TALA: SURROGATE-FREE QUERY-BASED BLACK-BOX ATTACKS ON IMAGE FORGERY LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cross-detector adversarial transfer for image forgery localization varies across surrogate–target pairs. Our analysis identifies substantial source-to-target effectiveness gaps on initially well-localized images, while attribution analysis suggests that the forensic evidence used for localization is both model- and input-dependent. These observations motivate us to explore whether victim localization feedback alone can guide effective attacks, without relying on cross-detector transfer. To this end, we propose Target-Adaptive Localization Attack (**TALA**), to the best of our knowledge, **the first** query-based attack specifically designed for image forgery localization that requires neither a surrogate detector nor a surrogate-trained attack policy. TALA iteratively updates the image through local perturbations, using victim feedback to adapt their location, spatial extent, and type. It combines victim-guided patch-center sampling with failure-aware spatial penalties and determines patch sizes from the entropy-based effective support of the guided distribution. A shared rejection-feedback mechanism coordinates spatial search and switching between complementary smooth and high-frequency perturbation engines. Across three heterogeneous detectors, five datasets, and four evaluation conditions, TALA achieves the highest mean attack success rate in **58 of 60** settings and ties for the highest in one additional setting against four baselines. It simultaneously achieves higher attack success, lower LPIPS, and higher PSNR than the strongest baseline in attack success rate in 51 settings. These findings establish surrogate-free target-side adaptation as an effective complementary approach to black-box robustness evaluation and expose the susceptibility of forgery localization detectors to local perturbations guided directly by their own feedback.

1 INTRODUCTION

Advances in image editing and generative models have made image manipulation increasingly accessible, raising concerns about the authenticity of visual content. Image forgery localization addresses this problem by identifying manipulated regions at the pixel level. Despite substantial progress (Kwon et al., 2022; Dong et al., 2022; Wu et al., 2022; Guillaro et al., 2023; Wu et al., 2025), localization models may remain vulnerable to deliberate perturbations designed to conceal manipulation traces. Evaluating their robustness against such attacks is therefore important, particularly in practical and challenging black-box settings, where attackers cannot access its parameters or gradients. However, existing attacks often relax this setting by assuming either white-box access or access to a surrogate detector. Transfer-based methods generate Adversarial Examples (AEs) on a surrogate and apply them to an unseen victim (Cao et al., 2024; Zeng & Pun, 2024; Zhu et al., 2023; Fang & Stamm, 2023; Zhuo et al., 2025). RLGC (Mo et al., 2025) and AdvFor (Mo et al., 2026) incorporate victim queries, but still rely on an attack policy trained on a surrogate detector. DiffCFG (Tailanián et al., 2024) avoids a surrogate detector through diffusion-based generation, but does not adapt its perturbations to victim feedback during the attack. Meanwhile, general-purpose query-based attacks developed for image classification (Guo et al., 2019; Andriushchenko et al., 2020; Al-Dujaili & O'Reilly, 2020) provide surrogate-free alternatives, but do not explicitly exploit the dense spatial predictions available in forgery localization. This raises a practical question: *How can we use localization feedback to guide an effective query-based attack without a surrogate detector?*

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.

Our empirical analysis provides further evidence for this direction. Although transfer-based attacks can achieve non-trivial cross-detector effectiveness, their performance remains sensitive to the source–target pair, and the transferability gap becomes particularly pronounced on images that are initially localized with high accuracy. Attribution analysis further suggests that the forensic cues exploited by different detectors vary across both models and inputs. These observations motivate exploring direct adaptation to the current victim and input as a complementary approach that requires neither cross-detector transfer nor a surrogate-trained attack policy.

We propose **T**arget-**A**daptive **L**ocalization **A**ttack (**TALA**), a surrogate-free query-based attack that jointly adapts *where* and *how* to perturb. For *where to perturb*, TALA constructs a victim-guided spatial distribution for patch-center sampling from the victim's evolving localization predictions, and reduces sampling weights around locations where previous perturbations failed to improve the attack objective. It also adjusts patch dimensions according to the entropy-derived spatial support of this distribution, allowing both perturbation location and spatial extent to adapt during the search. For *how to perturb*, TALA selects between complementary smooth and high-frequency perturbations. A shared rejection-feedback mechanism coordinates spatial search and perturbation selection, allowing the attack to change its strategy when no further improvement is observed. We evaluate TALA across three heterogeneous localization detectors and five datasets covering diverse image manipulation types, under different query budgets and initial localization accuracy conditions. TALA achieves the highest or tied-highest Attack Success Rate (ASR) in nearly all settings of the main comparison, with a favorable trade-off between attack effectiveness and perceptual quality. Comparisons with general image degradations and alternative perturbation designs further assess the benefits of the proposed search. Ablations examine the roles of victim-guided patch-center sampling, distribution-adaptive patch sizing, and complementary perturbation types. Our main contributions are:

- We identify source-to-target effectiveness gaps for the evaluated transfer attacks on initially well-localized images. Our attribution analysis further suggests that forensic cues vary across models and inputs, motivating direct victim-and-input adaptation as a complementary approach to transfer-based attacks.
- To the best of our knowledge, TALA is the first surrogate-free query-based black-box attack specifically designed for image forgery localization. It requires neither a surrogate detector nor a surrogate-trained attack policy, and directly adapts to each victim and input through victim-guided spatial search and complementary perturbation engines.
- Across 60 evaluation settings, TALA achieves the highest mean ASR in 58 and ties for the highest in one additional setting. In 51 settings, it simultaneously achieves higher ASR, lower LPIPS, and higher PSNR than the strongest baseline in ASR. Ablation studies further assess the contributions of its components.

2 Related Work

2.1 Image Forgery Localization Detectors

Image forgery detection and localization have evolved from handcrafted forensic analysis to deep learning models that directly learn manipulation-sensitive representations. Representative methods exploit diverse forensic cues and learning strategies. CAT-Net (Kwon et al., 2022) localizes forged regions by classifying DCT coefficients, while MVSS-Net (Dong et al., 2022) extracts forgery features through multi-view learning. IFOSN (Wu et al., 2022) introduces a robust training framework based on adversarial noise modeling to improve forgery detection under image transmission through online social networks, whereas TruFor (Guillaro et al., 2023) combines RGB information with a learned noise-sensitive fingerprint to extract complementary forensic clues. More recently, FOCAL (Wu et al., 2025) employs contrastive learning to model the relationships between forged and pristine regions in the feature space, enabling the capture of subtle forensic inconsistencies without relying on predefined artifacts. MLLM-based approaches such as FakeShield (Xu et al., 2025) combine forgery detection and localization with textual explanations. Our robustness evaluation focuses on IFOSN, TruFor, and FOCAL, which represent different forensic cues and learning strategies.

(a) Cross-detector transferability

Surrogate	Input	Target detector: F1		
		TruFor	FOCAL	IFOSN
–	Clean (no filter)	.6931	.8667	.5091
TruFor	Clean	.8424	.5767	.5024
	PGN	*.0816*	.2975	.0662
FOCAL	Clean	.6090	.7844	.4587
	PGN	.3186	*.0712*	.1821
IFOSN	Clean	.7370	.6404	.8281
	PGN	.3998	.3892	*.1013*

Italics denote attacks evaluated on the surrogate itself.

(b) Forensic feature misalignment

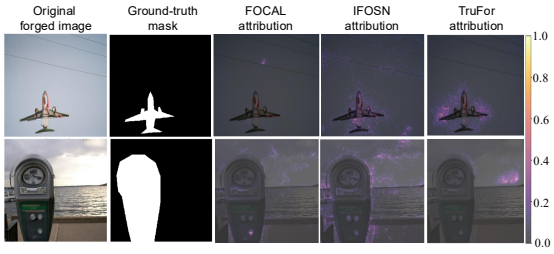


Figure 1: Diagnostic evidence for cross-detector transferability. (a) Clean and post-attack F1 scores on CASIAv1 using PGN (Ge et al., 2023). The first row reports clean performance on the same unfiltered images for all three detectors. Each subsequent surrogate group reports paired Clean and PGN results on images selected with initial localization F1 > 0.5 on that surrogate; these subsets can differ across groups. (b) Integrated Gradients visualizations illustrate model- and input-dependent attribution patterns for forgery localization.

2.2 Adversarial Attacks on Image Forgery Localization

Adversarial attacks deliberately modify images to disrupt forensic predictions. Several methods achieve black-box attacks by transferring AEs generated on accessible forensic models. White-box gradient-based attacks have been studied for image tampering localization (Cao et al., 2024). Zeng & Pun (2024) further combine low-frequency constraints with feature-gradient aggregation to improve image quality and cross-model transferability. Generative methods can also rely on transferability in black-box settings. Zhu et al. (2023) train a frequency-aware GAN against an accessible detector and transfer the generated examples to other models. Fang & Stamm (2023) train a generator to synthesize consistent forensic traces and demonstrate transfer to a localization model not used in attack training. SEAR (Zhuo et al., 2025) uses self-supervised learning and a surrogate detector to train a network that generates perturbations to conceal manipulation traces. When applied to other detectors without further training, it likewise relies on cross-detector transferability. DiffCFG (Tailanián et al., 2024) uses diffusion models to modify forged images without a surrogate forensic detector. Like the transfer-based methods above, however, it does not adapt the attack online to victim feedback. RLGC (Mo et al., 2025) incorporates victim queries, although its attack policy is trained on a surrogate detector. AdvFor (Mo et al., 2026) further extends RLGC from soft-label feedback (probability maps) to a hard-label, mask-only setting. In contrast, TALA requires neither a surrogate detector nor attack-policy training, and directly uses victim feedback to craft perturbations.

3 Observation: Cross-Detector Transferability Gap

Before introducing our attack, we examine the effectiveness of surrogate-based transfer across image forgery localization detectors. We find that transfer can be effective, but varies substantially across surrogate–target pairs, motivating exploring surrogate-free adaptation to the current victim and input as a complementary approach.

Cross-detector transferability gap. We conduct a diagnostic transfer study on CASIAv1 (Dong et al., 2013) using three heterogeneous detectors: TruFor, FOCAL, and IFOSN. AEs are generated with white-box access to a surrogate under an ℓ_∞ budget of $\epsilon \approx 0.020$ and evaluated on the other detectors without victim-side adaptation. The complete study in Appendix A evaluates multiple transfer attacks under two initial-F1 filtering conditions. On images with initial F1 > 0, PGN (Ge et al., 2023) achieves the strongest overall cross-detector transfer among the evaluated attacks. Fig. 1(a) highlights the performance of PGN on images with initial F1 > 0.5 on each surrogate, reporting paired clean and post-attack results. PGN substantially degrades all three detectors, but its effectiveness varies across surrogate–target pairs. With IFOSN as the surrogate, it reduces the source-model F1 from 0.8281 to 0.1013, whereas TruFor's F1 only decreases from 0.7370 to 0.3998 on the same images. Transfer can nevertheless be strong: with TruFor as the surrogate, IFOSN's F1 decreases from 0.5024 to 0.0662. These results show that strong surrogate-side effectiveness does not consistently translate into equally strong degradation of another detector.

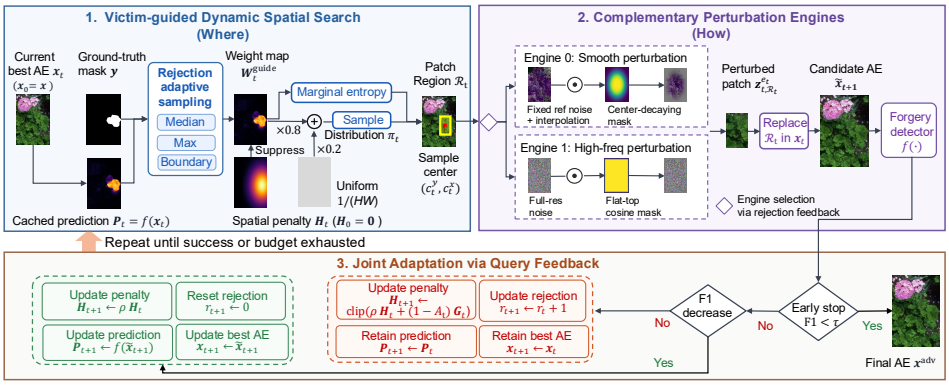


Figure 2: Overview of the proposed TALA. **Where:** Victim feedback and past failures guide patch-center sampling with 20% uniform exploration, while only the guided distribution determines the patch size. **How:** Complementary smooth and high-frequency perturbation engines generate candidate AEs. **Adaptation:** Rejection feedback jointly updates the spatial search (Where) and perturbation-engine selection (How) until attack success or query-budget exhaustion.

Forensic feature misalignment. To investigate a possible explanation for this variability, we analyze detector attribution patterns using Integrated Gradients (IG) (Sundararajan et al., 2017). As shown in Fig. 1(b), attribution patterns differ across detectors for the same forgery and across inputs for the same detector. We refer to this phenomenon as *forensic feature misalignment*: the attributed evidence underlying forgery localization is both *model-dependent* and *input-dependent*. Additional examples are provided in Appendix A. These observations suggest that perturbations optimized against a surrogate may target forensic evidence that is less influential for another detector, although attribution alone does not establish causality. These observations motivate exploring surrogate-free attacks that adapt directly to the current victim and input. Accordingly, TALA uses victim feedback to jointly adapt *where* and *how* to perturb, as detailed in the following section.

4 Method

4.1 Problem Formulation and Overview

Let $\boldsymbol{x} \in [0,1]^{C\times H\times W}$ be a manipulated image with a nonempty manipulation mask $\boldsymbol{y} \in \{0,1\}^{H\times W}$. We assume that the attacker creates the manipulation or assists its creator and thus knows $\boldsymbol{y}$. The attacker can query the victim's localization probabilities, but cannot access its parameters or gradients and uses no surrogate detector. Under a query budget Q, we seek an adversarial image $\boldsymbol{x}^{\mathrm{adv}}$ that minimizes the pixel-level localization F1 computed from the victim's probability map thresholded at 0.5, which we refer to as discrete F1:

$$\min_{\boldsymbol{x}^{\mathrm{adv}}} \quad \mathcal{L}(\boldsymbol{x}^{\mathrm{adv}}) := \mathrm{F1}\big(\mathbb{1}[f(\boldsymbol{x}^{\mathrm{adv}}) > 0.5], \boldsymbol{y}\big), \quad \text{s.t.} \quad \|\boldsymbol{x}^{\mathrm{adv}} - \boldsymbol{x}\|_\infty \le \epsilon, \quad \boldsymbol{x}^{\mathrm{adv}} \in [0,1]^{C\times H\times W}. \tag{1}$$

To optimize this objective, TALA iteratively updates the current best image through local perturbations, accepting only updates that strictly reduce discrete F1. This criterion aligns with the metric used to determine attack success and requires only binary masks for update decisions, supporting future extensions of the criterion to hard-label settings. An alternative based on continuous soft-F1, computed directly from probabilities without thresholding, yields comparable aggregate performance (Appendix B.1). Fig. 2 and Algorithm 1 summarize the framework. We next describe its three components: *where* to perturb through victim-guided spatial search, *how* to perturb using complementary engines, and *adaptation* of the search and engine selection based on victim feedback.

4.2 Victim-Guided Dynamic Spatial Search

Victim-guided patch-center sampling. At iteration t, we adapt patch-center sampling by selecting a state s_t according to the number of consecutive rejections r_t: *Median* for $r_t < 2$, *Max* for

$2 \le r_t < 4$, and *Boundary* for $r_t \ge 4$. The *Median* state favors true positives with confidence scores near their current median, providing a distribution-adaptive starting point. After repeated rejections, *Max* shifts the preference toward the highest-confidence true positives. Persistent failure activates *Boundary*, which removes the true-positive restriction and favors predictions near 0.5 across the full image. Thus, consecutive unsuccessful perturbations shift the search toward different types of spatial evidence. Given the selected state s_t, we use the best image found so far, $\boldsymbol{x}_t$, and its cached victim prediction $\boldsymbol{P}_t = f(\boldsymbol{x}_t) \in [0,1]^{H\times W}$ to construct a spatial guidance weight map $\boldsymbol{W}_t^{\text{guide}}$. The precise construction of this map is provided in Appendix B.2. We then combine $\boldsymbol{W}_t^{\text{guide}}$ with a spatial penalty map $\boldsymbol{H}_t$, which records the locations of unsuccessful perturbations and is updated using victim feedback as described in Sec. 4.4. The penalty suppresses recently unsuccessful regions, and the resulting weights are normalized to form a guided sampling distribution:

$$\widetilde{W}_t(i,j) = W_t^{\text{guide}}(i,j)\exp[-\lambda H_t(i,j)], \qquad \pi_t^{\text{guide}}(i,j) = \widetilde{W}_t(i,j)/\big(\sum\nolimits_{(\bar{i},\bar{j})\in\Omega}\widetilde{W}_t(\bar{i},\bar{j})\big). \quad (2)$$

Here $\lambda > 0$ controls the suppression strength, and $\Omega = \{0,\dots,H-1\}\times\{0,\dots,W-1\}$ denotes the image lattice, with i and j indexing rows and columns, respectively; $\bar{i}$ and $\bar{j}$ are dummy summation indices. Finally, we mix π_t^{guide} with a uniform distribution over the image and sample the patch center $\boldsymbol{c}_t$ from the resulting distribution:

$$\pi_t(i,j) = \eta\,\pi_t^{\text{guide}}(i,j) + (1-\eta)\pi^{\text{uni}}(i,j), \qquad \boldsymbol{c}_t \sim \pi_t. \quad (3)$$

Here $\pi^{\text{uni}}(i,j) = 1/(HW)$. We set $\eta = 0.8$, retaining 20% uniform exploration so that regions with low guided weights can still be sampled.

Distribution-adaptive patch size. After sampling the patch center $\boldsymbol{c}_t$, we determine the height and width of the local region $\mathcal{R}_t$ to be perturbed in the next query. We derive these dimensions from the spatial spread of the current guided distribution π_t^{guide}, so that the update region expands or contracts according to the current victim feedback. To measure this spread, we first compute the row and column marginals:

$$\mu_t^y(i) = \sum\nolimits_j \pi_t^{\text{guide}}(i,j), \qquad \mu_t^x(j) = \sum\nolimits_i \pi_t^{\text{guide}}(i,j). \quad (4)$$

Afterwards, for each axis $d \in \{y,x\}$, we characterize the spread of the corresponding marginal through its entropy-based effective support (Shannon, 1948; Hill, 1973):

$$n_t^d = \exp\big(-\sum\nolimits_k \mu_t^d(k)\log\mu_t^d(k)\big), \qquad u_t^d = (n_t^d - 1)/(L_d - 1), \quad (5)$$

where $L_y = H$ and $L_x = W$, and we adopt the convention $0\log 0 = 0$. The normalized support $u_t^d \in [0,1]$ is smaller for concentrated marginals and larger for diffuse marginals. We then map the normalized support to the allowable patch-size range for each axis:

$$b_{t,d} = \big\lfloor b_d^- + (b_d^+ - b_d^-)u_t^d + 1/2 \big\rfloor, \qquad d \in \{y,x\}, \quad (6)$$

where b_d^- and b_d^+ are the minimum and maximum allowable dimensions determined by $(r_{\min}, r_{\max})$ and the minimum patch side length $\ell_{\min}$. The resulting height $h_t = b_{t,y}$ and width $w_t = b_{t,x}$ define a nominal patch centered at $\boldsymbol{c}_t$, which is clipped to the image boundaries to obtain $\mathcal{R}_t$. The patch is therefore smaller for spatially concentrated evidence and larger for broadly distributed evidence. We compute u_t^d before adding the uniform exploration component, so that exploration does not artificially broaden the estimated spatial support. Unlike a predefined linear shrinking schedule (Appendix B.4), this design determines patch dimensions from the current guided distribution. Our ablation study shows that adaptive sizing improves perceptual quality while largely preserving ASR (Sec. 5.3). Exact bounds, patch-region construction, and boundary handling are provided in Appendix B.3.

4.3 Complementary Perturbation Engines

Having determined *where* to perturb through victim-guided spatial search, we next determine *how* to perturb the region $\mathcal{R}_t$. To this end, TALA uses two complementary engines, with the active engine $e_t \in \{0,1\}$ generating a perturbed patch $\boldsymbol{z}_{t,\mathcal{R}_t}^{(e_t)}$. *Engine 0 (smooth)* combines a fixed Rademacher-perturbed reference image with a bicubically upsampled low-resolution Rademacher (Vershynin,

Algorithm 1 TALA: joint spatial and perturbation adaptation

Require: Image $\boldsymbol{x}$, mask $\boldsymbol{y}$, victim f, budget ϵ, query limit $Q \geq 1$, target F1 $\tau > 0$
Ensure: Best image $\boldsymbol{x}_t$ with $\|\boldsymbol{x}_t - \boldsymbol{x}\|_\infty \leq \epsilon$

$t \leftarrow 0$; $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}$; $r_0 \leftarrow 0$; $\boldsymbol{H}_0 \leftarrow \mathbf{0}$; $e_0 \leftarrow 0$ ▷ Initialize with Engine 0
$\boldsymbol{P}_0 \leftarrow f(\boldsymbol{x}_0)$; $q \leftarrow 1$; $\mathcal{L}_0 \leftarrow \mathrm{F1}(\mathbb{1}[\boldsymbol{P}_0 > 0.5], \boldsymbol{y})$
Sample the fixed reference $\boldsymbol{x}^B$ (Eq. 19)
while $q < Q$ **and** $\mathcal{L}_t \geq \tau$ **do**
  Select s_t from r_t and construct $\boldsymbol{W}_t^{\mathrm{guide}}$ (Eq. 12) ▷ Where to perturb
  Form π_t^{guide} by normalizing $\boldsymbol{W}_t^{\mathrm{guide}} \odot \exp(-\lambda \boldsymbol{H}_t)$
  Sample perturbation center $\boldsymbol{c}_t \sim 0.8\,\pi_t^{\mathrm{guide}} + 0.2\ \mathrm{Uniform}(\Omega)$
  Compute perturbation size (h_t, w_t) from the marginal effective supports of π_t^{guide} (Sec. 4.2)
  Construct perturbation region $\mathcal{R}_t$ from $\boldsymbol{c}_t$ and (h_t, w_t), clipped to image boundaries
  Form $\widetilde{\boldsymbol{x}}_{t+1}$ by replacing $\boldsymbol{x}_t$ on $\mathcal{R}_t$ with $\boldsymbol{z}_{t,\mathcal{R}_t}^{(e_t)}$ (Eq. 7) ▷ How to perturb
  $\widetilde{\boldsymbol{P}}_{t+1} \leftarrow f(\widetilde{\boldsymbol{x}}_{t+1})$; $q \leftarrow q + 1$ ▷ Query the victim
  $\widetilde{\mathcal{L}}_{t+1} \leftarrow \mathrm{F1}(\mathbb{1}[\widetilde{\boldsymbol{P}}_{t+1} > 0.5], \boldsymbol{y})$; $A_t \leftarrow \mathbb{1}[\widetilde{\mathcal{L}}_{t+1} < \mathcal{L}_t]$
  If $A_t = 1$: $r_{t+1} \leftarrow 0$, $\boldsymbol{x}_{t+1} \leftarrow \widetilde{\boldsymbol{x}}_{t+1}$, $(\boldsymbol{P}_{t+1}, \mathcal{L}_{t+1}) \leftarrow (\widetilde{\boldsymbol{P}}_{t+1}, \widetilde{\mathcal{L}}_{t+1})$ ▷ Adaptation
  If $A_t = 0$: $r_{t+1} \leftarrow r_t + 1$, $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t$, $(\boldsymbol{P}_{t+1}, \mathcal{L}_{t+1}) \leftarrow (\boldsymbol{P}_t, \mathcal{L}_t)$
  Update $\boldsymbol{H}_{t+1}, e_{t+1}$ using A_t (Sec. 4.4)
  $t \leftarrow t + 1$
end while
return $\boldsymbol{x}_t$

2018) field and attenuates the perturbation progressively away from the patch center. *Engine 1 (high-frequency)* uses independent full-resolution Rademacher perturbations and a flat-top mask with cosine tapering only near the patch boundary. Their shared local construction is

$$\boldsymbol{z}_{t,\mathcal{R}_t}^{(e_t)} = \boldsymbol{x}_{\mathcal{R}_t} + \boldsymbol{M}_t^{(e_t)} \odot \big(\hat{\boldsymbol{x}}_{t,\mathcal{R}_t}^{(e_t)} - \boldsymbol{x}_{\mathcal{R}_t}\big), \qquad e_t \in \{0, 1\}, \tag{7}$$

where $\hat{\boldsymbol{x}}_{t,\mathcal{R}_t}^{(e_t)}$ is the engine-generated image patch, projected to satisfy the ℓ_∞ budget ϵ, and $\boldsymbol{M}_t^{(e_t)}$ is a spatial mask with entries in $[0, 1]$, broadcast across channels. The mask controls the spatial attenuation described above. Detailed constructions of both perturbations and their masks are provided in Appendix B.5. The two engines exhibit partially distinct success sets at approximately matched PSNR (Sec. 5.3), motivating adaptation between them. TALA retains the current engine after an acceptance and switches engines after sufficiently many consecutive rejections. This feedback-driven selection is described next.

4.4 Joint Adaptation via Query Feedback

After choosing where and how to perturb, TALA uses the victim's response to update the current best image and guide the next iteration's spatial search and engine selection. We first construct the candidate $\widetilde{\boldsymbol{x}}_{t+1}$ by replacing the pixels of $\boldsymbol{x}_t$ within $\mathcal{R}_t$ with $\boldsymbol{z}_{t,\mathcal{R}_t}^{(e_t)}$, leaving all other pixels unchanged. One victim query evaluates this candidate. We accept only strict improvement, with $A_t = \mathbb{1}[\mathcal{L}(\widetilde{\boldsymbol{x}}_{t+1}) < \mathcal{L}(\boldsymbol{x}_t)]$. If accepted, the current best image $\boldsymbol{x}_{t+1} = \widetilde{\boldsymbol{x}}_{t+1}$ and the cached response is replaced; otherwise, both remain unchanged. The acceptance decision updates the rejection count as $r_{t+1} = (1 - A_t)(r_t + 1)$, which determines the next sampling state s_{t+1} and whether to switch engines. The engine switches to $e_{t+1} = 1 - e_t$ when $r_{t+1} \geq K$, with $K = 6$, and otherwise remains unchanged. Thus, consecutive rejections first change the spatial sampling preference; from the sixth consecutive rejection onward, the engines alternate on each failure while sampling remains in *Boundary*. Acceptance resets the rejection count to zero, returning sampling to *Median* and retaining the improving engine. The same feedback updates the spatial penalty map:

$$\boldsymbol{H}_{t+1} = \mathrm{clip}_{[0, H_{\max}]}\big(\rho \boldsymbol{H}_t + (1 - A_t)\boldsymbol{G}_t\big). \tag{8}$$

Here $0 < \rho < 1$ controls penalty decay, $H_{\max} = 5$ sets the maximum penalty, and $\boldsymbol{G}_t$ is a Gaussian centered at the attempted location with patch-dependent scales (Appendix B.2). Rejected attempts add a local penalty, reducing the sampling weights of recently unsuccessful regions. Existing penalties decay at every iteration and are not reset by acceptance, allowing previously unsuccessful locations to be revisited.

Table 1: Attack effectiveness and perceptual quality on images with initial localization F1 > 0.5 under a maximum query budget of $Q = 50$. Each entry reports ASR (%) ↑ / LPIPS ↓, both as mean ± standard deviation over five independent random seeds. Across all 15 detector–dataset combinations, TALA achieves the highest mean ASR while also attaining lower mean LPIPS than the strongest baseline in ASR.

Victim	Method	CASIAv1	COCOGlide	COVERAGE	IMD2020	FaceApp
FOCAL	DiffCFG (wacv'24)	30.8 ± 4.1 / 0.1182 ± 0.0034	11.6 ± 3.3 / **0.1123** ± 0.0133	14.0 ± 4.0 / **0.1091** ± 0.0067	4.0 ± 2.0 / 0.1576 ± 0.0246	12.4 ± 4.3 / 0.1524 ± 0.0023
	AdvFor (TIFS'26)	25.6 ± 2.6 / 0.1663 ± 0.0117	14.4 ± 2.6 / 0.1675 ± 0.0158	5.2 ± 2.3 / 0.1525 ± 0.0206	3.6 ± 2.6 / 0.2549 ± 0.0349	14.4 ± 1.7 / 0.1693 ± 0.0211
	RLGC (AAAI'25)	37.6 ± 2.6 / 0.1990 ± 0.0180	26.0 ± 2.4 / 0.2794 ± 0.0110	26.4 ± 4.1 / 0.3343 ± 0.0205	13.6 ± 3.0 / 0.4255 ± 0.0523	21.6 ± 0.9 / 0.1904 ± 0.0388
	Square (ECCV'20)	58.4 ± 3.6 / 0.1355 ± 0.0028	60.8 ± 7.2 / 0.1566 ± 0.0135	28.0 ± 3.7 / 0.1156 ± 0.0115	34.8 ± 4.3 / 0.1604 ± 0.0360	90.0 ± 3.2 / 0.1471 ± 0.0013
	TALA (Ours)	**70.4** ± 4.3 / **0.0640** ± 0.0084	**61.6** ± 2.6 / 0.1241 ± 0.0131	**91.2** ± 1.8 / 0.1118 ± 0.0059	**43.2** ± 5.8 / **0.1424** ± 0.0167	**96.8** ± 1.8 / **0.1053** ± 0.0077
TruFor	DiffCFG (wacv'24)	27.6 ± 3.8 / 0.1181 ± 0.0028	16.4 ± 1.7 / 0.1180 ± 0.0129	19.2 ± 2.7 / 0.1157 ± 0.0068	12.0 ± 4.0 / 0.1656 ± 0.0036	2.8 ± 1.1 / 0.1512 ± 0.0156
	AdvFor (TIFS'26)	28.8 ± 4.1 / 0.1430 ± 0.0119	40.4 ± 2.6 / 0.2460 ± 0.0115	20.4 ± 1.7 / 0.1918 ± 0.0103	4.4 ± 0.9 / 0.1856 ± 0.0108	31.2 ± 2.7 / 0.1621 ± 0.0101
	RLGC (AAAI'25)	43.2 ± 2.3 / 0.1866 ± 0.0101	71.2 ± 1.8 / 0.2811 ± 0.0074	40.4 ± 0.9 / 0.3350 ± 0.0067	39.2 ± 1.8 / 0.4609 ± 0.0067	69.6 ± 2.6 / 0.3248 ± 0.0115
	Square (ECCV'20)	51.2 ± 3.0 / 0.1028 ± 0.0018	42.8 ± 4.1 / 0.2123 ± 0.0161	37.2 ± 3.0 / **0.1027** ± 0.0081	25.6 ± 2.6 / **0.1294** ± 0.0120	83.6 ± 4.3 / 0.1804 ± 0.0049
	TALA (Ours)	**78.0** ± 2.0 / **0.0260** ± 0.0063	**94.0** ± 2.0 / **0.1116** ± 0.0069	**87.2** ± 1.8 / 0.1207 ± 0.0035	**70.4** ± 5.2 / 0.1998 ± 0.0207	**84.8** ± 3.3 / **0.1373** ± 0.0064
IFOSN	DiffCFG (wacv'24)	32.0 ± 8.0 / **0.1086** ± 0.0043	2.4 ± 0.9 / 0.1113 ± 0.0125	8.4 ± 2.9 / 0.1577 ± 0.0528	5.6 ± 0.9 / **0.1322** ± 0.0085	0.0 ± 0.0 / –
	AdvFor (TIFS'26)	29.6 ± 1.7 / 0.1587 ± 0.0079	44.8 ± 1.1 / 0.1707 ± 0.0027	33.7 ± 2.9 / 0.1332 ± 0.0063	5.2 ± 1.1 / 0.2321 ± 0.0240	19.2 ± 1.8 / **0.1878** ± 0.0090
	RLGC (AAAI'25)	56.8 ± 1.1 / 0.2759 ± 0.0122	69.6 ± 0.9 / 0.2751 ± 0.0132	92.6 ± 2.9 / 0.2903 ± 0.0171	57.2 ± 1.8 / 0.4006 ± 0.0176	30.8 ± 2.3 / 0.4155 ± 0.0312
	Square (ECCV'20)	66.8 ± 1.8 / 0.1934 ± 0.0021	56.4 ± 7.3 / 0.2406 ± 0.0133	38.9 ± 6.0 / 0.0980 ± 0.0117	11.2 ± 2.7 / 0.1331 ± 0.0265	73.6 ± 1.7 / 0.2828 ± 0.0047
	TALA (Ours)	**88.8** ± 2.3 / 0.1143 ± 0.0077	**93.2** ± 4.1 / **0.1047** ± 0.0151	**96.8** ± 4.7 / **0.0713** ± 0.0121	**66.8** ± 4.6 / 0.2052 ± 0.0036	**94.8** ± 3.9 / 0.2366 ± 0.0111

DiffCFG, AdvFor, RLGC: no directly adjustable ℓ_∞ or ℓ_2 perturbation budget in their original attack design. Square, TALA: a directly adjustable norm budget is part of the original design. Bold and underlined values indicate the best and second-best mean, respectively.

5 Experiments

The following experiments first evaluate whether TALA provides a favorable trade-off between attack effectiveness and perceptual quality compared with existing baselines, and then examine how its key components contribute to this performance. We further conduct diagnostic experiments to assess whether generic global image degradations can achieve comparable attack effectiveness.

5.1 Main Experimental Setup

We evaluate TALA on CASIAv1 (Dong et al., 2013), COCOGlide (Guillaro et al., 2023), COVERAGE (Wen et al., 2016), IMD2020 (Novozámskỳ et al., 2020), and FaceApp (FaceApp Limited, 2017), using released checkpoints of FOCAL (Wu et al., 2025), TruFor (Guillaro et al., 2023), and IFOSN (Wu et al., 2022). Baselines include Square (Andriushchenko et al., 2020), DiffCFG (Tailanián et al., 2024), RLGC (Mo et al., 2025), and AdvFor (Mo et al., 2026), covering general-purpose query-based search, diffusion-based counter-forensics, and surrogate-based query attacks. Other related attacks lack official implementations or could not be reliably reproduced from available code. Appendix C details baseline configurations and global-transformation diagnostics.

We evaluate images with initial localization F1 > 0.1 or > 0.5, using $Q \in \{50, 100\}$ for each criterion. An attack succeeds if it reduces localization F1 below 0.1 within the query budget. We sweep the perturbation budgets of TALA and Square, keeping other settings fixed. We use Square's ℓ_2 variant, which significantly outperforms its ℓ_∞ variant in our evaluation. Since norm constraints differ, and RLGC, AdvFor and DiffCFG lack directly comparable perturbation budget controls, we compare ASR and perceptual quality, primarily using LPIPS, rather than matching numerical norm budgets. See Appendix C.2 for budget-selection details. ASR is computed over all eligible images; average/median query counts (Avg.Q/Med.Q), LPIPS, PSNR, and post-attack F1 and IoU are computed only over successful attacks. Appendix C.1 provides further implementation details.

5.2 Comparison with Baselines

Attack effectiveness. Table 1 reports ASRs and perceptual distortion across three victim detectors and five datasets under a maximum query budget of $Q = 50$, using images with initial localization F1 > 0.5 on the corresponding victim. Following the rationale of our transferability analysis, this criterion focuses evaluation on initially well-localized images. TALA achieves the highest mean ASR in all 15 detector–dataset combinations. In every combination, it also achieves lower mean LPIPS than the baseline with the second-highest ASR. Thus, **TALA consistently improves both attack effectiveness and perceptual quality relative to its closest competitor in ASR under this setting.** For example, on TruFor, TALA achieves a mean ASR of 82.9% across the five datasets,

exceeding the strongest baseline by mean ASR, RLGC (52.7%), by 30.2 percentage points. On FOCAL–COVERAGE, TALA increases ASR from Square's 28.0% to 91.2% while reducing LPIPS from 0.1156 to 0.1118. The strongest baseline varies across detector–dataset combinations, whereas TALA maintains the highest mean ASR throughout. These results support the effectiveness of adapting directly to victim feedback without requiring a surrogate detector. Appendix C.3 reports additional results for initial F1 > 0.5 with $Q = 100$ and initial F1 > 0.1 with $Q \in \{50, 100\}$, together with detailed query counts, perceptual quality metrics, and post-attack localization scores. Across all 60 detector–dataset–threshold–budget combinations, TALA achieves the highest mean ASR in 58 and ties for the highest in one additional case. In 51 combinations, it simultaneously achieves higher mean ASR, lower mean LPIPS, and higher mean PSNR than the strongest baseline in ASR.

Perceptual quality. To examine performance beyond the configurations in Table 1, Fig. 3 compares ASR and LPIPS on CASIAv1 under $Q = 50$ and initial F1 > 0.5. We vary the perturbation budgets of TALA and Square, while evaluating DiffCFG, AdvFor, and RLGC at their standard configurations. Each point represents an evaluated configuration. Across all three victims, TALA achieves higher ASR at comparable LPIPS levels. Although some baselines attain lower LPIPS at particular operating points, their ASRs are substantially lower. Additional results on the challenging IMD2020 dataset (Fig. 5 in Appendix C.3) further support this trade-off: TALA's ASR generally increases rapidly as greater perceptual distortion is allowed, whereas Square shows limited improvements or occasional declines across the evaluated configurations. These results highlight TALA's ability to translate increased perceptual distortion into substantially greater attack success under the same query budget, achieving a more favorable effectiveness–quality trade-off. Qualitative examples in Appendix C.6 show adversarial images, amplified perturbations, and victim predictions. TALA effectively disrupts localization while largely preserving the visual appearance of the input. Despite competitive ASR and PSNR, Square can introduce visible colored block artifacts, while RLGC may cause global color distortion and fine block-like textures. In contrast, TALA produces smoother and less visually salient perturbations.

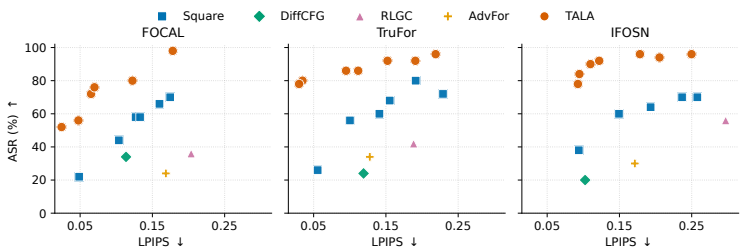


Figure 3: ASR–LPIPS comparison on CASIAv1 with $Q = 50$ and initial F1 > 0.5. Higher ASR and lower LPIPS are preferred.

Comparison with global image degradations (GIDs). We compare TALA with 11 global image degradations on FOCAL, including global noise patterns similar to those used by our perturbation engines. JPEG compression achieves the highest ASR among these transformations on all five datasets, but remains less effective than TALA while producing higher LPIPS and lower PSNR. These comparisons indicate that applying the evaluated degradations globally does not reproduce TALA's effectiveness–quality trade-off, supporting the value of its adaptive perturbation strategy. Full results and degradation settings are provided in Appendix C.4.

5.3 Ablation Studies

Victim-guided patch-center sampling. To isolate the effect of patch-center sampling, we compare our victim-guided strategy with uniform sampling over the entire image, keeping adaptive engine switching and all other components unchanged. Both variants use the same linear patch-size schedule, decreasing the patch-size ratio from $r_{\max}$ to $r_{\min}$ as queries progress. This replaces distribution-adaptive patch size and prevents differences in patch size from confounding the comparison. As shown in Table 2, at $\epsilon = 0.098$, dynamic sampling increases the mean ASR from 77.0% to 86.1%, while reducing query counts, improving perceptual quality, and lowering post-attack F1 and IoU.

Table 2: Ablation study of TALA, averaged over three victim detectors and five datasets, evaluating *Patch-center sampling*, *Patch-size adaptation*, and *Perturbation-engine selection*. *Adaptive E0+E1* denotes the final engine design.

Module	Configuration	Variant	ASR↑	Avg.Q↓	Med.Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
Patch-center sampling	–	Uniform	77.0%	24	19	0.0760	33.12	0.051	0.027
		Dynamic	**86.1%**	**20**	**13**	**0.0547**	**34.35**	**0.047**	**0.025**
Patch-size adaptation	$(r_{\min}, r_{\max}) = (0.05, 0.25)$	Linear	**85.7%**	**19**	**13**	0.0535	33.13	**0.050**	**0.029**
		Adaptive	84.6%	26	19	**0.0439**	**36.14**	0.057	0.030
	$(r_{\min}, r_{\max}) = (0.2, 0.6)$	Linear	**96.3%**	**15**	**8**	0.1218	28.30	**0.039**	**0.020**
		Adaptive	95.8%	17	11	**0.0969**	**30.57**	0.044	0.023
	$(r_{\min}, r_{\max}) = (0.1, 0.9)$	Linear	**96.8%**	**12**	**7**	0.1597	26.58	**0.035**	**0.018**
		Adaptive	96.7%	16	11	**0.1150**	**30.10**	0.041	0.021
Perturbation engine selection	–	E0 (JPEG$_{75}$)	55.6%	27	21	**0.0205**	34.81	0.059	0.031
		E0 (DiffCFG)	61.8%	21	13	0.0368	35.23	0.055	0.029
		E0 (Rademacher)	69.8%	19	12	0.0400	**35.39**	0.051	0.027
		E1 (PSNR-matched)	81.2%	13	**6**	0.0865	34.95	**0.041**	**0.021**
		E1 (default)	88.3%	**12**	7	0.1017	29.65	0.045	0.023
		Adaptive E0+E1	**96.7%**	16	11	0.1150	30.10	**0.041**	**0.021**

Distribution-adaptive patch size. We compare distribution-adaptive patch size with a linear query-dependent schedule (Appendix B.4), keeping all other components fixed. Across the ranges in Table 2, allowing larger patches generally increases mean ASR at the cost of perceptual quality. Adaptive sizing improves both LPIPS and PSNR in every range, with PSNR gains of 2.3–3.5 dB, although it requires more queries. Its ASR gap relative to the linear schedule narrows from 1.1 to 0.5 and 0.1 percentage points as $r_{\max}$ increases. We therefore adopt the broadest range as our default, which retains nearly the same ASR as the linear schedule while improving perceptual quality.

Perturbation engines: Contribution. We disable switching and evaluate single-engine variants while keeping all other settings fixed. In Table 2, *E0 (Rademacher)* and *E1 (default)* correspond to the default configurations described in Sec. 4.3. They achieve ASRs of 69.8% and 88.3%, respectively, compared with 96.7% for *Adaptive E0+E1*, which is our default TALA. Because E1's flat-top mask preserves stronger perturbations over a larger portion of the patch, equal ℓ_∞ budgets do not necessarily produce comparable image distortion. We therefore reduce E1's budget to approximately match *E0 (Rademacher)*'s average PSNR. This *E1 (PSNR-matched)* variant achieves 81.2% ASR at 34.95 dB, compared with 69.8% at 35.39 dB for *E0 (Rademacher)*. The remaining ASR advantage suggests that E1's effectiveness is not explained solely by the PSNR difference. However, its LPIPS is higher (0.0865 vs. 0.0400), motivating the retention of E0 as a complementary option with lower perceptual distortion. **Complementarity.** To further examine engine complementarity, Appendix C.5 reports a diagnostic on FOCAL–CASIAv1 at approximately matched PSNR. E0 and E1 succeed on partially overlapping image sets, with a combined success rate of 75.4% across their independent runs. Adaptive E0+E1 achieves 80.2% ASR, exceeding this independent-run union and supporting the benefit of switching engines within a single attack trajectory. **E0 reference.** We also replace E0's fixed Rademacher-perturbed reference image with a JPEG-compressed image (*E0 (JPEG$_{75}$)*) or a DiffCFG-generated image (*E0 (DiffCFG)*). JPEG is included because it is the strongest global degradation in our GID comparison. These variants achieve lower ASRs of 55.6% and 61.8%, respectively, at PSNR values close to that of *E0 (Rademacher)*. The results support the choice of a simple Rademacher-perturbed reference for E0.

6 Conclusion

We presented TALA, a surrogate-free query-based attack on image forgery localization that adapts perturbation location, size, and type through victim feedback. Experiments across three heterogeneous detectors and five datasets demonstrate strong attack effectiveness and favorable perceptual quality. The results support direct target-side adaptation as a complementary approach to transfer-based attacks and expose the vulnerability of localization detectors to perturbations guided by their own feedback. The current setting assumes a known manipulation mask and pixel-level probability feedback. Extending TALA to unknown masks or more restrictive feedback, such as binary localization masks, remains future work.

AI USE STATEMENT

We used generative AI tools to assist with code revision under author-provided specifications, language polishing of manuscript drafts, checks of mathematical notation, and the formatting of tables and figures. The authors directed the research and made all final methodological and experimental decisions. The authors conducted all experiments and collected and compiled the reported numerical results. Generative AI tools were not used to conduct experiments autonomously, fabricate experimental data, or replace empirical evaluation. The literature review was conducted primarily by the authors, with AI tools used only to assist in identifying potentially relevant literature. All AI-assisted code was reviewed and tested by the authors, and all AI-assisted text, equations, tables, and figures were reviewed by the authors. The authors take full responsibility for the final manuscript, code, and reported findings, including all content developed with generative AI assistance.

ETHICS STATEMENT

This work studies black-box adversarial attacks against image forgery detectors and therefore has potential dual-use implications. While our goal is to expose vulnerabilities in existing forensic systems and facilitate the development of more robust detectors and defenses, similar techniques could potentially be misused to help manipulated images evade forensic analysis. Our experiments are conducted solely for non-commercial academic research and do not target deployed systems.

Our evaluation includes publicly released facial manipulation benchmarks containing FaceApp-generated images. All real facial images used in our experiments are obtained from publicly available academic datasets, and we do not collect additional personal data or use private or unauthorized facial images. The manipulated images are used only as benchmark data for evaluating the robustness of forgery detectors and adversarial attacks. No human-subject experiments are conducted in this study.

We report the attack assumptions and evaluation settings to facilitate transparent and reproducible robustness assessment. We aim to reveal potential weaknesses before they are exploited in practice and to support the development of more reliable image-forensics systems.

REPRODUCIBILITY STATEMENT

The source code is included in the supplementary files accompanying this submission, including source code, configuration files, and instructions for reproducing the main experiments. Upon acceptance, we will publicly release the complete codebase and reproducibility materials in a permanent repository. The main paper specifies the threat model, attack objective, and core algorithmic components, while the appendix provides additional implementation details, hyperparameter settings, baseline configurations, and experimental protocols. We also document the datasets, victim detectors, query budgets, evaluation metrics, and ablation settings used throughout the experiments. Detailed per-dataset and per-detector results are included in the appendix to facilitate comparison and verification of the reported findings.

REFERENCES

Abdullah Al-Dujaili and Una-May O'Reilly. Sign bits are all you need for black-box attacks. In *International conference on learning representations*, 2020.

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501, 2020. doi: 10.1007/978-3-030-58592-1_29.

Gang Cao, Yuqi Wang, Haochen Zhu, Zijie Lou, and Lifang Yu. Transferable adversarial attack on image tampering localization. *Journal of Visual Communication and Image Representation*, 102: 104210, 2024.

Chengbo Dong, Xinru Chen, Ruohan Hu, Juan Cao, and Xirong Li. MVSS-Net: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3539–3553, 2022.

Jing Dong, Wei Wang, and Tieniu Tan. CASIA image tampering detection evaluation database. In *IEEE China Summit and International Conference on Signal and Information Processing*, pp. 422–426, 2013.

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*, pp. 9185–9193, 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*, pp. 4307–4316, 2019.

FaceApp Limited. FaceApp. `https://www.faceapp.com/`, 2017. Accessed: 2025-08-16.

Shengbang Fang and Matthew C Stamm. Attacking image splicing detection and localization algorithms using synthetic traces. *IEEE Transactions on Information Forensics and Security*, 19: 2143–2156, 2023.

Zhijin Ge, Hongying Liu, Wang Xiaosen, Fanhua Shang, and Yuanyuan Liu. Boosting adversarial transferability by achieving flat local maxima. *Advances in Neural Information Processing Systems*, 36:70141–70161, 2023.

Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20606–20615, 2023. doi: 10.1109/CVPR52729.2023.01974.

Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International conference on machine learning*, pp. 2484–2493, 2019.

Mark O. Hill. Diversity and evenness: A unifying notation and its consequences. *Ecology*, 54(2): 427–432, 1973. doi: 10.2307/1934352.

Myung-Joon Kwon, Seung-Hun Nam, In-Jae Yu, Heung-Kyu Lee, and Changick Kim. Learning JPEG compression artifacts for image manipulation detection and localization. *International Journal of Computer Vision*, 130(8):1875–1895, 2022.

Xianbo Mo, Shunquan Tan, Bin Li, and Jiwu Huang. Query-efficient attack for black-box image inpainting forensics via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19503–19511, 2025.

Xianbo Mo, Shunquan Tan, Rongxuan Peng, Bin Li, and Jiwu Huang. Query-efficient hard-label attacks against black-box image forgery localization model via reinforcement learning. *IEEE Transactions on Information Forensics and Security*, 21:4578–4593, 2026. doi: 10.1109/TIFS.2026.3684843.

Adam Novozámskỳ, Babak Mahdian, and Stanislav Saic. IMD2020: A large-scale annotated dataset tailored for detecting manipulated images. In *IEEE Winter Applications of Computer Vision Workshops*, pp. 71–80, 2020.

Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328, 2017.

Matías Tailanián, Marina Gardella, Alvaro Pardo, and Pablo Musé. Diffusion models meet image counter-forensics. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3925–3935, 2024.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018. doi: 10.1017/9781108231596.

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*, pp. 1924–1933, 2021.

Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. COVERAGE: A novel database for copy-move forgery detection. In *IEEE International Conference on Image Processing*, pp. 161–165, 2016.

Haiwei Wu, Jiantao Zhou, Jinyu Tian, Jun Liu, and Yu Qiao. Robust image forgery detection against transmission over online social networks. *IEEE Transactions on Information Forensics and Security*, 17:443–456, 2022. doi: 10.1109/TIFS.2022.3144878.

Haiwei Wu, Yiming Chen, Jiantao Zhou, and Yuanman Li. Rethinking Image Forgery Detection via Soft Contrastive Learning and Unsupervised Clustering . *IEEE Transactions on Dependable and Secure Computing*, 22(06):6296–6308, 2025. ISSN 1941-0018. doi: 10.1109/TDSC.2025.3583167.

Zhipei Xu, Xuanyu Zhang, Runyi Li, Zecheng Tang, Qing Huang, and Jian Zhang. FakeShield: Explainable image forgery detection and localization via multi-modal large language models. In *International Conference on Learning Representations*, volume 2025, pp. 31186–31216, 2025.

Yijia Zeng and Chi-Man Pun. Frequency-constrained transferable adversarial attack on image manipulation detection and localization. *The Visual Computer*, 40(7):4817–4828, 2024.

Peifei Zhu, Genki Osada, Hirokatsu Kataoka, and Tsubasa Takahashi. Frequency-aware GAN for adversarial manipulation generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4292–4301, 2023.

Long Zhuo, Shenghai Luo, Shunquan Tan, Han Chen, Bin Li, and Jiwu Huang. Evading detection actively: Toward anti-forensics against forgery localization. *IEEE Transactions on Dependable and Secure Computing*, 22(2):852–869, 2025.

APPENDIX

The appendix is organized as follows:

- Section A: **Cross-Detector Transferability Analysis.**
  - Section A.1: Experimental Setup.
  - Section A.2: Complete Transferability Results.
- Section B: **Additional Method Details of TALA.**
  - Section B.1: Discrete versus Continuous Acceptance.
  - Section B.2: Spatial Sampling and Failure Penalties.
  - Section B.3: Distribution-Adaptive Patch Size.
  - Section B.4: Linear Patch-Size Baseline.
  - Section B.5: Perturbation Engine Constructions.
- Section C: **Additional Experimental Details and Results.**
  - Section C.1: Implementation Details and Hyperparameter Settings.
  - Section C.2: Baseline Configurations.
  - Section C.3: Additional Baseline Comparisons.
  - Section C.4: Global Image Degradation Details.
  - Section C.5: Additional Ablation Results on Complementary Perturbation Engines.
  - Section C.6: Qualitative Results and Visualizations.

A Cross-Detector Transferability Analysis

A.1 Experimental Setup

We provide additional details and complete results for the diagnostic transferability study introduced in Sec. 3. Experiments are conducted on CASIAv1 (Dong et al., 2013) using three heterogeneous forgery localization detectors: TruFor, FOCAL, and IFOSN. All adversarial perturbations are constrained by an ℓ_∞ budget of $\epsilon = 5/255 \approx 0.020$. For each surrogate, adversarial examples are generated with full white-box access and then evaluated on all three detectors. The other two detectors are neither queried nor used to adapt the perturbations during attack generation.

Using TruFor as the surrogate, we first evaluate MI-FGSM (Dong et al., 2018), TI-MI-FGSM (Dong et al., 2019), VMI-FGSM (Wang & He, 2021), PGN (Ge et al., 2023), and AITL (Cao et al., 2024). The first four are general-purpose adversarial attacks commonly used to improve transferability, whereas AITL is specifically designed for image forgery localization. AITL serves as a representative forgery-localization-specific transfer attack discussed in Sec. 2.2. Other methods in this category are omitted because their implementations are not publicly available or we could not successfully reproduce them in our experimental environment. Among the attacks evaluated with TruFor, PGN achieves the lowest post-attack F1 and IoU on both transfer targets, followed by TI-MI-FGSM and MI-FGSM. We therefore evaluate these three attacks with FOCAL and IFOSN as additional surrogates.

For reference, we also include clean images and three non-adversarial degradations: JPEG compression, Gaussian blur, and Gaussian noise. These degradations are not constrained by the same ℓ_∞ budget and are included as reference transformations rather than budget-matched attacks.

We consider two initial-F1 filtering conditions. The first uses images with initial F1 > 0 on the surrogate for a broader comparison of the evaluated attacks. The second evaluates PGN on images with initial F1 > 0.5 on the surrogate, focusing on images that it initially localizes well.

A.2 Complete Transferability Results

Table 3 reports post-attack F1 and IoU, where lower values indicate stronger attacks. Italic values denote evaluations on the surrogate itself. Under the initial F1 > 0 condition, PGN achieves the lowest F1 and IoU on both transfer targets within every surrogate group. Using TruFor as the surrogate, PGN obtains an average transfer F1 of 0.0930 and IoU of 0.0624, compared with 0.3918 and 0.3421 for TI-MI-FGSM. PGN also achieves the best average transfer results with FOCAL and IFOSN as surrogates, obtaining average F1 scores of 0.1307 and 0.1447, respectively. Strong performance on the surrogate alone, however, does not ensure strong transfer. MI-FGSM achieves the lowest source-model F1 on TruFor (0.0045), but its F1 scores on FOCAL and IFOSN remain 0.5520 and 0.3610, giving an average transfer F1 of 0.4565. PGN has a higher source-model F1 (0.0483), yet transfers more effectively to both targets. This comparison illustrates the distinction between optimizing an attack against its source model and obtaining perturbations that transfer across detectors.

Under the initial F1 > 0.5 condition, PGN's effectiveness continues to vary across surrogate–target pairs. With TruFor as the surrogate, the average transfer F1 is 0.1818, but this average combines different outcomes: F1 reaches 0.0662 on IFOSN and 0.2975 on FOCAL. The paired clean results show that these scores decrease from 0.5024 and 0.5767, respectively. PGN therefore degrades both targets, while transferring more effectively to IFOSN on this subset. With FOCAL as the surrogate, PGN achieves a source-model F1 of 0.0712 and an average transfer F1 of 0.2504, with target scores of 0.3186 on TruFor and 0.1821 on IFOSN. The gap is particularly clear with IFOSN as the surrogate. PGN reduces IFOSN's F1 from 0.8281 to 0.1013, whereas TruFor's F1 decreases from 0.7370 to 0.3998 and FOCAL's from 0.6404 to 0.3892. The resulting average transfer F1 is 0.3945. Thus, the attack substantially degrades the surrogate while leaving more localization performance on the transfer targets. Because the selected subsets differ across surrogate groups, differences between their transfer averages reflect both the detector pairs and the evaluated images.

Overall, PGN demonstrates that strong cross-detector transfer is possible, but its effectiveness varies across detector pairs and evaluation subsets. Even the strongest evaluated transfer method can leave substantial localization performance on some targets despite strongly degrading its surrogate. These

Table 3: Cross-detector transferability on CASIAv1 under two initial-F1 filtering conditions. Transfer attacks are generated on the surrogate under an ℓ_∞ budget of $\epsilon \approx 0.020$ and evaluated on the other detectors without victim-side adaptation. Non-adversarial degradations are reference transformations and are not constrained by the same ℓ_∞ budget. For transfer attacks and their paired Clean rows in (b), Average is computed over the two non-surrogate target detectors. For Clean and reference degradations in (a), Average is computed over all three detectors. In (b), Clean and PGN use the same images within each surrogate group, selected with initial F1 > 0.5 on that surrogate. Italic values denote attacks evaluated on the surrogate itself. In (b), post-attack F1 values above 0.1 and their corresponding IoU values are shaded in light red.

			TruFor		FOCAL		IFOSN		Average	
Type	Surrogate	Method	F1↓	IoU↓	F1↓	IoU↓	F1↓	IoU↓	F1↓	IoU↓
(a) Initial localization F1 > 0										
Reference	–	Clean	0.6931	0.6291	0.8667	0.7291	0.5091	0.4653	0.6896	0.6078
Non-adversarial degradations	–	JPEG_{70}	0.6041	0.5360	0.5259	0.4523	0.3234	0.2833	0.4845	0.4239
	–	Gaussian Blur	0.6625	0.6000	0.6337	0.5657	0.5104	0.4647	0.6022	0.5435
	–	Gaussian Noise	0.5870	0.5245	0.6047	0.5347	0.4750	0.4340	0.5556	0.4977
Transfer attacks	TruFor	VMI-FGSM (Wang & He, 2021)	*0.5424*	*0.4829*	0.6213	0.5508	0.5048	0.4602	0.5631	0.5055
		AITL (Cao et al., 2024)	*0.0747*	*0.0398*	0.5682	0.5015	0.4217	0.3777	0.4950	0.4396
		MI-FGSM (Dong et al., 2018)	***0.0045***	***0.0023***	0.5520	0.4847	0.3610	0.3265	0.4565	0.4056
		TI-MI-FGSM (Dong et al., 2019)	*0.0303*	*0.0178*	0.4887	0.4206	0.2948	0.2636	0.3918	0.3421
		PGN (Ge et al., 2023)	*0.0483*	*0.0270*	**0.1605**	**0.1101**	**0.0255**	**0.0146**	**0.0930**	**0.0624**
	FOCAL	TI-MI-FGSM	0.5416	0.4802	*0.1186*	*0.0723*	0.4080	0.3670	0.4748	0.4236
		MI-FGSM	0.5357	0.4759	*0.1139*	*0.0684*	0.4070	0.3662	0.4713	0.4211
		PGN	**0.1693**	**0.1346**	***0.0383***	***0.0221***	**0.0921**	**0.0750**	**0.1307**	**0.1048**
	IFOSN	TI-MI-FGSM	0.4443	0.3827	0.4286	0.3618	*0.0907*	*0.0574*	0.4365	0.3722
		MI-FGSM	0.3630	0.3131	0.4345	0.3699	***0.0740***	*0.0449*	0.3987	0.3415
		PGN	**0.1531**	**0.1101**	**0.1364**	**0.0872**	*0.0748*	***0.0420***	**0.1447**	**0.0987**
(b) Initial localization F1 > 0.5										
Reference	TruFor	Clean	0.8424	0.7540	0.5767	0.4867	0.5024	0.4385	0.5396	0.4626
Transfer attack		PGN	*0.0816*	*0.0466*	0.2975	0.2275	0.0662	0.0394	0.1818	0.1335
Reference	FOCAL	Clean	0.6090	0.5459	0.7844	0.6706	0.4587	0.3833	0.5339	0.4646
Transfer attack		PGN	0.3186	0.2711	*0.0712*	*0.0419*	0.1821	0.1458	0.2504	0.2085
Reference	IFOSN	Clean	0.7370	0.6732	0.6404	0.5581	0.8281	0.7361	0.6887	0.6157
Transfer attack		PGN	0.3998	0.3382	0.3892	0.3270	*0.1013*	*0.0573*	0.3945	0.3326

results motivate TALA's use of direct victim feedback to adapt perturbations to the current detector and input.

Additional attribution analysis. To further examine the forensic feature misalignment discussed in Sec. 3, Fig. 4 provides additional Integrated Gradients (IG) visualizations for forged images sampled from different datasets. Different detectors can assign importance to different regions of the same image, and the attribution patterns of a given detector also vary across images. These examples support the model- and input-dependent forensic evidence discussed in Sec. 3. They offer a possible explanation for variable transferability, but do not by themselves establish its cause.

B Additional Method Details of TALA

B.1 Discrete versus continuous acceptance

We compare the discrete F1-based acceptance criterion used in the main method with a continuous soft-F1 criterion, while retaining discrete F1 in Eq. 1 to determine attack success. For an evaluated image $\boldsymbol{x}$, let $\boldsymbol{P} = f(\boldsymbol{x}) \in [0,1]^{H\times W}$ denote the victim's predicted manipulation probabilities. The soft-F1 objective is

$$\mathcal{L}_{\text{soft}}(\boldsymbol{x}) = \frac{2\sum_{(i,j)\in\Omega} P(i,j)y(i,j)}{\sum_{(i,j)\in\Omega} P(i,j) + \sum_{(i,j)\in\Omega} y(i,j) + 10^{-8}}. \tag{9}$$

Here Ω denotes the image lattice and $\boldsymbol{y}$ is the binary manipulation mask. Unlike discrete F1, this objective uses the predicted probabilities directly without thresholding. A candidate is accepted if

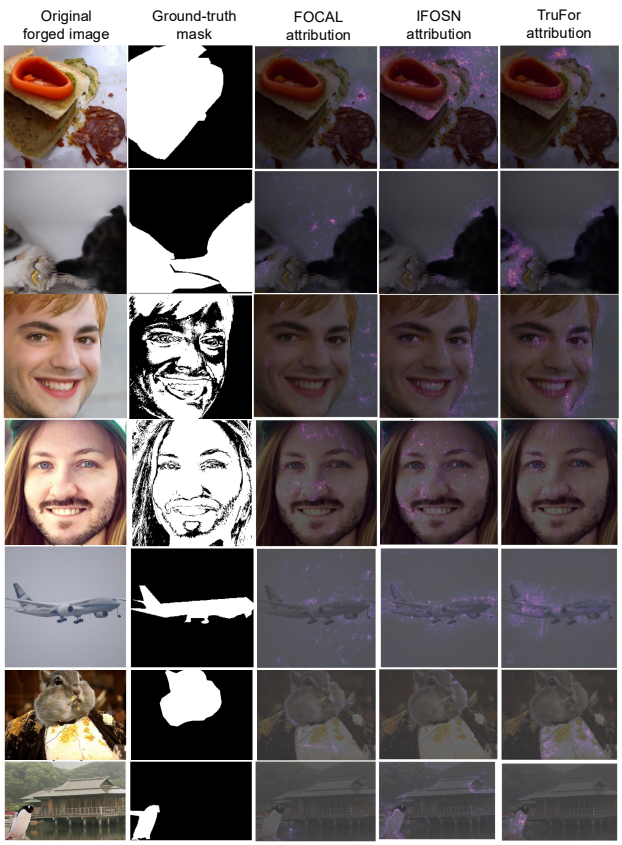


Figure 4: Additional examples of forensic feature misalignment across heterogeneous localization detectors. Integrated Gradients (IG) visualizations of forged images sampled from different datasets show that FOCAL, IFOSN, and TruFor rely on different forensic evidence for the same input. Their attribution patterns also vary across inputs, providing further examples of both model- and input-dependent forensic evidence.

it strictly decreases $\mathcal{L}_{\text{soft}}$. Candidates that satisfy the discrete attack-success criterion are always accepted and terminate the attack.

As shown in Table 4, the two criteria yield comparable aggregate performance: mean ASR is 89.1% with discrete F1 and 89.4% with soft F1. Soft F1 slightly reduces average query counts and perceptual distortion, but its ASR gains are not consistent across settings, with six improvements, six decreases, and three ties. We therefore retain discrete F1 as the default criterion: it directly matches the evaluation metric and can be computed from binary localization masks without probability scores, facilitating future extension to hard-label feedback.

B.2 Spatial Sampling and Failure Penalties

State-dependent weights W_t^{guide}. Let $m_t = \text{median}\{P_t(i,j) : (i,j) \in \mathbb{T}_t\}$, where $\mathbb{T}_t = \{(i,j) \in \Omega : y(i,j) = 1,\ P_t(i,j) > 0.5\}$ denotes the true-positive set. Since the ground-truth mask is nonempty, $\mathbb{T}_t = \varnothing$ implies zero F1. The algorithm stops before computing weights for an empty set. The normalized distance from the median is

$$d_t(i,j) = \frac{|P_t(i,j) - m_t|}{\max_{(\bar{i},\bar{j})\in\mathbb{T}_t} |P_t(\bar{i},\bar{j}) - m_t| + \delta}, \qquad (i,j) \in \mathbb{T}_t, \tag{10}$$

Table 4: Ablation of the acceptance criterion on images with initial localization F1 > 0.5. Within each detector–dataset pair, the first row uses discrete F1 and the second uses soft F1 for step acceptance. Both variants use discrete F1 to determine attack success. All other settings are fixed, with $(r_{\min}, r_{\max}) = (0.1, 0.9)$, $\ell_{\min} = 15$, and perturbation budget $\epsilon = 0.118$. Mean denotes the unweighted average of each reported metric across the 15 settings.

Victim	Dataset	Criterion	ASR (%)↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
IFOSN	CASIAv1	Discrete F1	90.0	16.0	15.0	**0.1099**	**29.88**	**0.018**	**0.009**
		Soft F1	**96.0**	**14.5**	**13.0**	0.1115	29.71	0.027	0.014
	COCOGlide	Discrete F1	**94.0**	8.7	**4.0**	**0.0839**	**30.84**	0.034	0.017
		Soft F1	**94.0**	**8.2**	**4.0**	0.0974	30.53	**0.031**	**0.016**
	COVERAGE	Discrete F1	**100.0**	9.4	5.0	0.0715	32.19	0.026	0.013
		Soft F1	94.7	**7.6**	**4.0**	**0.0609**	**32.51**	**0.023**	**0.012**
	IMD2020	Discrete F1	72.0	**19.9**	20.5	**0.1798**	**27.45**	**0.039**	**0.020**
		Soft F1	**76.0**	20.2	**17.5**	0.1970	27.44	0.044	0.023
	FaceApp	Discrete F1	88.0	16.9	14.0	0.2271	26.29	0.032	**0.016**
		Soft F1	**96.0**	**15.5**	**12.5**	**0.2236**	**26.66**	**0.031**	**0.016**
TruFor	CASIAv1	Discrete F1	86.0	**22.4**	**23.0**	**0.0949**	**30.75**	0.049	0.026
		Soft F1	**94.0**	22.6	24.0	0.1000	30.30	**0.037**	**0.019**
	COCOGlide	Discrete F1	**98.0**	**11.5**	8.0	**0.1180**	29.98	**0.034**	**0.017**
		Soft F1	**98.0**	11.8	**7.0**	0.1240	**30.03**	**0.034**	**0.017**
	COVERAGE	Discrete F1	**100.0**	13.5	11.0	0.1257	29.04	**0.017**	**0.009**
		Soft F1	96.0	**12.4**	**9.0**	**0.1097**	**30.41**	0.036	0.019
	IMD2020	Discrete F1	**84.0**	29.1	**30.0**	**0.2122**	26.25	0.045	0.023
		Soft F1	78.0	**27.7**	31.0	0.2227	**26.48**	**0.041**	**0.021**
	FaceApp	Discrete F1	**100.0**	12.4	9.5	0.1598	27.39	0.066	0.034
		Soft F1	**100.0**	**10.6**	**9.0**	**0.1371**	**28.26**	**0.061**	**0.032**
FOCAL	CASIAv1	Discrete F1	80.0	16.0	14.5	**0.1224**	28.52	0.035	0.018
		Soft F1	**84.0**	**15.8**	**14.0**	0.1257	**28.86**	**0.028**	**0.014**
	COCOGlide	Discrete F1	**74.0**	20.8	17.0	0.1471	27.28	0.043	0.022
		Soft F1	68.0	**17.4**	**16.5**	**0.1301**	**28.11**	**0.040**	**0.021**
	COVERAGE	Discrete F1	**92.0**	**12.4**	8.5	0.1195	29.35	**0.026**	**0.013**
		Soft F1	88.0	12.5	**7.0**	**0.1067**	**30.17**	0.030	0.016
	IMD2020	Discrete F1	**80.0**	**20.9**	20.0	**0.2102**	**26.63**	0.049	0.026
		Soft F1	78.0	22.0	**19.0**	0.2230	26.08	**0.045**	**0.023**
	FaceApp	Discrete F1	98.0	**10.7**	**10.0**	**0.1248**	**29.19**	**0.060**	**0.031**
		Soft F1	**100.0**	12.2	**10.0**	0.1363	28.64	0.064	0.034
Mean		Discrete F1	89.1	16.0	14.0	0.1405	28.74	**0.038**	**0.020**
		Soft F1	**89.4**	**15.4**	**13.2**	**0.1404**	**28.95**	**0.038**	**0.020**

where $\delta = 10^{-8}$ is a numerical stabilizer, and $\bar{i}, \bar{j}$ are dummy indices. The three unnormalized weight maps are

$$\begin{aligned} W_t^{\mathrm{med}}(i,j) &= \begin{cases} (1 - d_t(i,j) + \beta)^{\gamma}, & (i,j) \in \mathbb{T}_t, \\ 0, & (i,j) \notin \mathbb{T}_t, \end{cases} \\ W_t^{\max}(i,j) &= \mathbb{1}[(i,j) \in \mathbb{T}_t] P_t(i,j)^{\gamma}, \\ W_t^{\mathrm{bnd}}(i,j) &= \left(1 - |2P_t(i,j) - 1| + 0.1\right)^2, \qquad (i,j) \in \Omega. \end{aligned} \tag{11}$$

We set $\beta = 0.1$ and $\gamma = 3$. The offset β prevents locations in the true-positive set from receiving vanishingly small weights, while the exponent γ sharpens the preference toward responses close to the median confidence without enforcing a hard selection. *Median* adapts its target to the confidence distribution, *Max* emphasizes strong remaining responses, and *Boundary* extends the preference to uncertain predictions across the image. The active map is

$$\boldsymbol{W}_t^{\mathrm{guide}} = \begin{cases} \boldsymbol{W}_t^{\mathrm{med}}, & s_t = \mathrm{Median}, \\ \boldsymbol{W}_t^{\max}, & s_t = \mathrm{Max}, \\ \boldsymbol{W}_t^{\mathrm{bnd}}, & s_t = \mathrm{Boundary}. \end{cases} \tag{12}$$

Spatial penalties. We construct the Gaussian penalty matrix $\boldsymbol{G}_t$ used to update the spatial penalty map after a rejected perturbation. For the sampled patch center $\boldsymbol{c}_t = (c_t^y, c_t^x)$, its entries and spatial scales are defined as

$$\begin{aligned} G_t(i,j) &= \exp\left[-\frac{(i - c_t^y)^2}{2\sigma_{y,t}^2} - \frac{(j - c_t^x)^2}{2\sigma_{x,t}^2} \right], \\ \sigma_{y,t} &= \max(h_t/2.5, 1), \qquad \sigma_{x,t} = \max(w_t/2.5, 1). \end{aligned} \tag{13}$$

Here h_t and w_t are the nominal patch dimensions defined in Sec. 4.2. The update in Eq. 8 decays existing penalties at every iteration and adds $\boldsymbol{G}_t$ only upon rejection. The resulting map is clipped element-wise to $[0, H_{\max}]$, with $H_{\max} = 5$. Thus, $\boldsymbol{H}_t$ guides sampling of the current proposal, while the updated map $\boldsymbol{H}_{t+1}$ guides the next. Acceptance introduces no new penalty but still applies decay.

B.3 Distribution-Adaptive Patch Size

We provide additional implementation details for the distribution-adaptive patch sizing described in Sec. 4.2. For each axis $d \in \{y, x\}$, the allowable patch dimensions are bounded by

$$\begin{aligned} b_d^- &= \min\left\{L_d, \max\left(1, \ell_{\min}, \lfloor r_{\min} L_d \rfloor\right)\right\}, \\ b_d^+ &= \min\left\{L_d, \max\left(b_d^-, \lfloor r_{\max} L_d \rfloor\right)\right\}, \end{aligned} \tag{14}$$

where $L_y = H$ and $L_x = W$. Here, $r_{\min}$ and $r_{\max}$ specify the minimum and maximum patch side lengths relative to the corresponding image dimension, while $\ell_{\min}$ specifies an absolute minimum side length in pixels. Together with the normalized spatial support u_t^d defined in Eq. 5, these bounds determine the nominal patch height and width $b_{t,d}$ according to Eq. 6. Unless otherwise specified, we set $\ell_{\min} = 15$ pixels, $r_{\min} = 0.1$, and $r_{\max} = 0.9$. The uniform exploration component is used only for center sampling and is excluded from the patch-size calculation.

Patch construction and boundary handling. Following Eq. 6, we define the nominal patch height and width as $h_t = b_{t,y}$ and $w_t = b_{t,x}$, respectively. Given the sampled center $\boldsymbol{c}_t = (c_t^y, c_t^x)$ and these nominal dimensions, we construct the patch region as

$$\mathcal{R}_t = [y_t^-, y_t^+) \times [x_t^-, x_t^+), \tag{15}$$

where the intervals contain integer pixel coordinates and

$$\begin{aligned} y_t^- &= \max(0, c_t^y - \lfloor h_t/2 \rfloor), \quad y_t^+ = \min(H, c_t^y + \lceil h_t/2 \rceil), \\ x_t^- &= \max(0, c_t^x - \lfloor w_t/2 \rfloor), \quad x_t^+ = \min(W, c_t^x + \lceil w_t/2 \rceil). \end{aligned} \tag{16}$$

The floor/ceiling convention preserves the nominal dimensions, including odd sizes, away from image boundaries. After clipping, the actual dimensions are $\bar{h}_t = y_t^+ - y_t^-$ and $\bar{w}_t = x_t^+ - x_t^-$. Both perturbation engines use these actual dimensions to construct their local fields and masks, whereas the Gaussian penalty scales use the nominal dimensions (h_t, w_t). The same patch construction is used for the linear baseline described below.

B.4 Linear Patch-Size Baseline

To evaluate the benefit of adapting patch dimensions to the current guided distribution, we compare our design with a predefined query-dependent schedule in the ablation study. This baseline uses the same patch-center sampling strategy but determines the patch size solely from query progress, decreasing it linearly as the query budget is consumed.

Using the same notation as Appendix B.3, the nominal patch dimensions are

$$\begin{aligned} a_t &= r_{\max}(1 - t/Q) + r_{\min} t/Q, \\ b_{t,d} &= \min\left\{L_d,\ \max\left(1, \ell_{\min}, \lfloor a_t L_d \rfloor\right)\right\}, \qquad d \in \{y, x\}, \end{aligned} \tag{17}$$

where $L_y = H$, $L_x = W$, $h_t = b_{t,y}$, and $w_t = b_{t,x}$. The parameters $r_{\min}$, $r_{\max}$, and $\ell_{\min}$ have the same meanings as in the adaptive variant. Unlike distribution-adaptive sizing, this schedule depends only on query progress. The resulting dimensions (h_t, w_t) define $\mathcal{R}_t$ using the same boundary handling as the adaptive variant (Eqs. 15–16).

B.5 Perturbation Engine Constructions

Notation. For the selected region $\mathcal{R}_t$, we define the element-wise feasible projection

$$\Pi_{\mathcal{R}_t}(\mathbf{a}) = \mathrm{clip}_{[\max(\mathbf{0}, \boldsymbol{x}_{\mathcal{R}_t} - \epsilon),\ \min(\mathbf{1}, \boldsymbol{x}_{\mathcal{R}_t} + \epsilon)]}(\mathbf{a}), \tag{18}$$

where $\mathbf{a}$ denotes a candidate image patch within $\mathcal{R}_t$. The projection constrains each pixel to the valid image range $[0, 1]$ and the ℓ_∞ neighborhood of the corresponding original patch $\boldsymbol{x}_{\mathcal{R}_t}$. As detailed below, each engine first generates a candidate patch within $\mathcal{R}_t$, applies this projection, and then combines the projected patch with its spatial mask to obtain the perturbed patch.

Engine 0: smooth perturbation. At the beginning of each attack, we sample a fixed bounded reference image $\boldsymbol{x}^B$:

$$\boldsymbol{x}^B = \mathrm{clip}_{[0,1]}(\boldsymbol{x} + \boldsymbol{\delta}^B), \qquad \delta^B_{cij} = \epsilon s_{cij}, \quad s_{cij} \overset{\text{iid}}{\sim} \mathrm{Rad}(1/2), \tag{19}$$

where c, i, and j index the channel, row, and column, respectively, and $\mathrm{Rad}(1/2)$ takes -1 and $+1$ with equal probability. Thus, s_{cij} is an independent random sign and $\delta^B_{cij} \in \{-\epsilon, +\epsilon\}$ is the corresponding pixel-wise perturbation added to the original image $\boldsymbol{x}$. The resulting $\boldsymbol{x}^B$ is clipped to the valid image range and remains fixed throughout the attack.

For each Engine 0 perturbation, we additionally sample an independent low-resolution random field:

$$\begin{aligned} \boldsymbol{U}_t &\in \{-\epsilon, +\epsilon\}^{C \times h'_t \times w'_t}, \\ h'_t &= \max(4, \lfloor \bar{h}_t/4 \rfloor), \qquad w'_t = \max(4, \lfloor \bar{w}_t/4 \rfloor), \\ \boldsymbol{\delta}^{\text{smooth}}_t &= \mathrm{Bicubic}(\boldsymbol{U}_t; \bar{h}_t, \bar{w}_t). \end{aligned} \tag{20}$$

Here, $\bar{h}_t$ and $\bar{w}_t$ denote the actual height and width of the selected region $\mathcal{R}_t$ after clipping it to the image boundaries, while h'_t and w'_t are the corresponding low-resolution dimensions. The entries of $\boldsymbol{U}_t$ are independently sampled from $\{-\epsilon, +\epsilon\}$ with equal probability, and bicubic interpolation upsamples this field to the actual patch size $\bar{h}_t \times \bar{w}_t$.

We combine the perturbed reference patch with the smooth random field and project the result onto the feasible set:

$$\hat{\boldsymbol{x}}^{(0)}_{t,\mathcal{R}_t} = \Pi_{\mathcal{R}_t}\left(\alpha \boldsymbol{x}^B_{\mathcal{R}_t} + (1-\alpha)\boldsymbol{\delta}^{\text{smooth}}_t\right), \qquad \alpha = 0.8. \tag{21}$$

The weighted combination forms a candidate image patch, and projection ensures that it satisfies the perturbation budget and valid pixel range.

For normalized patch coordinates $(u, v) \in [-1, 1]^2$, the spatial mask is

$$M^{(0)}_t(u, v) = \exp[-3(u^2 + v^2)^2]. \tag{22}$$

The mask progressively attenuates the modification away from the patch center without forcing it to vanish at the boundary. Combining this mask with the projected candidate through Eq. 7 yields the perturbed patch.

Engine 1: high-frequency perturbation. We first sample an independent full-resolution random field over the selected patch, add it to the original image patch, and project the result onto the feasible set:

$$\begin{aligned} \delta^{\text{HF}}_{t,cij} &= \epsilon s_{t,cij}, \qquad s_{t,cij} \overset{\text{iid}}{\sim} \mathrm{Rad}(1/2), \\ \hat{\boldsymbol{x}}^{(1)}_{t,\mathcal{R}_t} &= \Pi_{\mathcal{R}_t}(\boldsymbol{x}_{\mathcal{R}_t} + \boldsymbol{\delta}^{\text{HF}}_t). \end{aligned} \tag{23}$$

We then apply a flat-top spatial mask that preserves the projected perturbation in the patch interior and attenuates only its outer band. With $d_\infty(u, v) = \max(|u|, |v|)$, the mask is defined as

$$M^{(1)}_t(u, v) = \begin{cases} 1, & d_\infty(u, v) < 0.9, \\ \dfrac{1 + \cos\left(\pi \dfrac{d_\infty(u, v) - 0.9}{0.1}\right)}{2}, & 0.9 \le d_\infty(u, v) \le 1. \end{cases} \tag{24}$$

Combining this mask with the projected candidate through Eq. 7 yields the perturbed patch. For a spatial axis of length one, we set its normalized coordinate to zero.

C Additional Experimental Details and Results

C.1 Implementation Details and Hyperparameter Settings

Unless otherwise specified, we use the hyperparameter settings summarized in Table 5 throughout our experiments. Because relatively few images in some datasets satisfy the initial F1 > 0.5 criterion, and for evaluation efficiency, we evaluate up to 50 eligible images per detector–dataset pair, using all eligible images when fewer than 50 are available. Under the initial F1 > 0.1 criterion, we use a fixed subset of 100 images per dataset and evaluate all images within this subset whose initial F1 score exceeds 0.1 for the corresponding detector. All compared methods use the same selected images within each setting.

Table 5: Default hyperparameters of TALA.

Parameter	Symbol	Default Value
Success threshold	τ	0.1
Guided distribution weight	η	0.8
Penalty weight	λ	3
Penalty decay rate	ρ	0.9
Maximum patch size ratio	$r_{\max}$	0.9
Minimum patch size ratio	$r_{\min}$	0.1
Minimum patch size	$\ell_{\min}$	15 pixels

Table 6: Diagnostic analysis of the dual-engine interaction on FOCAL–CASIAv1 under approximately matched PSNR.

Strategy	ASR (%) ↑	Unique Success (%)
E0 only	66.1	5.1
E1 only	70.3	9.3
Independent-run union	75.4	–
Adaptive E0+E1	**80.2**	–

Table 7: Black-box attack performance against TruFor across five datasets under a maximum query budget of $Q = 50$, evaluated on images with an initial localization F1> 0.5. Results are reported as mean±standard deviation over five independent random seeds where available. Bold and underlined values indicate the best and second-best mean, respectively, for each metric within each dataset.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	27.6 ± 3.8%	–	–	0.1181 ± 0.0028	30.82 ± 0.32	**0.014** ± 0.007	**0.007** ± 0.004
	AdvFor	28.8 ± 4.1%	**3.3** ± 0.8	**1.4** ± 0.5	0.1430 ± 0.0119	33.73 ± 0.16	0.026 ± 0.004	0.014 ± 0.002
	RLGC	43.2 ± 2.3%	16.5 ± 0.5	7.7 ± 1.2	0.1866 ± 0.0101	35.17 ± 0.25	0.046 ± 0.005	0.024 ± 0.003
	Square	51.2 ± 3.0%	10.0 ± 0.7	5.8 ± 1.4	0.1028 ± 0.0018	35.06 ± 0.02	0.017 ± 0.003	0.009 ± 0.002
	TALA (Ours)	**78.0** ± 2.0%	16.2 ± 1.4	14.0 ± 1.9	**0.0260** ± 0.0063	**37.51** ± 1.01	0.048 ± 0.004	0.025 ± 0.002
COCOGlide	DiffCFG	16.4 ± 1.7%	–	–	0.1180 ± 0.0129	33.87 ± 0.39	**0.009** ± 0.007	**0.005** ± 0.004
	AdvFor	40.4 ± 2.6%	**2.6** ± 0.2	**1.3** ± 0.4	0.2460 ± 0.0115	**34.74** ± 0.28	0.026 ± 0.002	0.013 ± 0.001
	RLGC	71.2 ± 1.8%	15.1 ± 0.8	12.2 ± 0.8	0.2811 ± 0.0074	30.84 ± 0.39	0.052 ± 0.006	0.027 ± 0.003
	Square	42.8 ± 4.1%	14.1 ± 4.3	10.7 ± 6.0	0.2123 ± 0.0161	28.46 ± 0.13	0.029 ± 0.013	0.015 ± 0.007
	TALA (Ours)	**94.0** ± 2.0%	12.1 ± 1.1	7.1 ± 2.1	**0.1116** ± 0.0069	31.05 ± 0.28	0.037 ± 0.006	0.019 ± 0.003
COVERAGE	DiffCFG	19.2 ± 2.7%	–	–	0.1157 ± 0.0068	**35.32** ± 0.25	**0.017** ± 0.006	**0.009** ± 0.003
	AdvFor	20.4 ± 1.7%	**4.2** ± 1.0	**3.9** ± 0.5	0.1918 ± 0.0103	34.14 ± 0.32	0.040 ± 0.006	0.021 ± 0.003
	RLGC	40.4 ± 0.9%	21.0 ± 0.5	18.4 ± 0.8	0.3350 ± 0.0067	30.71 ± 0.26	0.056 ± 0.007	0.029 ± 0.004
	Square	37.2 ± 3.0%	14.2 ± 1.4	11.4 ± 1.9	**0.1027** ± 0.0081	30.38 ± 0.38	0.023 ± 0.005	0.012 ± 0.003
	TALA (Ours)	**87.2** ± 1.8%	22.4 ± 0.9	21.7 ± 1.9	0.1207 ± 0.0035	31.52 ± 0.16	0.029 ± 0.004	0.015 ± 0.002
IMD2020	DiffCFG	12.0 ± 4.0%	–	–	0.1656 ± 0.0036	34.79 ± 0.62	**0.018** ± 0.012	**0.009** ± 0.006
	AdvFor	4.4 ± 0.9%	**2.1** ± 1.4	**2.0** ± 1.1	0.1856 ± 0.0108	**36.86** ± 0.85	0.020 ± 0.028	0.010 ± 0.015
	RLGC	39.2 ± 1.8%	30.6 ± 1.1	27.7 ± 0.7	0.4609 ± 0.0067	27.12 ± 0.27	0.066 ± 0.006	0.035 ± 0.003
	Square	25.6 ± 2.6%	5.5 ± 1.3	4.6 ± 0.8	**0.1294** ± 0.0120	26.60 ± 0.32	0.029 ± 0.005	0.015 ± 0.003
	TALA (Ours)	**70.4** ± 5.2%	28.6 ± 0.9	28.5 ± 2.0	0.1998 ± 0.0207	27.73 ± 0.61	0.052 ± 0.005	0.027 ± 0.003
FaceApp	DiffCFG	2.8 ± 1.1%	–	–	0.1512 ± 0.0156	34.65 ± 0.91	**0.026** ± 0.033	**0.013** ± 0.017
	AdvFor	31.2 ± 2.7%	**2.8** ± 0.4	**2.0** ± 0.0	0.1621 ± 0.0101	**35.71** ± 0.29	0.055 ± 0.007	0.028 ± 0.004
	RLGC	69.6 ± 2.6%	21.3 ± 1.2	19.7 ± 1.3	0.3248 ± 0.0115	30.97 ± 0.52	0.075 ± 0.001	0.039 ± 0.001
	Square	83.6 ± 4.3%	12.0 ± 0.4	9.3 ± 0.7	0.1804 ± 0.0049	30.40 ± 0.07	0.038 ± 0.002	0.020 ± 0.001
	TALA (Ours)	**84.8** ± 3.3%	24.7 ± 1.7	24.2 ± 2.3	**0.1373** ± 0.0064	31.27 ± 0.24	0.067 ± 0.003	0.035 ± 0.002

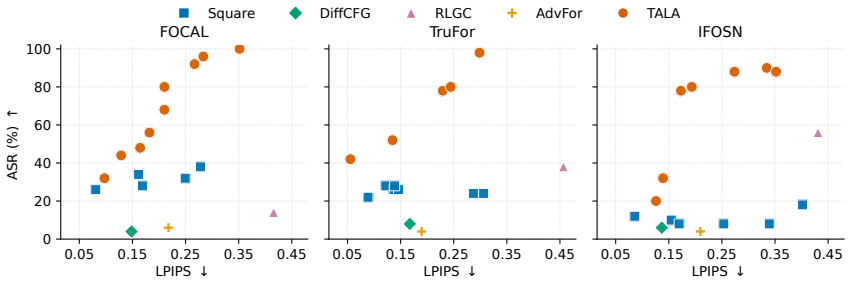


Figure 5: Attack effectiveness and perceptual quality on IMD2020 under $Q = 50$ and initial localization F1 > 0.5. Each point represents an evaluated configuration. Higher ASR and lower LPIPS are preferred.

Dataset coverage. The evaluated datasets cover copy-move and splicing (CASIAv1), diffusion-based inpainting (COCOGlide), copy-move (COVERAGE), diverse real-world manipulations (IMD2020), and facial edits generated with FaceApp.

Table 8: Black-box attack performance against IFOSN across different datasets under a maximum query budget of $Q = 50$, evaluated on images with initial F1> 0.5.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	32.0 ± 8.0%	–	–	**0.1086** ± 0.0043	31.04 ± 0.24	**0.010** ± 0.005	**0.005** ± 0.003
	AdvFor	29.6 ± 1.7%	**2.9** ± 0.5	**2.1** ± 0.2	0.1587 ± 0.0079	**32.36** ± 0.39	0.022 ± 0.007	0.011 ± 0.004
	RLGC	56.8 ± 1.1%	24.5 ± 0.2	25.8 ± 0.4	0.2759 ± 0.0122	26.94 ± 0.23	0.058 ± 0.007	0.030 ± 0.004
	Square	66.8 ± 1.8%	8.6 ± 1.5	3.1 ± 1.6	0.1934 ± 0.0021	27.28 ± 0.14	0.026 ± 0.002	0.013 ± 0.001
	TALA (Ours)	**88.8** ± 2.3%	16.3 ± 0.7	14.7 ± 2.2	0.1143 ± 0.0077	29.51 ± 0.34	0.027 ± 0.006	0.014 ± 0.003
COCOGlide	DiffCFG	2.4 ± 0.9%	–	–	0.1113 ± 0.0125	**35.82** ± 0.27	0.024 ± 0.020	**0.012** ± 0.011
	AdvFor	44.8 ± 1.1%	**1.9** ± 0.2	**1.0** ± 0.0	0.1707 ± 0.0027	35.31 ± 0.21	**0.023** ± 0.006	**0.012** ± 0.003
	RLGC	69.6 ± 0.9%	18.5 ± 0.5	14.8 ± 1.3	0.2751 ± 0.0132	27.50 ± 0.26	0.063 ± 0.004	0.033 ± 0.002
	Square	56.4 ± 7.3%	8.0 ± 1.8	1.4 ± 0.9	0.2406 ± 0.0133	25.60 ± 0.17	0.032 ± 0.005	0.017 ± 0.003
	TALA (Ours)	**93.2** ± 4.1%	9.0 ± 0.9	4.5 ± 1.1	**0.1047** ± 0.0151	30.33 ± 0.45	0.033 ± 0.004	0.017 ± 0.002
COVERAGE	DiffCFG	8.4 ± 2.9%	–	–	0.1577 ± 0.0528	34.99 ± 0.53	0.064 ± 0.025	0.033 ± 0.013
	AdvFor	33.7 ± 2.9%	**2.3** ± 0.7	**1.7** ± 0.4	0.1332 ± 0.0063	**36.10** ± 0.43	**0.026** ± 0.008	**0.014** ± 0.004
	RLGC	92.6 ± 2.9%	20.3 ± 0.6	20.2 ± 1.1	0.2903 ± 0.0171	30.41 ± 0.12	0.061 ± 0.002	0.032 ± 0.001
	Square	38.9 ± 6.0%	6.9 ± 3.6	3.4 ± 2.3	0.0980 ± 0.0117	29.19 ± 0.30	0.028 ± 0.013	0.015 ± 0.007
	TALA (Ours)	**96.8** ± 4.7%	8.3 ± 0.9	4.0 ± 0.7	**0.0713** ± 0.0121	32.33 ± 0.30	0.027 ± 0.004	**0.014** ± 0.002
IMD2020	DiffCFG	5.6 ± 0.9%	–	–	**0.1322** ± 0.0085	**34.91** ± 0.79	**0.013** ± 0.015	**0.007** ± 0.008
	AdvFor	5.2 ± 1.1%	**3.3** ± 1.2	**3.7** ± 1.7	0.2321 ± 0.0240	34.60 ± 0.98	0.060 ± 0.015	0.031 ± 0.008
	RLGC	57.2 ± 1.8%	25.8 ± 0.8	27.1 ± 0.8	0.4006 ± 0.0176	28.15 ± 0.25	0.073 ± 0.005	0.038 ± 0.003
	Square	11.2 ± 2.7%	20.8 ± 5.2	21.8 ± 7.2	0.1331 ± 0.0265	27.61 ± 0.67	0.060 ± 0.019	0.031 ± 0.010
	TALA (Ours)	**66.8** ± 4.6%	28.4 ± 1.7	28.9 ± 2.5	0.2052 ± 0.0036	27.80 ± 0.36	0.042 ± 0.003	0.022 ± 0.002
FaceApp	DiffCFG	0.0 ± 0.0%	–	–	–	–	–	–
	AdvFor	19.2 ± 1.8%	**2.4** ± 0.3	1.8 ± 0.4	**0.1878** ± 0.0090	**36.24** ± 0.37	0.039 ± 0.004	0.020 ± 0.002
	RLGC	30.8 ± 2.3%	31.2 ± 1.8	34.9 ± 1.7	0.4155 ± 0.0312	26.37 ± 0.57	0.075 ± 0.005	0.039 ± 0.003
	Square	73.6 ± 1.7%	7.6 ± 1.3	**1.0** ± 0.0	0.2828 ± 0.0047	25.95 ± 0.05	**0.022** ± 0.003	**0.011** ± 0.002
	TALA (Ours)	**94.8** ± 3.9%	17.2 ± 1.0	15.8 ± 2.0	0.2366 ± 0.0111	26.19 ± 0.33	0.035 ± 0.007	0.018 ± 0.003

Table 9: Black-box attack performance against FOCAL across five datasets under a maximum query budget of $Q = 50$, evaluated on images with an initial localization F1> 0.5.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	30.8 ± 4.1%	–	–	0.1182 ± 0.0034	31.12 ± 0.35	**0.017** ± 0.003	**0.009** ± 0.001
	AdvFor	25.6 ± 2.6%	**3.1** ± 0.7	**2.4** ± 0.7	0.1663 ± 0.0117	32.85 ± 0.60	0.024 ± 0.005	0.013 ± 0.003
	RLGC	37.6 ± 2.6%	17.3 ± 2.2	12.0 ± 3.2	0.1990 ± 0.0180	**33.13** ± 1.04	0.040 ± 0.007	0.021 ± 0.004
	Square	58.4 ± 3.6%	10.4 ± 1.3	6.2 ± 1.9	0.1355 ± 0.0028	31.92 ± 0.05	0.022 ± 0.009	0.011 ± 0.005
	TALA (Ours)	**70.4** ± 4.3%	16.5 ± 2.8	14.3 ± 2.3	**0.0640** ± 0.0084	32.95 ± 0.97	0.031 ± 0.006	0.016 ± 0.003
COCOGlide	DiffCFG	11.6 ± 3.3%	–	–	**0.1123** ± 0.0133	32.96 ± 0.84	**0.026** ± 0.005	**0.014** ± 0.003
	AdvFor	14.4 ± 2.6%	**3.0** ± 1.4	**1.8** ± 0.8	0.1675 ± 0.0158	**34.45** ± 0.82	0.040 ± 0.009	0.021 ± 0.005
	RLGC	26.0 ± 2.4%	21.3 ± 1.1	19.9 ± 1.7	0.2794 ± 0.0110	28.48 ± 0.55	0.053 ± 0.007	0.028 ± 0.003
	Square	60.8 ± 7.2%	15.3 ± 0.8	11.3 ± 2.0	0.1566 ± 0.0135	28.83 ± 0.16	0.027 ± 0.007	**0.014** ± 0.003
	TALA (Ours)	**61.6** ± 2.6%	20.5 ± 0.5	18.6 ± 1.3	0.1241 ± 0.0131	30.23 ± 0.34	0.042 ± 0.006	0.022 ± 0.003
COVERAGE	DiffCFG	14.0 ± 4.0%	–	–	**0.1091** ± 0.0067	33.42 ± 0.29	**0.026** ± 0.010	**0.013** ± 0.005
	AdvFor	5.2 ± 2.3%	**3.7** ± 1.2	**3.8** ± 1.5	0.1525 ± 0.0206	**34.92** ± 0.91	0.052 ± 0.022	0.027 ± 0.012
	RLGC	26.4 ± 4.1%	25.1 ± 2.7	25.3 ± 3.9	0.3343 ± 0.0205	29.65 ± 1.31	0.054 ± 0.006	0.028 ± 0.003
	Square	28.0 ± 3.7%	16.0 ± 2.2	10.8 ± 5.7	0.1156 ± 0.0115	28.62 ± 0.46	0.049 ± 0.008	0.025 ± 0.004
	TALA (Ours)	**91.2** ± 1.8%	13.1 ± 0.8	7.9 ± 0.9	0.1118 ± 0.0059	29.83 ± 0.27	0.031 ± 0.004	0.016 ± 0.002
IMD2020	DiffCFG	4.0 ± 2.0%	–	–	0.1576 ± 0.0246	**36.38** ± 1.60	0.043 ± 0.021	0.020 ± 0.010
	AdvFor	3.6 ± 2.6%	**4.2** ± 0.9	**4.0** ± 0.8	0.2549 ± 0.0349	35.13 ± 0.87	**0.017** ± 0.013	**0.008** ± 0.007
	RLGC	13.6 ± 3.0%	28.6 ± 4.0	35.3 ± 7.9	0.4255 ± 0.0523	29.99 ± 2.05	0.057 ± 0.009	0.030 ± 0.005
	Square	34.8 ± 4.3%	7.9 ± 1.6	4.4 ± 2.2	0.1604 ± 0.0360	23.56 ± 0.75	0.038 ± 0.008	0.020 ± 0.004
	TALA (Ours)	**43.2** ± 5.8%	26.2 ± 2.7	26.2 ± 4.4	**0.1424** ± 0.0167	32.58 ± 0.41	0.044 ± 0.009	0.023 ± 0.005
FaceApp	DiffCFG	12.4 ± 4.3%	–	–	0.1524 ± 0.0023	34.79 ± 0.50	**0.021** ± 0.008	**0.011** ± 0.004
	AdvFor	14.4 ± 1.7%	**4.2** ± 1.7	**2.9** ± 1.7	0.1693 ± 0.0211	35.65 ± 0.84	0.035 ± 0.008	0.018 ± 0.004
	RLGC	21.6 ± 0.9%	13.1 ± 2.6	8.3 ± 1.6	0.1904 ± 0.0388	**36.09** ± 1.45	0.048 ± 0.003	0.025 ± 0.002
	Square	90.0 ± 3.2%	13.4 ± 1.2	10.3 ± 0.7	0.1471 ± 0.0013	34.63 ± 0.03	0.030 ± 0.002	0.016 ± 0.001
	TALA (Ours)	**96.8** ± 1.8%	16.2 ± 1.1	14.0 ± 1.6	**0.1053** ± 0.0077	30.55 ± 0.13	0.054 ± 0.001	0.028 ± 0.001

Input resolutions. We follow the input-resolution requirements of each method. TruFor and IFOSN process images at their original resolution, whereas FOCAL resizes inputs to 1024×1024 as part of its inference pipeline. The perturbation generators of RLGC and AdvFor process inputs at 256×256 and 384×256, respectively. DiffCFG divides images into 256×256 patches before applying the diffusion process. The Stable Diffusion reconstruction baseline uses inputs resized to

Table 10: Black-box attack performance against TruFor across five datasets under a maximum query budget of $Q = 100$, evaluated on images with an initial localization F1> 0.5.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	27.6 ± 3.8%	–	–	0.1181 ± 0.0028	30.82 ± 0.32	**0.014** ± 0.007	**0.007** ± 0.004
	AdvFor	29.6 ± 0.8%	**3.1** ± 0.4	**1.3** ± 0.6	0.1211 ± 0.0102	**34.01** ± 0.43	0.022 ± 0.006	0.011 ± 0.003
	RLGC	68.0 ± 3.3%	38.7 ± 2.7	40.9 ± 4.0	0.3043 ± 0.0110	28.26 ± 0.65	0.054 ± 0.006	0.028 ± 0.003
	Square	70.8 ± 2.7%	10.7 ± 0.8	3.5 ± 0.4	0.1603 ± 0.0036	29.80 ± 0.06	0.018 ± 0.007	0.009 ± 0.003
	TALA (Ours)	**94.0** ± 1.8%	28.0 ± 1.7	24.2 ± 4.0	**0.0879** ± 0.0035	31.26 ± 0.56	0.042 ± 0.005	0.022 ± 0.003
COCOGlide	DiffCFG	16.4 ± 1.7%	–	–	0.1180 ± 0.0129	33.87 ± 0.39	**0.009** ± 0.007	**0.005** ± 0.004
	AdvFor	39.2 ± 2.0%	**2.3** ± 0.2	**1.2** ± 0.4	0.2432 ± 0.0038	**35.32** ± 0.22	0.029 ± 0.005	0.015 ± 0.002
	RLGC	89.6 ± 0.8%	26.3 ± 0.4	16.5 ± 1.3	0.3419 ± 0.0028	27.61 ± 0.19	0.060 ± 0.003	0.031 ± 0.001
	Square	63.2 ± 1.6%	20.7 ± 2.4	11.5 ± 2.0	0.2060 ± 0.0077	27.09 ± 0.24	0.031 ± 0.005	0.016 ± 0.003
	TALA (Ours)	**97.2** ± 1.0%	11.9 ± 2.2	6.0 ± 0.9	**0.1146** ± 0.0095	30.28 ± 0.38	0.035 ± 0.002	0.018 ± 0.001
COVERAGE	DiffCFG	19.2 ± 2.7%	–	–	0.1157 ± 0.0068	**35.32** ± 0.25	**0.017** ± 0.006	**0.009** ± 0.003
	AdvFor	17.6 ± 2.3%	**4.1** ± 0.3	**3.7** ± 0.2	0.1921 ± 0.0059	33.95 ± 0.27	0.038 ± 0.009	0.020 ± 0.005
	RLGC	65.2 ± 1.6%	41.9 ± 1.2	33.0 ± 1.3	0.4366 ± 0.0067	25.86 ± 0.27	0.068 ± 0.003	0.035 ± 0.001
	Square	52.4 ± 2.3%	18.6 ± 3.1	12.3 ± 3.3	**0.1020** ± 0.0070	28.44 ± 0.12	0.024 ± 0.004	0.012 ± 0.002
	TALA (Ours)	**99.6** ± 0.8%	13.4 ± 0.7	9.5 ± 1.3	0.1172 ± 0.0048	29.71 ± 0.28	0.030 ± 0.003	0.016 ± 0.002
IMD2020	DiffCFG	12.0 ± 4.0%	–	–	0.1656 ± 0.0036	34.79 ± 0.62	**0.018** ± 0.012	**0.009** ± 0.006
	AdvFor	4.4 ± 0.8%	**2.2** ± 0.4	**2.1** ± 0.4	0.1907 ± 0.0219	**36.62** ± 0.33	**0.018** ± 0.020	**0.009** ± 0.011
	RLGC	62.8 ± 1.6%	45.9 ± 1.6	46.0 ± 0.6	0.5286 ± 0.0041	23.89 ± 0.19	0.070 ± 0.005	0.037 ± 0.002
	Square	30.0 ± 1.3%	12.7 ± 2.5	6.5 ± 1.9	**0.1601** ± 0.0160	23.72 ± 0.40	0.033 ± 0.008	0.017 ± 0.004
	TALA (Ours)	**94.4** ± 3.4%	35.5 ± 2.3	32.0 ± 2.0	0.2342 ± 0.0095	25.62 ± 0.44	0.043 ± 0.004	0.022 ± 0.002
FaceApp	DiffCFG	2.8 ± 1.1%	–	–	0.1512 ± 0.0156	34.65 ± 0.91	**0.026** ± 0.033	**0.013** ± 0.017
	AdvFor	30.8 ± 1.0%	**2.9** ± 0.3	**2.4** ± 0.5	0.1713 ± 0.0101	**35.31** ± 0.39	0.057 ± 0.004	0.029 ± 0.002
	RLGC	90.0 ± 1.3%	32.5 ± 0.9	26.7 ± 1.2	0.3889 ± 0.0051	28.29 ± 0.27	0.079 ± 0.001	0.041 ± 0.001
	Square	93.6 ± 1.5%	11.9 ± 1.0	8.8 ± 0.4	0.1930 ± 0.0028	28.71 ± 0.04	0.038 ± 0.003	0.020 ± 0.002
	TALA (Ours)	**98.8** ± 1.0%	23.6 ± 2.3	19.0 ± 1.9	**0.1462** ± 0.0041	29.36 ± 0.08	0.065 ± 0.004	0.034 ± 0.002

Table 11: Black-box attack performance against IFOSN across five datasets under a maximum query budget of $Q = 100$, evaluated on images with an initial localization F1> 0.5.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	32.0 ± 8.0%	–	–	**0.1086** ± 0.0043	31.04 ± 0.24	**0.010** ± 0.005	**0.005** ± 0.003
	AdvFor	30.0 ± 1.3%	**2.7** ± 0.1	2.0 ± 0.0	0.1614 ± 0.0084	**32.36** ± 0.36	0.025 ± 0.005	0.013 ± 0.002
	RLGC	86.4 ± 0.8%	38.4 ± 0.6	31.2 ± 1.0	0.3701 ± 0.0027	23.76 ± 0.13	0.062 ± 0.005	0.032 ± 0.003
	Square	76.0 ± 4.4%	5.8 ± 1.3	**1.0** ± 0.0	0.3007 ± 0.0141	22.06 ± 0.21	0.017 ± 0.002	0.009 ± 0.001
	TALA (Ours)	**99.6** ± 0.8%	22.0 ± 1.5	17.7 ± 2.8	0.1301 ± 0.0113	28.94 ± 0.32	0.030 ± 0.004	0.015 ± 0.002
COCOGlide	DiffCFG	2.4 ± 0.9%	–	–	**0.1113** ± 0.0125	**35.82** ± 0.27	**0.024** ± 0.020	**0.012** ± 0.011
	AdvFor	45.6 ± 1.5%	**1.9** ± 0.1	**1.0** ± 0.0	0.1725 ± 0.0021	35.28 ± 0.19	0.026 ± 0.002	0.014 ± 0.001
	RLGC	84.8 ± 1.6%	29.0 ± 0.5	21.3 ± 0.9	0.3536 ± 0.0038	24.88 ± 0.18	0.067 ± 0.001	0.035 ± 0.001
	Square	68.4 ± 2.9%	9.8 ± 1.0	1.2 ± 0.4	0.2711 ± 0.0116	23.67 ± 0.13	0.027 ± 0.005	0.014 ± 0.002
	TALA (Ours)	**97.2** ± 1.6%	11.0 ± 0.7	5.4 ± 0.5	0.1175 ± 0.0070	29.84 ± 0.21	0.031 ± 0.004	0.016 ± 0.002
COVERAGE	DiffCFG	8.4 ± 2.9%	–	–	0.1577 ± 0.0528	34.99 ± 0.53	0.064 ± 0.025	0.033 ± 0.013
	AdvFor	33.7 ± 4.2%	**2.1** ± 0.7	**1.6** ± 0.5	0.1397 ± 0.0097	**36.10** ± 0.73	**0.022** ± 0.012	**0.012** ± 0.006
	RLGC	**100.0** ± 0.0%	23.4 ± 0.7	20.6 ± 1.4	0.3333 ± 0.0086	29.68 ± 0.34	0.064 ± 0.002	0.033 ± 0.001
	Square	55.4 ± 7.9%	13.3 ± 6.3	5.8 ± 2.4	0.1198 ± 0.0089	27.51 ± 0.33	0.033 ± 0.006	0.017 ± 0.003
	TALA (Ours)	**100.0** ± 0.0%	10.2 ± 0.7	4.0 ± 0.0	**0.0747** ± 0.0088	32.47 ± 0.35	0.034 ± 0.003	0.018 ± 0.001
IMD2020	DiffCFG	5.6 ± 0.9%	–	–	**0.1322** ± 0.0085	**34.91** ± 0.79	**0.013** ± 0.015	**0.007** ± 0.008
	AdvFor	6.0 ± 0.0%	**4.5** ± 0.9	**4.6** ± 0.8	0.2569 ± 0.0089	33.30 ± 0.58	0.062 ± 0.005	0.032 ± 0.002
	RLGC	74.4 ± 0.8%	33.8 ± 0.6	31.6 ± 1.0	0.4784 ± 0.0018	26.27 ± 0.11	0.078 ± 0.002	0.041 ± 0.001
	Square	14.8 ± 3.0%	23.8 ± 7.3	14.0 ± 9.3	0.1878 ± 0.0172	24.88 ± 0.63	0.051 ± 0.008	0.026 ± 0.004
	TALA (Ours)	**90.4** ± 0.8%	26.8 ± 0.7	22.4 ± 0.8	0.2325 ± 0.0103	26.45 ± 0.21	0.052 ± 0.004	0.027 ± 0.002
FaceApp	DiffCFG	0.0 ± 0.0%	–	–	–	–	–	–
	AdvFor	20.0 ± 3.3%	**2.8** ± 0.6	2.1 ± 0.7	**0.1912** ± 0.0092	**36.11** ± 0.41	0.039 ± 0.009	0.020 ± 0.005
	RLGC	70.0 ± 2.2%	53.9 ± 1.0	53.0 ± 3.2	0.5522 ± 0.0062	22.03 ± 0.23	0.082 ± 0.002	0.043 ± 0.001
	Square	82.4 ± 3.4%	6.3 ± 1.9	**1.0** ± 0.0	0.3281 ± 0.0125	24.18 ± 0.22	**0.017** ± 0.003	**0.009** ± 0.002
	TALA (Ours)	**98.0** ± 1.3%	20.6 ± 2.0	15.3 ± 1.2	0.2461 ± 0.0082	25.97 ± 0.14	0.034 ± 0.003	0.017 ± 0.002

512×512. The resulting adversarial images are restored to the original image dimensions before computing PSNR and LPIPS.

Table 12: Black-box attack performance against FOCAL across five datasets under a maximum query budget of $Q = 100$, evaluated on images with an initial localization F1> 0.5.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	30.8 ± 4.1%	–	–	0.1182 ± 0.0034	31.12 ± 0.35	**0.017** ± 0.003	**0.009** ± 0.001
	AdvFor	23.6 ± 4.6%	**2.7** ± 0.5	**2.1** ± 0.7	0.1613 ± 0.0215	33.19 ± 0.97	0.025 ± 0.007	0.013 ± 0.004
	RLGC	44.8 ± 1.6%	29.8 ± 5.0	21.1 ± 4.4	0.2820 ± 0.0289	29.54 ± 1.57	0.042 ± 0.005	0.021 ± 0.003
	Square	63.2 ± 1.6%	17.7 ± 3.9	9.1 ± 1.1	0.1372 ± 0.0029	31.86 ± 0.03	0.023 ± 0.003	0.012 ± 0.002
	TALA (Ours)	**75.2** ± 2.4%	24.4 ± 2.2	15.6 ± 2.5	**0.0683** ± 0.0106	**33.20** ± 0.77	0.036 ± 0.007	0.019 ± 0.004
COCOGlide	DiffCFG	11.6 ± 3.3%	–	–	**0.1123** ± 0.0133	32.96 ± 0.84	**0.026** ± 0.005	**0.014** ± 0.003
	AdvFor	15.6 ± 2.9%	**4.3** ± 1.3	**3.9** ± 1.4	0.1836 ± 0.0159	**33.12** ± 0.64	0.039 ± 0.011	0.020 ± 0.006
	RLGC	34.8 ± 2.7%	31.5 ± 2.6	29.5 ± 4.9	0.3472 ± 0.0227	25.27 ± 1.13	0.051 ± 0.004	0.027 ± 0.002
	Square	76.0 ± 4.6%	13.9 ± 1.8	9.6 ± 0.6	0.1867 ± 0.0104	24.77 ± 0.27	0.033 ± 0.002	0.017 ± 0.001
	TALA (Ours)	**91.2** ± 2.7%	25.1 ± 1.0	18.6 ± 2.0	0.1851 ± 0.0110	25.81 ± 0.35	0.046 ± 0.004	0.024 ± 0.002
COVERAGE	DiffCFG	14.0 ± 4.0%	–	–	**0.1091** ± 0.0067	33.42 ± 0.29	**0.026** ± 0.010	**0.013** ± 0.005
	AdvFor	5.2 ± 2.0%	**4.8** ± 1.7	**4.8** ± 1.7	0.1821 ± 0.0236	**34.30** ± 0.72	0.046 ± 0.028	0.024 ± 0.015
	RLGC	44.4 ± 2.0%	45.2 ± 3.2	45.6 ± 4.5	0.4385 ± 0.0142	24.43 ± 0.91	0.054 ± 0.004	0.028 ± 0.002
	Square	42.4 ± 2.9%	29.9 ± 4.6	24.2 ± 4.8	0.1257 ± 0.0055	28.13 ± 0.30	0.053 ± 0.004	0.028 ± 0.002
	TALA (Ours)	**93.2** ± 1.0%	15.6 ± 1.0	9.4 ± 2.2	0.1180 ± 0.0088	29.28 ± 0.30	0.029 ± 0.003	0.015 ± 0.001
IMD2020	DiffCFG	4.0 ± 2.0%	–	–	**0.1576** ± 0.0246	**36.38** ± 1.60	0.043 ± 0.021	0.020 ± 0.010
	AdvFor	4.0 ± 1.3%	**6.7** ± 1.9	7.0 ± 1.6	0.2595 ± 0.0405	34.23 ± 0.95	**0.010** ± 0.006	**0.005** ± 0.003
	RLGC	29.2 ± 3.7%	55.6 ± 3.2	57.8 ± 4.4	0.5537 ± 0.0094	22.97 ± 0.51	0.048 ± 0.006	0.025 ± 0.003
	Square	47.2 ± 6.4%	10.6 ± 3.2	**3.2** ± 1.1	0.2322 ± 0.0194	20.17 ± 0.33	0.036 ± 0.008	0.019 ± 0.004
	TALA (Ours)	**89.6** ± 3.4%	29.8 ± 2.4	25.2 ± 2.5	0.2336 ± 0.0160	25.67 ± 0.60	0.045 ± 0.003	0.023 ± 0.002
FaceApp	DiffCFG	12.4 ± 4.3%	–	–	0.1524 ± 0.0023	34.79 ± 0.50	**0.021** ± 0.008	**0.011** ± 0.004
	AdvFor	13.2 ± 2.7%	**3.7** ± 1.7	**3.6** ± 2.7	0.1663 ± 0.0267	**35.72** ± 1.02	0.040 ± 0.012	0.021 ± 0.006
	RLGC	32.8 ± 5.6%	28.1 ± 4.2	16.3 ± 5.4	0.3028 ± 0.0282	31.72 ± 1.50	0.055 ± 0.009	0.028 ± 0.005
	Square	99.2 ± 1.6%	7.5 ± 0.4	6.1 ± 0.7	0.1855 ± 0.0017	31.47 ± 0.02	0.032 ± 0.005	0.017 ± 0.002
	TALA (Ours)	**99.6** ± 0.8%	21.1 ± 2.0	18.6 ± 1.0	**0.0938** ± 0.0047	31.85 ± 0.29	0.055 ± 0.003	0.029 ± 0.002

Table 13: Black-box attack performance against FOCAL across five datasets under a maximum query budget of $Q = 50$, evaluated on images with an initial localization F1> 0.1.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	44.8 ± 5.8%	–	–	0.1199 ± 0.0024	30.90 ± 0.13	**0.017** ± 0.003	**0.009** ± 0.001
	AdvFor	42.9 ± 3.5%	**2.9** ± 0.9	**1.7** ± 0.7	0.1675 ± 0.0050	33.03 ± 0.45	0.024 ± 0.008	0.012 ± 0.004
	RLGC	55.6 ± 3.7%	14.8 ± 2.5	7.9 ± 0.9	0.1751 ± 0.0337	35.70 ± 1.33	0.043 ± 0.003	0.022 ± 0.002
	Square	47.9 ± 3.0%	11.5 ± 1.1	4.0 ± 2.2	0.0769 ± 0.0047	38.02 ± 0.02	0.033 ± 0.006	0.017 ± 0.003
	TALA (Ours)	**75.2** ± 2.7%	18.3 ± 1.8	15.7 ± 0.7	**0.0393** ± 0.0046	**39.24** ± 0.53	0.037 ± 0.005	0.019 ± 0.003
COCOGlide	DiffCFG	16.2 ± 4.4%	–	–	**0.1081** ± 0.0038	32.06 ± 0.33	**0.037** ± 0.005	**0.019** ± 0.003
	AdvFor	26.0 ± 3.5%	**2.6** ± 0.4	**1.4** ± 0.5	0.1451 ± 0.0075	**34.83** ± 0.51	0.043 ± 0.003	0.022 ± 0.002
	RLGC	40.3 ± 3.3%	19.2 ± 1.4	19.3 ± 2.9	0.2605 ± 0.0148	29.58 ± 0.91	0.060 ± 0.005	0.031 ± 0.002
	Square	69.1 ± 4.8%	14.2 ± 0.5	10.4 ± 0.5	0.1467 ± 0.0056	29.16 ± 0.04	**0.037** ± 0.004	**0.019** ± 0.002
	TALA (Ours)	**83.8** ± 3.1%	14.5 ± 0.5	11.1 ± 1.1	0.1212 ± 0.0060	29.44 ± 0.23	0.050 ± 0.003	0.026 ± 0.001
COVERAGE	DiffCFG	20.7 ± 3.5%	–	–	0.1189 ± 0.0051	34.09 ± 0.47	**0.023** ± 0.004	**0.012** ± 0.002
	AdvFor	14.9 ± 1.6%	**2.5** ± 0.5	**1.1** ± 0.2	0.1606 ± 0.0075	**36.06** ± 0.39	0.028 ± 0.007	0.014 ± 0.004
	RLGC	36.3 ± 1.7%	21.0 ± 1.3	18.3 ± 2.9	0.2877 ± 0.0142	32.83 ± 0.75	0.054 ± 0.005	0.028 ± 0.002
	Square	33.3 ± 1.6%	14.2 ± 1.7	12.0 ± 2.9	0.0877 ± 0.0035	34.41 ± 0.07	0.047 ± 0.004	0.024 ± 0.002
	TALA (Ours)	**66.9** ± 3.9%	22.3 ± 0.8	21.5 ± 0.7	**0.0787** ± 0.0051	35.74 ± 0.21	0.035 ± 0.006	0.018 ± 0.003
IMD2020	DiffCFG	7.3 ± 1.8%	–	–	0.1656 ± 0.0075	35.13 ± 0.94	0.044 ± 0.011	0.023 ± 0.006
	AdvFor	5.2 ± 0.9%	**3.4** ± 1.6	**3.4** ± 2.3	0.1974 ± 0.0117	**35.81** ± 1.32	**0.027** ± 0.009	**0.014** ± 0.005
	RLGC	20.9 ± 1.2%	19.5 ± 3.0	15.4 ± 2.4	0.3312 ± 0.0383	33.36 ± 1.90	0.056 ± 0.006	0.029 ± 0.003
	Square	35.4 ± 2.2%	9.0 ± 0.7	5.5 ± 1.0	0.1236 ± 0.0121	26.63 ± 0.43	0.036 ± 0.005	0.018 ± 0.002
	TALA (Ours)	**42.6** ± 3.2%	22.8 ± 1.4	20.7 ± 2.0	**0.1103** ± 0.0053	35.07 ± 0.37	0.044 ± 0.004	0.023 ± 0.002
FaceApp	DiffCFG	31.2 ± 6.5%	–	–	0.1535 ± 0.0028	34.14 ± 0.19	**0.042** ± 0.006	**0.022** ± 0.003
	AdvFor	44.0 ± 5.5%	**2.9** ± 0.2	**2.1** ± 0.2	0.1535 ± 0.0044	36.18 ± 0.15	0.054 ± 0.006	0.028 ± 0.003
	RLGC	51.6 ± 3.0%	10.2 ± 1.0	6.4 ± 1.1	0.1507 ± 0.0193	39.34 ± 0.65	0.062 ± 0.003	0.032 ± 0.002
	Square	48.7 ± 4.8%	16.9 ± 1.1	11.9 ± 0.9	0.0870 ± 0.0040	38.92 ± 0.02	0.047 ± 0.001	0.024 ± 0.001
	TALA (Ours)	**70.2** ± 2.8%	19.8 ± 0.9	17.7 ± 1.6	**0.0336** ± 0.0060	**41.08** ± 0.65	0.061 ± 0.003	0.032 ± 0.001

C.2 Attack Configurations and Perturbation-Budget Selection

Square. We use the official ℓ_2 variant of Square (Andriushchenko et al., 2020), which achieves substantially higher ASR than its ℓ_∞ variant in our preliminary evaluation.

Table 14: Black-box attack performance against TruFor across five datasets under a maximum query budget of $Q = 50$, evaluated on images with an initial localization F1> 0.1.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	36.5 ± 1.4%	–	–	0.1194 ± 0.0036	31.23 ± 0.18	**0.013** ± 0.007	**0.007** ± 0.004
	AdvFor	43.8 ± 1.6%	**2.4** ± 0.8	**1.0** ± 0.0	0.1279 ± 0.0083	34.87 ± 0.33	0.022 ± 0.006	0.012 ± 0.003
	RLGC	57.7 ± 3.0%	15.0 ± 1.4	6.3 ± 1.1	0.1727 ± 0.0145	36.06 ± 0.80	0.045 ± 0.005	0.024 ± 0.002
	Square	67.6 ± 1.0%	9.5 ± 2.0	5.7 ± 0.7	0.0908 ± 0.0011	35.89 ± 0.01	0.022 ± 0.006	0.012 ± 0.003
	TALA (Ours)	**91.9** ± 3.4%	14.4 ± 0.9	12.9 ± 1.5	**0.0276** ± 0.0073	**37.15** ± 0.95	0.045 ± 0.003	0.023 ± 0.002
COCOGlide	DiffCFG	23.4 ± 2.9%	–	–	0.1034 ± 0.0054	32.72 ± 0.33	**0.018** ± 0.003	**0.009** ± 0.002
	AdvFor	48.3 ± 3.0%	**2.4** ± 0.5	**1.1** ± 0.2	0.1904 ± 0.0124	**34.93** ± 0.34	0.044 ± 0.004	0.023 ± 0.002
	RLGC	80.3 ± 0.7%	13.5 ± 0.3	11.4 ± 0.4	0.2243 ± 0.0040	32.88 ± 0.22	0.061 ± 0.002	0.032 ± 0.001
	Square	51.6 ± 2.7%	14.9 ± 1.4	10.4 ± 1.5	0.1506 ± 0.0041	30.41 ± 0.08	0.048 ± 0.005	0.025 ± 0.003
	TALA (Ours)	**88.3** ± 2.8%	13.8 ± 1.7	7.3 ± 2.7	**0.0869** ± 0.0112	33.52 ± 0.55	0.039 ± 0.004	0.020 ± 0.002
COVERAGE	DiffCFG	29.0 ± 3.6%	–	–	0.1217 ± 0.0027	35.00 ± 0.17	**0.016** ± 0.003	**0.008** ± 0.002
	AdvFor	24.9 ± 3.0%	**3.3** ± 0.7	**3.1** ± 0.5	0.1547 ± 0.0046	**35.53** ± 0.33	0.039 ± 0.004	0.020 ± 0.002
	RLGC	45.8 ± 1.6%	16.6 ± 0.6	14.5 ± 1.0	0.2642 ± 0.0051	34.16 ± 0.27	0.062 ± 0.003	0.032 ± 0.002
	Square	33.3 ± 2.6%	14.1 ± 2.1	10.8 ± 3.4	0.0926 ± 0.0034	34.49 ± 0.05	0.041 ± 0.006	0.021 ± 0.003
	TALA (Ours)	**84.7** ± 2.3%	19.0 ± 1.1	19.6 ± 1.7	**0.0828** ± 0.0033	35.04 ± 0.21	0.033 ± 0.004	0.017 ± 0.002
IMD2020	DiffCFG	16.1 ± 3.1%	–	–	0.1696 ± 0.0061	34.66 ± 0.48	**0.025** ± 0.004	**0.013** ± 0.002
	AdvFor	6.4 ± 0.5%	**1.5** ± 0.3	**1.0** ± 0.0	0.1470 ± 0.0038	**37.15** ± 0.26	0.050 ± 0.001	0.026 ± 0.001
	RLGC	38.1 ± 2.1%	26.8 ± 0.4	26.3 ± 0.4	0.4194 ± 0.0054	28.65 ± 0.11	0.067 ± 0.002	0.035 ± 0.001
	Square	31.2 ± 2.6%	10.0 ± 1.4	6.5 ± 0.7	**0.0844** ± 0.0091	29.19 ± 0.44	0.034 ± 0.004	0.018 ± 0.002
	TALA (Ours)	**55.0** ± 3.7%	27.2 ± 1.3	28.3 ± 2.7	0.1463 ± 0.0094	31.00 ± 0.36	0.050 ± 0.002	0.026 ± 0.001
FaceApp	DiffCFG	17.9 ± 2.4%	–	–	0.1490 ± 0.0046	34.33 ± 0.30	**0.039** ± 0.014	**0.020** ± 0.007
	AdvFor	56.8 ± 2.3%	**3.0** ± 0.3	**2.1** ± 0.2	0.1456 ± 0.0021	**36.26** ± 0.15	0.065 ± 0.002	0.034 ± 0.001
	RLGC	96.4 ± 1.7%	15.4 ± 0.6	10.5 ± 1.1	0.2124 ± 0.0070	36.19 ± 0.28	0.077 ± 0.002	0.040 ± 0.001
	Square	88.0 ± 3.2%	9.8 ± 0.9	5.5 ± 1.3	0.1829 ± 0.0030	31.48 ± 0.03	0.053 ± 0.005	0.027 ± 0.003
	TALA (Ours)	**98.8** ± 1.0%	10.5 ± 0.9	8.3 ± 1.2	**0.1222** ± 0.0097	30.06 ± 0.33	0.060 ± 0.003	0.031 ± 0.002

Table 15: Black-box attack performance against IFOSN across five datasets under a maximum query budget of $Q = 50$, evaluated on images with an initial localization F1> 0.1.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	48.9 ± 8.0%	–	–	0.1129 ± 0.0015	31.05 ± 0.12	**0.010** ± 0.003	**0.005** ± 0.002
	AdvFor	42.1 ± 4.9%	**2.4** ± 0.2	1.2 ± 0.4	0.1461 ± 0.0068	**33.56** ± 0.21	0.025 ± 0.006	0.013 ± 0.003
	RLGC	77.4 ± 1.2%	19.3 ± 0.7	17.1 ± 1.7	0.2370 ± 0.0077	30.31 ± 0.52	0.062 ± 0.002	0.032 ± 0.001
	Square	75.8 ± 1.1%	9.4 ± 0.6	**1.0** ± 1.0	0.1544 ± 0.0024	31.29 ± 0.02	0.028 ± 0.004	0.015 ± 0.002
	TALA (Ours)	**97.0** ± 2.4%	10.9 ± 1.1	7.3 ± 1.1	**0.0737** ± 0.0062	32.79 ± 0.38	0.026 ± 0.005	0.013 ± 0.002
COCOGlide	DiffCFG	10.2 ± 1.3%	–	–	**0.0997** ± 0.0154	29.99 ± 0.66	0.033 ± 0.019	0.017 ± 0.010
	AdvFor	48.8 ± 1.6%	**2.2** ± 0.4	**1.0** ± 0.0	0.1750 ± 0.0019	**34.72** ± 0.17	**0.026** ± 0.007	**0.014** ± 0.004
	RLGC	72.1 ± 2.3%	19.1 ± 1.0	18.2 ± 1.2	0.3062 ± 0.0112	28.05 ± 0.49	0.067 ± 0.002	0.035 ± 0.001
	Square	48.6 ± 2.7%	11.4 ± 2.2	6.3 ± 4.7	0.1697 ± 0.0027	30.33 ± 0.05	0.038 ± 0.010	0.020 ± 0.005
	TALA (Ours)	**92.6** ± 1.9%	8.9 ± 1.2	4.8 ± 0.8	0.1074 ± 0.0190	30.98 ± 0.65	0.031 ± 0.006	0.016 ± 0.003
COVERAGE	DiffCFG	6.6 ± 2.9%	–	–	0.1289 ± 0.0162	35.06 ± 0.44	0.041 ± 0.017	0.021 ± 0.009
	AdvFor	41.8 ± 2.8%	**2.4** ± 0.4	**1.2** ± 0.4	0.1614 ± 0.0063	**36.22** ± 0.31	**0.038** ± 0.003	**0.020** ± 0.002
	RLGC	87.2 ± 0.8%	15.0 ± 0.3	9.7 ± 0.7	0.2499 ± 0.0035	34.45 ± 0.18	0.068 ± 0.001	0.036 ± 0.001
	Square	28.8 ± 2.4%	13.0 ± 3.0	6.4 ± 4.7	**0.0861** ± 0.0025	33.98 ± 0.14	0.052 ± 0.004	0.027 ± 0.002
	TALA (Ours)	**91.9** ± 3.6%	20.4 ± 1.2	18.5 ± 1.7	0.0904 ± 0.0048	34.86 ± 0.29	0.041 ± 0.004	0.021 ± 0.002
IMD2020	DiffCFG	12.2 ± 1.6%	–	–	0.1563 ± 0.0088	35.32 ± 0.48	**0.038** ± 0.008	**0.020** ± 0.004
	AdvFor	13.2 ± 1.5%	**2.0** ± 0.4	**1.0** ± 0.0	0.1524 ± 0.0090	**38.46** ± 0.55	0.057 ± 0.004	0.030 ± 0.002
	RLGC	68.2 ± 1.2%	20.5 ± 0.4	21.3 ± 0.4	0.3579 ± 0.0028	31.95 ± 0.14	0.076 ± 0.001	0.040 ± 0.001
	Square	22.1 ± 0.7%	11.1 ± 3.4	4.5 ± 3.0	**0.1195** ± 0.0054	29.52 ± 0.32	0.053 ± 0.008	0.028 ± 0.004
	TALA (Ours)	**76.2** ± 1.7%	21.6 ± 1.3	19.7 ± 1.0	0.1647 ± 0.0036	30.15 ± 0.26	0.047 ± 0.003	0.024 ± 0.002
FaceApp	DiffCFG	3.6 ± 1.4%	–	–	0.1527 ± 0.0099	34.75 ± 0.66	0.053 ± 0.032	0.028 ± 0.017
	AdvFor	48.2 ± 3.3%	**2.7** ± 0.3	**1.1** ± 0.2	0.1495 ± 0.0062	**37.16** ± 0.23	**0.035** ± 0.005	**0.018** ± 0.003
	RLGC	43.6 ± 1.8%	18.5 ± 2.6	13.4 ± 1.5	0.2724 ± 0.0282	33.05 ± 0.78	0.076 ± 0.004	0.040 ± 0.002
	Square	79.6 ± 2.2%	11.8 ± 1.2	6.9 ± 2.5	0.1803 ± 0.0024	33.12 ± 0.03	0.047 ± 0.006	0.024 ± 0.003
	TALA (Ours)	**86.2** ± 4.3%	24.4 ± 1.2	23.4 ± 1.8	**0.0875** ± 0.0103	35.42 ± 0.26	0.040 ± 0.005	0.021 ± 0.002

DiffCFG. DiffCF (Tailanián et al., 2024) provides an unguided variant, DiffCF, and an SSIM-guided variant, DiffCFG. We evaluate both and report DiffCFG, which achieves higher ASR in our evaluation. DiffCFG generates images through SSIM-guided reverse diffusion sampling, without explicitly constraining the output to an ℓ_∞ or ℓ_2 ball around the input. Imposing such a constraint would require modifying its original sampling procedure or post-processing its outputs. We therefore retain its recommended configuration.

Table 16: Black-box attack performance against FOCAL across five datasets under a maximum query budget of $Q = 100$, evaluated on images with an initial localization F1> 0.1.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	44.8 ± 5.8%	–	–	0.1199 ± 0.0024	30.90 ± 0.13	**0.017** ± 0.003	**0.009** ± 0.001
	AdvFor	37.5 ± 2.4%	**2.6** ± 0.5	**1.8** ± 0.4	0.1706 ± 0.0135	32.94 ± 0.30	0.028 ± 0.007	0.014 ± 0.004
	RLGC	63.8 ± 2.3%	21.2 ± 1.8	9.0 ± 1.9	0.2166 ± 0.0180	**33.69** ± 0.90	0.048 ± 0.002	0.025 ± 0.001
	Square	87.9 ± 1.3%	10.7 ± 1.6	3.0 ± 1.1	0.1673 ± 0.0040	29.72 ± 0.04	0.022 ± 0.003	0.011 ± 0.002
	TALA (Ours)	**95.6** ± 1.6%	17.1 ± 1.2	11.7 ± 0.7	**0.0703** ± 0.0100	32.82 ± 0.56	0.035 ± 0.003	0.018 ± 0.002
COCOGlide	DiffCFG	16.2 ± 4.4%	–	–	**0.1081** ± 0.0038	32.06 ± 0.33	0.037 ± 0.005	0.019 ± 0.003
	AdvFor	26.0 ± 3.4%	**2.4** ± 0.4	**1.8** ± 0.4	0.1426 ± 0.0140	**34.76** ± 0.51	0.042 ± 0.003	0.022 ± 0.002
	RLGC	50.8 ± 1.0%	26.9 ± 1.9	21.8 ± 2.2	0.3044 ± 0.0187	27.40 ± 0.91	0.063 ± 0.004	0.033 ± 0.002
	Square	72.4 ± 3.1%	27.0 ± 1.3	18.6 ± 2.1	0.1413 ± 0.0029	30.50 ± 0.09	**0.036** ± 0.004	**0.018** ± 0.002
	TALA (Ours)	**86.8** ± 2.4%	25.6 ± 1.4	21.4 ± 2.5	0.1150 ± 0.0049	30.92 ± 0.24	0.052 ± 0.003	0.027 ± 0.001
COVERAGE	DiffCFG	20.7 ± 3.5%	–	–	0.1189 ± 0.0051	34.09 ± 0.47	**0.023** ± 0.004	**0.012** ± 0.002
	AdvFor	13.3 ± 2.6%	**2.4** ± 0.8	**1.4** ± 0.4	0.1690 ± 0.0055	**35.79** ± 0.40	0.027 ± 0.007	0.014 ± 0.004
	RLGC	54.7 ± 1.9%	39.7 ± 1.3	36.9 ± 2.5	0.3932 ± 0.0096	27.69 ± 0.59	0.051 ± 0.002	0.027 ± 0.001
	Square	55.2 ± 2.3%	17.0 ± 2.5	7.3 ± 2.6	0.1414 ± 0.0015	26.21 ± 0.12	0.043 ± 0.004	0.023 ± 0.002
	TALA (Ours)	**93.5** ± 0.9%	14.7 ± 1.6	8.6 ± 1.6	**0.1004** ± 0.0072	31.50 ± 0.28	0.032 ± 0.003	0.017 ± 0.002
IMD2020	DiffCFG	7.3 ± 1.8%	–	–	**0.1656** ± 0.0075	35.13 ± 0.94	0.044 ± 0.011	0.023 ± 0.006
	AdvFor	5.4 ± 0.7%	**5.0** ± 2.7	**3.4** ± 1.1	0.2004 ± 0.0100	**35.83** ± 0.38	0.043 ± 0.016	0.022 ± 0.009
	RLGC	31.6 ± 1.1%	39.0 ± 5.2	40.7 ± 8.0	0.4424 ± 0.0231	27.85 ± 1.20	0.053 ± 0.003	0.028 ± 0.001
	Square	48.0 ± 3.2%	9.7 ± 1.7	8.3 ± 1.9	0.2400 ± 0.0119	19.98 ± 0.43	**0.038** ± 0.007	**0.020** ± 0.004
	TALA (Ours)	**90.4** ± 1.3%	27.0 ± 0.7	22.6 ± 0.9	0.2164 ± 0.0050	26.53 ± 0.16	0.046 ± 0.003	0.024 ± 0.001
FaceApp	DiffCFG	31.2 ± 6.5%	–	–	0.1535 ± 0.0028	34.14 ± 0.19	**0.042** ± 0.006	**0.022** ± 0.003
	AdvFor	41.8 ± 4.7%	**3.3** ± 0.7	**2.1** ± 0.5	0.1564 ± 0.0097	36.15 ± 0.41	0.056 ± 0.004	0.029 ± 0.002
	RLGC	57.1 ± 4.4%	17.2 ± 1.5	8.7 ± 1.3	0.2049 ± 0.0190	**36.78** ± 0.96	0.065 ± 0.003	0.034 ± 0.001
	Square	93.8 ± 2.2%	16.9 ± 2.9	11.0 ± 0.6	0.1431 ± 0.0019	34.58 ± 0.02	0.044 ± 0.003	**0.022** ± 0.003
	TALA (Ours)	**97.8** ± 1.4%	20.7 ± 1.1	15.2 ± 2.2	**0.0643** ± 0.0032	36.49 ± 0.29	0.060 ± 0.003	0.031 ± 0.001

Table 17: Black-box attack performance against TruFor across five datasets under a maximum query budget of $Q = 100$, evaluated on images with an initial localization F1> 0.1.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	36.5 ± 1.4%	–	–	0.1194 ± 0.0036	31.23 ± 0.18	**0.013** ± 0.007	**0.007** ± 0.004
	AdvFor	41.5 ± 3.6%	**2.0** ± 0.4	**1.0** ± 0.0	0.1299 ± 0.0042	**34.68** ± 0.46	0.014 ± 0.006	**0.007** ± 0.003
	RLGC	77.3 ± 2.6%	28.2 ± 1.6	12.5 ± 3.5	0.2486 ± 0.0128	32.10 ± 0.79	0.049 ± 0.003	0.026 ± 0.002
	Square	87.5 ± 1.6%	6.8 ± 0.4	2.4 ± 0.5	0.1765 ± 0.0045	28.69 ± 0.08	0.016 ± 0.004	0.008 ± 0.002
	TALA (Ours)	**96.9** ± 1.6%	19.2 ± 1.1	15.2 ± 2.2	**0.0614** ± 0.0046	33.68 ± 0.47	0.042 ± 0.003	0.022 ± 0.002
COCOGlide	DiffCFG	23.4 ± 2.9%	–	–	0.1034 ± 0.0054	32.72 ± 0.33	**0.018** ± 0.003	**0.009** ± 0.002
	AdvFor	49.5 ± 1.8%	**2.1** ± 0.3	**1.0** ± 0.0	0.1914 ± 0.0022	**35.02** ± 0.20	0.039 ± 0.004	0.020 ± 0.002
	RLGC	96.6 ± 0.6%	22.9 ± 0.3	12.9 ± 0.5	0.2811 ± 0.0023	29.79 ± 0.17	0.067 ± 0.002	0.035 ± 0.001
	Square	73.2 ± 3.2%	18.4 ± 1.6	11.9 ± 2.6	0.1648 ± 0.0066	28.42 ± 0.07	0.042 ± 0.006	0.022 ± 0.003
	TALA (Ours)	**97.2** ± 0.6%	14.3 ± 0.9	6.0 ± 1.4	**0.0905** ± 0.0040	33.00 ± 0.22	0.038 ± 0.004	0.020 ± 0.002
COVERAGE	DiffCFG	29.0 ± 3.6%	–	–	0.1217 ± 0.0027	35.00 ± 0.17	**0.016** ± 0.003	**0.008** ± 0.002
	AdvFor	24.1 ± 2.1%	**3.9** ± 0.7	**2.8** ± 0.5	0.1562 ± 0.0081	**35.41** ± 0.31	0.043 ± 0.006	0.022 ± 0.003
	RLGC	65.8 ± 0.9%	34.7 ± 0.8	23.6 ± 1.2	0.3687 ± 0.0050	29.43 ± 0.29	0.068 ± 0.002	0.035 ± 0.001
	Square	62.2 ± 2.5%	14.6 ± 2.1	9.0 ± 1.7	0.1106 ± 0.0050	29.02 ± 0.13	0.022 ± 0.004	0.011 ± 0.002
	TALA (Ours)	**99.2** ± 0.7%	16.1 ± 1.2	12.5 ± 1.0	**0.0971** ± 0.0052	32.69 ± 0.22	0.031 ± 0.004	0.016 ± 0.002
IMD2020	DiffCFG	16.1 ± 3.1%	–	–	0.1696 ± 0.0061	34.66 ± 0.48	**0.025** ± 0.004	**0.013** ± 0.002
	AdvFor	6.4 ± 0.4%	**1.5** ± 0.3	**1.2** ± 0.4	0.1473 ± 0.0032	**37.13** ± 0.44	0.048 ± 0.005	0.025 ± 0.003
	RLGC	62.5 ± 1.6%	43.7 ± 1.7	41.1 ± 4.0	0.4994 ± 0.0088	24.85 ± 0.42	0.074 ± 0.001	0.038 ± 0.001
	Square	34.7 ± 2.2%	13.7 ± 2.4	8.5 ± 1.4	**0.1076** ± 0.0084	26.76 ± 0.25	0.040 ± 0.002	0.021 ± 0.001
	TALA (Ours)	**93.8** ± 0.9%	36.7 ± 1.8	33.8 ± 2.8	0.2097 ± 0.0054	27.32 ± 0.20	0.049 ± 0.003	0.026 ± 0.001
FaceApp	DiffCFG	17.9 ± 2.4%	–	–	0.1490 ± 0.0046	34.33 ± 0.30	**0.039** ± 0.014	**0.020** ± 0.007
	AdvFor	54.8 ± 3.2%	**3.0** ± 0.6	**2.1** ± 0.2	0.1501 ± 0.0064	**36.18** ± 0.28	0.063 ± 0.006	0.033 ± 0.003
	RLGC	99.6 ± 0.8%	17.0 ± 0.8	10.6 ± 0.8	0.2236 ± 0.0087	35.67 ± 0.40	0.077 ± 0.002	0.040 ± 0.001
	Square	96.7 ± 1.6%	8.2 ± 0.6	4.7 ± 1.3	0.2035 ± 0.0091	29.35 ± 0.05	0.043 ± 0.001	0.022 ± 0.001
	TALA (Ours)	**100.0** ± 0.0%	11.6 ± 1.4	8.8 ± 1.3	**0.1265** ± 0.0114	29.82 ± 0.53	0.059 ± 0.003	0.031 ± 0.001

RLGC and AdvFor. RLGC (Mo et al., 2025) and AdvFor (Mo et al., 2026) progressively accumulate perturbations during the query process without an explicit ℓ_∞ or ℓ_2 budget. Each update depends on the current perturbed image and the victim's localization feedback; projecting candidates onto an additional norm ball would alter this feedback and the subsequent attack trajectory.

Table 18: Black-box attack performance against IFOSN across five datasets under a maximum query budget of $Q = 100$, evaluated on images with an initial localization F1 score above 0.1.

Dataset	Method	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	48.9 ± 8.0%	–	–	0.1129 ± 0.0015	31.05 ± 0.12	**0.010** ± 0.003	**0.005** ± 0.002
	AdvFor	39.6 ± 3.5%	**2.0** ± 0.4	**1.2** ± 0.4	0.1384 ± 0.0069	**33.95** ± 0.52	0.022 ± 0.007	0.011 ± 0.004
	RLGC	95.7 ± 0.0%	28.2 ± 0.4	23.0 ± 1.7	0.2961 ± 0.0031	28.08 ± 0.12	0.063 ± 0.005	0.032 ± 0.003
	Square	84.0 ± 3.3%	8.1 ± 2.1	1.3 ± 0.4	0.1988 ± 0.0088	28.00 ± 0.12	0.021 ± 0.005	0.011 ± 0.003
	TALA (Ours)	**100.0** ± 0.0%	13.1 ± 0.7	8.8 ± 0.9	**0.0785** ± 0.0103	32.23 ± 0.41	0.029 ± 0.003	0.015 ± 0.002
COCOGlide	DiffCFG	10.2 ± 1.3%	–	–	**0.0997** ± 0.0154	29.99 ± 0.66	0.033 ± 0.019	0.017 ± 0.010
	AdvFor	49.3 ± 2.7%	**2.1** ± 0.2	**1.0** ± 0.0	0.1709 ± 0.0079	**34.83** ± 0.28	0.032 ± 0.002	0.017 ± 0.001
	RLGC	90.2 ± 0.9%	30.2 ± 0.6	24.0 ± 0.6	0.3626 ± 0.0049	25.33 ± 0.22	0.070 ± 0.003	0.037 ± 0.001
	Square	69.7 ± 2.9%	10.2 ± 1.5	1.6 ± 0.5	0.2363 ± 0.0190	25.35 ± 0.17	**0.031** ± 0.004	**0.016** ± 0.002
	TALA (Ours)	**97.1** ± 1.1%	13.0 ± 1.3	5.0 ± 1.2	0.1208 ± 0.0108	30.41 ± 0.28	0.034 ± 0.006	0.017 ± 0.003
COVERAGE	DiffCFG	6.6 ± 2.9%	–	–	0.1289 ± 0.0162	35.06 ± 0.44	0.041 ± 0.017	0.021 ± 0.009
	AdvFor	42.1 ± 2.2%	**2.4** ± 0.3	**1.1** ± 0.2	0.1580 ± 0.0033	**36.27** ± 0.26	**0.038** ± 0.004	**0.020** ± 0.002
	RLGC	**99.7** ± 0.6%	23.0 ± 0.5	13.2 ± 0.4	0.2973 ± 0.0032	32.52 ± 0.26	0.071 ± 0.001	0.037 ± 0.001
	Square	50.5 ± 4.0%	15.9 ± 1.7	3.1 ± 1.7	0.1368 ± 0.0064	29.15 ± 0.38	0.042 ± 0.003	0.022 ± 0.001
	TALA (Ours)	95.9 ± 1.7%	14.9 ± 1.6	5.5 ± 0.5	**0.0982** ± 0.0072	32.55 ± 0.21	0.045 ± 0.002	0.024 ± 0.001
IMD2020	DiffCFG	12.2 ± 1.6%	–	–	0.1563 ± 0.0088	35.32 ± 0.48	**0.038** ± 0.008	**0.020** ± 0.004
	AdvFor	12.9 ± 0.7%	**2.3** ± 0.3	**1.1** ± 0.2	**0.1530** ± 0.0170	**38.21** ± 0.49	0.061 ± 0.005	0.032 ± 0.003
	RLGC	83.3 ± 0.5%	29.1 ± 0.5	26.4 ± 0.5	0.4123 ± 0.0028	29.51 ± 0.12	0.078 ± 0.002	0.041 ± 0.001
	Square	31.8 ± 3.6%	16.3 ± 3.8	5.3 ± 3.5	0.1645 ± 0.0130	24.83 ± 0.35	0.055 ± 0.003	0.029 ± 0.002
	TALA (Ours)	**94.5** ± 1.8%	21.5 ± 1.3	15.2 ± 1.3	0.1805 ± 0.0035	28.71 ± 0.22	0.051 ± 0.003	0.027 ± 0.002
FaceApp	DiffCFG	3.6 ± 1.4%	–	–	0.1527 ± 0.0099	34.75 ± 0.66	0.053 ± 0.032	0.028 ± 0.017
	AdvFor	48.2 ± 3.0%	**1.9** ± 0.2	**1.0** ± 0.0	0.1357 ± 0.0069	**37.54** ± 0.26	**0.031** ± 0.004	**0.016** ± 0.002
	RLGC	85.6 ± 1.3%	48.5 ± 2.1	52.9 ± 4.0	0.4765 ± 0.0146	25.10 ± 0.46	0.083 ± 0.001	0.044 ± 0.000
	Square	88.5 ± 3.5%	14.9 ± 0.9	5.9 ± 0.8	0.1949 ± 0.0032	31.38 ± 0.04	0.039 ± 0.002	0.020 ± 0.001
	TALA (Ours)	**96.8** ± 2.2%	23.4 ± 1.9	20.4 ± 2.3	**0.1161** ± 0.0049	31.91 ± 0.21	0.039 ± 0.004	0.020 ± 0.002

We therefore retain their original attack designs and use their configurations as reference operating points.

Perturbation-budget selection. We sweep the perturbation budgets of TALA and Square while keeping other settings fixed. TALA uses an ℓ_∞ constraint and Square an ℓ_2 constraint, so their numerical budgets are not directly comparable. We therefore compare their effectiveness–quality trade-offs, using LPIPS as the primary perceptual-quality metric and PSNR as a secondary consideration. For the tables, we select representative operating points that balance attack effectiveness and perceptual distortion, aiming for high ASR while keeping LPIPS low and PSNR high. The perturbation budgets used for each victim–dataset setting are detailed in Table 20. To avoid relying on individual operating points, the ASR–LPIPS plots additionally report the broader range of evaluated perturbation budgets.

Notably, DiffCFG or AdvFor achieves low distortion in some settings among Tables 7–18, partly because perceptual metrics are averaged only over successful attacks. Thus, different methods may be evaluated on different image subsets, while higher-ASR methods additionally account for more difficult-to-attack images. Additionally, to achieve higher ASR than methods without directly adjustable perturbation strength, such as RLGC, we allow larger perturbations for TALA in some settings. As further analyzed in Sec. C.3, TALA still exhibits a favorable effectiveness–quality trade-off with reduced perturbation budgets.

All baselines use their official implementations and recommended settings, except for the Square perturbation-budget variations described above.

C.3 Additional Baseline Comparisons

Attack effectiveness and perceptual quality. Tables 7–18 report detailed results across three victim detectors and five datasets under initial F1 thresholds of 0.5 and 0.1 and query budgets of $Q \in \{50, 100\}$. They include ASR, average and median query counts, LPIPS, PSNR, and post-attack F1 and IoU.

Under the initial F1 > 0.5 criterion, TALA achieves the highest mean ASR in all 15 detector–dataset combinations at $Q = 50$ and the highest or tied-highest mean ASR in all 15 combinations

Table 19: Attack effectiveness under different perceptual-quality operating points against FOCAL. DiffCFG and AdvFor use their standard configurations, while TALA is evaluated with different perturbation budgets ϵ. The maximum query budget is $Q = 100$, and evaluated images have initial localization F1 > 0.1. Bold TALA values indicate strictly better performance than both DiffCFG and AdvFor. Among highlighted entries, darker blue indicates a better rank.

Dataset	Method	ϵ	ASR↑	Avg. Q↓	Med. Q↓	LPIPS↓	PSNR↑	F1↓	IoU↓
CASIAv1	DiffCFG	–	47.6%	–	–	0.1174	30.78	0.019	0.010
	AdvFor	–	36.5%	3	2	0.1924	32.64	0.025	0.013
	TALA	0.047	**88.9%**	20	15	**0.0276**	**37.74**	0.038	0.020
	TALA	0.031	**77.8%**	19	13	**0.0161**	**41.46**	0.036	0.019
	TALA	0.019	**77.8%**	24	19	**0.0120**	**43.99**	0.041	0.021
COCOGlide	DiffCFG	–	11.1%	–	–	0.1078	31.73	0.032	0.017
	AdvFor	–	23.8%	3	2	0.1290	34.84	0.037	0.019
	TALA	0.047	**77.8%**	25	21	**0.0426**	**37.16**	0.058	0.030
	TALA	0.031	**63.5%**	23	18	**0.0285**	**40.52**	0.061	0.032
	TALA	0.019	**58.7%**	30	28	**0.0181**	**43.66**	0.055	0.029
COVERAGE	DiffCFG	–	24.1%	–	–	0.1183	33.62	0.025	0.013
	AdvFor	–	9.2%	2	2	0.1739	35.52	0.027	0.014
	TALA	0.047	**83.9%**	22	20	**0.0394**	**36.70**	0.042	0.022
	TALA	0.031	**81.6%**	26	21	**0.0349**	**39.01**	0.040	0.021
	TALA	0.019	**66.7%**	28	24	**0.0186**	**43.07**	0.039	0.020
IMD2020	DiffCFG	–	8.6%	–	–	0.1617	34.05	0.055	0.028
	AdvFor	–	5.4%	6	4	0.1975	35.41	0.054	0.028
	TALA	0.047	**63.4%**	34	30	**0.0749**	**35.23**	**0.054**	0.028
	TALA	0.031	**46.2%**	34	29	**0.0524**	**38.44**	0.057	0.030
	TALA	0.019	**36.6%**	35	31	**0.0384**	**41.98**	**0.052**	**0.027**
FaceApp	DiffCFG	–	32.7%	–	–	0.1499	33.89	0.039	0.020
	AdvFor	–	43.6%	3	2	0.1478	36.38	0.056	0.029
	TALA	0.047	**98.2%**	16	12	**0.0276**	**38.77**	0.065	0.034
	TALA	0.031	**85.5%**	21	16	**0.0227**	**41.14**	0.065	0.034
	TALA	0.019	**67.3%**	23	18	**0.0140**	**44.93**	0.067	0.035

at $Q = 100$. Under the initial F1 > 0.1 criterion, it achieves the highest mean ASR in all 15 combinations at $Q = 50$ and in 14 of 15 combinations at $Q = 100$. These results extend the main comparison in Table 1 to additional query budgets and initial-localization conditions.

At the reported operating points, TALA achieves higher mean ASR, lower mean LPIPS, and higher mean PSNR than the strongest baseline in ASR in 51 of the 60 evaluated combinations. For example, on TruFor–IMD2020 with initial F1 > 0.1 and $Q = 50$, TALA achieves an ASR of 55.0%, compared with RLGC's 38.1%, while reducing LPIPS from 0.4194 to 0.1463 and increasing PSNR from 28.65 to 31.00 dB. Overall, the results support improved attack effectiveness and perceptual quality across most evaluated settings. Although TALA does not consistently minimize query counts, it achieves lower average and median query counts than the strongest baseline in ASR in 26 and 25 of the 60 settings, respectively. As in the main experiments, metrics other than ASR are computed over each method's successful attacks. TALA's higher ASR may include harder cases requiring more queries, so these conditional query counts should be interpreted alongside ASR rather than as a standalone measure of efficiency.

Effectiveness–quality trade-off. Beyond the CASIAv1 comparison in Fig. 3, Fig. 5 evaluates the more challenging IMD2020 dataset under $Q = 50$ and initial F1 > 0.5. Across the three victims, TALA achieves higher ASR at comparable LPIPS levels. As greater distortion is allowed, its ASR generally increases rapidly before leveling off, whereas Square shows limited gains or occasional declines across the evaluated configurations. These results show that TALA's favorable effectiveness–quality trade-off extends beyond CASIAv1.

Table 19 complements the ASR–LPIPS curves with a focused comparison against DiffCFG and AdvFor, additionally reporting PSNR. These baselines achieve high perceptual quality in some settings despite relatively low ASR. We examine whether TALA retains its effectiveness advantage as its perturbation budget is reduced. The comparison uses FOCAL under $Q = 100$ and initial F1 > 0.1.

DiffCFG and AdvFor retain their standard configurations, while we vary only TALA's perturbation budget ϵ. Reducing ϵ improves TALA's perceptual quality while generally decreasing ASR. Nevertheless, at both $\epsilon = 0.031$ and $\epsilon = 0.019$, TALA achieves higher ASR, lower LPIPS, and higher PSNR than both baselines on all five datasets. For example, on IMD2020 with $\epsilon = 0.031$, TALA achieves an ASR of 46.2%, compared with 8.6% for DiffCFG and 5.4% for AdvFor. Its LPIPS is also lower (0.0524 vs. 0.1617 and 0.1975), and its PSNR is higher (38.44 vs. 34.05 and 35.41 dB). Thus, TALA retains a clear ASR advantage even with better perceptual quality on both metrics.

C.4 Global Image Degradation Details

Since this experiment is intended to diagnose whether generic image degradation alone can explain attack effectiveness, rather than to benchmark GIDs as competing attack methods, we conduct it on FOCAL as a representative victim across all five datasets.

We additionally evaluate a set of conventional transformations applied globally to the entire input image, collectively referred to as *Global Image Degradations (GIDs)*. Unlike our query-based attack, each GID is applied once to the original image, without spatial localization or iterative victim-side adaptation. For each degradation, we vary its corresponding strength parameter and select an operating point whose average image quality is comparable to or lower than that of the baselines in our main experiments, thereby providing the GIDs with a sufficiently large perturbation magnitude to avoid underestimating their attack effectiveness. All outputs are clipped to the valid image range $[0, 1]$. The specific implementation and strength parameter of each degradation are described below.

Gaussian blur. We apply channel-wise Gaussian filtering. The degradation strength is mainly controlled by the Gaussian standard deviation σ, while the kernel size is enlarged accordingly to provide sufficient spatial support. Larger σ produces stronger spatial smoothing.

Median filtering. Each pixel is replaced by the median value within a local $k \times k$ neighborhood using reflection padding. We vary the kernel size k to control the filtering strength, where a larger kernel generally removes more local texture and high-frequency forensic traces.

JPEG compression. We recompress the image using JPEG and vary the JPEG quality factor q. A lower quality factor corresponds to stronger compression and larger image degradation.

Rademacher noise. We add independent Rademacher noise to each pixel and channel of the entire image,

$$x'_{c,p,q} = \mathrm{clip}\left(x_{c,p,q} + \epsilon_b b_{c,p,q},\, 0,\, 1\right), \qquad b_{c,p,q} \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}\{-1, +1\}, \tag{25}$$

where ϵ_b controls the perturbation magnitude. Increasing ϵ_b directly increases the maximum additive distortion.

Color jitter. We sequentially perturb image brightness, contrast, and saturation using random multiplicative factors. For a strength parameter s, the corresponding factor is sampled from $[1-s, 1+s]$. We jointly vary the brightness, contrast, and saturation strengths to control the overall color-jitter magnitude.

Resize/resampling. We downsample the input to a fraction of its original spatial resolution and resample it back to the original size. The resize scale controls the degradation strength; smaller intermediate resolutions generally induce stronger resampling artifacts. We use bilinear interpolation unless otherwise specified.

Low-frequency noise. We first sample Gaussian noise $\boldsymbol{n}$ at a reduced spatial resolution and bicubically upsample it to the input resolution. The resulting field is normalized by its maximum absolute value before being scaled by ϵ_{LF}:

$$\boldsymbol{x}' = \mathrm{clip}\left(\boldsymbol{x} + \epsilon_{\mathrm{LF}} \frac{\mathrm{Upsample}(\boldsymbol{n})}{\|\,\mathrm{Upsample}(\boldsymbol{n})\|_\infty}, 0, 1\right). \tag{26}$$

We primarily vary ϵ_{LF} to control perturbation magnitude, while the downsampling ratio determines its spatial frequency: a larger ratio produces a smoother, lower-frequency perturbation.

Brightness, contrast, and saturation. We further evaluate the three photometric transformations independently. Brightness is implemented by globally scaling the image intensity, $\boldsymbol{x}' = \beta\boldsymbol{x}$. Contrast interpolates between the input and its image-wise grayscale mean, whereas saturation interpolates between the RGB image and its grayscale version. Their respective factors control the degradation strength, with values farther from 1 producing stronger transformations.

Stable Diffusion reconstruction. For SD reconstruction, we use Stable Diffusion v1.4 with a DDIM scheduler and 50 diffusion steps. The input is first resized to the diffusion input resolution when necessary, encoded into the latent space, inverted along the DDIM trajectory, and then reconstructed through the reverse trajectory. Both the conditional and unconditional prompts are empty, and the classifier-free guidance scale is set to 1, thereby minimizing prompt-induced semantic changes. The reconstructed image is finally resized back to the original resolution using bilinear interpolation and clipped to $[0, 1]$. No additional additive noise is introduced in this global SD baseline.

These Rademacher noise and low-frequency noise baselines resemble the high-frequency and smooth perturbations used by our engines, respectively, but are applied globally without iterative victim-feedback optimization. They therefore provide a reference for the effect of the noise alone, rather than a controlled comparison of global and local perturbations.

Comparison with TALA. Table 21 reports the complete results of the evaluated GIDs. Although several global degradations substantially reduce localization performance, they remain less effective than TALA while introducing greater perceptual distortion. On FOCAL, using images with initial F1 > 0.1, JPEG compression achieves ASRs of 39%–65% across the five datasets, whereas TALA achieves 86.8%–97.8% with a query budget of $Q = 100$. The GIDs are applied directly without query-based optimization. These JPEG operating points yield PSNR values of approximately 25 dB on most datasets and 31 dB on FaceApp. TALA achieves PSNR above 30 dB on four datasets and 26.53 dB on IMD2020, while also attaining lower LPIPS. Thus, the evaluated global degradations incur greater image distortion yet achieve lower attack success. These results support the benefit of TALA's local perturbations adapted to victim feedback, beyond applying generic degradations across the entire image.

C.5 Additional Ablation Results on Complementary Perturbation Engines

We provide additional analysis of the complementary behavior of E0 and E1 discussed in Sec. 5.3. This diagnostic differs from the aggregate engine ablation in Table 2, which evaluates the overall contribution of each engine under the common evaluation protocol. Here, we instead use all 761 eligible paired samples on FOCAL–CASIAv1 to examine the overlap between the success sets of E0 and E1 at the individual-image level. Because the overlap and engine-specific success cases represent only subsets of the evaluated samples, we use a larger paired sample set to obtain sufficient instances for characterizing their complementary success patterns. In addition, we adjust the perturbation budget of E1 to approximately match the PSNR of E0, enabling comparison of their success sets at a similar overall distortion level. Because of these differences in evaluation samples and perturbation settings, the individual ASRs in Table 6 are not directly comparable to those in the aggregate ablation.

As shown in Table 6, although E1 achieves a higher individual ASR than E0 (70.3% vs. 66.1%), their successful attacks do not fully overlap. Among all eligible images, 5.1% are successfully attacked only by E0, whereas 9.3% are successfully attacked only by E1. Thus, neither engine's success set fully contains that of the other, indicating that their different perturbation constructions can succeed on different inputs. Taking the union of the success sets from two independent runs increases the coverage to 75.4%, further demonstrating this complementarity.

Importantly, the independent-run union is an oracle set statistic rather than a practical engine-selection strategy: it determines success after executing E0 and E1 independently and combining their successful samples. It therefore requires the queries of two separate attack runs. In contrast, adaptive E0+E1 switches between the two engines within a single attack trajectory according to rejection feedback and achieves an even higher ASR of 80.2%. This improvement over the independent-run union suggests that the benefit of adaptation is not limited to selecting whichever engine would succeed independently. Instead, perturbations generated by one engine can change

the current attack state before the other engine is applied, allowing sequential adaptation to exploit interactions that are absent when the two engines are run separately. Together with the aggregate results in the main text, this analysis supports the use of complementary perturbation engines with rejection-driven adaptive switching in the final design.

C.6 Qualitative Results and Visualizations

Figure 6 provides qualitative examples of the adversarial images, post-attack localization responses, and amplified perturbations produced by different attacks. The evaluated methods exhibit substantially different perturbation patterns. Square introduces spatially scattered block-like modifications, while DiffCFG produces broadly distributed fine-grained perturbations. RLGC modifies a large portion of the image and exhibits pronounced structured patterns. AdvFor produces relatively fine-grained perturbations distributed across the image, with additional localized structures in some regions. In contrast, TALA concentrates its perturbations on a smaller number of adaptively selected regions, consistent with its victim-guided spatial search. Despite these localized perturbations, TALA effectively suppresses or misaligns the predicted manipulation regions in both examples. These examples illustrate how TALA can adapt the spatial placement and structure of its perturbations to disrupt localization while avoiding unnecessary modifications over the entire image.

Table 20: Attack parameters used in the main comparison experiments (Tables 7-18). TALA parameter values are rounded to three decimal places for presentation; the exact values are provided in the supplementary code.

			Initial F1 > 0.5		Initial F1 > 0.1	
MaxQ	Victim	Dataset	Square ϵ_2	TALA ϵ_∞	Square ϵ_2	TALA ϵ_∞
50	TruFor	CASIAv1	10	0.059	9	0.071
		COCOGlide	20	0.098	15	0.071
		COVERAGE	30	0.059	15	0.047
		IMD2020	80	0.118	70	0.071
		FaceApp	18	0.098	15	0.118
	IFOSN	CASIAv1	30	0.118	16	0.118
		COCOGlide	30	0.118	15	0.118
		COVERAGE	30	0.118	15	0.040
		IMD2020	50	0.071	50	0.078
		FaceApp	32	0.118	12	0.031
	FOCAL	CASIAv1	15	0.071	7	0.027
		COCOGlide	20	0.071	19	0.118
		COVERAGE	35	0.118	15	0.035
		IMD2020	140	0.040	90	0.035
		FaceApp	10	0.078	6	0.020
100	TruFor	CASIAv1	20	0.098	23	0.098
		COCOGlide	25	0.118	29	0.078
		COVERAGE	40	0.118	35	0.078
		IMD2020	120	0.118	100	0.098
		FaceApp	23	0.071	20	0.118
	IFOSN	CASIAv1	60	0.118	25	0.118
		COCOGlide	40	0.118	30	0.118
		COVERAGE	40	0.118	30	0.098
		IMD2020	80	0.118	100	0.118
		FaceApp	40	0.118	15	0.055
	FOCAL	CASIAv1	15	0.059	20	0.078
		COCOGlide	40	0.138	15	0.071
		COVERAGE	35	0.118	50	0.098
		IMD2020	250	0.118	250	0.118
		FaceApp	15	0.059	10	0.035

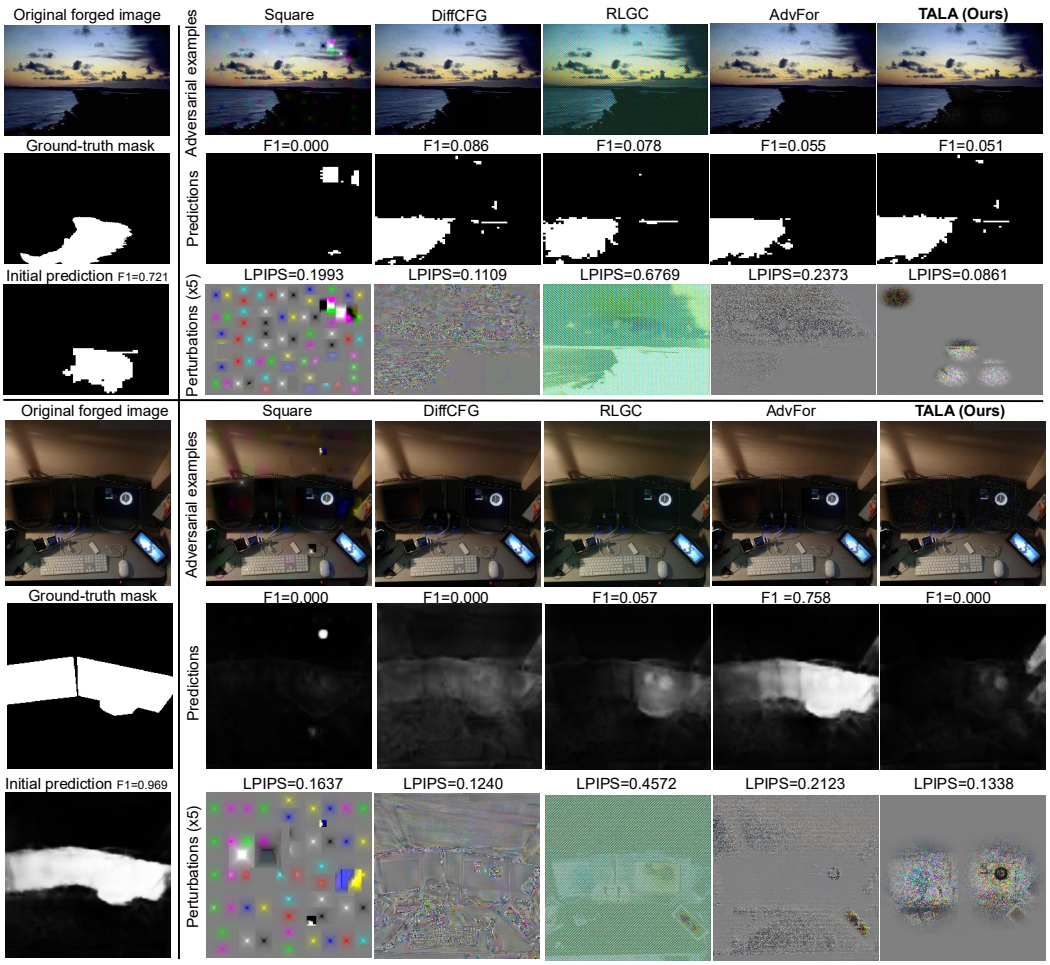


Figure 6: Qualitative comparison of perturbed images and localization responses. The first column shows the original forged image, ground-truth mask, and initial victim prediction. The remaining columns show the perturbed images, post-attack predictions, and $5\times$ amplified perturbations produced by Square, DiffCFG, RLGC, AdvFor and TALA. F1 and LPIPS values are reported for the post-attack predictions and perturbed images, respectively. The top example is from CASIAv1 with FOCAL as the victim, and the bottom example is from COCOGlide with TruFor as the victim.

Table 21: Performance of Global Image Degradations (GIDs) against FOCAL across five datasets, evaluated on images with an initial localization F1> 0.1. We use FOCAL as a representative victim to examine whether generic global degradation can achieve high attack success under a favorable distortion budget.

Dataset	Method	ASR ↑	LPIPS ↓	PSNR ↑	F1 ↓	IoU ↓
CASIAv1	SD Recon.	47.6%	0.2465	24.06	0.019	0.010
	Gaussian Blur	27.0%	0.1962	26.54	0.013	0.007
	Median Filter	47.6%	0.3451	23.59	0.011	0.006
	JPEG	**65.1%**	0.4112	25.15	0.013	0.007
	Rademacher Noise	42.9%	0.3882	22.32	0.025	0.013
	Color Jitter	11.1%	0.0555	21.72	0.022	0.011
	Resize	33.3%	0.2300	26.85	0.020	0.010
	Low-freq. Noise	25.4%	0.3306	**27.45**	0.020	0.010
	Brightness	4.8%	0.0323	18.36	**0.009**	**0.004**
	Contrast	3.2%	**0.0319**	24.31	0.045	0.024
	Saturation	12.7%	0.1452	23.92	0.016	0.008
COCOGlide	SD Recon.	33.3%	0.2775	24.25	0.031	0.016
	Gaussian Blur	27.0%	0.3646	24.52	0.020	0.010
	Median Filter	44.4%	0.3273	24.56	0.023	0.012
	JPEG	**47.6%**	0.4488	25.01	0.035	0.018
	Rademacher Noise	33.3%	0.4386	20.54	0.025	0.013
	Color Jitter	3.2%	**0.0307**	**25.11**	0.078	0.040
	Resize	44.4%	0.3897	23.43	0.032	0.016
	Low-freq. Noise	14.3%	0.3789	24.69	0.022	0.011
	Brightness	6.3%	0.0529	15.87	0.053	0.027
	Contrast	9.5%	0.0488	22.44	0.036	0.019
	Saturation	6.3%	0.1139	23.91	**0.018**	**0.009**
COVERAGE	SD Recon.	40.2%	0.3555	21.08	0.016	0.008
	Gaussian Blur	29.9%	0.2837	25.68	0.014	0.007
	Median Filter	25.3%	0.2768	25.67	0.023	0.012
	JPEG	**49.4%**	0.5446	23.52	0.022	0.011
	Rademacher Noise	41.4%	0.4901	20.34	0.015	0.008
	Color Jitter	5.7%	0.0383	20.04	**0.007**	**0.004**
	Resize	44.8%	0.3414	24.57	0.010	0.005
	Low-freq. Noise	18.4%	0.4321	24.86	0.019	0.010
	Brightness	3.4%	0.0393	15.11	0.010	0.005
	Contrast	8.0%	**0.0296**	23.45	0.017	0.009
	Saturation	4.6%	0.0691	**25.77**	0.029	0.015
IMD2020	SD Recon.	31.2%	0.3218	22.09	0.033	0.017
	Gaussian Blur	21.5%	0.3234	25.16	0.028	0.014
	Median Filter	25.8%	0.3848	24.06	0.038	0.020
	JPEG	**39.8%**	0.5308	24.87	0.032	0.016
	Rademacher Noise	25.8%	0.4754	20.88	0.019	0.010
	Color Jitter	1.1%	0.0837	24.50	**0.001**	**0.000**
	Resize	24.7%	0.3355	24.61	0.029	0.015
	Low-freq. Noise	19.4%	0.4548	**25.69**	0.025	0.013
	Brightness	0.0%	–	–	–	–
	Contrast	2.2%	0.3247	13.11	0.074	0.038
	Saturation	1.1%	**0.0396**	24.83	0.088	0.046
FaceApp	SD Recon.	20.0%	0.2566	26.26	0.038	0.019
	Gaussian Blur	3.6%	0.1727	31.22	**0.034**	**0.017**
	Median Filter	32.7%	0.2695	29.70	0.046	0.024
	JPEG	**54.5%**	0.3217	31.06	0.044	0.023
	Rademacher Noise	32.7%	0.2297	30.14	0.045	0.024
	Color Jitter	3.6%	**0.0039**	31.22	0.074	0.039
	Resize	9.1%	0.1865	**31.52**	0.049	0.026
	Low-freq. Noise	30.9%	0.3225	30.48	0.038	0.020
	Brightness	14.5%	0.0363	16.64	0.040	0.021
	Contrast	12.7%	0.0448	23.46	0.049	0.026
	Saturation	12.7%	0.0240	29.69	0.059	0.031