# Seeing but Not Detecting: Privacy-Preserving Scene Text Attack via Hybrid Adversarial Policy Learning

**Feiran Li**[1] **Jiahao Lyu**[2] **Ke Jiang**[1] **Yu Zhou**[3,†] **Jinqiao Shi**[1,†]

[1]Beijing University of Posts and Telecommunications
[2]Institute of Information Engineering, Chinese Academy of Sciences
[3]Nankai University
feiranli@bupt.edu.cn yzhou@nankai.edu.cn shijinqiao@bupt.edu.cn

## Abstract

Scene text encodes high-density semantic information, raising privacy risks for automated vision systems. Existing privacy-preserving adversarial methods mainly target Scene Text Recognition (STR) by inducing character-level misrecognition, yet text regions remain detectable, enabling manual inspection or downstream recovery. We argue that stronger privacy requires suppressing Scene Text Detection (STD) during localization, hiding text as background to bypass subsequent OCR pipelines. To this end, we propose **AR-Attacker** (**A**ttention-guided **R**einforcement learning text **Attacker**), a black-box attack framework that generates sparse, visually imperceptible perturbations to effectively suppress STD outputs. We formulate the attack as a Markov Decision Process (MDP) and design an attention-guided action space, which leverages a frozen surrogate model to focus perturbations on salient text regions. Furthermore, AR-Attacker employs a discrete-continuous hybrid policy optimized with Proximal Policy Optimization (PPO), which jointly selects perturbation locations and magnitudes. Empirical results show that AR-Attacker surpasses state-of-the-art baselines by 30% in text removal rate while simultaneously reducing query counts by 40%. Remarkably, the framework maintains high visual fidelity (*PSNR* $>$ 45 dB), advancing the "visible to humans, invisible to machines" privacy paradigm through a robust RL-based approach.

## 1 Introduction

In the digital era, scene text images are ubiquitous, widely found in shop billboards, traffic signs, daily photographs, etc. As a key semantic cue in visual environments, text plays a vital role in enabling intelligent systems to perceive the physical world. However, automatic extraction of sensitive information, such as identity numbers, home addresses, and phone numbers, from these images by third-party services or malicious actors poses significant privacy risks [8, 54, 42]. Therefore, mechanisms to safeguard sensitive text from unauthorized detection are increasingly crucial.

To mitigate this risk, recent works have explored adversarial perturbations to protect scene text. Effective scene text understanding generally relies on Scene Text Detection (STD) to localize text regions and Scene Text Recognition (STR) to decode them into machine-readable character sequences. Existing adversarial attacks predominantly target the downstream STR task [49, 13, 19], aiming to induce character-level errors which are easily restorable via semantic context. Furthermore, STR attacks fully localize text regions, producing visually readable but semantically nonsensical outputs against common sense, and thus easily expose the adversarial attack. While effective in certain contexts, we argue that relying solely on STR-level protection is insufficient for comprehensive

†Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

privacy preservation. Therefore, it is essential to investigate STD attacks that suppress text at the detection stage, preventing automated systems from localizing sensitive regions and blocking privacy leakage at its source.

In practical privacy protection scenarios, attackers lack internal model knowledge, making black-box attacks most realistic. Furthermore, STD attacks are more viable than STR attacks in black-box settings, needing only bounding box outputs (universally provided by APIs), while STR attacks require character-level confidence scores rarely exposed commercially. Existing black-box strategies generally comprise transfer-based methods [6], which rely on surrogate models and suffer from weak transferability, and query-based methods [38, 37, 32], which iteratively probe the model but suffer low efficiency, incurring massive redundant queries on irrelevant background pixels. Many methods boost effectiveness with high-contrast noise, violating the imperceptibility needed for privacy. In conclusion, as illustrated by Figure 1, attacking STD presents formidable challenges: (1) attackers should operate under **black box** settings; (2) text regions are sparse, demanding **high efficiency** exploration; (3) perturbations must be **visually imperceptible** to humans; and (4) the detector's localization capability must be suppressed to achieve **machine undetectability**.




Figure 1: Overview of **AR-Attacker**. A black-box STD attack framework achieving superior efficiency, machine undetectability, and visual imperceptibility through attention-guided MDP modeling and PPO optimization.

To tackle these challenges, we propose AR-Attacker, a prior-guided black-box reinforcement learning framework. We model the adversarial generation process as a Markov Decision Process (MDP) and employ Proximal Policy Optimization (PPO) [34] to learn optimal attack strategies through online interactions. Central to our method is a discrete-continuous hybrid action space, which enables the agent to jointly optimize discrete pixel locations and continuous perturbation magnitudes. To further enhance efficiency, the action space is strategically constrained to high-saliency regions via an attention-guided prior, effectively pruning the search space and enabling sparse, targeted modifications. By aligning RL with detection suppression, AR-Attacker effectively eradicates sensitive text from machine perception, improving the text removal rate by 30% and reducing query counts by approximately 40% compared to SOTA, while maintaining superior visual imperceptibility with PSNR consistently above 45 dB. Our contributions can be summarized as follows:

- We propose AR-Attacker, the first query-based black-box framework to block automated extraction at the source by suppressing text detection. It uses reinforcement learning to optimize perturbations via online interactions with detector outputs only.
- We introduce an attention-guided action space and discrete-continuous hybrid actions, enabling efficient and targeted perturbations that significantly improve query efficiency and attack success.
- We introduce a rigorous "machine undetectability" objective, where an attack succeeds only when no bounding boxes are detected, ensuring no downstream tasks are triggered.
- Our method marks a big step towards machine-invisible but human-readable attacks. Extensive experiments demonstrate that AR-Attacker outperforms SOTA in text detection suppression and query efficiency, while preserving high visual imperceptibility ($PSNR > 45$ dB).

## 2 Related Work

### 2.1 Adversarial Attacks on Scene Text

Most research on scene text privacy has focused on STR adversarial attacks. Xu et al. [49] proposed a white-box strategy that iteratively updates perturbations to disrupt both CTC-based and attention-based STR. AD$^2$E [50] optimizes Differential Evolution (DE) to accelerate the optimization process in black-box attacks. ProTegO [13] exploited the human visual flicker-fusion effect, using high-frequency alternating complementary patterns to mask adversarial textures. More recently, Li et al.

[19] decompose global optimization into sub-objectives to address overly concentrated perturbation pixels. However, these methods only induce character-level misrecognition, leaving text regions localized and visible. Therefore, it is essential to investigate STD attacks that prevent sensitive information from being localized at the source. However, existing STD attacks are predominantly transfer-based. Wu et al. [47] utilize pixel-level probability maps to target both anchor-based and anchor-free detectors, while UDUP [5] generates universal adversarial backgrounds to disrupt text localization. These methods depend on the architectural similarity between the surrogate and target models, highlighting the need for a strict black-box method requiring no prior model knowledge.

### 2.2 Scene Text Detector

Scene text detector contributes to localizing texts in scene images. Early CNN-based methods [10, 14, 22, 23] followed the object detection method to detect the scene text. [11] and its variants [48, 7, 51] represent the oriented and curve text contours. EAST [55] is an end-to-end anchor-free method adopting pixel-level regression techniques to handle multi-oriented text instances. [29, 28, 43, 44, 25, 39] also explore the efficient representation operators. Given the success of Transformer in vision tasks, the fundamental architecture has been gradually embedded in the scene text detector [31, 41, 25, 2]. [31] first introduces the Deformable DETR [56] to detect oriented text instances. To alleviate the burden of the Transformer, [41] samples the features of text instances and reduces the computational cost. [52] notices the reading of inverse-like scene text. [2] combines the advantages of top-down and bottom-up and proposes a transformer-powered text detector.

### 2.3 Query-based Attacks

Query-based black-box attacks are increasingly practical for real-world scenarios where attackers only have query access to model outputs. While query-based attacks have been extensively studied for image classification [38, 9, 32], object detection [20, 21, 37] and STR [50], their application to STD remains underexplored. OnePixel [38] generates adversarial sample images using the Differential Evolution (DE) [4]. RFPAR [37] leverages a remember-and-forget reinforcement learning framework [40] to suppress detections with minimal pixel modifications. Xu et al. [50] propose an adaptive discrete differential evolution algorithm that protects text privacy by perturbing an extremely limited number of pixels. However, these methods typically perform a global search across the entire image. Due to the characteristic sparsity of text regions within natural scene images, such a global search strategy is inefficient, leading to a substantial number of redundant queries and slow convergence.

## 3 Method

This section details AR-Attacker. We first formulate the generation of adversarial examples as MDP in Sec. 3.1. Then, in Sec. 3.2, we present an overview of the overall framework. Subsequently, we detail two key technical innovations: the attention-guided action space for efficient search (Sec. 3.3) and the policy learning mechanism for handling discrete-continuous hybrid actions (Sec. 3.4).

### 3.1 Problem Formulation as MDP

We study the problem of query-based black-box attacks against STD. Given a clean image $x$ and a target detector $D$ that outputs text bounding boxes with confidence scores, our goal is to generate an adversarial example $x'$ such that $D(x')$ produces no detections while satisfying two practical constraints: (i) the perturbation is imperceptible, and (ii) the number of queries to $D$ is minimized.

To achieve this under black-box conditions, we formulate the attack as a Markov Decision Process (MDP) defined by tuple $(S, A, P, R, \gamma)$, where the state $s_t \in S$ captures the current attack status at step $t$ through the triple $s_t = (x'_t, M, D(x'_t))$ which includes the current adversarial image $x'_t$, a binary attention mask $M$ indicating the salient text regions and detector output $D(x'_t)$; the action $a_t \in A$ specifies pixel locations and perturbation magnitudes within $M$; the progress-aware state transition $P$ is deterministic and only accepts improvements (i.e., reduced detection confidence or box count); and the reward $r_t \in R$ quantifies this improvement. Our objective is to learn an attack

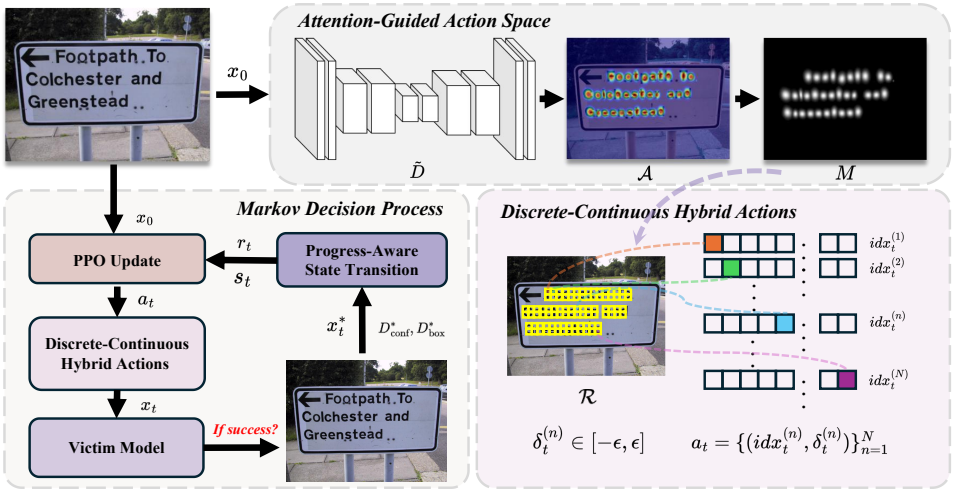



Figure 2: The Overall Pipeline of AR-Attacker. The process consists of three main stages: 1) A clean image is first processed by an attention mechanism to identify salient text regions; 2) A hybrid policy network iteratively selects pixels and perturbation magnitudes; 3) The victim model provides feedback to drive a progress-aware state transition $(P)$ and reward calculation for PPO policy updates. The process repeats until no text boxes are detected or the step limit is reached.

policy $\pi_\theta(a_t \mid s_t)$ that maximizes cumulative reward over $T$ steps.

$$J(\theta) = \mathbb{E}_{\pi_\theta}\Big[\sum_{t=0}^{T} \gamma^t r_t\Big], \tag{1}$$

where $\gamma \in (0, 1]$ is the discount factor controlling the trade-off between short-term and long-term gains. We employ PPO [34] to optimize $\pi_\theta$, with further details in A.2. PPO's clipped objective function ensures stable and reliable policy updates, which is vital given the high cost per query to $D$.

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathbb{E}_t\Big[\min\big(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-c, 1+c)\hat{A}_t\big)\Big], \tag{2}$$

where $r_t(\theta) = \dfrac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$ and $c$ is the clipping parameter. We estimate advantages $\hat{A}_t$ with Generalized Advantage Estimation (GAE) [33].

### 3.2 Overall Framework

The proposed AR-Attacker is an attention-guided reinforcement learning method for black-box adversarial attacks for STD. As shown in Figure 2, the framework is designed as a sequential decision-making process that learns to strategically perturb an image through iterative interactions based solely on detector outputs.

The process begins with a clean input image $x_0$ through an attention mechanism to identify salient text regions, yielding a sparse set of candidate pixels (in Sec. 3.3). At each step, it observes the current state $s_t = (x'_t, M, D(x'_t))$, comprising the latest adversarial image and the detector's output, and uses a policy network to decide which pixels to modify and how much to perturb them. This hybrid action mechanism (in Sec. 3.4), combines discrete selection with continuous perturbation strength. After applying the perturbation and querying the victim detector, a progress-aware transition rule $(P)$ is employed to update the state, accepting a candidate next state $s^*_{t+1}$ only if it demonstrably degrades the detector's performance.

$$s^*_{t+1} = f(s_t, a_t) = \big(x^*_{t+1},\ M,\ D(x^*_{t+1})\big) \tag{3}$$

where $x^*_{t+1}$ is the image $x_t$ after applying $a_t$, $f(\cdot)$ is the deterministic transition function that constructs the next candidate state (see Sec. 3.4 for details).

We measure the improvement in detection performance before and after applying the action $a_t$ as:

$$\begin{aligned} \Delta_{t+1} D_{\text{conf}} &= D_{\text{conf}}(x'_t) - D_{\text{conf}}(x^*_{t+1}), \\ \Delta_{t+1} D_{\text{box}} &= D_{\text{box}}(x'_t) - D_{\text{box}}(x^*_{t+1}) \end{aligned} \tag{4}$$

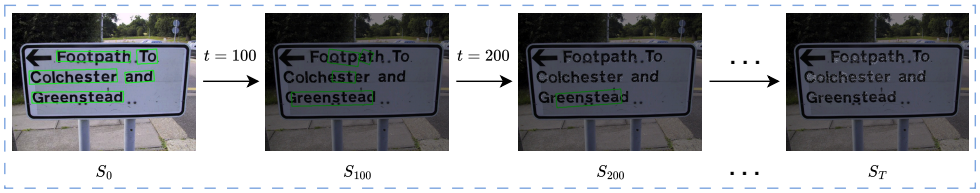



Figure 3: Visualization of the attack process by AR-Attacker at different time steps. To make the adversarial perturbations more visually apparent, we enhance the contrast of the perturbation maps.

where $D_{\text{conf}}(\cdot)$ and $D_{\text{box}}(\cdot)$ respectively denote the aggregated confidence of the detected boxes and the total number of boxes returned by $D(\cdot)$.

The state transition is accepted only if the action yields a positive improvement:

$$x'_{t+1} = \begin{cases} x^*_{t+1}, & \text{if } \Delta_{t+1} D_{\text{conf}} > 0 \text{ or } \Delta_{t+1} D_{\text{box}} > 0, \\ x'_t, & \text{otherwise.} \end{cases} \tag{5}$$

This provides the agent with clear and high-quality learning signals, guiding it to explore truly effective attack patterns while avoiding blind exploration of the state space due to ineffective perturbations.

Then a corresponding reward is provided to guide policy updates. We define immediate reward $r_t$ as the sum of the reductions in both detection metrics, as defined in Eq. 4:

$$r_t = \Delta_{t+1} D_{\text{conf}} + \lambda \cdot \Delta_{t+1} D_{\text{box}} \tag{6}$$

where $\lambda$ is a hyperparameter to balance $\Delta_{t+1} D_{\text{conf}}$ and $\Delta_{t+1} D_{\text{box}}$.

The process continues until no text boxes are detected or the step limit is reached, as detailed in Algorithm 1 in Appendix B.

### 3.3 Attention-Guided Action Space

Scene text is typically sparse and localized and occupies only a fraction of the image. Conducting blind global searches in such high-dimensional pixel space is highly inefficient, as most perturbations often fall on irrelevant background pixels. To address this challenge, we introduce an attention-guided mechanism that strategically confines adversarial perturbations to semantically critical text regions.

Specifically, we employ a surrogate model $\tilde{D}$ to generate an attention map $\mathcal{A}$ using Grad-CAM [35], which effectively highlights text regions in the image. The generated attention map $\mathcal{A}$ is then binarized into a mask $M$ based on a predefined threshold $\tau$:

$$M_{i,j} = \begin{cases} 1, & \text{if } \mathcal{A}_{i,j} \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where $i, j$ denotes the spatial coordinates of a pixel and the threshold $\tau$ is set to $0.3$.

The action space is then restricted exclusively to the set of foreground pixel coordinates:

$$\mathcal{R} = \{(i, j) \mid M_{i,j} = 1\} \tag{8}$$

which is sparse and spatially fragmented, resulting in a discontinuous and irregular search space.

### 3.4 Policy Learning with Discrete-Continuous Hybrid Actions

We formulate the attack action as selecting a set of $N$ pixels from the candidate region $\mathcal{R}$ and applying continuous perturbations to them, formally defined as $a_t = \{(a^{(n)}_{t,1}, a^{(n)}_{t,2})\}_{n=1}^{N}$, where $a^{(n)}_{t,1}$ denotes the spatial index of the $n$-th perturbed pixel within the constrained action space $\mathcal{R}$, and $a^{(n)}_{t,2}$ represents the corresponding perturbation magnitude applied to that pixel. However, the spatially discrete nature of the candidate set $\mathcal{R}$ presents a fundamental challenge for policy learning, as conventional reinforcement learning algorithms are designed for continuous or simple discrete action spaces.

To handle the discrete nature of $\mathcal{R}$, we avoid modeling a categorical probability distribution on all candidate pixels. Specifically, we apply the following transformations:

$$\begin{aligned} \tilde{u}_t^{(n)} &= \text{sigmoid}\left(a_{t,1}^{(n)}\right) \in [0,1], \\ \delta_t^{(n)} &= \epsilon \cdot \tanh\left(a_{t,2}^{(n)}\right) \in [-\epsilon, \epsilon] \end{aligned} \tag{9}$$

$\mathcal{R}$ is flattened to an ordered list of size $K = |\mathcal{R}|$, where each index uniquely corresponds to a foreground pixel $(i, j)$. The continuous index $idx_t^{(n)}$ is mapped to a discrete pixel index by:

$$idx_t^{(n)} = \lfloor \tilde{u}_t^{(n)} \cdot K \rfloor \tag{10}$$

Finally, the adversarial image is generated by sequentially accumulating the selected $N$ perturbations in the current state. Let $x_{t,0}^* = x_t'$. For each action $n \in \{1, \ldots, N\}$, the targeted pixel $(i^{(n)}, j^{(n)})$ is updated and immediately clipped:

$$x_{t,n}^*[i^{(n)}, j^{(n)}] = \text{Clip}\left(x_{t,n-1}^*[i^{(n)}, j^{(n)}] + \delta_t^{(n)},\ L^{(n)},\ U^{(n)}\right) \tag{11}$$

where the $\text{Clip}(\cdot)$ function ensures the pixel value strictly remains within the valid image range $[0, 1]$ and the $L_\infty$ perturbation budget $\epsilon$ relative to the original clean image $x_0$:

$$L^{(n)} = \max(0, x_0[i^{(n)}, j^{(n)}] - \epsilon), \quad U^{(n)} = \min(1, x_0[i^{(n)}, j^{(n)}] + \epsilon) \tag{12}$$

After applying $N$ pixel modifications in action $a_t$, the next state is $x_{t+1}^* = x_{t,N}^*$. Figure 3 illustrates the progressive effect of AR-Attacker in attack steps. At each step, a hybrid action selects pixels and sets perturbation magnitudes, gradually weakening detector response. This design enables effective navigation of discrete text regions while maintaining precise control over perturbation intensity.

# 4 Experiments

## 4.1 Experiment Settings

**Datasets and Models.** To comprehensively evaluate the effectiveness of our method, we conduct extensive experiments on three benchmark datasets that encompass diverse text scenarios: **ICDAR 2013** [15] for horizontal English text, **Total-Text** [3] for curved and multi-oriented text, and **ReCTS** [53] for Chinese street view text. This diverse selection enables a comprehensive evaluation of the method's generalization ability across varying text geometries, languages, and scene complexities. For ReCTS, we randomly sampled 1,000 test images. We further benchmark four representative STD models, including a regression-based model (e.g., TextBoxes++ [23]), two segmentation-based models (e.g., DBNet [24], PSENet [45]), and a transformer-based model (e.g., DPText-DETR [52]).

**Implementation Details.** In our experiments, the maximum number of attack steps is set to $T = 1000$. At each step, $N = \lfloor \alpha \cdot K \rfloor$ pixels are perturbed, where $K$ is the number of candidate pixels in the attention-guided action space obtained from the DPText-based mask $M$ [52], and $\alpha$ range from 0.001 to 0.005. The per-pixel perturbation magnitude is bounded by $\epsilon = 20/255$. We set the hyperparameters $\lambda$ to 0.5. PPO training details and hyperparameters are provided in the Appendix A.1. All experiments are conducted on a single NVIDIA A100 GPU with 40GB of graphical memory.

**Evaluation Metrics.** Precision (P), Recall (R), and F-measure (F) are standard metrics for evaluating STD performance, which we use to assess the general impact of our attack. To quantify the extent of box suppression, we report the Text Removal Rate (TR), defined as the relative reduction in the number of detected bounding boxes. Following our machine undetectability definition, we introduce the Strict Success Rate (SSR): an attack succeeds only when the detector returns zero bounding boxes. Additionally, Query Count (Q) measures search efficiency in the black-box setting. Finally, visual quality is quantified via Peak Signal-to-Noise Ratio (PSNR) and Mean Opinion Score (MOS), where 20 participants rated 100 random adversarial images per dataset 1-5 (detailed in Appendix D).

## 4.2 Comparison with STD Attacks

We compare AR-Attacker with the only two existing attack methods specifically tailored for STD [47, 5]. Both follow a transfer-based paradigm, utilizing CRAFT [1] surrogate to generate adversarial examples that are applied to four representative detectors across three datasets for evaluation.

Table 1: Comparison with state-of-the-art STD attack methods.

| Model | Method | ICDAR 2013 | | | | | Total-Text | | | | | ReCTS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | P ↓ | R ↓ | F↓ | TR ↑ | SSR ↑ | P ↓ | R ↓ | F ↓ | TR ↑ | SSR ↑ | P ↓ | R ↓ | F ↓ | TR ↑ | SSR ↑ |
| Textbox++ [23] | Wu et al. [47] | 0.63 | 0.70 | 0.67 | 0.22 | 0.07 | 0.72 | 0.74 | 0.73 | 0.18 | 0.05 | 0.83 | 0.90 | 0.86 | 0.10 | 0.05 |
| | UDUP [5] | 0.58 | 0.62 | 0.60 | 0.31 | 0.12 | 0.58 | 0.60 | 0.59 | 0.25 | 0.10 | 0.55 | 0.49 | 0.52 | 0.18 | 0.09 |
| | Ours | **0.22** | **0.35** | **0.27** | **0.81** | **0.63** | **0.31** | **0.24** | **0.27** | **0.79** | **0.61** | **0.39** | **0.18** | **0.24** | **0.75** | **0.58** |
| DBNet [24] | Wu et al. [47] | 0.69 | 0.74 | 0.71 | 0.18 | 0.06 | 0.74 | 0.69 | 0.71 | 0.16 | 0.03 | 0.83 | 0.78 | 0.81 | 0.16 | 0.04 |
| | UDUP [5] | 0.55 | 0.65 | 0.60 | 0.33 | 0.15 | 0.56 | 0.68 | 0.62 | 0.28 | 0.11 | 0.56 | 0.72 | 0.63 | 0.25 | 0.09 |
| | Ours | **0.29** | **0.26** | **0.28** | **0.78** | **0.59** | **0.35** | **0.18** | **0.24** | **0.76** | **0.58** | **0.31** | **0.24** | **0.27** | **0.72** | **0.55** |
| PSENet [45] | Wu et al. [47] | 0.71 | 0.76 | 0.73 | 0.15 | 0.06 | 0.78 | 0.60 | 0.68 | 0.12 | 0.04 | 0.77 | 0.70 | 0.73 | 0.13 | 0.04 |
| | UDUP [5] | 0.61 | 0.65 | 0.63 | 0.25 | 0.12 | 0.65 | 0.70 | 0.68 | 0.25 | 0.09 | 0.62 | 0.66 | 0.64 | 0.22 | 0.11 |
| | Ours | **0.27** | **0.25** | **0.26** | **0.80** | **0.61** | **0.30** | **0.21** | **0.25** | **0.77** | **0.59** | **0.35** | **0.18** | **0.24** | **0.72** | **0.56** |
| DPText [52] | Wu et al. [47] | 0.78 | 0.74 | 0.76 | 0.15 | 0.03 | 0.84 | 0.62 | 0.71 | 0.17 | 0.04 | 0.85 | 0.74 | 0.79 | 0.12 | 0.04 |
| | UDUP [5] | 0.68 | 0.72 | 0.70 | 0.21 | 0.09 | 0.71 | 0.75 | 0.73 | 0.20 | 0.08 | 0.70 | 0.78 | 0.74 | 0.17 | 0.06 |
| | Ours | **0.37** | **0.21** | **0.27** | **0.71** | **0.55** | **0.40** | **0.16** | **0.23** | **0.70** | **0.52** | **0.47** | **0.21** | **0.29** | **0.67** | **0.53** |

Table 2: Comparison with state-of-the-art query-based attack methods.

| Model | Method | ICDAR 2013 | | | Total-Text | | | ReCTS | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | TR ↑ | SSR ↑ | Q ↓ | TR ↑ | SSR ↑ | Q ↓ | TR ↑ | SSR ↑ | Q ↓ |
| Textbox++ [23] | $AD^2E$ [50] | 0.38 | 0.21 | 3842 | 0.33 | 0.20 | 3791 | 0.37 | 0.24 | 3655 |
| | RFPAR [37] | 0.53 | 0.40 | 1226 | 0.57 | 0.43 | 1413 | 0.59 | 0.45 | 1670 |
| | MPCS [19] | 0.43 | 0.30 | 2533 | 0.39 | 0.27 | 2601 | 0.36 | 0.23 | 2423 |
| | Ours | **0.81** | **0.63** | **794** | **0.79** | **0.61** | **875** | **0.75** | **0.58** | **896** |
| DBNet [24] | $AD^2E$ [50] | 0.33 | 0.18 | 3577 | 0.30 | 0.15 | 3612 | 0.35 | 0.20 | 3639 |
| | RFPAR [37] | 0.46 | 0.37 | 1458 | 0.47 | 0.35 | 1431 | 0.53 | 0.42 | 1492 |
| | MPCS [19] | 0.36 | 0.20 | 2694 | 0.39 | 0.23 | 2531 | 0.36 | 0.22 | 2613 |
| | Ours | **0.78** | **0.59** | **729** | **0.76** | **0.58** | **811** | **0.72** | **0.55** | **895** |
| PSENet [45] | $AD^2E$ [50] | 0.35 | 0.20 | 3632 | 0.34 | 0.18 | 3715 | 0.34 | 0.15 | 3704 |
| | RFPAR [37] | 0.66 | 0.49 | 1344 | 0.54 | 0.42 | 1697 | 0.62 | 0.46 | 1563 |
| | MPCS [19] | 0.42 | 0.28 | 2566 | 0.40 | 0.25 | 2583 | 0.35 | 0.19 | 2712 |
| | Ours | **0.80** | **0.61** | **698** | **0.77** | **0.59** | **846** | **0.72** | **0.56** | **879** |
| DPText-DETR [52] | $AD^2E$ [50] | 0.33 | 0.15 | 3825 | 0.31 | 0.15 | 3874 | 0.28 | 0.13 | 3952 |
| | RFPAR [37] | 0.41 | 0.33 | 1632 | 0.45 | 0.35 | 1690 | 0.55 | 0.43 | 1796 |
| | MPCS [19] | 0.39 | 0.22 | 2607 | 0.35 | 0.21 | 2685 | 0.31 | 0.17 | 2736 |
| | Ours | **0.71** | **0.55** | **873** | **0.70** | **0.52** | **899** | **0.67** | **0.53** | **913** |

As summarized in Table 1, our method consistently outperforms prior methods across all detectors and datasets. Wu et al. and UDUP are highly surrogate-dependent and transfer poorly to other architectures, especially to transformer-based DPText (F-measure $> 70\%$). In contrast, AR-Attacker sharply reduces P, R, and F, illustrating that our attack disrupts core detection functionality. Crucially, the superiority of our method is even more evident in TR and SSR that directly reflect the effectiveness of privacy protection. Our method achieves a TR exceeding 70% on average, which is a substantial improvement (averaging over 40% absolute gain). Furthermore, SSR consistently exceed 50% across all settings, which represents the most challenging goal. In stark contrast, baselines perform poorly, achieving SSR values mostly below 15% and sometimes as low as 3%. These results demonstrate that AR-Attacker overcomes the weak transferability and surrogate-dependency of prior methods, remaining architecture-agnostic. Using online RL for direct feedback, it suppresses the output of bounding boxes systematically, offering a stronger solution for real-world privacy protection.

### 4.3 Comparison with Query-based Attacks

Table 2 compares AR-Attacker with query-based attacks adapted for the STD setting. Our method significantly outperforms baselines across all metrics, delivering a superior balance of attack potency and efficiency. AR-Attacker achieves substantial TR and breakthrough SSR, particularly on the more challenging DPText model, representing a nearly two-fold improvement over most baselines in terms of machine undetectability. Furthermore, a key advantage is its significantly lower query consumption. While $AD^2E$ requires over 3,500 queries and MPCS exceeds 2,500, our method reduces the average query count to under 900, representing an approximately 40% reduction compared to RFPAR. This advantage is further reinforced under limited budgets (Table 4), our method can achieve an almost 0.5 text removal rate, nearly doubling the performance of RFPAR. As the budget increases to 2000, our method further widens the performance gap. The exceptional performance of AR-Attacker is attributed to two key design innovations. First, the attention-guided mechanism that restricts the action space to salient text regions. Second, by formulating the attack as an MDP and employing a

discrete-continuous hybrid policy, the agent learns to optimize perturbation patterns through direct online feedback, effectively avoiding the inefficient and redundant global blind searching.

As shown in Appendix C.3, AR-Attacker achieves an approximate $10\times$ speedup over traditional optimizers (DE, NES, BO) and cuts wall-clock time by roughly 50% compared to SOTA RFPAR, requiring only 2.62 minutes per image. 700 to 900 queries represent the known optimal efficiency level under the setting of this problem. Moreover, since privacy protection is typically an offline preprocessing step before sharing images with third-party services, a per-image optimization cost of a few minutes is practically acceptable. More comparisons with traditional black-box optimizers are provided in Appendix C.2.

### 4.4 Evaluation of Visual Imperceptibility

To evaluate the "human-visible" aspect of our framework, we compare AR-Attacker with SOTA methods using objective (PSNR) and subjective (MOS) metrics. As summarized in Table 3, AR-Attacker consistently achieves better image fidelity, maintaining PSNR above 46 dB and MOS exceeding 4.4 across all models and datasets, outperforming the best baseline by approximately 15–20 dB. As illustrated in Figure 4, while prior methods introduce noticeable global artifacts or high-frequency noise that disrupts natural appearance of the scene text, AR-Attacker generates sparse, strategically localized perturbations that are nearly indistinguishable from original clean images.

Table 3: Comparison of visual imperceptibility among different attack methods. PSNR (dB) and MOS represent objective and subjective visual assessments.

| Model | Method | ICDAR 2013 | | Total-Text | |
|---|---|---|---|---|---|
| | | PSNR ↑ | MOS ↑ | PSNR ↑ | MOS ↑ |
| Textbox++ [23] | Wu et al. [47] | 34.62 | 3.61 | 32.43 | 3.55 |
| | AD$^2$E [50] | 35.28 | 3.54 | 34.59 | 3.41 |
| | RFPAR [37] | 30.25 | 2.88 | 29.77 | 2.78 |
| | Ours | **48.53** | **4.80** | **47.28** | **4.65** |
| DBNet [24] | Wu et al. [47] | 34.62 | 3.61 | 32.43 | 3.55 |
| | AD$^2$E [50] | 33.58 | 3.39 | 34.23 | 3.44 |
| | RFPAR [37] | 34.33 | 2.82 | 31.96 | 2.75 |
| | Ours | **48.58** | **4.78** | **48.33** | **4.72** |
| PSENet [45] | Wu et al. [47] | 34.62 | 3.61 | 32.23 | 3.55 |
| | AD$^2$E [50] | 36.05 | 3.86 | 36.29 | 3.75 |
| | RFPAR [37] | 28.32 | 2.60 | 29.45 | 2.83 |
| | Ours | **48.94** | **4.76** | **47.99** | **4.58** |
| DPText-DETR [52] | Wu et al. [47] | 34.62 | 3.61 | 32.23 | 3.55 |
| | AD$^2$E [50] | 37.18 | 3.74 | 36.93 | 3.60 |
| | RFPAR [37] | 28.65 | 2.38 | 28.80 | 2.52 |
| | Ours | **46.81** | **4.52** | **47.55** | **4.46** |



(a) Clean (b) Wu et al. (c) RFPAR (d) Ours

Figure 4: Visualization of adversarial examples (Top: ICDAR2013; Bottom: Total-Text). More results are in Appendix: Figures 6 and 7.

### 4.5 Ablation Study

In this section, we conduct comprehensive ablation experiments on three key components: the attention mechanism, the surrogate model $\tilde{D}$, and the pixel attack rate $\alpha$. Further hyperparameter ablations are detailed in Appendix ( C.4, C.5 and C.6).

**Effect of the Attention Mechanism.** To validate the contribution of our attention-guided perturbation strategy, we compare the full method against a variant without attention that perturbs pixels uniformly at random. Since the absence of attention leads to substantially less effective perturbations and thus requires more queries to suppress text detections, we increase the maximum number of attack steps ($T$) to 3,000 for the w/o Att variant. As shown in Figure 5, the attention-based approach consistently achieves higher TR with significantly fewer queries (Q). Specifically, our method consistently achieves TR > 0.70 across all detectors with 500–1,000 queries, while the random variant fails to reach TR = 0.5 even with 2,500–3,000 queries. This dramatic performance gap demonstrates that the attention mechanism plays a crucial role in focusing perturbations on semantically relevant areas and improving overall attack efficiency.

**Effect of Surrogate Model Choice.** A key design choice is using a surrogate model to generate an attention mask that guides perturbations. As shown in Table 5, attack performance correlates positively with surrogate accuracy. The high-precision DPText-DETR surrogate yields the strongest suppression, as it provides a more precise attention mask that guides the PPO agent to focus on the

most sensitive semantic regions. Notably, even a weaker surrogate (e.g., CRAFT) maintains robust performance, indicating that the RL policy can still discover effective strategies despite imperfect guidance. Crucially, the surrogate model serves only to generate a spatial prior that narrows the RL agent's search space, and its selection is entirely independent of the victim's architecture. Even a CNN-based surrogate (CRAFT) successfully attacks a Transformer-based victim (DPText-DETR), highlighting the exceptional cross-model generalization of our framework in black-box scenarios.

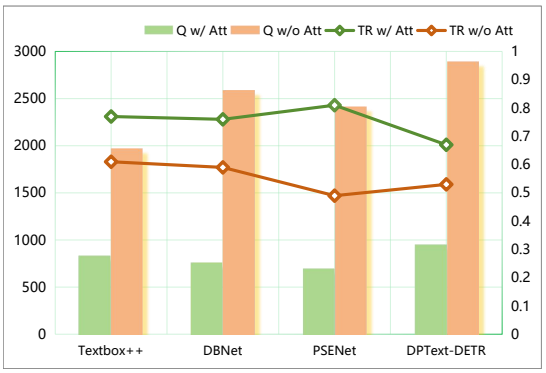



Figure 5: Ablation study of the attention mechanism. The bar chart illustrates the average query counts (Q) for attacks with attention (w/ Att) versus without attention (w/o Att) across various text detection models. The line chart presents the corresponding text removal rates (TR). All experiments are conducted on the ICDAR2013 dataset.

**Effect of Pixel Attack Rate.** As shown in Table 6, increasing $\alpha$ from 0.001 to 0.005 improves TR from 0.65 to 0.80 (DBNet) and 0.60 to 0.76 (DPText-DETR), indicating that perturbing more informative pixels enhances attack effectiveness. Higher $\alpha$ reduces query count by accelerating convergence but lowers PSNR, reflecting the trade-off between attack efficiency and imperceptibility. A value of $\alpha = 0.003$ offers a favorable balance, achieving strong performance while maintaining high visual quality.

Table 4: Performance comparison between AR-Attacker and RFPAR under various maximum query constraints in ICDAR 2013.

| Max Q | Method | DBNet | | | DPText-DETR | | |
|---|---|---|---|---|---|---|---|
| | | TR ↑ | Q ↓ | PSNR ↑ | TR ↑ | Q ↓ | PSNR ↑ |
| 500 | RFPAR | 0.27 | 478 | 35.63 | 0.21 | 486 | 34.79 |
| | Ours | 0.57 | 486 | 49.58 | 0.46 | 492 | 49.69 |
| 1000 | RFPAR | 0.29 | 967 | 33.98 | 0.29 | 974 | 32.11 |
| | Ours | 0.78 | 729 | 48.58 | 0.71 | 873 | 46.81 |
| 2000 | RFPAR | 0.46 | 1822 | 28.88 | 0.41 | 1834 | 27.65 |
| | Ours | 0.81 | 1649 | 42.54 | 0.79 | 1702 | 41.85 |

Table 5: Impact of surrogate model choice on attack performance across victim models (Total-Text). $F_{\text{sur}}$ denotes the F-measure of the surrogate model itself on clean images.

| **Surrogate** ($\tilde{D}$) | $F_{\text{sur}}$ (Clean) | DBNet F ↓ / TR ↑ | PSENet F ↓ / TR ↑ | DPText-DETR F ↓ / TR ↑ |
|---|---|---|---|---|
| CRAFT | 0.83 | 0.36 / 0.69 | 0.30 / 0.73 | 0.31 / 0.66 |
| DBNet | 0.85 | 0.28 / 0.73 | 0.26 / 0.73 | 0.29 / 0.69 |
| PSENet | 0.84 | 0.30 / 0.72 | 0.29 / 0.75 | 0.30 / 0.66 |
| **DPText-DETR** | **0.89** | **0.24 / 0.76** | **0.25 / 0.77** | **0.23 / 0.70** |

Table 6: Ablation study on the pixel attack rate $\alpha$.

| $\alpha$ | DBNet | | | DPText-DETR | | |
|---|---|---|---|---|---|---|
| | TR ↑ | Q ↓ | PSNR ↑ | TR ↑ | Q ↓ | PSNR ↑ |
| $\text{Ours}_{0.001}$ | 0.65 | 856 | 50.67 | 0.60 | 994 | 49.12 |
| $\text{Ours}_{0.002}$ | 0.72 | 831 | 50.12 | 0.66 | 947 | 48.45 |
| $\text{Ours}_{0.003}$ | 0.76 | 758 | 48.33 | 0.67 | 944 | 45.08 |
| $\text{Ours}_{0.004}$ | 0.77 | 733 | 42.20 | 0.71 | 886 | 43.28 |
| $\text{Ours}_{0.005}$ | 0.80 | 738 | 40.05 | 0.76 | 822 | 41.77 |

### 4.6 Robustness against Defense Mechanisms

To evaluate the practical resilience of AR-Attacker, we investigate its performance against common defensive image pre-processing techniques, including JPEG compression ($q = 70$), Gaussian blurring, and data augmentation (e.g., rotation, scaling and cropping). As shown in Table 7, while defenses impact performance, our attack remains effective. While JPEG compression and blurring attempt to erase high-frequency adversarial perturbations, the victim model's performance still remains suppressed. Notably, under JPEG compression, AR-Attacker achieves a TR of 52%, surpassing the undefended RFPAR baseline (TR=0.47) reported in Table 2. The attack also shows strong robustness to spatial transformations (TR=0.69). This demonstrates that perturbations target semantically critical regions rather than superficial noise, ensuring robustness under realistic defenses. More detailed evaluations regarding robustness against stronger defenses are provided in Appendix C.9.

Table 7: Performance of AR-Attacker under various defensive image pre-processing techniques (Victim: DBNet, Dataset: Total-Text).

| Defense Method | P ↓ | R ↓ | F ↓ | TR ↑ |
|---|---|---|---|---|
| No Defense | 0.35 | 0.18 | 0.24 | 0.76 |
| JPEG Compression ($q = 70$) | 0.53 | 0.58 | 0.55 | 0.52 |
| Gaussian Blur | 0.41 | 0.34 | 0.37 | 0.64 |
| Data Augmentation | 0.38 | 0.25 | 0.30 | 0.69 |

## 5 Conclusion

This paper presents AR-Attacker, an RL-based black-box adversarial attack framework designed to suppress scene text detection for privacy protection. Unlike traditional methods that target character recognition, AR-Attacker shifts the paradigm toward detection suppression, blocking automated extraction at the localization stage. By combining reinforcement learning with an attention-guided perturbation mechanism and discrete–continuous hybrid actions, our method efficiently generates sparse, targeted perturbations through online interactions with the target model. Empirical results show that AR-Attacker surpasses SOTA baselines by 30% in text removal rate and 40% in query efficiency while preserving high visual imperceptibility. AR-Attacker establishes a robust, architecture-agnostic foundation for privacy-preserving vision systems.

**Broader Impacts.** The positive social impact of our work lies in providing a practical mechanism to protect sensitive textual information from unauthorized automated extraction. While the adversarial framework could potentially be misused to bypass legitimate security or monitoring systems, its primary intent is defensive. Furthermore, by exposing current detector vulnerabilities, this study encourages the development of more robust STD models via adversarial training. We advocate for the ethical application of this technology, prioritizing individual privacy rights in the digital era.

## 6 Acknowledgments

This work was supported by the BUPT - China Unicom Joint Innovation Project (2025-STHZ-BJYDDX), the National Natural Science Foundation of China (Grant Nos. 62376266 and 62406318) and the CAAI-Ant Group Research Fund.

## References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[2] Qingwen Bu, Sungrae Park, Minsoo Khang, and Yichuan Cheng. SRFormer: Text detection transformer with incorporated segmentation and regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 855–863, 2024.

[3] Chee Kheng Chng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 International Conference on Document Analysis and Recognition*, volume 1, pages 935–942. IEEE, 2017.

[4] Swagatam Das and Ponnuthurai Nagaratnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.

[5] JiaCheng Deng, Li Dong, Jiahao Chen, Diqun Yan, Rangding Wang, Dengpan Ye, Lingchen Zhao, and Jinyu Tian. Universal defensive underpainting patch: Making your text invisible to optical character recognition. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7559–7568, 2023.

[6] Yang Deng, Weibin Wu, Jianping Zhang, and Zibin Zheng. Blurred-dilated method for adversarial attacks. In *Annual Conference on Neural Information Processing Systems*, 2023.

[7] Bo Du, Jian Ye, Jing Zhang, Juhua Liu, and Dacheng Tao. I3CL: Intra-and inter-instance collaborative learning for arbitrary-shaped scene text detection. *International Journal of Computer Vision*, 130(8):1961–1977, 2022.

[8] Jun Du, Chunxiao Jiang, Kwang-Cheng Chen, Yong Ren, and H. Vincent Poor. Community-structured evolutionary game for privacy protection in social networks. *IEEE Transactions on Information Forensics and Security*, 13(3):574–589, 2018.

[9] Yan Feng, Baoyuan Wu, Yanbo Fan, Li Liu, Zhifeng Li, and Shutao Xia. Boosting black-box attack with partially transferred conditional adversarial distribution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[10] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localization in natural images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[13] Yanru He, Kejiang Chen, Guoqiang Chen, Zehua Ma, Kui Zhang, Jie Zhang, Huanyu Bian, Han Fang, Weiming Zhang, and Nenghai Yu. Protego: Protect text content against ocr extraction attack. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7424–7434, 2023.

[14] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116:1–20, 2016.

[15] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *International Conference on Document Analysis and Recognition*, pages 1484–1493, 2013.

[16] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. ICDAR 2015 competition on robust reading. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 1156–1160, 2015.

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, October 2023.

[18] Annu Lambora, Kunal Gupta, and Kriti Chopra. Genetic algorithm- a literature review. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing*, pages 380–384, 2019.

[19] Jingyu Li, Pengwen Dai, Mingqing Zhu, Chengwei Wang, Haolong Liu, and Xiaochun Cao. Towards irreversible attack: Fooling scene text recognition via multi-population coevolution search. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[20] Siyuan Liang, Baoyuan Wu, Yanbo Fan, Xingxing Wei, and Xiaochun Cao. Parallel rectangle flip attack: A query-based black-box attack against object detection. In *IEEE International Conference on Computer Vision*, 2021.

[21] Siyuan Liang, Longkang Li, Yanbo Fan, Xiaojun Jia, Jingzhi Li, Baoyuan Wu, and Xiaochun Cao. A large-scale multiple-objective method for black-box attack against object detection. In *European Conference on Computer Vision*, 2022.

[22] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[23] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE Transactions on Image Processing*, 27(8):3676–3690, 2018.

[24] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11474–11481, 2020.

[25] Ruijin Liu, Ning Lu, Dapeng Chen, Cheng Li, Zejian Yuan, and Wei Peng. Pbformer: Capturing complex scene text shape with polynomial band Transformer. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2112–2120, 2023.

[26] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[27] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recogn.*, 90:337–345, 2019.

[28] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. ABCNet: Real-time scene text spotting with adaptive bezier-curve network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9809–9818, 2020.

[29] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *European Conference on Computer Vision*, pages 20–36, 2018.

[30] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*, 2018.

[31] Zobeir Raisi, Mohamed A Naiel, Georges Younes, Steven Wardell, and John S Zelek. Transformer-based text detection in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 3162–3171, 2021.

[32] Soumyendu Sarkar, Ashwin Ramesh Babu, Sajad Mousavi, Sahand Ghorbanpour, Vineet Gundecha, Antonio Guillen, Ricardo Luna, and Avisek Naug. Robustness with query-efficient adversarial attack using reinforcement learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2330–2337, 2023.

[33] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *Computer ence*, 2015.

[34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[35] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, October 2019. ISSN 1573-1405.

[36] Satya Narayan Shukla, Anit Kumar Sahu, Devin Willmott, and Zico Kolter. Simple and efficient hard label black-box adversarial attacks in low query budget regimes. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, page 1461–1469, 2021.

[37] Dongsu Song, Daehwa Ko, and Jay Hoon Jung. Amnesia as a catalyst for enhancing black box pixel attacks in image classification and object detection. In *Annual Conference on Neural Information Processing Systems*, volume 37, pages 87159–87183, 2024.

[38] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[39] Yuchen Su, Zhineng Chen, Zhiwen Shao, Yuning Du, Zhilong Ji, Jinfeng Bai, Yong Zhou, and Yu-Gang Jiang. Lranet: Towards accurate and efficient scene text detection with low-rank approximation network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4979–4987, 2024.

[40] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Annual Conference on Neural Information Processing Systems*, volume 12, 1999.

[41] Jingqun Tang, Wenqing Zhang, Hongye Liu, MingKun Yang, Bo Jiang, Guanglong Hu, and Xiang Bai. Few could be better than all: Feature sampling and grouping for scene text detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4563–4572, 2022.

[42] Xianhao Tian, Peijia Zheng, and Jiwu Huang. Robust privacy-preserving motion detection and object tracking in encrypted streaming video. *IEEE Transactions on Information Forensics and Security*, 16:5381–5396, 2021.

[43] Fangfang Wang, Yifeng Chen, Fei Wu, and Xi Li. TextRay: Contour-based geometric modeling for arbitrary-shaped scene text detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 111–119, 2020.

[44] Wei Wang, Yu Zhou, Jiahao Lv, Dayan Wu, Guoqing Zhao, Ning Jiang, and Weipinng Wang. TPSNet: Reverse thinking of thin plate splines for arbitrary shape scene text representation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5014–5025, 2022.

[45] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9336–9345, 2019.

[46] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, and Jürgen Schmidhuber. Natural evolution strategies, 2011.

[47] Shudeng Wu, Tao Dai, Guanghao Meng, Bin Chen, Jian Lu, and Shu-Tao Xia. Transferable adversarial attacks for deep scene text detection. In *Proceedings of the International Conference on Pattern Recognition*, pages 8945–8951, 2021.

[48] Shanyu Xiao, Liangrui Peng, Ruijie Yan, Keyu An, Gang Yao, and Jaesik Min. Sequential deformation for accurate scene text detection. In *European Conference on Computer Vision*, pages 108–124, 2020.

[49] Xing Xu, Jiefu Chen, Jinhui Xiao, Lianli Gao, Fumin Shen, and Heng Tao Shen. What machines see is not what they get: Fooling scene text recognition models with adversarial text images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12301–12311, 2020.

[50] Yikun Xu, Pengwen Dai, Zekun Li, Hongjun Wang, and Xiaochun Cao. The best protection is attack: Fooling scene text recognition with minimal pixels. *IEEE Transactions on Information Forensics and Security*, 18:1580–1595, 2023.

[51] Jian Ye, Zhe Chen, Juhua Liu, and Bo Du. Textfusenet: Scene text detection with richer fused features. In *International Joint Conferences on Artificial Intelligence*, pages 516–522, 2020.

[52] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Bo Du, and Dacheng Tao. DPText-DETR: Towards better scene text detection with dynamic points in Transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3241–3249, 2023.

[53] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun Yang, and et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *2019 International Conference on Document Analysis and Recognition*, pages 1577–1581. IEEE, 2019.

[54] Shuaitao Zhang, Yuliang Liu, Lianwen Jin, Yaoxiong Huang, and Songxuan Lai. Ensnet: Ensconce text in the wild. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[55] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: an efficient and accurate scene text detector. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.

[56] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

# A Implementation Details

## A.1 PPO Hyperparameter Configuration

We train our policy using Proximal Policy Optimization (PPO) [34] with a set of hyperparameters selected based on empirical validation and standard practices in the RL literature. The complete configuration is summarized in Table 8.

Table 8: PPO Hyperparameter Configuration.

| **PPO Hyperparameter** | **Value** |
|---|---|
| Feature Extractor | ResNet-18 |
| Learning Rate (Actor) | 1e-4 |
| Learning Rate (Critic) | 3e-4 |
| Optimizer | Adam |
| Discount Factor ($\gamma$) | 0.99 |
| GAE Parameter ($\lambda$) | 0.95 |
| Clip Range ($c$) | 0.20 |
| Epochs per Update | 10 |
| Entropy Coefficient | 0.01 |
| Value Function Coefficient | 0.50 |

## A.2 PPO Policy Optimization

We provide a detailed schematic of our policy network, which is built on a ResNet-18 backbone. The input image $x'_t$ is first processed by the backbone to extract high-level visual features, which are then flattened and passed through a shared fully-connected layer with ReLU activation. The resulting feature vector branches into two separate heads: an actor head that predicts the mean of a bivariate Gaussian distribution over two continuous action dimensions: An **actor head** that predicts the mean $\mu_t = (\mu_{t,1}, \mu_{t,2})$ of a bivariate Gaussian policy over two continuous action dimensions: $\mu_{t,1}$ corresponds the perturbation location, and $\mu_{t,2}$ represents the perturbation magnitude. And a **critic head** that estimates the state value $V(s_t)$.

The standard deviation $\sigma_t = (\sigma_{t,1}, \sigma_{t,2})$ of the Gaussian policy is parameterized via a learnable log-standard-deviation vector $\log \sigma_t$, shared across all states for training stability. To generate the $N$ individual pixel-level actions $A_t = \{a_{t,n}\}_{n=1}^{N}$ at step $t$, we draw $N$ independent samples from this bivariate distribution:

$$a_{t,n} \sim \mathcal{N}\left(\mu_t, \mathrm{diag}(\sigma_t^2)\right), \quad \forall n \in \{1, 2, \ldots, N\},$$

where each $a_{t,n}$ is a bivariate vector representing a target pixel index candidate and a perturbation amplitude.

Since these $N$ actions are sampled independently given the state $s_t$, the joint probability $P(A_t|s_t)$ is the product of their individual probabilities. Accordingly, the joint log-probability $\log \pi_\theta(A_t|s_t)$ is computed as the sum of the log-probabilities of all sampled actions:

$$\log \pi_\theta(A_t|s_t) = \sum_{n=1}^{N} \log \pi_\theta(a_{t,n}|s_t).$$

This joint log-probability is then directly incorporated into the standard PPO clipped surrogate objective function to compute the importance sampling ratio.

The policy is trained using PPO to stabilize updates by limiting step size through a clipped surrogate objective. Let $\pi_\theta$ denote the current policy and $\pi_{\theta_{\text{old}}}$ the policy used to collect the data. The PPO objective is defined as:

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathbb{E}_t \left[ \min \left( r_t(\theta)\hat{A}_t,\ \text{clip}\big(r_t(\theta), 1-c, 1+c\big)\hat{A}_t \right) \right],$$

where $r_t(\theta) = \frac{\pi_\theta(A_t|s_t)}{\pi_{\theta_{\text{old}}}(A_t|s_t)} = \exp\left(\log \pi_\theta(A_t|s_t) - \log \pi_{\theta_{\text{old}}}(A_t|s_t)\right)$ is the importance sampling ratio, $c$ is a clipping hyperparameter, and $\hat{A}_t$ is the estimated advantage using Generalized Advantage Estimation (GAE). which balances bias and variance via exponentially weighted multi-step returns. Given

a trajectory of rewards $r_t, r_{t+1}, \ldots$ and value estimates $V(s_t), V(s_{t+1}), \ldots$, the GAE estimator at time step $t$ is defined as:

$$\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta_{t+l},$$

where the temporal difference (TD) residual is

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t),$$

$\gamma \in [0, 1]$ is the discount factor, and $\lambda \in [0, 1]$ controls the bias-variance trade-off (we use $\gamma = 0.99$, $\lambda = 0.95$).

The total loss combines the PPO actor loss, a value function (critic) loss, and an entropy bonus to encourage exploration:

$$\mathcal{L}(\theta) = \mathcal{L}^{\mathrm{PPO}}(\theta) - \beta_1 \mathcal{L}^{\mathrm{VF}}(\theta) + \beta_2 \mathcal{L}^{\mathrm{Entropy}}(\theta),$$

where $\mathcal{L}^{\mathrm{VF}} = \mathbb{E}_t\big[(V(s_t) - V_t^{\mathrm{target}})^2\big]$ is the mean squared error between predicted and target values, $\mathcal{L}^{\mathrm{Entropy}} = \mathbb{E}_t\big[-\sum_{n=1}^{N} \log \pi_\theta(a_{t,n}|s_t)\big]$, and $\beta_1, \beta_2$ are hyperparameters weighting the value function loss and the entropy bonus, respectively.

### A.3 Attention Mask Generation

To focus adversarial perturbations on text-relevant regions while operating in a black-box setting, we generate an attention mask for each input image using a fixed surrogate model. We use DPText-DETR [52] as our surrogate model. This mask is used solely to define candidate pixel locations for perturbation and does not require access to the internal structure or gradients of the target scene text detector. Given an input image $x \in \mathbb{R}^{3\times H\times W}$, we first feed it into the surrogate model, which outputs character-level score maps. We then apply Grad-CAM [35] on the final convolutional layer of DPText-DETR to produce a coarse localization map highlighting regions that most strongly contribute to the presence of text. Concretely,

$$y_c = \sum_{i,j,k} \mathcal{S}_{i,j,k}$$

denote the total activation over all spatial positions and channels of the DPText-DETR's output score map $\mathcal{S}$, serving as a scalar proxy for "text confidence". The Grad-CAM weights are computed as:

$$\alpha_k = \frac{1}{Z} \sum_{i,j} \frac{\partial y_c}{\partial A_{k,i,j}},$$

where $A_k$ is the $k$-th channel of the last convolutional feature map and $Z$ is the number of spatial elements. The raw attention map is obtained by:

$$\mathcal{A}^{\mathrm{raw}} = \mathrm{ReLU}\left(\sum_k \alpha_k A_k\right).$$

We resize $\mathcal{A}^{\mathrm{raw}}$ to the original image resolution via bilinear interpolation and normalize its values to $[0, 1]$. During attack execution, we restrict perturbations to regions where the attention score exceeds a threshold $\tau = 0.3$, i.e., the set of valid pixel locations is defined as

$$\mathcal{R} = \{(i, j) \,|\, \mathcal{A}_{i,j} > 0.3\}.$$

The entire pipeline—including DPText-DETR inference and gradient computation—is implemented in PyTorch and runs on GPU. Critically, the surrogat model is frozen and used only once per image to generate the attention prior, no fine-tuning or adaptation to the target detector is performed. This ensures compatibility with strict black-box query-based evaluation protocols.

## B Detailed Algorithmic Procedure

**Algorithm 1** AR-Attacker

**Input:** Clean image $x$, victim STD $D$, surrogate model $\tilde{D}$, max steps $T$, pixels per step $N$, perturbation budget $\epsilon$
**Output:** Adversarial image $x'$

Initialize: $x'_0 \leftarrow x, t \leftarrow 0$
Generate attention map $\mathcal{A} \leftarrow$ Grad-CAM$(\tilde{D}, x'_0)$
Binarize mask: $M_{i,j} \leftarrow \mathbb{I}[\mathcal{A}_{i,j} \geq \tau]$
Build candidate set $\mathcal{R} \leftarrow \{(i,j) \mid M_{i,j} = 1\}$
Flatten to index: $K \leftarrow |\mathcal{R}|$
**for** $t = 0$ **to** $T - 1$ **do**
  $x_{temp} \leftarrow x'_t$
  State: $s_t = (x'_t, M, D(x'_t))$
  Actions: $a_t = \{(a^{(n)}_{t,1}, a^{(n)}_{t,2})\}_{n=1}^{N}$
  **for** $n = 1$ **to** $N$ **do**
    $\tilde{u}^{(n)}_t = \text{sigmoid}(a^{(n)}_{t,1})$ (Eq. 9)
    $\delta^{(n)}_t = \epsilon \cdot \tanh(a^{(n)}_{t,2})$
    $idx^{(n)}_t = \lfloor \tilde{u}^{(n)}_t \cdot K \rfloor$ (Eq. 10)
    $(i^{(n)}, j^{(n)}) = \mathcal{R}[idx^{(n)}_t]$
    $x_{temp}[i^{(n)}, j^{(n)}] = x_{temp}[i^{(n)}, j^{(n)}] + \delta^{(n)}_t$ (Eq. 11)
    Clip $x_{temp}[i^{(n)}, j^{(n)}]$ to valid image range $[0, 1]$ and the $L_\infty$ perturbation budget $\epsilon$. (Eq. 12)
  **end for**
  Set the candidate next state as $x^*_{t+1} = x_{temp}$
  Query detector: $D^*_{\text{conf}}, D^*_{\text{box}} \leftarrow D(x^*_{t+1})$
  **if** $D^*_{\text{box}} = 0$ **then**
    **return** $x^*_{t+1}$ {Attack succeeded}
  **end if**
  Compute improvement (Eq. 4):
  Progress-aware state transition (Eq. 5)
  Compute reward: $r_t = \Delta_{t+1} D_{\text{conf}} + \lambda \cdot \Delta_{t+1} D_{\text{box}}$
  Store transition $(s_t, a_t, r_t, s_{t+1})$ for PPO update
**end for**
**return** $x'$

Table 9: Attack performance of AR-Attacker against different black-box victim models.

| Victim Model | TR ↑ | Q ↓ | PSNR ↑ | SSIM ↑ |
|---|---|---|---|---|
| EAST [55] | 0.83 | 792 | 47.89 | 0.95 |
| PANet [26] | 0.79 | 815 | 47.02 | 0.93 |
| CRAFT [1] | 0.82 | 731 | 48.51 | 0.97 |
| MaskRCNN [12] | 0.75 | 899 | 46.03 | 0.95 |
| TextSnake [29] | 0.76 | 870 | 45.76 | 0.93 |
| ABCNet [28] | 0.73 | 942 | 45.65 | 0.92 |
| EasyOCR | 0.72 | 866 | 46.35 | 0.95 |

# C Additional Experimental Results

## C.1 Attack Performance Across Diverse Black-Box Victim Models

To validate the generalizability of AR-Attacker in realistic black-box scenarios, we evaluate it against five widely used text detection models as victims, while keeping the surrogate model fixed as DPText-DETR [52] and $\tau = 0.3$. All victim models are treated as true black boxes—no architecture or gradient information is used during attack. The results on ICDAR2013 are summarized in Table 9.

AR-Attacker consistently achieves high text removal rates in all victims, with TR ranging from 72% to 83%. Notably, even for contour-sensitive models like TextSnake and ABCNet, which explicitly model

Table 10: Comparison of AR-Attacker with traditional black-box optimization algorithms under the same attention-guided constraint (Dataset: ICDAR 2013, Max Query Budget: 10,000).

| Model | Method | TR ↑ | Q ↓ | PSNR (dB) ↑ |
|---|---|---|---|---|
| Textbox++ [23] | DE [4] | 0.53 | 8240 | 36.12 |
| | NES [46] | 0.58 | 6692 | 36.77 |
| | GA [18] | 0.54 | 9143 | 35.96 |
| | BO [36] | 0.49 | 9651 | 33.28 |
| | **Ours** | **0.91** | **2874** | **45.13** |
| DBNet [24] | DE [4] | 0.51 | 9433 | 32.85 |
| | NES [46] | 0.56 | 7209 | 35.16 |
| | GA [18] | 0.49 | 9532 | 33.57 |
| | BO [36] | 0.43 | 9784 | 32.63 |
| | **Ours** | **0.88** | **3047** | **44.85** |
| PSENet [45] | DE [4] | 0.65 | 9185 | 34.60 |
| | NES [46] | 0.69 | 6927 | 36.58 |
| | GA [18] | 0.41 | 9233 | 32.34 |
| | BO [36] | 0.57 | 9606 | 30.92 |
| | **Ours** | **0.89** | **3112** | **44.37** |
| DPText-DETR [52] | DE [4] | 0.37 | 9461 | 30.64 |
| | NES [46] | 0.46 | 7889 | 35.18 |
| | GA [18] | 0.43 | 9855 | 31.27 |
| | BO [36] | 0.29 | 9936 | 28.95 |
| | **Ours** | **0.82** | **3595** | **42.19** |

text geometry—the attack remains effective (TR $\geq$ 73%), suggesting that localized perturbations within salient regions can disrupt both bounding-box and curve-based detectors. Moreover, all adversarial examples maintain high visual fidelity: PSNR exceeds 45.6 dB, and SSIM remains above 0.92, indicating minimal perceptual distortion.

We further include EasyOCR, a widely adopted open-source OCR system that integrates its own detection module as a representative of real-world deployed pipelines. AR-Attacker achieves a 72% text removal rate against EasyOCR with strong visual fidelity (PSNR = 46.35 dB, SSIM = 0.95), demonstrating its practical relevance beyond academic detector benchmarks. This result underscores that our method generalizes well to OCR systems commonly used in industrial applications.

### C.2 Comparison with Alternative Black-Box Optimization Algorithms

To further validate the superiority of our reinforcement learning-based approach, we compared AR-Attacker against several classical black-box optimization algorithms: Differential Evolution (DE) [4], Natural Evolution Strategy (NES) [46], Genetic Algorithm (GA) [18], and Bayesian Optimization (BO) [36]. To ensure a fair comparison, all baseline algorithms were constrained to the exact same attention-guided search space (i.e., the DPText-based mask) as AR-Attacker. The experiments were conducted on the ICDAR 2013 dataset with a uniformly set maximum query budget of 10,000.

As shown in Table 10, AR-Attacker overwhelmingly outperforms traditional black-box optimizers in all metrics when evaluated under the exact same attention-guided constraint. Traditional evolutionary and Bayesian methods struggle with the highly sparse space, often falling into local optima or wasting massive queries on ineffective random mutations. Consequently, they yield significantly lower TR (TR < 0.60) and suffer from severe visual degradation (PSNR drops to 28–36 dB) due to the accumulation of redundant, high-frequency noise. In contrast, our RL-based approach leverages progress-aware MDP feedback to systematically joint-optimize pixel selections and perturbation magnitudes. This highly directed search strategy not only effectively doubles the attack success rate and halves the query consumption compared to the best baseline (NES), but also strictly minimizes unnecessary modifications to consistently maintain superior visual imperceptibility.

Table 11: Comparison of average query counts and wall-clock time per image (Dataset: ICDAR 2013, Victim Detector: DBNet).

| Method | Queries (Q) ↓ | Avg. Time (mins) ↓ |
|---|---|---|
| DE [4] | 9433 | 25.06 |
| NES [46] | 7209 | 19.33 |
| BO [36] | 9784 | 26.80 |
| RFPAR (RL-based) [37] | 1458 | 5.58 |
| Ours | **729** | **2.62** |

### C.3 Evaluation of Computational Efficiency

To evaluate the practical efficiency and computational cost of our method, we measured the average wall-clock time required to generate a successful adversarial example per image. All evaluations were executed on a single NVIDIA A100 GPU with 40GB of graphical memory. To ensure a fair comparison, traditional optimization algorithms were evaluated under the exact same attention-mask constraint (i.e., the DPText-based mask) as our method.

As shown in Table 11, compared to traditional zeroth-order optimization methods like DE, NES, and BO, AR-Attacker achieves a 10x speedup in wall-clock time. While traditional search methods struggle with the high-dimensional search space (even when masked) and require thousands of queries to estimate gradients or explore candidates. Even compared to the SOTA RFPAR, which also utilizes a RL-based framework with the "remember and forget" search mechanism, AR-Attacker reduces the query count and wall-clock time by 50%. This efficiency gain stems from our discrete-continuous hybrid action space and PPO's stable policy updates. Given the massive improvement in Text Removal rate and visual quality as reported in our paper, we believe that this test-time optimization is a justifiable and efficient trade-off for robust privacy protection.

**Parallel Scalability and Memory Efficiency.** Although optimizing a single image requires 700–900 sequential queries, the computational overhead per query is exceptionally low. For instance, a sequential single-image rollout maintains a highly efficient memory footprint, utilizing only about 27% of VRAM on a 40GB A100 GPU. For large-scale batch processing, the time overhead of these serial queries can be effectively mitigated through parallelization, where attack tasks for different images are distributed and executed simultaneously across multiple parallel workers, multiple GPUs, or multiple batch streams on a single GPU.

### C.4 Effect of Perturbation Magnitude

In Table 12, we examine the influence of the perturbation magnitude $\epsilon$, ranging from $8/255$ to $24/255$. Performance improves substantially as $\epsilon$ increases: on DBNet, TR rises from 0.44 at $8/255$ to 0.78 at $20/255$, while the required query count simultaneously decreases. This trend indicates that allowing a larger perturbation budget enables more effective disruption of detector predictions and accelerates convergence, thereby improving query efficiency. Increasing $\epsilon$ to $24/255$ yields only marginal TR gains but noticeably degrades visual quality (PSNR = 43.69 dB) and slightly increases query consumption, likely due to over-perturbation that introduces redundant or detrimental noise. Overall, the results show that $\epsilon = 20/255$ strikes the best balance, jointly optimizing attack effectiveness, query efficiency, and perceptual fidelity.

### C.5 Effect of Attention Threshold $\tau$

To analyze the sensitivity of AR-Attacker to the attention threshold $\tau$, we evaluate its performance across a range of values from $0.1$ to $0.9$. As summarized in Table 13, $\tau$ acts as a critical regulator of the action space. At lower thresholds (e.g., $\tau = 0.1$), the attack covers a broader region, yielding a high Text Removal rate but at the cost of excessive queries ($Q = 863$). Conversely, as $\tau$ increases beyond $0.5$, the action space becomes overly restrictive, leading to a precipitous drop in TR to $0.26$ at $\tau = 0.9$. Notably, we observe an unconventional trend: while increasing $\tau$ initially improves efficiency, an excessively high threshold (e.g., $\tau \geq 0.7$) leads to a rebound in query count $Q$. Consequently, the PPO agent struggles to find a valid adversarial path within these limited candidates, leading to more failed iterations and a higher number of interactions with the victim model. In contrast, $\tau = 0.3$

Table 12: Ablation study on the maximum perturbation magnitude $\epsilon$.

| $\epsilon$ | DBNet | | | DPText-DETR | | |
|---|---|---|---|---|---|---|
| | TR ↑ | Q ↓ | PSNR ↑ | TR ↑ | Q ↓ | PSNR ↑ |
| Ours$_{8/255}$ | 0.44 | 981 | 48.23 | 0.46 | 977 | 48.11 |
| Ours$_{12/255}$ | 0.68 | 956 | 47.54 | 0.61 | 915 | 46.22 |
| Ours$_{16/255}$ | 0.71 | 821 | 47.64 | 0.67 | 908 | 46.88 |
| Ours$_{20/255}$ | 0.78 | 729 | 48.58 | 0.71 | 873 | 46.81 |
| Ours$_{24/255}$ | 0.79 | 735 | 43.69 | 0.76 | 815 | 45.53 |

strikes the optimal balance. It provides a sufficiently focused yet expressive action space that allows the agent to identify effective perturbations with minimal exploration. This phenomenon can be attributed to the over-constrained search space. When $\tau$ is too high, the attention mask becomes overly sparse, filtering out not only background noise, but also potential vulnerable pixels necessary for a successful attack. In particular, $\tau = 0.3$ emerges as the optimal balance, achieving a peak TR of $0.78$ while significantly reducing the query count to $729$ and maintaining a high PSNR of $48.58$ dB. This suggests that a moderately calibrated attention mask successfully filters out redundant background pixels while preserving enough vulnerable pixels for the PPO agent to exploit, thereby maximizing both attack potency and search efficiency.

Table 13: Ablation study on attention threshold $\tau$.

| $\tau$ | TR ↑ | Q ↓ | PSNR ↑ |
|---|---|---|---|
| 0.1 | 0.78 | 863 | 46.94 |
| 0.3 | 0.78 | 729 | 48.58 |
| 0.5 | 0.70 | 733 | 48.95 |
| 0.7 | 0.59 | 781 | 47.82 |
| 0.9 | 0.26 | 829 | 48.66 |

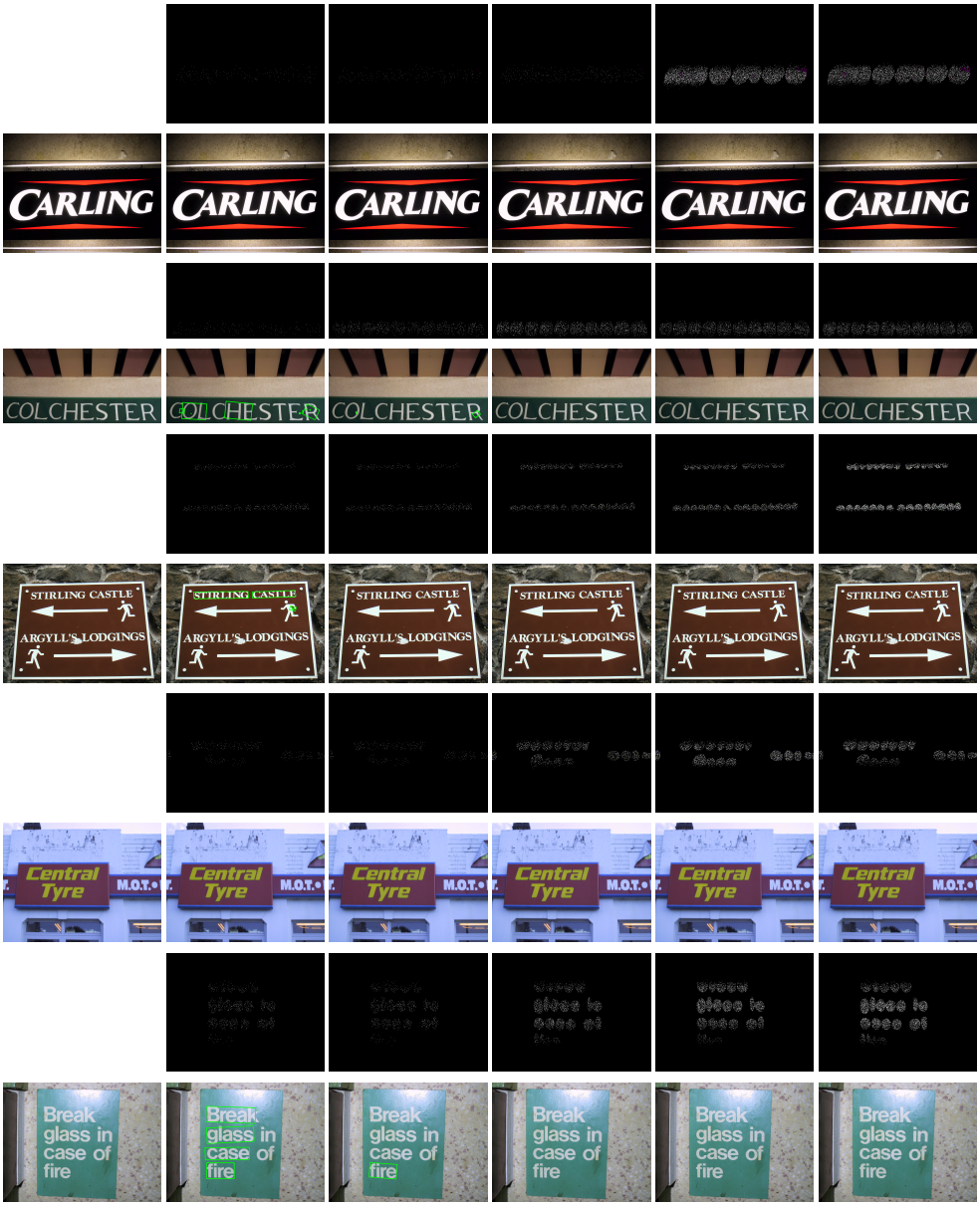
### C.6 Different Methods for Extracting Attention Mechanisms

We evaluate the effectiveness of different attention mechanisms in guiding our RL agent. As shown in Table 14, using Grad-CAM [35] derived from DPText-DETR [52] as a surrogate attention source outperforms the Segment Anything Model (SAM) [17] across all metrics. Specifically, the Grad-CAM guided strategy reduces the Recall (R) to 0.16 and elevates TR rate to 0.70.

Specifically, SAM focuses on geometric completeness and object boundaries, providing a mask of all potential text pixels regardless of their contribution to the detector's decision-making process. However, adversarial attacks are most efficient when they perturb critical decision pixels regions that the detector's localization head relies on most heavily. Since SAM is agnostic to the victim detector's internal features, it fails to reflect the model-specific saliency, leading to a blind perturbation strategy that wastes query budget on semantically redundant regions. In contrast, Grad-CAM (even from a surrogate model) captures the detector's attention, allowing the RL agent to prioritize regions that yield the maximum suppression impact.

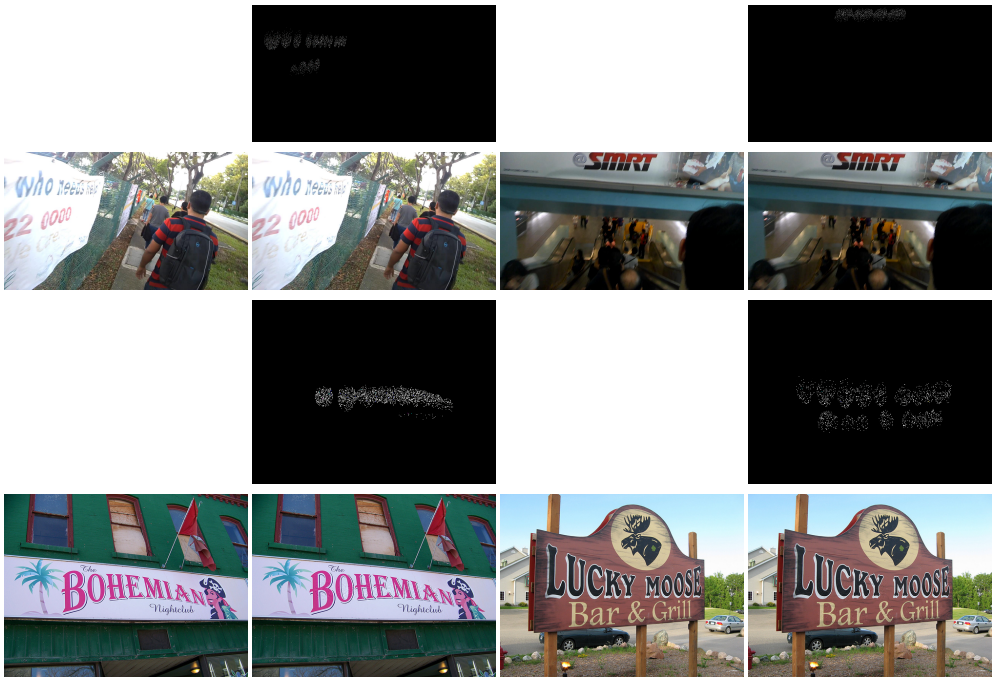
Table 14: Performance of different strategies for extracting attention mechanisms (Victim: DPText-DETR, Dataset: Total-Text).

| Method | P ↓ | R ↓ | F ↓ | TR ↑ |
|---|---|---|---|---|
| SAM [17] | 0.45 | 0.38 | 0.41 | 0.63 |
| Grad-CAM(DPText-DETR) [35] | 0.40 | 0.16 | 0.23 | 0.70 |




(a) Clean Image (b) $\alpha = 0.001$ (c) $\alpha = 0.002$ (d) $\alpha = 0.003$ (e) $\alpha = 0.004$ (f) $\alpha = 0.005$

Figure 6: Visualization of adversarial examples generated with different pixel attack rate $\alpha$. Each row corresponds to one original image and its adversarial counterparts under varying $\alpha$ values and the first column shows the clean input image.




(a) Clean Image (b) Adv Image (c) Clean Image (d) Adv Image

Figure 7: Cross-dataset visualization of successful adversarial attacks: clean vs. adversarial images from ICDAR 2015 (top) and CTW1500 (bottom).

### C.7 Effect of Pixel Attack Rate $\alpha$

To investigate the impact of the pixel attack rate $\alpha$, which controls the magnitude of perturbations applied at each query step—we generate adversarial examples using AR-Attacker with $\alpha$ ranging from 0.001 to 0.005. For each input image, we visualize both the resulting adversarial image and the corresponding perturbation map ($\delta$). As shown in Figure 6, small values of $\alpha$ (e.g., 0.001) produce subtle, visually imperceptible changes that are often sufficient to fool the detector, while larger values lead to more aggressive perturbations that may compromise visual fidelity but ensure higher attack success. This qualitative analysis aligns with our quantitative results, confirming that a moderate $\alpha$ (e.g., 0.003) offers the best trade-off between effectiveness and stealth.

### C.8 Cross-Dataset Generalization

To evaluate the cross-dataset generalizability of our method, we test our method on two challenging datasets with distinct text characteristics: (1) **ICDAR 2015** [16], which contains incidental scene text with arbitrary orientations; (2) **CTW1500** [27], which features densely annotated curved and multi-oriented text instances.

Table 15: Cross-dataset attack performance.

| Model | ICDAR 2015 | | | | CTW1500 | | | |
|---|---|---|---|---|---|---|---|---|
| | P ↓ | R ↓ | F↓ | PSNR ↑ | P ↓ | R ↓ | F ↓ | PSNR ↑ |
| Textbox++ [23] | 0.36 | 0.31 | 0.33 | 47.21 | 0.43 | 0.35 | 0.39 | 46.85 |
| DBNet [24] | 0.32 | 0.28 | 0.30 | 46.10 | 0.39 | 0.35 | 0.37 | 47.50 |
| PSENet [45] | 0.38 | 0.32 | 0.35 | 45.22 | 0.33 | 0.25 | 0.28 | 47.38 |
| DPText-DETR [52] | 0.46 | 0.38 | 0.44 | 46.72 | 0.40 | 0.35 | 0.37 | 47.16 |

As shown in Table 15, AR-Attacker demonstrates strong cross-dataset generalization, effectively degrading the performance of various text detectors on both ICDAR2015 and CTW1500 without any retraining or dataset-specific tuning. For the ICDAR2015 dataset which contains arbitrarily oriented text, our attack drops the F-measure to the 0.3–0.38 range, while maintaining precision below 0.46 for all tested models. On the more challenging CTW1500 dataset, which contains densely distributed curved text instances, our method still achieves substantial degradation (F-measure drops to 0.28–0.39), despite the increased geometric complexity. Across all settings, adversarial examples maintain high visual fidelity, with PSNR values consistently above 45 dB (as shown in Figure 7). These results confirm that the attention-guided query strategy generalizes well to diverse text layouts and shapes, reinforcing the practicality of AR-Attacker in real-world black-box scenarios.

### C.9 Robustness against Stronger Defenses

To further evaluate performance against adversarial training, we fine-tuned DBNet and DPText-DETR using PGD-based adversarial training on the Total-Text training set, following the standard adversarial training protocol [30]. We then evaluated AR-Attacker's attack performance against these adversarially fine-tuned models on the Total-Text test set, using the same attack configuration as in our main experiments.

Table 16: Attack performance of AR-Attacker against adversarially trained models (Dataset: Total-Text).

| Model | Text Removal Rate (TR ↑) | Queries (Q ↓) |
|---|---|---|
| DBNet | 0.76 | 811 |
| AT-DBNet | 0.69 | 1238 |
| DPText-DETR | 0.70 | 899 |
| AT-DPText-DETR | 0.64 | 1372 |

As summarized in Table 16, while AT increases the difficulty of the attack, AR-Attacker maintains high potency. We attribute this resilience to the online, adaptive optimization mechanism of PPO. Since it does not rely on pre-calculated static perturbations, our PPO-based agent dynamically interacts with the victim to iteratively learn and adapt to the specific, modified decision boundaries of the AT-model. The increase in average query counts when attacking AT-models is a direct reflection of this active probing process, where the agent spends more iterations until it successfully suppresses the detection output of the fine-tuned model.

Our existing evaluation (Table 7 ) covers simple frequency-domain purification methods (e.g., JPEG compression, Gaussian blur), where AR-Attacker's effectiveness remains robust (TR drops from 0.76 to 0.52–0.64). However, robustness against stronger, learning-based purification defenses (such as diffusion-based purification) remains a limitation of the current work. We acknowledge that such generative purification methods, which reconstruct pixel values using learned image priors rather than simple filtering, could plausibly remove a larger portion of our perturbation than the defenses currently tested. Nonetheless, our perturbations specifically target text regions, which are themselves inherently high-frequency, any purification strong enough to fully remove them risks also degrading the legibility of the underlying text, undermining the human-readability property central to our privacy-preservation goal. We leave a rigorous empirical evaluation of this trade-off to future work.

## D Details of the Subjective User Study (MOS)

To evaluate the human perceptibility of the adversarial perturbations generated by AR-Attacker, we conducted a subjective user study to obtain Mean Opinion Scores (MOS).

### D.1 Experimental Procedure

We recruited 20 participants for this study. Prior to the experiment, all participants were informed that the study was for academic research purposes and that their data would be processed anonymously. Participation was entirely voluntary and did not receive monetary compensation and informed consent was obtained from each subject. Participants were presented with 100 adversarial images per dataset.

The images were displayed individually in a randomized order. For each image, participants were asked to evaluate its visual quality based on two primary criteria: (1) Does the image look like a real-world photograph without unnatural digital artifacts? (2) Is the scene text in the image clearly readable without effort?

### D.2 Instructions

The participants were given the following instructions: In this study, you will see a series of images containing scene text. Your task is to rate the visual quality of each image on a scale of 1 to 5, focusing on whether the image looks natural and whether the text is easy to read. Please use the following scale for your rating:

- 5 (Excellent): The image looks completely natural and text is perfectly legible.
- 4 (Good): The image is high quality. Very minor artifacts may exist but are not distracting. Text is easily readable.
- 3 (Fair): Some noise or blur is noticeable, but the image remains acceptable and the text is still legible.
- 2 (Poor): Noticeable distortions or artifacts make the image look unnatural. Text may be slightly difficult to read.
- 1 (Very Poor): Heavy distortions or extreme artifacts are present. The image quality is severely degraded.

## E Discussion and limitations

In this paper, we advocate for a paradigm shift in scene text privacy protection: moving from conventional Scene Text Recognition (STR) attacks to Scene Text Detection (STD) suppression. By attacking STD, AR-Attacker effectively prevents automated systems from even localizing the text bounding boxes, fundamentally cutting off downstream OCR pipelines. A core insight of our work is the formulation of the black-box attack as a Markov Decision Process (MDP) coupled with an attention-guided prior. Natural scene images typically have high resolutions with sparse text regions. Traditional query-based attacks (e.g., based on Differential Evolution or global random search) suffer from the curse of dimensionality, wasting a massive number of queries on irrelevant background pixels. By introducing a surrogate-guided attention mask to constrain the action space, combined with a discrete-continuous hybrid policy optimized via PPO, AR-Attacker efficiently navigates the complex search space. It not only accelerates convergence but also ensures that perturbations remain imperceptible by confining them to visually complex semantic text regions.The substantial performance margin over previous heuristic-based query attacks highlights the superiority of RL-based sequential decision-making for sparse adversarial tasks.

While AR-Attacker shows basic resilience against standard pre-processing (Section 4.6), its performance still degrades, and its robustness against advanced defenses (e.g., adversarial training, input purification) remains unexplored. Furthermore, its effectiveness against more sophisticated adversarial defense strategies (such as adversarial training, input purification, or advanced structural filtering) remains to be fully explored. Enhancing robustness against SOTA defenses while preserving visual fidelity remains a critical direction for future work. Our current framework optimizes the RL attack policy strictly per-image. While AR-Attacker successfully reduces the average query count by approximately 40% compared to SOTA query-based baselines, the agent must still interact with the victim model 700 to 900 times per image. Future work will explore meta-learning and offline RL to extract generalizable adversarial priors across large datasets. This shift toward a universal meta-policy aims to achieve single-forward-pass attacks or few-query fine-tuning on unseen images.

## F Future work

To further push the boundaries of efficiency, our future work will transition from per-image optimization to generalized policy learning. By leveraging meta-learning and offline RL, we aim to train a policy on large-scale scene text datasets to extract universal adversarial priors across different

detectors. This allows the agent to be warm-started with a high-quality initial policy for any unseen image, reducing the queries required for real-time privacy protection.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction in Section 1 clearly state the main claims. These claims are systematically validated in Section 4 through quantitative comparisons on multiple datasets and architectures, as well as qualitative visual quality assessments, ensuring that the stated scope matches the empirical evidence provided.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: In Appendix E, we also describe the limitations of the proposed method.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper does not include theoretical results. We present only experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We clearly describe the key steps in our method in Section 3. . The experiment settings are also detailed in Section 4.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All of the data used in our experiments is publicly available and easily obtained. Full experimental settings are provided for all aspects of the experiments in Section 4.1 and Appendix A.1 so that results can be reproduced. Code is not currently provided but can be made available on acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: In Section 4.1, we have specified the key hyperparameters used in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars, as our method has a very significant improvement in Table 1 and 2. Also, the number of experiments is too large, and the cost is too high.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: We report the time of execution and the environment we used in Appendix C.3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, we conducted in the paper conforms, in every respect. Specifically, we briefly discussed the potential negative impacts on society in Section 5.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the societal impacts in Section 5.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We are not releasing any models or data.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The used data, models, and code are open source and properly credited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We conducted a subjective user study with 20 participants to evaluate the visual quality (MOS) of our adversarial examples. The full text of instructions, recruitment details and participant compensation are provided in Appendix D

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The user study involved a simple visual assessment task with minimal risk to participants. All subjects were informed of the study's nature and provided their consent prior to participation. The study followed the ethical standards of the authors' institution for human-centered research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We do not use LLMs as any important, original, or non-standard components in this research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.

- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.