ before channel-wise normalization, which is applied before model input. Unless otherwise specified, the common preprocessing pipeline applies no additional stochastic data augmentation. Table 14 summarizes the normalization constants.

G Adaptation of Baselines

Following the baseline-adaptation practice of (Su et al., 2025; Zhao et al., 2021), we adapt prior TTA poisoning attacks to our targeted benign co-batch misclassification setting, where poisoned inputs are crafted to induce benign co-batched samples to be predicted as the target label ℓ_{tar}, without the target-model access or target queries, real benign co-batches, or training-similar auxiliary data dur-

Table 15: Baseline parameter settings.

Baseline	Method-Specific Setting	Value
TePA	Momentum / DI probability / min. resize ratio	$1.0/0.5/0.8$
FCA	Feature location / centroid EMA momentum	avgpool / 0.6
FCA	Cosine numerical constant	10^{-6}
FCA	Dispersion / collapse / centroid coefficients	$+1/-1/+1$
NHE	Multiplier init. / frequency / step	0 / once per step / 10^{-4}
BLE	Multiplier init. / frequency / step	0 / once per step / 10^{-3}
BLE	Class-wise EMA momentum	0.8

ing crafting. Specifically, target-model access or queries are replaced, where required, by the same surrogate model f_ω, while unavailable benign co-batches or training-distribution-matched auxiliary data are replaced using the same available auxiliary source $\mathcal{D}_{\text{aux}}$. Unless otherwise specified, all attacks use the same perturbation budget $\Lambda = 8/255$, $R = 20$ crafting iterations, and step size $\alpha = \Lambda/10$. For each trial, compared attacks use the same target batch, attacker/benign split, and target-label draw.

We form a size-matched auxiliary co-batch $\widetilde{\mathcal{D}}_{\text{aux}}$ by repeating the samples in $\mathcal{D}_{\text{aux}}$, such that it has the same size as the proxy co-batch used by PCAC, i.e., $|\widetilde{\mathcal{D}}_{\text{aux}}| = |\mathcal{S}|$, following prior settings with batch-conditioned co-batch adaptation (Cong et al., 2024; Rifat et al., 2025), and motivated by the size-matched control in Table 9, variant (ii). For methods requiring a batch-conditioned adapted model state, we denote by ω' the surrogate state obtained through the single-level recomputation approximation on $\mathcal{B}_a^{\boldsymbol{\delta}} \cup \widetilde{\mathcal{D}}_{\text{aux}}$, following (Wu et al., 2023; Su et al., 2025).

Following targeted-attack formulations (Zhao et al., 2021; Kim et al., 2023; Li et al., 2020), for baselines without an explicit target-label term, we introduce a target-conditioning term toward ℓ_{tar}, while retaining their method-specific objective components. Following (Wu et al., 2023; Cong et al., 2024; Rifat et al., 2025; Su et al., 2025) and corresponding implementations[6] [7] [8] [9], we report the baseline parameter settings in Table 15.

TargetCE. TargetCE directly optimizes the attacker-controlled samples toward the target specified label ℓ_{tar}. Its crafting loss is

$$\mathcal{L}^{\text{adapt}}_{\text{TargetCE}}(\boldsymbol{\delta}) = \frac{1}{|\mathcal{B}_a|} \sum_{x'_i \in \mathcal{B}_a} \text{CE}\left(f_\omega(x'_i + \delta_i), \ell_{\text{tar}}\right).$$

DIA. DIA originally crafts poisoned samples using real benign co-batched samples and target-model access (Wu et al., 2023). We replace the target model with the surrogate model and the co-batch with $\widetilde{\mathcal{D}}_{\text{aux}}$. The adapted DIA loss is

$$\mathcal{L}^{\text{adapt}}_{\text{DIA}}(\boldsymbol{\delta}) = \frac{1}{|\widetilde{\mathcal{D}}_{\text{aux}}|} \sum_{u \in \widetilde{\mathcal{D}}_{\text{aux}}} \text{CE}\left(f_{\omega'}(u), \ell_{\text{tar}}\right).$$

TePA. For TePA (Cong et al., 2024), we replace its training-distribution-matched seed data with $\widetilde{\mathcal{D}}_{\text{aux}}$, retain its entropy-based term and method-specific crafting updates, and introduce target-label conditioning on the attacker-controlled samples. The adapted TePA loss is

$$\begin{aligned}\mathcal{L}^{\text{adapt}}_{\text{TePA}}(\boldsymbol{\delta}) = \mathcal{L}^{\text{base}}_{\text{TePA}}(\boldsymbol{\delta}) \\ + \lambda_{\text{tar}} \frac{1}{|\mathcal{B}_a|} \sum_{x'_i \in \mathcal{B}_a} \text{CE}\left(f_{\omega'}(x'_i + \delta_i), \ell_{\text{tar}}\right),\end{aligned}$$

where $\mathcal{L}^{\text{base}}_{\text{TePA}}$ denotes TePA's entropy-based term expressed under the minimization convention, and $\lambda_{\text{tar}} > 0$ controls the strength. The retained momentum and input-diversity settings are reported in Table 15.

[6] https://github.com/inspire-group/tta_risk
[7] https://github.com/tianshuocong/TePA
[8] https://github.com/Restuccia-Group/tta-adv
[9] https://github.com/Gorilla-Lab-SCUT/RTTDP

FCA. For FCA (Rifat et al., 2025), we compute its feature-based term using surrogate features extracted at the adaptive average-pooling layer and replace the real benign co-batch set with $\widetilde{\mathcal{D}}_{\mathrm{aux}}$. We apply target specified-label conditioning to the auxiliary co-batch samples. The adapted FCA loss is

$$\mathcal{L}_{\mathrm{FCA}}^{\mathrm{adapt}}(\boldsymbol{\delta}) = \mathcal{L}_{\mathrm{FCA}}^{\mathrm{base}}(\boldsymbol{\delta}) + \lambda_{\mathrm{tar}} \frac{1}{|\widetilde{\mathcal{D}}_{\mathrm{aux}}|} \sum_{u \in \widetilde{\mathcal{D}}_{\mathrm{aux}}} \mathrm{CE}\left(f_{\omega'}(u), \ell_{\mathrm{tar}}\right).$$

Here, $\mathcal{L}_{\mathrm{FCA}}^{\mathrm{base}}$ denotes feature-dispersion, feature-collapse, and centroid-alignment terms expressed under the minimization convention. The FCA settings are reported in Table 15.

NHE and BLE. Following the formulations of NHE and BLE in (Su et al., 2025), we use their method-specific attack terms and feature-consistency regularization. Under our no-query setting, we use f_ω in place of the surrogate updated through target-model query feedback.

The adapted NHE loss is

$$\mathcal{L}_{\mathrm{NHE}}^{\mathrm{adapt}}(\boldsymbol{\delta}) = \mathcal{L}_{\mathrm{NHE}}^{\mathrm{base}}(\boldsymbol{\delta}) + \lambda_{\mathrm{tar}} \frac{1}{|\mathcal{B}_a|} \sum_{x'_i \in \mathcal{B}_a} \mathrm{CE}\left(f_{\omega'}(x'_i + \delta_i), \ell_{\mathrm{tar}}\right).$$

The adapted BLE loss is

$$\mathcal{L}_{\mathrm{BLE}}^{\mathrm{adapt}}(\boldsymbol{\delta}) = \mathcal{L}_{\mathrm{BLE}}^{\mathrm{base}}(\boldsymbol{\delta}) + \lambda_{\mathrm{tar}} \frac{1}{|\mathcal{B}_a|} \sum_{x'_i \in \mathcal{B}_a} \mathrm{CE}\left(f_{\omega'}(x'_i + \delta_i), \ell_{\mathrm{tar}}\right).$$

Here, $\mathcal{L}_{\mathrm{NHE}}^{\mathrm{base}}$ and $\mathcal{L}_{\mathrm{BLE}}^{\mathrm{base}}$ denote the corresponding method-specific attack terms together with their feature-consistency regularization at the current multiplier values, expressed under the minimization convention. Following the procedure in released implementation of (Su et al., 2025), the feature multipliers are initialized at zero and updated once per crafting step using step sizes 10^{-4} for NHE and 10^{-3} for BLE. BLE additionally uses class-wise EMA momentum 0.8.

White-box Crafting. We use the target-side crafting objective $J_t(\delta; \mathcal{B}_b, \theta)$ defined in Section 2.2 as a white-box reference. This setting has access to the target model f_θ, the real benign co-batch set $\mathcal{B}_b$, and the deployed TTA procedure, and directly optimizes the perturbations against the resulting batch-adapted target model f'_θ.