DAIA: Dynamic Ahead Iterative Attack for Highly Transferable Adversarial Examples

Yongqing Zhao[1,2], Fuqiang Wang[3,1,2,*], Wenqing Guo[1,2]
[1]*Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China*
[2] *Shandong Provincial Key Laboratory of Industrial Network and Information System Security, Shandong Fundamental Research Center for Computer Science,Jinan, China*
[3]*Quan Cheng Laboratory, jinan, China*
[*]Corresponding Author: Fuqiang Wang, email:wangfq@sdas.org

***Abstract*—Adversarial example attacks are considered a serious threat to Deep Neural Network (DNN) models. While generating adversarial examples in a white-box setting has been extensively studied, creating transferable adversarial examples that can successfully attack black-box models remains challenging. This work proposes DAIA (Dynamic Ahead Iterative Attack), a two-stage transferable adversarial attack method based on momentum-adaptive optimization. In the early stage of the attack, DAIA introduces a delayed-start phase, performing pre-iterations with a relatively large step amplification factor to accumulate a stable momentum direction and suppress early gradient oscillations. During the formal iterative stage, DAIA adaptively adjusts the step size by computing the cosine similarity between consecutive momentum vectors, combined with an Exponential Moving Average (EMA) smoothing mechanism, to balance stability and exploration.Empirical evaluations on the ImageNet dataset demonstrate that DAIA achieves superior portability compared to existing input-transformation-based methods under both single-model and defense settings. Furthermore, DAIA can be seamlessly integrated with input-transformation approaches to further enhance portability, highlighting the flexibility and effectiveness of our method.**

***Index Terms*—Transfer-based attack, Adversarial transferability, Robustness, Model Security**

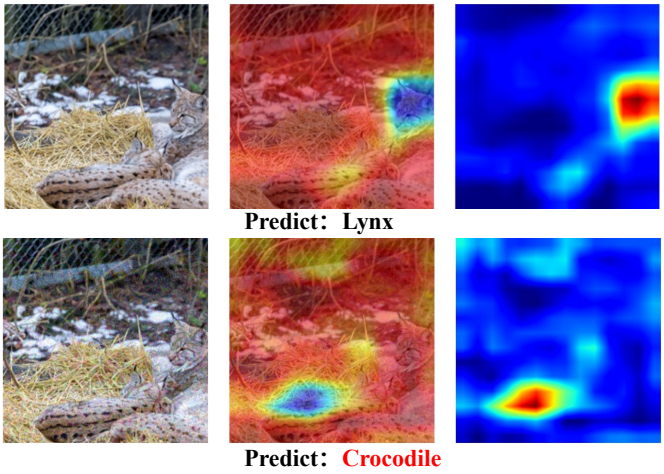


Fig. 1: Attention maps of a clean image and its adversarial example crafted by DAIA on RN-50. The target model is VGG-16.

I. Introduction

In recent years, Deep Neural Networks (DNNs) have achieved remarkable success in computer vision tasks such as autonomous driving [1] and medical diagnosis [2]. However, researchers have demonstrated that safety-critical applications powered by deep learning are highly vulnerable to adversarial attacks [3], raising serious concerns about model security. Adversarial examples can be crafted by adding carefully designed, imperceptible perturbations to the original inputs. Although these subtle perturbations are invisible to the human eye, they can cause models to make incorrect predictions [4]. Therefore, generating adversarial examples has become an effective approach to revealing model vulnerabilities and improving robustness. Based on the knowledge of model information, adversarial attacks are generally divided into white-box and black-box attacks. In a white-box attack, the attacker has full access to the victim model's information, including its architecture, parameters, weights, and the loss function used for training. Therefore, they can directly obtain the model's gradients to generate adversarial examples. In contrast, in a black-box attack, the attacker has no specific knowledge about the victim model. Considering the difficulty of obtaining model information in real-world applications, black-box attacks are of greater practical significance.

Black-box attacks can be categorized into query-based and transfer-based attacks according to their attack strategies. In query-based attacks, the attacker needs to access the target model multiple times to obtain information such as the predicted labels of samples, which greatly increases the risk of being detected. In contrast, transfer-based attacks can generate high-quality adversarial examples without querying the target model. The core idea is to craft adversarial examples on a white-box surrogate model and transfer them to the target black-box model for attack. Although existing studies have made notable progress in improving transferability, we observe that 1) the gradient direction in the early iterations of existing methods is often unstable, easily leading to local optima; 2)

the fixed step size used during updates fails to adapt to the dynamic nature of the optimization process. These issues result in inefficient perturbation optimization and limited cross-model transferability. To address the above issues, this paper proposes a novel Dynamic Ahead Iterative Attack (DAIA) method. An example is provided to show the misdirection capacity of DAIA in Fig. 1. DAIA adopts a two-phase dynamic optimization framework that enhances perturbation optimization through momentum stabilization modeling and adaptive step-size adjustment. In the delayed-start phase, DAIA performs pre-iterations guided by a step amplification factor to accumulate stable momentum directions and suppress early gradient oscillations. The perturbation is then reinitialized while retaining the stabilized momentum information, preventing early-stage noise from affecting subsequent optimization. In the dynamic step-size iteration phase, DAIA measures the stability of optimization by computing the cosine similarity between consecutive momentum vectors and applies an exponential moving average (EMA) for smoothing. The step size is dynamically adjusted according to momentum stability—increasing when the direction is stable to accelerate convergence, and decreasing when it fluctuates to maintain robustness—thus achieving a dynamic balance between exploration and stability during optimization.

Extensive experiments on the ImageNet dataset demonstrate that DAIA significantly improves the success rate of black-box attacks. Our main contributions are summarized as follows:

- We propose a transfer-based black-box attack from a new perspective: Dynamic Ahead Iterative Attack (DAIA). DAIA stabilizes directions via a delayed-start momentum pre-convergence, and dynamically adjusts the update magnitude in the formal iterations using a momentum-similarity-guided adaptive step size.
- We demonstrate the effectiveness of DAIA. Compared with existing transfer-based attack methods, DAIA significantly improves the transferability of adversarial examples.
- We show that DAIA can be easily integrated with existing transfer-based attacks to produce even better performance, indicating desirable flexibility.

II. Related Work

A. Gradient-Based Attack

Fast Gradient Sign Method (FGSM) [5] adds perturbations in the direction of the input gradient. Iterative FGSM (I-FGSM) [6] extends FGSM to an iterative version, demonstrating improved white-box attack performance but generating adversarial examples with poor transferability. The Momentum Iterative Fast Gradient Sign Method (MIFGSM) [7] incorporates a momentum term in each iteration, which facilitates rapid escape from local optima and the discovery of adversarial samples with higher transferability. The Nesterov Iterative Fast Gradient Sign Method (NIFGSM) [8] proposes utilizing Nesterov's accelerated gradient to effectively look ahead in the gradient direction during computation, thereby stabilizing the update direction to enhance transferability. IE-FGSM [9] employs a two-step gradient calculation based on the augmented Euler method to correct the gradient descent direction from a numerical analysis perspective. GNP [10] introduces a gradient norm penalty (GNP), which drives the loss function optimization process to converge to a flat region of local optima within the loss landscape. Global Momentum Initialization(GI) [11] analyzes the limitations of existing optimization methods from the perspective of gradient consistency and introduces global momentum, providing global momentum knowledge to mitigate gradient vanishing.

B. Input Transformation-Based Attack

Diverse Input Method (DIM) [12] enhances input images by combining random resizing and padding transformations, then uses the processed images to generate adversarial examples. Translation Invariant Method (TIM) [13] improves the transferability of adversarial examples by applying a convolution operation using a Gaussian kernel W to the gradient information. Scale invariant method (SIM) [14] calculates the gradient of several scaled images. Block Shuffle and Rotation (BSR) [15] divides the input image into several blocks. By randomly shuffling and rotating these blocks, it disrupts the image's inherent relationships and constructs a new set of images for gradient computation.Mixed-Frequency Inputs (MFI) [16]is a method based on a frequency domain perspective. By considering high-frequency components from various images during the gradient calculation process, MFI helps mitigate the overfitting of adversarial examples to the source model.

C. Adversarial Defenses

Researchers have been working to mitigate the threat of adversarial examples, such as incorporating them into model training to enhance robustness. Adversarial Training (AT) [17] is one of the most effective methods, where adversarial examples are incorporated during training to enhance model robustness. Input purification methods [18] [19] [20] aim to remove adversarial perturbations so as to restore adversarial examples to natural samples as much as possible. For instance, Liao et al. [18] designed a high-level representation-guided denoiser (HGD) based on U-Net to eliminate adversarial perturbations. Naseer et al. [21] proposed an adaptive training neural representation purifier (NRP) to purify input samples, demonstrating strong effectiveness against transfer-based adversarial examples.

III. Method

A. Preliminaries

Let x and y denote a clean image and its corresponding true label, respectively. Given a classification model $f(x) \rightarrow y$, where y is the output category for input x. Our goal is to generate an adversarial sample $x^{adv} = x + \delta$ such that the classifier misclassifies it, i.e.,$f(x^{adv}) \neq y$, while the generated adversarial sample remains visually indistinguishable from the

Algorithm 1 Framework of DAIA

Input: A classifier f with loss function $\ell(\cdot,\cdot)$, maximum perturbation ϵ, number of iterations T, input image x with label y, delayed-start iterations P, step amplification factor β, maximum step scale $\alpha_{\max}$, EMA coefficient γ, similarity factor k, minimum step scale $\alpha_{\min}$.
Output: Adversarial example x^{adv}.

$g_0 = 0$, $x_0^{adv} = x$, $\alpha = \epsilon/T$.
// Delayed Start Phase
for $t = 0$ to $P-1$ **do**
  $g_{t+1} = \mu g_t + \frac{\nabla_x \ell(f(x_t^{\text{adv}}),y)}{\|\nabla_x \ell(f(x_t^{\text{adv}}),y)\|_1}$
  $x_{t+1}^{adv} = \text{Clip}\big(x_t^{adv} + \beta \cdot \alpha \cdot \text{sign}(g_{t+1})\big)$
end for
Set $g_0 = g_P$, $s_0 = 1$.
// Dynamic Step Size Iteration
for $t = 0$ to $T-1$ **do**
  $g_{t+1} = \mu g_t + \frac{\nabla_x \ell(f(x_t^{\text{adv}}),y)}{\|\nabla_x \ell(f(x_t^{\text{adv}}),y)\|_1}$
  $\cos_{t+1} = \frac{\langle g_{t+1}, g_t \rangle}{\|g_{t+1}\|_2 \|g_t\|_2 + \varepsilon}$
  $s_{t+1} = \gamma s_t + (1-\gamma)\cos_{t+1}$
  $\text{scale} = \text{Clamp}\big(1 + k s_{t+1},\ \alpha_{\min},\ \alpha_{\max}\big)$
  $\alpha_{t+1} = \alpha \cdot \text{scale}$
  $x_{t+1}^{adv} = \text{Clip}\big(x_t^{adv} + \alpha_{t+1} \cdot \text{sign}(g_{t+1})\big)$
end for
$x^{adv} = x_T^{adv}$
return x^{adv}

original image. The generation of adversarial examples can be formulated as the following optimization problem:

$$\arg\max_{x^{adv}} \ell(x^{adv}, y) \quad \text{s.t.} \quad \|x^{adv} - x\|_p \le \epsilon \tag{1}$$

where the loss function $\ell(\cdot,\cdot)$ is typically the cross-entropy loss, ϵ denotes the perturbation budget and the L_p norm is used to measure the difference between x and x^{adv}. In this work, we adopt the most commonly used $p = \infty$ in the literature. However, in a black-box attack environment, the gradient of the loss function cannot be directly computed. To address this, we generate adversarial examples on a white-box model that can successfully attack other black-box target models. Therefore, we need to investigate how to enhance the transferability of adversarial examples—that is, their ability to successfully attack other black-box models.

B. DAIA

DAIA comprises two phases: the delayed initiation phase and the dynamic step phase. The former stabilizes the momentum direction, while the latter adaptively controls disturbance steps.

Delayed Start Phase.During the initial optimization phase, gradient directions fluctuate significantly. If formal iteration begins immediately, it may lead to momentum oscillations and local optima. Therefore, DAIA performs p pre-updates with "amplified step sizes" during the delayed initiation phase, specifically expressed as:

$$g_{t+1} = \mu g_t + \frac{\nabla_x \ell(f(x_t^{\text{adv}}), y)}{\|\nabla_x \ell(f(x_t^{\text{adv}}), y)\|_1} \tag{2}$$

$$x_{t+1}^{\text{adv}} = x_t^{\text{adv}} + \alpha\beta \cdot \text{sign}(g_t) \tag{3}$$

Where, β represents the step amplification factor during the delayed start phase. The objective at this stage is not to immediately generate effective perturbations, but rather to accumulate a more stable momentum direction.

Dynamic Step Size Iteration Phase. The formal iteration process executes a total of T steps. During this phase, DAIA no longer employs a fixed step size but dynamically adjusts it based on the stability of the momentum direction. First, we calculate the cosine similarity between two consecutive momentum vectors:

$$\cos(t+1) = \frac{\langle g_{t+1}, g_t \rangle}{\|g_{t+1}\|_2 \cdot \|g_t\|_2 + \varepsilon} \tag{4}$$

When similarity is high, it indicates directional stability, allowing for an appropriate increase in step size. Conversely, when similarity is low, the step size should be appropriately reduced to avoid overly aggressive updates.Additionally, to mitigate the impact of noise on similarity, exponential moving average (EMA) is applied to smooth the similarity values:

$$s_{t+1} = \gamma s_t + (1-\gamma)\cos(t+1) \tag{5}$$

where, s_{t+1} denotes the smoothed momentum similarity, and γ represents the smoothing coefficient.Next, dynamically compute the step size for this iteration based on the smoothed similarity:

$$\alpha_{t+1} = \alpha \cdot \text{Clamp}\left(1 + k s_{t+1}, \alpha_{\min}, \alpha_{\max}\right) \tag{6}$$

$$\text{Clamp}(v, a, b) = \min(b, \max(a, v)) \tag{7}$$

where, k is the momentum influence coefficient, which controls the strength of momentum similarity's effect on the step size. $\alpha_{\min}$is the lower bound of the step size, preventing excessive step size reduction that could cause optimization stagnation. $\alpha_{\max}$ is the upper bound of the step size, limiting the maximum amplification factor to ensure update stability. Then update the adversarial perturbations:

$$x_{t+1}^{adv} = \text{Clip}_{x,\epsilon}\left(x_t^{adv} + \alpha_t \cdot \text{sign}(g_t)\right) \tag{8}$$

The attack process of the proposed DAIA method can be summarized as Algorithm. 1.

IV. Experiment

A. Experimental Setup

Models. To validate the performance of our proposed method, we evaluated it on three categories of models: 1) **CNN-based models**, including four widely used architectures: ResNet-50 (RN-50) [22], VGG-16 [23], DenseNet-121 (DN-121) [24], and Inception-v3 (Inc-v3) [25]; 2) **Transformer-based models**, including ViTB [26], PiT [27], Visformer [28],

TABLE I: Attack success rates (%) for eight models under different attack methods. Attackers were constructed based on RN-50, VGG-16, DN-121, and Inc-v3. * denotes white-box attacks.

Model	Attack	RN-50	VGG-16	DN-121	Inc-v3	ViT	PiT	Visformer	Swin
RN-50	MIFGSM	99.9*	59.1	53.4	37.0	12.1	22.7	26.3	28.1
	NIFGSM	**100.0***	65.1	63.5	39.3	13.1	23.0	29.8	28.7
	IEFGSM	**100.0***	67.3	67.4	43.0	15.6	28.2	37.2	36.8
	GNP	**100.0***	68.3	69.7	46.2	15.7	29.7	38.1	38.1
	GIFGSM	**100.0***	73.5	67.5	50.1	16.1	**29.9**	39.0	37.1
	DAIA	**100.0***	**78.6**	**75.7**	**53.0**	**16.4**	**29.9**	**42.4**	**39.7**
VGG-16	MIFGSM	59.7	99.8*	78.3	55.9	15.8	29.6	44.5	42.5
	NIFGSM	60.7	99.9*	76.2	57.7	15.8	30.3	47.4	46.6
	IEFGSM	68.0	99.8*	84.3	63.0	17.8	33.9	52.9	49.4
	GNP	68.9	99.8*	84.2	62.6	17.6	**34.1**	**53.4**	50.7
	GIFGSM	66.3	99.9*	82.8	63.5	17.9	31.4	49.1	49.0
	DAIA	**73.3**	**100.0***	**84.6**	**66.2**	**18.0**	32.3	50.2	**51.9**
DN-121	MIFGSM	74.5	81.2	**100.0***	63.3	21.3	32.7	51.0	46.7
	NIFGSM	80.7	85.8	**100.0***	67.1	24.1	35.2	53.6	50.2
	IEFGSM	86.0	86.8	**100.0***	73.7	26.9	37.7	61.8	55.2
	GNP	85.7	87.8	99.9*	72.2	26.1	40.5	60.5	55.7
	GIFGSM	85.4	90.6	**100.0***	74.9	26.1	**40.9**	58.1	54.9
	DAIA	**87.8**	**92.6**	**100.0***	**76.8**	**27.4**	**40.9**	**61.9**	**56.0**
Inc-v3	MIFGSM	43.7	69.6	65.5	**100.0***	17.1	24.9	32.3	31.1
	NIFGSM	53.9	74.9	72.7	**100.0***	19.9	30.2	39.4	37.5
	IEFGSM	54.4	75.4	73.7	**100.0***	21.6	31.0	39.8	37.2
	GNP	58.0	77.9	76.2	**100.0***	21.2	32.3	43.1	41.1
	GIFGSM	57.2	78.9	73.9	**100.0***	20.7	31.5	41.7	39.2
	DAIA	**63.3**	**83.3**	**80.6**	**100.0***	**22.5**	**34.5**	**45.4**	**43.5**

and Swin [29]; 3) **Defense models**, where we employed two advanced defense methods: AT [17] and HGD [18].

Dataset. We conducted experiments using a subset of ImageNet. This subset dataset consists of 1000 images and was used in the NIPS 2017 adversarial competition [30]. These images were correctly classified and belong to distinct categories as our original inputs.

Baselines.To verify the effectiveness of DAIA, we choose five competitive gradient-based attacks as our baselines, i.e. MIFGSM [7], NIFGSM [8], IEFGSM [9], GNP [10] and GIFGSM [11]. Additionally, to validate that this method can seamlessly integrate with existing approaches to enhance attack success rates, we selected four transformation methods—DIM [12], SIM [14], TIM [13], and BSR [15]—to combine with our method for effectiveness analysis.

Hyper-parameters. We set the maximum perturbation of the parameter $\epsilon = 16$, the step size $\alpha = 1.6$. The total number of iterations for all methods is set to 15, and the decay factor μ is set to 1. For GNP, we set the step size r to 0.01 and the regularization coefficient to 0.8. GI-FGSM has pre-convergence it erations = 5 and global search factor = 10. DIM has its diversity probability p = 0.5. TIM utilizes a kernel size of 15 × 15. The number of copies of SIM is 5. BSR splits the image into 2 × 2 blocks and calculates the gradients on N = 20 transformed images. In DAIA, we introduce several hyperparameters to control the number of iterations in the delayed activation phase, the search intensity, and the step scaling intensity in the dynamic step size phase. Initial hyperparameter search on 1000 ImageNet validation images established default values of $P = 5$, $T = 10$, $\beta = 8$, $\gamma = 0.8$, $k = 1.0$, $\alpha_{\min} = 0.5$ and $\alpha_{\max} = 5$, which were subsequently fixed across all experiments.

B. Evaluation on Normally Trained Models

We first evaluated the performance of various gradient-based adversarial attacks. We generated adversarial examples on four standard training models and evaluated them on eight models, comprising four CNN-based models and four Transformer-based models. The Attack Success Rate (ASR) is summarized in Table. I., where the first column indicates the surrogate model, the second column lists the attack method, and the remaining columns represent the ASR of different classification models under attack.

At first glance, our proposed DAIA outperforms all baseline attack methods across all models. For instance, when the proxy model is RN-50 [22], the attack success rates of MIFGSM [7], NIFGSM [8], IEFGSM [9], GNP [10], and GIFGSM [11] against the target model VGG-16 [23] are 59.1%, 65.1%, 67.3%, 68.3%, and 73.5%, respectively. In contrast, our DAIA method achieves a significantly higher success rate of 78.6%, which is 5.1% higher than GIFGSM [11] and 10.3% higher than GNP [10].

C. Integrated with Transformation-based Attacks

To further demonstrate the effectiveness and flexibility of DAIA, we combined the DAIA method with four transform-based approaches, including DIM [12], SIM [14], TIM [13], and BSR [15]. We report the attack success rates of adversarial examples generated on RN-50 [22]in Table. II. Our proposed DAIA method significantly enhances the transferability of these input-transformation-based attacks. For instance, when

TABLE II: Attack success rate (%) when combining DAIA with different input transformation methods. Attacker based on RN-50. * denotes white-box attacks.↑ indicates the increase in attack success rate after combining with DAIA.

Attack	RN-50	VGG-16	DN-121	Inc-v3	ViT	PiT	Visformer	Swin
DIM	98.4*	69.4	71.6	57.4	25.6	40.3	50.3	46.2
DIM-DAIA	**99.6***$^{\uparrow 1.2}$	**87.1**$^{\uparrow 17.7}$	**88.3**$^{\uparrow 16.7}$	**77.8**$^{\uparrow 20.4}$	**35.3**$^{\uparrow 9.7}$	**51.8**$^{\uparrow 11.5}$	**66.8**$^{\uparrow 16.5}$	**60.4**$^{\uparrow 14.2}$
SIM	100*	69.5	73.0	51.9	22.0	35.9	47.4	43.6
SIM-DAIA	**100***$^{\uparrow 0.0}$	**86.4**$^{\uparrow 16.9}$	**90.1**$^{\uparrow 17.1}$	**71.4**$^{\uparrow 19.5}$	**28.1**$^{\uparrow 6.1}$	**44.9**$^{\uparrow 9.0}$	**61.6**$^{\uparrow 14.2}$	**56.2**$^{\uparrow 12.6}$
TIM	97.7*	57.5	53.3	40.0	10.7	15.4	22.6	19.4
TIM-DAIA	**99.7***$^{\uparrow 2.0}$	**81.0**$^{\uparrow 23.5}$	**78.5**$^{\uparrow 25.2}$	**58.5**$^{\uparrow 18.5}$	**12.8**$^{\uparrow 2.1}$	**18.8**$^{\uparrow 3.4}$	**30.2**$^{\uparrow 7.6}$	**24.2**$^{\uparrow 4.8}$
BSR	99.1*	96.6	97.4	90.3	54.6	80.1	89.2	84.6
BSR-DAIA	**99.9***$^{\uparrow 0.8}$	**99.3**$^{\uparrow 2.7}$	**99.3**$^{\uparrow 1.9}$	**94.6**$^{\uparrow 4.3}$	**56.0**$^{\uparrow 1.4}$	**82.6**$^{\uparrow 2.5}$	**91.1**$^{\uparrow 1.9}$	**84.7**$^{\uparrow 0.1}$

TABLE III: Attack success rates (%) of adversarial examples generated using various attack methods across eight classification models under AT.

Attack	RN-50	VGG-16	DN-121	Inc-v3	ViT	PiT	Visformer	Swin
MIFGSM	40.6	41.8	42.3	42.6	42.3	42.1	41.9	41.2
NIFGSM	41.0	41.9	42.7	42.0	42.6	41.9	41.8	41.3
IEFGSM	40.9	41.7	42.9	42.3	42.6	42.2	42.3	41.4
GNP	40.7	41.8	42.5	42.6	42.9	42.7	42.3	41.8
GIFGSM	41.4	**43.2**	43.6	42.7	43.9	42.4	42.8	41.9
DAIA	**41.4**	43.0	**43.8**	**42.8**	**44.0**	**42.8**	**42.9**	**42.2**

TABLE IV: Attack success rates (%) of adversarial examples generated using various attack methods across eight classification models under HGD.

Attack	RN-50	VGG-16	DN-121	Inc-v3	ViT	PiT	Visformer	Swin
MIFGSM	18.2	36.0	50.3	44.0	28.4	28.0	33.9	13.2
NIFGSM	19.6	35.4	53.4	51.6	29.9	29.2	36.4	13.3
IEFGSM	24.0	43.3	**62.3**	52.8	32.0	31.3	36.3	12.9
GNP	24.9	**43.7**	**62.3**	56.4	34.9	34.2	40.9	14.7
GIFGSM	26.2	37.5	59.7	52.2	37.6	35.8	42.3	15.3
DAIA	**26.3**	36.1	59.2	**56.8**	**38.7**	**36.8**	**43.1**	**16.0**

attacking VGG16 [23], DN-121 [24], and Inc-v3 [25]respectively, combining DAIA with DIM [12]increases the attack success rate by 17.7%, 16.7%, and 20.4%. This effectively demonstrates the efficacy and flexibility of our proposed DAIA method.

D. Evaluations on Defense Method

Although adversarial training models offer some defensive capabilities, to further validate the effectiveness of our proposed DAIA method, we evaluated it on two more challenging advanced defense models: AT [17] and HGD [18]. The evaluation results are shown in Table. III and Table. IV, respectively. Our DAIA maintains outstanding performance across all baselines. These results validate the robustness and effectiveness of our approach against diverse adversaries.

E. Ablation Study

To validate the independent contributions of each DAIA component to transfer performance, we conducted component ablation experiments on four white-box models (ResNet-50, VGG-16, DenseNet-121 and Inception-v3). We decomposed DAIA into two components: Delay and DynStep. We added each component individually to the baseline (MIFGSM) and compared them against both the baseline and the complete

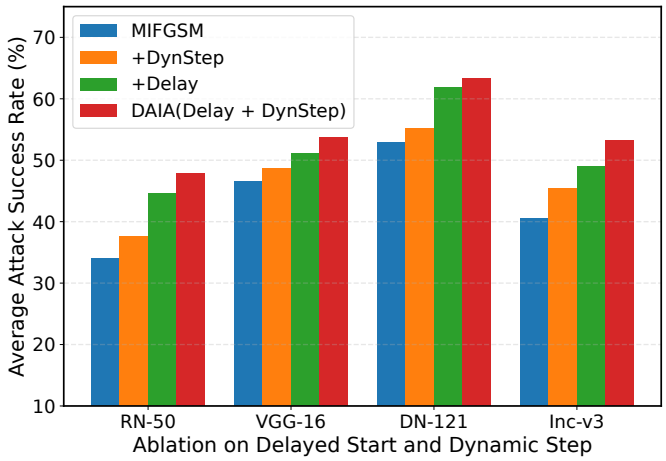
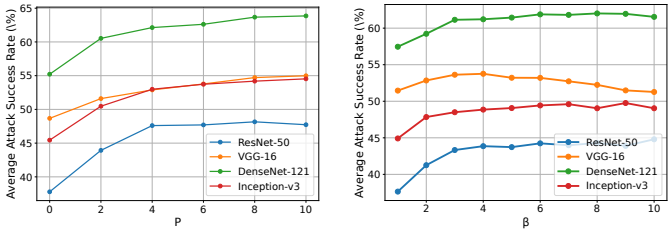


Fig. 2: Average attack success rate (%) of adversarial examples generated using MIFGSM, MI+Delay, MI+Dynstep, and DAIA on four models.



(a) Delayed start iterations P. (b) Step amplification factor β.

Fig. 3: Average attack success rate (%) under different hyperparameter values adversarial examples were generated on RN-50.

DAIA. As shown in Fig. 2, experimental results demonstrate that both components independently enhance transferability, confirming their positive effects. Additionally, we observe that the Delay component slightly outperforms DynStep. Based on this finding, we focused our parameter ablation analysis on two core hyperparameters p and β during the Delay phase to further explore their mechanisms.

Analysis of p, as shown in Fig. 3a, reveals that when p=0 (equivalent to disabling delayed activation entirely), all

models exhibit the lowest transfer success rates. As p increases, all four models demonstrate a clear upward trend, further validating the effectiveness of our delayed activation component. Performance stabilizes when p is increased to 4–6, with all four models achieving optimal results within this range. Ultimately, to achieve the optimal balance between performance and efficiency, we set p to 5. Analysis of β is shown in Fig. 3b. Experiments indicate that as β increases, transfer performance first improves rapidly and then gradually stabilizes. However, when β exceeds 8, it causes excessive deviation in the momentum direction, slightly impairing subsequent iterations. Therefore, a moderate β is most conducive to establishing a stable and high-quality initial momentum direction. Considering overall performance, we set β to 8.

V. Conclusion

This paper proposes a simple and efficient dynamic advance iterative attack method, DAIA. By dynamically predicting the gradient iteration process, it adaptively adjusts the update direction, effectively mitigating gradient drift and overfitting issues in the substitute model, thereby significantly improving the black-box transfer performance of adversarial samples. Extensive experimental results demonstrate the effectiveness and flexibility of our method.

ACKNOWLEDGMENT

This work was supported by The National Key R&D Program of China under Grant No. 2023YFC3306304. Qilu University of Technology (Shandong Academy of Sciences) Major Project under Grant No. 2025ZDZX02. Major project of Quan Cheng Laboratory under Grant No. QCL20250105.

References

[1] L. K. Sahoo and V. Varadarajan, "Deep learning for autonomous driving systems: technological innovations, strategic implementations, and business implications-a comprehensive review," *Complex Engineering Systems*, vol. 5, no. 1, pp. N–A, 2025.

[2] D. C. Lepcha, B. Goyal, A. Dogra, A. Alkhayyat, P. K. Sahu, A. Ali, and V. Kukreja, "Deep learning in medical image analysis: A comprehensive review of algorithms, trends, applications, and challenges," *Computer Modeling in Engineering & Sciences*, vol. 145, no. 2, p. 1487, 2025.

[3] H. Chen and M. A. Babar, "Security for machine learning-based software systems: A survey of threats, practices, and challenges," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–38, 2024.

[4] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. Ieee, 2017, pp. 39–57.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[6] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[7] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[8] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," *arXiv preprint arXiv:1908.06281*, 2019.

[9] A. Peng, Z. Lin, H. Zeng, W. Yu, and X. Kang, "Boosting transferability of adversarial example via an enhanced euler's method," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[10] T. Wu, T. Luo, and D. C. Wunsch, "Gnp attack: Transferable adversarial examples via gradient norm penalty," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 3110–3114.

[11] J. Wang, Z. Chen, K. Jiang, D. Yang, L. Hong, P. Guo, H. Guo, and W. Zhang, "Boosting the transferability of adversarial attacks with global momentum initialization," *Expert Systems with Applications*, vol. 255, p. 124757, 2024.

[12] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2730–2739.

[13] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4312–4321.

[14] J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, "Nesterov accelerated gradient and scale invariance for adversarial attacks," *arXiv preprint arXiv:1908.06281*, 2019.

[15] K. Wang, X. He, W. Wang, and X. Wang, "Boosting adversarial transferability by block shuffle and rotation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 24 336–24 346.

[16] Y. Qian, K. Chen, B. Wang, Z. Gu, S. Ji, W. Wang, and Y. Zhang, "Enhancing transferability of adversarial examples through mixed-frequency inputs," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 7633–7645, 2024.

[17] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in neural information processing systems*, vol. 32, 2019.

[18] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1778–1787.

[19] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli, "A self-supervised approach for adversarial robustness," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 262–271.

[20] W. Nie, B. Guo, Y. Huang, C. Xiao, A. Vahdat, and A. Anandkumar, "Diffusion models for adversarial purification," *arXiv preprint arXiv:2205.07460*, 2022.

[21] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli, "A self-supervised approach for adversarial robustness," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 262–271.

[22] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[26] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[27] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 11 936–11 945.

[28] Z. Chen, L. Xie, J. Niu, X. Liu, L. Wei, and Q. Tian, "Visformer: The vision-friendly transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 589–598.

[29] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[30] Nips 2017 adversarial attacks and defenses competition. [Online]. Available: https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition/dataset