READ THE TRAJECTORY, NOT JUST THE ENDPOINT: CALIBRATED UTILITY READOUT FOR TRANSFER ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Iterative transfer attacks are widely used to evaluate adversarial vulnerability on unseen models. Although their optimization trajectories contain multiple legal adversarial actions, existing methods almost universally deploy the final iterate. However, source optimization and cross-model transferability need not peak at the same iteration, and the preferred action can vary across images. Replacing the endpoint with one global checkpoint is therefore insufficient, while target-guided action selection would violate the zero-query transfer setting. To address these problems, we propose **Calibrated Pointwise Relative Utility (CPRU)**, a target-output-free trajectory readout that estimates the signed utility of each saved action relative to the conventional final iterate. We show that the target-clean mean of this utility is exactly equal to the paired attack success rate (ASR) gain. CPRU uses four target-disjoint proxy models and an attack-specific utility regressor to score all non-final actions, independently returning the highest predicted positive-utility action while retaining final as an exact zero-utility fallback. Extensive experiments across different iterative attacks, data sets, source models, and target architectures demonstrate the effectiveness of image-specific trajectory readout.

1 INTRODUCTION

Adversarial examples can transfer across models, allowing attacks crafted on a white-box surrogate to affect unseen targets (Szegedy et al., 2014; Goodfellow et al., 2015; Papernot et al., 2017). This property underlies query-free black-box attacks and robustness evaluation. Existing approaches mainly modify trajectory generation through momentum (Dong et al., 2018), input transformations (Xie et al., 2019), translation-invariant gradients (Dong et al., 2019), variance tuning (Wang & He, 2021), intermediate representations (Huang et al., 2019), modified backpropagation or feature importance (Wu et al., 2020; Wang et al., 2021; Zhang et al., 2022), spatial redistribution (Gao et al., 2020), adaptive coordinate-wise dynamics (Tao et al., 2026), or local perturbation augmentation (Mi et al., 2026). Transferability also depends strongly on the surrogate–target pair and attack configuration (Demontis et al., 2019; Li et al., 2023), yet the final iterate is still almost universally deployed after generation.

This endpoint convention need not align with cross-model transferability. Continued optimization can specialize a perturbation to the source after transfer has saturated or deteriorated (Demontis et al., 2019; Gubri et al., 2023; Tao et al., 2026), while intermediate representations, checkpoints, and discarded candidates may preserve transferable information (Huang et al., 2019; Li et al., 2025; Yang et al., 2025). Because the useful trajectory location can differ across images, endpoint-only deployment discards an already generated sample-dependent degree of freedom. A globally fixed early checkpoint still assigns the same action to every image, target-guided selection violates the query-free setting, and averaging intermediate candidates constructs a new perturbation rather than choosing one saved legal state (Yang et al., 2025). These limitations motivate a target-output-free rule that predicts which action is preferable to final for each image.

We introduce **Calibrated Pointwise Relative Utility (CPRU)**, which models the signed utility of every intermediate action relative to the conventional endpoint. Four fixed proxy models encode the trajectory state, and an attack-specific regressor predicts the target-clean value of replacing fi-

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.

nal. Each image receives its highest predicted positive-utility action, while final remains an exact zero-utility fallback. The signed construction distinguishes recovery of a transfer success from destruction of a success already achieved by final; its empirical mean is exactly the paired target-clean ASR change. CPRU therefore aligns reader training with evaluation, adapts deployment by image, and leaves the underlying attack and perturbation budget unchanged.

Our **main contributions** can be summarized as follows:

1. We reveal endpoint selection as a previously overlooked degree of freedom in transferable adversarial attacks. We formulate the value of an intermediate action by its signed utility relative to the conventional final iterate and show that its target-clean empirical mean is exactly equal to the paired ASR change. We further establish the expected advantage available to image-specific trajectory selection over a globally fixed checkpoint.

2. We develop CPRU, a target-output-free pointwise trajectory readout method. We show that target-disjoint proxy trajectories contain predictive information about the relative utility of intermediate actions on unseen formal targets, enabling image-specific action selection without altering attack generation or querying the target.

3. We conduct extensive experiments across attacks, targets, sources, and data sets. Five iterative attacks are evaluated on CIFAR-10 and native-resolution ImageNet-1K, with unseen-target, second-source, and LPAA comparisons. Controlled studies further examine proxy number, trajectory density, relative-to-final modeling, development-set size, and reader reuse.

2 Related Works and Unsolved Problems

We first introduce the standard framework of iterative transfer attacks and then formalize the limitations induced by final-iterate deployment.

2.1 Iterative Transfer Attacks

Let $f_s : [0,1]^d \to \mathbb{R}^C$ be a white-box source classifier, $x \in [0,1]^d$ be an image with label y, and ϵ be the perturbation budget. The feasible adversarial region is

$$\mathcal{B}(x,\epsilon) = \{x' \in [0,1]^d : \|x' - x\|_\infty \le \epsilon\}. \tag{1}$$

A standard untargeted attack searches for

$$\max_{x' \in \mathcal{B}(x,\epsilon)} \mathcal{L}(f_s(x'), y). \tag{2}$$

Iterative attacks approximately solve 2 by producing a trajectory

$$\mathcal{A}(x) = \{x_t\}_{t=1}^{T}, \qquad x_t \in \mathcal{B}(x,\epsilon), \tag{3}$$

whose projected sign-gradient update can be written as

$$x_{t+1} = \Pi_{\mathcal{B}(x,\epsilon)}(x_t + \alpha_t \operatorname{sign}(\nabla_{x_t} \mathcal{L}(f_s(x_t), y))). \tag{4}$$

MI-FGSM accumulates historical gradients through momentum (Dong et al., 2018), DI^2-FGSM differentiates through stochastic input transformations (Xie et al., 2019), and translation-invariant or variance-tuned attacks aggregate local gradient information (Dong et al., 2019; Wang & He, 2021). PI-FGSM redistributes projected overflow (Gao et al., 2020), MDCS adapts coordinate-wise dynamics (Tao et al., 2026), and LPAA aggregates gradients from masked, scaled perturbation subspaces with a dedicated initialization (Mi et al., 2026). Intermediate-level attacks and modified backward paths further reshape the trajectory (Huang et al., 2019; Wu et al., 2020; Wang et al., 2021; Zhang et al., 2022). These generation methods usually share the deployment rule

$$x_{\text{adv}} = x_T. \tag{5}$$

Thus an attack generates T legal states but conventionally retains only the last one.

2.2 Problems of Final-Iterate Deployment

Let f_q be an unseen target classifier. For image i, define clean correctness and attack success at step t by

$$C_{q,i} = \mathbf{1}\{f_q(x_i) = y_i\}, \qquad A_{q,i,t} = \mathbf{1}\{f_q(x_{i,t}) \ne y_i\}. \tag{6}$$

The target-clean ASR of a fixed trajectory position is

$$\mathrm{ASR}_q(t) = \frac{\sum_i C_{q,i} A_{q,i,t}}{\sum_i C_{q,i}}. \tag{7}$$

The source loss $\ell_s(t) = \mathcal{L}(f_s(x_t), y)$ is optimized on f_s. It does not imply monotonic improvement on f_q,

$$\ell_s(t+1) \geq \ell_s(t) \not\Longrightarrow \mathrm{ASR}_q(t+1) \geq \mathrm{ASR}_q(t). \tag{8}$$

This discrepancy agrees with analyses of source–target alignment, local geometry, and surrogate overfitting (Demontis et al., 2019; Gubri et al., 2023; Li et al., 2023). It also accords with evidence that useful transferable information can occur before the terminal representation (Huang et al., 2019; Tao et al., 2026).

The conventional rule implicitly treats T as the preferred action. For diagnostic analysis, suppose formal-target outcomes were available and define the best globally fixed trajectory position as

$$t_q^{\mathrm{fix}} \in \arg\max_{1 \leq t \leq T} \mathrm{ASR}_q(t). \tag{9}$$

If target information were available, an image-specific oracle could choose

$$t_{q,i}^{\star} \in \arg\max_{1 \leq t \leq T} A_{q,i,t}. \tag{10}$$

Because T is included in the fixed-checkpoint search, the resulting performance obeys

$$\mathrm{ASR}_q^{\mathrm{oracle}} \geq \mathrm{ASR}_q(t_q^{\mathrm{fix}}) \geq \mathrm{ASR}_q(T). \tag{11}$$

The second inequality measures target-side global endpoint suboptimality. The first captures additional headroom created by image-dependent trajectory positions. This target-optimal diagnostic is distinct from the deployable Dev-fixed baseline selected only on development models in Section 4.2. A global checkpoint cannot exploit image-dependent heterogeneity.

2.3 Existing Remedies and the Remaining Problem

A checkpoint chosen on a development population is deployable without target feedback, but it applies one step to every image. Target-guided selection can adapt by image, but it violates zero-query evaluation. Adversarial Example Soups reuses discarded candidates by averaging them (Yang et al., 2025). CPRU instead selects one saved legal action and preserves the original perturbation set. Absolute proxy success is also an incomplete decision statistic. A candidate with high proxy ASR can merely duplicate final or can replace a final success with a failure. The required target is the incremental value of replacement,

$$\mathbb{E}[A_{q,i,t} - A_{q,i,T} \mid Z_{i,t}], \tag{12}$$

where $Z_{i,t}$ contains only target-disjoint evidence. This relative objective connects trajectory readout to calibration (Guo et al., 2017), uplift modeling (Gutierrez & Gérardy, 2017), and selective prediction (Geifman & El-Yaniv, 2017; 2019). The unresolved problem is to estimate 12 for every image and action while retaining final as a valid fallback and keeping the formal target completely outside the decision process.

3 Methodology

An iterative attack produces legal actions $\mathcal{A}_i = \{x_{i,t}\}_{t=1}^{T}$ and conventionally deploys $x_{i,T}$. CPRU assigns each action a signed value relative to this endpoint,

$$u_{q,i,t} = A_{q,i,t} - A_{q,i,T} \in \{-1, 0, 1\}, \qquad u_{q,i,T} = 0. \tag{13}$$

The method must align learning with target-clean ASR, adapt the returned step to each image, and remain independent of the formal target. Here, *calibrated* means that reader scores use the signed relative-utility scale aligned with paired target-clean ASR, rather than probabilistic calibration. Figure 1 summarizes the complete frozen-readout pipeline.

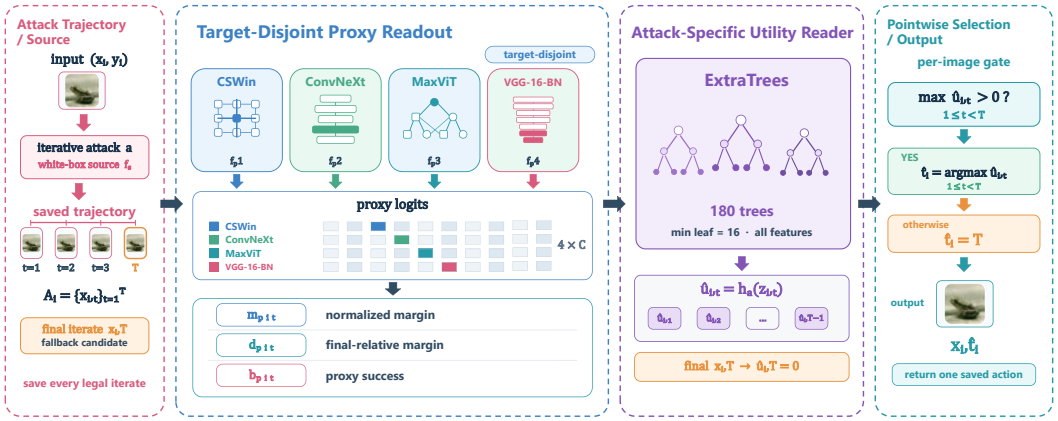


Figure 1: The CPRU methodology. An unchanged source attack produces legal trajectory states; target-disjoint proxies encode every state, a frozen attack-specific reader estimates final-relative utility, and a pointwise gate returns one saved action without formal-target access.

3.1 Signed Relative Utility and Exact ASR Alignment

For a target q, let $C_{q,i} = \mathbf{1}\{f_q(x_i) = y_i\}$ be clean correctness and let $A_{q,i,t} = \mathbf{1}\{f_q(x_{i,t}) \neq y_i\}$ be attack success. Utility $+1$ in 13 is a recovery because action t succeeds while final fails. Utility -1 is a destruction because action t fails while final succeeds. Utility zero covers agreement and the final fallback.

For a development population of N images and Q development models, define $N_q = \sum_{i=1}^{N} C_{q,i}$. The regression target is

$$y_{i,t} = \frac{1}{Q} \sum_{q=1}^{Q} \frac{N}{N_q} C_{q,i} u_{q,i,t}. \tag{14}$$

The factor N/N_q preserves each target-clean denominator when utilities are averaged by image.

Proposition 1 (Exact utility alignment)**.** *For any target-output-free reader π, the target-clean empirical mean of its selected signed utilities equals the paired ASR gain over final. Equal averaging across targets preserves the identity. The finite-sample proof is provided in Appendix A.1.*

For one target, Proposition 1 gives

$$\mathrm{ASR}_q(\pi) - \mathrm{ASR}_q(T) = \frac{\sum_i C_{q,i} u_{q,i,\pi(i)}}{\sum_i C_{q,i}}. \tag{15}$$

The multi-target empirical mean of 14 is therefore the equal-target MacroASR difference. The identity separates recovery from destruction and requires neither model calibration nor an independence assumption.

Conditioned on target-disjoint information $Z_i = z$, the macro-compatible conditional utility is

$$\begin{aligned} \mu_t(z) &= \mathbb{E}[y_{i,t} \mid Z_i = z] \\ &= \frac{1}{Q} \sum_{q=1}^{Q} \frac{N}{N_q} \Big[\mathbb{P}(C_{q,i} = 1, A_{q,i,t} = 1, A_{q,i,T} = 0 \mid Z_i = z) \\ &\qquad - \mathbb{P}(C_{q,i} = 1, A_{q,i,t} = 0, A_{q,i,T} = 1 \mid Z_i = z) \Big]. \end{aligned} \tag{16}$$

The first weighted term is target-clean recovery mass and the second is target-clean destruction mass. The zero reference accepts a non-final action only when estimated recovery exceeds estimated destruction. Absolute proxy success cannot make this comparison because it does not encode the outcome already achieved by final.

3.2 Pointwise Utility Maximization

The ideal target-output-free reader uses observable Z_i to choose

$$\pi^*(Z_i) \in \arg\max_{t\in\{1,\dots,T\}} \mu_t(Z_i), \qquad \mu_T(Z_i) = 0. \tag{17}$$

A global checkpoint instead chooses one t for the entire population.

Proposition 2 (Pointwise dominance)**.** *The Bayes pointwise reader satisfies*

$$\mathbb{E}_i\Big[\max_t \mu_t(Z_i)\Big] \geq \max_t \mathbb{E}_i[\mu_t(Z_i)] \geq 0. \tag{18}$$

The first lower bound is the best globally fixed checkpoint and the second is final. Appendix A.2 proves the proposition, its strictness condition, and the fitted-reader approximation bound.

Let

$$t^{\text{fix}} \in \arg\max_t \mathbb{E}_i[\mu_t(Z_i)]. \tag{19}$$

The additional capacity of pointwise selection is the heterogeneity gap

$$\Delta_{\text{het}} = \mathbb{E}_i\Big[\max_t \mu_t(Z_i) - \mu_{t^{\text{fix}}}(Z_i)\Big] \geq 0. \tag{20}$$

The inequality is strict when a positive-probability set of images has a utility-maximizing action different from t^{fix} with a positive margin. This condition formalizes image-dependent transfer peaks.

A fitted reader predicts $\hat{\mu}_t(Z_i)$. If

$$\max_t |\hat{\mu}_t(Z_i) - \mu_t(Z_i)| \leq \varepsilon_i, \tag{21}$$

then its selected action $\hat{t}_i \in \arg\max_t \hat{\mu}_t(Z_i)$ obeys

$$\mu_{\pi^*(Z_i)}(Z_i) - \mu_{\hat{t}_i}(Z_i) \leq 2\varepsilon_i. \tag{22}$$

Consequently, the fitted reader exceeds the best fixed checkpoint whenever $\Delta_{\text{het}} > 2\mathbb{E}_i[\varepsilon_i]$. This result concerns conditional utility under the development estimand; transfer of the learned ordering to disjoint formal targets is evaluated empirically in Section 4.

3.3 Target-Disjoint Proxy Trajectory Representation

Let $\mathcal{M}_s = \{f_s\}$, $\mathcal{M}_p = \mathcal{P} = \{f_p\}_{p=1}^P$, $\mathcal{M}_{\text{dev}} = \mathcal{Q}_{\text{dev}}$, and $\mathcal{M}_{\text{formal}} = \mathcal{Q}_{\text{formal}}$ be the source, proxy, development-utility, and formal-target model sets. Their roles are checkpoint-disjoint,

$$\mathcal{M}_r \cap \mathcal{M}_{r'} = \varnothing \qquad \text{for every distinct } r, r' \in \{s, p, \text{dev}, \text{formal}\}. \tag{23}$$

Only $\mathcal{M}_s$, $\mathcal{P}$, and $\mathcal{Q}_{\text{dev}}$ can affect the fitted reader. Formal targets are unavailable until the returned action has been frozen. For proxy logits $f_p(x) \in \mathbb{R}^C$, the per-image clean scale is

$$s_{p,i} = \max\left\{\sqrt{\frac{1}{C}\sum_{c=1}^{C} f_{p,c}(x_i)^2}, 10^{-8}\right\}. \tag{24}$$

For action t, define the normalized adversarial margin, its final-relative change, and proxy success by

$$m_{p,i,t} = \frac{\max_{c\neq y_i} f_{p,c}(x_{i,t}) - f_{p,y_i}(x_{i,t})}{s_{p,i}}, \tag{25}$$

$$d_{p,i,t} = m_{p,i,t} - m_{p,i,T}, \tag{26}$$

$$b_{p,i,t} = \mathbf{1}\{\arg\max_c f_{p,c}(x_{i,t}) \neq y_i\}. \tag{27}$$

Let $\text{stat}(v) = (\text{mean}(v), \text{std}(v), \min(v), \max(v))$ over proxies. The complete descriptor is

$$\begin{aligned} z_{i,t} = \big[&\{m_{p,i,t}\}_{p=1}^P, \{d_{p,i,t}\}_{p=1}^P, \{b_{p,i,t}\}_{p=1}^P, \\ &\text{stat}(m_{i,t}), \text{stat}(d_{i,t}), \text{mean}(b_{i,t}), \text{std}(b_{i,t}), t/T, (t/T)^2\big] \in \mathbb{R}^{3P+12}. \end{aligned} \tag{28}$$

The canonical $P = 4$ panel yields 24 dimensions. Appendix A.3 gives the component-wise construction and access audit.

3.4 Reader Learning and Pointwise Inference

Each attack a receives an ExtraTrees regressor $h_a : \mathbb{R}^{24} \to \mathbb{R}$. Every development image contributes one record for each non-final action. The sample weight is

$$w_{i,t} = 1 + 30 \cdot \mathbf{1}\{y_{i,t} > 0\} + 2 \cdot \mathbf{1}\{y_{i,t} < 0\}, \tag{29}$$

so positive, negative, and zero utilities have weights 31, 3, and 1. The regressor uses 180 trees, minimum leaf size 16, and all features at each split. Five-fold out-of-fold prediction is grouped by image, which keeps all actions from one image on the same side of a fold. The final reader is refitted on the full development population and then frozen.

For an unseen image, CPRU predicts $\hat{u}_{i,t} = h_a(z_{i,t})$ for all $t < T$ and inserts final with exact score $\hat{u}_{i,T} = 0$. Its pointwise decision is

$$\hat{t}_i = \begin{cases} \arg\max\limits_{1 \le t < T} \hat{u}_{i,t}, & \max_{t<T} \hat{u}_{i,t} > 0, \\ T, & \text{otherwise.} \end{cases} \tag{30}$$

The returned action for one image is independent of all other deployed images. Formal-target parameters, logits, predictions, gradients, and feedback are unavailable throughout fitting and selection. For P proxies and T saved attack steps, the reader requires

$$B_{\text{proxy}} = P(T+1) \tag{31}$$

proxy forwards per image and attack, including the clean image. The canonical $P = 4$ and $T = 10$ setting uses 44 proxy forwards and no additional source-gradient step; the last two stages of Figure 1 visualize this frozen inference.

3.5 Attack Chain on the Unseen Target

After CPRU freezes the pointwise action $\hat{t}_i$, the deployed adversarial example is

$$x_i^{\text{adv}} = x_{i,\hat{t}_i}, \qquad \delta_i = x_i^{\text{adv}} - x_i, \qquad \|\delta_i\|_\infty \le \epsilon. \tag{32}$$

The following chain characterizes how this frozen perturbation affects an unseen target classifier f_q. Let $h_q^{(\ell)}(\cdot)$ be the representation at target layer ℓ. The adversarial input induces the layer-wise deviation

$$\Delta h_{q,i}^{(\ell)} = h_q^{(\ell)}(x_i^{\text{adv}}) - h_q^{(\ell)}(x_i). \tag{33}$$

Under a local first-order approximation, successive deviations satisfy

$$\Delta h_{q,i}^{(\ell+1)} \approx J_q^{(\ell)} \Delta h_{q,i}^{(\ell)}, \tag{34}$$

where $J_q^{(\ell)}$ is the local Jacobian from layer ℓ to $\ell+1$, evaluated along the clean activation path. This approximation only describes target-side propagation; it is neither a CPRU optimization objective nor a guarantee of boundary crossing.

The accumulated representation change perturbs the target logits. Define the target classification margin as

$$M_{q,i}(x) = f_{q,y_i}(x) - \max_{c \ne y_i} f_{q,c}(x). \tag{35}$$

For a target-clean image, $M_{q,i}(x_i) > 0$. A successful transferred adversarial example shifts this margin across zero,

$$M_{q,i}(x_i^{\text{adv}}) < 0, \tag{36}$$

which implies

$$\arg\max_c f_{q,c}(x_i^{\text{adv}}) \ne y_i. \tag{37}$$

With $\Delta M_{q,i} = M_{q,i}(x_i^{\text{adv}}) - M_{q,i}(x_i)$ and $\hat{y}_i = \arg\max_c f_{q,c}(x_i^{\text{adv}})$, the target-side chain is

$$\boxed{\delta_i \to \Delta h_{q,i}^{(1)} \to \cdots \to \Delta h_{q,i}^{(L)} \to \Delta M_{q,i} \to \hat{y}_i \ne y_i.} \tag{38}$$

Figure 2 connects the frozen pointwise readout to this unseen-target mechanism while preserving the zero-query boundary before evaluation.

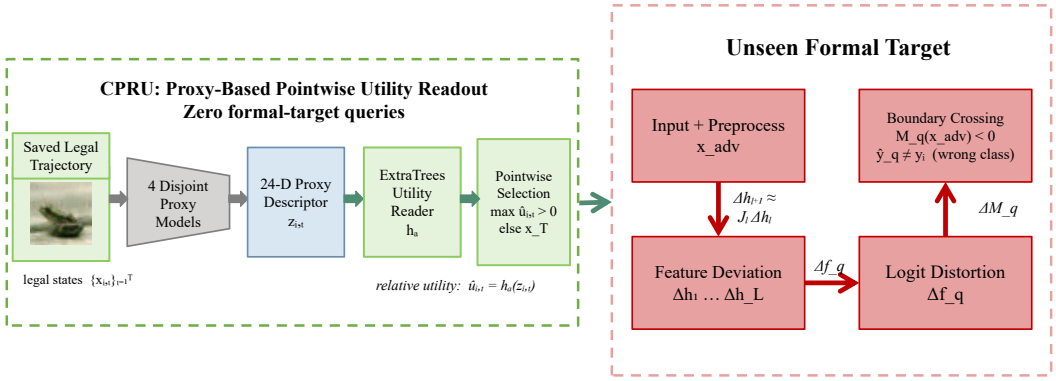


Figure 2: Frozen CPRU readout and the unseen-target attack chain. The left panel ends before any formal-target query. The right panel characterizes how the returned perturbation can propagate through target features and logits until the classification margin crosses zero.

4 Experiments

4.1 Experimental Setting

Protocol and attacks. We define protocol R as an untargeted margin-loss attack with ten iterations, a clean start, and an ℓ_∞ budget of $8/255$. All nine non-final actions and the final action are saved. We evaluate BP14, PGD-A15 (Madry et al., 2018), MDCS (Tao et al., 2026), PI-FGSM (Gao et al., 2020), and DI^2-FGSM (Xie et al., 2019). PGD-A15 uses step size $1.5\epsilon/T$. DI^2-FGSM uses diversity probability 0.5, nearest-neighbor resizing, zero padding, and step size $2/255$. Every attack update remains unchanged, and CPRU changes only the returned trajectory state.

CIFAR-10 roles and populations. The primary data set is CIFAR-10 (Krizhevsky, 2009). The source, proxy, development, and formal-target roles are checkpoint-disjoint; Appendix A.4 lists every checkpoint. One seeded permutation assigns 2,000 images to reader development and the remaining 8,000 images to a held-out pool. A fixed random seed is used to sample 1,000 evaluation images from this held-out pool. The evaluation population is therefore determined independently of CPRU scores, proxy outputs, reader predictions, and formal-target outcomes, and the same 1,000 images are used for all attacks and deployment policies. The evaluation population is fixed before CPRU inference, and formal targets are not loaded until the returned action for every evaluation image has been frozen.

ImageNet-1K roles and populations. The ImageNet-1K validation experiment uses original images and the standard 224×224 crop (Deng et al., 2009). Two development and eight held-out images per class produce development and held-out populations of 2,000 and 8,000 images, respectively. A fixed random seed is used to sample 1,000 evaluation images from the held-out population before CPRU inference and formal-target evaluation. The source is torchvision ViT-B/32. The proxies are AugReg ViT-S/16, ConvNeXt-Tiny, MaxViT-Tiny, and VGG-16-BN. Development utility uses MobileNetV3-Large, PVTv2-B1, RegNetY-4GF, ResNeXt-50, and Swin-T. Formal targets are DeiT-Small, CoAtNet-0, CaiT-XXS24, NesT-Tiny, and PiT-Ti. All checkpoints are ImageNet-pretrained and role-disjoint.

Metric and uncertainty. ASR is computed only over images classified correctly by the evaluated target before perturbation and is averaged equally over formal targets. Gains are CPRU minus final in percentage points. Confidence intervals resample image indices with replacement for 20,000 replicates and carry every target observation associated with the sampled image. This image-cluster bootstrap preserves dependence across actions and targets.

LPAA baseline. Local Perturbation Augmentation Attack (LPAA) aggregates source gradients over multiple masked, scaled perturbation subspaces and uses perturbation initialization to improve

transferability (Mi et al., 2026). We run the official LPAA procedure on the same pre-sampled 1,000-image CIFAR-10 population, source checkpoint, perturbation budget, and formal-target panel. Because LPAA changes attack generation while CPRU changes only trajectory readout, the comparison reports one standalone LPAA run against each compatible CPRU attack. LPAA is not applicable to the projected PGD-A15 update under protocol R, so no PGD-A15 comparison is reported. Appendix A.4.2 gives the complete configuration and cost accounting.

4.2 Primary CIFAR-10 Results

Table 1 shows gains of 4.368–9.455 points for every attack and 7.617 points on average; every attack-wise interval excludes zero. Non-final selection occurs for nearly every image, and recovered target-clean successes exceed destroyed final successes.

Table 1: Primary CIFAR-10 results under protocol R. Dev-fixed is one attack-specific checkpoint selected on the disjoint development population and applied uniformly to every evaluation image. ASR values are percentages; differences are percentage points. The reported 95% CI corresponds to (CPRU–Final).

Attack	Final	Dev-fixed	CPRU	(CPRU–Final)	(CPRU–Dev-fixed)	95% CI	Non-final
BP14	59.535	59.535	**66.672**	+7.138	+7.138	[5.977, 8.319]	100.0%
PGD-A15	33.876	33.876	**38.244**	+4.368	+4.368	[3.412, 5.316]	99.5%
MDCS	57.963	57.963	**66.191**	+8.228	+8.228	[7.015, 9.475]	100.0%
PI-FGSM	55.288	57.061	**64.185**	+8.897	+7.124	[7.683, 10.116]	100.0%
DI^2-FGSM	66.831	66.831	**76.286**	+9.455	+9.455	[8.312, 10.609]	99.5%
Equal-attack mean	54.699	55.053	**62.316**	**+7.617**	**+7.262**	**[6.975, 8.265]**	–

Development chooses final for four attacks and step 6 for PI-FGSM, raising the mean by only about 0.355 points. CPRU exceeds Dev-fixed by 7.262 points (95% CI [6.624, 7.906]); its gain therefore cannot be explained by one globally preferred iteration. Appendix A.5.1 reports all 25 attack–target cases; every gain is positive.

4.3 Comparison with LPAA

Table 2 compares CPRU with the standalone LPAA reference on the same pre-sampled 1,000-image population. Each CPRU entry averages the four compatible attacks for one formal target. CPRU reaches 68.334 equal-target MacroASR versus 63.281 for LPAA, a descriptive advantage of 5.053 points, and its target-wise advantage ranges from 3.118 to 8.595 points. We define complete wall-clock time as the interval from process launch to emission of the output for the fixed workload. For target j and CPRU attack a, this time is

$$T^{(a)}_{\mathrm{CPRU},j} = T^{(a)}_{\mathrm{load}} + T^{(a)}_{\mathrm{reader/decision}} + T^{(a)}_{\mathrm{source\ trajectory}} + T^{(a)}_{\mathrm{formal},j} + T^{(a)}_{\mathrm{output}}. \tag{39}$$

Appendix A.5.2 gives complete attack–target results and cost accounting.

Table 2: Target-wise (CPRU–LPAA) comparison on CIFAR-10. Final, LPAA, and CPRU average BP14, MDCS, PI-FGSM, and DI^2-FGSM for each target. The last two columns report complete single-attack wall-clock times in seconds. Values are percentages; PGD-A15 is excluded because LPAA is not defined for its projected update under protocol R.

Target	Final	LPAA	CPRU	CPRU total (s)	LPAA total (s)
DeiT-Tiny	73.969	80.550	**83.668**	25.073	468.784
CoAtNet-0	42.982	41.692	**47.859**	29.255	469.885
CaiT-XXS24	67.392	73.557	**77.072**	29.132	469.158
NesT-Tiny	57.026	64.316	**68.184**	25.828	469.247
PiT-Ti	58.150	56.289	**64.885**	25.346	468.812
Equal-target mean	59.904	63.281	**68.334**	**26.927**	**469.177**

4.4 Unseen Target and Source Transfer

Table 3 evaluates two complementary shifts. Panel (a) applies the frozen ViT-B/32-source readers to the unseen Twins-SVT-Small target while keeping the source, proxies, reader, action slate, and population fixed. All attacks improve by 6.542–14.849 points and the mean gain is 10.696 points. Panel (b) changes only the trajectory source to ConvNeXt-Small, which is distinct from the ConvNeXt proxy. All intervals again exclude zero and the mean gain is 4.548 points. Appendix A.5.4 gives target-wise results.

Table 3: Unseen-target and second-source evaluations. Panel (a) uses frozen ViT-B/32-source readers on Twins-SVT-Small, and panel (b) uses ConvNeXt-Small as the source. ASR is in percent and intervals are in percentage points.

	(a) Twins-SVT-Small target				(b) ConvNeXt-Small source			
Attack	Final	CPRU	Gain	95% CI	Final	CPRU	Gain	95% CI
BP14	60.540	70.717	+10.177	[7.867, 12.591]	51.319	53.703	+2.384	[1.358, 3.429]
PGD-A15	34.891	41.433	+6.542	[4.588, 8.524]	24.803	27.583	+2.779	[1.937, 3.626]
MDCS	58.048	71.443	+13.396	[10.870, 15.942]	49.486	53.334	+3.848	[2.799, 4.926]
PI-FGSM	52.544	67.394	+14.849	[12.253, 17.451]	53.405	56.254	+2.849	[1.830, 3.885]
DI^2-FGSM	77.051	85.566	+8.515	[6.348, 10.733]	38.113	48.994	+10.881	[9.666, 12.102]
Mean	56.615	**67.310**	**+10.696**	[9.442, 11.950]	43.425	**47.973**	**+4.548**	[3.983, 5.129]

4.5 ImageNet-1K at Native Resolution

At native 224×224 resolution, Table 4 reports positive point estimates for all five attacks and an equal-attack mean gain of 2.280 points (95% CI [1.832, 2.728]). Attack-wise intervals exclude zero for MDCS, PI-FGSM, and DI^2-FGSM but include zero for BP14 and PGD-A15. Appendix A.5.3 gives all 25 target-wise results.

Table 4: ImageNet-1K at native resolution under protocol R (ASR in %). Attack-wise intervals use image-cluster bootstrap.

Attack	Final ASR	CPRU ASR	Gain	95% CI	Non-final
BP14	56.510	56.883	+0.374	[−0.195, 0.943]	49.6%
PGD-A15	35.438	35.934	+0.496	[−0.356, 1.330]	99.6%
MDCS	55.150	56.285	+1.135	[0.247, 2.044]	100.0%
PI-FGSM	49.986	56.777	+6.791	[5.482, 8.107]	75.9%
DI^2-FGSM	54.310	56.915	+2.605	[1.826, 3.409]	54.2%
Equal-attack mean	50.279	**52.559**	**+2.280**	**[1.832, 2.728]**	–

5 Conclusion and Discussion

This paper addresses the mismatch between source-side optimization and endpoint deployment in transferable adversarial attacks. From a trajectory-readout perspective, we develop Calibrated Pointwise Relative Utility (CPRU) to exploit transferable information contained in intermediate attack states. Specifically, we show that the relative-utility objective is exactly aligned with paired target-clean ASR gain, and that pointwise selection can exploit image-dependent trajectory information beyond a globally fixed checkpoint. Experimental results across different iterative attacks, data sets, source models, and target architectures demonstrate that CPRU effectively improves endpoint deployment without modifying the underlying attack optimization or querying the formal target. Furthermore, the reader-sharing study shows that leave-one-attack-out training remains effective, revealing reusable structural cues across attack trajectories. This result motivates a fully plug-and-play reader as future work, with the goal of generalizing to new attacks without attack-specific refitting.

REFERENCES

Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pp. 321–338, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, June 2009. doi: 10.1109/CVPR.2009.5206848.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. *Patch-Wise Attack for Fooling Deep Neural Network*, volume 12373 of *Lecture Notes in Computer Science*, pp. 307–322. Springer International Publishing, 2020. doi: 10.1007/978-3-030-58604-1_19.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 09–15 Jun 2019.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Martin Gubri, Maxime Cordy, and Yves Le Traon. Going further: Flatness at the rescue of early stopping for adversarial example transferability, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 06–11 Aug 2017.

Pierre Gutierrez and Jean-Yves Gérardy. Causal inference and uplift modelling: A review of the literature. In Claire Hardgrove, Louis Dorard, Keiran Thompson, and Florian Douetteau (eds.), *Proceedings of The 3rd International Conference on Predictive Applications and APIs*, volume 67 of *Proceedings of Machine Learning Research*, pp. 1–13. PMLR, 11–12 Oct 2017.

Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, April 2009.

Qizhang Li, Yiwen Guo, Wangmeng Zuo, and Hao Chen. Towards evaluating transfer-based attacks systematically, practically, and fairly. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 41707–41726. Curran Associates, Inc., 2023. doi: 10.52202/075280-1807.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20685–20694, June 2025.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Jian-Xun Mi, Xuanhui Zhong, and Weisheng Li. Improving adversarial transferability with local perturbation augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20639–20649, June 2026.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pp. 506–519. ACM, April 2017. doi: 10.1145/3052973.3053009.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Wei Tao, Yang Dai, Jincai Huang, and Qing Tao. The power of decaying steps: Enhancing attack stability and transferability for sign-based optimizers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 42300–42309, June 2026.

Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1924–1933, June 2021.

Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7639–7648, October 2021.

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with ResNets. In *International Conference on Learning Representations*, 2020.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Bo Yang, Hengwei Zhang, Jindong Wang, Yulong Yang, Chenhao Lin, Chao Shen, and Zhengyu Zhao. Adversarial example soups: Improving transferability and stealthiness for free. *IEEE Transactions on Information Forensics and Security*, 20:1882–1894, 2025. doi: 10.1109/TIFS.2025.3536611.

Jianping Zhang, Weibin Wu, Jen-tse Huang, Yizhan Huang, Wenxuan Wang, Yuxin Su, and Michael R. Lyu. Improving adversarial transferability via neuron attribution-based attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14993–15002, June 2022.

A APPENDIX

This appendix provides the complete mathematical analysis, implementation contract, experimental specifications, disaggregated results, and controlled ablations supporting the main paper.

A.1 PROOF OF PROPOSITION 1: EXACT UTILITY ALIGNMENT

Let

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \tag{A.1}$$

be an arbitrary finite evaluation population and let f_q be one target classifier. Define

$$C_{q,i} = \mathbf{1}\{f_q(x_i) = y_i\}, \qquad A_{q,i,t} = \mathbf{1}\{f_q(x_{i,t}) \neq y_i\}, \qquad N_q = \sum_{i=1}^{N} C_{q,i} > 0. \tag{A.2}$$

For a reader π, the returned action for image i is $x_{i,\pi(i)}$. Its target-clean ASR and the final policy are

$$\mathrm{ASR}_q(\pi) = \frac{1}{N_q} \sum_{i=1}^{N} C_{q,i} A_{q,i,\pi(i)}, \tag{A.3}$$

$$\mathrm{ASR}_q(T) = \frac{1}{N_q} \sum_{i=1}^{N} C_{q,i} A_{q,i,T}. \tag{A.4}$$

The signed relative utility is

$$u_{q,i,t} = A_{q,i,t} - A_{q,i,T}, \qquad u_{q,i,t} \in \{-1, 0, 1\}, \qquad u_{q,i,T} = 0. \tag{A.5}$$

Lemma 1 (Recovery–destruction decomposition)**.** *For any $a, b \in \{0, 1\}$,*

$$a - b = \mathbf{1}\{a = 1, b = 0\} - \mathbf{1}\{a = 0, b = 1\}. \tag{A.6}$$

Proof. The four possible pairs are

$$(a, b) \in \{(0, 0), (1, 0), (0, 1), (1, 1)\}. \tag{A.7}$$

For these cases, $a - b$ equals $0, +1, -1, 0$. The right side of A.6 takes the same four values, which proves the identity.

Applying Lemma 1 with $a = A_{q,i,\pi(i)}$ and $b = A_{q,i,T}$ gives

$$\begin{aligned} u_{q,i,\pi(i)} = {} & \mathbf{1}\{A_{q,i,\pi(i)} = 1, A_{q,i,T} = 0\} \\ & - \mathbf{1}\{A_{q,i,\pi(i)} = 0, A_{q,i,T} = 1\}. \end{aligned} \tag{A.8}$$

Thus the selected action contributes $+1$ exactly when it recovers a final failure and -1 exactly when it destroys a final success.

Proof of Proposition 1. Subtracting A.4 from A.3 yields

$$\begin{aligned} \mathrm{ASR}_q(\pi) - \mathrm{ASR}_q(T) &= \frac{1}{N_q} \sum_{i=1}^{N} C_{q,i} \left(A_{q,i,\pi(i)} - A_{q,i,T} \right) \\ &= \frac{1}{N_q} \sum_{i=1}^{N} C_{q,i} u_{q,i,\pi(i)}. \end{aligned} \tag{A.9}$$

This is an exact finite-sample identity. It does not invoke asymptotics, independence, or calibration.

To expose both directions of replacement, define

$$R_q(\pi) = \sum_{i=1}^{N} C_{q,i} \mathbf{1}\{A_{q,i,\pi(i)} = 1, A_{q,i,T} = 0\}, \tag{A.10}$$

$$D_q(\pi) = \sum_{i=1}^{N} C_{q,i} \mathbf{1}\{A_{q,i,\pi(i)} = 0, A_{q,i,T} = 1\}. \tag{A.11}$$

Substituting A.8 into A.9 gives

$$\mathrm{ASR}_q(\pi) - \mathrm{ASR}_q(T) = \frac{R_q(\pi) - D_q(\pi)}{N_q}. \tag{A.12}$$

Now let $\mathcal{Q} = \{1, \ldots, Q\}$ be a finite collection of targets and define

$$\mathrm{MacroASR}(\pi) = \frac{1}{Q} \sum_{q=1}^{Q} \mathrm{ASR}_q(\pi). \tag{A.13}$$

Averaging A.9 over q and exchanging the finite sums gives

$$\begin{aligned} &\mathrm{MacroASR}(\pi) - \mathrm{MacroASR}(T) \\ &= \frac{1}{Q} \sum_{q=1}^{Q} \frac{1}{N_q} \sum_{i=1}^{N} C_{q,i} u_{q,i,\pi(i)} \\ &= \frac{1}{N} \sum_{i=1}^{N} \left[\frac{1}{Q} \sum_{q=1}^{Q} \frac{N}{N_q} C_{q,i} u_{q,i,\pi(i)} \right]. \end{aligned} \tag{A.14}$$

The bracketed term is precisely the selected-action version of 14. The factor N/N_q is therefore not an arbitrary balance weight. It follows from

$$\frac{1}{N} \frac{N}{N_q} = \frac{1}{N_q}, \tag{A.15}$$

which restores the target-clean denominator after image-wise averaging. This completes the proof.

A.2 Proof of Proposition 2: Pointwise Dominance and Approximation Error

Let Z_i collect all target-output-free information available when image i is read. For every action t, define

$$\mu_t(Z_i) = \mathbb{E}[y_{i,t} \mid Z_i], \qquad \mu_T(Z_i) = 0. \tag{A.16}$$

The Bayes pointwise reader and the best fixed checkpoint are

$$\pi^*(Z_i) \in \arg \max_{t \in \{1,\ldots,T\}} \mu_t(Z_i), \tag{A.17}$$

$$t^{\mathrm{fix}} \in \arg \max_{t \in \{1,\ldots,T\}} \mathbb{E}[\mu_t(Z_i)]. \tag{A.18}$$

Lemma 2 (Selected-action conditional expectation)**.** *For any measurable target-output-free policy* $\pi(Z_i)$*,*

$$\mathbb{E}[y_{i,\pi(Z_i)} \mid Z_i] = \mu_{\pi(Z_i)}(Z_i). \tag{A.19}$$

Proof. Because $\pi(Z_i)$ is measurable with respect to Z_i,

$$\begin{aligned} \mathbb{E}[y_{i,\pi(Z_i)} \mid Z_i] &= \sum_{t=1}^{T} \mathbf{1}\{\pi(Z_i) = t\} \mathbb{E}[y_{i,t} \mid Z_i] \\ &= \sum_{t=1}^{T} \mathbf{1}\{\pi(Z_i) = t\} \mu_t(Z_i) = \mu_{\pi(Z_i)}(Z_i). \end{aligned} \tag{A.20}$$

Proof of Proposition 2. For every realization of Z_i and every fixed action t_0,

$$\max_t \mu_t(Z_i) \geq \mu_{t_0}(Z_i). \tag{A.21}$$

Taking expectations, applying Lemma 2, and then maximizing the right side over t_0 yields

$$\begin{aligned}\mathbb{E}[y_{i,\pi^*(Z_i)}] &= \mathbb{E}\Big[\max_t \mu_t(Z_i)\Big] \\ &\geq \max_t \mathbb{E}[\mu_t(Z_i)] = \mathbb{E}[y_{i,t^{\text{fix}}}].\end{aligned} \tag{A.22}$$

Final is an admissible action with $\mu_T = 0$. Therefore

$$\mathbb{E}[y_{i,\pi^*(Z_i)}] \geq \mathbb{E}[y_{i,t^{\text{fix}}}] \geq 0. \tag{A.23}$$

The difference between pointwise and fixed capacity is

$$\begin{aligned}\Delta_{\text{het}} &= \mathbb{E}\big[\mu_{\pi^*(Z_i)}(Z_i) - \mu_{t^{\text{fix}}}(Z_i)\big] \\ &= \mathbb{E}\Big[\max_t \mu_t(Z_i) - \mu_{t^{\text{fix}}}(Z_i)\Big] \geq 0.\end{aligned} \tag{A.24}$$

Suppose a measurable set S has positive probability and there is $\gamma > 0$ such that

$$\max_t \mu_t(Z_i) - \mu_{t^{\text{fix}}}(Z_i) \geq \gamma \quad \text{for every } Z_i \in S. \tag{A.25}$$

Then

$$\Delta_{\text{het}} \geq \gamma\, \mathbb{P}(Z_i \in S) > 0. \tag{A.26}$$

Thus pointwise dominance is strict whenever a non-negligible image population prefers a different action with positive utility margin.

The fitted reader uses predictions $\hat{\mu}_t(Z_i)$ and returns

$$\hat{t}_i \in \arg\max_t \hat{\mu}_t(Z_i). \tag{A.27}$$

Assume the image-wise uniform error bound

$$\max_t |\hat{\mu}_t(Z_i) - \mu_t(Z_i)| \leq \varepsilon_i. \tag{A.28}$$

Let $t_i^* = \pi^*(Z_i)$. By optimality of $\hat{t}_i$ under $\hat{\mu}$,

$$\hat{\mu}_{t_i^*}(Z_i) - \hat{\mu}_{\hat{t}_i}(Z_i) \leq 0. \tag{A.29}$$

Adding and subtracting predicted utilities gives

$$\begin{aligned}\mu_{t_i^*}(Z_i) - \mu_{\hat{t}_i}(Z_i) &= \mu_{t_i^*} - \hat{\mu}_{t_i^*} + \hat{\mu}_{t_i^*} - \hat{\mu}_{\hat{t}_i} + \hat{\mu}_{\hat{t}_i} - \mu_{\hat{t}_i} \\ &\leq |\mu_{t_i^*} - \hat{\mu}_{t_i^*}| + |\hat{\mu}_{\hat{t}_i} - \mu_{\hat{t}_i}| \\ &\leq 2\varepsilon_i.\end{aligned} \tag{A.30}$$

Taking expectations produces

$$\mathbb{E}[\mu_{\hat{t}_i}(Z_i)] \geq \mathbb{E}[\mu_{t_i^*}(Z_i)] - 2\mathbb{E}[\varepsilon_i]. \tag{A.31}$$

Subtracting the best fixed utility and using A.24 yields

$$\mathbb{E}[\mu_{\hat{t}_i}(Z_i)] - \mathbb{E}[\mu_{t^{\text{fix}}}(Z_i)] \geq \Delta_{\text{het}} - 2\mathbb{E}[\varepsilon_i]. \tag{A.32}$$

Therefore $\Delta_{\text{het}} > 2\mathbb{E}[\varepsilon_i]$ is sufficient for the fitted pointwise reader to exceed the best fixed checkpoint in expectation. This decomposition separates available trajectory heterogeneity from utility-estimation error.

A.3 CPRU Implementation Details

A.3.1 Model Roles and Access Boundary

CPRU separates trajectory generation, trajectory observation, utility supervision, and formal evaluation into distinct model roles. Let

$$\mathcal{M}_{\text{src}}, \qquad \mathcal{P}, \qquad \mathcal{Q}_{\text{dev}}, \qquad \mathcal{Q}_{\text{formal}} \tag{A.33}$$

be the source, proxy, development-utility, and formal-target model sets, respectively. The corresponding checkpoints are role-disjoint,

$$\mathcal{M}_r \cap \mathcal{M}_{r'} = \varnothing, \qquad r \neq r', \quad r, r' \in \{\text{src}, \text{proxy}, \text{dev}, \text{formal}\}. \tag{A.34}$$

For attack a, the source model first generates a legal trajectory

$$\mathcal{A}_a(x_i) = \{x_{i,t}^{(a)}\}_{t=1}^{T}, \qquad \|x_{i,t}^{(a)} - x_i\|_\infty \leq \epsilon. \tag{A.35}$$

Once this trajectory has been generated, CPRU does not alter any trajectory state. Its readout information for image i is restricted to

$$\mathscr{I}_{i,a}^{\text{pre}} = \left\{ x_i, y_i, \mathcal{A}_a(x_i), \{f_p(x_i), f_p(x_{i,t}^{(a)})\}_{p\in\mathcal{P},\, t=1}^{T}, h_a \right\}, \tag{A.36}$$

where h_a denotes the frozen attack-specific utility reader. The selected action

$$\widehat{t}_i^{(a)} = \pi_a(\mathscr{I}_{i,a}^{\text{pre}}) \tag{A.37}$$

is therefore a function only of the saved source trajectory, the known label, target-disjoint proxy outputs, and the frozen reader.

No member of $\mathcal{Q}_{\text{formal}}$ contributes to A.36. In particular, before $\widehat{t}_i^{(a)}$ is frozen,

$$N_{\text{query}}^{\text{formal}} = N_{\text{forward}}^{\text{formal}} = N_{\text{gradient}}^{\text{formal}} = 0. \tag{A.38}$$

Formal targets are used only after the returned action has been fixed, and only for evaluating its target-clean attack success. Table A.1 summarizes this access contract.

Table A.1: Model-access contract for CPRU. "Pre-freeze" denotes information that may influence the returned trajectory action.

Operation	Source	Proxy	Development	Formal target
Trajectory gradient	Yes	No	No	No
Descriptor forward	No	Yes	No	No
Utility supervision	No	No	Training only	No
Reader fitting	No	Descriptor input	Utility label	No
Frozen-action evaluation	No	No	No	Forward only

A.3.2 Descriptor and Reader Construction

Let

$$f_p(x) = [f_{p,1}(x), \ldots, f_{p,C}(x)]^\top \in \mathbb{R}^C \tag{A.39}$$

be the logit vector of proxy p. Since different proxy architectures may produce logits with different numerical scales, CPRU normalizes each proxy response using the clean-image root-mean-square logit magnitude,

$$s_{p,i} = \max\left\{ \sqrt{\frac{1}{C}\sum_{c=1}^{C} f_{p,c}(x_i)^2}, 10^{-8} \right\}. \tag{A.40}$$

For trajectory action $x_{i,t}$, define the normalized untargeted margin

$$m_{p,i,t} = \frac{\max_{c\neq y_i} f_{p,c}(x_{i,t}) - f_{p,y_i}(x_{i,t})}{s_{p,i}}. \tag{A.41}$$

The decision studied by CPRU is not the absolute adversarial strength of $x_{i,t}$, but whether it should replace the conventional endpoint. We therefore additionally define its final-relative margin,

$$d_{p,i,t} = m_{p,i,t} - m_{p,i,T}. \tag{A.42}$$

The binary proxy-success indicator is

$$b_{p,i,t} = \mathbf{1}\left\{ \arg\max_c f_{p,c}(x_{i,t}) \neq y_i \right\}. \tag{A.43}$$

Stacking these quantities over all P proxies gives

$$\begin{aligned}\mathbf{m}_{i,t} &= [m_{1,i,t}, \ldots, m_{P,i,t}]^\top, \\ \mathbf{d}_{i,t} &= [d_{1,i,t}, \ldots, d_{P,i,t}]^\top, \\ \mathbf{b}_{i,t} &= [b_{1,i,t}, \ldots, b_{P,i,t}]^\top.\end{aligned} \tag{A.44}$$

For a proxy-response vector $\mathbf{v}$, define the aggregation operator

$$\Gamma(\mathbf{v}) = [\mathrm{Mean}(\mathbf{v}), \mathrm{Std}(\mathbf{v}), \mathrm{Min}(\mathbf{v}), \mathrm{Max}(\mathbf{v})]^\top. \tag{A.45}$$

Let

$$r_t = \frac{t}{T} \tag{A.46}$$

be the normalized trajectory position. The complete action descriptor is

$$z_{i,t} = \begin{bmatrix} \mathbf{m}_{i,t} \\ \mathbf{d}_{i,t} \\ \mathbf{b}_{i,t} \\ \Gamma(\mathbf{m}_{i,t}) \\ \Gamma(\mathbf{d}_{i,t}) \\ \mathrm{Mean}(\mathbf{b}_{i,t}) \\ \mathrm{Std}(\mathbf{b}_{i,t}) \\ r_t \\ r_t^2 \end{bmatrix}. \tag{A.47}$$

Its dimensionality is

$$\dim(z_{i,t}) = P + P + P + 4 + 4 + 1 + 1 + 2 = 3P + 12. \tag{A.48}$$

The canonical reader uses $P = 4$, and hence

$$z_{i,t} \in \mathbb{R}^{24}. \tag{A.49}$$

This decomposition exactly matches the current 24-dimensional descriptor: four normalized margins, four final-relative margins, four proxy-success indicators, eight margin summary statistics, two success statistics, and two trajectory-position variables.

For attack a, every development image contributes one training record for each non-final action. The resulting regression collection is

$$\mathcal{R}_a = \left\{ (z_{i,t}^{(a)}, y_{i,t}^{(a)}, w_{i,t}^{(a)}) : i \in \mathcal{I}_{\mathrm{dev}},\ 1 \le t < T \right\}, \tag{A.50}$$

where $y_{i,t}^{(a)}$ is the macro-compatible signed relative utility derived in Appendix A.1.

The training weight is

$$w_{i,t} = 1 + 30 \cdot \mathbf{1}\{y_{i,t} > 0\} + 2 \cdot \mathbf{1}\{y_{i,t} < 0\}, \tag{A.51}$$

or equivalently,

$$w_{i,t} = \begin{cases} 31, & y_{i,t} > 0, \\ 3, & y_{i,t} < 0, \\ 1, & y_{i,t} = 0. \end{cases} \tag{A.52}$$

Each attack uses an extremely randomized tree regressor

$$h_a : \mathbb{R}^{24} \to \mathbb{R} \tag{A.53}$$

with 180 trees, minimum leaf size 16, and all descriptor dimensions available at each split. These settings and the $31/3/1$ weighting are exactly those reported by the current implementation.

To prevent trajectory actions from the same image from appearing on both sides of a development split, cross-validation is grouped by image rather than by action. Let

$$\mathcal{I}_{\mathrm{dev}} = \bigcup_{k=1}^{K} \mathcal{I}_k, \qquad \mathcal{I}_k \cap \mathcal{I}_l = \varnothing\ (k \neq l), \qquad K = 5. \tag{A.54}$$

be the five image-level folds. If image i belongs to fold k, all of its non-final action records remain in that fold,

$$i \in \mathcal{I}_k \quad \Longrightarrow \quad (i,t) \in \mathcal{R}_{a,k}, \qquad \forall t < T. \tag{A.55}$$

Let $h_a^{(-k)}$ be the reader trained without images in $\mathcal{I}_k$. The corresponding out-of-fold prediction is

$$\widehat{u}_{i,t}^{\mathrm{OOF}} = h_a^{(-\kappa(i))}(z_{i,t}), \tag{A.56}$$

where $\kappa(i)$ is the fold containing image i. Thus, no trajectory action from image i is used to fit the reader that produces its out-of-fold prediction.

After development fitting is completed, the final attack-specific reader h_a is trained on the complete development population and frozen before formal-target evaluation. The formal target models do not participate in either A.51 or A.56.

A.3.3 Frozen Pointwise Inference

For a frozen attack reader h_a, CPRU evaluates every non-final action independently through

$$\widehat{u}_{i,t} = h_a(z_{i,t}), \qquad 1 \le t < T, \tag{A.57}$$

The conventional final action is not estimated by the regressor. Instead, its relative utility is inserted exactly as

$$\widehat{u}_{i,T} = 0. \tag{A.58}$$

The returned trajectory index is

$$\widehat{t}_i = \begin{cases} \arg\max_{1 \le t < T} \widehat{u}_{i,t}, & \max_{1 \le t < T} \widehat{u}_{i,t} > 0, \\ T, & \max_{1 \le t < T} \widehat{u}_{i,t} \le 0, \end{cases} \tag{A.59}$$

Consequently, a non-final action must satisfy two conditions: it must maximize predicted utility among all non-final actions and its predicted utility must exceed the exact zero-utility endpoint. Otherwise, CPRU returns the original endpoint $x_{i,T}$.

The inference decision is pointwise. For any two distinct images $i \neq j$,

$$\widehat{t}_i = \pi_a(\{z_{i,t}\}_{t<T}) \tag{A.60}$$

does not depend on $\{z_{j,t}\}_{t<T}$. Thus, the readout of one image does not require batch-level ranking or decisions from other evaluation images.

For each proxy, CPRU forwards the clean image once and all T trajectory actions once. Therefore, the total number of proxy forward passes per image and attack is

$$N_{\text{proxy-forward}} = P(T+1). \tag{A.61}$$

Under the canonical setting $P = 4$ and $T = 10$,

$$N_{\text{proxy-forward}} = 4 \times 11 = 44. \tag{A.62}$$

The reader itself evaluates only the $T - 1$ non-final actions,

$$N_{\text{reader}} = T - 1 = 9. \tag{A.63}$$

No additional source gradient is required after the original trajectory has been generated,

$$N_{\text{additional source gradient}} = 0. \tag{A.64}$$

Hence the additional inference cost of CPRU consists of target-disjoint proxy forward passes and tree-reader evaluation, while the underlying attack trajectory and perturbation budget remain unchanged. The current ablation record independently confirms the 44-forward cost of the four-proxy configuration.

A.4 Experimental Details

A.4.1 Data Sets and Model Checkpoints

The primary evaluation is conducted on CIFAR-10 (Krizhevsky, 2009). ViT-B/32 serves as the white-box source model. The proxy panel consists of CSWin, ConvNeXt, MaxViT, and VGG-16-BN. These checkpoints are used only to construct target-disjoint trajectory descriptors.

Utility supervision is provided by five separate development models: MobileNetV3-Large, PVTv2, RegNetY-4GF, ResNeXt-50 32×4d, and Swin-T. The primary formal-target panel consists of DeiT-Tiny, CoAtNet-0, CaiT-XXS24, NesT-Tiny, and PiT-Ti. No checkpoint is reused across source, proxy, development, and formal-target roles. These exact role assignments are those used by the primary CIFAR-10 experiment.

Two architecture-transfer settings extend the primary experiment. The unseen-target experiment retains the frozen ViT-B/32-source readers and evaluates the returned actions on Twins-SVT-Small, which is absent from reader construction and from the primary formal-target panel. The source-transfer experiment replaces ViT-B/32 with ConvNeXt-Small while retaining the remaining evaluation-role contract.

ImageNet-1K evaluation uses original validation images at the standard 224×224 input resolution (Deng et al., 2009). The source is an ImageNet-pretrained ViT-B/32. Four ImageNet-pretrained proxy checkpoints and five role-disjoint development utility models are used to construct the corresponding readers. Formal evaluation is conducted on DeiT-Small, CoAtNet-0, CaiT-XXS24, NesT-Tiny, and PiT-Ti.

All experiments preserve the same logical separation,

$$\text{Source} \perp_{\text{role}} \text{Proxy} \perp_{\text{role}} \text{Development} \perp_{\text{role}} \text{Formal Target}, \tag{A.65}$$

where $\perp_{\text{role}}$ denotes checkpoint-role disjointness rather than statistical independence. Table A.2 lists every concrete checkpoint family assigned to these roles in each setting.

Table A.2: Concrete model-role configuration in every experimental setting. Names in the ImageNet-1K block identify the pretrained checkpoint families used in the archived configuration.

Setting	Role	Model or checkpoint panel
Primary CIFAR-10	Source	ViT-B/32
	Proxies	CSWin, ConvNeXt, MaxViT, VGG-16-BN
	Development	MobileNetV3-Large, PVTv2, RegNetY-4GF, ResNeXt-50 32×4d, Swin-T
	Formal targets	DeiT-Tiny, CoAtNet-0, CaiT-XXS24, NesT-Tiny, PiT-Ti
Unseen target CIFAR-10	Source	ViT-B/32
	Proxies	CSWin, ConvNeXt, MaxViT, VGG-16-BN
	Development	MobileNetV3-Large, PVTv2, RegNetY-4GF, ResNeXt-50 32×4d, Swin-T
	Formal target	Twins-SVT-Small
Changed source CIFAR-10	Source	ConvNeXt-Small
	Proxies	CSWin, ConvNeXt, MaxViT, VGG-16-BN
	Development	MobileNetV3-Large, PVTv2, RegNetY-4GF, ResNeXt-50 32×4d, Swin-T
	Formal targets	DeiT-Tiny, CoAtNet-0, CaiT-XXS24, NesT-Tiny, PiT-Ti
ImageNet-1K 224×224	Source	torchvision ViT-B/32 (ImageNet-1K V1)
	Proxies	AugReg ViT-S/16, ConvNeXt-Tiny, MaxViT-Tiny, VGG-16-BN
	Development	MobileNetV3-Large, PVTv2-B1, RegNetY-4GF, ResNeXt-50 32×4d, Swin-Tiny
	Formal targets	DeiT-Small, CoAtNet-0, CaiT-XXS24, NesT-Tiny, PiT-Ti

A.4.2 ATTACK CONFIGURATIONS

All attacks follow the common protocol R. The adversarial trajectory starts from the clean input, uses the untargeted margin objective, runs for

$$T = 10 \tag{A.66}$$

iterations, and satisfies the ℓ_∞ perturbation constraint

$$\|x_{i,t} - x_i\|_\infty \le \epsilon, \qquad \epsilon = \frac{8}{255}. \tag{A.67}$$

All ten legal trajectory states are retained,

$$\mathcal{A}(x_i) = \{x_{i,1}, \dots, x_{i,9}, x_{i,10}\}, \tag{A.68}$$

where $t = 10$ is the conventional final action and actions $1, \dots, 9$ constitute the non-final readout slate.

The five evaluated attacks preserve their original update rules. BP14 uses the frozen adaptive-power momentum trajectory recorded by the primary implementation. PGD-A15 uses the projected margin update with step size

$$\alpha_{\mathrm{PGD}} = \frac{1.5\epsilon}{T}. \tag{A.69}$$

MDCS follows its published coordinate-wise decay strategy (Tao et al., 2026), while PI-FGSM redistributes projected overflow over neighboring pixels according to its patch-wise update rule (Gao et al., 2020).

DI^2-FGSM applies stochastic input diversity before gradient computation (Xie et al., 2019). Its diversity probability and step size are

$$p_{\mathrm{div}} = 0.5, \qquad \alpha_{\mathrm{DI}^2} = \frac{2}{255}. \tag{A.70}$$

For CIFAR-10, the transformed input is resized to a randomly selected intermediate spatial scale between 90% and 100% of the original size using nearest-neighbor interpolation and is then zero-padded back to 32×32. The stochastic transform is made reproducible by deriving each random draw deterministically from the immutable image index and attack iteration. These attack-specific settings reproduce the current experimental protocol.

LPAA comparison protocol. Local Perturbation Augmentation Attack (LPAA) constructs multiple masked perturbation subspaces at each outer iteration, scales the retained perturbation by an augmentation coefficient, averages the resulting source gradients, and initializes the attack with its published perturbation-initialization procedure (Mi et al., 2026). The comparison changes only the generation rule: it uses the same ViT-B/32 source, pre-sampled 1,000-image CIFAR-10 evaluation population, five formal targets, untargeted objective, $T = 10$, and $\epsilon = 8/255$ as protocol R.

The dimensionless LPAA settings follow the official configuration: momentum decay $\mu = 1$, $N = 20$ local subspaces per outer step, masking ratio $p = 0.95$, augmentation coefficient $\beta = 35$, initialization projection factor $\eta = 3$, and $T_{\mathrm{LPAPI}} = 5$ initialization steps. To retain the shared perturbation budget, the step sizes scale with ϵ,

$$\alpha_{\mathrm{LPAA}} = \frac{\epsilon}{T} = \frac{0.8}{255}, \qquad \alpha_{\mathrm{LPAPI}} = \frac{2\epsilon}{T} = \frac{1.6}{255}. \tag{A.71}$$

All other initialization and projection operations follow the released algorithm. The result is one standalone LPAA adversarial set, evaluated on the same five targets and repeated in the four row-wise comparisons only to make each (CPRU–LPAA) difference explicit. LPAA is not reported for PGD-A15 because its momentum-based local-subspace update is not defined for the projected PGD-A15 rule used by protocol R; inserting it would create an unprespecified hybrid rather than reproduce LPAA.

For the 1,000-image workload, each CPRU attack retains the $10{,}000$ source-gradient evaluations used to generate its ten-step trajectory and adds $44{,}000$ proxy forwards. The instrumented LPAA

execution records 305,000 source-gradient evaluations, including initialization and local-subspace aggregation. The corresponding operation accounting is

$$\begin{aligned} B_{\mathrm{CPRU}} &= 10{,}000 \text{ source gradients} + 44{,}000 \text{ proxy forwards}, \\ B_{\mathrm{LPAA}} &= 305{,}000 \text{ source gradients}. \end{aligned} \tag{A.72}$$

Recorded CPRU wall-clock times and the complete operation counts appear in Table A.3. Wall-clock values are reported for one attack and one formal target, while the gradient and forward counts provide the hardware-independent workload comparison. Because no paired LPAA bootstrap record is available, (CPRU–LPAA) differences are descriptive rather than accompanied by confidence intervals.

Table A.3: Complete (CPRU–LPAA) computation record for the fixed 1,000-image workload. Each row reports the complete single-attack wall-clock time for one formal target. LPAA/CPRU is the ratio of the recorded wall-clock times.

Target	CPRU (s)	LPAA (s)	LPAA/CPRU	CPRU grad.	Proxy fwd.	LPAA grad.
DeiT-Tiny	25.073	468.784	**18.70**×	10,000	44,000	305,000
CoAtNet-0	29.255	469.885	**16.06**×	10,000	44,000	305,000
CaiT-XXS24	29.132	469.158	**16.10**×	10,000	44,000	305,000
NesT-Tiny	25.828	469.247	**18.17**×	10,000	44,000	305,000
PiT-Ti	25.346	468.812	**18.50**×	10,000	44,000	305,000
Five-target mean	**26.927**	**469.177**	**17.42**×	**10,000**	**44,000**	**305,000**

A.4.3 Statistical Inference

The primary metric is target-clean attack success rate. For attack a, target q, and deployment policy π, define

$$C_{q,i} = \mathbf{1}\{f_q(x_i) = y_i\}, \tag{A.73}$$

and

$$A^{(a)}_{q,i,\pi} = \mathbf{1}\Big\{f_q(x^{(a)}_{i,\pi(i)}) \neq y_i\Big\}. \tag{A.74}$$

The corresponding target-clean ASR is

$$\mathrm{ASR}_{q,a}(\pi) = \frac{\sum_{i=1}^{N} C_{q,i} A^{(a)}_{q,i,\pi}}{\sum_{i=1}^{N} C_{q,i}}. \tag{A.75}$$

For a formal-target panel

$$\mathcal{Q}_{\mathrm{formal}} = \{q_1, \ldots, q_Q\}, \tag{A.76}$$

we report the equal-target macro average

$$\mathrm{MacroASR}_a(\pi) = \frac{1}{Q} \sum_{q=1}^{Q} \mathrm{ASR}_{q,a}(\pi). \tag{A.77}$$

The paired CPRU gain for attack a is

$$\Delta_a = \mathrm{MacroASR}_a(\mathrm{CPRU}) - \mathrm{MacroASR}_a(T). \tag{A.78}$$

All gains are reported in percentage points.

Confidence intervals are obtained using image-cluster bootstrap resampling. Let

$$I^{(b)}_1, \ldots, I^{(b)}_N \tag{A.79}$$

be the image indices sampled with replacement for bootstrap replicate b, where

$$b = 1, \ldots, B, \qquad B = 20{,}000. \tag{A.80}$$

All target observations belonging to the same sampled image are carried together. For target q, the bootstrap ASR is recomputed as

$$\widehat{\mathrm{ASR}}^{(b)}_{q,a}(\pi) = \frac{\sum_{r=1}^{N} C_{q,I^{(b)}_r} A^{(a)}_{q,I^{(b)}_r,\pi}}{\sum_{r=1}^{N} C_{q,I^{(b)}_r}}. \tag{A.81}$$

The target-clean denominator is therefore re-estimated in every replicate rather than being held fixed. The corresponding bootstrap macro-average is

$$\widehat{\mathrm{MacroASR}}_a^{(b)}(\pi) = \frac{1}{Q}\sum_{q=1}^{Q} \widehat{\mathrm{ASR}}_{q,a}^{(b)}(\pi), \tag{A.82}$$

and the paired gain is

$$\widehat{\Delta}_a^{(b)} = \widehat{\mathrm{MacroASR}}_a^{(b)}(\mathrm{CPRU}) - \widehat{\mathrm{MacroASR}}_a^{(b)}(T). \tag{A.83}$$

The reported two-sided 95% percentile interval is

$$\mathrm{CI}_{0.95}(\Delta_a) = \left[Q_{0.025}(\{\widehat{\Delta}_a^{(b)}\}_{b=1}^{B}), Q_{0.975}(\{\widehat{\Delta}_a^{(b)}\}_{b=1}^{B})\right]. \tag{A.84}$$

where $Q_\tau(\cdot)$ denotes the empirical τ-quantile.

When a joint attack-level bootstrap record is available, the equal-attack gain is computed by

$$\overline{\Delta}^{(b)} = \frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} \widehat{\Delta}_a^{(b)}. \tag{A.85}$$

A.5 Complete Experimental Results

A.5.1 Primary CIFAR-10 Results

The 25 cases in Table A.4 are the Cartesian product of five attacks and five formal targets. CPRU improves every attack–target pair, with gains ranging from 1.651 points for PI-FGSM on CoAtNet-0 to 13.158 points for PI-FGSM on NesT-Tiny. Thus, the positive equal-target results in the main paper are not produced by one favorable target: every attack improves on all five formal targets. The variation in gain magnitude also reinforces the pointwise motivation, because the amount of recoverable utility depends on both the generated trajectory and the evaluated architecture.

Table A.4: Complete primary CIFAR-10 results. Values are target-clean ASR percentages and percentage-point gains.

Attack	Target	Final	CPRU	Gain
BP14	DeiT-Tiny	76.744	84.144	+7.400
BP14	CoAtNet-0	41.692	45.820	+4.128
BP14	CaiT-XXS24	69.465	76.495	+7.030
BP14	NesT-Tiny	55.474	66.211	+10.737
BP14	PiT-Ti	54.298	60.692	+6.394
PGD-A15	DeiT-Tiny	47.252	52.643	+5.391
PGD-A15	CoAtNet-0	18.679	20.743	+2.064
PGD-A15	CaiT-XXS24	36.097	42.288	+6.191
PGD-A15	NesT-Tiny	38.947	43.158	+4.211
PGD-A15	PiT-Ti	28.407	32.390	+3.983
MDCS	DeiT-Tiny	74.101	83.510	+9.408
MDCS	CoAtNet-0	39.835	45.201	+5.366
MDCS	CaiT-XXS24	67.576	76.705	+9.129
MDCS	NesT-Tiny	55.053	65.684	+10.632
MDCS	PiT-Ti	53.249	59.853	+6.604
PI-FGSM	DeiT-Tiny	66.913	79.598	+12.685
PI-FGSM	CoAtNet-0	44.169	45.820	+1.651
PI-FGSM	CaiT-XXS24	60.965	72.403	+11.438
PI-FGSM	NesT-Tiny	46.947	60.105	+13.158
PI-FGSM	PiT-Ti	57.442	62.998	+5.556
DI^2-FGSM	DeiT-Tiny	78.118	87.421	+9.302
DI^2-FGSM	CoAtNet-0	46.233	54.592	+8.359
DI^2-FGSM	CaiT-XXS24	71.563	82.686	+11.123
DI^2-FGSM	NesT-Tiny	70.632	80.737	+10.105
DI^2-FGSM	PiT-Ti	67.610	75.996	+8.386

A.5.2 LPAA COMPARISON

Table A.5 disaggregates the (CPRU–LPAA) comparison over all five formal targets for the four compatible CPRU attacks. CPRU exceeds LPAA in 17 of 20 attack–target comparisons. The three reversals occur for PI-FGSM on DeiT-Tiny, CaiT-XXS24, and NesT-Tiny; PI-FGSM still exceeds LPAA on CoAtNet-0 and PiT-Ti and has a positive equal-target difference of 0.904 points. The LPAA target values repeat across attack blocks because they come from one standalone LPAA output set. Differences are computed from the archived unrounded values, so a displayed difference can differ by 0.001 from subtraction of the rounded cells.

Table A.5: Complete target-wise (CPRU–LPAA) comparison on the fixed 1,000-image CIFAR-10 evaluation population. ASR values are percentages and differences are percentage points. Bold marks the higher of CPRU and LPAA within each row; positive (CPRU–LPAA) differences are also bold.

CPRU attack	Target	Final	CPRU	(CPRU–Final)	LPAA	(CPRU–LPAA)
BP14	DeiT-Tiny	76.744	**84.144**	+7.400	80.550	**+3.594**
BP14	CoAtNet-0	41.692	**45.820**	+4.128	41.692	**+4.128**
BP14	CaiT-XXS24	69.465	**76.495**	+7.030	73.557	**+2.938**
BP14	NesT-Tiny	55.474	**66.211**	+10.737	64.316	**+1.895**
BP14	PiT-Ti	54.298	**60.692**	+6.394	56.289	**+4.403**
MDCS	DeiT-Tiny	74.101	**83.510**	+9.408	80.550	**+2.960**
MDCS	CoAtNet-0	39.835	**45.201**	+5.366	41.692	**+3.509**
MDCS	CaiT-XXS24	67.576	**76.705**	+9.129	73.557	**+3.148**
MDCS	NesT-Tiny	55.053	**65.684**	+10.632	64.316	**+1.368**
MDCS	PiT-Ti	53.249	**59.853**	+6.604	56.289	**+3.564**
PI-FGSM	DeiT-Tiny	66.913	79.598	+12.685	**80.550**	−0.951
PI-FGSM	CoAtNet-0	44.169	**45.820**	+1.651	41.692	**+4.128**
PI-FGSM	CaiT-XXS24	60.965	72.403	+11.438	**73.557**	−1.154
PI-FGSM	NesT-Tiny	46.947	60.105	+13.158	**64.316**	−4.211
PI-FGSM	PiT-Ti	57.442	**62.998**	+5.556	56.289	**+6.709**
DI^2-FGSM	DeiT-Tiny	78.118	**87.421**	+9.302	80.550	**+6.871**
DI^2-FGSM	CoAtNet-0	46.233	**54.592**	+8.359	41.692	**+12.900**
DI^2-FGSM	CaiT-XXS24	71.563	**82.686**	+11.123	73.557	**+9.129**
DI^2-FGSM	NesT-Tiny	70.632	**80.737**	+10.105	64.316	**+16.421**
DI^2-FGSM	PiT-Ti	67.610	**75.996**	+8.386	56.289	**+19.706**

A.5.3 IMAGENET-1K RESULTS

Table A.6 uses the frozen pointwise result for each attack on the same pre-sampled evaluation population. Twenty of the 25 attack–target gains are positive, four are negative, and one is exactly zero. The negative changes are limited to 0.618 points in magnitude, whereas the largest positive change is 9.563 points for PI-FGSM on NesT-Tiny. PI-FGSM and DI^2-FGSM improve on all five targets, while the other attacks exhibit isolated target-specific reversals. This disaggregation supports the positive attack-level means summarized in Section 4.5 without claiming uniform improvement for every ImageNet attack–target pair.

The attack-level averages clarify the same pattern. BP14 and PGD-A15 improve by 0.374 and 0.496 points, respectively, but their intervals include zero; MDCS, PI-FGSM, and DI^2-FGSM improve by 1.135, 6.791, and 2.605 points with intervals excluding zero. The 2.280-point descriptive mean therefore combines strong gains for several attacks with near-zero average changes for others, rather than implying a uniform effect size across attack families.

Table A.6: Complete ImageNet-1K 224×224 results. Values are target-clean ASR percentages and percentage-point gains.

Attack	Target	Final	CPRU	Gain
BP14	DeiT-Small	66.348	67.064	+0.716
BP14	CoAtNet-0	42.398	41.780	−0.618
BP14	CaiT-XXS24	65.261	66.253	+0.993
BP14	NesT-Tiny	41.677	41.558	−0.118
BP14	PiT-Ti	66.866	67.761	+0.896
PGD-A15	DeiT-Small	42.601	43.317	+0.716
PGD-A15	CoAtNet-0	20.766	21.137	+0.371
PGD-A15	CaiT-XXS24	42.308	41.811	−0.496
PGD-A15	NesT-Tiny	24.203	26.092	+1.889
PGD-A15	PiT-Ti	47.313	47.313	0.000
MDCS	DeiT-Small	65.155	67.184	+2.029
MDCS	CoAtNet-0	40.667	42.027	+1.360
MDCS	CaiT-XXS24	62.779	63.151	+0.372
MDCS	NesT-Tiny	40.732	43.093	+2.361
MDCS	PiT-Ti	66.418	65.970	−0.448
PI-FGSM	DeiT-Small	59.666	65.513	+5.847
PI-FGSM	CoAtNet-0	38.072	44.870	+6.799
PI-FGSM	CaiT-XXS24	55.707	64.020	+8.313
PI-FGSM	NesT-Tiny	33.648	43.211	+9.563
PI-FGSM	PiT-Ti	62.836	66.269	+3.433
DI^2-FGSM	DeiT-Small	63.842	67.303	+3.461
DI^2-FGSM	CoAtNet-0	38.443	40.791	+2.349
DI^2-FGSM	CaiT-XXS24	60.546	63.524	+2.978
DI^2-FGSM	NesT-Tiny	39.315	41.913	+2.597
DI^2-FGSM	PiT-Ti	69.403	71.045	+1.642

A.5.4 Unseen-Target and Source-Transfer Results

Table A.7 records the unseen Twins-SVT-Small target. All five frozen ViT-B/32-source readers improve on this target, with gains from 6.542 to 14.849 points; every attack-wise interval and the 10.696-point mean interval exclude zero. This result tests target-architecture transfer because Twins-SVT-Small is absent from both reader development and the five-target formal panel.

Table A.7: Frozen-reader results on the unseen Twins-SVT-Small target.

Attack	Final	CPRU	Gain	95% CI
BP14	60.540	70.717	+10.177	[7.867, 12.591]
PGD-A15	34.891	41.433	+6.542	[4.588, 8.524]
MDCS	58.048	71.443	+13.396	[10.870, 15.942]
PI-FGSM	52.544	67.394	+14.849	[12.253, 17.451]
DI^2-FGSM	77.051	85.566	+8.515	[6.348, 10.733]
Mean	56.615	67.310	+10.696	[9.442, 11.950]

Table A.8 gives the complete ConvNeXt-Small-source disaggregation. Twenty-one of the 25 pair-wise gains are positive, and every attack retains a positive equal-target mean in the main source-transfer summary. The four negative entries are confined to one PGD-A15 pair and three PI-FGSM pairs, whereas DI^2-FGSM improves on every target by 7.756–13.209 points. Changing the source therefore alters the detailed gain pattern, but does not remove the aggregate benefit of frozen point-wise readout.

Table A.8: Complete ConvNeXt-Small-source CIFAR-10 results. Values are target-clean ASR percentages and percentage-point gains.

Attack	Target	Final	CPRU	Gain
BP14	DeiT-Tiny	43.268	44.707	+1.439
BP14	CoAtNet-0	41.982	45.250	+3.269
BP14	CaiT-XXS24	48.663	52.984	+4.321
BP14	NesT-Tiny	83.075	85.449	+2.374
BP14	PiT-Ti	39.607	40.124	+0.517
PGD-A15	DeiT-Tiny	12.847	13.669	+0.822
PGD-A15	CoAtNet-0	15.526	18.284	+2.758
PGD-A15	CaiT-XXS24	15.226	16.049	+0.823
PGD-A15	NesT-Tiny	68.215	78.122	+9.907
PGD-A15	PiT-Ti	12.203	11.789	−0.414
MDCS	DeiT-Tiny	41.624	43.885	+2.261
MDCS	CoAtNet-0	41.062	44.740	+3.677
MDCS	CaiT-XXS24	45.679	51.337	+5.658
MDCS	NesT-Tiny	82.250	86.171	+3.922
MDCS	PiT-Ti	36.815	40.538	+3.723
PI-FGSM	DeiT-Tiny	50.462	50.154	−0.308
PI-FGSM	CoAtNet-0	50.664	49.234	−1.430
PI-FGSM	CaiT-XXS24	53.292	56.893	+3.601
PI-FGSM	NesT-Tiny	55.934	68.937	+13.003
PI-FGSM	PiT-Ti	56.670	56.050	−0.620
DI^2-FGSM	DeiT-Tiny	27.338	36.177	+8.839
DI^2-FGSM	CoAtNet-0	28.907	41.573	+12.666
DI^2-FGSM	CaiT-XXS24	32.716	44.650	+11.934
DI^2-FGSM	NesT-Tiny	75.026	88.235	+13.209
DI^2-FGSM	PiT-Ti	26.577	34.333	+7.756

A.6 Controlled Ablation Studies

A.6.1 Cross-Setting Readout Behavior

Figure A.1 compares attack-wise gains across the four principal evaluations. The effect remains positive in equal-attack mean under an unseen target, a changed source, and ImageNet-1K, but its magnitude is setting dependent. PI-FGSM is strongest on the unseen Twins-SVT-Small target and on ImageNet-1K, whereas DI^2-FGSM is strongest after replacing the source with ConvNeXt-Small. BP14 and PGD-A15 yield smaller gains on ImageNet-1K and their intervals include zero, while the remaining three intervals exclude zero. The changing attack ordering is inconsistent with a rule that simply applies one common early step; instead, the reader responds to trajectory-specific proxy evidence whose usefulness changes with the source, target, and data set.

The equal-attack gains provide a complementary scale comparison: 7.617 points in the primary CIFAR-10 setting, 10.696 on the unseen target, 4.548 after changing the source, and a descriptive 2.280 on ImageNet-1K. These values should not be read as a controlled ranking because the evaluation populations and, for ImageNet-1K, the model panel also change. They do show that positive readout value persists under three distinct shifts, while the disaggregated intervals identify where the evidence is strong and where it remains uncertain. The reader is therefore robust across the tested settings without being claimed invariant to architecture or data-set changes.

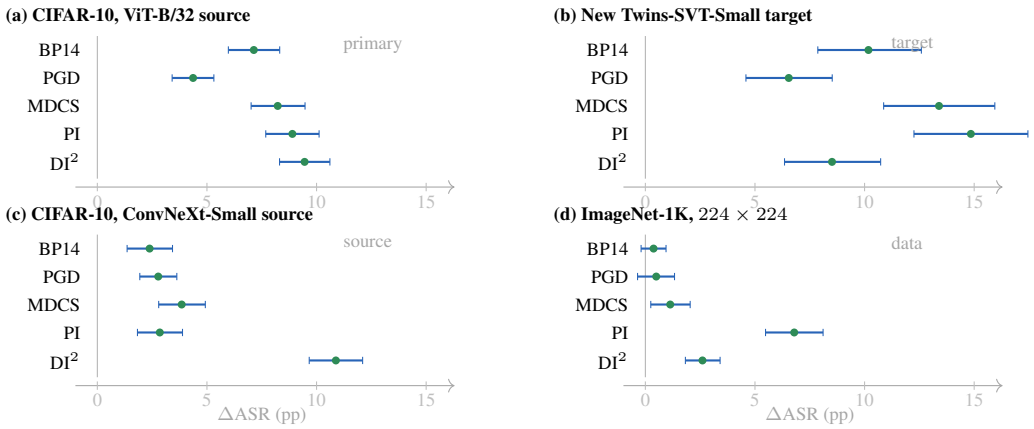


Figure A.1: Attack-wise CPRU gains and image-cluster bootstrap intervals across the four principal evaluations. The vertical line marks zero gain.

A.6.2 Effect of Proxy Count

Table A.9 varies only the number of proxies. The gain rises from 4.989 points with one proxy to 6.261 with two and 6.694 with four, after which the increments become smaller. Four proxies retain 91.8% of the nine-proxy gain while reducing the forward count from 99 to 44, a 55.6% reduction. The non-final rate stays between 98.1% and 99.4%, so the performance change is driven mainly by which non-final action is selected rather than whether CPRU leaves final.

Table A.10 reports the corresponding absolute ASR for every attack–target pair and every proxy count. Table A.11 then reports a separately frozen, synchronized four-versus-nine replay. Its absolute gains therefore need not equal the independently recorded sweep in Table A.9; the paired nine-minus-four comparison is computed only within the strict replay. The replay gives gains of 6.698 and 7.260 points and a paired difference of 0.562 points (95% CI [0.262, 0.867]). The positive interval confirms that additional proxies can still help, while the smaller absolute increment supports four proxies as the canonical accuracy–cost operating point rather than a claim that four is statistically indistinguishable from nine. Table A.12 disaggregates this replay into the same 25 cases.

Table A.9: Proxy-count ablation on identical trajectories, labels, reader hyperparameters, and pre-sampled evaluation images.

Proxies	Final	CPRU	Gain [95% CI]	Non-final	Forwards
1	54.829	59.818	4.989 [4.423, 5.536]	99.4%	11
2	54.829	61.090	6.261 [5.702, 6.855]	99.0%	22
4	54.829	61.523	6.694 [6.107, 7.233]	98.8%	44
6	54.829	61.766	6.937 [6.363, 7.504]	98.7%	66
9	54.829	62.122	7.293 [6.707, 7.836]	98.1%	99

Table A.10: Target-level absolute ASR for the proxy-count sweep. Final is the shared endpoint baseline, and each P column reports CPRU with that many proxies.

Attack	Target	Final	P=1	P=2	P=4	P=6	P=9
BP14	DeiT-Tiny	76.850	80.867	81.818	82.030	82.664	83.721
BP14	CoAtNet-0	41.486	44.169	45.201	45.511	46.027	45.820
BP14	CaiT-XXS24	69.360	74.816	75.131	75.341	75.761	76.495
BP14	NesT-Tiny	55.789	61.158	63.158	63.158	62.842	63.158
BP14	PiT-Ti	54.403	58.910	59.853	60.273	60.797	61.216
PGD-A15	DeiT-Tiny	47.252	51.691	52.431	52.748	53.805	53.911
PGD-A15	CoAtNet-0	18.782	18.576	20.433	20.640	20.743	20.846
PGD-A15	CaiT-XXS24	36.097	40.818	41.763	41.763	41.343	43.022
PGD-A15	NesT-Tiny	38.947	41.684	43.053	43.158	43.895	44.316
PGD-A15	PiT-Ti	28.512	30.503	31.447	31.551	31.866	31.971
MDCS	DeiT-Tiny	74.207	80.444	82.664	82.135	82.452	82.558
MDCS	CoAtNet-0	39.732	43.447	42.415	43.653	44.066	44.582
MDCS	CaiT-XXS24	67.786	74.292	74.292	74.816	74.397	74.711
MDCS	NesT-Tiny	54.842	61.263	62.632	63.263	63.368	63.053
MDCS	PiT-Ti	53.249	57.338	59.015	59.853	58.176	59.015
PI-FGSM	DeiT-Tiny	67.230	75.581	78.118	79.387	80.233	79.915
PI-FGSM	CoAtNet-0	44.376	45.408	45.717	45.820	45.408	45.924
PI-FGSM	CaiT-XXS24	61.805	68.625	70.829	71.144	71.668	71.354
PI-FGSM	NesT-Tiny	47.579	55.684	58.526	60.211	60.526	60.526
PI-FGSM	PiT-Ti	56.604	60.377	61.216	62.264	62.264	62.893
DI^2-FGSM	DeiT-Tiny	78.330	85.624	87.315	87.209	87.738	88.055
DI^2-FGSM	CoAtNet-0	46.440	52.322	54.902	53.870	54.489	55.624
DI^2-FGSM	CaiT-XXS24	71.773	79.748	80.693	81.952	82.581	82.791
DI^2-FGSM	NesT-Tiny	71.053	78.316	79.263	80.526	80.316	80.526
DI^2-FGSM	PiT-Ti	68.239	73.795	75.367	75.786	76.730	77.044

Table A.11: Strict paired four-versus-nine-proxy replay on the same locked trajectories, labels, reader hyperparameters, and pre-sampled evaluation images.

Panel	Final	CPRU	Gain	95% CI
Four proxies	54.829	61.526	6.698	[6.138, 7.274]
Nine proxies	54.829	62.089	7.260	[6.690, 7.842]
Nine minus four	–	+0.562	+0.562	[0.262, 0.867]

Table A.12: Target-level absolute ASR in the strict paired four-versus-nine-proxy replay.

Attack	Target	Final	Four proxies	Nine proxies
BP14	DeiT-Tiny	76.850	82.241	83.510
BP14	CoAtNet-0	41.486	45.511	45.408
BP14	CaiT-XXS24	69.360	75.131	76.076
BP14	NesT-Tiny	55.789	63.263	62.947
BP14	PiT-Ti	54.403	60.168	61.111
PGD-A15	DeiT-Tiny	47.252	53.066	53.805
PGD-A15	CoAtNet-0	18.782	20.846	21.053
PGD-A15	CaiT-XXS24	36.097	41.658	43.022
PGD-A15	NesT-Tiny	38.947	43.158	44.421
PGD-A15	PiT-Ti	28.512	31.656	31.971
MDCS	DeiT-Tiny	74.207	82.135	82.875
MDCS	CoAtNet-0	39.732	43.963	44.479
MDCS	CaiT-XXS24	67.786	74.816	74.921
MDCS	NesT-Tiny	54.842	63.158	62.947
MDCS	PiT-Ti	53.249	59.434	59.119
PI-FGSM	DeiT-Tiny	67.230	79.704	79.915
PI-FGSM	CoAtNet-0	44.376	45.614	46.130
PI-FGSM	CaiT-XXS24	61.805	71.354	71.773
PI-FGSM	NesT-Tiny	47.579	59.158	60.316
PI-FGSM	PiT-Ti	56.604	62.264	62.998
DI^2-FGSM	DeiT-Tiny	78.330	87.209	88.055
DI^2-FGSM	CoAtNet-0	46.440	54.180	55.315
DI^2-FGSM	CaiT-XXS24	71.773	81.742	82.581
DI^2-FGSM	NesT-Tiny	71.053	80.842	80.632
DI^2-FGSM	PiT-Ti	68.239	75.891	76.834

A.6.3 Effect of Trajectory Representation

Table A.13 holds the action set and reader family fixed while changing the descriptor or supervision. Removing final-relative features reduces the gain from 6.694 to 3.581 points, the largest drop among the tested variants. By contrast, aggregate-only and mean-only representations remain within 0.022 and 0.210 points of the full representation, respectively. Much of the usable signal is therefore captured by compact cross-proxy summaries, but those summaries are most informative when expressed relative to the final action.

Replacing signed regression with a binary positive-utility label reduces the gain by 0.181 points. This comparison isolates the value of predicting utility magnitude: two actions can both be beneficial relative to final, yet the signed target permits the reader to distinguish the larger improvement. Together, the variants identify relative-to-final modeling as the essential component and signed magnitude supervision as a smaller but consistent refinement. Table A.14 gives the absolute target-level ASR underlying every reported variant.

Table A.13: Trajectory-representation and supervision ablation. Gain and change are equal-attack percentage points relative to final.

Variant	Final	CPRU	Gain	Change
Full signed representation	54.829	61.523	6.694	0.000
No relative-to-final features	54.829	58.410	3.581	−3.113
Aggregate features only	54.829	61.501	6.672	−0.022
Mean features only	54.829	61.313	6.484	−0.210
Binary positive-utility label	54.829	61.342	6.513	−0.181

Table A.14: Target-level absolute ASR for the trajectory-representation and supervision ablation. All columns use the same endpoint baseline and evaluation cases.

Attack	Target	Final	Full	No-relative	Aggregate	Mean	Binary
BP14	DeiT-Tiny	76.850	82.030	81.818	81.501	82.135	82.770
BP14	CoAtNet-0	41.486	45.511	44.169	46.336	44.685	45.408
BP14	CaiT-XXS24	69.360	75.341	72.928	75.341	75.131	75.761
BP14	NesT-Tiny	55.789	63.158	62.947	63.684	64.000	64.105
BP14	PiT-Ti	54.403	60.273	58.910	59.644	59.434	59.748
PGD-A15	DeiT-Tiny	47.252	52.748	50.634	52.748	52.114	51.586
PGD-A15	CoAtNet-0	18.782	20.640	18.989	21.053	21.053	20.640
PGD-A15	CaiT-XXS24	36.097	41.763	38.825	41.973	42.183	41.028
PGD-A15	NesT-Tiny	38.947	43.158	40.211	43.263	43.053	42.526
PGD-A15	PiT-Ti	28.512	31.551	29.036	31.342	31.132	31.447
MDCS	DeiT-Tiny	74.207	82.135	81.078	81.818	81.501	82.664
MDCS	CoAtNet-0	39.732	43.653	42.312	42.621	43.240	43.756
MDCS	CaiT-XXS24	67.786	74.816	72.928	74.502	74.397	74.921
MDCS	NesT-Tiny	54.842	63.263	61.158	62.105	62.421	64.632
MDCS	PiT-Ti	53.249	59.853	58.281	58.805	59.224	58.700
PI-FGSM	DeiT-Tiny	67.230	79.387	73.679	80.761	79.598	79.810
PI-FGSM	CoAtNet-0	44.376	45.820	44.066	46.543	46.027	45.614
PI-FGSM	CaiT-XXS24	61.805	71.144	66.946	70.829	70.304	70.199
PI-FGSM	NesT-Tiny	47.579	60.211	53.368	60.105	59.895	58.737
PI-FGSM	PiT-Ti	56.604	62.264	57.757	62.159	61.530	60.587
DI^2-FGSM	DeiT-Tiny	78.330	87.209	82.241	87.738	87.209	87.209
DI^2-FGSM	CoAtNet-0	46.440	53.870	49.329	54.902	54.180	54.592
DI^2-FGSM	CaiT-XXS24	71.773	81.952	74.816	81.847	81.847	81.742
DI^2-FGSM	NesT-Tiny	71.053	80.526	74.316	80.421	80.632	79.684
DI^2-FGSM	PiT-Ti	68.239	75.786	69.497	75.472	75.891	75.681

A.6.4 Effect of Action Density

Table A.15 shows that dense trajectory access exposes utility missed by sparse checkpoint sets. Each sparse slate uses the same frozen action subset for every attack and evaluation image. Two early actions plus final gain 3.964 points, four early actions gain 4.980 points, and all nine non-final actions gain 6.694 points. Dense access therefore adds 2.730 points over the two-action slate and 1.714 points over the four-action slate.

The two sparse variants have the same 94.2% non-final rate despite a 1.016-point gain difference. Selection frequency alone therefore does not explain readout quality: a denser slate improves the utility of the selected action even when the reader leaves final equally often. The 98.8% non-final rate of the dense setting further shows that useful intermediate states are distributed broadly enough that restricting the available actions discards measurable value. Table A.16 provides the corresponding absolute ASR in all 25 attack–target cases.

Table A.15: Action-density ablation.

Action slate	Final	CPRU	Gain	Non-final
Final only	54.829	54.829	0.000	0.0%
2 early actions + final	54.829	58.793	3.964	94.2%
4 early actions + final	54.829	59.808	4.980	94.2%
9 early actions + final	54.829	61.523	6.694	98.8%

Table A.16: Target-level absolute ASR for the action-density ablation. The headers denote the number of non-final actions plus the final fallback.

Attack	Target	Final	$2{+}F$	$4{+}F$	$9{+}F$
BP14	DeiT-Tiny	76.850	81.924	82.241	82.030
BP14	CoAtNet-0	41.486	44.169	45.614	45.511
BP14	CaiT-XXS24	69.360	74.502	74.921	75.341
BP14	NesT-Tiny	55.789	62.421	62.000	63.158
BP14	PiT-Ti	54.403	59.748	59.853	60.273
PGD-A15	DeiT-Tiny	47.252	50.106	49.471	52.748
PGD-A15	CoAtNet-0	18.782	20.021	19.917	20.640
PGD-A15	CaiT-XXS24	36.097	39.035	39.664	41.763
PGD-A15	NesT-Tiny	38.947	40.421	40.526	43.158
PGD-A15	PiT-Ti	28.512	29.560	29.979	31.551
MDCS	DeiT-Tiny	74.207	80.867	81.290	82.135
MDCS	CoAtNet-0	39.732	42.828	44.066	43.653
MDCS	CaiT-XXS24	67.786	74.292	73.452	74.816
MDCS	NesT-Tiny	54.842	61.789	62.842	63.263
MDCS	PiT-Ti	53.249	58.595	59.119	59.853
PI-FGSM	DeiT-Tiny	67.230	72.093	76.321	79.387
PI-FGSM	CoAtNet-0	44.376	44.788	44.685	45.820
PI-FGSM	CaiT-XXS24	61.805	64.113	68.206	71.144
PI-FGSM	NesT-Tiny	47.579	50.737	56.737	60.211
PI-FGSM	PiT-Ti	56.604	58.071	61.530	62.264
DI^2-FGSM	DeiT-Tiny	78.330	83.615	83.932	87.209
DI^2-FGSM	CoAtNet-0	46.440	50.464	51.290	53.870
DI^2-FGSM	CaiT-XXS24	71.773	77.440	78.489	81.952
DI^2-FGSM	NesT-Tiny	71.053	76.421	76.211	80.526
DI^2-FGSM	PiT-Ti	68.239	71.803	72.851	75.786

A.6.5 Effect of Development Size

Table A.17 averages five seeds for 250, 500, and 1,000 images. For each seed and size, images are sampled without replacement from the fixed 2,000-image development pool, and all five attack readers are refit with the same saved trajectories, descriptors, hyperparameters, and pre-sampled evaluation population. The 2,000-image row is the single canonical fit. This protocol changes only the amount of reader supervision and leaves the formal evaluation fixed.

Mean gain increases monotonically from 6.128 points at 250 images to 6.607 at 1,000 and 6.776 in the canonical fit. Across the replicated sizes, the standard deviation falls from 0.584 to 0.183 as the development population grows. CPRU is therefore already effective with limited supervision, while additional development images primarily improve stability and produce progressively smaller gains in the mean. Table A.18 reports target-level absolute ASR, averaging the five archived fits at each replicated development size.

Table A.17: Development-size sensitivity. Standard deviation is across five seeds when available.

Images	Fits	Final	CPRU	Gain	Gain std.
250	5	54.829	60.957	6.128	0.584
500	5	54.829	61.130	6.301	0.327
1,000	5	54.829	61.436	6.607	0.183
2,000	1	54.829	61.605	6.776	–

Table A.18: Target-level absolute ASR for the development-size ablation. Values for 250, 500, and 1,000 images average five reader fits; 2,000 images is the canonical single fit.

Attack	Target	Final	250	500	1,000	2,000
BP14	DeiT-Tiny	76.850	81.247	81.755	82.220	82.135
BP14	CoAtNet-0	41.486	44.892	45.490	45.862	46.543
BP14	CaiT-XXS24	69.360	74.565	74.795	74.879	74.711
BP14	NesT-Tiny	55.789	61.811	63.242	63.263	63.368
BP14	PiT-Ti	54.403	58.994	59.266	59.560	60.901
PGD-A15	DeiT-Tiny	47.252	52.770	52.875	52.685	52.537
PGD-A15	CoAtNet-0	18.782	20.660	20.929	20.743	20.846
PGD-A15	CaiT-XXS24	36.097	41.826	41.742	41.763	41.658
PGD-A15	NesT-Tiny	38.947	42.779	43.179	42.989	43.158
PGD-A15	PiT-Ti	28.512	31.698	31.279	31.426	31.656
MDCS	DeiT-Tiny	74.207	80.846	81.121	82.030	81.501
MDCS	CoAtNet-0	39.732	43.137	43.426	43.798	42.828
MDCS	CaiT-XXS24	67.786	74.208	74.313	74.711	75.446
MDCS	NesT-Tiny	54.842	62.168	62.611	63.411	64.000
MDCS	PiT-Ti	53.249	58.658	58.700	59.161	58.805
PI-FGSM	DeiT-Tiny	67.230	78.901	78.837	79.429	79.704
PI-FGSM	CoAtNet-0	44.376	45.552	45.655	45.573	46.543
PI-FGSM	CaiT-XXS24	61.805	70.535	69.780	71.018	71.459
PI-FGSM	NesT-Tiny	47.579	58.989	58.316	59.453	60.000
PI-FGSM	PiT-Ti	56.604	61.782	61.300	62.159	62.474
DI^2-FGSM	DeiT-Tiny	78.330	86.850	87.315	87.357	87.421
DI^2-FGSM	CoAtNet-0	46.440	54.118	54.530	54.572	54.489
DI^2-FGSM	CaiT-XXS24	71.773	81.637	81.763	81.826	81.952
DI^2-FGSM	NesT-Tiny	71.053	79.474	80.063	80.126	80.316
DI^2-FGSM	PiT-Ti	68.239	75.828	75.975	75.891	75.681

A.6.6 Reader Reuse Across Attacks

Table A.19 compares attack-specific fitting with shared alternatives. The pooled variants combine records from all five attacks, either without an attack covariate or with a categorical attack indicator. Leave-one-attack-out (LOAO) fits on four attacks, evaluates the held-out fifth, and averages the five held-out results. These variants test whether the descriptor captures a common notion of relative utility rather than attack-name-specific shortcuts.

A pooled reader without an attack identifier reaches 6.978 points, the indicator variant reaches 6.815 points, and LOAO remains positive at 6.490 points. The pooled results show that explicit attack identity is unnecessary when all attacks are represented during fitting. More importantly, LOAO retains 97.0% of the attack-specific gain while never fitting on the evaluated attack, providing direct evidence of reusable cross-attack structure. The canonical method remains attack specific because that is the prespecified primary protocol; the reuse study demonstrates robustness of the representation rather than redefining the main method after evaluation. Table A.20 exposes the absolute ASR for every attack–target pair and reader-sharing rule.

Table A.19: Reader-sharing ablation on the same trajectories and pre-sampled evaluation images.

Reader training	Final	CPRU	Gain	Non-final
Attack-specific	54.829	61.523	6.694	98.8%
Pooled without attack ID	54.829	61.807	6.978	99.3%
Pooled with attack ID	54.829	61.644	6.815	98.8%
Leave-one-attack-out	54.829	61.319	6.490	99.8%

Table A.20: Target-level absolute ASR for the reader-sharing ablation. Pooled+ID includes an attack indicator, while LOAO excludes the evaluated attack during fitting.

Attack	Target	Final	Specific	Pooled	Pooled+ID	LOAO
BP14	DeiT-Tiny	76.850	82.030	82.981	82.241	82.347
BP14	CoAtNet-0	41.486	45.511	45.201	45.820	45.201
BP14	CaiT-XXS24	69.360	75.341	75.971	75.236	76.076
BP14	NesT-Tiny	55.789	63.158	64.842	63.579	65.053
BP14	PiT-Ti	54.403	60.273	60.692	60.273	59.539
PGD-A15	DeiT-Tiny	47.252	52.748	52.748	52.960	52.748
PGD-A15	CoAtNet-0	18.782	20.640	20.433	20.640	21.156
PGD-A15	CaiT-XXS24	36.097	41.763	42.078	42.288	42.078
PGD-A15	NesT-Tiny	38.947	43.158	43.053	43.263	43.053
PGD-A15	PiT-Ti	28.512	31.551	31.761	31.866	31.656
MDCS	DeiT-Tiny	74.207	82.135	81.924	81.712	82.558
MDCS	CoAtNet-0	39.732	43.653	44.582	43.550	44.788
MDCS	CaiT-XXS24	67.786	74.816	76.076	74.816	75.551
MDCS	NesT-Tiny	54.842	63.263	64.737	63.474	65.158
MDCS	PiT-Ti	53.249	59.853	58.700	58.805	58.595
PI-FGSM	DeiT-Tiny	67.230	79.387	79.281	79.598	76.427
PI-FGSM	CoAtNet-0	44.376	45.820	47.265	46.027	47.162
PI-FGSM	CaiT-XXS24	61.805	71.144	71.668	71.354	68.520
PI-FGSM	NesT-Tiny	47.579	60.211	59.789	60.000	56.737
PI-FGSM	PiT-Ti	56.604	62.264	61.635	62.264	61.635
DI^2-FGSM	DeiT-Tiny	78.330	87.209	87.104	87.844	86.469
DI^2-FGSM	CoAtNet-0	46.440	53.870	54.592	54.592	53.457
DI^2-FGSM	CaiT-XXS24	71.773	81.952	82.162	82.371	82.162
DI^2-FGSM	NesT-Tiny	71.053	80.526	80.211	80.842	79.474
DI^2-FGSM	PiT-Ti	68.239	75.786	75.681	75.681	75.367

AI USE STATEMENT

In this work, we used generative AI tools only for grammar correction and minor language edits to improve readability. We have not used generative AI tools for other tasks. We have reviewed all AI-assisted work, and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.