Geometrically Adaptive Dictionary Attack on Face Recognition

Junyoung Byun, Hyojun Go, Changick Kim
Korea Advanced Institute of Science and Technology (KAIST)
{bjyoung, gohyojun15, changick}@kaist.ac.kr

Abstract

CNN-based face recognition models have brought remarkable performance improvement, but they are vulnerable to adversarial perturbations. Recent studies have shown that adversaries can fool the models even if they can only access the models' hard-label output. However, since many queries are needed to find imperceptible adversarial noise, reducing the number of queries is crucial for these attacks. In this paper, we point out two limitations of existing decision-based black-box attacks. We observe that they waste queries for background noise optimization, and they do not take advantage of adversarial perturbations generated for other images. We exploit 3D face alignment to overcome these limitations and propose a general strategy for query-efficient black-box attacks on face recognition named Geometrically Adaptive Dictionary Attack (GADA). Our core idea is to create an adversarial perturbation in the UV texture map and project it onto the face in the image. It greatly improves query efficiency by limiting the perturbation search space to the facial area and effectively recycling previous perturbations. We apply the GADA strategy to two existing attack methods and show overwhelming performance improvement in the experiments on the LFW and CPLFW datasets. Furthermore, we also present a novel attack strategy that can circumvent query similarity-based stateful detection that identifies the process of query-based black-box attacks.

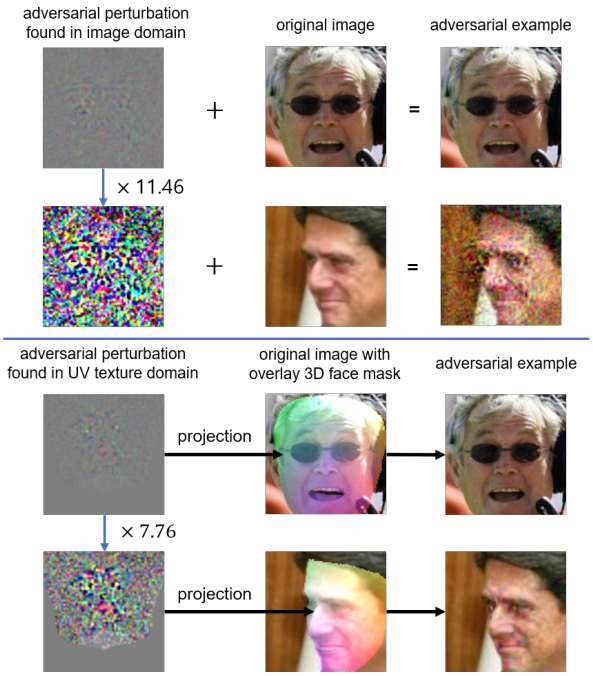

Figure 1: Illustrations of our intuitions. Finding adversarial perturbations in the UV texture map with 3D face alignment is beneficial for both search space reduction and effective utilization of previous perturbations. We repetitively multiply the previous perturbations by 1.05 until the model misclassifies the image. For visualization, we add 0.5 to the adversarial perturbations and multiply them by 5. The sample images are from the LFW dataset [15].

1. Introduction

Convolutional neural networks have brought remarkable performance improvement in face recognition, but maliciously crafted inputs can fool them with small noise called adversarial perturbations. Since facial recognition can be used in a wide range of areas, such as payment, finance, and criminal identification, adversarial attacks pose a great threat to their security. Recent studies [4, 9, 6] have shown that this attack is feasible even when adversaries cannot access the target models' interiors but can only obtain the hard-label predictions. However, decision-based black-box attacks require many queries to find an imperceptible adversarial perturbation for an image. Since making a lot of queries can cause a financial burden, time consumption, and even system administrators' suspicion, reducing the number of queries is crucial for these decision-based attacks.

In line with this research direction, Dong *et al.* [9] propose EA, an evolutionary algorithm-based decision-based attack on face recognition. It accelerates the convergence of perturbations by updating the sampling distribution of random noise and shows the superior query efficiency over

previous attack methods.

However, most decision-based attacks on face recognition [9, 6], including EA, do not take advantage of the characteristics of face recognition in that it optimizes the noise in the whole image area. Even with the same person, the background can change drastically, so we can naturally assume that most of the key features for face recognition locate in facial areas. If we optimize perturbations only for facial areas, we can effectively reduce query waste for optimizing background noise.

Furthermore, to the best of our knowledge, existing query-based black-box attacks do not utilize previous adversarial perturbations created for other images. Prior works on query-efficient black-box attacks reduce the number of required queries by exploiting the gradients of substitute networks [11], the gradients of previous iterations or the local correlation in an image as prior information [17]. However, any attempts to recycle adversarial perturbations of previous images are hardly found in literature.

It may be because directly applying a previous adversarial perturbation may not work in most cases, as it is not optimized for the current image. Even if we magnify the previous adversarial perturbation to make it effective against the image, it may harm the convergence rate because of the misalignment of the perturbation to the face. Nevertheless, if the previous adversarial perturbation properly aligns with the face, it may provide a better initial point with a smaller magnitude.

The above two limitations are difficult to overcome with naive solutions. For the background noise optimization, since it is difficult to estimate the gradient towards a target class in a decision-based black-box setting, an adversary needs to initialize an image with the target person's face and gradually reduce the norm of perturbation afterward. However, it is hard to replace the face with the target identity's face via simple copy and paste due to differences in facial pose and size. On the other hand, for the second limitation, one may try traditional face-warping based on Delaunay triangulation [20] of face landmarks to align the adversarial perturbation, but the warping may not work properly for large poses where some face landmarks are invisible.

In this paper, we exploit 3D face alignment to deal with the above limitations more flexibly and propose *a general attack strategy* for query-efficient black-box attacks on face recognition. Our strategy is to create adversarial perturbations in the UV texture map and project them onto the 3D face in the image. This separation of *the generation of adversarial perturbations* and *the alignment process* enables efficient recycling of previous adversarial perturbations as shown in Fig. 1. Moreover, it reduces unnecessary queries for background noise optimization by limiting the perturbation search space to the facial area that is important for face recognition.

Our major contributions can be listed as follows:

1) We propose Geometrically Adaptive Dictionary Attack (GADA), *a general strategy* for query-efficient black-box attacks on face recognition using 3D face alignment. It remarkably improves query efficiency by limiting the perturbation search space to facial areas of images.

2) GADA can efficiently utilize previous adversarial perturbations created in the common UV texture map since it can align the perturbations to the faces. To the best of our knowledge, this is the first query-based black-box attack that takes advantage of previous adversarial perturbations created for other images.

3) To evaluate memory-based black-box attacks, we also propose a novel evaluation protocol that measures and compares the average number of queries used for finding norm-bounded adversarial perturbations for a predetermined sequence of images. This evaluates how effectively a black-box attack method utilizes previous attack experiences against a target model.

4) We present an attack strategy to evade stateful detection that computes the perceptual similarity of query images to detect the process of query-based black-box attacks. GADA can confuse similarity-based detectors by injecting noise into the background area while gradually reducing perturbations in the facial area. To our knowledge, this is the first effective way to bypass these detection techniques.

5) We demonstrate that GADA brings overwhelming performance improvement when applied to two attack methods, EA [9] and SFA [6], through experiments on the LFW [15] and CPLFW [26] datasets. Specifically, for dodging attacks on the LFW dataset, compared to EA, our proposed strategy almost halves the perturbation norm when the query budget is 1K. It also drops the average number of queries required to find an adversarial perturbation whose ℓ_2 norm is two by more than 2,100.

2. Background

2.1. Face recognition

Face recognition involves two sub-tasks: face verification and face identification. Face verification is a task of comparing a candidate face to another, and verifying whether the two face images are of the same identity, and face identification is a task to classify an image into one of the gallery's identities. In this paper, we deal with face verification task, but the proposed method can be adapted to face identification task as it is a problem of finding the face with the closest distance in the gallery.

In face verification, a neural network f encodes an input image $\boldsymbol{x}$ into a feature vector $f(\boldsymbol{x}) \in \mathbb{R}^D$, where D indicates the feature dimension. For a pair of images, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we can compute their ℓ_2-normalized Euclidean distance

(a proxy for cosine distance) as follows.

$$Dist(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left\| \frac{f(\boldsymbol{x}_1)}{\|f(\boldsymbol{x}_1)\|_2} - \frac{f(\boldsymbol{x}_2)}{\|f(\boldsymbol{x}_2)\|_2} \right\|_2^2. \quad (1)$$

If $Dist(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is less than a threshold γ, the model recognizes that the people in two images represent the same identity. Otherwise, they are considered different. The performance of face verification models has been improved through various angular margin losses ranging from SphereFace[18], CosFace [18], and ArcFace[8]. These losses are designed to increase the inter-class distance while shrinking the intra-class distance. Recently, Huang *et al.* [16] further improve the accuracy by incorporating the idea of curriculum learning into their loss function to induce the models to treat easy samples in early stages and hard ones in later.

2.2. Adversarial setting

We now construct a black-box threat model that wraps a face verification model. First, let us denote $\boldsymbol{x}_1$ as $\boldsymbol{x}_A$ and $\boldsymbol{x}_2$ as $\boldsymbol{x}_S$ to clarify that each of them is owned by an adversary and a server, respectively. The target model performs face verification by comparing the query input $\boldsymbol{x}_A$ with $\boldsymbol{x}_S$ in the server. As a black-box threat model, $\boldsymbol{x}_S$ is inaccessible to the attacker, and thus, the adversary can only modify $\boldsymbol{x}_A$ and make queries to check the hard-label predictions of the model. Based on the above setting, we can represent a black-box face verification model that returns a hard label $h \in \{1, 0\}$ for $\boldsymbol{x}_A$ as follows:

$$h_{\boldsymbol{x}_S}(\boldsymbol{x}_A) = \begin{cases} 1, & \text{if } Dist(\boldsymbol{x}_A, \boldsymbol{x}_S) < \gamma \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

In the following, we will omit the subscript of h for notational convenience.

An adversary has a clean image $\boldsymbol{x}_A$ and wants to generate an adversarial example that has minimal perturbation while successfully fooling the target model h. Then, we can represent the adversary's objective as follows:

$$\arg\min_{\boldsymbol{\delta}_q} \|\boldsymbol{\delta_q}\|_p, \text{ s.t. } h(\boldsymbol{x}_A + \boldsymbol{\delta}_q) \neq h(\boldsymbol{x}_A) \text{ and } q \leq Q, \quad (3)$$

where Q is the query budget of the adversary, and q is the number of queries used to make $\boldsymbol{\delta}_q$ and $p \geq 0$. Unless otherwise noted, we use $p = 2$ (i.e., ℓ_2 norm) for the perturbation norm objective. In this paper, we also assume that pixel values of images are normalized into $[0, 1]$. Depending on the value of $h(\boldsymbol{x}_A)$ (i.e., whether the pair of images originally represent the same identity), we call the attacks differently either dodging attacks ($h(\boldsymbol{x}_A)$=1) or impersonation attacks ($h(\boldsymbol{x}_A)$=0). These two types of attacks differ in their initial values and objectives, and we will explain them in detail in Section 3.

2.3. 3D face alignment

3D face alignment identifies the 3D geometric shape of faces in an image, and it helps to find the semantic meanings of facial pixels. 3D face alignment is widely used in face frontalization [27] and face presentation attack detection [22, 19] but has not yet been used for adversarial attacks to the best of our knowledge.

Among learning-based 3D face alignment methods, Guo *et al.* [10] propose 3DDFA_V2 that shows superior performance in terms of accuracy, speed, and stability of predictions. This method regresses the parameters of the 3D Morphable Model (3DMM) [2] through a light model. The 3DMM used in [10] describes the 3D face with PCA, and it can be represented as follows:

$$\boldsymbol{S} = \bar{\boldsymbol{S}} + \boldsymbol{A}_{id}\boldsymbol{\alpha}_{id} + \boldsymbol{A}_{exp}\boldsymbol{\alpha}_{exp}, \quad (4)$$

where $\boldsymbol{S}$ is the 3D mesh of face model, $\bar{\boldsymbol{S}}$ is the mean 3D face shape, $\boldsymbol{\alpha}_{id}$ and $\boldsymbol{\alpha}_{exp}$ are the facial shape and expression parameters, respectively. The 3DDFA_V2 model predicts the following parameters of the 3DMM, $\boldsymbol{p} = (\boldsymbol{R}, \boldsymbol{\alpha}_{id}, \boldsymbol{\alpha}_{exp}, \boldsymbol{t}_{2d})$, where $\boldsymbol{R}$ and $\boldsymbol{t}_{2d}$ are the rotation matrix and the translation vector, respectively. From the above predicted parameters, the 3D face can be reconstructed as follows.

$$\boldsymbol{V}_{3D} = \boldsymbol{R}(\bar{\boldsymbol{S}} + \boldsymbol{A}_{id}\boldsymbol{\alpha}_{id} + \boldsymbol{A}_{exp}\boldsymbol{\alpha}_{exp}) + [\boldsymbol{t}_{2d}, 0]^T. \quad (5)$$

Furthermore, the reconstructed 3D Face can be rendered on a 2D image using rasterization with z-buffer as follows.

$$\boldsymbol{x}^* = Render(\boldsymbol{x}, \boldsymbol{V}_{3D}, \boldsymbol{C}_V, \boldsymbol{Z}), \quad (6)$$

where $Render$ function displays the 3D triangular meshes on the image $\boldsymbol{x}$ with the vertex coordinates $\boldsymbol{V}_{3D}$ and z-Buffer $\boldsymbol{Z}$ with vertex colors $\boldsymbol{C}_V$. $\boldsymbol{Z}$ contains the depth of reconstructed 3D face and it helps to avoid rendering the occluded area.

3. Proposed method

As an overview, GADA creates adversarial perturbations in UV texture maps and projects them onto the 3D face obtained by 3D face alignment. A UV texture map is a planar representation of a 3D model's surface used to paint the surface of the 3D model. We can project face textures in a UV map onto a 3D face model's surface through UV mapping. Likewise, we can also extract the face texture from the 3D face mask as a UV texture map according to the relative coordinates of the face. Since GADA is a general attack strategy for query-efficient black-box attacks on face recognition, we explain its general scheme, but it can be adapted to various attacks for improving query efficiency.

Initialization of 3D face alignment. In face verification, it is assumed that each image has a face, and thus, a 3D face

mask can be found via 3D face alignment. Since the face position in an image is fixed, there is no need to perform 3D face alignment for each query, so GADA performs 3D face alignment only once at the initial stage to get the vertex coordinates of the 3D face. We also compute $\boldsymbol{Z}$ at the initial stage to prevent rendering of occluded areas in the future.

UV mapping. The adversarial perturbations in the UV texture map contain noise values according to the relative coordinates of the face. By exploiting the UV mapping, adversarial perturbations can be properly projected onto the 3D face with $\boldsymbol{C}_V$ that retains the noise of RGB colors for the vertices of the 3D face model. It operates as projecting a translucent UV texture map on the 3D face. To project the perturbation of $\boldsymbol{C}_V$ onto the image, we use the modified $Render$ function that adds the corresponding vertex color of the point in the 3D face to the original pixel values in rasterization. Since GADA finds adversarial perturbations in the UV space, we also need to use UV $\rightarrow \boldsymbol{C}_V$ conversion. For this conversion, we obtain $\boldsymbol{C}_V$ from the UV texture map using bilinear interpolation with the UV coordinates.

Rendering details of GADA. When rendering each triangular face of the 3D mesh, the color of an inner point is calculated as the combination of colors of its three vertices weighted by its distances. However, this slows down the convergence of perturbations as the color of one point in an image depends on the three vertices of a triangle. So, for the inner points of each triangle, we use the color of the triangle's first vertex. It improves the convergence rate, especially in ℓ_∞ norm-based attacks such as SFA [6].

The reason for the use of the UV texture map. Vertex colors are stored in a matrix $\boldsymbol{C}_V$ of $N_V \times 3$, where N_V is the number of vertices of the 3DMM. Since we use a dense 3D face model, N_V can be larger than the dimension of an image (in our experimental setting, N_V is 38,365, and the spatial dimension of an image is $112 \times 112 = 12,544$). If adversaries try to find the optimal adversarial perturbations in $\boldsymbol{C}_V$, such a large search space can degrade query efficiency because many queries are required for search. Meanwhile, existing query-efficient attacks that operate in the image space tend to find perturbations in the down-scaled image to reduce the search space. However, in our case, the adjacency of projected vertices of $\boldsymbol{V}_{3D}$ in the image varies from image to image, so it is not easy to utilize common reduced space like the image-space attacks. In addition, if we find perturbations in $\boldsymbol{C}_V$, recycling efficiency becomes poor because only a relatively small percentage of the vertex colors are used in an image as our rendering function maps each image pixel to one vertex. To effectively reduce the search space and facilitate recycling, we find perturbations in the UV texture map and convert it to $\boldsymbol{C}_V$ rather than directly searching for them in $\boldsymbol{C}_V$. We set the size of the UV texture map as the same size as the image, but this can be set arbitrarily regardless of the image size.

Figure 2: Examples of initial images of GADA for impersonation attacks on the LFW dataset [15]. The top and middle row images are the source and target images, respectively. As shown in the bottom row, the target images' faces are replaced with the source images' faces. We apply various data augmentation to the source images' textures to increase the chance of prediction to the target identity. An example is shown in the rightmost image.

3.1. Dodging attacks

For dodging attacks, GADA initializes the UV texture map as a uniformly random image (and subtract the original UV texture as it adds the texture values when rendering) for misclassification of the target model and gradually reduce the perturbation from it.

Dictionary attacks. The adversarial perturbation created in the common UV space can provide a better initial state when attacking other images. We can naturally expect that the perturbation for a person can be effective in different poses of the same identity. Furthermore, We also conjecture that the perturbation for a person can be more effective against similar-looking people than others. From this motivation, when an attack is complete, GADA saves the minimal adversarial perturbation and the face feature in its dictionary. Therefore, if we use dictionary attacks, we initialize an image with a previous perturbation if it is available. When GADA draws a previous adversarial perturbation from its dictionary, it fetches the perturbation corresponding to the feature vector closest to that of the current image. This is because the closer the feature vectors of two images, the more likely they have similarities. After it fetches a perturbation, it repetitively multiplies the perturbation by 1.05 until the target model misclassifies the image pair.

3.2. Impersonation attacks

In the decision-based black-box setting, an impersonation attack starts from a target identity's image (source image) and gradually update the image to look like the target image. Note that the target image differs from $\boldsymbol{x}_S$, which

is inaccessible to attackers. Except for the difference in the initialization of the image, the attack process is similar to that of dodging attacks. Since the proposed GADA strategy applies perturbations only to facial areas, it is necessary to extract the UV face texture map from the target identity's source image to make an initial image. To do this, it extracts $\boldsymbol{C}_V$ from the image and converts it into a UV texture map. We first obtain $\boldsymbol{C}_V$ from an image using bilinear interpolation with the image coordinates. To convert $\boldsymbol{C}_V$ to a UV texture map, we can use the $Render$ function as follows.

$$\boldsymbol{x}_{uv} = Render(\mathbf{0}, \boldsymbol{V}_{UV}, \boldsymbol{C}_V, -10^8 \times \mathbf{1}), \tag{7}$$

where $\boldsymbol{V}_{UV}$ is the vertex coordinates in the UV space. With the obtained UV texture map, we can replace the target image's face with the source image's face. Since the $Render$ function adds the vertex colors to the corresponding original pixel values, we use the UV texture of the target image subtracted by that of the source image. We illustrate some initial images of impersonation attacks in Fig. 2.

In impersonation attacks, we do not use the dictionary for the following reasons: (1) if the dictionary does not have the adversarial perturbation for the target identity, it is impossible to make the image to be recognized as the target identity; (2) even if the dictionary has the perturbation for the target identity, the projected perturbation may not work due to changes in the facial pose and scale. In score-based attacks where adversaries can obtain the distance between two facial features, they may exploit the dictionary for impersonation attacks.

Data augmentation on the UV texture map. Even if the source image's face is projected on the target image, there may be some cases in which the target model does not recognize it as the target identity due to errors in texture mapping or missing textures. To increase the chance to find an image recognized as the target identity, GADA applies data augmentation on the UV texture map of the source image with random horizontal flip and random color jittering (brightness, contrast, saturation, hue). If it fails to find an image recognized as the target identity even after 200 attempts, it applies the original image-space attack for the image.

4. Experiments

4.1. Experimental settings

We evaluated the performance improvement of GADA with Labeled Faces in the Wild (LFW) [15] and Cross-Pose LFW (CPLFW) [26] datasets. We used the datasets for the following reasons: (1) the LFW dataset is one of the most representative datasets for face recognition; (2) Since the CPLFW dataset has more diverse facial pose changes than the LFW dataset, it is appropriate to show that the GADA strategy works well for general cases. We found the best threshold with the highest accuracy using 10-fold splits for each dataset and made the model classify images based on that threshold.

For creating test image sequences for dodging attacks, we randomly extracted 500 pairs from each dataset, each pair of which represents an identity. When composing test image sequences for impersonation attacks, we permuted each pair's left image (i.e., $\boldsymbol{x}_A$) so that all pairs are perceived as different identities and used them for target images for impersonation attacks. Meanwhile, we used the original images as the source images for the target identities. The maximum number of queries available in each image is set to 10K for both types of attacks. For measuring the query efficiency of attacks, we used two types of metrics: (1) the smallest norm of adversarial perturbations found within a specific query budget; (2) the average number of queries used to generate an adversarial example whose norm is less than or equal to a threshold. This metric is useful for comparing the average number of queries spent to find a sufficiently small adversarial perturbation.

In our experiments, we used the ArcFace ResNet-50 [8, 13] model[1] trained on the 112×112 aligned MS-Celeb-1M dataset [12] as the target model for the black-box attacks. For GADA's dictionary attacks, a feature embedding for $\boldsymbol{x}_A$ needs to be computed. Since we assumed a realistic black-box setting, we used a network which is different from the target model. Specifically, we used FaceNet[2] [21] trained on the VGGFace2 dataset [3]. For 3D face alignment in GADA, we used a pre-trained model[3] of 3DDFA_V2 provided by the authors.

The proposed method can be applied to various existing query-based black-box attacks for improving their query efficiency. In this paper, we applied GADA to two different decision-based attacks, EA [9] and SFA [6], and show their performance improvement.

Notations. We used diverse variants of attack methods depending on their additional functions in our experiments. To refer to them, we attach 'G' to the name of an attack when using geometrically adaptive attacks and add 'D' when using dictionary attacks. For example, EAGD is the improved version of the EA attack [9] with the proposed geometrically adaptive attacks and dictionary attacks.

4.2. Quantitative results

We evaluated the query efficiency of the variants of EA. For comprehensive comparisons, we also evaluated other state-of-the-art decision-based attack methods, HSJA [4] and Sign-OPT [7]. GADA can also be applied to the above

[1]We use the pre-trained ArcFace model provided from `https://github.com/ZhaoJ9014/face.evoLVe.PyTorch`

[2]We use the pre-trained FaceNet model provided from `https://github.com/timesler/facenet-pytorch`

[3]We use the default model whose backbone is MobileNet_V1 [14] from `https://github.com/cleardusk/3DDFA_V2`

Dodging attacks												
	LFW dataset [15]						CPLFW dataset [26]					
	Minimum perturbation norm with query budget				Avg. # queries for perturbation with norm		Minimum perturbation norm with query budget				Avg. # queries for perturbation with norm	
Attack method	1K	2K	5K	10K	4	2	1K	2K	5K	10K	4	2
Sign-OPT [7]	17.54	11.19	4.66	2.57	5738	8460	16.87	10.52	4.07	2.18	5016	7498
HSJA [4]	11.4	7.62	3.8	2.32	4692	7769	11.31	7.27	3.41	2.01	4191	6833
EA [9]	13.73	7.17	2.74	1.44	3561	6445	13.30	6.63	2.42	1.27	3231	5683
EAD	10.20	5.50	2.27	1.27	2807	5705	10.20	5.50	2.27	1.27	2807	5705
EAG	7.99	4.20	1.81	1.20	2155	4697	7.83	4.00	1.68	1.09	2086	4263
EAGD	**6.39**	**3.58**	**1.71**	**1.19**	**1778**	**4275**	**6.42**	**3.45**	**1.55**	**1.04**	**1757**	**3845**
Impersonation attacks												
	LFW dataset [15]						CPLFW dataset [26]					
	Minimum perturbation norm with query budget				Avg. # queries for perturbation with norm		Minimum perturbation norm with query budget				Avg. # queries for perturbation with norm	
Attack method	1K	2K	5K	10K	4	2	1K	2K	5K	10K	4	2
Sign-OPT [7]	22.19	16.47	8.29	4.20	7574	9076	20.45	13.67	5.16	2.09	5163	7083
HSJA [4]	18.72	13.53	6.40	3.58	6566	8538	15.48	9.47	3.47	1.71	3844	5963
EA [9]	14.84	8.50	3.43	1.82	4363	7323	11.58	5.61	1.92	0.96	2652	4537
EAG	**10.85**	**6.22**	**2.63**	**1.57**	**3219**	**6277**	**8.40**	**4.04**	**1.43**	**0.83**	**1925**	**3505**

Table 1: Evaluation of decision-based advesrarial attacks on the two datasets.

two attacks, but we applied GADA to EA because EA shows the best query efficiency in our experiments. We described hyperparameter settings for each attack method in supplementary material. In EAD and EAGD, to prevent the evolutionary algorithm from being stuck at the boundary of the pixel value range, we clip the image of perturbation-applied areas into the range between 0.2 and 0.8 when they utilize a previous perturbation.

Table 1 shows the evaluation results of decision-based adversarial attacks on the two datasets. Note that for all methods, we measured the norm of perturbations in the image space, not in the UV space. For dodging attacks on the LFW dataset, EAGD reduces the smallest perturbation norm by 1.03 compared to EA. The results of EAGD clearly show that using a dictionary for utilizing previous perturbation helps to improve query efficiency. Besides, compared to EA, EAGD reduces the number of queries required to find an adversarial perturbation whose norm is less than or equal to two by 2,170 for dodging attacks and 1,046 for impersonation attacks. This query efficiency can save considerable resources of attackers. We illustrated the perturbation norm curves that visually shows the query efficiency of each method in supplementary material.

4.3. Qualitative results

Figure 3 shows adversarial examples from EA and its variants. EAG generates perturbations only in the facial area, so unlike EA, background noise does not exist. Since GADA reduces the search space, it can be seen that when Q=3K, the perturbation is significantly reduced compared to EA. Dictionary attacks of EAGD help to start at a smaller perturbation and the results show that when Q=1K, EAGD reduces the perturbation norm to $0.37\times$ compared to EAG.

For impersonation attacks, existing methods start from a source image of the target identity and iteratively update the adversarial example to look like the target image. However, as shown in Fig. 3, if the target image has a flat background, the adversarial perturbation becomes more noticeable. In contrast, as GADA initially replace the target image's face with the source image's face and optimizes the perturbation in the facial area, the noise becomes far more imperceptible when the query budget is only 2K. For more qualitative comparisons with other attacks, we include more exemplary results in supplementary material. We also conducted the above experiments with a deeper target model, Curricular Face ResNet-100 [16, 13], and the experimental results are listed in supplementary material.

4.4. Adaptation to ℓ_∞ norm-based attacks

Since GADA is a general strategy applicable to various query-based black-box attacks, we applied it to SFA [6] that is particularly effective for ℓ_∞ norm constraint. Figure 4 shows the results of SFA and its variants for dodging attacks on the LFW dataset. SFAGD clearly reduces the perturbation norm faster than SFA. When Q=10K, the difference of ℓ_∞ norm between SFAD and SFAGD is small, but SFAGD almost halves ℓ_2 norm of the perturbation since SFAGD perturbs the facial area only.

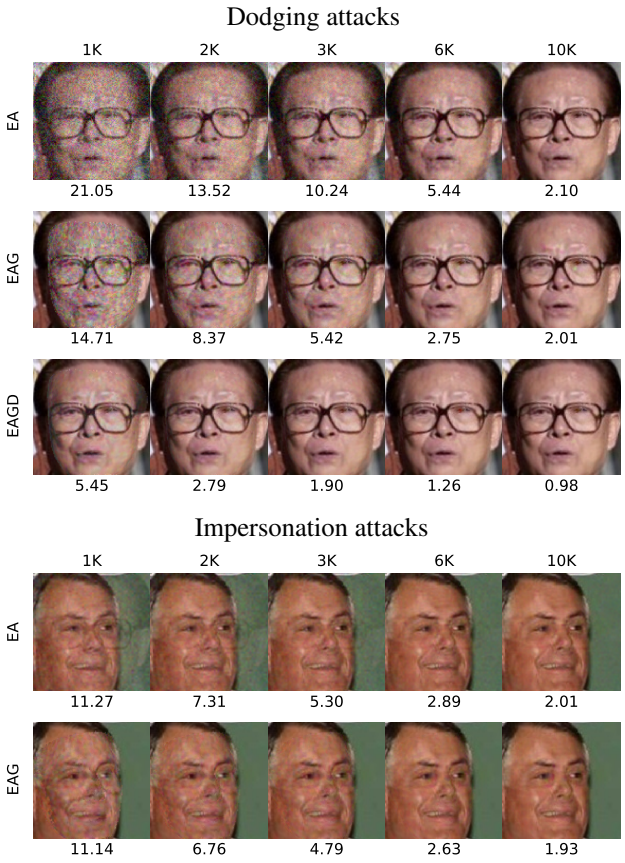

Figure 3: Qualitative results of dodging and impersonation attacks on the LFW dataset [15]. For each attack, we illustrate the minimum norm-adversarial examples in each query budget. The ℓ_2 norm of perturbation is displayed under each image.

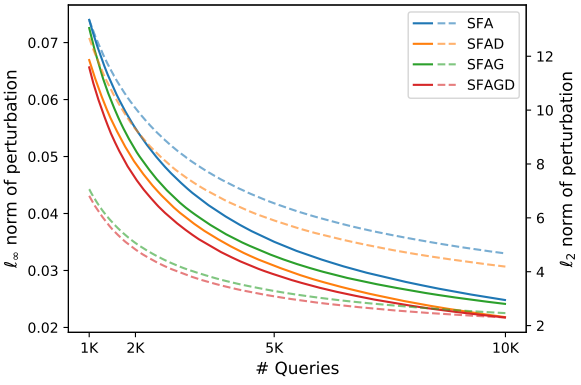

Figure 4: Perturbation norm curves of SFA and its variants for dodging attacks on the LFW dataset. The solid lines represent ℓ_∞ norm of perturbations, and the dotted lines represent l_2 norm of perturbations.

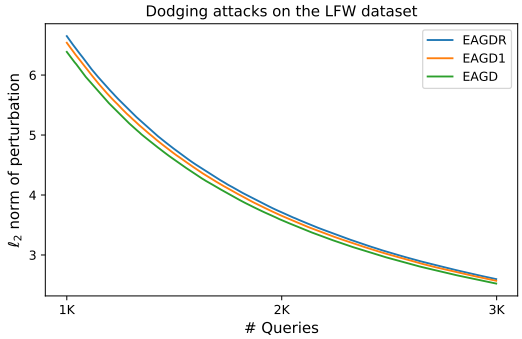

Figure 5: Perturbation norm curves for different ways of fetching perturbations.

4.5. Different ways to fetch a perturbation

Originally, GADA stores the adversarial perturbations of many identities in a dictionary to fetch the perturbation of the closest identity in the feature space for better initialization in the future. In this ablation study, we find out the effectiveness of this way of fetching. We evaluated two different ways of fetching: EAGD1 and EAGDR. EAGD1 has a dictionary with only one memory slot so that it fetches and updates the perturbation for each attack. EAGDR stores adversarial perturbations of many identities like EAGD, but it fetches a perturbation in the dictionary randomly unless the same face feature exists in the dictionary. Figure 5 shows the results of EAGD along with the above variants. The results show that EAGD indeed has superior query efficiency than the other variants in the early stages. One may take the closest top-k perturbations and multiply them by a large number and gradually scale them down to find the smallest adversarial perturbation. Devising an efficient way to find a more useful perturbation in the dictionary can be an interesting research topic in the future.

4.6. Evading stateful detection

Query-based black-box attacks inevitably need to send a large number of perceptually similar images for queries in their processes. By targeting this commonality, a memory-based detection technique [5] has been proposed recently. It stores perceptual similarity embeddings of recent queries and detects the generation of adversarial examples if too many similarity embeddings are close. In detail, it encodes each image into a feature with a similarity encoder and stores it in its memory. If the k-nearest neighbor (k-NN) distance of the current image's embedding is smaller than a threshold, the detector judges the query as an adversarial attack. Existing decision-based black-box attacks are nearly impossible to avoid this detection technique due to their nature.

Despite the difficulties, we present a novel attack strat-

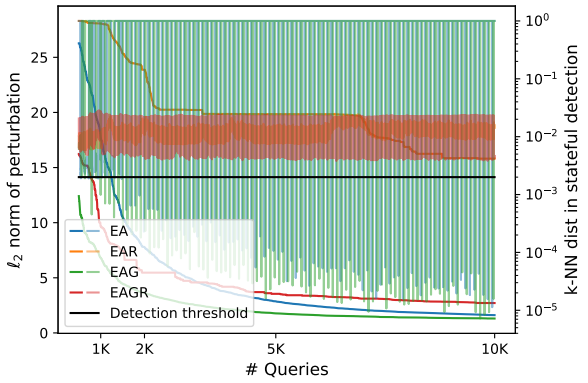

Figure 6: Perturbation norm and k-NN distance curves of EA and its variants. If the k-NN distance falls below the threshold, the query is detected as an adversarial attack.

egy that avoids this detection by injecting noise differently into the foreground and the background. Specifically, we disturb the similarity detector by consistently adding small random noise to the background while optimizing the foreground perturbation. An essential premise for this is that background noise should not affect the prediction of the target model. Since most face recognition models focus on inner part of faces, we assume that this premise is established for general cases.

To evaluate this approach in our experimental setting, we need a new similarity encoder for larger-sized images as Chen *et al.* [5] use perceptual similarity encoders for the 32x32 sized images of the CIFAR-10 dataset. Instead, we used Learned Perceptual Image Patch Similarity (LPIPS) [25] as a similarity encoder, which can measure general perceptual similarity. We used a pre-trained SqueezeNet model for LPIPS v0.1 and used k=50 for k-NN distance and 2e-3 for the threshold of detection. For its memory, we used a circular buffer containing 100 recent queries. The detection operates only when more than $k-1$ queries are stored in the buffer. Following [5], we flush the buffer whenever an adversarial attack is detected. We evaluated this detection against dodging attacks on the first 100 images of the test sequence of the LFW dataset. For SO, HSJA, EA, and EAG, the average number of detection per image is 166, 191, 188, 192, respectively. Considering that the total query budget of 10K divided by k (50) is 200, the above results show that most attacks are immediately detected when the detector starts to operate.

Before we describe the results of the new attack strategy against the detection, we explain our attack strategy in detail. We add random Gaussian noise into the background with $\sigma = 0.01$ divided by the background area ratio in the image. This makes larger noise when the background area is relatively small. We also applied random Gaussian noise with $\sigma = 0.02$ to the entire image for EA to avoid detection. We name the two variants EAGR and EAR, respectively. Note that the ratio of the background is less than 30% on average, so we used a smaller σ for EAR in most cases. For both methods, we make a query without background noise every i iterations to reduces the perturbation norm by excluding the background noise. We used $i = 20$ in our experiments, but i should be set in proportional to the detector's buffer size. With the same experimental settings as the other attacks, both methods are never detected for the 100 test images. However, when the query budget is 10K, EAR and EAGR have 11.69 and 3.47, respectively, in their smallest perturbation norm on average. It means that unlike EAR, EAGR can successfully reduce the perturbation norm while avoiding detection. Figure 6 shows the k-NN distance and perturbation norm curves for a test image.

5. Related work

In the following, we briefly introduce related studies and their difference to our strategy. Dabouei *et al.* utilize spatial transformation of images with face landmarks to fool face recognition models. It creates an adversarial example by displacing the position of the facial landmarks in the white-box threat model. It is common with our strategy in that it creates perturbation based on facial geometry, but GADA significantly differs as GADA is an intensity-based query-efficient black-box attack strategy that finds perturbations in the UV map. On the other hand, there are several 3D model-based adversarial attacks [1, 23, 24] which render 3D models in 2D space and find adversarial shapes or textures. However, their aim is not to query-efficient black-box attacks but to robust attacks under diverse views.

6. Conclusion

In this paper, we propose a general strategy for query-efficient black-box attacks on face recognition. It creates an adversarial perturbation in a common UV texture map and projects it onto the face area through 3D face alignment. By separating the facial areas and the background, we also suggest that injecting noise into the background that hardly affects predictions can help to circumvent stateful detection. In this paper, we opened a new research avenue for memory-based black-box attacks that can efficiently utilize previously found perturbations. We will release GADA's code and the test image sequences' indices for fruitful exploration with other researchers. The generalized core ideas of our work are limiting the perturbation search space to the region of interest and recycling previously found perturbations with object-aware semantic correspondence. In this paper, we deal with face recognition, but the above ideas can be effective for attacks on other tasks. We leave them for future work.

References

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[3] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[4] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1277–1294. IEEE, 2020.

[5] Steven Chen, Nicholas Carlini, and David Wagner. Stateful detection of black-box adversarial attacks. *arXiv preprint arXiv:1907.05587*, 2019.

[6] Weilun Chen, Zhaoxiang Zhang, Xiaolin Hu, and Baoyuan Wu. Boosting decision-based black-box adversarial attacks with random sign flip. In *Proceedings of the European Conference on Computer Vision*, 2020.

[7] Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations*, 2020.

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[9] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7714–7722, 2019.

[10] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. *arXiv preprint arXiv:2009.09960*, 2020.

[11] Yiwen Guo, Ziang Yan, and Changshui Zhang. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. In *Advances in Neural Information Processing Systems*, pages 3825–3834, 2019.

[12] 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*, pages 87–102. Springer, 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[15] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[16] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5901–5910, 2020.

[17] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2018.

[18] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[19] Yunxiao Qin, Chenxu Zhao, Xiangyu Zhu, Zezheng Wang, Zitong Yu, Tianyu Fu, Feng Zhou, Jingping Shi, and Zhen Lei. Learning meta model for zero-and few-shot face anti-spoofing. *arXiv preprint arXiv:1904.12490*, 2019.

[20] D. Ruprecht and H. Muller. Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications*, 15(2):37–43, 1995.

[21] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[22] Zezheng Wang, Zitong Yu, Chenxu Zhao, Xiangyu Zhu, Yunxiao Qin, Qiusheng Zhou, Feng Zhou, and Zhen Lei. Deep spatial gradient and temporal depth learning for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5042–5051, 2020.

[23] Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6898–6907, 2019.

[24] Xiaohui Zeng, Chenxi Liu, Yu-Siang Wang, Weichao Qiu, Lingxi Xie, Yu-Wing Tai, Chi-Keung Tang, and Alan L Yuille. Adversarial attacks beyond the image space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4302–4311, 2019.

[25] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[26] T. Zheng and W. Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Technical Report 18-01, Beijing University of Posts and Telecommunications, February 2018.

[27] Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. Rotate-and-render: Unsupervised photorealistic face

rotation from single-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5911–5920, 2020.