Unsourced Adversarial CAPTCHA: A Bi-Phase Adversarial CAPTCHA Framework

Xia Du, Xiaoyuan Liu, Jizhe Zhou*, Zheng Lin, Chi-man Pun, *Senior Member, IEEE*, Tao Li, Zhe Chen, *Member, IEEE*, Wei Ni, *Fellow, IEEE*, Jun Luo, *Fellow, IEEE*

***Abstract*—Traditional CAPTCHA schemes are increasingly vulnerable to automated attacks powered by deep neural networks (DNNs). Existing adversarial attack methods often rely on original image characteristics, resulting in distortions that hinder human interpretation and limiting their applicability in scenarios with no initial input images. To address these challenges, we propose the Unsourced Adversarial CAPTCHA (UAC), a novel framework generating high-fidelity adversarial examples guided by attacker-specified text prompts. Leveraging a Large Language Model (LLM), UAC enhances CAPTCHA diversity and supports both targeted and untargeted attacks. For targeted attacks, the EDICT method optimizes dual latent variables in a diffusion model for superior image quality. In untargeted attacks, especially for black-box scenarios, we introduce bi-path unsourced adversarial CAPTCHA (BP-UAC), a two-step optimization strategy employing multimodal gradients and bi-path optimization for efficient misclassification. Experiments show BP-UAC achieves high attack success rates across diverse systems, generating natural CAPTCHAs indistinguishable to humans and DNNs.**

***Index Terms*—adversarial attacks, diffusion model, CAPTCHA, Large Language Model**

I. Introduction

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) is a foundational cybersecurity mechanism. Its core function is to distinguish legitimate human users from automated bots. This technology presents specific computational challenges that are easily solvable by humans but difficult for machines. Common implementations can be categorized into text-based CAPTCHAs e.g., interpreting distorted alphanumeric sequences and solving arithmetic problems, image-based CAPTCHAs e.g., recognizing objects in image grids and completing slider puzzles, and audio-based CAPTCHAs. CAPTCHA systems establish critical barriers against automated abuse. They are widely deployed to prevent malicious activities, such as credential stuffing, spam registrations, ticket scalping, data scraping, and comment spam. By filtering out automated attacks, these mechanisms safeguard digital services while maintaining system integrity.

As AI systems become more proficient at tasks traditionally requiring human intelligence, they also pose unintended challenges in areas where maintaining human control and security is critical. For instance, image recognition technologies based on DNN model have grown highly effective at deciphering even the most complex CAPTCHA systems, which are

Fig. 1. Practical application scenarios of adversarial examples in CAPTCHA images. To allow readers can distinguish the source of the images, we label the adversarial examples generated by our method with red boxes, the adversarial examples generated by the traditional method with yellow boxes, and the clean images that are not labeled.

Xiaoyuan Liu and Xia Du are with the School of Computer and Information Engineering, Xiamen University of Technology, Xiamen, 361000, China (email: 2322071027@stu.xmut.edu.cn; duxia@xmut.edu.cn;).

Jizhe Zhou is with the School of Computer Science, Engineering Research Center of Machine Learning and Industry Intelligence, Sichuan University, Chengdu, China, 610020, China (email: yb87409@um.edu.mo).

Z. Lin and T. Li are with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pok Fu Lam, Hong Kong, China (e-mail: linzheng@eee.hku.hk; lthku999@connect.hku.hk).

Chi-man Pun is with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, 999078, China (email: cmpun@umac.mo).

Z. Chen is with the Institute of Space Internet, Fudan University, Shanghai 200438, China, and the School of Computer Science, Fudan University, Shanghai 200438, China (e-mail: zhechen@fudan.edu.cn).

W. Ni is with Data61, CSIRO, Marsfield, NSW 2122, Australia, and the School of Computing Science and Engineering, and the University of New South Wales, Kensington, NSW 2052, Australia (e-mail: wei.ni@ieee.org).

J. Luo is with the School of Computer Engineering, Nanyang Technological University, Singapore (e-mail: junluo@ntu.edu.sg).

* denotes Corresponding author.

Corresponding author: Jizhe Zhou (yb87409@um.edu.mo)

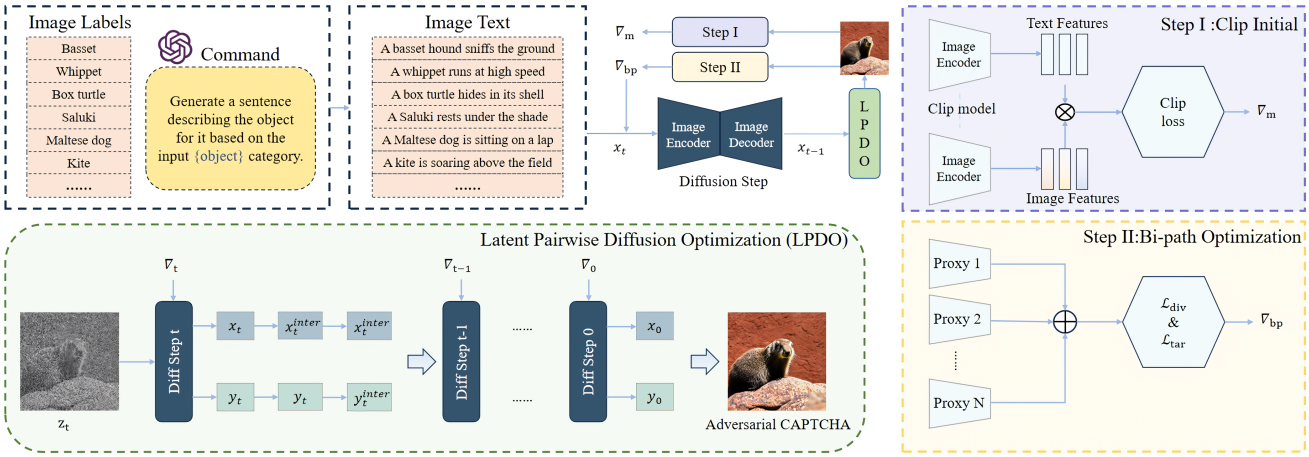

Fig. 2. Our proposed bi-path unsourced adversarial CAPTCHA (BP-UAC) attack framework

widely used to prevent automated bots and malicious activities. CAPTCHAs, originally designed to differentiate between humans and machines, are now increasingly vulnerable to attacks powered by advanced deep learning models. These models can break CAPTCHA defenses with remarkable precision, rendering conventional CAPTCHA systems less effective at safeguarding online platforms against automated attacks.

Adversarial attacks, which exploit the sensitivity of deep learning models to specific, carefully designed perturbations, have emerged as a powerful tool to evaluate model robustness [1], [2]. These attacks introduce subtle, targeted noise to mislead models into incorrect predictions, serving as an effective tool for evaluating the robustness of AI systems. This concept has been creatively extended to adversarial CAPTCHAs, where adversarial perturbations are used to enhance CAPTCHA robustness against automated recognition. This novel approach has started gaining traction in recent years, particularly among leading AI enterprises, and represents an innovative shift in security strategies for CAPTCHA systems [3], [4].

Although adversarial CAPTCHAs represent a significant advancement in securing systems against automated attacks, adversarial CAPTCHA systems still face several challenges. First, the limited variety of CAPTCHA images, often constrained by issues like copyright, restricts the diversity of generated adversarial examples, which in turn can inflate the cost of model training. Second, to effectively resist recognition attempts by various black-box models, adversarial CAPTCHAs require strong generalization abilities, often achieved by amplifying the intensity of adversarial perturbations. However, increasing the strength of these perturbations can introduce more visual noise, creating difficulties for legitimate users during the verification process and adversely impacting the usability and overall user experience of CAPTCHA systems. The potential of diffusion models offers a promising direction for further enhancing CAPTCHA security and usability, ensuring robustness against an even wider array of attack methodologies.

To address the above issues, we propose the unsourced adversarial CAPTCHA (UAC) attack framework in the white-box scenario. Specifically, before initiating the attack, we use a large language model to convert the attacker's input prompt into a concise and clear sentence, thereby ensuring the accuracy of the generated image. Unlike other adversarial attack methods based on diffusion model, we adopt EDICT's [5] dual latent variable optimization approach. This method approximates the initial input by performing one-step denoising at each time step, allowing the model gradients to propagate backward through the entire chain and resolving the instability and error accumulation issues inherent in adversarial attack methods based on diffusion model. Through the precise inversion process, UAC effectively avoids error propagation seen in traditional methods, ensuring the stability and high quality of adversarial example generation. In the more challenging and practical black-box attack scenario, we further introduce the bi-path unsourced adversarial CAPTCHA (BP-UAC). This method enhances optimization efficiency and attack performance by integrating multi-model gradient optimization and proposing a novel bi-path optimization strategy during the attack process. It ensures that the generated adversarial examples exhibit high transferability in black-box environments.

In summary, our contributions are as follows:

• We propose the first unsourced attack framework based on text-guided generation of adversarial examples, which enables the generation of adversarial examples during the diffusion process instead of relying on the original input images. The method effectively improves the diversity of generated adversarial examples and solves the optimization problem of adversarial example perturbation concealment, significantly improving the search space for adversarial attacks.

• We propose different attack frameworks for the white-box and black-box scenarios. In the white-box scenario, we propose the unsourced adversarial CAPTCHA (UAC) framework, which uses a large language model to convert the attacker's input into a clear sentence, ensuring accurate image generation. We also employ EDICT's dual latent variable optimization to

address instability and error propagation in other adversarial attack methods based on diffusion model, ensuring stable and high-quality adversarial examples. In the black-box scenario, we further propose BP-UAC, which enhances optimization efficiency by integrating multi-model gradient optimization and employing a bi-path optimization strategy, improving the transferability and robustness of adversarial examples.

• Extensive experiments have demonstrated that our approach can guarantee high attack success rate (ASR) for deep neural network models with different structures under different white- and black-box scenarios while generating unsourced Adversarial CAPTCHAs that are indistinguishable from clean examples.

To the best of our knowledge, we are the first to generate unsourced adversarial CAPTCHA using the diffusion model. At the same time, by means of a novel bi-path optimization strategy, we achieve for the first time a near 100% ASR against an unknown black-box model when generating adversarial examples using the generative model. This opens up new possibilities for adversarial research.

II. RELATED WORK

As the domain of adversarial attacks continues to mature, the endeavor to render adversarial perturbations increasingly inconspicuous has emerged as a pivotal topic.

In recent years, there have been various innovative approaches to adversarial image attacks. MUTEN [6] enhances the success rate and robustness of gradient-based adversarial attacks by utilizing diverse variant models. GADT [7] improves the migrability of adversarial examples by optimizing the data enhancement parameters, which is particularly suitable for black-box and query attacks. MGAA [8] improves the mobility of attacks using meta-learning methods and enhances the success rate of attacks by narrowing the difference in gradient directions in white-box and black-box environments. U-GAN [9] constructs unconstrained adversarial exexample through GAN networks, breaking through the restriction on the perturbation range of traditional gradient-based attack methods, making the generated adversarial example more aggressive and effective in bypassing many existing defense mechanisms. AdvDiff [10] generates unconstrained adversarial example by utilizing the denoising process of the diffusion model, demonstrating the potential of this model in adversarial attacks. DiffAttack [11] proposes an attack strategy specifically for countering the purification defense of the diffusion model, which successfully generates adversarial example by bypassing the purification process, revealing potential loopholes in the defense mechanisms of the diffusion model and driving new challenges in the field of adversarial attacks.

Not only in the field of image recognition, but also in the field of security, researchers have similarly tried to incorporate undetectable perturbations into CAPTCHAs to fight against machine intrusion. Shi *et al.* [12] proposed an aCAPTCHA system, which enhances the security of ordinary CAPTCHAs by generating adversarial examples. This approach makes it difficult for deep learning models to recognize them by adding adversarial perturbations to the images, while still allowing human users to pass normally. Wen *et al.* [13] explored methods to generate stronger CAPTCHA challenges through adversarial attacks such as Iterative FGSM (I-FGSM) [14] and DeepFool [15]. Their method, which focuses on perturbation processing, improves the resistance of these visual challenges to machine learning models, making them harder for automated systems to crack. Zhang *et al.* [17] explored the application of adversarial examples on different image classes of CAPTCHAs, and investigated how to improve the robustness of image CAPTCHAs by using adversarial methods by analyzing the effect of adversarial example perturbation on it.

While all of the above methods optimize the perturbations at different levels, it is always necessary to generate image adversarial examples based on the corresponding benign images, and the fundamental flaws of the adversarial attack methods in this regard limit the search space of the perturbations and the high intensity of the attack performance can make the perturbations too noisy, which affects their visualization by humans. In contrast to the above work, our focus is on introducing gradient information of DNNs in the diffusion generation process and enabling their stable integration into the diffusion model to generate high-quality adversarial samples.

TABLE I
CHARACTERISTICS OF ADVERSARIAL ATTACK METHODS. ○/◑/● INDICATE THE LOW/MIDDLE/HIGH QUALITY OF THE METHODS.

Attack Methods	Method Characteristics			
	Dependent input	Diversity	Quality	transferability
MUTEN [6]	○	○	◑	○
GADT [7]	○	○	◑	◑
MGAA [8]	○	○	◑	●
U-GAN [16]	○	◑	●	○
DiffAttack [11]	○	◑	●	○
AdvDiff [10]	○	◑	●	◑
UAC (ours)	●	●	●	○
BP-UAC (ours)	●	●	●	●

III. BACKGROUND

A. Traditional adversarial scenarios

Traditional adversarial attacks can be categorized into untargeted and targeted attacks. To cope with different application scenarios, our approach discusses targeted and untargeted attacks, respectively. Targeted attacks aim to shift the model's predictions to a targeted category specified by the attacker. This attack can be considered "targeted" because the attacker wants the model to output a specific mislabel when confronted with a modified example. Specifically, a traditional targeting attack is formulated as follows :

$$\begin{aligned} x' &= \arg\min_{x'} \ell(f(x'), y_{\text{target}}) \\ &\text{s.t. } x' = x + \delta \end{aligned} \tag{1}$$

Instead, the untargeted attack aims to make the prediction results different from the original category without specifying the wrong category. The attacker only needs the model to out-

put any of the wrong categories when it sees the antagonistic example.

$$x' = \arg\max_{x'} \ell(f(x'), y_{\text{target}}) \quad \text{s.t. } x' = x + \delta \tag{2}$$

From Eq. 1 and Eq. 2, it can be concluded that **traditional adversarial attacks mainly rely on specify benign images x as the basis for generating adversarial examples x'.**

B. Diffusion model

Denoising Diffusion Models (DDMs) [18] are a type of generative model that produces images by progressively removing noise, starting from pure noise and iterating until a clean image is generated. The process consists of two main stages: forward diffusion and reverse diffusion. Forward Diffusion Process: In this stage, clean images are corrupted by adding Gaussian noise in a series of steps, ultimately transforming the image into pure noise. At each step t, the noisy image x_t is computed as:

$$x_{t+1} = \sqrt{a_t x_t + \sqrt{1-\alpha_t}\epsilon} \tag{3}$$

where is a_t noise-scaling factor and ϵ represents Gaussian noise.

Reverse Diffusion Process: Starting from pure noise, the model denoises the image iteratively, generating a series of intermediate images that gradually resemble the original image. The reverse process is modeled by:

$$p_\theta\left(x_{t-1} \mid x_t\right) = \mathcal{N}\left(x_{t-1}; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right) \tag{4}$$

where μ_θ and Σ_θ are the learned parameters for mean and variance, guiding the denoising process.

To accelerate sampling, Denoising Diffusion Implicit Models (DDIM) [19] introduce a non-stochastic, efficient sampling approach. DDIM avoids fully stochastic sampling, offering a faster alternative by approximating the reverse steps with an implicit formula:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \cdot \mu_\theta(x_t, t) + \sqrt{1-\bar{\alpha}_{t-1}} \cdot \epsilon \tag{5}$$

This formula reduces the number of steps required to generate an example, making the model more practical for real-time applications.

C. EDICT

In recent years, EDICT [5] has been proposed to address the problem of reconstruction and content loss in model-generated images due to error propagation during DDM [20] backpropagation. EDICT mathematically and precisely inverts real and model-generated images by coupling latent noise variables. Specifically, EDICT utilizes two alternating sequences, x_t and y_t, in the symmetric coupling layer of the DDM. These sequences are used to track and reconstruct the states of the latent variables during the generation process. The forward updating of x_t and y_t during DDM inverse propagation is then realized through the following coupling steps:

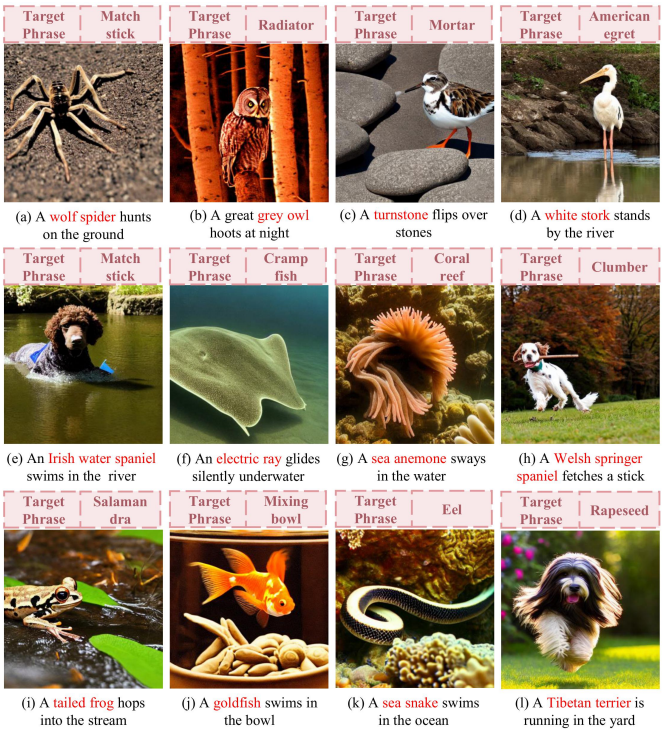

Fig. 3. The adversarial examples generated by our proposed method.

$$\begin{aligned} x_t^{inter} &= a_t \cdot x_t + b_t \cdot \Theta_{(t,C)}(y_t) \\ y_t^{inter} &= a_t \cdot y_t + b_t \cdot \Theta_{(t,C)}(x_t^{inter}) \\ x_{t-1} &= p \cdot x_t^{inter} + (1-p) \cdot y_t^{inter} \\ y_{t-1} &= p \cdot y_t^{inter} + (1-p) \cdot x_{t-1} \end{aligned} \tag{6}$$

where a_t and b_t denote the coefficients from time to time, EDICT includes the internal transformations and backward updates of the variables. It generates a new x_{t-1} and y_{t-1} by mixing the internal representations of x_t and y_t through the averaging parameter $p \in (0,1)$, which is derived using linear equations through the above equation to ensure that the entire diffusion process can be performed exactly in reverse:

$$\begin{aligned} y_{t+1}^{inter} &= (y_t - (1-p) \cdot x_t)/p \\ x_{t+1}^{inter} &= (x_t - (1-p) \cdot y_{t+1}^{inter})/p \\ y_{t+1} &= (y_{t+1}^{inter} - b_{t+1} \cdot \Theta_{(t+1,C)}(x_{t+1}^{inter}))/a_{t+1} \\ x_{t+1} &= (x_{t+1}^{inter} - b_{t+1} \cdot \Theta_{(t+1,C)}(y_{t+1}))/a_{t+1} \end{aligned} \tag{7}$$

IV. METHODOLOGY

In this section, we introduce the UAC framework in the context of white-box attacks, where we integrate both the LLM and EDICT frameworks to generate unsourced adversarial examples while ensuring the quality of the generated images. Subsequently, to address the broader applicability and higher complexity of black-box attack scenarios, we further extend the UAC framework by incorporating the clip model and the bi-path optimization strategy. This enhancement ensures semantic consistency during the diffusion model's generation

process and overcomes the limitations of traditional search spaces, enabling the generation of unsourced adversarial CAPTCHAs suitable for practical application scenarios. The overall framework is shown in Fig. 2.

A. Threat model

Considering the distinct contexts of image recognition and CAPTCHA security, we adopt a threat model based on image classification and examine both white-box and black-box attack scenarios. In this scenario, the attacker only needs to provide the target class and the desired adversarial instance class during the generation phase and manipulate the test image during the inference phase without involvement in the training phase of the model.

B. Unsourced adversarial CAPTCHA

1) Prompt guidance: Firstly, we introduce an LLM to ensure the accuracy and diversity of the generation process. Although the initial attack aims to generate the corresponding adversarial examples based on the object categories entered by the attacker, in the experiments, we found that the use of only short prompts P' tends to lead to generation errors. This is mainly because brief prompts lack sufficient contextual information, making the diffusion model ambiguous in understanding the target image features. Diffusion models rely on the semantics of the prompter to gradually guide image generation. Still, when the prompter is too simple, for example, a single word or an ambiguous description, the model may associate it with multiple different potential image features, resulting in generation results that deviate from expectations.

To address this issue, we introduce an LLM [21]–[23] to enhance the semantic information of the prompt. The LLM guides the generator's inputs by generating richer and contextually relevant prompts like "A goldfish swims in the bowl", which not only ensures the diversity and stability of the generated examples but also enables the attacker to generate more accurate adversarial examples given the category and the specific target. That is :

$$P' = f_{\mathrm{LLM}}(P) \tag{8}$$

Next, we use a diffusion model G to generate images step by step, and the generation process of this model is conditionally guided by the extended prompt. Specifically, we first input the extended prompt P' into G for conditional generation, and G generates a representation of the intermediate latent variable z_t based on the latent variables and the prompt at each time step t. Specifically, this process can be formulated as

$$I_t = G(z_t, P', \epsilon_t) \tag{9}$$

where ϵ_t is the noise term we introduce in the generation process t to ensure diversity in the generation.

2) Latent pairwise diffusion optimization: In our approach, gradient guidance and merging of latent variables are key steps to ensure that the representation of latent spaces during generation accurately guides the optimization of the generator G. Specifically, the gradient of the target model f needs to be used first to guide the latent variable pairs:

$$\begin{aligned} x_{t-1}^{\mathrm{inter}} &= x_t - \eta_x \nabla_{x_t} \ell(x_t, y_{target}) \\ y_{t-1}^{\mathrm{inter}} &= y_t - \eta_y \nabla_{y_t} \ell(y_t, y_{target}) \end{aligned} \tag{10}$$

where L denotes the loss function, ∇ indicates the gradient of f, y_{target} indicates the attacker specifies the target label of the attack, and η denote the learning rate of the latent variable pairs (x_t, y_t)

After the gradient-guided optimization, we merge the two optimized latent variable pairs (x_{t-1}, y_{t-1}) into a single unified latent variable z_{t-1} to continue the diffusion model generation process:

$$\begin{gathered} z_{t-1} = \alpha \cdot x_{t-1}^{\mathrm{inter}} + (1-\alpha) \cdot y_{t-1}^{\mathrm{inter}} \\ \text{s.t. } \alpha \in [0,1] \end{gathered} \tag{11}$$

where α is a weight parameter to control the proportion of the contribution of the two latent variables in the merger. The goal of the merging process is to fuse the optimized properties of the two latent spaces to produce a more robust and integrated latent representation that better supports the generation of diffusion models.

3) Adversarial example synthesis: In UAC, P' is provided to the generator G as the input and initial condition for the adversarial example generation process. During the reverse diffusion process of the generator, EDICT is employed as a reversible temporal diffusion framework to iteratively optimize the latent variable pair (x_t, y_t). At each step, gradient guidance is utilized to progressively refine the generated image toward the target class y_{target}. Specifically, for each latent variable

Algorithm 1 Adversarial example synthesis in UAC

Require: Initial prompt P from the attacker, generator G, LLM f_{LLM}, target category y_{target}, iterations T, learning rates η_x, η_y, weight α, cross-entropy loss L
Ensure: $f(I_t) = y_{target}$
1: $P' = f_{LLM}(P)$
2: Initialize latent variables x and y
3: **for** $t = T$ to 1 **do**
4: $I_t = G(z_t, P', \epsilon_t)$
5: **if** $f(I_t)$ = y_{target} **then**
6: exit the loop and proceed to Step 17
7: **else**
8: Introduce gradients ∇ of f
9: Compute loss L between generated image I_t and y_{target}
10: Update latent variables x and y:
11: $x_{t-1}^{inter} = x_t - \eta_x * \nabla_{x_t} \ell(x_t, y_{target})$,
12: $y_{t-1}^{inter} = y_t - \eta_y * \nabla_{y_t} \ell(y_t, y_{target})$
13: Merge optimized latent variables into a unified representation:
$z_{t-1} = \alpha * x_{t-1}^{inter} + (1-\alpha) * y_{t-1}^{inter}$
14: $I_{t-1} = G(z_{t-1})$
15: **end if**
16: **end for**
17: **return** Adversarial image I^*.

z_t in the generation process, joint optimization of (x_t, y_t) is performed, leveraging the gradient information of the target loss to guide the latent variable closer to the target class during optimization. At each time step t UAC adjusts the variable pair (x_t, y_t) through gradient guidance, controlling the diffusion and denoising processes. Ultimately, the synergy between gradient guidance and latent variable optimization ensures that the generated adversarial examples not only precisely align with the target conditions but also achieve optimal performance in terms of distribution consistency and attack efficacy. In summary, our adversarial example generation can be expressed by the following equation:

$$z' = \arg\min_{z_t} \ell(z_t, P, y_{target}) \tag{12}$$

where z' denotes the final latent variant. After this, we pass the final latent variable z' to the generator G to obtain the final output of the adversarial examples $I^* = G(z')$. The specific adversarial example synthesis process is shown in algorithm 1.

C. Bi-path unsourced adversarial CAPTCHA

Although UAC can effectively deceive the machine into producing false recognition results by utilizing the original model information in a white-box environment, the premise of a white-box environment is that the attacker has access to the complete structure and parameters of the target model. However, in practical applications, especially in adversarial CAPTCHA scenarios, this premise often does not hold because the detailed information of the target model is not available, and direct access to the original image source is also challenging due to the copyright issues of the image source. Therefore, we extend the UAC method into a more practical black-box attack method called BP-UAC.

In BP-UAC, before the attack process, we introduce the gradient of the clip model [24] to guide G to generate the initial states x_m and y_m, and synthesize the two initial states obtained into a temporary latent variable z_m through Eq. 11 after the conclusion of the bootstrap optimization of the clip model. To ensure the consistency of the generated image with the P'.

This consistency not only enhances the naturalness and visual quality of generated images but also prevents distribution drift. Additionally, incorporating clip model's gradient significantly improves the stealthiness of adversarial examples by aligning the generated images with the prompt description, making them harder for human observers and detection methods to identify.

Then, we integrate the gradients of multiple models during the attack and propose a bi-path optimization strategy to address the limitations of white-box environments and adapt to black-box scenarios. By merging predictions from multiple models, we can approximate the behavior of the target model, enhancing the ASR on unknown models. The bi-path strategy balances the loss from the target class and second-highest posterior probabilities in the target model's output, effectively capturing vulnerabilities in the decision boundary to improve adversarial robustness and ASR.

Compared to traditional attacks, BP-UAC excels in generating perturbations by exploring the input space more comprehensively through bi-path optimization strategy. This increases the diversity of adversarial examples and enhances the attack's robustness, allowing it to exploit weaknesses in the target model more effectively. As a result, BP-UAC achieves higher success rates in complex and uncertain black-box environments.

To accomplish the above, we need to add the assumption that we have three known proxy models f_1 ,f_2 and f_3 , which have classification predicted probability distribution of $g_1(x)$, $g_2(x)$, $g_3(x)$, and parameters β_1, β_2 and β_3, respectively. Using the predict probability distribution of these models, we update the latent variables x_t and y_t. Based on UAC, we replace the known model gradients in Eq. 10 and Eq. 11 with the weighted average of the gradients from the three proxy models to enhance the attack's effectiveness against an unknown black-box model. Specifically, our integrated classification predicted probability distribution g_{x_t} and g_{y_t} can be expressed by the following equation:

$$\begin{aligned} g_1(x) = \mathrm{u}_1 &= \left[u_1^{(1)}, u_1^{(2)}, \ldots, u_1^{(M)}\right]^T \\ g_2(x) = \mathrm{u}_2 &= \left[u_2^{(1)}, u_2^{(2)}, \ldots, u_2^{(M)}\right]^T \\ g_3(x) = \mathrm{u}_3 &= \left[u_3^{(1)}, u_3^{(2)}, \ldots, u_3^{(M)}\right]^T \\ g_{x_t} &= \frac{\beta_1 g_1(x) + \beta_2 g_2(x) + \beta_{3g_3(x)}}{\beta_1 + \beta_2 + \beta_3} \end{aligned} \tag{13}$$

where $g_1, g_2, g_3 : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^M$, the calculation of g_{y_t} is the same as for g_{x_t}.

Leveraging the collective gradient information from these proxy models, we further guide the model to generate adversarial examples toward the second-highest and target class probability classes. The bi-path optimization strategy identifies a more robust path across different loss spaces, enhancing the transferability and success rate of adversarial examples in black-box models. This approach effectively improves the performance of black-box attacks, making adversarial examples more deceptive and robust.

Specifically, we focus on guiding the generation process by minimizing the loss associated with both the second-highest and target class probability classes:

$$\begin{aligned} \ell_{div} &= -\ell(x_t, y_{second}) \\ \ell_{tar} &= \ell(x_t, y_{target}) \end{aligned} \tag{14}$$

Finally, the new loss ℓ_{BP} is derived by aggregating the losses from multiple directional objectives, effectively balancing the guidance effects of both the target and auxiliary classes during optimization, thereby overcoming the challenges associated with black-box model attacks:

$$\ell_{x_t} = \frac{\ell_{div} + \ell_{tar}}{2} \tag{15}$$

Compute gradient information through loss ℓ_{BP}:

$$\nabla_{x_t} = \frac{\partial \ell_{x_t}}{\partial g x_t} \tag{16}$$

Algorithm 2 Adversarial example synthesis in BP-UAC

Require: Initial prompt P from the attacker, clip model M, origin category y_{origin}, learning rates η_x, η_y, proxy models f_1, f_2 and f_3, target model f_t

Ensure: $f(I_t) \neq y_{origin}$

$P' = f_{LLM}(P)$
Initialize latent variables x and y
Using Eq. 9 and Eq. 10 optimize the latent variables x and y through m, get x_m and y_m
for $t = T$ to 1 **do**
  $I_t = G(z_t, P', \epsilon_t)$
  if $f_t(I_t) \neq y_{origin}$ **then**
    exit the loop and proceed to Step 20
  else
    Compute the probability distributions $g_1(x)$, $g_2(x)$, $g_3(x)$ of the proxy models
    Integrate $g_1(x)$, $g_2(x)$, $g_3(x)$ as g_{x_t} base on Eq. 12
    Compute loss of second-highest and target class ℓ_{div}, ℓ_{tar}
    Integrate ℓ_{div}, ℓ_{tar} as ℓ_{x_t} base on Eq. 14
    Compute gradient ∇ base on Eq. 15
    Update latent variables x and y:
    $x_{t_1}^{inter} = x - \eta_x * \nabla_{x_t} \ell_{x_t}$
    $y_{t-1}^{inter} = y - \eta_y * \nabla_{y_t} \ell_{y_t}$
    Merge optimized latent variables into a unified representation:
    $z_{t-1} = \alpha * x_{t-1}^{inter} + (1 - \alpha) * y_{t-1}^{inter}$
    $I_t = G(z_{t-1})$
  end if
end for
return Adversarial image I^*.

TABLE II
SOME OF THE PROMPT FOR LLM AND DIFFUSION MODEL.

Origin Class	Input Prompt
Tench	A tench is swimming in the pond
Goldfish	A goldfish swims in the bowl
White shark	A white shark is hunting
Hummingbird	A hummingbird hovers near the flower
White stork	A white stork stands by the river
Marmot	A marmot stands on a rock
Otter	An otter swims in the river
Gila monster	A Gila monster hides under a rock
School bus	A school bus carries the children
Toilet paper	Toilet paper is rolled on the holder

The specific adversarial example synthesis process is shown in algorithm 2.

V. EXPERIMENTS AND RESULTS

In this section, we will first validate the attack effectiveness of our method in white-box and black-box scenarios, respectively, by comparing the attack effectiveness with other methods and calculating the attack effectiveness of existing methods in the face of unknown models. Next, we launch a comprehensive ablation experiment to compare the attack effect, the quality of the generated images, and the efficiency of the attack, respectively, to prove the usefulness of our module and the rationality of the parameter settings.

A. Experimental setup

1) Dataset: Imagenet is one of the most commonly used datasets for image classification, image detection, and image localization in deep learning. Although this experiment did not require the images provided by the dataset as the basis for the input, to ensure the experiment's rigor, we chose to refer to Imagenet with 1000 classifications as input and attacked categories in this experiment.

2) Environment: All experiments are carried out on an Ubuntu 22.04.4 Server with an Intel(R) Xeon(R) Gold 6342 CPU @ 2.80GHz and NVIDIA A40 with 48G of memory.

3) Attack Model: All tasks in this experiment were performed based on eight models, including SeResNeXt101 [25], MobilenetV2 [26], Resnet50 [27], Resnet152, Googlenet [28], Efficientnet [29], inceptionV3 [30] and Alexnet [31].

4) Evaluation Metrics: To demonstrate the effectiveness and superiority of our sparse attack method, we employ the Attack Success Rate (ASR), the Clip Score [32], and the Average Attack Step in comparison with other methods. Specifically, they are defined as:

ASR measures the proportion of inputs successfully manipulated to produce the attacker's desired erroneous outputs.

$$\text{ASR} = \frac{N_{adv}}{N_{total}} \times 100\% \tag{17}$$

where N_{adv} is the number of adversarial examples that successfully mislead the target model, while N_{total} denotes the total number of generated adversarial examples.

Clip Score evaluates semantic alignment between images and text using the Clip model, commonly applied in tasks like image generation and text-to-image synthesis.

$$\text{Clip Score(I, C)} = max(100 * cos(E_I, E_C), 0) \tag{18}$$

where I is the input image, C is the input text description, and $cos(E_I, E_C)$ denotes the cosine similarity between the image vector E_I and the text vector E_C.

B. Comparison and analysis of attack effects

To prove the superior attack performance we demonstrate in white-box scenarios and black-box scenarios, we compare it with classic white-box attack methods such as FGSM [33], BIM [34], and PGD [35], and to prove the superior attack performance of our method, we compare our method with advanced adversarial attack methods based on diffusion model under unknown models, such as U-BigGAN [9], AdvDiffuse [36], DiffAttack [11], AdvDiff [10]. In addition, to prove that BP-UAC can show stable and excellent attack performance against unknown models, we conducted comprehensive experiments on seven traditional DNNs.

As shown in Table IV, we compare the ASR of four classical adversarial attack methods (PGD, FGSM, BIM, and

TABLE III
TARGETED ATTACK SUCCESS RATES (%) AGAINST BLACK-BOX TARGET MODELS WITH THE FOUR SOURCE MODELS. FOR EACH ATTACK, WE ALSO REPORTED THE AVERAGE ATTACK SUCCESS RATE. THE BEST RESULTS ARE IN BLUE. * INDICATES THAT SURROGATE MODEL AND TARGET MODEL ARE SAME.

Surrogate Model: RN-50	Method									
Target Model	SAE	ADer	ReColorADV	cAdv	tADV	NCF	ACE	ColorFool	ACA	Ours
RN-50	88.0*	55.7*	96.4*	97.2*	99.0*	99.1*	90.1*	91.4*	88.3*	98.1*
RN-152	46.5	7.8	33.3	37.0	30.2	15.2	21.0	60.5	61.7	79.9
MN-v2	63.2	15.5	40.6	44.2	43.4	32.8	41.6	71.2	69.3	89.3
Dense-161	41.9	8.4	28.3	36.8	28.8	16.1	18.6	48.5	61.9	87.9
Eff-b7	28.8	11.4	19.2	34.9	21.6	12.7	15.4	32.4	60.3	61.3
Inc-v3	25.9	7.7	17.7	25.3	27.0	9.4	9.8	33.6	61.6	75.0
Average	49.05	17.75	39.25	45.9	41.7	30.9	32.75	56.3	67.2	81.9
Surrogate Model: MN-v2	Method									
Target Model	SAE	ADer	ReColorADV	cAdv	tAdv	NCF	ACE	ColorFool	ACA	Ours
RN-50	53.2	8.4	33.7	39.6	31.5	17.9	25.7	65.9	62.6	88.9
RN-152	41.9	7.1	26.4	29.9	24.5	12.6	15.4	56.3	56.0	79.6
MN-v2	90.8*	56.6*	97.7*	96.6*	99.9*	99.1*	93.3*	93.2*	93.1*	91.6*
Dense-161	38.0	7.7	24.7	33.9	24.3	12.4	15.3	43.5	55.7	84.4
Eff-b7	26.9	10.9	20.7	32.7	22.4	11.7	13.4	33.0	51.0	69.5
Inc-v3	22.5	7.6	18.6	26.8	27.2	9.5	9.5	33.6	56.8	64.5
Average	45.55	16.4	37.0	43.25	38.3	27.2	28.8	54.25	62.5	79.75

TABLE IV
THE PERFORMANCE COMPARISON OF UAC AND SOME TRADITIONAL WHITE-BOX ADVERSARIAL ATTACK ON TARGET ATTACK SUCCESS RATE (TSR) AND UNTARGET ATTACK SUCCESS RATE (USR).

Attack Methods		**TSR Label** : Gila Monster						
		ResNet50	ResNet152	SeResNext101	Effecientnet	Googlenet	MobileNetV2	Alexnet
PGD	USR	99.8%	100%	99.8%	99.9%	99.6%	99.7%	99.9%
PGD	TSR	69.8%	65.3%	47.7%	46.8%	77.3%	81.1%	65.0%
FGSM	USR	100%	99.7%	100%	99.9%	100%	100%	100%
FGSM	TSR	75.3%	60.0%	62.3%	53.6%	81.7%	83.5%	64.3%
BIM	USR	100%	100%	100%	99.9%	100%	100%	100%
BIM	TSR	75.3%	67.9%	62.3%	53.6%	81.7%	83.5%	64.3%
UAC	USR	100%	100%	100%	100%	100%	100%	100%
UAC	TSR	100%	100%	100%	100%	**99%**	100%	**99%**

UAC) on seven deep learning models, including the untargeted attack success rate (USR) and the targeted attack success rate (TSR), with the target category "Gila Monster". The experimental results show that the UAC method exhibits a nearly 100% USR and TSR in all models and scenarios. It significantly outperforms the other attack methods in both USR and TSR, demonstrating its overall robustness and attack effectiveness advantages. In contrast, although the other methods also achieve a high success rate (more than 99%) in untargeted attack scenarios, the ASR is significantly lower in targeted scenarios, especially for the Efficientnet and SeResNext101 models, which suggests that there is still a large room for optimization of the performance of the traditional adversarial attack methods in targeted attacks. The results of this experiment show that untargeted attacks are very mature under the existing methods, and almost all the methods can achieve a very high USR under the white-box untargeted attack scenarios, whereas the effectiveness of targeted attacks is limited by the complexity of the method design and model architecture. The UAC methods perform well in both targeted and untargeted attack scenarios, show good generalization and robustness, and provide a strong benchmark for subsequent research.

As shown in Table V, we have evaluated the ASR using the BP-UAC method on different combinations of baseline and targeted attack models. Overall, the migration ASR of

TABLE V
THE PERFORMANCE COMPARISON OF BP-UAC AND SOME STATE-OF-ART ADVERSARIAL ATTACK METHODS RELYING ON GENERATIVE MODELS ON ASR.

Baseline Models			**Attack method** : BP-UAC						
			ResNet50	ResNet152	SeResNext101	Effecientnet	Googlenet	MobileNetV2	Alexnet
ResNet50	SeResNext101	Googlenet	100%	98.5%	100%	99.1%	100%	97.2%	98.0%
ResNet50	Effecientnet	Alexnet	100%	98.0%	98.0%	100%	97.6%	97.9%	100%
Googlenet	Effecientnet	Alexnet	97.7%	96.6%	97.2%	100%	100%	95.5%	100%
ResNet152	MobileNetV2	Alexnet	95.4%	100%	95.8%	96.2%	97.2%	100%	100%
ResNet152	SeResNext101	MobileNetV2	99.3%	100%	100%	99.4%	99.2%	100%	99.2%

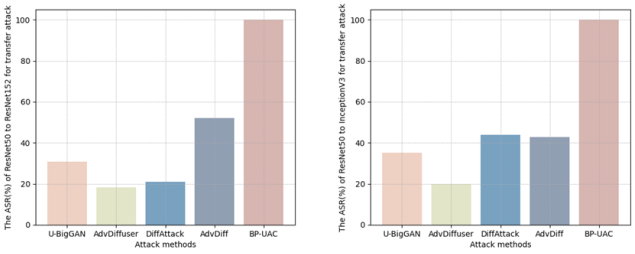

Fig. 4. Attack success rate of transfer attacks based on Resnet50 (left) and InceptionV3 (right).

the BP-UAC method is higher than 99% on all model combinations, regardless of the architectural differences between the baseline and target models, which demonstrates the strong generalization ability and robustness of the BP-UAC method in cross-model attacks. Especially on the models of ResNet series, Googlenet, and MobileNetV2, the success rates of the attacks are almost indistinguishable whether they are the baseline model or the target model, which demonstrates that the antiperturbation generated by the BP-UAC method is highly adaptable and stable. It is worth noting that even for models with relatively complex architectures (e.g., SeResNext101 and Efficientnet), the success rate of BP-UAC's migration attack also remains exceptionally high, further demonstrating its performance advantage in dealing with diverse deep learning models. Overall, the experimental results fully validate the efficiency and robustness of the BP-UAC approach in cross-model scenarios, reflect its potential as a generalized adversarial attack method, and provide critical experimental benchmarks and theoretical support for subsequent research.

As shown in Fig 4, we use the BP-UAC method to compare the migration attack performance with four adversarial attack methods based on generative models using ResNet50 as the baseline model on ResNet152 and InceptionV3 models. From the experimental data, it can be seen that the existing adversarial attack methods based on diffusion model have poor attack performance when facing unknown models, and only the AdvDiff method can barely exceed 50% ASR; in contrast, our method can still show a success rate of close to 100% when facing unknown models, which further illustrates the superiority of our method when facing unknown models. This further demonstrates the superiority of our method in the face of the unknown model. In the production of CAPTCHA, since the illegal model used by the attacker is unknown, therefore, in the production of adversarial CAPTCHA defense, the method is required to have a high degree of robustness in the face of the unknown model. This experiment proves that our method can significantly resist the theft of the CAPTCHA by the illegal attacker.

C. Ablation study

We compared adversarial examples generated using class names from ImageNet as prompts with those generated using sentences as prompts, and further incorporated Clip model gradients for optimization. As shown in the first column of Figure 5, although both types of prompts achieved an ASR of 100%, the average number of steps required for a successful attack was noticeably higher when the prompt was a single word. Additionally, the Clip Score of the generated images was lower, indicating poorer quality of the adversarial examples. By incorporating clip model gradients during the generation process, we ensured semantic consistency and visual alignment, which further improved the quality of the generated examples. Overall, the combination of prompt optimization through LLM and gradient optimization via clip model significantly enhanced both attack efficiency and adversarial example quality. Furthermore, as visually demonstrated in Figure 6, when the prompt consisted of a single word, the simplicity and lack of information in the semantic expression led to the generation of incorrect or unrealistic images. In contrast, our optimization strategy effectively guided the generation model to output high-quality images that better aligned with the target class. This comparison clearly demonstrates the crucial role of both LLM and clip model in our method, as they not only improved the quality of the generated examples but also significantly increased the attack efficiency.

D. Robustness evaluation

In traditional adversarial attack scenarios, various defense preprocessing methods are commonly employed to reduce the efficacy of adversarial samples. However, in practical applications of adversarial CAPTCHA systems, if an attacker preprocesses the CAPTCHA using these defense mechanisms during the attack process, the security of the CAPTCHA is significantly compromised. To validate the robustness of

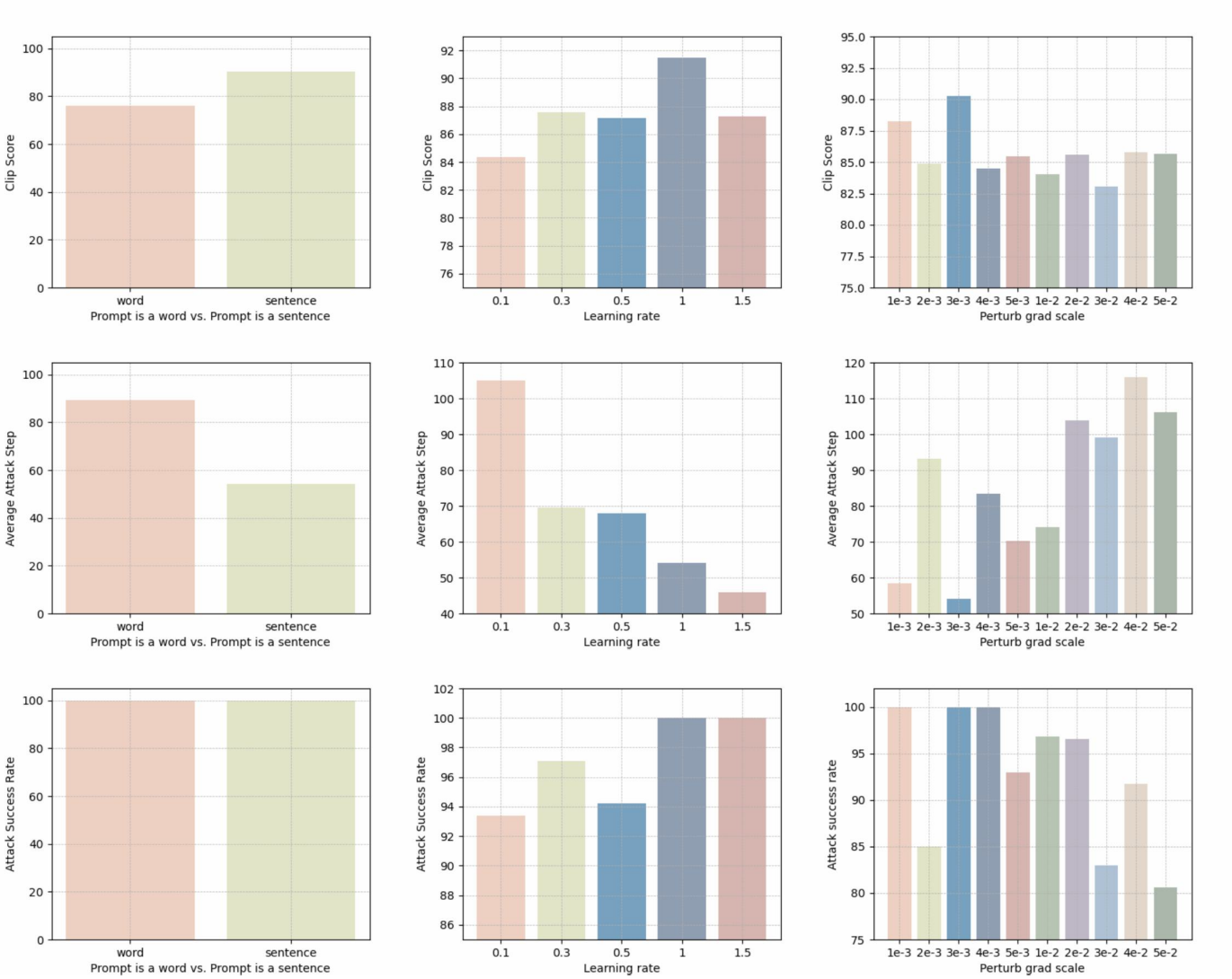

Fig. 5. Attack success rate, average attack steps, and Clip Score of different settings.

TABLE VI
THE ASR OF ADVERSARIAL ATTACKS WHEN AGAINSTING DIFFERENT DEFENCE METHODS.

Attack Models	**Baseline Model** : Resnet50			
	NRP	R&P	RS	HGD
U-BigGAN	30.9%	14.2%	34.5%	22.6%
AdvDiffuser	40.5%	15.4%	38.4%	10.8%
DiffAttack	38.5%	23.7%	40.8%	20.5%
AdvDiff	74.2%	56.8%	82.8%	53.8%
BP-UAC	100%	97.1%	100%	95.7%

our proposed adversarial CAPTCHA system, this study extensively analyzes the impact of various defense strategies including NRP [37], RS [38], R&P [39] and HGD [40] on adversarial samples. As illustrated in Figure 4, the potency of attack methods based on generative models is typically diminished after defense preprocessing, likely due to increased uncertainty in samples, which reduces the effectiveness of attacks based on generative models. Nonetheless, our BP-UAC, through a bi-path optimization strategy, overcomes the limitations of traditional attack approaches and effectively mitigates the impact of defense measures. By integrating gradient information from multiple models, the BP-UAC enhances the resilience of adversarial CAPTCHAs against a variety of defense mechanisms, while also alleviating the issue of model overfitting.

E. Influence and setting of Parameters

We also conducted a comprehensive ablation experiment to verify the reasonableness of the parameter settings. We calculated the Clip Score, average attack step and ASR of the adversarial examples generated with different learning rate and different perturb grad scale, as can be seen in the Fig. 7 and the second and third columns of Fig. 5, with the increase of the learning rate, the the average attack step required for the success of the attack decreases significantly, however, a higher learning rate η will cause the generative model to be

Fig. 6. Comparison plot of adversarial samples generated using detailed prompt versus using labels. It is easy to draw conclusion that: a more detailed prompt will result in a more accurate and higher quality generated image.

Fig. 7. Adversarial examples generated with different Learning rates.

over-guided in some steps during the generation process, thus decreasing the quality of the generated image, for this reason, we consider the quality of the image, the efficiency of the attack, and choose to set η to 1.0, and at the same time, for the size of the perturbation ϵ, in order to guarantee the method is effective, we must ensure the success rate of the attack, so in the case of ASR of 100%, we choose to set the perturb grad scale to the $3e^{-3}$ that requires the least number of attack steps and generates the highest Clip Score of the adversarial examples.

VI. CONCLUSION

In this paper, we propose a novel adversarial CAPTCHA generation framework, BP-UAC, which integrates LLM and generative models while introducing an innovative bi-path adversarial optimization strategy. By overcoming the limitations of traditional adversarial attack methods that add noise to original images, BP-UAC leverages gradients from multiple deep models and simultaneously guides the model toward generating adversarial examples in the directions of the second-highest and target class probability classes. This approach enables BP-UAC to identify more robust paths in different loss spaces, achieving exceptionally high ASR even against unknown models. Experimental results demonstrate that our method not only generates realistic images but also effectively deceives traditional DNN recognition models in various white-box and black-box application scenarios, providing a new direction for future adversarial attack research. Furthermore, given the widespread use of CAPTCHAs in daily life for identity verification and security protection, our method can effectively enhance the security of CAPTCHA systems, preventing unauthorized intrusions and providing essential technical support for CAPTCHA design and security upgrades. In addition, we plan to further explore how to generate higher-quality images in future work.

REFERENCES

[1] Y. Gong and C. Poellabauer, "Crafting adversarial examples for speech paralinguistics applications," in *Proceedings of 2018 DYnamic and Novel Advances in Machine Learning and Intelligent Cyber Security (DYNAMICS 2018) Workshop*, 2018.

[2] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini: Fooling deep structured prediction models," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, p. 6977–6987.

[3] M. Osadchy, J. Hernandez-Castro, S. Gibson, O. Dunkelman, and D. Pérez-Cabo, "No bot expects the deepcaptcha! introducing immutable adversarial examples, with applications to captcha generation," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2640–2653, 2017.

[4] S. Sivakorn, I. Polakis, and A. D. Keromytis, "I am robot: (deep) learning to break semantic image captchas," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016, pp. 388–403.

[5] B. Wallace, A. Gokul, and N. Naik, "Edict: Exact diffusion inversion via coupled transformations," 2022. [Online]. Available: https://arxiv.org/abs/2211.12446

[6] Y. Guo, Q. Hu, M. Cordy, M. Papadakis, and Y. L. Traon, "Muten: Boosting gradient-based adversarial attacks via mutant-based ensembles," 2021. [Online]. Available: https://arxiv.org/abs/2109.12838

[7] Y. Ma, X. Xu, L. Fang, and Z. Liu, "Gadt: Enhancing transferable adversarial attacks through gradient-guided adversarial data transformation," 2024. [Online]. Available: https://arxiv.org/abs/2410.18648

[8] Z. Yuan, J. Zhang, Y. Jia, C. Tan, T. Xue, and S. Shan, "Meta gradient adversarial attack," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 7728–7737.

[9] Y. Song, R. Shu, N. Kushman, and S. Ermon, "Constructing unrestricted adversarial examples with generative models," 2018. [Online]. Available: https://arxiv.org/abs/1805.07894

[10] X. Dai, K. Liang, and B. Xiao, "Advdiff: Generating unrestricted adversarial examples using diffusion models," 2024. [Online]. Available: https://arxiv.org/abs/2307.12499

[11] M. Kang, D. Song, and B. Li, "Diffattack: Evasion attacks against diffusion-based adversarial purification," 2024. [Online]. Available: https://arxiv.org/abs/2311.16124

[12] C. Shi, X. Xu, S. Ji, K. Bu, J. Chen, R. Beyah, and T. Wang, "Adversarial captchas," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6095–6108, 2022.

[13] Y. Wen, "Robust image-based captcha generation using adversarial attack," in *Conference on Intelligent Computing and Human-Computer Interaction*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:255801888

[14] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2017. [Online]. Available: https://arxiv.org/abs/1607.02533

[15] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," 2016. [Online]. Available: https://arxiv.org/abs/1511.04599

[16] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," 2019. [Online]. Available: https://arxiv.org/abs/1801.02610

[17] Y. Zhang, H. Gao, G. Pei, S. Kang, and X. Zhou, "Effect of adversarial examples on the robustness of captcha," in *2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2018, pp. 1–109.

[18] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," 2021. [Online]. Available: https://arxiv.org/abs/2102.09672

[19] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2022. [Online]. Available: https://arxiv.org/abs/2010.02502

[20] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2022. [Online]. Available: https://arxiv.org/abs/2112.10752

[21] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[22] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[23] "The claude 3 model family: Opus, sonnet, haiku." [Online]. Available: https://api.semanticscholar.org/CorpusID:268232499

[24] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021. [Online]. Available: https://arxiv.org/abs/2103.00020

[25] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," 2017. [Online]. Available: https://arxiv.org/abs/1611.05431

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014. [Online]. Available: https://arxiv.org/abs/1409.4842

[29] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2020. [Online]. Available: https://arxiv.org/abs/1905.11946

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015. [Online]. Available: https://arxiv.org/abs/1512.00567

[31] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, no. 2, 2012.

[32] J. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi, "Clipscore: A reference-free evaluation metric for image captioning," 2022. [Online]. Available: https://arxiv.org/abs/2104.08718

[33] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015. [Online]. Available: https://arxiv.org/abs/1412.6572

[34] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2017. [Online]. Available: https://arxiv.org/abs/1611.01236

[35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019. [Online]. Available: https://arxiv.org/abs/1706.06083

[36] D. Liu, X. Wang, C. Peng, N. Wang, R. Hu, and X. Gao, "Adv-diffusion: Imperceptible adversarial face identity attack via latent diffusion model," 2023. [Online]. Available: https://arxiv.org/abs/2312.11285

[37] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli, "A self-supervised approach for adversarial robustness," 2020. [Online]. Available: https://arxiv.org/abs/2006.04924

[38] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," 2019. [Online]. Available: https://arxiv.org/abs/1902.02918

[39] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," 2018. [Online]. Available: https://arxiv.org/abs/1711.01991

[40] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," 2018. [Online]. Available: https://arxiv.org/abs/1712.02976