STBA: Towards Evaluating the Robustness of DNNs for Query-Limited Black-box Scenario

Renyang Liu, *Graduate Student Member, IEEE,* Kwok-Yan Lam, *Senior Member, IEEE,* Wei Zhou*, *Member, IEEE,* Sixing Wu, Jun Zhao, *Member, IEEE,* Dongting Hu, Mingming Gong*, *Member, IEEE,*

***Abstract*—Extensive studies have revealed that deep neural networks (DNNs) are vulnerable to adversarial attacks, especially black-box ones, which can heavily threaten the DNNs deployed in the real world. Many attack techniques have been proposed to explore the vulnerability of DNNs and further help to improve their robustness. Despite the significant progress made recently, existing black-box attack methods still suffer from unsatisfactory performance due to the vast number of queries needed to optimize desired perturbations. Besides, the other critical challenge is that adversarial examples built in a noise-adding manner are abnormal and struggle to successfully attack robust models, whose robustness is enhanced by adversarial training against small perturbations. There is no doubt that these two issues mentioned above will significantly increase the risk of exposure and result in a failure to dig deeply into the vulnerability of DNNs. Hence, it is necessary to evaluate DNNs' fragility sufficiently under query-limited settings in a non-additional way. In this paper, we propose the Spatial Transform Black-box Attack (STBA), a novel framework to craft formidable adversarial examples in the query-limited scenario. Specifically, STBA introduces a flow field to the high-frequency part of clean images to generate adversarial examples and adopts the following two processes to enhance their naturalness and significantly improve the query efficiency: a) we apply an estimated flow field to the high-frequency part of clean images to generate adversarial examples instead of introducing external noise to the benign image, and b) we leverage an efficient gradient estimation method based on a batch of samples to optimize such an ideal flow field under query-limited settings. Compared to existing score-based black-box baselines, extensive experiments indicated that STBA could effectively improve the imperceptibility of the adversarial examples and remarkably boost the attack success rate under query-limited settings.**

***Index Terms*—Model Vulnerability, Model Robustness, Adversarial Attack, Query-Limited Black-Box Attack, Spatial Transform.**

This work is supported in part by the National Natural Science Foundation of China under Grant 62162067 and 62101480, and in part by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative, ABC Pte Ltd and XYZ association.

R. Liu is with the School of Information Science and Engineering, Yunnan University, Kunming 650500, China (e-mail: ryliu@nus.edu.sg)

K.Y. Lam and J. Zhao are with the College of Computing and Data Science, Nanyang Technological University, Singapore, 639798 (e-mail: {kwokyan.lam,junzhao}@ntu.edu.sg).

W. Zhou and S. Wu are with the School of Software and the Engineering Research Center of Cyberspace, Yunnan University, Kunming 650500, China (e-mail: {zwei, wusixing}@ynu.edu.cn).

D. hu and M. Gong are with Melbourne Centre for Data Science, School of Mathematics and Statistics, University of Melbourne, Parkville, VIC 3010, Australia (e-mail: dongting@student.unimelb.edu.au; Mingming.gong@unimelb.edu.au).

*Corresponding authors: Wei Zhou, Mingming Gong.

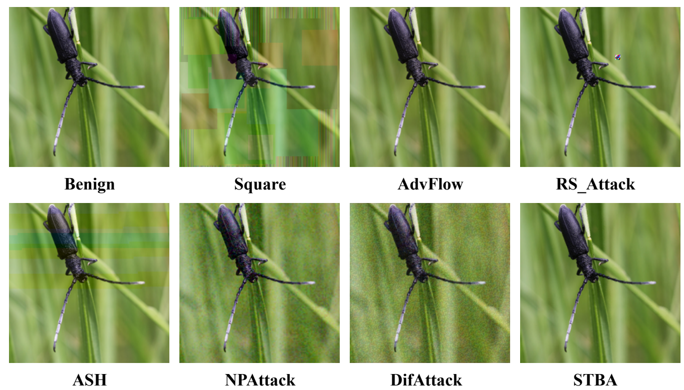


Fig. 1. The clean example and their corresponding adversarial example generated by baselines and the proposed STBA. The first one is the clean images, and the flowings are the adversarial example generated by Square Attack [1], AdvFlow [2], RS_Attack [3], ASH [4], NPAttack [5], and DifAttack [6], respectively.

I. Introduction

DEEP neural networks (DNNs) are susceptible to adversarial examples [7]–[9], which are crafted by subtly perturbing a clean input [10]–[13], especially for computer vision (CV) [14] and Natural Language Processing (NLP) tasks, such as image classification, face recognition [15], and language translation [16]. Except for the threat brought by such attacks, adversarial examples are the most effective way to evaluate existing models' vulnerability and can further impel the defender to improve the robustness of existing models and guide the design of new safety models. The critical point in carrying out adversarial attacks on CV models is how to generate adversarial examples with high imperceptibility and attack success rate. Various methods have been proposed to build adversarial examples; among them, the black-box attacks are more practical because they do not need to access the target model and only require the prediction results [17]–[20], which makes them are more suitable to destroy the DNN models deployed in the physical world.

Although most existing black-box attacks can obtain a high success rate by adding noise to the original image, they are not ideal in terms of naturalness since the added perturbations are not harmonious with the original clean image [21]. Meanwhile, most existing black-box attacks are query and optimization-based, which regard the process of calculating adversarial examples (or adversarial noise) as an optimization problem. They iteratively optimize the intermediate adversarial example with the query results until it can attack the target model successfully [22]. Those methods require a vast number of queries to ensure that most crafted adversarial examples for a dataset can be used to attack victim models successfully, which results in a heavy computation and time burden; worse, a mass of queries can be easily detected by defenders. Finally, most deployed DNN models are enhanced by various defense mechanisms; among them, adversarial training, which finetunes the DNN model on the pre-collected adversarial examples, is verified as the most effective way to improve the model's robustness against unknown attacks. Since such adversarial examples are often locally generated with noise-adding attacks, existing black-box attacks are more challenging to attack these robust models successfully.

To address the issues mentioned above, researchers have proposed various works to solve them separately. For improving the imperceptibility of adversarial examples [23], [24], some methods try to generate imperceptible adversarial examples, such as AdvFlow [2], which crafts adversarial examples by disturbing the latent space of image under L_p-norm constrained. Even though this method makes the adversarial perturbations more imperceptible to human eyes under black-box settings, it still impacts the whole clean image. For the challenge of a large number of queries required by optimized-based methods, some query-limited or query-efficient black-box attack methods, like evolutionary-based methods $\mathcal{N}$attack [25] and gradient estimation-based AdvFlow [2], declare that they can build suitable adversarial examples only in thousands or hundreds of queries. However, existing methods still need a large query budget to pursue a higher attack success rate. The empirical maximum query budget is usually 10,000 $\sim$ 100,000 [26], [27], which is still intolerable to online deployed DNN systems. As a consequence, a large number of queries yield an increased query cost and the probability of exposure.

To better evaluate the vulnerability and robustness of existing DNN models, including the robust ones, and improve the concealment of adversarial examples and the query efficiency of black-box attacks under the query-limited scenario, we formulate the issue of synthesizing adversarial examples beyond additive perturbations and propose a novel non-addition black-box attack method called **STBA**. More specifically, STBA uses spatial transformation techniques to generate adversarial examples with better concealment rather than directly adding well-designed noise to the benign image. The spatial transform technique can generate an eligible adversarial example by leveraging a well-optimized smooth flow field matrix $\boldsymbol{f}$ to move each pixel to a new location. Besides, to further enhance the imperceptibility of the generated adversarial noise, we first split the high- and low-frequency parts from clean images and then only apply the will-estimated flow field $\boldsymbol{f}$ to the high-frequency part to formulate the adversarial high-frequency counterpart, which will be combined to the original low-frequency part to synthesis the final adversarial examples. Furthermore, to address the query challenge of optimization-based black-box attacks, STBA leverages the scarce information, query accounts, attack success rate, and the risk of exposure in the query-limited situation, employed with gradient estimation to optimize such a flow field, which can effectively find the global optima with higher probability but require less information from the target model and straightforward the generation process. Therefore, the proposed STBA can achieve a higher attack success rate with a harsher query budget.

We draw the adversarial examples of different methods in Fig. 1, where the victim model is ResNet-152 [28] and the image is from the ImageNet [29] dataset. The visual results show that our proposed method alters the clean image slightly, thus ensuring the invisibility of the adversarial perturbations. Furthermore, we conducted extensive experiments on four different CV benchmark datasets, the statistical results indicate that the STBA can obtain the highest attack success rate with the max query budget limited as $Q_{max} = 1000$ or $Q_{max} = 10000$. Specifically, STBA can achieve a nearly 100% attack success rate on the normally trained models when $Q_{max} = 1000$ and on the robust trained models when $Q_{max} = 10000$ in most cases, while the baselines are lower than 90% and 70% in the most cases, respectively. Furthermore, we found that STBA can achieve the highest attack success rate with fewer average queries compared to the existing query-limited black-box methods on the target victim models while outperforming them concerning attack success rate. The main contributions could be summarized as follows:

- We propose a new black-box attack approach called STBA, which introduces the spatial transform techniques instead of noising-adding to black-box settings to address the low attack success rate and query efficiency of existing black-box methods on normally and robustly trained DNNs in query-limited scenarios.
- To further enhance the imperceptibility of adversarial examples, we integrate Gaussian blur techniques and dynamical strategy into our attack framework. Specifically, we first decomposed the image into high-frequency and low-frequency components. We then only applied the dynamically constrained flow field $\boldsymbol{f}$ to the high-frequency part to improve the generated adversarial images' visibility and image quality.
- Comparing with the existing score-based black-box methods, experimental results on four benchmark datasets show STBA's superiority in synthesizing adversarial examples with efficient queries, attack performance, better naturalness, and remarkable transferability.

The rest of this paper is organized as follows. We briefly review the methods relating to adversarial attacks in Sec. II. In Sec. III, we provide the details of the proposed STBA framework. The experiments are presented in Sec. IV, with the conclusion and limitation drawn in Sec. V.

II. Related Work

This section briefly reviews the most relevant attack methods to the proposed work, including spatial transform-based attacks and Query-Efficient black-box attacks.

A. Spatial transform-based attacks

The concept of spatial transformation is first mentioned in [30], which indicates that conventional neural networks are not robust to rotation, translation, and dilation. Next, [31] utilized the spatial transform techniques and proposed the stAdv method to craft adversarial examples with a high fooling rate and perceptually realistic beyond noise-adding way. They change each pixel position in the clean image by applying a well-designed flow field matrix $\boldsymbol{f}$ to the clean image to formulate the adversarial counterpart. Later, [32] proposed a new method to produce the universal adversarial examples by combining the spatial transform and pixel-level distortion, and it successfully increased the attack success rate against universal perturbation to more than 90%. In the literature [33], the authors applied spatial transform to the YUV space to generate adversarial examples with higher superiority in image quality. Also, Liu *et al*. [34] utilized spatial transformations by applying the flow field to the latent representations from the normalized flow model [35] to guarantee that generated adversarial examples are more perceptually similar to the original image.

B. Query-Efficient black-box attack

Although most optimized-based black-box attacks have achieved a high attack success rate, even reaching 100% in some cases, the tremendous need for queries to optimize adversarial examples is still a big challenge. Empirically, the average number of queries required to obtain a suitable adversarial example is from thousands to hundreds of thousands [27], [36], which is obviously not feasible in the physical world and can be detected by the target DNN models easily. Therefore, recently, researchers have proposed various methods to generate adversarial examples under query-limited queries or in a query-efficient way [37], [38].

The first work to limit the number of queries in black-box attacks is the Bayesian Optimization-Based Black-box Attack [39]. It uses the Bayesian optimization strategy to reduce the average number of queries to 1/10 of the random perturbation strategies. Then, Ilyas *et al*. proposed NES [36] to improve the query efficiency while maintaining a higher success rate in query and information-limited settings. In [40], the authors propose a query-efficient boundary-based attack (QEBA) that only relies on the predicted hard labels and presents an optimal analysis of gradient estimation based on dimensional reduction and gets good results. AdvFlow [2] improves the attack performance with a limited query budget by disturbing the image's latent representation obtained from pre-trained Normalize Flow. Square Attack [1] uses the random searched square-like adversarial perturbation to extend attacks. Accelerated Sign Hunter (ASH) [4] crafts the adversarial example by applying a Branch Prune Strategy that infers the unknown sign bits to avoid unnecessary queries and further improve the query efficiency within a limited query budget.

Besides, some researchers have proposed combining the transferability of adversarial attacks to improve query efficiency, which is dubbed query- and transfer-based attacks. F. Suya *et al*. [41] combined transfer-based attack [42] and optimize-based black-box and proposed Hybrid Batch Attacks (HBA). The HBA can use the information of the existing local models to generate candidate adversarial seeds and use these seeds to query the target model and optimize the generated images iteratively. Its seed-first strategy saves about 70% query times for an attack. MCG-Attack [43] improves the query efficiency by fully utilizing both example-level and model-level transferability. DifAttack [6] disentangles image latent into adversarial and visual features, trains an autoencoder with clean images and their adversarial examples of surrogate model(s), and iteratively optimizes adversarial features based on victim model feedback to create successful AEs while maintaining visual consistency.

However, these transfer-based attacks heavily rely on the similarity between the substitute model and the oracle model. In this paper, we only focus on score-based attacks, which are independent of local surrogate models. Therefore, adversarial attacks, especially for the Query-Efficient black-box attacks, pose the request for a method that is efficient and effective in building adversarial examples with high imperceptibility and quality in performing attacks on unknown DNN models within limited queries. On the other hand, with the development of defense mechanisms, higher requirements are placed on the invisibility of adversarial examples. To achieve these goals, we learn from the previous studies that adversarial examples can be gained in a beyond-noise-adding way. Hence, we are well motivated to develop a novel method to disturb the original image by applying a well-calculated flow field $\boldsymbol{f}$ to clean images to generate adversarial ones, which makes it possible to build qualified adversarial examples with high invisibility in dozens or hundreds of queries.

III. Methodology

A. Problem Definition

Given a well-trained DNN classifier $\boldsymbol{\mathcal{C}}$ and an input $\boldsymbol{x}$ with its corresponding label y, we have $\boldsymbol{\mathcal{C}}(\boldsymbol{x}) = y$. The adversarial example $\boldsymbol{x}^{adv}$ is a neighbor of $\boldsymbol{x}$ and satisfies that $\boldsymbol{\mathcal{C}}(\boldsymbol{x}^{adv}) \neq y$ and $\left\|\boldsymbol{x}^{adv} - \boldsymbol{x}\right\|_p \leq \epsilon$, where the $\boldsymbol{L}_p$-norm is used as the distance measure and ϵ is usually a small noise budget. With this definition, the problem of finding an adversarial example becomes a constrained optimization problem:

$$\boldsymbol{x}^{adv} = \underset{\left\|\boldsymbol{x}^{adv}-\boldsymbol{x}\right\|_p \leq \epsilon}{argmax}\, \mathcal{L}\left(\boldsymbol{\mathcal{C}}(\boldsymbol{x}^{adv}) \neq y\right), \tag{1}$$

where $\mathcal{L}$ stands for a loss function that measures the confidence of the model outputs.

Previous works craft an adversarial example $\boldsymbol{x}^{adv}$ by adding $\boldsymbol{L}_p$-norm constrained noise to the clean image $\boldsymbol{x}$, i.e.,

$$\boldsymbol{x}^{adv} = \boldsymbol{x} + noise,\ s.t.\ \left\|noise\right\|_p \leq \epsilon, \tag{2}$$

Different from this, we introduce the spatial transform techniques to the high-frequency part of the clean image to build the adversarial example $\boldsymbol{x}^{adv}$. As illustrated in Fig. 2, the proposed Spatial Transform Black-box Attack framework can be divided into flowing stages: The first stage is decomposing the different frequency information, i.e., the high-frequency part $\boldsymbol{x}_{high}$ and the low-frequency part x_{low}, from the original clean image $\boldsymbol{x}$ with the help of Gaussian Blur (we set the kernel size as 3x3). Second, we random sampling a mini-batch candidate flow fields $\boldsymbol{f}_i, i \in \{1, 2, 3..., n\}$ from an isometric normal distribution $\mathcal{N}$, which will be applied to the high-frequency part $\boldsymbol{x}_{high}$, and then combined with the low-frequency part

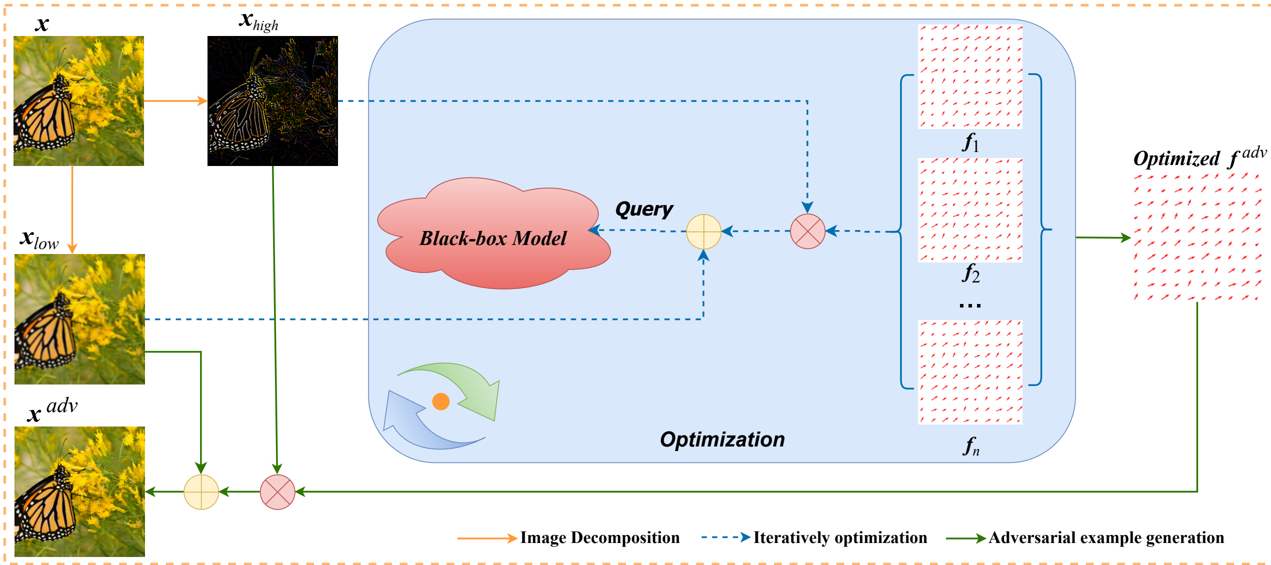


Fig. 2. Overview of Spatial Black-box Transform Attack (STBA). The **Black-box Model** is the victim model. The $\boldsymbol{x}$ and $\boldsymbol{x}^{adv}$ are the benign image and the corresponding adversarial counterpart, respectively. The $\boldsymbol{x}_{high}$ and $\boldsymbol{x}_{low}$ represent the high-frequency part and the low-frequency part of the benign image. The $\otimes$ represents the spatial transformation operation, and $\oplus$ means element addition. **The $\boldsymbol{f}_i, i \in \{1, 2, ..., n\}$ are the mini-batch candidate flow field sampled from an isometric normal distribution** $\mathcal{N}$, and the $\boldsymbol{f}^{adv}$ is the final optimized flow field that applies to the original clean image to formulate the final adversarial image $\boldsymbol{x}^{adv}$.

$\boldsymbol{x}_{low}$ to build the candidate adversarial examples used to query the black-box victim model to estimate the gradient in an iterative manner. Once the gradient is estimated, it can used to guide the optimization process to update the flow field $\boldsymbol{f}^{adv}$. Finally, we can obtain the adversarial example $\boldsymbol{x}^{adv}$ by applying the calculated flow field $\boldsymbol{f}^{adv}$ to the high-frequency part $\boldsymbol{x}_{high}$.

B. Adversarial Example Generation

We utilize the spatial transform method to build adversarial examples in a non-noise additional way. The spatial transform techniques using a flow field matrix $\boldsymbol{f} \in [0,1]^{2*h*w}$ to transform the original image $\boldsymbol{x}$ to $\boldsymbol{x}_{st}$ [31], where h and w are the height and width of the image. In particular, for our proposed STBA, we only need to apply such transform operation to the high-frequency part, i.e., $\boldsymbol{x}_{high}$. Specifically, assume the input is $\boldsymbol{x}_{high}$ and its transformed counterpart $\boldsymbol{x}^{st}_{high}$, for the i-th pixel in $\boldsymbol{x}^{st}_{high}$ at the pixel location (u^i_{st}, v^i_{st}), we need to calculate each element of the flow field as $\boldsymbol{f}_i = (\Delta u^i, \Delta v^i)$. So, the i-th pixel $\boldsymbol{x}^i_{high}$'s location in the transformed image can be indicated as:

$$(u^i, v^i) = (u^i_{st} + \Delta u^i, v^i_{st} + \Delta v^i). \tag{3}$$

To ensure the flow field $\boldsymbol{f}$ is differentiable, the bi-linear interpolation [30] is used to obtain the 4 neighboring pixels' value surrounding the location $(u^i_{st} + \Delta u^i, v^i_{st} + \Delta v^i)$ for the transformed image $\boldsymbol{x}^{st}_{high}$ as:

$${\boldsymbol{x}^{st}_{high}}^i = \sum_{q \in \boldsymbol{N}(u^i, v^i)} \boldsymbol{x}^q_{high}(1 - |u^i - u^q|)(1 - |v^i - v^q|), \tag{4}$$

where $\boldsymbol{N}(u^i, v^i)$ is the neighborhood, that is, the four positions (top-left, top-right, bottom-left, bottom-right) tightly surrounding the target pixel (u^i, v^i). For each pixel ${\boldsymbol{x}^{st}_{high}}^i$ in the adversarial high-frequency part $\boldsymbol{x}_{high}$, we can calculate it with with Eq. 4 to formulate the $\boldsymbol{x}^{st}_{high}$, i.e., the final adversarial high-frequency part $\boldsymbol{x}^{adv}_{high}$. Once the $\boldsymbol{f}$ for formulating $\boldsymbol{x}^{adv}_{high}$ has been computed, we can obtain the final $\boldsymbol{x}^{adv}$ by adding such adversarial high-frequency part $\boldsymbol{x}^{adv}_{high}$ with low-frequency part $\boldsymbol{x}_{low}$ of the clean image, which is given by:

$$\boldsymbol{x}^{adv} = \boldsymbol{x}^{adv}_{high} \oplus \boldsymbol{x}_{low}. \tag{5}$$

In practice, we regard the problem of calculating flow field $\boldsymbol{f}$ as an optimization task. Under the black-box settings, the only information that we can utilize is the output of the classifier $\boldsymbol{\mathcal{C}}(\cdot)$ regarding a specific input. To this end, following the previous works [2], [25], in this paper, we use gradient estimation to find the optimal flow $\boldsymbol{f}$ in an iterative manner.

C. Objective Functions

Taking the attack success rate and visual invisibility of the generated adversarial examples into account, we divide the objective function into two parts, where one is the adversarial loss $\mathcal{L}_{adv}$ and the other is a constraint for the flow field, i.e., flow loss $\mathcal{L}_{\boldsymbol{f}}$.

For adversarial attacks, the goal is making $\boldsymbol{\mathcal{C}}(\boldsymbol{x}^{adv}) \neq y$. We give the objective function as follows:

$$\mathcal{L}_{adv}(\boldsymbol{x}^{adv}, y, \boldsymbol{f}) = max[\boldsymbol{\mathcal{C}}(\boldsymbol{x}^{adv})_y - \underset{k \neq y}{max} \boldsymbol{\mathcal{C}}(\boldsymbol{x}^{adv})_k, k]. \tag{6}$$

To guarantee the visual imperceptibility of the generated adversarial examples, we extend the flow loss defined in [31] as $\mathcal{L}_f$ to constrain the distance of any two adjacent pixels, which compute the sum of spatial movement as:

$$\mathcal{L}_{\boldsymbol{f}}(\boldsymbol{f}) = \sum_p^A \sum_{q \in \mathcal{N}(p)} \sqrt{\left\|\triangle u^{(p)} - \triangle u^{(q)}\right\|_2^2 + \left\|\triangle v^{(p)} - \triangle v^{(q)}\right\|_2^2}, \tag{7}$$

where A represents all the pixels in the victim image.

In addition, we introduce a dynamic adjustment strategy for updating the flow field $\boldsymbol{f}$ flexible to further improve the generated adversarial examples' invisibility. More specifically, we initialize the flow budget ξ_{t_0} with a small value (such as $\xi_{t_0} = 10^{-1}$), then we update ξ with T steps, that is:

$$\xi_t = \xi_{t-1} + \alpha,\ \alpha = (\xi_{max} - \xi_{t_0})/T, \\ s.t.\ t \in \{1, 2, 3, ..., T\}, \quad (8)$$

where $T = Q_{max}/n_{sample}$, and the value of $\boldsymbol{f}$ matrices of each iterations can be obtained with a $clip(\cdot)$ operation, i.e., $\boldsymbol{f} = clip(\boldsymbol{f}, -\xi, +\xi)$.

The whole objective function can be written as:

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda * \mathcal{L}_{\boldsymbol{f}},\ s.t.\ \boldsymbol{f} = clip(\boldsymbol{f}, -\xi, +\xi), \quad (9)$$

where λ is a hyper-parameter to balance each loss item, in this paper, we set it as $\lambda = 5$ (For the reason why we set $\lambda = 5$, please refer to Sec. IV-F).

D. Optimization

Recall that our goal is to calculate the flow field $\boldsymbol{f}$, which will be applied to the clean image's high-frequency part $\boldsymbol{x}_{high}$ to generate its adversarial counterpart one $\boldsymbol{x}_{high}^{adv}$ and further formulate adversarial example $\boldsymbol{x}^{adv}$. we regard the flow field $\boldsymbol{f}$ comes from a unique distribution and re-defined the expected adversarial example $\boldsymbol{x}^{adv}$ by re-written the Eq.4 and Eq.5 as

$$\boldsymbol{x}^{adv} = \boldsymbol{x}_{high} \otimes \boldsymbol{f} \oplus \boldsymbol{x}_{low}\ s.t.\ \boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \sigma^2 \boldsymbol{I}), \quad (10)$$

where $\mathcal{N}$ is an isometric normal distribution with mean $\boldsymbol{\mu}$ and standard deviation σ, $\otimes$ stands for the spatial transform operation. As in AdvFlow [2], we flowing the *lazy statistician* [44] and re-write our attack definition in Eq. 9 as

$$\begin{aligned} J(\boldsymbol{\mu}, \sigma) &= \mathbb{E}_{p(\boldsymbol{x}^{adv}|\boldsymbol{\mu}, \sigma)}[\mathcal{L}(\boldsymbol{x}^{adv})] \\ &= \mathbb{E}_{\mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \sigma^2\boldsymbol{I})}[\mathcal{L}(\boldsymbol{x}_{high} \otimes \boldsymbol{f} \oplus \boldsymbol{x}_{low})], \end{aligned} \quad (11)$$

Further, we only required to minimize $J(\boldsymbol{\mu}, \sigma)$ w.r.t $\boldsymbol{\mu}$ [2], [25]. Likely, we derive the Jacobian of $J(\boldsymbol{\mu}, \sigma)$ as

$$\nabla J(\boldsymbol{\mu}, \sigma) = \mathbb{E}_{p(\boldsymbol{x}^{adv}|\boldsymbol{\mu}, \sigma^2 \boldsymbol{I})} \\ [\mathcal{L}(\boldsymbol{x}_{high} \otimes \boldsymbol{f} \oplus \boldsymbol{x}_{low}) \nabla_{\boldsymbol{\mu}} log \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \sigma^2 \boldsymbol{I})], \quad (12)$$

The value of expectation can be obtained by sampling from a distribution $\mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$ and the $\boldsymbol{\mu}$ can be optimized by executing a gradient descent step

$$\boldsymbol{\mu} \longleftarrow \boldsymbol{\mu} - lr \cdot \nabla_{\boldsymbol{\mu}} \boldsymbol{J}(\boldsymbol{\mu}, \sigma), \quad (13)$$

where the lr is the learning rate. We find the optimal flow field $\boldsymbol{f}$, which will lead the clean example $\boldsymbol{x}$ change to adversarial, by iteratively searching the optimization direction on a mini-batch-wised random sampled neighbor of $\boldsymbol{f}$ from $\mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$, here, we define the batch-size as $B = n_{sample}$. Finally, the adversarial example $\boldsymbol{x}^{adv}$ can be derived as

$$\boldsymbol{x}^{adv} = \boldsymbol{x}_{high} \otimes clip(\boldsymbol{f}^{adv}, -\xi, \xi) \oplus \boldsymbol{x}_{low}. \quad (14)$$

The whole algorithm of STBA is listed in Alg. 1 for ease of reproduction of our results, where lines 4-20 depict the core optimization process.

Algorithm 1 Spatial Transform for Query-Efficient Attack

Input: $\boldsymbol{\mathcal{C}}$: the target classifier to be attacked; $\boldsymbol{x}$: the clean image; $\boldsymbol{Q}_{max}$: the maximum querying number; q: the current querying number; ξ: the flow budget; lr: the learning rate; n_{sample}: the number of samples; $adjust_{num}$: the epoch to update the flow budget.
Parameter: The flow field $\boldsymbol{f} = [0, 1]^{2*h*w}$.
Output: The adversarial example $\boldsymbol{x}^{adv}$ used for attack.

1: Initialize the flow budget ξ with $\xi = 10^{-1}$.
2: Initialize $\boldsymbol{\mu}$ randomly.
3: Initialize the parameters of the flow $\boldsymbol{f_0}$ with zeros.
4: Decompose $\boldsymbol{x}$ to $\boldsymbol{x}_{high}$ and $\boldsymbol{x}_{low}$.
5: **while** $q <= \boldsymbol{Q}_{max}/n_{sample}$ **do**
6: Draw n_{sample} samples from $\boldsymbol{\delta_f} = \boldsymbol{\mu} + \sigma \cdot \boldsymbol{\epsilon}$ s.t. $\boldsymbol{\epsilon} \sim \boldsymbol{\mathcal{N}}(\epsilon|\boldsymbol{0}, \boldsymbol{I})$.
7: Set $\boldsymbol{f_k} = \boldsymbol{\delta_{f_k}}$ for all $k = 1, ..., n_{sample}$.
8: Compute ${\boldsymbol{x}_{high}^{adv}}^k$ by Eq. 14 for all $k = 1, ..., n_{sample}$.
9: Calculate $\boldsymbol{\mathcal{L}_k} = \boldsymbol{\mathcal{L}}({\boldsymbol{x}_{high}^{adv}}^k \oplus \boldsymbol{x}_{low})$ by Eq. 9 for all $k = 1, ..., n_{sample}$.
10: Normalize $\hat{\boldsymbol{\mathcal{L}}}_k = (\boldsymbol{\mathcal{L}}_k - mean(\boldsymbol{\mathcal{L}}))/std(\boldsymbol{\mathcal{L}})$.
11: Compute $\nabla_{\boldsymbol{\mu}} \boldsymbol{J}(\boldsymbol{\mu}, \sigma) = \frac{1}{n_{sample}} \sum_{k=1}^{n_{sample}} \hat{\mathcal{L}} \cdot \boldsymbol{\epsilon}_k$.
12: update $\boldsymbol{\mu} \longleftarrow \boldsymbol{\mu} - lr \nabla_{\boldsymbol{\mu}} \boldsymbol{J}(\boldsymbol{\mu}, \sigma)$.
13: **if** q % $adjust_{num} = 0$ **then**
14: Update ξ_t by Eq.8.
15: **end if**
16: Compute $\boldsymbol{x}_t^{adv} = \boldsymbol{x} \otimes clip(\boldsymbol{f}_0 + \boldsymbol{\mu}, -\xi_t, \xi_t) \oplus \boldsymbol{x}_{low}$.
17: **if** $\boldsymbol{x}_t^{adv}$ attack $\boldsymbol{\mathcal{C}}$ successfully **then**
18: break.
19: **end if**
20: **end while**
21: **return** $\boldsymbol{x}_t^{adv}$

IV. Experiments

A. Settings

The maximum number of queries is set to 1000 and 10000 for STBA and the baselines to simulate a realistic attack scenario. And the experimental results of those methods are reproduced by the code released in the original paper with default settings. All the experiments are conducted on a GPU server with 4 * NVIDIA Tesla A100 40GB GPU, 2 * Xeon Glod 6112 CPU, and RAM 512GB.

Dataset: We verify the performance of our method on four benchmark datasets for computer vision task, named CIFAR-10[1] [45], CIFAR-100[2] [45], STL-10[3] [46], and ImageNet-1k[4] [29]. In detail, CIFAR-10 contains 50,000 training images and 10,000 testing images with the size of 3x32x32 from 10 classes; CIFAR-100 has 100 classes, including the same number of training and testing images as the CIFAR-10; STL-10 has 10 classes, contains 500 training images and 800 testing images per class with image size of 3x96x96; ImageNet-

[1]http://www.cs.toronto.edu/ kriz/cifar.html
[2]http://www.cs.toronto.edu/ kriz/cifar.html
[3]https://cs.stanford.edu/ acoates/stl10/
[4]https://image-net.org/

TABLE I
THE VICTIM MODELS AND THEIR CORRESPONDING CLASSIFICATION ACCURACY (%) ON THE VALIDATION SET OF CIFAR-10, CIFAR-100, STL-10 AND THE SUBSET OF IMAGENET DATASET NIPS2017, RESPECTIVELY.

Type	CIFAR-10		CIFAR-100		STL-10		ImageNet	
	Model Name	Accuracy	Model Name	Accuracy	Model Name	Accuracy	Model Name	Accuracy
Normal	VGG-19	93.91	VGG-19	73.84	VGG-19	75.67	VGG-19	89.00
	ResNet-56	94.38	ResNet-56	72.60	ResNet-56	84.58	ResNet-152	94.00
	MobileNetV2	93.12	MobileNetV2	71.13	MobileNetV2	78.56	MobileNetV2	87.80
	ShuffleNetV2	93.98	ShuffleNetV2	75.49	ShuffleNetV2	79.46	DenseNet-121	70.20
Robust	Sehwag2020Hydra	88.98	Pang2022Robustness	65.56	FreeAT	63.77	Salman2020Do_R50	77.80
	Wang2020Improving	87.50	Addepalli2022Efficient	65.45	FastAdv	65.65	Engstrom2019Robustness	77.40
	Zhang2019Theoretically	84.92	Sehwag2021Proxy	65.93	TRADES	74.10	Wong2020Fast	62.70
	Wong2020Fast	83.33	Cui2020Learnable	70.24	MART	71.14	Salman2020Do_R18	63.10

TABLE II
ATTACK SUCCESS RATE (ASR,%) AGAINST NORMAL MODELS ON FOUR BENCHMARK DATASETS UNDER QUERY BUDGET IS SET TO **1000** AND **10000**

Max_Q		1000							10000						
Dataset	Model	Square	AdvFlow	RS	ASH	NP	Dif	STBA	Square	AdvFlow	RS	ASH	NP	Dif	STBA
CIFAR-10	VGG-19	90.27	64.71	88.99	96.47	90.91	96.89	**99.64**	99.65	97.05	91.38	96.51	99.70	99.97	**100.00**
	ResNet-56	97.14	69.19	62.79	98.44	94.31	99.13	**100.00**	99.94	99.64	67.74	98.44	**100.00**	**100.00**	**100.00**
	MobileNetv2	98.99	79.46	82.32	98.74	97.10	99.58	**99.97**	99.77	99.74	62.49	98.75	**100.00**	**100.00**	**100.00**
	ShuffleNetv2	99.56	65.78	84.18	96.83	92.01	97.12	**99.86**	99.89	98.80	58.13	96.83	**100.00**	**100.00**	**100.00**
CIFAR-100	VGG-19	92.93	64.37	90.48	95.12	88.31	91.58	**98.17**	98.83	96.03	93.12	95.12	98.66	99.42	**99.98**
	ResNet-56	99.82	90.32	84.18	98.93	95.47	99.06	**100.00**	99.86	99.93	86.88	98.93	99.97	**100.00**	**100.00**
	MobileNetv2	99.47	88.83	82.32	98.51	96.34	98.92	**100.00**	99.80	99.82	84.08	98.54	99.74	**100.00**	**100.00**
	ShuffleNetv2	97.83	77.76	68.79	94.78	90.61	95.80	**99.77**	99.84	98.99	71.25	94.78	99.79	99.96	**100.00**
STL-10	VGG-19	89.25	77.70	42.97	95.12	95.07	83.65	**99.00**	96.57	99.47	56.32	91.05	99.43	99.75	**100.00**
	ResNet-56	96.78	73.94	28.12	91.41	90.34	96.90	**99.80**	**100.00**	99.69	42.99	91.57	**100.00**	**100.00**	**100.00**
	MobileNetv2	**98.36**	77.32	33.59	91.99	96.66	91.95	97.10	99.85	99.40	64.24	89.65	**100.00**	**100.00**	**100.00**
	ShuffleNetv2	96.82	79.08	33.13	90.82	94.75	92.17	**98.50**	**100.00**	99.81	42.15	90.43	**100.00**	99.79	**100.00**
ImageNet	VGG-19	81.25	67.09	42.97	95.12	95.07	93.48	**99.52**	99.48	99.21	92.36	96.09	**100.00**	99.84	**100.00**
	ResNet-152	**96.78**	64.07	28.12	91.41	90.48	82.19	95.87	98.61	99.52	78.82	93.36	99.84	99.68	**100.00**
	MobileNetv2	98.36	79.01	33.59	91.99	96.66	95.55	**99.36**	**100.00**	99.84	83.16	92.38	**100.00**	**100.00**	**100.00**
	ShuffleNetv2	96.82	79.01	33.13	90.82	94.75	89.19	**99.05**	99.29	99.84	82.47	91.99	**100.00**	99.84	**100.00**

1K has 1,000 categories, containing about 1.3M samples for training and 50,000 samples for validation. In particular, in this paper, we carry our attack on the whole images in testing datasets of CIFAR-10, CIFAR-100 and STL-10, in terms of ImageNet-1k, flowing previous works [47]–[50], we are using its subset datasets from NIPS2017 Adversarial Learning Challenge[5].

Victim Models: For normal trained models, we adopted the pre-trained VGG-19_bn (VGG-19) [51], ResNet-56 [28], MobileNet-V2 [52] and ShuffleNet-V2 [53] for CIFAR-10 and CIFAR-100, all the models' weights are provided in the GitHub Repository [6], and we train these models on STL-10 by ourselves. For ImageNet, we use the PyTorch officially pre-trained model VGG-19 [51], ResNet-152 [28], MobileNet-V2 [52] and DenseNet-121 [54] as the victim models, respectively.

To investigate the performance of the proposed method in attacking robust models, we select some of the most recent defense techniques as follows: they are including HYDRA [55], Wang_{adv} [56], Zhang_{adv} [57], FastAdv [58] for CIFAR-10, Pang_{adv} [59], Efficient_{adv} [60], Proxy [61], Learn_{adv} [62] for CIFAR-100, FreeAT [63], FastAdv [58], TRADES [57] and MART [56] for STL-10, and Salman_{R50} [64], Engstrom_{adv} [65], FastAdv [58], Salman_{R18} [64] for ImageNet. For all the models of CIFAR-10, CIFAR-100, and ImageNet, we use their implementation in the robustbench toolbox[7] [66] and the models' weights are also provided in this toolbox [66]. For all these models, we chose their L_∞-norm version parameters because we carry out L_∞-norm attacks in this paper. Similar to normal models, we train the robust models of STL-10 by ourselves with the help of their officially provided codes, and we listed the recognition performance of all these aforementioned models on tested images in the Table. I.

Furthermore, three online vision models are involved in evaluating the attack in the physical world, including Google

[5] https://www.kaggle.com/competitions/nips-2017-non-targeted-adversarial-attack/data

[6] https://github.com/chenyaofo/pytorch-cifar-models

[7] https://github.com/RobustBench/robustbench

TABLE III
ATTACK SUCCESS RATE (ASR, %) AGAINST ROBUST MODELS ON FOUR BENCHMARK DATASETS UNDER QUERY BUDGET IS SET TO **1000** AND **10000**.

Max_Q		1000							10000						
Dataset	Model	Square	AdvFlow	RS	ASH	NP	Dif	STBA	Square	AdvFlow	RS	ASH	NP	Dif	STBA
CIFAR-10	HYDRA	13.33	36.24	29.87	28.99	7.49	21.89	**67.30**	20.86	58.60	31.56	29.18	19.68	47.19	**98.24**
	Wang_{adv}	13.32	33.58	24.10	22.13	8.79	20.79	**61.53**	20.67	54.82	25.12	22.42	20.38	43.50	**93.27**
	Zhang_{adv}	18.45	43.17	37.37	36.57	10.99	27.86	**78.67**	26.35	63.84	38.61	36.77	27.37	52.22	**98.14**
	FastAdv	23.14	50.39	37.27	43.26	13.79	33.49	**80.12**	34.05	72.88	38.67	43.33	32.97	61.57	**96.83**
CIFAR-100	Pang_{adv}	26.38	38.35	51.23	40.25	13.39	30.47	**82.17**	33.89	61.38	52.82	40.25	27.07	56.41	**98.34**
	Efficient_{adv}	31.60	43.11	40.14	39.83	12.99	35.27	**83.83**	41.96	68.81	41.96	39.83	30.48	**100.00**	98.52
	Proxy	27.78	39.14	51.84	39.34	11.09	31.36	**84.84**	53.47	64.46	53.47	39.62	27.17	**99.06**	98.87
	Learn_{adv}	28.16	36.42	53.88	48.21	9.99	32.69	**95.93**	55.03	67.78	55.03	49.12	28.97	87.76	**99.58**
STL-10	Free_AT	54.75	75.07	46.35	**79.04**	25.87	33.35	61.58	64.06	91.24	46.35	83.66	50.19	70.40	**97.34**
	FastAdv	47.80	**73.99**	35.85	73.31	25.77	33.86	63.20	60.95	92.49	35.85	79.95	51.36	71.43	**97.92**
	TRADES	60.52	69.49	24.78	80.21	51.25	54.75	**90.93**	82.01	94.93	24.78	85.09	92.22	93.57	**100.00**
	MART	53.23	71.73	35.85	73.24	48.55	49.69	**88.97**	72.59	92.60	29.61	80.08	82.88	87.02	**100.00**
ImageNet	Slman_$R18$	39.26	31.12	36.33	49.02	36.33	8.16	**70.02**	52.73	70.78	75.35	67.38	59.53	70.78	**97.53**
	Slman_$R50$	17.97	34.91	18.55	36.13	18.55	8.35	**54.84**	33.79	70.97	64.58	55.08	43.97	70.97	**95.45**
	Engsrom_$R18$	18.36	36.81	18.55	40.82	18.55	9.49	**44.97**	38.67	73.06	65.62	62.11	46.69	73.06	**95.64**
	FastAdv	36.72	51.80	30.86	**55.86**	30.86	18.41	50.47	55.47	86.15	76.39	72.27	53.70	86.15	**96.20**

Cloud Vision[8], Tencent Cloud[9] and Baidu AI Cloud[10].

Baselines: The baselines for query-limited score-based black-box attack settings are Square Attack [1], AdvFlow [2], RS_Attack [3], ASH [4], NPAttack [5] and DifAttack [6]. For Square Attack, AdvFlow, ASH, and NPAttack, we set the $L_\infty = 0.05$, and for RS_Attack, we set the patch size as 3x3 for CIFAR-10 and CIFAR-100, 8x8 for STL-10, and 20x20 for ImageNet. For AdvFlow, we use the officially pre-trained flow weights on CIFAR-10 and train on other datasets by ourselves, and for the DifAttack, we use the SqueezeNet [67] as the surrogate model.

Metrics: We compare our proposed method with several query-efficient black-box attack techniques concerned with Attack Success Rate (ASR), Average Query number (Average.Q) and Median Query number (Med.Q), and image quality involving LPIPS [68], DISTS [69], MAD [70], BRISQUE [71], FID [72], SSIM [73], PSNR, VIF [74] and FSIM [75].

B. Quantitative Comparison with the Query-Efficient Black-box Attacks

In this subsection, we will evaluate the proposed STBA and the baselines in attacking performance and query efficiency on the validation dataset of CIFAR-10, CIFAR-100, STL-10 and the whole NIPS2017 dataset. Note that, in this part, we use RS, NP, and Dif to represent RS_Attack, NPAttack, and DifAttack, respectively.

Table. II and Table. III show the attack results of all attacks on the normally trained and robustly trained models, respectively. As can be seen, STBA can improve upon the performance of baseline methods in all of the target models in terms of the attack success rate. For normally trained models, even when the max query is set to 1000, STBA can obtain nearly 100% attack success rate, while the baselines are struggling to achieve such a high attack success rate, in most cases, their attack success rates are lower than 95% in this query-limited situation. Besides, when attacking robust models, baseline methods perform poorly under the max query is limited to 1000, while STBA can obtain 61.53 $\sim$ 80.12%, 82.17 $\sim$ 95.93%, 61.58 $\sim$ 90.93% and 44.97 $\sim$ 70.02% attack success rate on the four different benchmark datasets, respectively. Even if we increased the max query budget to 10000, existing methods would still find it hard to attack the robust models successfully and can only get 19.68 $\sim$ 92.22% attack success rate. In contrast, STBA can obtain 93.27 $\sim$ 100% attack success rate.

The empirical results indicated that although previous techniques show fantastic performance in query-based black-box attack settings, once we put a relatively extreme limit on the maximum number of queries, these methods will lose their advantages and show dissatisfactory results. Besides, such results also verify our assumption that the previous techniques struggle to attack the robustly trained model successfully.

Furthermore, we exhibit the attack performance and query times of all methods in Fig. 3 and Fig. 4, respectively. From Fig. 4, attacking results on robust models, we can clearly find that under a limited query budget, i.e., 5000, the baselines can hardly achieve 40+% attack success rate on the CIFAR-10 dataset, while our method can obtain nearly 80% attack success rate. For the ImageNet dataset, our method can achieve a more satisfactory attack performance as the query budget grows and wins the best results with the maximal query budget. In contrast, others can only obtain about a 60% attack success rate and cannot even obtain a higher attack success rate by increasing the query budget. These

[8]https://cloud.google.com/vision
[9]https://cloud.tencent.com/
[10]https://cloud.baidu.com/

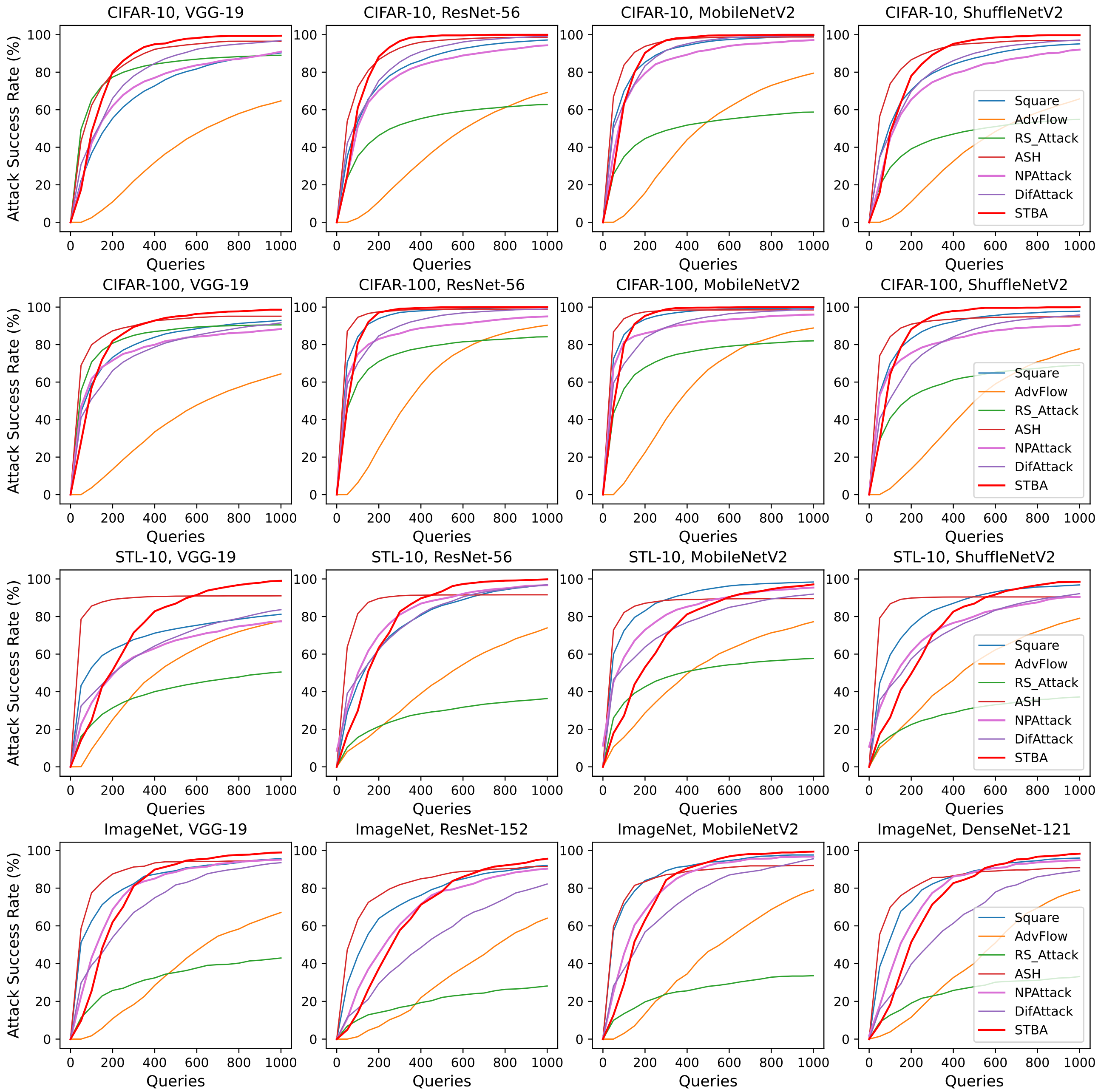


Fig. 3. The attack success rate $vs.$ query numbers on **clean** models of the baselines and the proposed method, respectively, where the max query numbers are set to **1000**. The **bold red** lines are the results of the proposed STBA.

results fully support our claims again: 1) Although existing black-box attacks can achieve high attack success rates, even reaching 100% attack success rate in some situations, they require a large amount of interaction from the victim model to obtain useful information to optimize perturbations. Once the number of queries is limited, their attack performance will be significantly affected. 2) Although existing black-box attack methods have achieved satisfactory attack results on normally trained black-box models, they are still difficult to successfully attack robust models with a high attack success rate, whose robustness is usually improved by adversarial training. This dramatically limits their attack effect in practical situations because most of the deep learning models deployed in the real world are relatively robust or equipment defense mechanisms. 3) Existing adversarial training-based robust models still show vulnerability against the newly proposed attacks, which are not involved in providing adversarial examples in the process of adversarial training or fine-tuning.

C. Transferability

The motivation of the current work states that the generation of adversarial examples is generally based on the assumption of transferability, which means that an adversarial example generated according to a model can be used to attack other unseen models. To see if this assumption is valid in black-box settings, here, we follow the previous works [2], [76] to examine the transferability of the generated adversarial examples across different models on CIFAR-10, CIFAR-100, STL-10

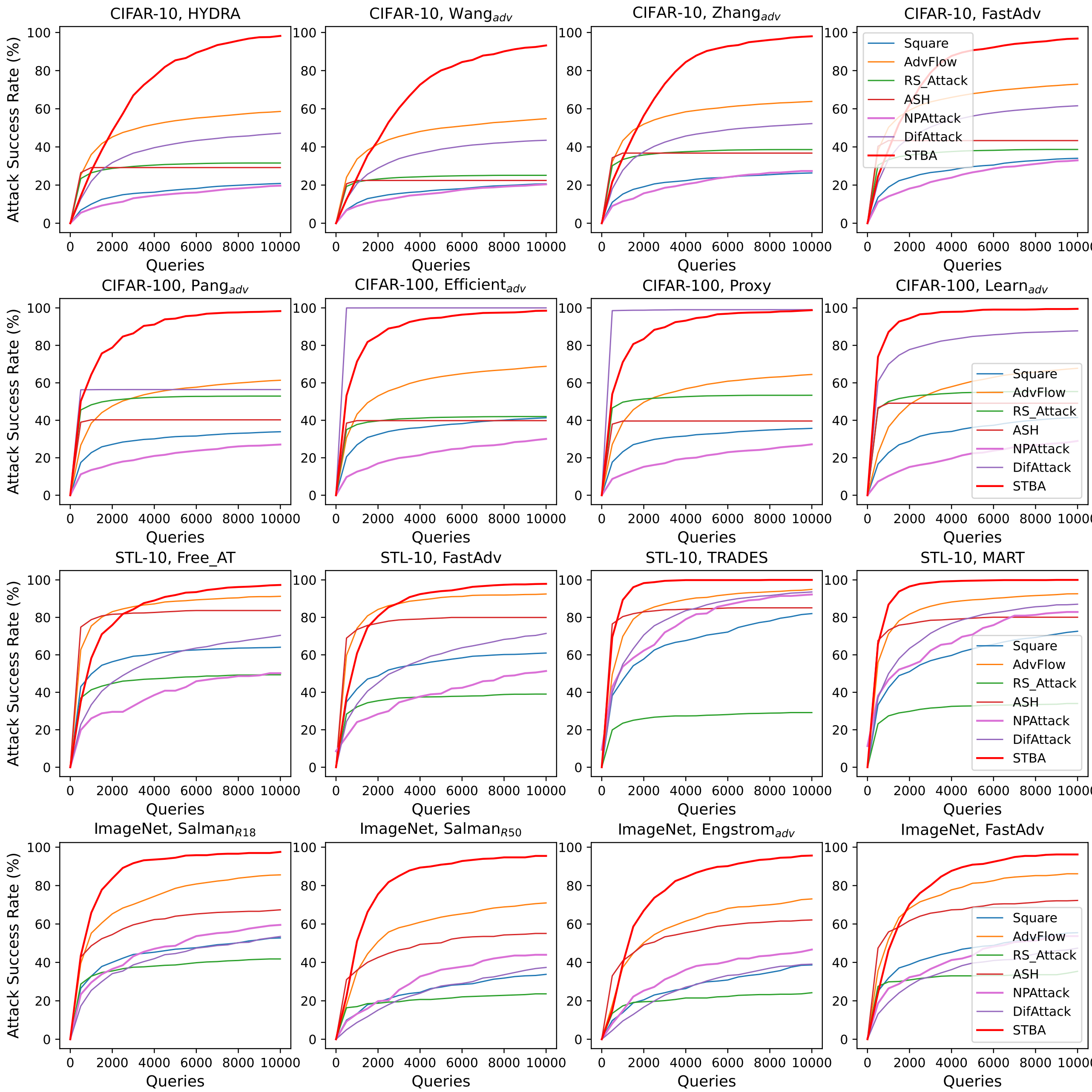


Fig. 4. The attack success rate $vs.$ query numbers on **robust** models of the baselines and the proposed method, respectively, where the max query numbers are set to **10000**. The **bold red** lines are the results of the proposed STBA.

and ImageNet. Specifically, we generate adversarial examples for the aforementioned eight models shown in Table I for each dataset, including four normal-trained and four adversarial-trained. We randomly selected 1000 images from the CIFAR-10, CIFAR-10, and STL-10 test set, and selected all images from ImageNet (NIPS2017), which are classified correctly by these models, whereas the corresponding adversarial examples are misclassified, for this experiment. Where such generated adversarial examples are used to attack the other models. Here, we set the maximal query number to 1000 for all baselines.

We illustrate the results of this part with the ASR confusion matrix on these four datasets, as shown in Fig. 5. The row represents which model is targeted during the generation of adversarial examples, while the column represents which model is attacked by the generated examples. From this figure, we can see that the highest transferability ASR on four datasets of STBA can achieve 74%, 83%, 57%, and 73%, respectively. In contrast, the comparable baseline method NPAttack is 66%, 84%, 47%, and 26%, respectively. The rest of the baselines even exhibit poor transferability. It means that the examples generated by STBA produce higher transfer ASR in most cases than baselines, especially for the STL-10 and ImageNet datasets, which validate the superior transferability of STBA. Meanwhile, this phenomenon shows that baselines heavily rely on the feedback of the target model during each query and cannot extract transferable features. By contrast, our method learns the adversarial flow field $\boldsymbol{f}$ that does not collapse to a specific model.

In addition, we were intrigued to discover that AEs generated on robust models exhibit superior transfer attack perfor-

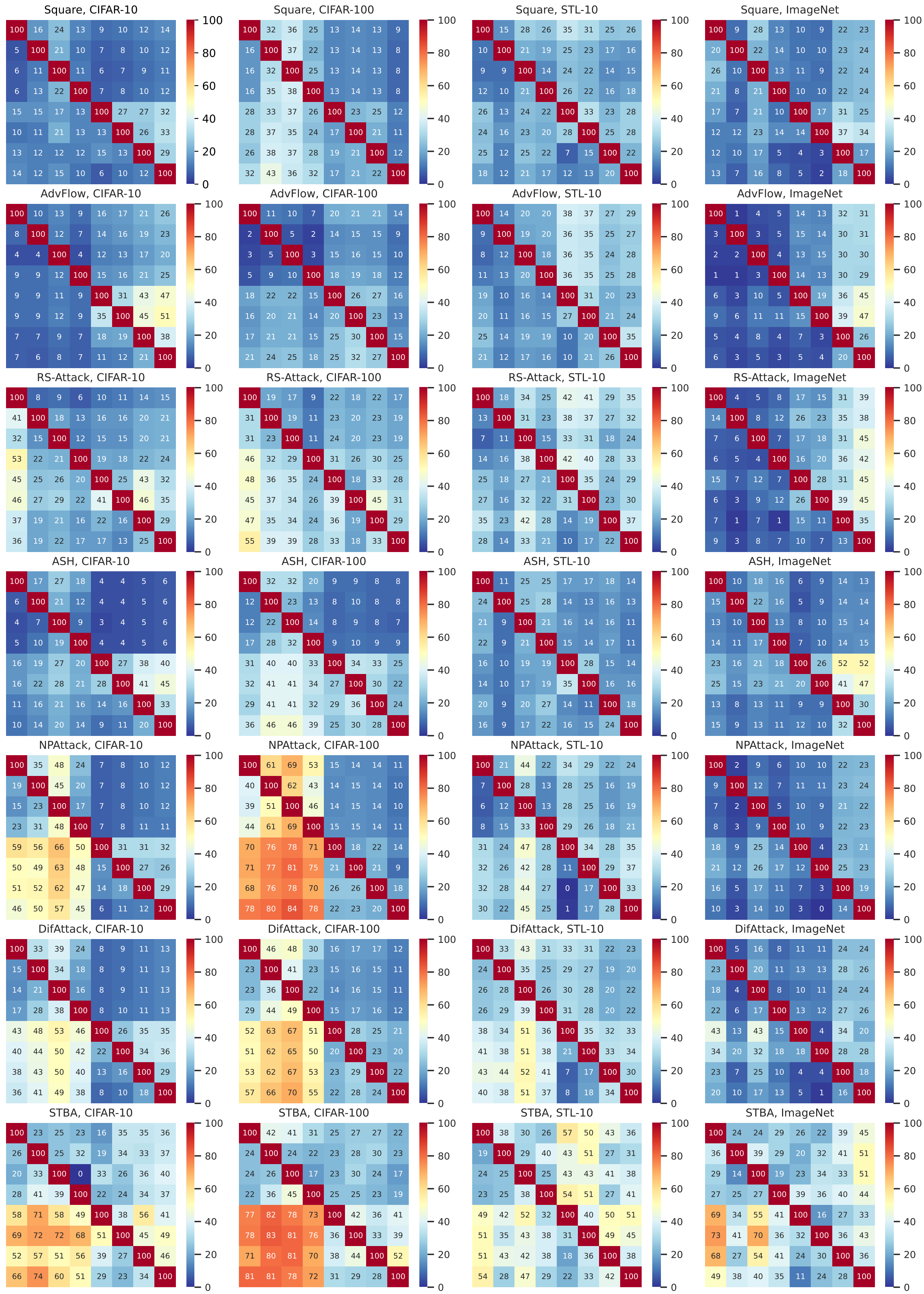


Fig. 5. Confusion matrix of transferability for adversarial attacks generated by baselines and the proposed STBA on CIFAR-10, CIFAR-100, STL-10 and ImageNet, respectively. The row represents which model is targeted during the generation of adversarial examples, while the column represents which model is attacked by such synthesized examples.

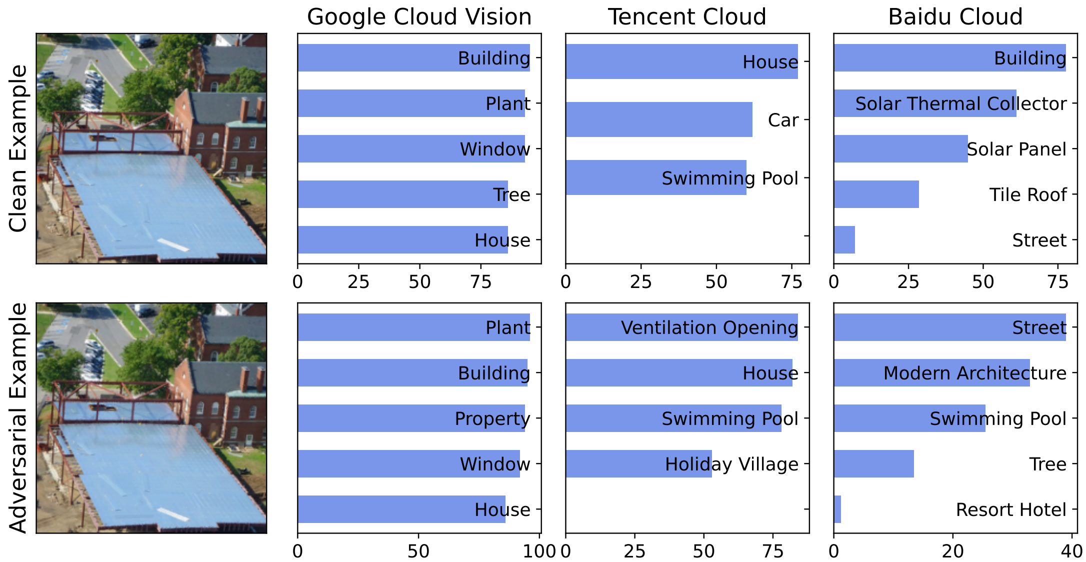


Fig. 6. The top-5 prediction confidence of the clean image and its corresponding adversarial counterpart returned by Google Cloud Vision, Tencent Cloud and Baidu Cloud, respectively.

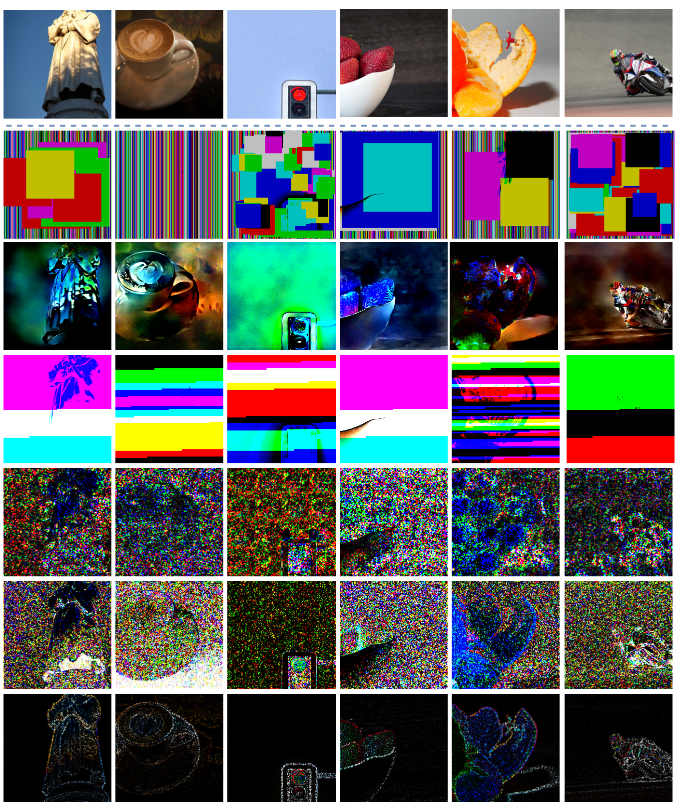


Fig. 7. The clean example and their corresponding adversarial noise generated by baselines and the proposed STBA. The first row is the clean images, and the flowing rows are the adversarial noise generated by Square Attack [1], AdvFlow [2], ASH [4], NPAttack [5], DifAttack [6], and STBA, respectively.

mance. This finding suggests that researchers can effectively leverage AEs crafted on robust models to conduct transfer attacks on various other models. Such a strategy allows for a more efficient exploration of vulnerabilities across existing models. Furthermore, this approach can significantly aid in the identification of critical weaknesses, thereby encouraging defenders to develop more robust and effective defense mechanisms. This could lead to enhancements in overall model robustness, promoting more secure AI applications.

D. Visual Quality

Different from the previous works using L_p-norm to evaluate the generated adversarial images' quality, in this paper, we follow previous work [33] and use the following perceptual metrics to evaluate the adversarial examples generated by our method and baselines, such as: Learned Perceptual Image Patch Similarity (LPIPS) metric [68], Deep Image Structure and Texture Similarity (DISTS) index [69], Mean Absolute Difference (MAD) [70], Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [71], Fréchet Inception Distance (FID) [72], Structure Similarity Index Measure (SSIM) [73], Peak Signal-to-Noise Ratio (PSNR), Visual Information Fidelity (VIF) [74] and Feature Similarity Index Measure (FSIM) [75] to assess the generated images' qualities in terms of various aspects. The image quality assessment toolkit we used in the experiments of this part is IQA_pytorch[11]. Note that we omitted RS_Attack in this part because it is a patch-based attack.

The quantitative experimental results of images' quality can be seen in Table. IV, which indicated that the proposed method has the lowest MAD and BRISQUE (the lower is better), are 3.0883 and 16.4823, respectively, and has the highest SSIM, PSNR, VIF and FSIM (the higher is better), achieving 0.9879, 37.3324, 0.8258 and 0.9873, respectively, in comparison to the baselines on ImageNet dataset. The results show that the proposed method is superior to the existing attack methods.

Furthermore, to better observe the difference between the adversarial examples generated by our method and the baselines from the visual aspect, we draw the adversarial perturbation generated on ImageNet by baselines and the proposed method in Fig. 7, the target model is pre-trained VGG-16. The first row is the benign examples, and the following are the adversarial noise of Square [1], AdvFlow [2], ASH [4],

[11] https://www.cnpython.com/pypi/iqa-pytorch

TABLE IV
VISUAL QUALITY EVALUATION RESULTS CALCULATED ON ADVERSARIAL EXAMPLES CRAFTED BY SQUARE, ADVFLOW, ASH, NPATTACK, DIFATTACK, AND THE PROPOSED STBA.

Metric	Square	AdvFlow	ASH	NPAttack	DifAttack	STBA
LPIPS ↓	0.2448	**0.0135**	0.1298	0.1168	0.2079	0.0162
DISTS ↓	0.2677	**0.0366**	0.1625	0.1340	0.1869	0.0680
MAD ↓	90.7684	29.4512	73.7850	66.6841	87.9537	**3.0883**
BRISQUE ↓	29.7395	17.0994	20.3779	17.2939	26.6368	**16.4823**
FID ↓	97.0623	**8.5572**	50.4029	29.9010	44.4464	13.9358
SSIM ↑	0.8202	0.9865	0.9516	0.9122	0.7969	**0.9879**
PSNR ↑	26.1883	33.3259	26.1909	31.3421	28.1149	**37.3324**
VIF ↑	0.6240	0.8137	0.7887	0.5404	0.4456	**0.8258**
FSIM ↑	0.9025	0.9721	0.9630	0.9410	0.9001	**0.9873**

NPAttack [5], DifAttack [6] and our method, respectively. Noted that, for better observation, we magnified the noise by multiplying a factor of 25. From Fig. 7, we can clearly observe that baseline methods distort the image without ordering. In contrast, the adversarial perturbations generated by our method are focused on the target object instead of disturbing the whole image, and its noise contains more semantic information and is more imperceptible to human eyes.

E. Performance on Physical Online Vision Models

To evaluate the adversarial examples' effectiveness in real-world scenarios, we conduct experiments of attacking the online models on Google Cloud Vision, Tencent Cloud and Baidu AI Cloud, whose structures and parameters and datasets used for training in their platforms are entirely sightless for us. We use the ImageNet dataset to carry out our attack on the pre-trained ResNet-152 model and then we randomly choose an adversarial image and its clean counterpart to validate on three online image recognition platforms. We draw the returned top-5 results in Fig. 6, which shows that the attack has successfully changed the image's top-5 prediction results to a large degree [12]. Taking Baidu AI Cloud for instance, its top-5 outputs of the clean image like Building (77.67%), Solar Thermal Collector (61.18%), Solar Panel (44.82%), Tile Roof (28.49%), Street (7.09%). However, after attacking, its top-5 predictions changed to Street (38.97%), Modern Architecture (32.94%), Swimming Pool (25.50%), Tree (13.43%) and Resort Hotel (1.17%). Besides, for Tencent Cloud, the adversarial example even increases the output labels from 3 to 4. This phenomenon indicates that our attack method is also applicable to the real physical world.

F. Ablation Study

In this section, we extend the ablation studies to verify the attack performance (including attack success rate and query times) and the generated adversarial examples' visual quality under different hyper-parameter settings.

We first investigate the effectiveness of the proposed strategy in terms of applying the spatial transform to the high-frequency part to improve the attack performance and the

[12]We call these APIs on March 24, 2024.

TABLE V
THE ATTACK PERFORMANCE, QUERY EFFICIENCY, AND VISUAL QUALITY OF APPLYING THE SPATIAL TRANSFORM TO **HIGH-** $vs.$ **LOW-FREQUENCY** PART.

Dataset	Frequency	ASR	Avg.Q	Med.Q	LPIPS	DISTS
CIAFAR-10	High	99.40	142.47	111.00	0.02	0.05
	Low	31.40	387.19	342.00	0.05	0.07
CIFAR-100	High	99.60	132.58	89.00	0.02	0.05
	Low	48.10	363.68	298.00	0.06	0.08
STL-10	High	99.10	241.65	190.00	0.02	0.06
	Low	43.40	385.62	337.00	0.05	0.10
ImageNet	High	99.50	205.34	169.00	0.02	0.06
	Low	27.70	293.26	169.00	0.06	0.09

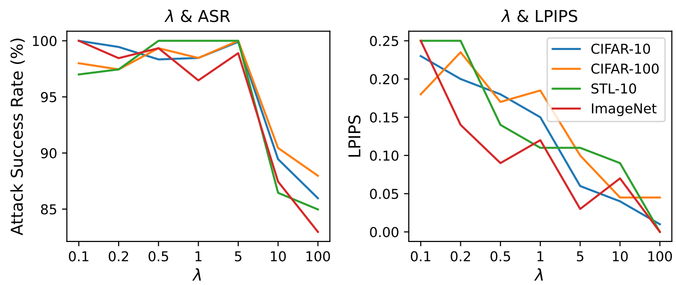


Fig. 8. Ablation study of attack performance and generated adversarial examples' quality under different loss weighting factor λ.

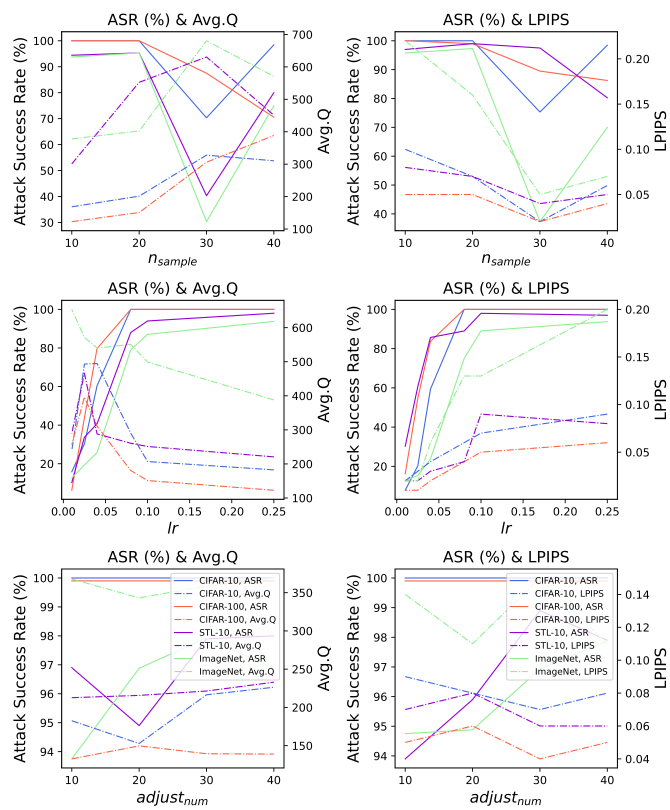


Fig. 9. Ablation study of attack performance and generated adversarial examples' visual quality under different hyper-parameters. The critical hyper-parameters involved here are the number of samples n_{sample}, the learning rate lr and the number of adjustments of flow field budget ξ $adjust_{num}$. We show the results of attack success rate, average query number and image quality in each row, respectively. The lines with different colors represent the result on different datasets, the solid lines represent the attack success rate, and the dashed lines indicate the average query number (Avg.Q) and image quality, respectively.

visual quality of generated adversarial examples. We list the results in Table V, which verified our claim and shows that applying spatial transform to the high-frequency part not only gets a good attack performance but also improves the query efficiency and image quality. The test images are randomly selected and correctly classified 1000 images for CIFAR-10, CIFAR-100, and STL-10, while the ImageNet (NIPS) are all correctly classified images. Here, the maximal query budget is set to 1000, and the victim model is VGG-19.

Besides, we also investigate the influence of loss function balancing factor λ on the performance of the proposed method. Specifically, we conduct a dissection experiment to systematically vary the balancing factor λ within the loss function and observe its effects on attack success rate and the image quality of the generated adversarial examples. The results Fig. 8 show that the attack can maintain a success rate when the λ is set to a small value, i.e., 0.1 to 5, and has an obvious drop when λ is greater than 5. While the image quality decreases with the λ increase. Therefore, to balance the attack performance and image quality of the generated images, we set the $\lambda = 5$ in this paper according to these empirical results.

Finally, in our method, other involved critical hyper-parameters are as follows: the number of samples n_{sample} to estimate the gradient, the learning rate lr and the number of adjustments of flow field budget ξ, $adjust_{num}$. First of all, we set $\sigma = 2 * lr$ and do the experiments under a predefined hyper-parameters set; we illustrate the results in Fig. 9. Specifically, we set the $n_{sample} \in \{10, 20, 30, 40\}$, $lr \in \{0.25, 0.1, 0.08, 0.04, 0.025, 0.01\}$ and $adjust_{num} \in \{10, 20, 30, 40\}$ and extent attacks on the ResNet (ResNet-56 for CIFAR and STL-10 dataset, and ResNet-152 for ImageNet dataset, respectively.) under the max query number is set to 1000. As the results show, for n_{sample}, we set $n_{sample} = 10$ to get the highest attack success rate and lowest average query number and set $n_{sample} = 30$ to get the best image quality, but the attack success rate will be faded for most victim datasets. For lr, we set $lr = 0.1$ to balance the attack success rate, average query number, and image quality. For $adjust_{num}$, STBA achieves the best attack performance and image quality when $adjust_{num} = 30$. Therefore, in our experiments, to obtain the adversarial images with high attack success and image quality, we set the most essential hyper-parameters as $n_{sample} = 10$, $lr = 0.1$ and $adjust_{num} = 20$, respectively.

V. Conclusions and Limitations

To evaluate the vulnerability and robustness of existing normally trained and robustly trained DNN models under query-limited black-box situations, in this paper, we present a novel non-noise additional method, called STBA, to build adversarial examples. The proposed STBA combines performing the spatial transformation to the high-frequency part of the image and gradient estimation to search for the best flow field to synthesize adversarial examples. Empirically, the adversarial examples generated by STBA are imperceptible to the human eyes and significantly increase query efficiency; it only needs dozens and hundreds of queries to obtain a practical adversarial example in attacking normally trained and robustly trained models, respectively. Extensive experiments show that the proposed method is superior to the existing methods in terms of attack success rate, query times and imperceptibility. Even on robust models, it can obtain $93.27 \sim 100\%$ attack success rate on benchmark datasets. Benefitting from generating adversarial examples without noise-adding, the proposed STBA provides a new efficient way to evaluate the vulnerability of existing classifiers and further enhance their robustness performance using techniques like fine-tuning or adversarial training.

Although our method can achieve better attack results and visual imperceptibility under query-limited black-box attack settings, we still cannot guarantee a 100% attack success rate all the time with such minor modifications. Besides, we find that the modification in some generated adversarial examples exhibits a jagged distortion because we struggle to calculate a more elaborate transform matrix with gradient estimation in a black-box attack setting. Further, we will solve this limitation by extending spatial transform operations in other image spaces, such as YUV, LAB, et al., instead of RGB space to generate more imperceptible adversarial examples in black-box scenarios under guaranteed attack performance.

Acknowledgments

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore and Infocomm Media Development Authority, Yunnan Province expert workstations, Yunnan Fundamental Research Projects.

References

[1] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *ECCV*, vol. 12368, 2020, pp. 484–501.

[2] H. M. Dolatabadi, S. M. Erfani, and C. Leckie, "Advflow: Inconspicuous black-box adversarial attacks using normalizing flows," in *NeurIPS*, 2020.

[3] F. Croce, M. Andriushchenko, N. D. Singh, N. Flammarion, and M. Hein, "Sparse-rs: A versatile framework for query-efficient sparse black-box adversarial attacks," in *AAAI*, 2022, pp. 6437–6445.

[4] S. Li, G. Huang, X. Xu, Y. Yang, and F. Shen, "Accelerated sign hunter: A sign-based black-box attack via branch-prune strategy and stabilized hierarchical search," in *ICMR*, 2022, pp. 462–470.

[5] Y. Bai, Y. Wang, Y. Zeng, Y. Jiang, and S. Xia, "Query efficient black-box adversarial attack on deep neural networks," *Pattern Recognit.*, vol. 133, p. 109037, 2023.

[6] J. Liu, J. Zhou, J. Zeng, and J. Tian, "Difattack: Query-efficient black-box adversarial attack via disentangled feature space," in *AAAI*, 2024, pp. 3666–3674.

[7] H. Xiong, R. Wan, J. Zhao, Z. Chen, X. Li, Z. Zhu, and J. Huan, "Grod: Deep learning with gradients orthogonal decomposition for knowledge transfer, distillation, and adversarial training," *ACM Trans. Knowl. Discov. Data*, vol. 16, no. 6, pp. 117:1–117:25, 2022.

[8] P. Li, Y. Zhang, L. Yuan, J. Zhao, X. Xu, and X. Zhang, "Adversarial attacks on video object segmentation with hard region discovery," *CoRR*, vol. abs/2309.13857, 2023.

[9] S. He, R. Wang, T. Liu, C. Yi, X. Jin, R. Liu, and W. Zhou, "Type-i generative adversarial attack," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 3, pp. 2593–2606, 2023.

[10] W. Xing, J. Yao, Z. Liu, W. Liu, S. Zhang, and L. Wang, "Contrastive JS: A novel scheme for enhancing the accuracy and robustness of deep models," *IEEE Trans. Multim.*, vol. 25, pp. 7881–7893, 2023.

[11] H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *Int. J. Autom. Comput.*, vol. 17, no. 2, pp. 151–178, 2020.

[12] Z. Zhang, X. Wang, G. Lu, F. Shen, and L. Zhu, "Targeted attack of deep hashing via prototype-supervised adversarial networks," *IEEE Trans. Multim.*, vol. 24, pp. 3392–3404, 2022.

[13] H. Yuan, Q. Chu, F. Zhu, R. Zhao, B. Liu, and N. Yu, "Automa: Towards automatic model augmentation for transferable adversarial attacks," *IEEE Trans. Multim.*, vol. 25, pp. 203–213, 2023.

[14] Y. Wu, F. Li, H. Bai, W. Lin, R. Cong, and Y. Zhao, "Bridging component learning with degradation modelling for blind image super-resolution," *IEEE Trans. Multim.*, 2022.

[15] X. Tu, Z. Ma, J. Zhao, G. Du, M. Xie, and J. Feng, "Learning generalizable and identity-discriminative representations for face anti-spoofing," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 5, pp. 60:1–60:19, 2020.

[16] S. Tang, D. Guo, R. Hong, and M. Wang, "Graph-based multimodal sequential embedding for sign language translation," *IEEE Trans. Multim.*, vol. 24, pp. 4433–4445, 2022.

[17] R. Liu, W. Zhou, T. Zhang, K. Chen, J. Zhao, and K. Lam, "Boosting black-box attack to deep neural networks with conditional diffusion models," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 5207–5219, 2024.

[18] W. Wang, B. Yin, T. Yao, L. Zhang, Y. Fu, S. Ding, J. Li, F. Huang, and X. Xue, "Delving into data: Effectively substitute training for black-box attack," in *CVPR*, 2021, pp. 4761–4770.

[19] X. Du and C. Pun, "Robust audio patch attacks using physical sample simulation and adversarial patch noise generation," *IEEE Trans. Multim.*, vol. 24, pp. 4381–4393, 2022.

[20] X. Zhang, X. Zhang, W. Liu, X. Zou, M. Sun, and J. Zhao, "Waveform level adversarial example generation for joint attacks against both automatic speaker verification and spoofing countermeasures," *Eng. Appl. Artif. Intell.*, vol. 116, p. 105469, 2022.

[21] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[22] P. Chen, H. Zhang, Y. Sharma, J. Yi, and C. Hsieh, "ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *AISec@CCS*, 2017, pp. 15–26.

[23] S. Zhang, D. Zuo, Y. Yang, and X. Zhang, "A transferable adversarial belief attack with salient region perturbation restriction," *IEEE Trans. Multim.*, vol. 25, pp. 4296–4306, 2023.

[24] Y. Cheng, Q. Guo, F. Juefei-Xu, S. Lin, W. Feng, W. Lin, and Y. Liu, "Pasadena: Perceptually aware and stealthy adversarial denoise attack," *IEEE Trans. Multim.*, vol. 24, pp. 3807–3822, 2022.

[25] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, "NATTACK: learning the distributions of adversarial examples for an improved black-box attack on deep neural networks," in *ICML*, vol. 97, 2019, pp. 3866–3876.

[26] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *ICLR*, 2018.

[27] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, "Simple black-box adversarial attacks," in *ICML*, vol. 97, 2019, pp. 2484–2493.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[29] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[30] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *NeurIPS*, 2015, pp. 2017–2025.

[31] C. Xiao, J. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," in *ICLR*, 2018.

[32] Y. Zhang, W. Ruan, F. Wang, and X. Huang, "Generalizing universal adversarial attacks beyond additive perturbations," in *ICDM*, 2020, pp. 1412–1417.

[33] A. Aydin, D. Sen, B. T. Karli, O. Hanoglu, and A. Temizel, "Imperceptible adversarial examples by spatial chroma-shift," in *ADVM*, 2021, pp. 8–14.

[34] R. Liu, X. Jin, D. Hu, J. Zhang, Y. Wang, J. Zhang, and W. Zhou, "Dualflow: Generating imperceptible adversarial examples by flow field and normalize flow-based model," *Frontiers Neurorobotics*, vol. 17, 2023.

[35] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *NeurIPS*, 2018, pp. 10 236–10 245.

[36] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *ICML*, vol. 80, 2018, pp. 2142–2151.

[37] M. Cheng, S. Singh, P. H. Chen, P. Chen, S. Liu, and C. Hsieh, "Sign-opt: A query-efficient hard-label adversarial attack," in *ICLR*, 2020.

[38] C. Ma, L. Chen, and J. Yong, "Simulating unknown target models for query-efficient black-box attacks," in *CVPR*, 2021, pp. 11 835–11 844.

[39] F. Suya, Y. Tian, D. Evans, and P. Papotti, "Query-limited black-box attacks to classifiers," *CoRR*, vol. abs/1712.08713, 2017.

[40] H. Li, X. Xu, X. Zhang, S. Yang, and B. Li, "QEBA: query-efficient boundary-based blackbox attack," in *CVPR*, 2020, pp. 1218–1227.

[41] F. Suya, J. Chi, D. Evans, and Y. Tian, "Hybrid batch attacks: Finding black-box adversarial examples with limited queries," in *USENIX Security*, 2020, pp. 1327–1344.

[42] L. Gao, Z. Huang, J. Song, Y. Yang, and H. T. Shen, "Push & pull: Transferable adversarial examples with attentive attack," *IEEE Trans. Multim.*, vol. 24, pp. 2329–2338, 2022.

[43] F. Yin, Y. Zhang, B. Wu, Y. Feng, J. Zhang, Y. Fan, and Y. Yang, "Generalizable black-box adversarial attack with meta learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 3, pp. 1804–1818, 2024.

[44] L. Wasserman, *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2004.

[45] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Computer Science Department, University of Toronto, Tech. Rep*, vol. 1, 01 2009.

[46] A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *AISTATS*, vol. 15, 2011, pp. 215–223.

[47] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. L. Yuille, "Mitigating adversarial effects through randomization," in *ICLR*, 2018.

[48] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *CVPR*, 2018.

[49] W. Peng, R. Liu, R. Wang, T. Cheng, Z. Wu, L. Cai, and W. Zhou, "Ensemblefool: A method to generate adversarial examples based on model fusion strategy," *Comput. Secur.*, vol. 107, p. 102317, 2021.

[50] R. Liu, J. Zhang, H. Li, J. Zhang, Y. Wang, and W. Zhou, "AFLOW: developing adversarial examples under extremely noise-limited settings," in *ICICS*, vol. 14252, 2023, pp. 502–518.

[51] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[52] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520.

[53] N. Ma, X. Zhang, H. Zheng, and J. Sun, "Shufflenet V2: practical guidelines for efficient CNN architecture design," in *ECCV*, vol. 11218, 2018, pp. 122–138.

[54] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017, pp. 2261–2269.

[55] V. Sehwag, S. Wang, P. Mittal, and S. Jana, "HYDRA: pruning adversarially robust neural networks," in *NeurIPS*, 2020.

[56] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *ICLR*, 2020.

[57] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *ICML*, vol. 97, 2019, pp. 7472–7482.

[58] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *ICLR*, 2020.

[59] T. Pang, M. Lin, X. Yang, J. Zhu, and S. Yan, "Robustness and accuracy could be reconcilable by (proper) definition," in *ICML*, vol. 162, 2022, pp. 17 258–17 277.

[60] S. Addepalli, S. Jain, and V. B. R., "Efficient and effective augmentation strategy for adversarial training," in *NeurIPS*, 2022.

[61] V. Sehwag, S. Mahloujifar, T. Handina, S. Dai, C. Xiang, M. Chiang, and P. Mittal, "Robust learning meets generative models: Can proxy distributions improve adversarial robustness?" in *ICLR*, 2022.

[62] J. Cui, S. Liu, L. Wang, and J. Jia, "Learnable boundary guided adversarial training," in *ICCV*, 2021, pp. 15 701–15 710.

[63] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. P. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" in *NeurIPS*, 2019, pp. 3353–3364.

[64] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry, "Do adversarially robust imagenet models transfer better?" in *NeurIPS*, 2020.

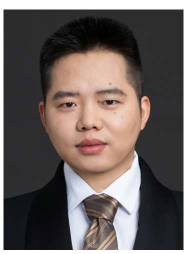
[65] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry, "Adversarial robustness as a prior for learned representations," *arXiv*, 2019.

[66] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein, "Robustbench: a standardized adversarial robustness benchmark," in *NeurIPS*, 2021.

[67] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and $<$1mb model size," *CoRR*, vol. abs/1602.07360, 2016.

[68] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018, pp. 586–595.
[69] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2567–2581, 2022.
[70] E. C. Larson and D. M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," *J. Electronic Imaging*, vol. 19, no. 1, p. 011006, 2010.
[71] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, 2012.
[72] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *NeurIPS*, 2017, pp. 6626–6637.
[73] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
[74] Y. Han, Y. Cai, Y. Cao, and X. Xu, "A new image fusion performance metric based on visual information fidelity," *Information Fusion*, vol. 14, no. 2, pp. 127–135, 2013.
[75] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "Fsim: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, 2011.
[76] P. Zhao, P. Chen, S. Wang, and X. Lin, "Towards query-efficient black-box adversary with zeroth-order natural gradient descent," in *AAAI*, 2020, pp. 6909–6916.

BIOGRAPHY SECTION



Renyang Liu (Graduate Student Member, IEEE) earned his B.E. in Computer Science from Northwest Normal University in 2017 and his Ph.D. from Yunnan University in 2024. He served as a Joint-training Ph.D. student at Nanyang Technological University's College of Computing and Data Science from 2022 to 2023. Currently, he is a Research Fellow at the Institute of Data Science, National University of Singapore. His research primarily focuses on security issues of multimodal models, with interests spanning AI security, machine unlearning, data privacy, and computer vision.

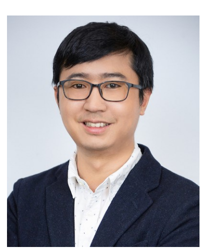
Kwok-Yan Lam (Senior Member, IEEE) received his B.Sc. degree (1st Class Hons.) from University of London, in 1987, and Ph.D. degree from University of Cambridge, in 1990. He is the Associate Vice President (Strategy and Partnerships) and Professor in the College of Computing and Data Science at the Nanyang Technological University, Singapore. He is currently also the Director of the Strategic Centre for Research in Privacy-Preserving Technologies and Systems (SCRiPTS). From August 2020, he is on part-time secondment to the INTERPOL as a Consultant at Cyber and New Technology Innovation. Prior to joining NTU, he has been a Professor of the Tsinghua University, PR China (2002–2010) and a faculty member of the National University of Singapore and the University of London since 1990. He was a Visiting Scientist at the Isaac Newton Institute, Cambridge University, and a Visiting Professor at the European Institute for Systems Security. In 1998, he received the Singapore Foundation Award from the Japanese Chamber of Commerce and Industry in recognition of his research and development achievement in information security in Singapore. His research interests include Distributed Systems, Intelligent Systems, IoT Security, Distributed Protocols for Blockchain, Homeland Security and Cybersecurity.

Wei Zhou received the Ph.D. degree from the University of Chinese Academy of Sciences. He is currently a Full Professor with the Software School, Yunnan University. His current research interests include the distributed data intensive computing and bio-informatics. He is currently a Fellow of the China Communications Society, a member of the Yunnan Communications Institute, and a member of the Bioinformatics Group of the Chinese Computer Society. He won the Wu Daguan Outstanding Teacher Award of Yunnan University in 2016, and was selected into the Youth Talent Program of Yunnan University in 2017. Hosted a number of National Natural Science Foundation projects.

Sixing Wu received his Ph.D. degree in computer software and theory from Peking University, China, in 2022. He is currently a Lecture with the Software School, Yunnan University. His research interests include artificial intelligence, natural language processing, and open-domain dialogue response generation. He published 20+ academic papers in international conferences and journals such as ACL, EMNLP, IJCAI, COLING, TASLP, WWW, WSDM, and ICASSP. He also served as a PC member (reviewer) of several international conferences, such as ACL (and ARR), EMNLP, AAAI, etc.

Jun Zhao (Member, IEEE) is currently an Assistant Professor in the College of Computing and Data Science at Nanyang Technological University (NTU), Singapore. He received a Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University (CMU), Pittsburgh, PA, USA, in May 2015, and a bachelor's degree in Information Engineering from Shanghai Jiao Tong University, China, in June 2010. One of his papers was a finalist for the best student paper award in IEEE International Symposium on Information Theory (ISIT) 2014. His research interests include A.I. and data science, security and privacy, control and learning in communications and networks.

Dongting Hu is a Ph.D student in data science at the School of Mathematics and Statistics, University of Melbourne. Prior to commencing PhD, he received his M.S. degree in Data Science from University of Melbourne in 2021. His research interests lie in 3D computer vision, including depth estimation and neural rendering.



Mingming Gong is a Senior Lecturer in Data Science at the School of Mathematics and Statistics, University of Melbourne, and an affiliated Associate Professor of Machine Learning at the Mohamed bin Zayed University of Artificial Intelligence. His research interests include causal learning and inference, deep learning, and computer vision. He has authored and co-authored more than 50 research papers in top venues such as ICML, NeurIPS, ICLR, CVPR, TPAMI, IJCV, TIP, TNNLS and served as a meta-reviewer for NeurIPS, ICML, ICLR, IJCAI, AAAI, TMLR, etc. He has won the Australian Artificial Intelligence Emerging Researcher Award and the Australian Research Council Early Career Researcher Award.