Aha! Adaptive History-driven Attack for Decision-based Black-box Models

Jie Li[1], Rongrong Ji[1,2,4*], Peixian Chen[1,6], Baochang Zhang[3], Xiaopeng Hong[5],
Ruixin Zhang[6], Shaoxin Li[6], Jilin Li[6], Feiyue Huang[6], Yongjian Wu[6]
[1]MAC Lab, School of Informatics, Xiamen University, [2]Peng Cheng Lab, [3]Beihang University,
[4]Institute of Artificial Intelligence, Xiamen University, [5]Xi'an Jiaotong University, [6]Youtu Lab, Tencent

Abstract

The decision-based black-box attack means to craft adversarial examples with only the top-1 label of the victim model available. A common practice is to start from a large perturbation and then iteratively reduce it with a deterministic direction and a random one while keeping it adversarial. The limited information obtained from each query and inefficient direction sampling impede attack efficiency, making it hard to obtain a small enough perturbation within a limited number of queries. To tackle this problem, we propose a novel attack method termed Adaptive History-driven Attack (AHA) which gathers information from all historical queries as the prior for current sampling. Moreover, to balance between the deterministic direction and the random one, we dynamically adjust the coefficient according to the ratio of the actual magnitude reduction to the expected one. Such a strategy improves the success rate of queries during optimization, letting adversarial examples move swiftly along the decision boundary. Our method can also integrate with subspace optimization like dimension reduction to further improve efficiency. Extensive experiments on both ImageNet and CelebA datasets demonstrate that our method achieves at least 24.3% lower magnitude of perturbation on average with the same number of queries. Finally, we prove the practical potential of our method by evaluating it on popular defense methods and a real-world system provided by MEGVII Face++.

1. Introduction

With the rapid development and the dominant performance, deep neural networks (DNNs) have been successfully deployed to improve productivity in many fields, *e.g.*, the voice assistant in smart speakers, image recognition APIs on the cloud, and automatic pilot in vehicles. Though many effort have been put into explaining the DNNs [1, 18, 19, 43], DNNs are still far from full controllable and have been proven to be vulnerable to carefully crafted imperceptible perturbations, *i.e.*, adversarial perturbations [41], which poses threats to the application of DNNs in security scenarios.

Therefore, many methods have been proposed to evaluate the robustness of the DNNs under different settings [13, 4, 21]. Among all the settings, the black-box setting is the most practical but challenging one since only the corresponding outputs are available. Some attack methods [29, 38, 37] craft adversarial examples on white-box models and transfer them to the victim model. These transfer-based methods consume fewer resources but can not guarantee a high attack success rate. Some adversaries turn to query the model repeatedly. Depending on the form of the outputs, query-based black-box attack methods can be further divided into the score-based attack and decision-based attack. Outputs of the former one are usually continuous and floating numbers (*e.g.*, class probabilities) responding to the change of input rapidly, which is able to guide the perturbation generation step by step. The decision-based attack setting is more challenging where the adversary can only fetch the result whether the input belongs to the same class as the target sample or not. Such a setting usually is correlated to a target attack whose goal is to craft an adversarial example classified as a target one.

The most classic decision-based attack, Boundary Attack [2], starts from an adversarial example and search along two directions: the source direction towards the source image directly for reducing perturbation and the spherical direction randomly sampled from the normal distribution for exploring. However, this method mainly depends on random sampling without utilizing information from prior queries efficiently, resulting in an enormous number of queries. Many methods have been proposed to improve it. Biased Boundary Attack [3] introduces three biases to improve the efficiency of direction sampling. Evolutionary Attack [10] reduces the solution space and models the local geometry via successful queries with (1+1)-CMA-ES optimization. However, without taking full advantage of all information from all queries, these methods still re-

*R. Ji (rrji@xmu.edu.cn) is the corresponding author.

quire a large number of queries to reduce the magnitude of the perturbations. Moreover, the trade-off of the two directions also impacts a lot. We argue that large coefficient for the direction reducing the perturbation brings more queries crossing the boundary and then failing, but large coefficient for the exploring direction will increase the number of queries. Existing methods adjust the corresponding coefficients based on whether the query is adversarial. Such a binary value gives a coarse guide thereby leaving coefficient adjustment inflexible.

In this paper, towards obtaining perturbations with smaller magnitude under fewer queries, we propose the Adaptive History-driven Attack (AHA) which makes use of information from all queries with an adaptive coefficient adjustment strategy. Following the boundary attack, AHA starts from a large perturbation, and then reduces it iteratively with a determinate direction (*i.e.*, source direction) and a random direction. Instead of randomly sampling from a standard normal distribution for the random direction, we gather information from historical queries and apply it as the prior for current sampling. Such a method is simple yet efficient without extra computation cost added. To balance between the source direction and the direction driven by historical queries, considering that the purpose of coefficient adjusting is to reduce the magnitude of the perturbations as much as possible, we dynamically adjust the coefficient based on the ratio of the actual reduction on perturbation's magnitude to the expected one. This strategy reduces the chance of getting stuck into the decision boundary. Besides, the optimization method is orthogonal to the existing subspace method like dimension reduction. These methods can be integrated to further improve performance. We conduct extensive experiments on various models including a real-world online system to demonstrate the efficiency of the proposed AHA. We conclude our contributions as:

- We propose a simple yet efficient decision-based attack method, termed Adaptive History-driven Attack (AHA), which utilizes information of both successful and failed historical queries as the prior for current sampling without complex optimization and extra computation cost added.

- To balance between two directions during the optimization process, we design a novel strategy to adjust the coefficient dynamically. Instead of on how often the optimization successes, the coefficient is adjusted based on the degree of the actual reduction on the magnitude compared with the expected one, which increases the probability of finding valid queries.

- Finally, we evaluate AHA on models for natural images and human faces. The perturbations generated by AHA are smaller than the state-of-the-art method with the same number of queries. Furthermore, the effectiveness of AHA on the real-world system, *i.e.*, face verification API from MEGVII Face++, is also verified with 24.9% smaller perturbations than baseline.

2. Related Work

Score-based Attack. Due to the fact that outputs fetched are continuous and floating numbers, every small change in input will give an instant response. It is natural to estimate the value of the gradient, and then perform the white-box attack. ZOO [6] estimates the value of the gradient using the finite-difference method. With such a dimension-wise way, it takes $2d$ queries each time to estimate the gradient. Instead of the finite-difference method, NES [20] utilizes the natural evolutionary strategy with random vectors sampled from the normal distribution to reduce the required number of queries. Bandits$_{TD}$ [21] method further introduces a data-dependent and a time-dependent prior to improve the efficiency of gradient estimation. Instead of gradient estimation, some methods adopt random search strategies. SimBA [15] crafts a set of orthonormal vectors first, then randomly picks one from the set and adds or subtracts it if the objective function decreases. PPBA [24] reduces the dimension with low-frequency constraint and performs random walk optimization on the low dimension space.

Decision-based Attack. Unlike the score-based attack, the outputs of models in the decision-based attack are only the labels. Such a hard-label setting increases the difficulty since the tiny change of the input may not reflect on the output. Opt-Attack [7] re-formulates this problem as a continuous optimization problem *w.r.t* the direction and distance to the decision boundary, and performs gradient estimation on it. However, this method is ineffective since the distance calculation and gradient estimation on the large dimension will consume an enormous number of queries. HSJA [5] directly performs the gradient estimation on the decision boundary with binary outputs. And QEBA [23] further improves the performance by adopted subspace on HSJA. However, hundreds of queries are needed for one time gradient estimation, which makes these methods still inefficient. Boundary attack [2] starts from a large adversarial perturbation and simultaneously reduces it with source direction and spherical direction. It bases on random walk optimization and rejects updating when not adversarial. This method is simple but the usage of standard normal distribution impedes efficiency. Biased Boundary Attack [3] introduces some biases to improve the boundary attack. Instead of the normal distribution, the Perlin distribution is adopted for low-frequency constraint, and the difference between the adversarial example and the source image is used as the weight for pixels. This method reduces the solution space, but it is not enough. Evolutionary method [10] replaces the normal distribution with a custom variance. The

variance is updated with (1+1)-CMA-ES when sampling is successful to model the weight for each pixel. However, the variance is sign-independent, which makes the sampling unstable. CAB [32] uses the square of the difference between the adversarial example and the source image as variance and accumulates the directions when failed for the mean. SurFree [28] tries to move along diverse directions guided by the geometrical properties of the decision boundary. Though these optimization methods are well-designed, they are complex and still not efficient enough.

3. Proposed Method

Throughout this paper, we focus on reducing the magnitude of the perturbations within limited queries under the decision-based target black-box attack setting. Based on the boundary attack, we perform the random walk optimization with historical queries as prior as detailed in Sec. 3.2. Coefficients of the two search directions play an important role as to move towards the source input more or to learn from the history more. To balance them, a novel adaptive adjusting strategy is proposed in Sec. 3.3. The optimization method can be further improved with the help of the subspace optimization as in Sec. 3.4. In the rest of this section, we will first introduce the preliminary knowledge and then give a thorough description of our proposed method.

3.1. Preliminaries

Suppose we have a source input sample $\boldsymbol{x_s}$, a target one $\boldsymbol{x_t}$, and a deep neural network based function $f(\boldsymbol{x_1}, \boldsymbol{x_2})\colon \mathcal{X} \times \mathcal{X} \mapsto \mathcal{Y}$ to determine whether the two input sample belong to the same class, where $\mathcal{X} = [0,1]^d$ is the space for images with d-dimension and $\mathcal{Y} = \{0, 1\}$ (1 denotes the two inputs share the same class). The aim of decision-based target attack is to find an adversarial example $\boldsymbol{x'}$ that close to the source input $\boldsymbol{x_s}$ as far as possible while keep $f(\boldsymbol{x'}, \boldsymbol{x_t}) = 1$. We have a objective function as:

$$\min_{\boldsymbol{x'}} L(\boldsymbol{x'}) = \mathcal{D}(\boldsymbol{x'}, \boldsymbol{x_s}) + \lambda \cdot (1 - f(\boldsymbol{x'}, \boldsymbol{x_t})), \quad (1)$$

where λ is a very large number to make sure $L(\boldsymbol{x'})$ large enough when the adversarial objective is unsatisfied, and $\mathcal{D}(\cdot, \cdot)$ is the distance function. In this paper, we select L_2 norm as the distance function, *i.e.*, $\mathcal{D}(\boldsymbol{x'}, \boldsymbol{x_s}) = \|\boldsymbol{x'} - \boldsymbol{x_s}\|_2$.

Following the common practice [2, 10], we start from an adversarial sample (with the same class as the target one), *e.g.*, the target sample x_t, and then move it close to the original sample x_s as much as possible iteratively with a constraint on the number of queries. A common update formula can be represented as:

$$\boldsymbol{x_{k+1}} = \boldsymbol{x_k} + \alpha \cdot \frac{\boldsymbol{x_s} - \boldsymbol{x_k}}{\|\boldsymbol{x_s} - \boldsymbol{x_k}\|_2} + \beta \cdot \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|_2}, \boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \quad (2)$$

where $\boldsymbol{x_k}$ is the adversarial example at the k-th steps and $\boldsymbol{x_s}$ is the source input, $(\boldsymbol{x_s} - \boldsymbol{x_k})$ and $\boldsymbol{\eta}$ are the source direction and spherical direction, respectively. α and β are the corresponding coefficients. The update value can be further multiplied by the distance between the current sample and the original sample to reduce the update value iteratively for better convergence:

$$\begin{aligned} \boldsymbol{x_{k+1}} &= \boldsymbol{x_k} + (\alpha \cdot \frac{\boldsymbol{x_s} - \boldsymbol{x_k}}{\|\boldsymbol{x_s} - \boldsymbol{x_k}\|_2} + \beta \cdot \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|_2}) \cdot \|\boldsymbol{x_s} - \boldsymbol{x_k}\|_2 \\ &= \boldsymbol{x_k} + \alpha \cdot (\boldsymbol{x_s} - \boldsymbol{x_k}) + \beta \cdot \frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|_2} \cdot \|\boldsymbol{x_s} - \boldsymbol{x_k}\|_2. \end{aligned} \quad (3)$$

Here a query is called a successful query if the x_{k+1} is still adversarial, and it will be called a failed one otherwise. Note that in some previous works, $\boldsymbol{x_{k+1}}$ will be accepted only if $L(\boldsymbol{x_{k+1}})$ is less than or equal to $L(\boldsymbol{x_k})$ as the common practice in the random walk optimization. To avoid falling into the local optimum, we only reject failed samples. If the $\boldsymbol{x_{k+1}}$ is rejected, then we set $\boldsymbol{x_{k+1}} = \boldsymbol{x_k}$ for the ease of representation.

3.2. Historical Prior Based Optimization

Reexamining Eq. 3 carefully, the only uncertainty lies in the random direction which influences the efficiency of the optimization method greatly. Previous methods also made efforts on it. The key question is that how to make the random direction sampled more efficiently. Some previous works have proven that the decision boundaries of deep neural networks have a quite small curvature in the vicinity of data samples [12], which indicates that the decision boundary at the neighborhood of adversarial example can be approximated locally with a hyperplane [25, 30]. Since the boundary is flat, we can confidently assume the current random direction and one of the last iteration or even more early iterations are consistent to some degree. Also, there are some previous works showing that historical information is helpful for current sampling [10, 21, 24, 32]. However, we argue that the exist methods utilizing historical prior is complicate and not thoroughgoing. For example, Evolutionary Attack [10] utilizes only successful queries while CAB [32] utilizes only failed queries, and Bandits$_{TD}$ [21] utilizes all queries but does not distinguish successful and failed queries well.

The flat boundaries and the lack of making the best of historical information motivate us to use a more simple but efficient way to guide the random direction. As [2, 10], we treat the historical prior as a custom gaussian distribution. Though the variance of the distribution can model the importance of each pixel naturally, modifying the variance also introduces instability since the variance is sign-independent which can not guide the direction well. Instead of variance, we focus on the mean $\boldsymbol{\mu}$ of the distribution and embed the information of both successful historical queries and failed ones in it. For the successful query x_{k+1} where

$f(\boldsymbol{x_{k+1}}, \boldsymbol{x_t}) = 1$, as talked above, the next direction will succeed with a high probability if they share similar direction since the decision boundary is flat. While for the failed query, as stated in [32], it also contains information about the decision boundary since the failed query crosses through the boundary. Similar to [32], its opposite direction is considered. In particular, we update the mean $\boldsymbol{\mu}$ with:

$$\boldsymbol{\mu} = \begin{cases} (1-\gamma)\cdot\boldsymbol{\mu} + \gamma\cdot\boldsymbol{\eta}, & f(\boldsymbol{x_{k+1}}, \boldsymbol{x_t}) = 1 \\ (1-\gamma)\cdot\boldsymbol{\mu} - \gamma\cdot\boldsymbol{\eta}, & f(\boldsymbol{x_{k+1}}, \boldsymbol{x_t}) = 0 \end{cases}, \quad (4)$$

where $\gamma \in (0, 1)$ is a coefficient to control how fast to forget the older information since the geometry properties of the decision boundary will change along with the shift of the adversarial example. Since the direction is driven by the historical queries, we named it the history-driven direction.

3.3. Coefficient Adaptive Adjusting

Another significant issue is how to balance between the source direction and the history-driven direction. A large coefficient α for source direction is helpful to reduce the magnitude of the perturbations quickly, while it raises the probability to hit the decision boundary and thereby leading to a failed query. On the contrary, a small α allows the optimization method to explore the decision boundary much, while it will decelerate the progress of approaching the source input and increase the number of queries. Therefore, an adaptive adjusting strategy is needed.

Previous methods also noticed such a problem and put effort into it. The boundary attack method samples more points with orthogonal directions to test the success rate, and reduce the coefficient if the success rate is much lower or increase it if the success rate is close to 50% or higher. The Biased Boundary Attack uses large coefficients at the beginning and decreases it when the number of failed queries increased. The coefficients are reset when a successful query occurs. As a evolution strategy, the evolutionary method utilizes a traditional method for hyper-parameter control in evolution strategies termed 1/5th success rule [31] to update the coefficient by multiplying $\exp(P_{\text{success}} - 1/5)$, where P_{success} denotes the success rate of several past iterations. Note that the existing methods are all based on the success rate, and every query can only offer coarse binary feedback (*i.e.*, successful or not).

Considering that the purpose of coefficient adjusting is to reduce the magnitude of the perturbations as much as possible, it motivates us to consider how the magnitude of perturbations is reduced. Therefore, instead of considering the success rate, we adjust the coefficient α based on the ratio of the actual magnitude reduction to the expected one. The actual magnitude reduction can be calculated straightly with the difference between the two distance as:

$$\begin{aligned} \mathcal{R}_{\text{actual}} &= \mathcal{D}(\boldsymbol{x_k}, \boldsymbol{x_s}) - \mathcal{D}(\boldsymbol{x_{k+1}}, \boldsymbol{x_s}) \\ &= \|\boldsymbol{x_k} - \boldsymbol{x_s}\|_2 - \|\boldsymbol{x_{k+1}} - \boldsymbol{x_s}\|_2. \end{aligned} \quad (5)$$

For the expected reduction, we then view the length of the projection of update part on the source direction as the expected reduction:

$$\begin{aligned} &\mathcal{R}_{\text{expected}} \\ =&\big(\alpha\cdot(\boldsymbol{x_s} - \boldsymbol{x_k}) + \beta\cdot\frac{\boldsymbol{\eta}}{\|\boldsymbol{\eta}\|_2}\|\boldsymbol{x_s} - \boldsymbol{x_k}\|_2\big)^{\mathrm{T}}\frac{\boldsymbol{x_s} - \boldsymbol{x_k}}{\|\boldsymbol{x_s} - \boldsymbol{x_k}\|_2} \\ =&\alpha\cdot\|\boldsymbol{x_s} - \boldsymbol{x_k}\|_2 + \beta\cdot\frac{\boldsymbol{\eta}^{\mathrm{T}}(\boldsymbol{x_s} - \boldsymbol{x_k})}{\|\boldsymbol{\eta}\|_2}. \end{aligned} \quad (6)$$

Depending on the value of $\mathcal{R}_{\text{actual}}$, there are three situations. When $\mathcal{R}_{\text{actual}} = 0$, we know that a failed query occurs and we should explore more by reducing α. When $\mathcal{R}_{\text{actual}} > 0$, we are moving towards the source sample and now $0 < \mathcal{R}_{\text{actual}} \leq \mathcal{R}_{\text{expected}}$. So the larger the $\mathcal{R}_{\text{actual}}/\mathcal{R}_{\text{expected}}$ is, the more important the source direction is, and the larger the corresponding coefficient, *i.e.*, α, should be. When $\mathcal{R}_{\text{actual}} < 0$, it is most likely that the optimization method gets stuck in local optimum, and should turn away from the source sample to escape it. In such a situation, $\mathcal{R}_{\text{actual}} \leq \mathcal{R}_{\text{expected}} < 0$. We calculate the ratio as $\mathcal{R}_{\text{expected}}/\mathcal{R}_{\text{actual}}$ for a value belonging to $[0, 1]$. Similarly, we prefer small α for a small ratio to help escape from the local optimum and large α for a large ratio to prevent the optimization method from keeping moving away from the source sample. Based on above discussion, we unify the ratio of $\mathcal{R}_{\text{actual}}$ and $\mathcal{R}_{\text{expected}}$ as:

$$r = \frac{\min\big(\text{abs}(\mathcal{R}_{\text{actual}}), \text{abs}(\mathcal{R}_{\text{expected}})\big)}{\max\big(\text{abs}(\mathcal{R}_{\text{actual}}), \text{abs}(\mathcal{R}_{\text{expected}})\big)}. \quad (7)$$

Note that the ratio $r \in [0, 1]$, and we should increase the coefficient α when r is large and decrease α otherwise. And for the failed query where $\boldsymbol{x_{k+1}} = \boldsymbol{x_k}$, the $\mathcal{R}_{\text{actual}}$ and the ratio r are equal to zero. Therefore, we can find that the success rate based method is just a particular case of our method when r is mapped as $sign(r)$. Note that too small α may result in forever exploring. So we reset the value of α when it is less than a threshold. Finally, we update the coefficient α as:

$$\alpha = \alpha\cdot h(\bar{r}), \alpha = \begin{cases} \alpha, & \alpha > \alpha_{\text{threshold}} \\ \alpha_{\text{initial}}, & \text{otherwise} \end{cases}, \quad (8)$$

where $\bar{r}$ is the mean of ratios r's over several past iterations, $h(\cdot)\colon [0, 1] \mapsto \mathbb{R}^+$ is a function that maps the ratio to a suitable value, $\alpha_{\text{threshold}}$ is the value preventing too small α, and α_{initial} is the initial value for α. $h(\cdot)$ should be monotonically increasing with $0 < h(0) < 1$ and $h(1) > 1$. In this paper, we experimentally select $h(\cdot)$ as $h(\bar{r}) = (\bar{r} + 0.8)^2$.

Algorithm 1: Adaptive History-driven Attack

Input: Victim model $f(\cdot,\cdot)$, source image $\boldsymbol{x_s}$, target image $\boldsymbol{x_t}$, maximum number of queries Q, initial direction coefficients α and β, coefficients γ, and interval i.

Output: Adversarial example $\boldsymbol{x'}$.

Initialize $q \leftarrow 0, q_{\text{last}} \leftarrow 0, \boldsymbol{x'} \leftarrow \boldsymbol{x_t}, \boldsymbol{\mu} \leftarrow \mathbf{0}, \bar{r} \leftarrow 0$
while $q < Q$ **do**
  Sample $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I})$
  Upscale $\boldsymbol{\eta}$ to the same dimension of x' for $\boldsymbol{\eta'}$
  $\boldsymbol{x}_{\text{temp}} \leftarrow \boldsymbol{x'} + \alpha \cdot (\boldsymbol{x_s} - \boldsymbol{x'}) + \beta \cdot \frac{\boldsymbol{\eta'}}{\|\boldsymbol{\eta'}\|_2} \cdot \|\boldsymbol{x_s} - \boldsymbol{x'}\|_2$
  if $f(\boldsymbol{x}_{temp}, \boldsymbol{x'}) = 1$ **then**
    $\boldsymbol{x'} \leftarrow \boldsymbol{x}_{\text{temp}}$
    $\boldsymbol{\mu} \leftarrow (1-\gamma) \cdot \boldsymbol{\mu} + \gamma \cdot \boldsymbol{\eta}$
  else
    $\boldsymbol{\mu} \leftarrow (1-\gamma) \cdot \boldsymbol{\mu} - \gamma \cdot \boldsymbol{\eta}$
  Calculate r according to Eq. 7
  // Calculate the running mean
  $\bar{r} \leftarrow \frac{q - q_{\text{last}}}{q - q_{\text{last}} + 1} \cdot \bar{r} + \frac{1}{q - q_{\text{last}} + 1} \cdot r$
  if $q - q_{last} = i$ **then**
    Update α by Eq. 8
    $q_{\text{last}} \leftarrow q$
    $\bar{r} \leftarrow 0$
  $q \leftarrow q + 1$
end
return $\boldsymbol{x'}$

3.4. Subspace Sampling

The large solution space is most blamed in black-box attacks, and many methods including dimension reduction [21, 10, 23] and low-frequency constraints [3, 24] have been proposed to reduce it and these methods do accelerate the attack process. These subspace optimization methods are orthogonal to our method and can be integrated to further improve the performance. Considering that dimension reduction with bilinear interpolation is more simple and fast compared with the other methods, we sample the direction $\boldsymbol{\eta}$ in the low dimensional space and then upscale it with bilinear interpolation to original input space. Following [23], the dimension of low space will be $1/16$ of the original one.

We refer to the method combining the three parts mentioned above as Adaptive History-driven Attack (AHA), and conclude the details in Alg. 1.

4. Experiments

4.1. Experimental Setups

Datasets and Victim Models. We mainly evaluate the effectiveness on the natural image dataset ImageNet [8] and human face dataset CelebA [26]. For the ImageNet dataset, we select the widely used pre-trained models including VGG-16 [33], ResNet50 [17], and Inception-V3 [35] as the victim models. We randomly select 100 pairs of images from the validation set for evaluation. The images in each pair are from different classes and are classified correctly by all three models. The input image size is $224 \times 224 \times 3$ for VGG-16 and ResNet50, and $299 \times 299 \times 3$ for Inception-V3, respectively. For the CelebA dataset, we evaluate the attack methods on state-of-the-art face recognition models, *i.e.*, CosFace [36] and ArcFace [9]. Both models are trained on MS1M dataset [16] with Inception-ResNet-152 [34] as backbone[1]. Similar to the ImageNet dataset, we also randomly select 100 pairs of faces from 200 different people that are distinguished well by the two models. These face images are pre-processed by MTCNN [42] with size of $112 \times 112 \times 3$. For defense methods, we perform attacks on 100 images randomly sampled from the CIFAR-10 dataset [22] with the Wide ResNets [40] as the target model. For the online model, we test the robustness of the face verification API proposed by Face++[2]. We choose 10 pairs of face images randomly from the CelebA with the same settings as the offline face verification experiments.

Evaluation Metrics. To judge how efficient an attack method is, we mainly check the mean L_2-norm of the final adversarial perturbations under the same number of queries since the adversarial example is guaranteed to be adversarial. The smaller the L_2-norm is, the more efficient the attack method is. To show how fast the optimization method can find a small perturbation, we depict the curve of the mean L_2-norm versus the number of queries. For quantitative comparison, we calculate the area under the curve (AUC), where the lower value denotes better performance. The attack success rate (ASR) is also a common metric for the adversarial attack. Considering that dimensions of the input image and degree of difficulty for different tasks are different, we define a successful adversarial example with dimension d as the one whose L_2-norm of the perturbation is less than $\sqrt{0.001 \cdot d}$ for the ImageNet dataset and $\sqrt{0.0001 \cdot d}$ for the CelebA dataset. We report the attack success rate on the final adversarial examples.

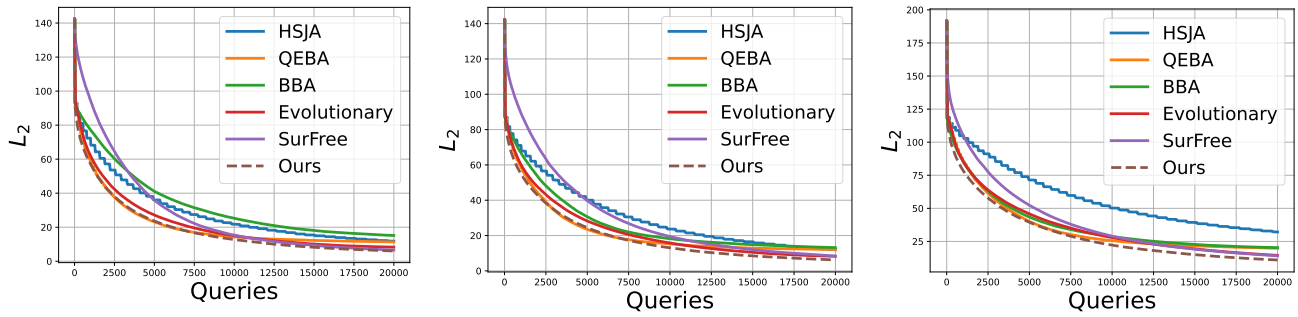
Compared Methods and Hyper-parameters Settings. We mainly compare our proposed AHA with four popular decision-based attack methods, *i.e.*, HSJA [5], QEBA [23], Biased Boundary Attack (BBA) [3], Evolutionary Attack [10], and SurFree [28]. For all the baselines, we use the source code kindly provided by the authors and the default parameters announced in their papers. Note that for a fair comparison, we do not utilize any extra surrogate model for all methods, thereby no surrogate model bias for the BBA. We believe all methods can benefit from the extra surrogate

[1]These models are trained following `https://github.com/ZhaoJ9014/face.evoLVe.PyTorch`

[2]`https://www.faceplusplus.com/face-comparing/`

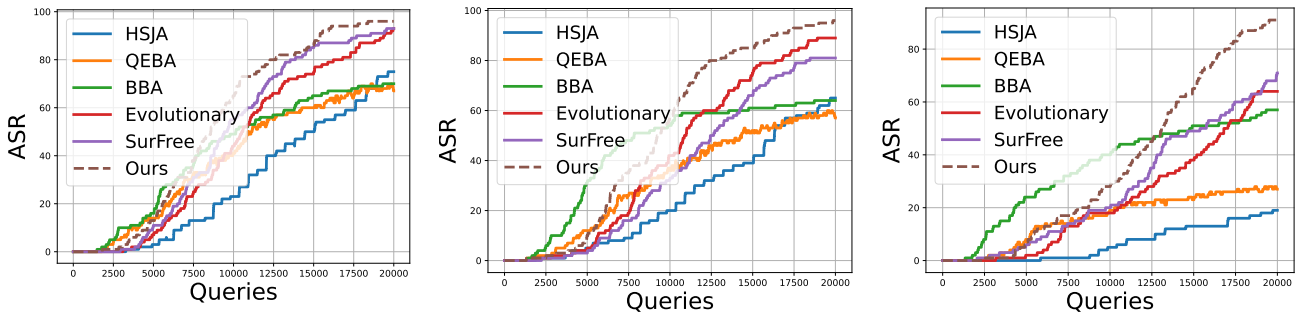
Methods	VGG-16			ResNet50			Inception-V3		
	mean L_2	AUC	ASR	mean L_2	AUC	ASR	mean L_2	AUC	ASR
HSJA	11.833	580609.2	75%	12.009	613671.2	65%	32.145	1161506.0	19%
QEBA	11.350	430284.8	67%	12.074	441363.8	57%	19.807	710941.6	27%
BBA	15.141	660683.2	70%	13.183	523421.3	64%	20.328	746100.3	57%
Evolutionary	8.305	445020.9	93%	8.238	451534.7	89%	14.494	725262.1	64%
SurFree	6.579	564050.6	93%	8.451	607074.5	81%	14.081	810286.5	71%
Ours	**6.013**	**386623.6**	**96%**	**6.203**	**389357.9**	**96%**	**10.976**	**616167.4**	**91%**

Table 1. Results of our proposed AHA and baselines on the ImageNet dataset. Note that the AUC denotes the area under the curve of L_2-norm versus the number of queries and ASR denotes the attack success rate of the final adversarial example.



(a) VGG-16 (b) ResNet50 (c) Inception-V3

Figure 1. The curves of mean L_2-norm versus the number of queries for the ImageNet dataset. (Lower is better)



(a) VGG-16 (b) ResNet50 (c) Inception-V3

Figure 2. The curves of attack success rate versus the number of queries for the ImageNet dataset. (Higher is better)

model. And we select QEBA-S for QEBA due to its best performance as shown in their paper. For hyper-parameters in our AHA, we set both the α and β to 0.01 initially following [2, 10]. We also set γ as 0.01 and the interval i as 30. Following [5, 23], we set the maximum number of queries as $20,000$ for ImageNet and CelabA models. The maximum number of queries for the defense methods is set as $50,000$. Considering that querying real-world online systems is costly and time-consuming, we limit the maximum number of queries for the online system to $5,000$.

4.2. Results on ImageNet and CelabA Models

We evaluate the performance of our proposed AHA along with the baselines in Tab. 1 for the ImageNet dataset and in Tab. 2 for the CelebA dataset, respectively. As in the Tab. 1 for the ImageNet dataset, we can find that our proposed AHA achieves the best performance on all three widely-used models. Particularly, the mean L_2-norm of perturbations found by our method is 27.6%, 24.7%, and 24.3% less than the second-best method, *i.e.*, the Evolutionary attack, on the VGG-16, ResNet50, and Inception-V3 respectively. The value of AUC represents how fast the optimization method can reduce the magnitude of the perturbation. It is worth noting that though the final mean L_2 is not small enough, the AUCs of the QEBA are less than other baselines. However, our proposed method still achieves a smaller AUC compared with the QEBA method with 10.1%, 11.8%, and 13.3% less. This can also be concluded from Fig. 1 where the curves of our method are at the bottom. The attack success rate represents how often the optimization method can find a valid adversarial example. From Tab. 1, we also find our proposed AHA achieves

Methods	CosFace			ArcFace		
	mean L_2	AUC	ASR	mean L_2	AUC	ASR
HSJA	3.517	169539.4	32%	2.506	131163.5	46%
QEBA	4.586	130791.0	2%	3.998	112046.8	6%
BBA	3.101	119237.1	30%	2.540	**87663.5**	43%
Evolutionary	2.960	137222.0	21%	2.604	120543.8	26%
Ours	**1.909**	**109084.4**	**58%**	**1.436**	93046.4	**78%**

Table 2. Results on the CelebA dataset.

the highest attack success rate among the attack methods, which demonstrates most of the adversarial examples found by AHA are valid. For example, for Inception-V3, our method achieves a 91% attack success rate, which is 27% higher than the state-of-the-art method. To give a more direct comparison, we draw the curves of attack success rates versus the number of queries in Fig. 2. At the beginning, the BBA method works well. And then after nearly $10,000$ queries, with enough historical accumulation, the attack success rate of our proposed AHA increases steeply and is higher than baselines. We also test the performance on the CelebA dataset in the Tab. 2. From the Tab. 2, we can conclude that AHA is not just efficient on the natural image dataset but also efficient on the human face dataset. Though the AUC on the ArcFace model of our AHA is larger than the one of the BBA method, our proposed method still achieves nearly the best performance over the three metrics. Also, we find that the CosFace model is more robust than the ArcFace against such a hard-label black-box attack. It is also interesting that the AUC of the QEBA method is lower than the HSJA while the mean L_2-norm and attack success rate of QEBA show no superiority for both datasets. Note that the QEBA can be viewed as the HSJA combined with the subspace optimization. We conclude that the subspace optimization helps faster convergence but leads to suboptimal results. So an adaptive subspace optimization will be helpful and we leave it as future work.

4.3. Attack Against Defense Methods

Along with the development of the attack methods, many studies proposed defense methods to protect their models. To verify the effectiveness of our AHA method under this setting, we evaluate the performance of AHA along with some baselines on 100 images randomly sampled from the CIFAR-10 test set with 10 images per class. For each image, we randomly select another image from the 100 images as the target. And we use the Wide ResNets architecture with 28 layers and a width multiplier of 10 (denoted as WRN-28-10) as the target model. For defense methods, we utilize widely used bit-depth reduction [39], JPEG compression [11] and adversarial training [27]. The resutls are presented in Tab. 3 where None denotes the vanilla model without defense. We can conclude that our proposed AHA still works well for all defense methods. For example, for the adversarial training model, the mean L_2-norm of our perturbations is 1.8825, which is 28.5% less than HSJA and 36.0% less than Evolutionary.

		None	Bit Depth	JPEG	Adv. Training
L_2 norm	HSJA	**0.2677**	6.1930	5.1632	2.6342
	QEBA	0.7554	2.8148	2.4491	2.3248
	BBA	0.5367	1.2898	2.7022	2.5145
	Evolutionary	0.5491	1.0516	2.2334	2.9410
	Ours	0.3185	**1.0207**	**2.2166**	**1.8825**
AUC	HSJA	**23480.4**	356472.4	299518.1	155950.6
	QEBA	42989.6	180617.4	191778.3	**127246.0**
	BBA	37163.5	163392.1	173135.0	293148.1
	Evolutionary	40127.3	143257.6	179670.6	173135.0
	Ours	36881.7	**141574.3**	**171625.6**	143037.2

Table 3. Results for different defense methods. None denotes vanilla model without defense.

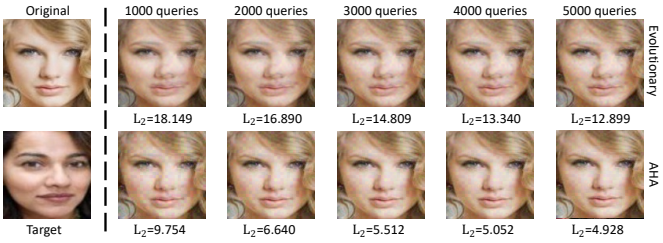


Figure 3. The example of AHA and Evolutionary on Face++ API. (Best view zoomed in)

4.4. Results on Online System

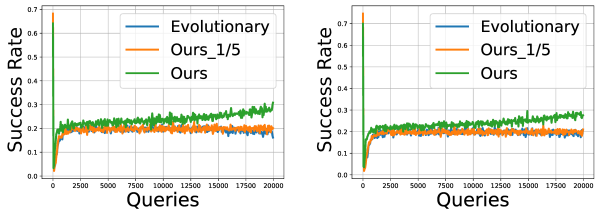
Turning attention to the robustness of real-world systems, we finally test the effectiveness of our method against the online system, *i.e.*, face verification API provided by MEGVII Face++. This API allows users to upload two face images and then returns a similarity score of them along with some thresholds. And two faces are recognized as the same person to some degree when the similarity score is higher than the corresponding threshold. Here we use the highest threshold returned and the attack method can only obtain whether the two faces belong to the same identity or not instead of the similarity score. Considering the querying is costly, we only attack this API using the Evolutionary and AHA on 10 pairs randomly selected and set the maximum query time to $5,000$. The final mean L_2-norm of the perturbations for the Evolutionary and ours are 14.544 and 10.910, respectively. We also give visual examples in Fig. 3 showing adversarial face images with different queries. Such results demonstrate that AHA is practical for real-world systems.

4.5. Ablation Study

In this subsection, we will examine how much the coefficient adaptive adjusting strategy and the subspace optimization contribute to the final performance. For the coefficient adjusting strategy, we compare the 1/5th Success Rule with our strategy on AHA. To save computing resources, we select some simple formulas for $h(\cdot)$ as listed in Tab. 4. We first find that with the 1/5th Success Rule, AHA is still more

Strategies	ResNet50		Inception-V3	
	mean L_2	AUC	mean L_2	AUC
1/5th Success Rule	6.638	404310.2	11.507	643280.9
$h(\bar{r}) = \exp(\bar{r} - 0.2)$	**6.154**	398307.0	11.222	641313.8
$h(\bar{r}) = \bar{r} + 0.8$	6.168	393999.3	**10.751**	620673.0
$h(\bar{r}) = (\bar{r} + 0.6)^2$	20.813	652617.2	29.989	953790.4
$h(\bar{r}) = (\bar{r} + 0.7)^2$	8.329	427764.1	13.654	660240.3
$h(\bar{r}) = (\bar{r} + 0.8)^2$	6.203	**389357.9**	10.976	**616167.4**
$h(\bar{r}) = (\bar{r} + 0.9)^2$	9.643	506123.7	16.604	782229.9
$h(\bar{r}) = (\bar{r} + 0.8)^3$	6.396	400115.1	11.211	628122.4

Table 4. Results for different coefficient adjusting strategies.



(a) VGG-16 (b) ResNet50

Figure 4. The curves of success rate for queries versus the number of queries. The curves are smoothed for better visualization. Ours_1/5 denotes our historical prior with the 1/5th Success Rule. (Best view in color)

efficient than baselines like the Evolutionary. And our coefficient adjusting strategy, with proper $h(\cdot)$, can further improve the performance. Here we select $h(\bar{r}) = (\bar{r}+0.8)^2$ for its stable AUC. To further prove the effectiveness of the proposed coefficient adjusting strategy, we check how often the optimization method can find a successful query. Therefore, we depict the curves of success rate versus the number of queries in Fig. 4 for the Evolutionary, our method with 1/5th Success Rule, and our method with the proposed adjusting strategy. For better visualization, we smooth the curves by drawing the mean of an interval of 40. The success rate of our proposed adjusting strategy increases sustainably and is higher than others, which helps the optimization method move along the decision boundary swiftly.

As mentioned in Sec. 3.4, the subspace optimization method is orthogonal to our method and can be combined for better performance. Note that some of the baselines also use the subspace optimization method to improve performance. The QEBA and the Evolutionary methods use bilinear interpolation for dimensionality reduction, and the BBA uses Perlin noise for the low-frequency space constraint. Here we check the influence of two widely used subspace optimization methods, *i.e.*, bilinear interpolation for dimensionality reduction (abbreviated as DR), and low-pass filtering via DCT (similar to Perlin noise but more simple). Following [23, 14], we set scale factor as 4 for both DR and DCT. The results of our proposed AHA with or without subspace optimization can be found in the Tab. 5. We can conclude that with single subspace optimization, performance can be improved significantly, *e.g.*, mean L_2-norm

Methods	ResNet50		Inception-V3	
	mean L_2	AUC	mean L_2	AUC
Ours	10.823	516663.8	22.196	846070.4
Ours+DR	**6.203**	**389357.9**	10.976	616167.4
Ours+DCT	6.515	406349.5	**10.788**	**615063.7**
Ours+DCT+DR	16.200	556612.0	28.012	888257.9

Table 5. Results with/without subspace optimization.

reduced from 10.823 to 6.203. DR works similar to DCT. However, the time DCT consumes is approximately 4 times longer than DR, which makes DR more competitive. We also find that combining DR and DCT results in the worst performance, we guess that too many constraints limit the optimization method. Based on the above observations, we choose DR as a part of our method.

We also give a brief study on using different historial queries. We evaluated the average of L_2 norm for ResNet50 on ImageNet of two cases, *i.e.*, only using failed queries (6.950) and only using successful queries (7.465). Since the success rate for queries is lower than 30% as shown in Fig. 4 and more queries tend to be failed, the former case utilizes more informative queries and performs better than the latter one. Besides, our method using all queries (6.203) is the best among them. We can conclude that utilizing more informative queries leads to higher performance, which also supports our motivation.

5. Conclusion

In this paper, we propose a simple yet efficient decision-based black-box attack method termed Adaptive History-driven Attack (AHA), which mainly utilizes information of historical queries as a prior to improve the random walk optimization. To balance between two directions during optimization, a novel coefficient adaptive adjusting strategy is also proposed based on how the magnitude of the perturbation is reduced. We evaluate our proposed method on both the natural image dataset and the human face dataset and show a higher attack performance of AHA compared with the state-of-the-art methods. Finally, we also check the effectiveness of our proposed method against some popular defense methods and a real-world face verification system provided by Face++. Our work shows that though difficult, the decision-based black-box attack is still achievable without any complicated method. We hope this work can serve as an inspiration in designing more robust models.

Acknowledgements. This work is supported by the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No.U1705262, No. 62072386, No. 62072387, No. 62072389, No. 62002305, No.61772443, No.61802324 and No.61702136), Guangdong Basic and Applied Basic Research Foundation (No.2019B1515120049) and the Fundamental Research Funds for the central universities (No. 20720200077, No. 20720200090 and No. 20720200091).

References

[1] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[2] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018.

[3] Thomas Brunner, Frederik Diehl, Michael Truong Le, and Alois Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *IEEE International Conference on Computer Vision*, 2019.

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.

[5] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE Symposium on Security and Privacy*, 2020.

[6] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM Workshop on Artificial Intelligence and Security*, 2017.

[7] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, JinFeng Yi, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *International Conference on Learning Representations*, 2019.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[9] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[11] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

[12] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, 2016.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[14] Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

[15] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, 2019.

[16] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, 2016.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[18] Jie Hu, Liujuan Cao, Qixiang Ye, Tong Tong, ShengChuan Zhang, Ke Li, Feiyue Huang, Rongrong Ji, and Ling Shao. Architecture disentanglement for deep neural networks. In *IEEE International Conference on Computer Vision*, 2021.

[19] Jie Hu, Rongrong Ji, ShengChuan Zhang, Xiaoshuai Sun, Qixiang Ye, Chia-Wen Lin, and Qi Tian. Information competing process for learning diversified representations. In *Advances in Neural Information Processing Systems*, 2019.

[20] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, 2018.

[21] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2019.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[24] Jie Li, Rongrong Ji, Hong Liu, Jianzhuang Liu, Bineng Zhong, Cheng Deng, and Qi Tian. Projection & probability-driven black-box attack. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[25] Yujia Liu, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. A geometry-inspired decision-based attack. In *IEEE International Conference on Computer Vision*, 2019.

[26] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, 2015.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[28] Thibault Maho, Teddy Furon, and Erwan Le Merrer. Surfree: a fast surrogate-free black-box attack. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[29] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[30] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. Geoda: A geometric framework for black-box adversarial attacks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[31] Ingo Rechenberg. Evolutionsstrategien. In *Simulationsmethoden in der Medizin und Biologie*. 1978.

[32] Yucheng Shi, Yahong Han, and Qi Tian. Polishing decision-based adversarial noise with a customized sampling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[34] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence*, 2016.

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[36] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[37] Yixu Wang, Jie Li, Hong Liu, Yongjian Wu, and Rongrong Ji. Black-box dissector: Towards erasing-based hard-label model stealing attack. *arXiv preprint arXiv:2105.00623*, 2021.

[38] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[39] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Network and Distributed System Security Symposium*, 2018.

[40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

[41] Christian Szegedy Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[42] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016.

[43] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *AAAI Conference on Artificial Intelligence*, 2018.