
Towards Irreversible Attack: Fooling Scene Text Recognition via Multi-Population Coevolution Search

Jingyu Li¹, Pengwen Dai^{1*}, Mingqing Zhu¹, Chengwei Wang¹, Haolong Liu¹, Xiaochun Cao¹

¹School of Cyber Science and Technology,
Shenzhen Campus of Sun Yat-sen University, Shenzhen, Guangdong 518107, China
{lijy768, zhmq8, liuhlong7}@mail2.sysu.edu.cn,
wangchw23@alumni.sysu.edu.cn,
{daipw, caoxiaochun}@mail.sysu.edu.cn

Abstract

Recent work has shown that scene text recognition (STR) models are vulnerable to adversarial examples. Different from non-sequential vision tasks, the output sequence of STR models contains rich information. However, existing adversarial attacks against STR models can only lead to a few incorrect characters in the predicted text. These attack results still carry partial information about the original prediction and could be easily corrected by an external dictionary or a language model. Therefore, we propose the Multi-Population Coevolution Search (MPCS) method to attack each character in the image. We first decompose the global optimization objective into sub-objectives to solve the attack pixel concentration problem existing in previous attack methods. While this distributed optimization paradigm brings a new joint perturbation shift problem, we propose a novel coevolution energy function to solve it. Experiments on recent STR models show the superiority of our method. The code is available at <https://github.com/Lee-Jingyu/MPCS>.

1 Introduction

Scene text recognition (STR) [29, 20, 30] has garnered increasing attention in the field of computer vision due to its wide applications, such as augmented reality [27], visual question answering [31], automatic driving [8], etc. Despite its great success, recent studies [32, 36, 37] have shown that STR models are vulnerable to adversarial examples, which deceive the models into making incorrect predictions by adding imperceptible perturbations on the input images. In practice, adversarial examples can be used to prevent private text [4, 37] in images from being recognized and exploited by malicious optical character recognition (OCR) systems. Therefore, adversarial attacks against STR models have become a valuable research topic.

Existing adversarial attacks in the computer vision field mainly focus on non-sequential tasks [9, 23, 33], such as image classification, object detection, face recognition, etc. The output of these models is usually a single label and contains limited information. In this scenario, once the model outputs a different label, the information conveyed by the output is completely altered. In contrast, the prediction of STR models can be seen as a series of character labels, which contain rich information. When attacking STR models, the altered prediction may retain partial information about the original prediction. With the retained information, the opponent could correct the mispredicted text through an extra dictionary [22, 30, 13] or large language models (LLMs) [34, 21, 15], as shown in Figure 1.

However, existing adversarial attacks against STR models consider merely altering the model prediction as a successful attack. For example, some works focus on attacking the text image with a clean

*Corresponding author: Pengwen Dai <daipw@mail.sysu.edu.cn>

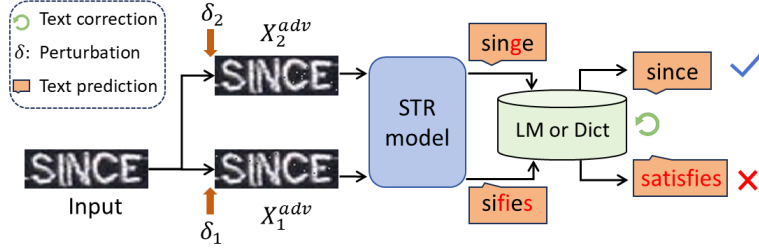


Figure 1: Adversarial attack results with a single incorrect character could be corrected through a language model or a dictionary. The attack results become more irreversible as the number of perturbed characters increases.

background [4, 11]. These methods may change the background or fail to attack due to complex backgrounds when dealing with scene text images. Differently, RoLMA [38] attacks license plate images, whose backgrounds are more complicated, by adding some light spots on the background. These light spots will also destroy the visual continuity of the background when dealing with scene text images. Compared to license plate images, scene text images are harder to attack due to their more complex backgrounds and diverse text styles. Xu et al. [36] develop an efficient optimization-based white-box attack method to fool STR models. Furthermore, in the black-box setting, AD²E [37] narrows the search space in the differential evolution algorithm to attack STR models. These methods overlook the fact that the attack results still contain rich information about the correct text.

In this work, we propose a black-box pixel-level attack method aiming at attacking all characters in the scene text image. We first discuss the reason for previous pixel-level attacks failing to attack more characters, *i.e.*, the attack pixel spatial concentration problem in their global optimization. Then we decompose the global optimization objective into several sub-objectives and introduce spatial constraints to avoid the attack pixel concentration. However, optimizing sub-objectives may undermine the global feasibility, manifested as the joint perturbation shift problem. Therefore, we propose the Multi-Population Coevolution Search (MPCS) method, which utilizes multiple evolutionary adversarial pixel populations in different image subspaces to individually search for each attack pixel in parallel. To solve the joint perturbation shift problem, MPCS introduces a coevolution energy function that implicitly embeds the global constraint into the local objective function. Experimental results on recent STR models demonstrate that MPCS produces substantially more incorrect predicted characters than state-of-the-art attack methods.

2 Related works

2.1 Attacks on common vision models

Existing adversarial attack methods mainly focus on non-sequential vision tasks, such as image classification, object detection, and facial recognition. These adversarial attack methods can be roughly categorized into white-box attacks and black-box attacks. White-box attacks use the whole information of the victim model, including model structure, parameters, and gradients. Fast gradient sign method (FGSM) [9] is proposed to generate adversarial perturbation according to the sign of the gradient. Deepfool [23] proposes to find the nearest classification hyperplane of the victim model to generate adversarial examples. GAT [12] composes multiple attack methods and utilizes a doubly stochastic matrix to optimize the attack order.

Differently, black-box attacks can only obtain the output of the victim model. OnePixel [33] employs the Differential Evolution (DE) algorithm to find a few adversarial pixels to attack image classification models. Transfer-based attacks [25, 1] utilize adversarial examples generated from attacking a source model to fool the target victim model. PRGF [5] introduces two prior-guided random gradient-free algorithms aimed at enhancing the efficiency of black-box adversarial attacks, which leverage a transfer-based prior provided by the gradient of a surrogate model. Furthermore, to enhance the transferability of adversarial perturbations, LLTA [7] generates generalized adversarial perturbations through data and model augmentation, and optimizes the final perturbation update through meta-learning.

These methods aim at attacking models that output non-sequential results, such as categories, positions, and scores. However, text recognition models predict text sequences, which usually contain more information for attackers to manipulate. We need to alter more predicted characters to ensure irreversible attack results.

2.2 Attacks on text recognition models

Text recognition models output text sequences containing more information than non-sequential outputs, which makes adversarial attacks aimed at them more challenging. [32] introduces an adversarial attack method targeting OCR systems, which involves making minor modifications to printed text images to deceive the OCR system into recognizing words with meanings opposite to the original text. [35] explores the effectiveness of the transfer-based attack algorithm in attacking real-world OCR systems. Another approach [4] involves manipulating the background of text images by replacing clean image backgrounds with random textures to disrupt the recognition by OCR systems. RoLMA [38] proposes a targeted attack method to fool license plate recognition models by adding some crafted light spots to images. Furthermore, Xu et al. [36] propose an optimization-based method to update the perturbation iteratively, attacking both CTC-based (connectionist temporal classification) and attention-based scene text recognition models. Based on the DE algorithm, AD²E [37] proposes to narrow down the continuous searching space to a discrete space, which can speed up the searching process. It generates adversarial examples by fixing a few pixels' values to 0 or 255 in the image.

These methods focus on changing the model output, which means attackers only need to change one character in the model predictions. However, a text with a few incorrect characters is easy to restore via the correction of an external dictionary or LLMs [34, 21, 15]. Therefore, in this work, we aim to attack all characters in images to make text recognition models predict more incorrect characters.

3 Preliminary

3.1 Problem formulation

Given a scene text image $X \in \mathbb{R}^{h \times w \times 3}$ (h and w are the height and width of the image), and its ground truth label $Y = [y_1, y_2, \dots, y_n]$ (n means the character number), the existing paradigm of crafting an adversarial example $X^{adv} = X + \delta$ is to add a small perturbation δ to X to change the output of the victim model $\mathcal{F}$, such that,

$$\mathcal{F}(X) \neq \mathcal{F}(X + \delta), \quad \text{s.t.} \quad \|\delta\|_p \leq \epsilon, \quad (1)$$

where ϵ represents the maximum perturbation.

However, when attacking scene text recognition models, merely altering the output is not enough. We need to mislead the victim model to predict as many erroneous characters as possible, rendering the attack results more irreversible. So we formulate a new objective as follows,

$$\delta^* = \arg \max_{\delta} D(\mathcal{F}(X), \mathcal{F}(X + \delta)), \quad \text{s.t.} \quad \|\delta\|_p \leq \epsilon, \quad (2)$$

where δ^* is the best adversarial perturbation. $D(\cdot, \cdot)$ is the edit distance [19] between the two predicted texts, which is measured by counting the minimum number of character operations required to transform one string into the other.

In the black-box attack settings, the structure, parameters, and gradients of the victim model are not accessible. The attacker can query the victim model to get the model output, which could be either a probability distribution or a character sequence with confidence scores. In practice, most OCR systems provide only the latter, which contains less information for attackers. Considering practicality, we choose the latter attack scenario.

3.2 Objective decomposition

In pixel-level attacks, the final perturbation δ consists of a series of pixel perturbations δ_i , *i.e.*, $\delta = \sum \delta_i$. And each pixel perturbation δ_i can be represented by a single attack pixel $p_i = (x_i, y_i, v_i)$, where x_i and y_i are the coordinates, and v_i is the pixel value. Former methods [33, 37] optimize δ in

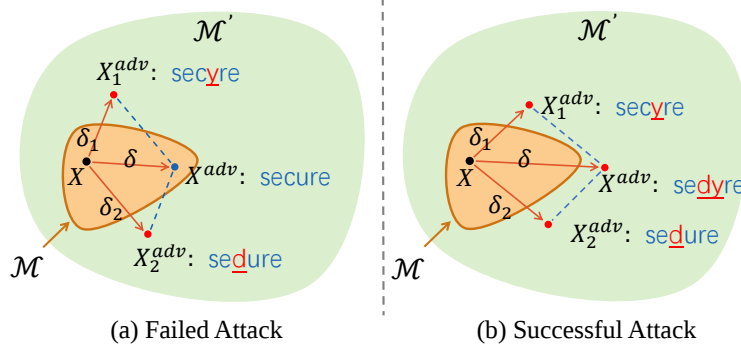


Figure 2: Illustration of the joint perturbation shift. (a) and (b) demonstrate two cases when combining two successful perturbations into one to attack text recognition models. Characters with underlines are incorrectly predicted.

the whole image space. This optimization paradigm is prone to fall into the local optimal solution where attack pixels are concentrated in the local space of the image. In this case, some characters will have no nearby attack pixel and then be correctly recognized.

To solve this problem, we decompose the optimization objective δ^* into multiple sub-objectives δ_i^* , turning it to be a distributed optimization problem. Then we limit the perturbation location to a specific region in each sub-optimization problem. Specifically, we divide the image into multiple non-overlapping subspaces $\{\mathcal{R}_i\}_N$ and find a local optimal attack pixel p_i in each subspace $\mathcal{R}_i$. In this way, a proper division of $\{\mathcal{R}_i\}_N$ could ensure that the attack pixels are not spatially concentrated.

3.3 The joint perturbation shift problem

In this distributed optimization problem, local optimization may undermine global feasibility. Even if each local optimized perturbation δ_i^* can successfully attack the target model $\mathcal{F}$, the global perturbation $\delta^* = \sum \delta_i^*$ has a certain probability of failing to attack, *i.e.*,

$$P\left(\mathcal{F}(X) = \mathcal{F}(X + \delta^*) \mid \bigwedge_{i=1}^N \mathcal{F}(X) \neq \mathcal{F}(X + \delta_i^*)\right) > 0. \quad (3)$$

We term this phenomenon the *joint perturbation shift* problem, and give a simple illustration of it in Figure 2. Let $\mathcal{M}$ and $\mathcal{M}'$ be the manifold area [1] where data samples can be correctly and incorrectly recognized by the target model, respectively. X is the original data sample with label “secure”. Although $X_1^{adv} = X + \delta_1$ and $X_2^{adv} = X + \delta_2$ are two successful adversarial samples located in $\mathcal{M}$, the joint perturbation $\delta = \delta_1 + \delta_2$ may shift the adversarial sample $X^{adv} = X + \delta$ to $\mathcal{M}'$ and lead to a failed attack.

4 Multi-population coevolution search

Following the distributed optimization paradigm, we propose the Multi-Population Coevolution Search (MPCS) method, which divides the image horizontally into N equal regions $\{\mathcal{R}_i\}_N$ and utilizes evolutionary populations to search in each region. Meanwhile, a coevolution energy function is introduced to address the joint perturbation shift problem. Next, we will elaborate on the MPCS algorithm.

Initialization. We first randomly initialize an adversarial pixel population $\mathbf{P}_i$ in each region $\mathcal{R}_i$. Each population $\mathbf{P}_i$ contains M adversarial pixels $\{p_i^j\}^M$, whose coordinates are randomly initialized and values are fixed to θ , which can be formulated as follows,

$$\mathbf{P}_i = \{p_i^j = (x_i^j, y_i^j, v_i^j) \mid (x_i^j, y_i^j) \in \mathcal{R}_i, v_i^j = \theta\}^M, \quad (4)$$

where x_i^j and y_i^j are the pixel coordinates, v_i^j is the pixel rgb value. Moreover, based on the finding from [37] that the adversarial pixel values of $\theta_{max} = (255, 255, 255)$ and $\theta_{min} = (0, 0, 0)$ have a

higher possibility to successfully attack than other values, we set θ as follows,

$$\theta = \begin{cases} \theta_{max}, & G(v_i^j) \leq \mu, \\ \theta_{min}, & G(v_i^j) > \mu, \end{cases} \quad (5)$$

where $G(v_i^j)$ denotes the gray value of v_i^j , and μ is set to 127 (the median of the set of pixel values). After that, we calculate the prior energy $\mathbf{E}^p$ of each adversarial pixel in populations as,

$$\mathbf{E}^p(p_i^j) = \text{Dist}(Y, \mathcal{F}(X + \Delta(p_i^j))), \quad (6)$$

where $\Delta(\cdot)$ is the perturbation generated by replacing the original pixel with the adversarial pixel, and $\text{Dist}(Y, \cdot)$ denotes the distance between the predicted label and the ground truth label Y (detailed in Appendix D). Here, $\mathbf{E}^p$ is equal to the local objective function.

Coevolution. We let the N populations coevolve in the direction of higher energy following the differential evolution (DE) algorithm [3, 37]. Briefly, each adversarial pixel p in a population has a probability of mutating into a new adversarial pixel p' . Among p and p' , the one with higher energy will be retained for the next generation.

However, as the existence of the joint perturbation shift problem, the prior energy function $\mathbf{E}^p$ cannot filter the best adversarial pixel. Therefore, we propose a novel coevolution energy function $\mathbf{E}^c$, which implicitly embeds the global constraint into the local objective function,

$$\mathbf{E}^c(p_i^j) = \text{Dist}(Y, \mathcal{F}(X + \Delta(\{p_i^j\} \cup \{p_k^*\}_{N-1}))), \quad \text{s.t. } 1 \leq k \leq N, k \neq i, \quad (7)$$

where $\{p_k^*\}_{N-1}$ is the set of adversarial pixels with the highest energy in the other $N-1$ populations. Note that in the first generation, we select $\{p_k^*\}_{N-1}$ according to their prior energy $\mathbf{E}^p$. After the first generation, we replace $\mathbf{E}^p$ with $\mathbf{E}^c$ to calculate the energy of adversarial pixels. In this case, the evolution of one population will influence the other populations, which is similar to the coevolution phenomenon in nature.

Early stop. We propose an early stop mechanism, which terminates the evolution of the populations whose maximum energy has not increased compared to the previous generation. When the maximum number of generations g_{max} is reached or when all populations have ceased evolution, we obtain the best adversarial pixels $\{p_i^*\}_N$ from each population.

Post process. If $\{p_i^*\}_N$ can successfully attack the victim model, we take them as the final attack pixels $\{p_i^{adv}\}_N$; otherwise, we will employ a pixel set population $\mathbf{P}_v$ to search for better pixel values following DE algorithm,

$$\mathbf{P}_v = \{\mathbf{Q}^t\}^L, \quad \mathbf{Q}^t = \{p_i^t | (x_i^t, y_i^t) = (x_i^*, y_i^*)\}_N, \quad (8)$$

where L is the population size of $\mathbf{P}_v$, $\mathbf{Q}^t$ is the t -th adversarial pixel set whose pixel coordinates are frozen to be the same as $\{p_i^*\}_N$ and values are randomly initialized. The energy function of $\mathbf{P}_v$ is calculated as,

$$\mathbf{E}^v(\mathbf{Q}^t) = \text{Dist}(\mathcal{F}(X), \mathcal{F}(X + \Delta(\mathbf{Q}^t))). \quad (9)$$

At the max generation g'_{max} of $\mathbf{P}_v$, we obtain the best adversarial pixel set $\mathbf{Q}^{best}$ as the final attack pixels $\{p_i^{adv}\}_N$. With the final attack pixels, the adversarial example is generated as follows,

$$X^{adv} = X + \Delta(\{p_i^{adv}\}_N). \quad (10)$$

5 Analysis

We make a theoretical analysis of the superiority of our proposed coevolutionary energy function $\mathbf{E}^c$ over the prior energy function $\mathbf{E}^p$. For a certain pixel p_i^j in population $\mathbf{P}_i$, we define:

$$A := \mathbb{I}\{\mathcal{F}(X) \neq \mathcal{F}(X + \Delta(p_i^j))\}, \quad \mathbf{E}^p(p_i^j) \propto P(A = 1), \quad (11)$$

where $\mathbb{I}$ is the indicator function whose value is 0 or 1, and the estimator $\mathbf{E}^p(p_i^j)$ approximately reflects the relative strength of the probability of $A = 1$. Similarly, we define:

$$B := \mathbb{I}\{\mathcal{F}(X) \neq \mathcal{F}(X + \Delta(\{p_i^j\} \cup \{p_k^*\}_{N-1}))\}, \quad \mathbf{E}^c(p_i^j) \propto P(B = 1), \quad (12)$$

where $k \neq i$ and $\{p_k^*\}_{N-1}$ denotes the local optimal pixels in the other $N - 1$ populations.

Our final goal is to increase the expectation of Z , which is defined as,

$$Z := \mathbb{I}\{\mathcal{F}(X) \neq \mathcal{F}(X + \Delta(\{p_k^*\}_N))\}, \quad (13)$$

where $\{p_k^*\}_N$ denotes the local optimal pixels in all N populations.

Next, according to the joint perturbation shift problem illustrated in Sec 3.3, we have:

$$P(Z = 1|A = 1) < 1, \quad P(Z = 0|A = 1) > 0, \quad (14)$$

which means the conditional distribution $P(Z|A = 1)$ is not a deterministic distribution. Therefore, the conditional entropy of Z given $A = 1$ is strictly positive, *i.e.*,

$$H(Z|A = 1) = - \sum_{z \in \{0,1\}} P(Z = z|A = 1) \log P(Z = z|A = 1) > 0. \quad (15)$$

On the other side, $\mathbf{E}^c(p_i^j) \leq \mathbf{E}^c(p_i^*)$ holds for all pixels in population $\mathbf{P}_i$. It means that if a successful attack can be achieved by the joint perturbation $\Delta(\{p_i^j\} \cup \{p_k^*\}_{N-1})$, then the joint perturbation of all the local optimal pixels $\Delta(\{p_k^*\}_N)$ must also yield a successful attack. Therefore:

$$P(Z = 1|B = 1) = 1, \quad P(Z = 0|B = 1) = 0. \quad (16)$$

Similarly, we have $H(Z|B = 1) = 0 < H(Z|A = 1)$, which reveals that when $B = 1$ (joint perturbation succeeds), the prediction of the global attack result Z is completely deterministic, unlike when $A = 1$ (local perturbation succeeds). It means that the prior energy function $\mathbf{E}^p$ has a certain degree of information loss when estimating the attack performance of pixel p_i^j , while the coevolution energy function $\mathbf{E}^c$ fills the loss.

6 Experiments

6.1 Experiment settings

Datasets and models. We validate the effectiveness of our proposed attack method on four frequently-used datasets, including ICDAR13 [17] (horizontal regular scene text, 1015 images), SVTP [26] (perspective distorted text, 645 images), CUTE80 [28] (curved scene text, 288 images), and IC-DAR15 [16] (irregular scene text, 2077 images). We fool three mainstream paradigms of scene text recognition, including the CTC-based paradigm (CRNN [29]), the attention-based paradigm (ASTER [30] and SAR [20]), and the multi-modal paradigm (IGTR [6]). All the models are from MMOCR [18] or OpenOCR [24]. If not specified, the experiments are conducted on the CRNN model and the CUTE80 dataset for convenience.

Metrics. We employ four metrics including the attack success rate before/after text correction (**SR/SR***), the perturbation rate (**PR**), and the (**L2**) distance between the adversarial image and the original image. Specifically, we employ GPT-4o mini [14] to correct the mispredicted text results and then calculate the **SR*** metric. The **PR** metric measures the proportion of the mispredicted characters,

$$\mathbf{PR} = \sum_{i=1}^S \frac{D(Y_i, \mathcal{F}(X_i^{adv}))}{\text{len}(Y_i)}, \quad (17)$$

where S represents the total number of successful adversarial examples, $\text{len}(\cdot)$ denotes the length of the text, and $D(\cdot, \cdot)$ is the edit distance between the two predicted texts. The **L2** metric measures the image perturbation degree.

Text correction. Many STR methods use an extra dictionary to correct their prediction [22, 30, 13]. Language models can do the same job, and they are more flexible. So we choose to employ GPT-4o mini [14] for text correction and then calculate the **SR*** metric. Further details are in Appendix C.

6.2 Comparison with pixel-level attacks

We replicate two state-of-the-art black-box pixel-level attack methods, AD²E [37] and OnePixel [33], which are also DE-based methods, and compare their attack performance with ours, as shown

Table 1: Comparison with other pixel-level attack methods. **SR*** is the attack success rate after correction by GPT-4o mini [14]. **PR** denotes the character perturbation rate.

Model	Method	CUTE80				IC13				SVTP			
		SR. $\uparrow$	SR*. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$	SR. $\uparrow$	SR*. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$	SR. $\uparrow$	SR*. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$
CRNN	OnePixel	95.60	60.09	52.26	4.25	95.31	43.78	49.82	3.98	96.76	58.06	42.99	3.93
	AD ² E	95.60	49.73	29.75	4.20	96.31	36.24	29.77	3.97	96.99	42.61	23.50	3.91
	MPCS	87.36	86.75	94.54	4.30	85.47	70.12	90.19	4.02	85.65	77.67	80.07	4.04
ASTER	OnePixel	57.98	20.59	39.18	4.23	41.86	14.05	39.63	3.95	68.17	28.93	33.41	3.89
	AD ² E	57.14	16.93	28.67	4.22	47.66	13.48	32.94	3.96	67.45	22.35	22.06	3.92
	MPCS	51.26	27.77	74.93	4.32	43.31	21.60	83.05	3.98	57.17	42.14	51.79	3.95
SAR	OnePixel	65.98	19.26	43.40	4.20	42.52	16.16	41.40	3.91	58.45	18.89	28.67	3.87
	AD ² E	70.08	25.00	30.68	4.18	42.52	14.10	28.51	3.88	57.67	17.30	21.56	3.87
	MPCS	58.61	33.76	66.02	4.19	34.22	15.77	55.31	3.82	48.35	25.96	45.01	3.83
IGTR	OnePixel	41.38	24.29	61.55	4.35	24.27	17.19	55.66	4.02	41.41	24.04	54.22	4.03
	AD ² E	42.24	25.00	36.04	4.31	23.95	14.95	33.14	3.97	42.29	18.94	30.28	4.02
	MPCS	61.21	49.14	89.86	4.44	39.67	32.05	82.86	4.16	54.87	37.95	79.29	4.22

in Table 1. In specific, we set the number of attack pixels to 10 (the same as ours) when reproducing these methods. As shown in Table 1, our method achieves the highest perturbation rate (PR) on four mainstream STR models and three widely used datasets. For example, when attacking the SOTA STR model IGTR [6], our method achieves 89.86%, 82.86%, and 79.29% PRs on the CUTE80, IC13, and SVTP datasets respectively, surpassing the second-best method by 28.31%, 27.21%, and 25.07%.

Furthermore, when attacking the CRNN, ASTER, and SAR models on the CUTE80 dataset, our method achieves 26.66%, 7.18%, and 14.5% higher SR* metrics than the second-best method, although our SR metrics is slightly lower due to the joint perturbation shift. It demonstrates that our attack method is much more irreversible and that the PR metric can effectively evaluate the attack’s irreversibility.

Table 2: Comparison with other attack paradigms. Sparse-rs [2] is a black-box attack method targeting image classification models. UDUP [4] is a transfer-based attack method targeting OCR systems. Xu et al. [36] propose a white-box attack method targeting STR models.

Model	Method	CUTE80			IC13			SVTP		
		SR. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$	SR. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$	SR. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$
CRNN	Sparse-rs	63.73	40.23	11.22	56.77	38.10	11.44	76.11	52.70	9.80
	UDUP	2.76	82.86	0.63	0.23	13.33	0.14	0	0	0.19
	Xu et al.	33.51	30.35	2.88	30.95	41.20	3.06	49.65	21.47	2.85
	MPCS	87.36	94.54	4.30	85.47	90.19	4.02	85.65	80.07	4.04
ASTER	Sparse-rs	12.55	7.65	11.41	9.50	6.84	11.46	22.18	11.27	9.88
	UDUP	0.97	30.00	1.20	0.12	16.67	0.15	0	0	0.20
	Xu et al.	36.26	35.38	2.85	32.84	38.70	3.04	56.29	23.04	2.82
	MPCS	51.26	74.93	4.32	43.31	83.05	3.98	57.17	51.79	3.95
SAR	Sparse-rs	19.23	11.65	11.43	12.47	9.54	11.45	24.77	12.03	9.76
	UDUP	3.70	107.45	1.78	0	0	0.20	0.18	16.67	0.17
	MPCS	58.61	66.02	4.19	34.22	55.31	3.82	48.35	45.01	3.83

6.3 Comparison with other attack paradigms

To demonstrate the superiority of our proposed method in attacking STR models, we compare our method with several other attack paradigms, as shown in Table 2. Xu et al. [36] propose a white-box attack method that updates the perturbation iteratively by optimization. When reproducing its code, we set the upper limit of the L2 metric to 10 to prevent image distortion caused by excessive perturbation, while our L2 metrics are less than 5. On the CRNN model and CUTE80 dataset, the SR and PR of this method are 53.85% and 64.19% lower than ours. This is because this method can hardly find successful perturbations under low perturbation limitations.

Table 3: Ablation study of our method. **MP** means the multi-population mechanism, **CEF** denotes the coevolution energy function, **MG** represents that the population will evolve to the maximum generation (*i.e.*, it will not stop when the attack is successful), and **ES** is the early stop mechanism.

Method	MP	CEF	MG	ES	SR. $\uparrow$	SR*. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$
Baseline					95.6	55.58	29.75	4.20
MPCS	✓				76.37	52.95	52.37	4.32
	✓	✓			88.46	75.42	71.43	4.30
	✓	✓	✓		90.66	87.20	99.72	4.29
	✓	✓	✓	✓	87.36	86.75	94.54	4.30

Besides, we compare our method with UDUP [4], a transfer-based attack method that fools OCR models by generating adaptive underpainting. As shown in Table 2, this method has extremely low SR when attacking STR models, up to 84.6% lower than ours. This is because UDUP cannot successfully add underpainting to scene text images with complicated backgrounds. Furthermore, we adopt Sparse-rs [2], a black-box attack method designed for image classification models, for attacking text recognition models by modifying its victim models and loss function. Our method outperforms Sparse-rs with +23.63% SR, +54.31% PR, and -6.92 L2 on CUTE80 and CRNN. These results demonstrate that attack methods tailored for image classification models are not suitable for attacking text recognition models.

6.4 Ablation study

In this section, we conduct comprehensive ablation experiments on the four main components of our proposed method. We reproduce AD²E [37] under our experiment settings and use it as our baseline. As shown in Table 3, after replacing the single population in the baseline with our proposed multi-population (**MP**), the PR increases by 22.62% while the SR decreases by 19.23% (due to the joint perturbation shift problem). As shown in Figure 3, with the spatial constraints imposed by **MP**, adversarial pixels within each population gradually converge to small regions that are evenly distributed across the entire image during the evolution process. Next, we employ the coevolution energy function (**CEF**) to get SR and PR increased by 12.09% and 19.06%, demonstrating its effectiveness.

Furthermore, in order to achieve a better performance, we force all the populations to evolve until the maximum generation (**MG**) rather than stopping evolution when the attack is successful. The fourth row shows that this solution gains an improvement of 2.2% and 28.29% on SR and PR, respectively. Meanwhile, we propose the early stop (**ES**) mechanism to balance the extra time consumption caused by **MG**. According to the last row in Table 3, it has a small impact on the attack performance. And the average total number of evolutionary generations with ES is less than that without ES, as shown in Figure 4.



Figure 3: Population evolution visualization.

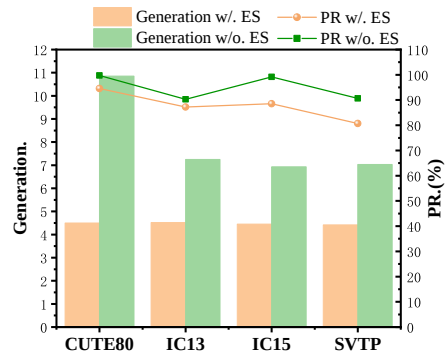


Figure 4: Effect of the early stop (ES).

Table 4: Influence of the population number N and the population size M . We set up 6 experiments on two datasets while keeping the sum of all population sizes constant.

N, M	CUTE80			IC13		
	SR. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$	SR. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$
(4, 150)	75.27	61.84	2.75	69.60	56.19	2.59
(5, 120)	77.47	70.23	3.09	73.19	60.75	2.90
(6, 100)	84.61	71.83	3.37	74.41	63.93	3.17
(10, 60)	87.35	94.54	4.30	80.11	87.23	4.02
(20, 30)	91.20	124.52	6.03	88.27	133.77	5.72

6.5 Influence of hyper-parameters

Firstly, we investigate the influence of the population number N and the population size M of $\mathbf{P}_i$, as shown in Table 4. We keep the sum of all population sizes constant, *i.e.*, $N \times M = 600$, and adjust the combination of N and M . As N increases to 20 on CUTE80, the SR, PR, and L2 rise to 91.20%, 124.52%, and 6.03, respectively. We adopt $N = 10$, which also has a promising attack performance with smaller perturbation.

6.6 Correction by the dataset dictionary

We also conduct experiments on a more traditional text correction method, *i.e.*, through an extra dictionary. We use the test set vocabulary of the IC15 dataset as the dictionary to correct the results of successful attacks through it. As shown in Table 5, among the attack results on the three STR models, our method exhibits much higher SR and PR compared to both OnePixel [33] and AD²E [37]. These results demonstrate that our method still has better attack irreversibility under this setting.

6.7 Effect range of adversarial pixels

We conduct experiments to investigate the effect range of adversarial pixels. In specific, we restrict the location of adversarial pixels to ensure that some characters do not have adversarial pixels around them. As shown in Table 6, when the adversarial pixels are restricted to the left half and right half of the image, the PR metric decreases by 20.42% and 13.68%, respectively. These results demonstrate that the adversarial pixel has a higher possibility of affecting nearby characters rather than farther characters.

Table 5: The attack results on the IC15 dataset. **SR*** is the attack success rate after correction by the dataset dictionary.

Method	CRNN		ASTER		SAR	
	SR*. $\uparrow$	PR. $\uparrow$	SR*. $\uparrow$	PR. $\uparrow$	SR*. $\uparrow$	PR. $\uparrow$
OnePixel	42.15	77.93	19.19	62.18	21.10	65.20
AD ² E	29.63	50.31	18.32	51.17	17.23	50.24
MPCS	71.19	94.48	35.53	78.27	27.84	73.87

Table 6: The attack performance on the CUTE80 dataset when the perturbation area is limited to the left and right half of the image, respectively.

Perturbation Area	SR. $\uparrow$	PR. $\uparrow$	L2. $\downarrow$
Left	32.97	74.12	3.38
Right	39.01	80.68	3.65
All	87.36	94.54	4.30

7 Discussion and limitations

In this work, we propose a pixel-level black-box attack method targeting at altering each character in STR models’ prediction, termed as MPCs. By modifying only a few pixels in the image, MPCs can mislead STR models to predict more incorrect characters than previous methods, while maintaining the whole visual semantic information. We believe MPCs offers valuable applications while requiring careful consideration of its societal implications. On the positive side, the generated adversarial examples can strengthen STR models through adversarial training, improving their robustness against real-world perturbations. Additionally, MPCs enables effective privacy protection by preventing

unauthorized OCR extraction of sensitive text with minimal visual distortion. For example, when users upload images to social platforms, we could add tiny perturbation pixels to the images so that the private texts may not be correctly recognized and collected by malicious OCR systems. Besides, the potential misuse of such adversarial attacks raises important concerns. Malicious actors could exploit MPCs to bypass security systems, manipulate automated text recognition, or evade content moderation, highlighting the need for responsible research practices and countermeasures.

Meanwhile, although MPCs do not affect the viewing of text images by human eyes, the pixels it disturbs are still quite different from the original pixels. In future work, we will investigate finding a better pixel value optimization method to minimize the deviation between adversarial and original pixel values, thus enhancing the attack imperceptibility.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (62302532, 62411540034), the Natural Science Foundation of Guangdong (2025A1515011224), and Shenzhen Science and Technology Program (202206193000001, 20220816225523001, KQTD20221101093559018).

References

- [1] Yanbo Chen and Weiwei Liu. A theory of transfer-based black-box attacks: Explanation and implications. In *Advances in Neural Information Processing Systems*, volume 36, pages 13887–13907, 2023.
- [2] Francesco Croce, Maksym Andriushchenko, Naman D Singh, Nicolas Flammarion, and Matthias Hein. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6437–6445, 2022.
- [3] Swagatam Das and Ponnuthurai Nagarathnam Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2010.
- [4] 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 ACM International Conference on Multimedia*, pages 7559–7568, 2023.
- [5] Yinpeng Dong, Shuyu Cheng, Tianyu Pang, Hang Su, and Jun Zhu. Query-efficient black-box adversarial attacks guided by a transfer-based prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9536–9548, 2021.
- [6] Yongkun Du, Zhineng Chen, Yuchen Su, Caiyan Jia, and Yu-Gang Jiang. Instruction-guided scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(4):2723–2738, 2025.
- [7] Shuman Fang, Jie Li, Xianming Lin, and Rongrong Ji. Learning to learn transferable attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 571–579, 2022.
- [8] Alvaro Gonzalez, Luis M Bergasa, and J Javier Yebes. Text detection and recognition on traffic panels from street-level imagery using visual appearance. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):228–238, 2013.
- [9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [10] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, page 369–376, 2006.

- [11] 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 ACM International Conference on Multimedia*, pages 7424–7434, 2023.
- [12] Lei Hsiung, Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Towards compositional adversarial robustness: Generalizing adversarial training to composite semantic perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24658–24667, 2023.
- [13] Mingxin Huang, Dezhi Peng, Hongliang Li, Zhenghao Peng, Chongyu Liu, Dahua Lin, Yuliang Liu, Xiang Bai, and Lianwen Jin. Swintextspotter v2: Towards better synergy for scene text spotting. *International Journal of Computer Vision*, pages 1–21, 2025.
- [14] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [15] Niddal H. Imam, Vassilios G. Vassilakis, and Dimitris Kolovos. Ocr post-correction for detecting adversarial text images. *Journal of Information Security and Applications*, 66:103170, 2022.
- [16] Dimosthenis Karatzas, Lluís Gomez-Bigorda, Angelos 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 *International Conference on Document Analysis and Recognition*, pages 1156–1160, 2015.
- [17] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluís Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluís Pere De Las Heras. Icdar 2013 robust reading competition. In *International Conference on Document Analysis and Recognition*, pages 1484–1493, 2013.
- [18] Zhanghui Kuang, Hongbin Sun, Zhizhong Li, Xiaoyu Yue, Tsui Hin Lin, Jianyong Chen, Huaqiang Wei, Yiqin Zhu, Tong Gao, Wenwei Zhang, Kai Chen, Wayne Zhang, and Dahua Lin. Mmocr: A comprehensive toolbox for text detection, recognition and understanding. In *Proceedings of the ACM International Conference on Multimedia*, page 3791–3794, 2021.
- [19] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710, 1966.
- [20] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8610–8617, 2019.
- [21] Jiquan Li, Junliang Guo, Yongxin Zhu, Xin Sheng, Deqiang Jiang, Bo Ren, and Linli Xu. Sequence-to-action: Grammatical error correction with action guided sequence generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10974–10982, 2022.
- [22] Canjie Luo, Lianwen Jin, and Zenghui Sun. Moran: A multi-object rectified attention network for scene text recognition. *Pattern Recognition*, 90:109–118, 2019.
- [23] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.
- [24] <https://github.com/Topdu/OpenOCR>.
- [25] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.
- [26] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 569–576, 2013.

- [27] Rishi R, Abhishek S, Anudeep N, and V Vivek. A survey on advanced text recognition and projection in augmented reality. In *International Conference on Advances in Computing, Communication Control and Networking*, pages 1085–1089, 2022.
- [28] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014.
- [29] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2016.
- [30] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2035–2048, 2019.
- [31] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019.
- [32] Congzheng Song and Vitaly Shmatikov. Fooling ocr systems with adversarial text images. *arXiv preprint arXiv:1802.05385*, 2018.
- [33] 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.
- [34] Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. Grammatical error correction using pseudo learner corpus considering learner’s error tendency. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32, 2020.
- [35] Shudeng Wu, Tao Dai, Guanghao Meng, Bin Chen, Jian Lu, and Shu-Tao Xia. Transferable adversarial attacks for deep scene text detection. In *International Conference on Pattern Recognition*, pages 8945–8951, 2021.
- [36] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12304–12314, 2020.
- [37] 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.
- [38] Mingming Zha, Guozhu Meng, Chaoyang Lin, Zhe Zhou, and Kai Chen. Rolma: a practical adversarial attack against deep learning-based lpr systems. In *Information Security and Cryptology*, pages 101–117, 2020.

NeurIPS Paper Checklist

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 Sec. 1 clearly state the main claims.

Guidelines:

- The answer NA 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 NA 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: Please refer to Sec. 7.

Guidelines:

- The answer NA 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: [\[NA\]](#)

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA 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 Sec. 4. The experiment settings are also detailed in Sec. 6.

Guidelines:

- The answer NA 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: [Yes]

Justification: We provide the code as the supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.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://nips.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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Please refer to Sec. 6.

Guidelines:

- The answer NA 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, as shown in Table 1. Also, the number of experiments is too large, and the cost is too high.

Guidelines:

- The answer NA 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: Please refer to Sec 6.

Guidelines:

- The answer NA 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: We have carefully checked the Ethics Guidelines to make sure our research is with it.

Guidelines:

- The answer NA 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 Sec 7.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

- The answer NA 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: [NA]

Justification: We do not use existing assets.

Guidelines:

- The answer NA 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: [\[Yes\]](#)

Justification: The documentation is provided in the code repository.

Guidelines:

- The answer NA 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: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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: [\[NA\]](#)

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLM is employed to correct the attack results. The details could be found in Sec 6 and Appendix C.

Guidelines:

- The answer NA 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 (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

A Implementation details

In our experiments, we set the population number N and the population size M to 10 and 60, respectively. For population P_v , we set the population size L to 600. The max generation g_{max} and g'_{max} are set to be 10 and 1 for a low query budget. The height h and width w of the input image are 32 and 100, respectively. The experiments are conducted on a workstation with a single GeForce RTX 3090 GPU, a 2.90 GHz Intel(R) Xeon(R) Gold 6226R CPU, and 64G RAM, based on Pytorch v2.0.1.

B Visualization

We visualize our adversarial examples and their corresponding attack results, and compare them with other attacks, as shown in 5. Furthermore, we visualize the generated adversarial examples against the CRNN model under different settings in Table 4, as shown in Figure 6. Typically, the attack performance becomes better with the increasing N . Besides, we notice that only a few pixels are needed to attack a single character, as shown in the second and third rows. Therefore, we can dynamically adjust the attack pixel number according to specific images to weaken the perturbation in practice.

Method	Origin	Xu et al.	OnePixel	AD ² E	Ours
Image					
Label	"carp"	"ca <u>l</u> p"	"ca <u>i</u> rl <u>e</u> "	"ca <u>l</u> p"	" <u>d</u> oe <u>i</u> l <u>i</u> t <u>s</u> "
Image					
Label	"islands"	"islan <u>s</u> "	"is <u>e</u> ane <u>y</u> s"	"is <u>b</u> ands"	" <u>s</u> sean <u>i</u> dn <u>g</u> "
Image					
Label	"estd"	"es <u>t</u> "	"es <u>i</u> ti <u>p</u> "	"es <u>t</u> p"	" <u>f</u> isi <u>t</u> rio"
Image					
Label	"tokyo"	"toky <u>a</u> "	"toky <u>i</u> e"	"toky <u>e</u> "	" <u>i</u> ok <u>i</u> e"
Image					
Label	"20"	"20 <u>0</u> "	"sa <u>o</u> v"	"20 <u>0</u> "	" <u>g</u> eallan"
Image					
Label	"dreams"	"drean <u>s</u> "	"dren <u>i</u> ms"	"drean <u>s</u> "	" <u>b</u> re <u>r</u> ing"

Figure 5: Visualization of the adversarial examples and their attack results. Characters with underlines are incorrectly predicted. Compared to other methods, our approach exhibits extraordinary attack performance. Zoom in to get a better view.

C Text correction by LLM

We employ GPT-4o mini [14] to correct the mispredicted text results and calculate the SR* metric. The prompt is formed as follows:

"Correct the spelling of the following words:

slacrest =>

N	4	5	6	10	20	Origin
Image						
Label	"ca ll b"	"ca ll b"	"aa ll d"	"la bl s"	"fa bt ies"	"club"
Image						
Label	"a44 4 "	"ha ll y"	"ni dely "	"ma af "	"a lail "	"14"
Image						
Label	"ce ss "	"co d "	"co d "	"ce ss "	"ge swi "	"6"
Image						
Label	"ru bb ack"	"plis lne k"	"ru bb ick"	"fu ij strer"	"fois haotes "	"public"

Figure 6: Visualization of our adversarial examples against the CRNN model under different attack settings. N is the population number, which is equal to the attack pixel number. The sum of population sizes is 600 in all experiments.

peach =>

staron =>

....."

The output of the model may be like:

"Here's the corrected list of words:

slacrest => Seacrest

peach => peach (unchanged)

staron => station

....."

After we get the model output, we first remove the annotations like "(unchanged)" to avoid their influence. Then we convert all words to lowercase and compare the predicted words with corresponding labels in lowercase. If the predicted word is different from its label, we treat it as a failed attack; otherwise, it is a successful attack.

D The calculation of distance

We employ CTC Loss [10] to calculate the distance $Dist(Y, \mathcal{F}(X^{adv}))$. $Y = [y_1, y_2, \dots, y_n]$ (n means the gt character number) is the ground truth label, and $\mathcal{F}(X^{adv})$ denotes the model output. Since we employ the black-box attack setting where the model output $\mathcal{F}(X^{adv})$ includes a character sequence $[\hat{y}_1, \hat{y}_2, \dots, \hat{y}_m]$ (m means the predicted character number) and a confidence score sequence $[\hat{s}_1, \hat{s}_2, \dots, \hat{s}_m]$ ($s \in [0, 1]$), we need to preprocess it to facilitate the CTC loss calculation.

Firstly, we define the character dictionary $Dict$. Next, we generate a $m \times (l + 1)$ matrix T as,

$$T[a, b] = \begin{cases} \hat{s}_a, & Dict(\hat{y}_a) = b, \\ \frac{1}{l+1}, & Dict(\hat{y}_a) \neq b, \end{cases} \quad (18)$$

where $Dict(\hat{y}_a)$ represents the index of character $\hat{y}_a$ in the dictionary. Then we apply the log-softmax function along the second dimension of tensor T to obtain the matrix T^* of logarithmic probabilities. After that, we can calculate the CTC loss between T^* and Y .

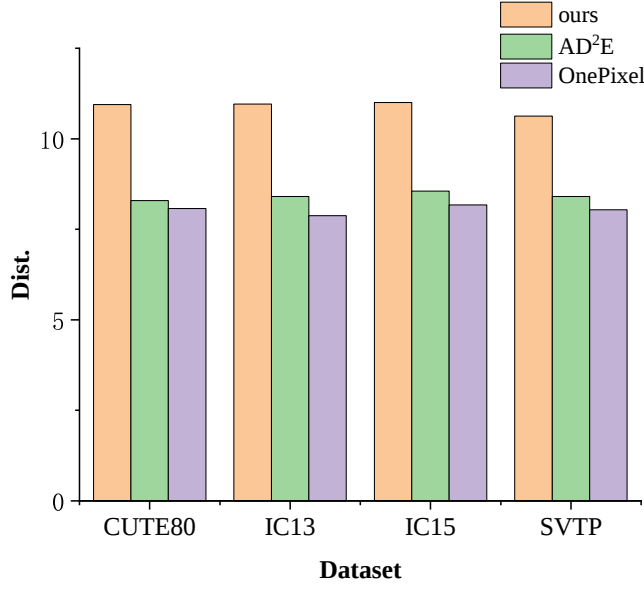


Figure 7: The average shortest distance between adversarial pixels on the CRNN model.

	Method	Origin	OnePixel	AD ² E	MPCS
CRNN	Image				
	Label	“disneyland”	“disneyland”	“disneyland”	“orsuagiera”
ASTER	Image				
	Label	“disneyland”	“disteeland”	“disneyland”	“otskeyland”
SAR	Image				
	Label	“disneyland”	“disneyland”	“disneyland”	“01.5.2y12nd”
CRNN	Image				
	Label	“chocolates”	“choicololes”	“chocololes”	“shiacidloied”
ASTER	Image				
	Label	“chocolates”	“chocalates”	“chacelates”	“chiacaiates”
SAR	Image				
	Label	“chocolates”	“chacolates”	“chocolates”	“thetciates”

Figure 8: Visualized comparison on the CRNN, ASTER, and SAR models. The underlined characters are incorrectly predicted. The red rectangles in images outline the characters with no nearby adversarial pixel. Zoom in to get a better view.

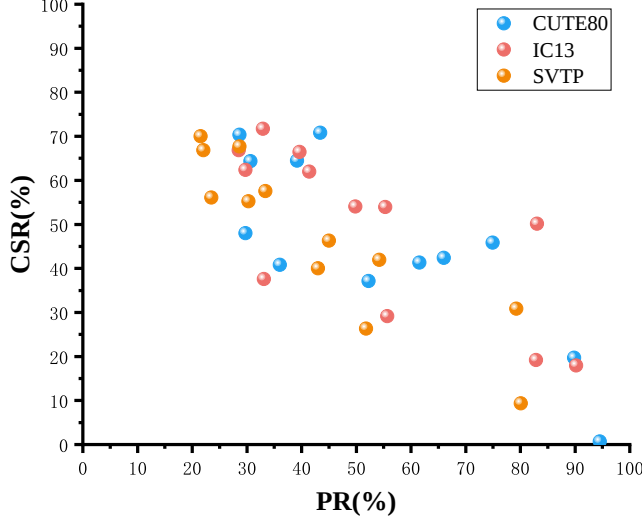


Figure 9: The relation between the correction success rate (CSR) by GPT-4o mini and the perturbation rate (PR) of texts.

E Attack pixel spatial concentration problem

In this section, we demonstrate the attack pixel spatial concentration problem that exists in previous pixel-level attack methods [33, 37]. Firstly, we give a quantitative analysis by calculating the average shortest distance between adversarial pixels. Given N adversarial pixels $[p_1, p_2, \dots, p_N]$, we calculate the distance between each adversarial pixel and its nearest adversarial pixel to get $[d_1, d_2, \dots, d_N]$, and then calculate the average value of d . As shown in Figure 7, our method has greater average distances on four datasets than previous methods. This is because our proposed multi-population mechanism restricts the adversarial pixel location in the horizontal direction.

Furthermore, we give a qualitative comparison by visualizing the attack results of OnePixel [33] and AD²E [37] with ours on the CRNN [29], ASTER [30], and SAR [20] models, respectively. The attack pixel number is 10. As shown in Figure 8, both OnePixel and AD²E have the adversarial pixel local aggregation problem when attacking the three STR models. The attack pixels concentrate within localized regions of the image, resulting in some characters having no attack pixels nearby. Consequently, these characters can be correctly recognized by the model. Meanwhile, they can only alter one or two predicted characters in most cases. Differently, our method effectively solves the adversarial pixel local aggregation problem through the multi-population mechanism. The attack pixels we generated are evenly distributed across the entire image along the horizontal direction, ensuring that each character in the image has at least one nearby attack pixel.

F The perturbation rate metric

To evaluate whether PR metrics can accurately reflect the irreversibility of attacks, we visualize the relationship between the correction success rate (CSR) by GPT-4o mini and the perturbation rate (PR) of texts. As shown in Figure 9, on the CUTE80 [28], IC13 [17] and SVTP [26] datasets, the overall correction success rate decreases as the perturbation increases. It demonstrates the effectiveness of the perturbation rate.

G Transfer attack on commercial OCR system

We implement transfer attacks on BaiduOCR (<https://cloud.baidu.com/doc/OCR/index.html>) using adversarial examples that can successfully attack SAR. As shown in Figure 10, the transfer attack achieves 53.44%, 63.51%, and 62.65% LPR on IC13, IC15, and SVTP datasets, respectively. Some qualitative results are also shown in Figure 10. These results demonstrate that our

Dataset	IC13	IC15	SVTP
PR	53.44	63.51	62.65
Image			
Pred	for → a	premier → b	kyro → u
Image			
Pred	30 → e	soymilk → 50	beads → beal

Figure 10: The result of conducting a transfer attack on BaiduOCR using adversarial samples obtained by attacking the SAR model.

$$\begin{aligned}
X: & \text{SEACREST} & Y: & \text{"seacrest"} \\
X^{adv}: & \text{SEACREST} & F(X^{adv}): & \text{"slagrese"}
\end{aligned}$$

Figure 11: Metric computation example.

method has good transferability and is still capable of misleading the model to predict a significant number of incorrect characters during the transfer attack.

H Metric computation example

We provide a real metric computation example in this section. As shown in Figure 11, the original image X has a predicted label $Y = \text{"seacrest"}$, and the adversarial example image X^{adv} is recognized by the STR model $\mathcal{F}$ as $\mathcal{F}(X^{adv}) = \text{"slagrese"}$. Then, the edit distance between their predicted labels is $D(Y, \mathcal{F}(X^{adv})) = 3$, and the length of the ground truth is $\text{len}(Y) = 9$. Therefore, the perturbation rate (PR) of this adversarial example is $PR = \frac{D(Y, \mathcal{F}(X^{adv}))}{\text{len}(Y)} = 33.33\%$. Besides, the L2 metric evaluates the average L2 distance between the original image and the adversarial example image: $L2 = \frac{1}{S} \sum_{i=1}^S \|X_i - X_i^{adv}\|_2$. In this example, the L2 distance is 3.62. As for the success rate before/after correction (SR/SR*), for example, if the number of samples is 1000, and 800 of them are successfully attacked, then the SR is 80%. Next, we use an LLM to correct the mispredicted texts. If 200 of them cannot be corrected, then the SR* is 20%.

I Evaluation on non-Latin scripts

To further examine the generality of our proposed method beyond Latin scripts, we additionally conducted experiments on a Chinese text recognition benchmark. Our main experiments rely on STR models provided by MMOCR and OpenOCR, both of which are trained on large-scale English datasets. Consequently, our primary evaluation focuses on four widely used English benchmarks.

However, as our method does not depend on any script-specific assumptions, we expect it to generalize to non-Latin scripts as well. To validate this, we performed a comparative study on the dataset introduced in "Benchmarking Chinese Text Recognition: Datasets, Baselines, and an Empirical Study." Specifically, we randomly sampled 200 images from this benchmark and applied both our attack and the baseline method to the IGTR model, which was trained on Chinese datasets by OpenOCR. Our method achieved a success rate (SR) of 51.85% and a perturbation rate (PR) of 34.44%, outperforming the baseline (SR: 38.89%, PR: 21.12%). These results demonstrate that our approach maintains strong effectiveness on non-Latin scripts, further confirming its generalizability across different languages and writing systems.