Transferable Black-Box Attack Against Face Recognition With Spatial Mutable Adversarial Patch

Haotian Ma, Ke Xu, *Member, IEEE*, Xinghao Jiang, *Senior Member, IEEE*, Zeyu Zhao, and Tanfeng Sun, *Senior Member, IEEE*

***Abstract*— Deep Neural Networks (DNNs) are vulnerable to adversarial patch attacks, which raises security concerns for face recognition systems using DNNs. Previous attack methods focus on the perturbation texture and generate adversarial patches with fixed shapes at random or pre-designed locations, which causes poor adversarial transferability. This paper proposes a Spatial Mutable Adversarial Patch (SMAP) method to generate a dynamic mutable patch to be injected into the face. In the proposed SMAP, the texture, position and shape of the patch are optimized simultaneously and the patch generation pipeline is end-to-end differentiable. Specifically, a Patch Location Selection Scheme is designed to find the critical patch position with the most significant influence on the target identity by the step-based gradient search. By innovatively bridging the pre-defined mask and the dynamic update of the patch, the patch position and shape are changed based on the affine transformation and sampling mechanism in each iteration, which maintains the importance of the injected patch to the adversarial objective. To evaluate the vulnerability of face recognition models, we explore more threatening impersonation attacks under the black-box setting and design a strict evaluation metric that aligns with the real-world scenario. Extensive experiments show that the proposed SMAP improves attack performance across various face recognition models and datasets. Moreover, SMAP achieves better transferability on commercial face recognition systems than existing methods.**

***Index Terms*— Adversarial patch, face recognition, impersonation attack, joint optimization, spatial mutability.**

I. Introduction

DEEP Neural Networks (DNNs) achieve excellent performance in many application domains, including computer vision [1], [2], and natural language processing [3], [4] and meet or exceed human performance (like accuracy and speed). Face recognition based on DNNs [5], [6] as a fundamental vision task is used for many application systems, such as payment and phone unlock. Therefore, it is crucial to guarantee the security and reliability of face recognition.

However, some work [7], [8], [9], [10] have shown that deep neural networks are vulnerable to adversarial examples crafted by adding elaborate perturbations to images. These adversarial samples usually cause the neural network to make incorrect predictions. Adversarial attacks can usually be divided into white-box attacks [7], [8] and black-box attacks [11], [12]. Under the white-box attack setting, the attacker has full access to the model. In a black-box attack, the attacker only knows the inputs and outputs of the model. Black-box attacks include query-based attacks [13], [14], [15], which optimize the attack objective by querying the models, and transfer-based attacks [11], [16], which leverage the transferability of the adversarial examples against the models.

Due to the global and imperceptible perturbation pattern, these primitive methods are difficult to be captured by cameras and be applied to real scenes. This causes them to be used only in the digital domain. Later, several works propose adversarial patch attacks [12], [16], [17], [18], which perturb pixels within a restricted area. Compared to the impracticality of global perturbation, patch attacks generate local perturbation with sufficient magnitude to be applied in the real world. Some patch methods have successfully implemented attacks in different application scenarios. For example, Hu et al. [19] make T-shirts and skirts with adversarial texture to evade person detectors. Sharif et al. [20] print adversarial eyeglasses to fool face recognition systems. Komkov and Petiushko [21] make adversarial hats to attack face id systems. Surprisingly, some researchers have successfully unlocked the face recognition of smartphones using adversarial patch.[1] The attacks in realistic scenarios threaten face recognition systems, and it is urgent to explore the vulnerability of face recognition models. On the other hand, some researchers are also working on the security of face recognition systems by detecting adversarial face attacks to ensure the security of people's information and privacy.[2]

Currently, most existing adversarial patch attack methods [16], [17], [18], [20], [22], [23] only consider the texture of the patch perturbation but not the location of the patch. Class Activation Mapping [24], [25] technology provides visual explanations of DNN model predictions. It uses the back-propagation gradients to locate the attention of the model.

Manuscript received 4 May 2023; revised 24 July 2023 and 15 August 2023; accepted 22 August 2023. Date of publication 30 August 2023; date of current version 20 September 2023. This work was supported by the National Natural Science Foundation of China under Grant 62272299, Grant 62272297, and Grant 62002220. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chia-Mu Yu. *(Corresponding author: Xinghao Jiang.)*

Haotian Ma, Ke Xu, and Zeyu Zhao are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: htmaaaaaa@sjtu.edu.cn; l13025816@sjtu.edu.cn; 329161318zzy@sjtu.edu.cn).

Xinghao Jiang and Tanfeng Sun are with the National Engineering Laboratory for Information Content Analysis Technology, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xhjiang@sjtu.edu.cn; tfsun@sjtu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2023.3310352

[1] https://zhuanlan.zhihu.com/p/348847509

[2] https://www.realai.ai/products/55.html

Through this technology, it is observed that the importance of pixels in different regions varies. Therefore, some works [26], [27] focus on finding a critical area to achieve a high attack success rate. However, the pixels in this region do not always play an important role in each iteration. There is no guarantee that the pre-designed position of most significance to the adversarial objective will stay the same. Rao et al. [28] propose to move the patch in each iteration to find an optimal location. GDPA [29] try to train a generator that can generate perturbation and position simultaneously. These methods either cost many queries or require additional time to train on a specific target identity. In addition, it has been shown that the shape of the patch has an impact on the attack. DAPatch [30] introduces a joint optimization for the shape and texture to enhance attack performance. However, it is a white-box attack and does not consider the patch location. With the above work, attack success rate is influenced by multiple factors. These existing methods suffer from two main shortcomings: (1) They focus on one or two influencing factors that are insufficient for better attack performance. These factors are interrelated but the previous methods could not optimize them simultaneously. (2) Since key pixels will change when the patch is updated, these methods struggle to use feedback from the model to guide the optimization of patch position and shape.

To address the above issues, a new Spatial Mutable Adversarial Patch (SMAP) method is proposed to optimize the texture, position and shape of the patch simultaneously and generate a dynamic mutable mask with the optimal patch in each iteration. Specifically, a Patch Location Selection Scheme (PLSS) is designed to search the critical patch position at the beginning of the attack. Then the position of the patch is dynamically change in each iteration according to the feedback to maintain its sensitivity. In this way, the adversarial transferability of the patch is effectively enhanced. Considering the effect of patch shape on attack performance, the patch is not restricted to a fixed shape. The deformation scheme is enabled to optimize the patch shape. Moreover, instead of using non-differentiable coordinate localization operations, the generation pipeline of SMAP is differentiable, making the texture, position and shape of the patch to be updated simultaneously. This paper will focus on evaluating the vulnerability of face recognition models under the black-box setting where model architectures and parameters are not accessible. Due to the limitation of the number of queries of the face recognition system, the surrogate model will be leveraged to generate patches that have adversarial transferability to the black-box models. Face attacks are divided into dodging (untargeted) attacks and impersonation (targeted) attacks according to the attack's goal. The more threatening impersonation attack is chosen, which generates an adversarial face to be recognized as the target identity.

The contributions can be summarized below:

- Aiming to simultaneously update the perturbation and mask, a new SMAP is proposed. By innovatively bridging the pre-defined mask and the dynamic update of the patch position and shape, SMAP maintains the sensitivity of the patch to the adversarial objective.
- We verify that different face recognition models generally focus on the same regions of the face. The PLSS is designed to find the critical patch position, and black-box transferability is significantly improved by perturbing the pixels in this region.
- We illustrate that perturbation texture, position and shape of the patch are equally important. The joint optimization of all three factors allows us to generate optimal adversarial patches end-to-end.
- Extensive experiments are conducted on well-known datasets in impersonation task and are evaluated in a strict evaluation metric that aligns with the real-world scenario. The results on state-of-the-art (SOTA) face recognition models and commercial face recognition systems show that the proposed method can improve success rate and transferability.

The remainder of the paper is organized as follows. Sec. II reviews the literature related to adversarial patch. In Sec. III, the formulation of the adversarial patch on face recognition is given. Then how to generate spatial mutable adversarial patch end-to-end is presented. The framework of the proposed method is illustrated in Sec. IV. Sec. V evaluates the performance of the proposed method on open-source and commercial face recognition models. Finally, the paper is concluded in Sec. VI.

II. Related Work

A. Adversarial Patch

Adversarial attacks [8], [31], [32] usually add global L_2 or L_∞ norm-based perturbations to the image. These perturbations are concealed and challenging for the human eye to detect, but are sensitive to the neural networks. In contrast to imperceptible attacks, adversarial patch focuses on practicability. The perturbation of this attack is limited to a small local region, but has a large enough magnitude to be captured by the cameras. As a result, it has been used in many realistic scenarios, such as person detectors [33], object detectors [33], [34] and face recognition [20], [21].

Earlier methods usually concern the texture of the adversarial patch, so they use random or fixed patch positions. Brown et al. [17] presented a method to create universal adversarial patches in the real world. Karmon et al. [18] proposed LaVAN to generate smaller, localized patches that can be applied to different images and locations. TnTs [12] proposed to generate universal adversarial patches that are robust, naturalistic and less malicious-looking by a pre-trained generator.

Some methods try to optimize the patch's location to increase the success rate. PS-GAN [26] captures the attention map of the attacked network to determine the critical areas. Wu et al. [27] proposed ROA to use exhaustive or gradient-guided search to find the patch location. Patch-Fool [35] uses the saliency map to guide the patch selection. However, these pre-designed positions are only significant at the beginning of the attack, and the key positions may change during the iteration. Rao et al. [28] moved the patch in a set of candidate directions with a fixed stride to maximize adversarial loss in each iteration. This method has to attempt a large number

of search directions, which causes greater computation. Li and Ji [29] proposed GDPA to train a generator to generate patch texture and location altogether. However, it depends on the target identity images and target model to train the generator and is a white-box attack that is difficult to apply to unknown black-box scenarios. The generator generalizes poorly and only generates positions and perturbations for a specific identity. It needs to retrain the generator when the target identity or target model changes. Compared to GDPA, our approach will be extended to generate perturbations for the known and unknown identities by simply modifying the input image of the identity and surrogate model without large-scale training. Chen et al. [30] optimize the patch shape and texture but do not consider the location.

B. Adversarial Patch on Face Recognition

Currently, face recognition is widely used in applications involving security and privacy, such as phone unlocking and face payment. The implementation of adversarial patch attacks raises security concerns for DNN-based face recognition. Sharif et al. [20] first proposed to generate adversarial eyeglasses to impersonate another identity or evade face recognition. Advhat [21] conducted the attack by wearing an adversarial hat. These patches are developed under the white-box setting. Dong et al. [36] proposed an evolutionary attack algorithm under the query-based black-box setting. Wei et al. [37] simultaneously optimize the patch position and perturbation using reinforcement learning with a few of queries.

Due to the excessive queries are not attainable for face recognition systems, recent works focus on transfer-based methods by leveraging surrogate models to generate adversarial samples. Adv-Makeup [38] developed a makeup generation method to synthesize imperceptible eye shadow and improved transferability by a fine-grained meta-learning strategy. Xiao et al. [16] regularized the adversarial patches on the low dimensional data manifold that is represented by generative models. However, the main limitation of these methods is that they focus on the texture of the patch, but do not consider the effect of position and shape on adversarial transferability. In addition, these methods struggle to dynamically update the mask in each iteration which makes it difficult for the patch to maintain sensitivity. In order to solve the problems discussed above and improve the transferability of the attack, the proposed SMAP will optimize the texture, position and shape of the patch simultaneously. And it will generate a dynamic mutable mask under the transfer-based black-box setting.

III. Spatial Mutable Adversarial Patch

The human face has its unique feature area making different face recognition models focus on it. Therefore, the success rate and transferability will be limited when generating perturbation with a fixed shape at a random or pre-designed location. It motivates us to generate perturbation in the optimal position to cover the critical feature area.

In this section, we formulate the adversarial patch on face recognition and then detail the proposed SMAP attack.

A. Problem Formulation

Let $f(x) : \mathcal{X} \rightarrow \mathbb{R}^k$ denote a face recognition model that inputs an image $x \in \mathcal{X}$ where $\mathcal{X}$ is a set of images of size $H \times W$ and x is in the range $[0, 255]$, and outputs a fixed-length vector in $\mathbb{R}^k$. This paper will focus on the impersonation attack against face recognition models where attackers need to generate an adversarial image that can be recognized as a target identity. Face recognition includes face verification and face identification. For face verification, the generated adversarial image can be identified as the same identity as another image, which is different from the identity of the source image. For face identification, the adversarial image is recognized as a specific identity in the face gallery search. Given a pair of face images $\{x_s, x_t\}$ where x_s is from the source identity and x_t is from the target identity, the proposed method aims to generate a patch that is placed on x_s to obtain image x_{adv} that can be misidentified as the identity of x_t by the face recognition model.

Formally, the perturbed face x_{adv} which is injected with an adversarial patch can be formulated as

$$x_{adv} = (1 - M) \odot x_s + M \odot \delta \tag{1}$$

where $M \in \{0, 1\}^{H \times W}$ is a binary mask used to locate the patch, $\odot$ is the element-wise product, δ is the perturbation.

Face recognition systems usually calculate the distance (e.g. Euclidean distance or cosine similarity) between two faces based on face features to obtain their similarity. When the similarity exceeds the pre-defined threshold of the model, the system determines that the two faces belong to the same identity. In this work, the distance between the two faces x_s and x_t is measured by calculating the cosine similarity of their feature vectors:

$$Sim_f(x_s, x_t) = \frac{f(x_s)^\top f(x_t)}{\|f(x_s)\|_2 \cdot \|f(x_t)\|_2} \tag{2}$$

where $\top$ is the transpose symbol.

Due to the limited number of queries in face recognition applications, the proposed method focuses on the transferability of adversarial examples. The features of different face images of the same identity are clustered in the feature space. Different face recognition models will compute the same feature clusters for face images of the same identity. Since the gradient information of the target model is not accessible, a white-box surrogate model f_w is used to generate images that can remain adversarial for the black-box model f_b. The surrogate model f_w will help us to compute the features of the face image and guide the generation of the adversarial image.

For the impersonation attack, the following optimization problem is solved to obtain an adversarial image:

$$\max_{\delta} \ \mathcal{L}_{adv}(x_{adv}, x_t) = Sim_{f_w}(x_{adv}, x_t) \tag{3}$$

where $\mathcal{L}_{adv}$ is an adversarial loss function.

The perturbation δ is updated with a small step α by maximizing the loss $\mathcal{L}_{adv}(x_{adv}, x_t)$ based on the Iterative Fast Gradient Sign Method (I-FGSM) [39]:

$$\delta^{k+1} = \delta^k + \alpha \cdot sign(\nabla_{\delta^k} \mathcal{L}_{adv}(x_{adv}^k, x_t)) \tag{4}$$

where k is the current number of iterations, $sign(\cdot)$ is the sign function.

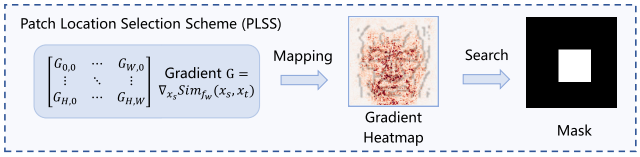

Fig. 1. Process of the proposed patch location selection scheme.

B. Strategy for Patch Generation

In order to generate a local patch, we need not only the texture of the patch, but also the location of the patch. The previous method usually initializes the patch position to the mask in a random or pre-designed way, *i.e.*, M in Eq. 1 is fixed. In order to improve attack performance, the position and shape of the patch as key factors also need to be optimized. Therefore, a Spatial Mutable Adversarial Patch is proposed to update M and δ in Eq. 1 simultaneously. The details of the mask update are shown below.

For face recognition, the feature region varies for different input face images. Therefore the adversarial performance of a patch with a random or fixed position is unsatisfactory. Moreover, for the same face, different face recognition models pay attention to generally the same area, which corresponds to the most critical features of the face (see Sec. V-E2). It means that black-box transferability is significantly improved by perturbing the pixels in this region.

In order to find this region, the model's feedback is utilized to obtain the patch location for source image x_s. The loss function is calculated and the gradients are obtained by backpropagation as:

$$G = \nabla_{x_s} Sim_{f_w}(x_s, x_t). \tag{5}$$

Then the optimal region is chosen by searching.

Specifically, the patch is assumed to be a square of size $N \times N$ and the search stride is set to S. Then $\lfloor (H - N)/S \rfloor \times \lfloor (W - N)/S \rfloor$ candidate regions are obtained. For each region in row i and column j where i is in range $[0, \lfloor (H - N)/S \rfloor)$ and j is in range $[0, \lfloor (W - N)/S \rfloor)$, the coordinate of the left-top corner $(h_{lt}^{(i,j)}, w_{lt}^{(i,j)})$ and right-bottom corner $(h_{rb}^{(i,j)}, w_{rb}^{(i,j)})$ are determined as

$$\begin{aligned}(h_{lt}^{(i,j)}, w_{lt}^{(i,j)}) &= (i \times S, j \times S)\\ (h_{rb}^{(i,j)}, w_{rb}^{(i,j)}) &= (i \times S + N, j \times S + N).\end{aligned} \tag{6}$$

Let G_k denote the gradient value for pixel k in image x_s. Then the absolute gradient values $|G_k|$ in every region are summed as the regional influence intensity value and select the optimal region with the largest intensity value. As shown in Fig. 1, the gradient heatmap represents the visualization of gradients and the mask locates the searched region.

Although the mask with the optimal patch location has been chosen based on the regional influence intensity value, the pixels in this patch region do not always play an important role in each iteration. There is no guarantee that the location of the patch selected as the most significant for the adversarial loss will remain the same. In this method with the pre-defined mask, only the perturbation δ is optimized and the update of the mask M is ignored, just like the previous method. It inspires us that the mask can be simultaneously optimized for better attack performance.

The coordinates of the patch in the mask are used to locate its position. However, it is not feasible to implement the patch change by directly defining the offset variables for the patch location and shape since this localization operation is not differentiable. *So how can the position and shape of the patch be updated simultaneously?*

In order to solve the above problem, the affine transformation [29], [40] is used on the mask to generate a sampling grid and then employ the sampling mechanism to output the mask with a new patch. It solves the problem of critical region changes. The position and shape of the patch are dynamically changed in each iteration based on the current gradient feedback. This way, the final optimal patch position can be obtained instead of the initial one. By innovatively bridging the pre-defined optimal position of the patch and the subsequent dynamic update of the mask, the position and shape of the patch can be optimized simultaneously with the perturbation in each iteration.

The operation is a 2D affine transformation and does not involve a perspective transformation. In this case, the transformation is:

$$\begin{aligned}Gd &= \begin{bmatrix} Gd_x^t \\ Gd_y^t \end{bmatrix} = affine(A_\theta, M^s)\\ &= A_\theta \begin{bmatrix} M_x^s \\ M_y^s \\ 1 \end{bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{bmatrix} M_x^s \\ M_y^s \\ 1 \end{bmatrix}\end{aligned} \tag{7}$$

where the grid Gd is the flow field (sampling grid), (Gd_x, Gd_y) are the coordinates of the grid Gd which specify the sampling pixel locations, A_θ is the affine transformation matrix, (M_x^s, M_y^s) are the coordinates of the mask M^s before the transformation. The function $affine(\cdot)$ refers to this process. In the transformation operation, the patch shape is not restricted. The transformation will allow translation, scale, and rotation to be applied to the patch. Therefore all 6 degrees of freedom of the matrix A_θ will be updated.

To compute the output M^t using known values of each coordinate in the input mask M^s and pixel locations from grid Gd, bilinear sampling is applied:

$$M^t = sample(Gd, M^s) \tag{8}$$

where $sample(\cdot)$ is the sampling operation.

Define M^{plss} as the mask obtained from the proposed PLSS algorithm. Based on the above strategy, the generation pipeline of mask M is denoted as

$$M = sample(affine(A_\theta, M^{plss}), M^{plss}) \tag{9}$$

The differentiable sampling mechanism will allow adversarial loss gradients can be backpropagated to the affine transformation matrix A_θ, which enables us to update the parameters θ.

The parameters for scale in the affine matrix A_θ are not fixed. The patch size affects the performance of the attack (See Sec. V-E2). The larger the patch area, the stronger the performance of the attack. In order to maximize the loss $\mathcal{L}_{adv}$, the parameters for scale will be updated in the direction that increases the patch area. This will cause the patch area to

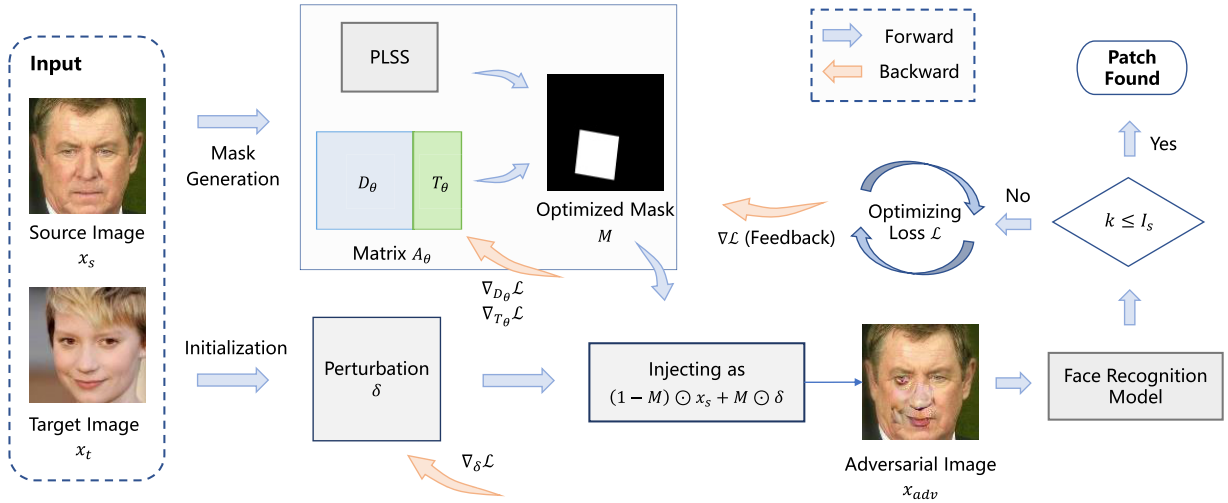

Fig. 2. The framework of the proposed SMAP. Patch Location Selection Scheme (PLSS) is applied to generate the initial mask. Affine transformation mechanism is introduced to dynamically update the location and shape of the patch based on the differentiable affine matrix. After injecting the patch, the adversarial image x_{adv} is generated and later fed into the face recognition model. The method performs the attack iteratively by optimizing the perturbation and affine matrix through gradient feedback.

increase continuously. Therefore, a loss $\mathcal{L}_{area}$ is introduced into the adversarial objective to constrain the patch area, and this loss function is calculated by

$$\mathcal{L}_{area}(M) = \sum_i M_i \tag{10}$$

where M_i is the pixel i in the mask. $\mathcal{L}_{area}$ is integrated into Eq. 3 and the adversarial objective is rewritten as:

$$\max_{\delta, A_\theta} \mathcal{L}(x_{adv}) = \begin{cases} \mathcal{L}_{adv}(x_{adv}, x_t) & area \leq T_{PA} \\ \mathcal{L}_{adv}(x_{adv}, x_t) - \lambda\mathcal{L}_{area}(M) & area > T_{PA} \end{cases} \tag{11}$$

where λ is a hyperparameter to balance these two losses, T_{PA} is the threshold that enables patch area stabilization. Compared with the previous method, our optimization objective is expanded from a single optimization of δ to a joint optimization of δ and A_θ.

In order to improve the success rate and transferability, the perturbation δ is initialized as x_t (see Sec. V-E1). After integrating the loss function $\mathcal{L}_{area}$, δ is updated as:

$$\delta^{k+1} = \delta^k + \alpha \cdot sign(\nabla_{\delta^k}\mathcal{L}(x^k_{adv})). \tag{12}$$

Since the parameters θ of the matrix A_θ control different transformations, θ will be updated using different strategies. The matrix A_θ is divided into two block matrices, where D_θ controls scale and rotation, T_θ controls translation:

$$A_\theta = \left[\, D_\theta \,|\, T_\theta \,\right] = \left[\begin{array}{cc|c} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{array}\right]. \tag{13}$$

In each iteration, block matrices D_θ and T_θ are separately updated in steps of γ_d and γ_t:

$$\begin{aligned} D^{k+1}_\theta &= D^k_\theta + \gamma_d \cdot sign(\nabla_{D^k_\theta}\mathcal{L}(x^k_{adv})) \\ T^{k+1}_\theta &= Clip_{[-1,1]}(T^k_\theta + \gamma_t \cdot sign(\nabla_{T^k_\theta}\mathcal{L}(x^k_{adv}))). \end{aligned} \tag{14}$$

As shown in Fig. 3, the position and shape of the patch will be transformed based on the affine matrix. Therefore, by optimizing the D_θ and T_θ, spatial mutable adversarial patches can be obtained.

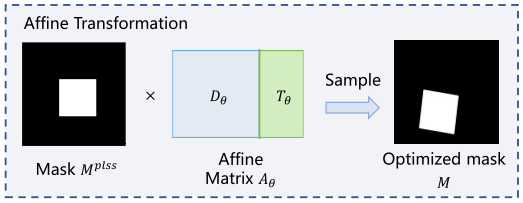

Fig. 3. Diagram of the affine transformation of the mask.

IV. Overall Framework

Based on the selection scheme of the mask and the process of simultaneous optimization, the overall framework of the proposed method is designed and illustrated in Fig. 2. And the steps are as follows:

Step 1: Select a source face image x_s and a face image x_t of the target identity from the dataset. Feed them into the selected white-box surrogate face recognition model to calculate the similarity and the gradient of the similarity. The critical region is located to generate the initial optimal mask M^{plss} with the proposed PLSS. Note that SMAP uses the target face image to initialize the perturbation *i.e.* $\delta^0 \leftarrow x_t$.

Step 2: Parameters in block matrices D_θ and T_θ are initialized as $\theta_{11} = 1, \theta_{12} = 0, \theta_{13} = 0, \theta_{21} = 0, \theta_{22} = 1, \theta_{23} = 0$. In order to stabilize the patch location update, it is necessary to decay the steps γ_d and γ_t. Therefore the learning rate schedule is applied and θ is stopped updating when reaching the $I_s - th$ iteration. The maximum number of iterations is set to I_m. In the first phase of the iteration, block matrices D_θ and T_θ are updated as in Eq. 14. The mask M is optimized by the differentiable affine transformation and sampling to maintain the location and shape sensitivity as in Eq. 9. When the parameters of the affine matrix converge, M stops updating at the $I_s - th$ iteration. Since the interpolation technique introduces small noise burr during the patch update,

Algorithm 1 Spatial Mutable Adversarial Patch (SMAP)

Input: A source face image x_s; a face image x_t of the target identity; patch size N; stride S; perturbation step α; balance parameter λ; transformation step γ_d and γ_t; patch area threshold T_{PA}; the number of iterations for updating I_s; the total number of iterations I_m.
Input: A face recognition model f_w.
Output: Adversarial image x_{adv}.

Initialize $\delta^0 \leftarrow x_t$;
Initialize D_θ^0 and T_θ^0;
Obtain M^{plss} according to PLSS;
for k in range $(0, I_m)$ **do**
  if $k \leq I_s$ **then**
    $M = sample(A_\theta^k M^{plss})$;
  else
    $M = deburr(M^{I_s})$;
  end if
  $x_{adv}^k = Clip((1-M) \odot x_s + M \odot \delta^k)$;
  Obtain the gradients of the loss $\mathcal{L}(x_{adv}^k)$;
  Update δ^{k+1} as in Eq. 12;
  if $k \leq I_s$ **then**
    Update D_θ^{k+1} and T_θ^{k+1} as in Eq. 14;
    $\gamma_d.step()$, $\gamma_t.step()$;
  end if
end for
return $x_{adv} = Clip((1-M) \odot x_s + M \odot \delta^{I_m})$;

the noise is removed when the first update is finished, which is called *deburr* operation. Therefore, M is obtained in the second phase of the iteration as

$$M = deburr(M^{I_s}). \tag{15}$$

Step 3: With the mask generated, the adversarial face image x_{adv} is obtained as in Eq. 1. And then it is constrained in range $[0, 255]$ by $Clip(\cdot)$ which is the function that performs pixel clipping of the image. The adversarial perturbation δ is continuously optimized with the gradients of the loss function in both two phase iterations as in Eq. 12. The spatial mutable adversarial patch can be generated after successive iteration. Finally, the adversarial image can be used to evaluate the vulnerability of face recognition systems.

The complete process of the proposed method is depicted in Algorithm 1.

V. Experiments and Analysis

In this section, the performance of the proposed SMAP will be evaluated. The experimental setting is described in Sec. V-A. Then the comparison with other methods is presented in Sec. V-D and Sec. V-B. Sec. V-C presents the results on commercial face recognition systems. Sec. V-E performs ablation studies to show the effectiveness of our method. Sec. V-F shows the visualization.

A. Experimental Setup

1) Compared Methods: The performance of the proposed SMAP is compared with existing patch attack methods, including GenAP [16], ROA [27], Patch-Fool [35], Adv-Glasses [20] and Adv-Hat [21], Patch-RL [37] and TnTs [12]. The proposed parameter settings are used in their configurations, and we try to improve their performance in our evaluate metric.

2) Types of Face Recognition: Experiments are conducted in two types of face recognition: similarity-based and classification-based face recognition. For the former, the feature will be output based on the face input. For the latter, the class will be output based on the face input.

3) Datasets: For similarity-based face recognition, two face datasets are used: Labeled Faces in the Wild (LFW) [41] and Large-scale CelebFaces Attributes High Quality (CelebA-HQ) [42]. LFW contains a total of 13233 low-quality face images. CelebA-HQ is a high-quality dataset generated by ProGAN [42]. We increase the difficulty of adversarial attacks. 300 pairs of images from different identities are selected for the face verification task. Moreover, another 300 images of the target identity are selected for evaluation. For the face identification task, we select 300 images of 300 different identities as the gallery set and corresponding 300 images of the same identities to form a probe set. And an identity is randomly chosen in the probe set as the target identity.

For classification-based face recognition, VGGFace dataset [43] is used. We choose ten subjects and one of the subjects is selected as the target class. A total of 150 pairs of images are sampled where x_t is from the target subject and x_s is from the rest subjects.

4) Face Recognition Models: For similarity-based face recognition, four SOTA face recognition models are used: ArcFace [5], CosFace [6], SphereFace+ (SFace+) [44], SphereFaceR (SFaceR) [45]. These face recognition models all use excellent neural networks as the backbone, such as iResNet [5], SFNet [45]. Among them, ArcFace uses iResNet-50 as the backbone, CosFace and SphereFaceR use iResNet-100, and SphereFace+ uses SFNet-64. In addition, they incorporate a form of angular-margin-based loss to introduce angle information in the feature space and make learned features separable with a larger angular distance. For these four models, the input image size is 112×112 and the output feature size is 512. The features output by the backbone are used to calculate similarity. In order to obtain the threshold of these four models, we do the calculation on the LFW validation set which contains 3000 pairs of images from the same identity and different identities respectively. The optimal threshold that gives the highest accuracy is chosen.

For classification-based face recognition, VGGFace model [43] is based on the VGGNet [46] and fine-tuned by Triplet loss [47] which minimizes the distance of positive samples and maximizes the distance of negative samples. For VGGFace in our experiments, the input image size is 224×224 and the output size is 10. The output class determines the identity to which the face image belongs.

In transfer-base attacks, it belongs to a white-box attack when the surrogate model is the same as the test model. On the contrary, it belongs to a black-box attack when the surrogate model differs from the test model.

5) Evaluation Metric: For similarity-based face recognition, the cosine distance between the adversarial image and the target image are measured and compared with the threshold. For face verification, it is considered a success when the

TABLE I

THE SUCCESS RATES OF DIFFERENT PATCH ATTACK METHODS UNDER FACE VERIFICATION TASK. THE ADVERSARIAL PATCHES ARE GENERATED ON ARCFACE, COSFACE, SFACE+ AND SFACER RESPECTIVELY AND EVALUATED ON DIFFERENT MODELS. THE SURROGATE MODELS ARE LISTED IN THE ROWS, AND THE TEST MODELS ARE IN THE COLUMNS. * INDICATES WHITE-BOX ATTACKS. HIGHER SUCCESS RATES ARE IN BOLD

Model	Method	CelebA-HQ				LFW				Average
		ArcFace	CosFace	SFace+	SFaceR	ArcFace	CosFace	SFace+	SFaceR	(black-box)
ArcFace	GenAP [16]	0.7267*	0.3233	0.3667	0.2400	0.8433*	0.3133	0.3767	0.2167	0.3061
	ROA [27]	0.9067*	0.3767	0.4633	0.2933	**0.9933***	0.3433	0.5100	**0.2700**	0.3761
	Patch-Fool [35]	0.7467*	0.1167	0.1900	0.1033	0.8300*	0.0700	0.1967	0.0733	0.1250
	SMAP (ours)	**0.9133***	**0.4500**	**0.5400**	**0.3567**	0.9867*	**0.3533**	**0.5500**	**0.2700**	**0.4200**
CosFace	GenAP [16]	0.3967	0.7133*	0.4300	0.2733	0.3733	0.8267*	0.4300	0.2767	0.3633
	ROA [27]	0.5233	0.9067*	0.5000	0.3400	0.4400	**0.9867***	0.4633	0.3000	0.4277
	Patch-Fool [35]	0.1667	0.7000*	0.1767	0.1067	0.1233	0.7633*	0.1467	0.0833	0.1339
	SMAP (ours)	**0.5467**	**0.9100***	**0.5700**	**0.4167**	**0.4567**	0.9733*	**0.5367**	**0.3167**	**0.4739**
SFace+	GenAP [16]	0.2033	0.1933	0.7667*	0.2033	0.1733	**0.1633**	0.9033*	0.1967	0.1888
	ROA [27]	0.2667	0.2033	**0.8933***	0.3033	0.1900	0.1100	**0.9833***	0.2133	0.2144
	Patch-Fool [35]	0.0833	0.0500	0.7500*	0.0867	0.0500	0.0267	0.8467*	0.0467	0.0572
	SMAP (ours)	**0.3067**	**0.2167**	0.8867*	**0.3400**	**0.2433**	0.1367	**0.9833***	**0.2633**	**0.2511**
SFaceR	GenAP [16]	0.2567	0.2600	0.4167	0.7133*	0.2600	0.2667	0.4400	0.7767*	0.3167
	ROA [27]	0.3333	0.3200	0.4900	0.8900*	0.2700	0.2367	0.5500	**0.9900***	0.3667
	Patch-Fool [35]	0.1133	0.0600	0.2100	0.6800*	0.0667	0.0267	0.1767	0.7833*	0.1089
	SMAP (ours)	**0.3767**	**0.3433**	**0.5433**	**0.8967***	**0.3200**	**0.2900**	**0.5967**	0.9800*	**0.4116**

similarity of the adversarial image and the second image of the target identity exceeds the threshold. For face identification, it succeeds when the similarity of the adversarial image generated by image pair in the probe set with the target image exceeds the similarity with other images in the gallery set and the threshold. This evaluation is more realistic because the image of the target identity in the face database is not accessible. For classification-based face recognition, it is considered a success when the adversarial image is is classified as the target class.

The attack success rate (the number of successful attacks devided by the total number of image pairs) is used to evaluate the effectiveness of an attack. The average black-box success rate is calculated to evaluate the transferability of an attack.

6) Implementation Details: For a fair comparison, the patch area is limited to 14.3% of the image pixels in all experiments. We set the number of iterations I_m to 255, I_s to 200, stride S to 5, perturbation step α to 1/255, balance parameter λ to 0.0001, transformation step $\gamma_d = \gamma_t$ are in the range [0.05, 0.02, 0.005, 0.002] which are updated every 50 iterations. In order to prevent the patch area from slightly exceeding T_{PA} at the last update of I_s, the threshold T_{PA} is set smaller than the patch area to ensure fairness. The codes are implemented in PyTorch. All experiments are conducted on NVIDIA GeForce RTX 3090 GPUs.

B. Experiments on Similarity-Based Face Recognition

We experimented with two types of black-box methods, including transfer-based and query-based. The transfer-based methods includes GenAP [16], ROA [27], Patch-Fool [35], Adv-Glasses [20] and Adv-Hat [21]. GenAP [16] uses the eyeglass region to generate the patches. The gradient-based version of ROA is used. For the query-based method Patch-RL [37], we limit the number of queries to 10, since excessive queries are not attainable.

For fair comparison, the perturbations of ROA, Patch-Fool and Patch-RL are initialized with the target face x_t instead of a zero or random initialization. The perturbation magnitude δ is not constrained among all experiments but image pixels are constrained in [0, 255].

1) Comparison With Transfer-Based Methods: We first conduct the comparison of methods for generating human-like face patches that can be pasted on the face including GenAP, ROA and Patch-Fool. The success rates of four attacks against the models under the face verification task are presented in Tab. I and under the face identification task in Tab. II.

For face verification, the similarity between the adversarial face and the face of the target identity needs to exceed the threshold. From Tab. I, it is observed that the success rates of GenAP are not satisfactory due to the fixed patch position. ROA finds a position with maximum loss and then applies PGD [31] for pixels inside the rectangle. Therefore, ROA achieves a higher success rate with a better patch location than GenAP. Due to the shortcomings of the optimization method, Patch-Fool method generates samples with low success rate and poor transferability. By overcoming the weakness that key locations may change during iterations and integrating the deformation scheme, the proposed method outperforms other methods in most cases. The proposed SMAP has strengths in black-box success rate. For example, the average black-box success rate of SMAP on ArcFace surrogate model is 42% which achieves 11.39%, 4.39% and 29.5% improvement compared to GenAP, ROA and Patch-Fool. We notice that ROA method has higher white-box success rates than SMAP in some cases, but low black-box success rates. For example, ROA achieves a success rate of 98.67% using CosFace and 99.00% using SFaceR on LFW, but its black-box success rates are both lower than SMAP. It indicates that ROA only focuses on optimizing the texture of the patch at a pre-defined location making the generated patch overfit the surrogate model. In contrast, the patches generated by our balanced optimization of texture, position and shape achieve better performance on different face recognition methods, showing the effectiveness in improving transferability.

Tab. II shows the attack results on the face identification task. The generated adversarial face needs to be recognized as the target identity in the face image gallery. The similarity

TABLE II

THE SUCCESS RATES OF DIFFERENT PATCH ATTACK METHODS UNDER FACE IDENTIFICATION TASK. THE ADVERSARIAL PATCHES ARE GENERATED ON ARCFACE, COSFACE, SFACE+ AND SFACER RESPECTIVELY. THE SURROGATE MODELS ARE LISTED IN THE ROWS, AND THE TEST MODELS ARE IN THE COLUMNS. * INDICATES WHITE-BOX ATTACKS. HIGHER SUCCESS RATES ARE IN BOLD

Model	Method	CelebA-HQ				LFW				Average
		ArcFace	CosFace	SFace+	SFaceR	ArcFace	CosFace	SFace+	SFaceR	(black-box)
ArcFace	GenAP [16]	0.5533*	0.1900	0.2567	0.1700	0.6000*	**0.1767**	0.2600	0.1600	0.2022
	ROA [27]	**0.8933***	0.2600	0.3400	0.2467	**0.9600***	0.1533	0.3433	0.1867	0.2550
	Patch-Fool [35]	0.5400*	0.0367	0.0933	0.0533	0.5100*	0.0033	0.0367	0.0167	0.0400
	SMAP (ours)	0.8767*	**0.3000**	**0.4367**	**0.2767**	0.9433*	0.1633	**0.3800**	**0.2033**	**0.2933**
CosFace	GenAP [16]	0.2033	0.5533*	0.3167	0.2167	0.2167	0.6633*	0.3033	0.2033	0.2433
	ROA [27]	0.3467	**0.8933***	0.4033	0.3067	0.1900	**0.9500***	0.2867	0.2033	0.2894
	Patch-Fool [35]	0.0733	0.5267*	0.0800	0.0500	0.0100	0.4667*	0.0333	0.0100	0.0427
	SMAP (ours)	**0.3733**	**0.8933***	**0.4400**	**0.3400**	**0.2433**	0.9400*	**0.3467**	**0.2233**	**0.3277**
SFace+	GenAP [16]	0.0867	0.0833	0.6933*	0.1400	0.0600	**0.0800**	0.8033*	0.1300	0.0967
	ROA [27]	0.1567	0.1000	**0.8867***	0.2400	0.0733	0.0367	**0.9733***	0.1267	0.1222
	Patch-Fool [35]	0.0333	0.0100	0.5867*	0.0467	0.0000	0.0033	0.5833*	0.0100	0.0172
	SMAP (ours)	**0.1800**	**0.1267**	0.8667*	**0.2633**	**0.0833**	0.0367	0.9567*	**0.1633**	**0.1422**
SFaceR	GenAP [16]	0.1267	0.1433	0.2833	0.6533*	**0.1433**	**0.1367**	0.3167	0.7033*	0.1916
	ROA [27]	0.2033	0.1900	0.3733	0.8833*	0.1033	0.1033	0.3833	**0.9767***	0.2260
	Patch-Fool [35]	0.0333	0.0100	0.0800	0.5400*	0.0100	0.0033	0.0400	0.5767*	0.0294
	SMAP (ours)	**0.2267**	**0.2200**	**0.4233**	**0.8900***	0.1100	0.1033	**0.4167**	0.9667*	**0.2500**

TABLE III

THE SUCCESS RATES OF DIFFERENT PATCH ATTACK METHODS ON CELEBA-HQ. THE ADVERSARIAL PATCHES ARE GENERATED ON ARCFACE AND EVALUATED ON DIFFERENT MODELS. * INDICATES WHITE-BOX ATTACKS. HIGHER SUCCESS RATES ARE IN BOLD

Method	ArcFace	CosFace	SFace+	SFaceR
Face Verification				
Adv-Glasses [20]	**1.0000***	0.0233	0.0467	0.0200
Adv-Hat [21]	0.3700*	0.0000	0.0100	0.0300
SMAP (ours)	0.9133*	**0.4500**	**0.5400**	**0.3567**
Face Identification				
Adv-Glasses [20]	0.1367*	0.0000	0.0000	0.0000
Adv-Hat [21]	0.0600*	0.0000	0.0100	0.0100
SMAP (ours)	**0.8767***	**0.3000**	**0.4367**	**0.2767**

TABLE IV

THE SUCCESS RATES OF DIFFERENT PATCH ATTACK METHODS ON LFW. THE ADVERSARIAL PATCHES ARE GENERATED ON ARCFACE AND EVALUATED ON DIFFERENT MODELS. * INDICATES WHITE-BOX ATTACKS. HIGHER SUCCESS RATES ARE IN BOLD

Method	ArcFace	CosFace	SFace+	SFaceR
Face Verification				
Adv-Glasses [20]	**1.0000***	0.0100	0.0433	0.0200
Adv-Hat [21]	0.5100*	0.0000	0.0200	0.0600
SMAP (ours)	0.9867*	**0.3533**	**0.5500**	**0.2700**
Face Identification				
Adv-Glasses [20]	0.1633*	0.0000	0.0000	0.0000
Adv-Hat [21]	0.0900*	0.0000	0.0000	0.0100
SMAP (ours)	**0.9433***	**0.1633**	**0.3800**	**0.2033**

between the generated face and the face of the target identity needs to exceed not only the threshold, but also the similarity with another face of the source identity. Because of the more challenging attack objective, the success rates of these methods drop by almost half (Patch-Fool even more) compared to the face verification task. At the same time, the proposed SMAP still maintains the highest success rate in most cases. In addition, our approach maintains better black-box success rates. For example, the average black-box success rate of SMAP on CosFace model reaches 32.77%, which is 8.44%, 3.83% and 28.5% higher than the results of GenAP, ROA and Patch-Fool. Moreover, same as face verification task, ROA method has a high white-box success rate in some cases, showing the shortcoming of its poor transferability. It is observed that SMAP and ROA perform poorly against CosFace on LFW compared to GenAP. The reason can be attributed to the fact that the deep features generated by GenAP on the LFW are more deceptive to CosFace.

Then we conduct the comparison of methods for generating wearable patches including Adv-Glasses and Adv-Hat. The success rates on CelebA-HQ are presented in Tab. III and those on LFW are shown in Tab. IV. It can be seen that Adv-Glasses achieves a 100% white-box success rate on both datasets under face verification task. But the image generated by Adv-Glasses maintains a high similarity to the target image, making it perform poorly under face identification task. Adv-Hat suffers from a low success rate due to the limitation of the hat template. In addition, both methods lack transferability and perform poorly on black-box attacks. Our approach achieves high transferability through joint optimization of the patch texture, position and shape and better patch initialization.

2) Comparison With Query-Based Methods: For Patch-RL, we make some special arrangements. First, the attack is launched using the surrogate model, the first image of the source and target identities. Then, the second image of the source and target identities are used for querying the target model to obtain rewards for policy update. Such arrangements more closely align with the querying in real-world scenarios.

The results are shown in Tab. V. It can be seen that when ArcFace or CosFace are used as the surrogate model and Sphereface+ or SphereFaceR as the query model, Patch-RL performs well. However, the performance is poor when the roles are reversed. The performance of Patch-RL relies on the model and is very sensitive to the choice of model. In addition, Patch-RL performs better in white-box attacks. This can be attributed to Patch-RL launching an attack directly after determining the position, while our method updates the location of the patch during iterations, which reduces the attack effect of previously accumulated perturbation. Undeniably, querying the target model will bring better attack performance. However, our method exhibits more stable performance across different models.

TABLE V

THE COMPARISON OF SUCCESS RATES WITH PATCH-RL. THE SURROGATE MODELS ARE LISTED IN THE ROWS, AND THE TARGET MODELS ARE IN THE COLUMNS. QUERY MEANS THE QUERY-BASED ATTACK, AND TRANSFER MEANS THE TRANSFER-BASED ATTACK. * INDICATES WHITE-BOX ATTACKS. HIGHER SUCCESS RATES ARE IN BOLD

Model	Method	Attack Type	CelebA-HQ				LFW				Average (black-box)
			ArcFace	CosFace	SFace+	SFaceR	ArcFace	CosFace	SFace+	SFaceR	
Face Verification											
ArcFace	Patch-RL [37]	Query	**0.9200***	**0.5533**	**0.5733**	**0.5733**	**0.9967***	**0.4733**	**0.6233**	**0.5467**	**0.5572**
	SMAP (ours)	Transfer	0.9133*	0.4500	0.5400	0.3567	0.9867*	0.3533	0.5500	0.2700	0.4200
CosFace	Patch-RL [37]	Query	0.2600	0.9067*	0.4133	**0.4567**	0.1900	**0.9967***	0.4333	**0.3967**	0.3583
	SMAP (ours)	Transfer	**0.5467**	**0.9100***	**0.5700**	0.4167	**0.4567**	0.9733*	**0.5367**	0.3167	**0.4739**
SFace+	Patch-RL [37]	Query	0.1633	**0.2267**	**0.9033***	**0.4800**	0.1200	**0.1867**	**0.9967***	**0.4733**	**0.2750**
	SMAP (ours)	Transfer	**0.3067**	0.2167	0.8867*	0.3400	**0.2433**	0.1367	0.9833*	0.2633	0.2511
SFaceR	Patch-RL [37]	Query	0.0800	0.1500	0.3067	**0.9100***	0.0567	0.0900	0.3633	**0.9967***	0.1744
	SMAP (ours)	Transfer	**0.3767**	**0.3433**	**0.5433**	0.8967*	**0.3200**	**0.2900**	**0.5967**	0.9800*	**0.4116**
Face Identification											
ArcFace	Patch-RL [37]	Query	**0.9167***	0.2967	**0.4500**	**0.3767**	**0.9933***	**0.1667**	**0.4400**	**0.2400**	**0.3283**
	SMAP (ours)	Transfer	0.8767*	**0.3000**	0.4367	0.2767	0.9433*	0.1633	0.3800	0.2033	0.2933
CosFace	Patch-RL [37]	Query	0.2000	**0.9133***	0.3500	**0.3500**	0.1000	**0.9967***	0.3267	**0.2633**	0.2650
	SMAP (ours)	Transfer	**0.3733**	0.8933*	**0.4400**	0.3400	**0.2433**	0.9400*	**0.3467**	0.2233	**0.3277**
SFace+	Patch-RL [37]	Query	0.0900	0.1000	**0.9033***	**0.3033**	0.0367	**0.0433**	**0.9967***	**0.1633**	0.1233
	SMAP (ours)	Transfer	**0.1800**	**0.1267**	0.8667*	0.2633	**0.0833**	0.0367	0.9567*	**0.1633**	**0.1422**
SFaceR	Patch-RL [37]	Query	0.0700	0.1033	0.2867	**0.9100***	0.0167	0.0367	0.3233	**0.9933***	0.1394
	SMAP (ours)	Transfer	**0.2267**	**0.2200**	**0.4233**	0.8900*	**0.1100**	**0.1033**	**0.4167**	0.9667*	**0.2500**

TABLE VI

THE COMPARISON OF SUCCESS RATES WITH PATCH-RL. THE TARGET MODELS ARE IN THE COLUMNS. THE QUERY MODEL FOR PATCH-RL IS ARCFACE. QUERY MEANS THE QUERY-BASED ATTACK, AND TRANSFER MEANS THE TRANSFER-BASED ATTACK. HIGHER SUCCESS RATES ARE IN BOLD

Method	Attack Type	CelebA-HQ			LFW			Average
		CosFace	SFace+	SFaceR	CosFace	SFace+	SFaceR	
Face Verification								
Patch-RL [37]	Query	0.5533	0.5733	0.5733	0.4733	0.6233	0.5467	0.5572
SMAP-Ensemble (ours)	Transfer	**0.6733**	**0.6833**	**0.6467**	**0.6500**	**0.7800**	**0.6967**	**0.6883**
Face Identification								
Patch-RL [37]	Query	0.2967	0.4500	0.3767	0.1667	0.4400	0.2400	0.3283
SMAP-Ensemble (ours)	Transfer	**0.5400**	**0.6067**	**0.6033**	**0.4433**	**0.6433**	**0.6000**	**0.5727**

TABLE VII

FID, LPIPS AND SSIM SCORES OF DIFFERENT METHODS GENERATED ON ARCFACE. THE BETTER SCORES ARE IN BOLD

Method	Source Image → Adversarial Image					
	CelebA-HQ			LFW		
	FID	LPIPS	SSIM	FID	LPIPS	SSIM
GenAP [16]	**67.32**	0.1299	0.8595	96.80	0.1871	0.8598
ROA [27]	80.04	0.0996	0.8731	107.79	0.1545	0.8750
Patch-Fool [35]	78.39	0.1031	0.8671	**91.68**	0.1423	0.8708
Adv-Glasses [20]	164.02	0.2523	0.8223	177.41	0.3199	0.8227
Adv-Hat [21]	116.69	0.1495	0.8588	145.50	0.1891	0.8643
Patch-RL [37]	158.04	0.2224	0.8399	202.59	0.3162	0.8375
SMAP (ours)	77.50	**0.0888**	**0.8822**	110.54	**0.1377**	**0.8850**

As these two methods are different types of attacks, Patch-RL queries the target model to update model parameters which achieves a higher success rate than ours in some situations. Therefore, in order to make the comparison more fair, we perform ensemble attacks for our SMAP. When one model is set as the black-box model, the remaining three are designated as surrogate models. For Patch-RL, we select its best-performing data on black-box attack for comparison, which is when the surrogate model is Arcface. The results are shown in Tab. VI. After the ensemble attacks, our SMAP performs better and is more stable than Patch-RL against different models.

3) Image Quality Assessment: The adversarial examples generated by different methods against ArcFace under the impersonation attack task are visualized in Fig. 4. Specifically, samples are selected from CelebA-HQ and LFW, and Arcface is used as a surrogate model. It can be seen that the patches generated by SMAP are more focused on the critical areas of the face. In addition, compared to other methods, the images generated by our SMAP appear more natural.

We calculate Frechet Inception Distance (FID) [48], Learned Perceptual Image Patch Similarity (LPIPS) [49] and Structural Similarity (SSIM) [50] to evaluate the quality of images. These three metrics are used to measure perceptual similarity between images. For FID and LPIPS, the lower the score, the more similar the images. For SSIM, the higher the score, the lower the image distortion. As shown in Tab. VII, our SMAP ranks relatively high in both two datasets for FID, and performs the best for LPIPS and SSIM.

C. Experiments on Commercial Systems

To evaluate the transferability of SMAP, the patch attacks are conducted on two commercial face recognition systems,

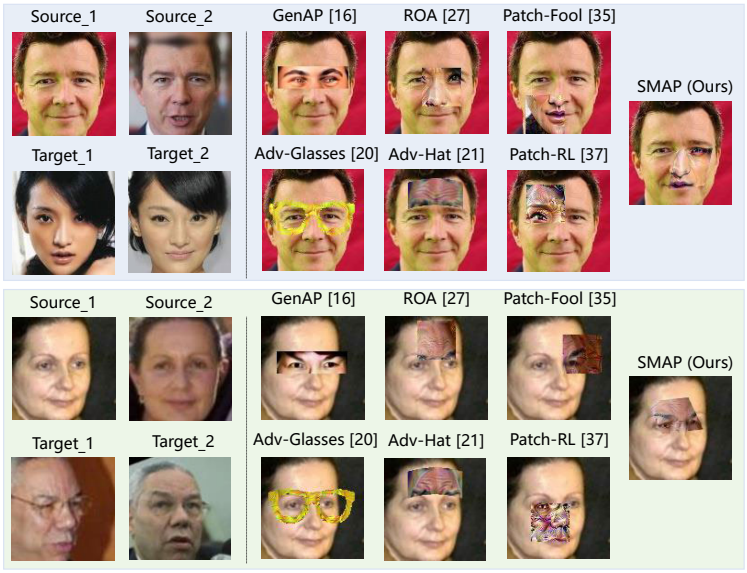

Fig. 4. Examples generated by different adversarial patch methods against ArcFace under the impersonation attack task. The gallery above is from the CelebA-HQ and the one below is from the LFW. Source_1 and Target_1 are used for image generation. Source_2 and Target_2 are used for evaluation.

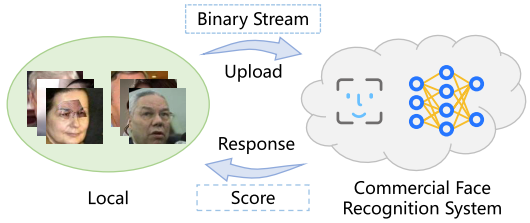

Fig. 5. Process of face verification on the commercial system. It allows us to upload two face images and then output a score indicating the similarity.

Tencent Cloud (TCloud)[3] from Tencent and Face++[4] from Megvii. As shown in Fig. 5, the binary stream of another face of the target identity and the adversarial face generated by different methods using surrogate models are uploaded. Then the response is obtained to evaluate the performance of these methods. The threshold with a false recognition rate of one ten thousandth is used to measure success rates, which is given by these two commercial face recognition systems.

The results are shown in Tab. VIII and IX. Note that since the commercial face recognition model is not accessible, we set the surrogate model and the target model to be the same and evaluate them on the commercial models for Patch-RL. It is seen that the proposed method outperforms GenAP on Face++ and Patch-Fool, Patch-Rl, Adv-Glasses and Adv-Hat on two commercial systems by a large margin, and the adversarial images generated by our method have more stable transferability. For example, the proposed SMAP achieves the highest average success rate of 31.08% using ArcFace model against the commercial models, which is 15.25%, 3.42%, 21.08% and 20.75% higher than those of GenAP, ROA, Patch-Fool and Patch-RL. Due to the inability to query commercial models, the performance of Patch-RL decreases. In addition, SMAP achieves the highest average success rate of 30.16% using ArcFace model against the commercial models, which is 27.49% and 28.16% higher than Adv-Glasses and Adv-Hat. These results illustrate not only the better performance of SMAP on open-source models but also the effectiveness to commercial models compared to other methods.

In addition, there is not much difference in the success rate of the adversarial faces generated using surrogate models with different architectures against unknown commercial models. It shows the vulnerability of real-world face recognition models which will bring security risks.

[3]https://www.tencentcloud.com/products/facerecognition

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

D. Experiments on Classification-Based Face Recognition

To evaluate the performance on classification-based face recognition, we compare with TnTs [12] on VGGFace model which is fine-tuned on the face images of the sampled 10 subjects. Due to the change of the adversarial objective from feature similarity to the target class, $\mathcal{L}_{adv}$ in Eq. 11 is replaced by the cross-entropy loss such that the adversarial face x_{adv} can be classified as target class by the model. For classification-based face recognition, the index of the maxima of the output of the fully connected (FC) layer is usually considered as the result. There is no threshold to evaluate the quality of the adversarial face. Therefore, the output of the FC layer is fed into the softmax function to calculate the confidence. Two experiments are conducted: (1) It is considered a success when confidence is greater than 0.9 (Conf $>$ 0.9). (2) It is considered a success as long as the confidence is the highest (Unlimited).

The results in Tab. X demonstrate that SMAP performs better compared to TnTs in both evaluation metrics. This can be attributed to the fact: (1) SMAP perturbs the critical regions of the face, while TnTs generate a patch at the lower-right corner. (2) Naturalistic and less malicious-looking flower patch in TnTs is difficult to convert to facial feature and mislead the classification network compared to face-like patch. It indicates that the texture representation and position of the patch is crucial for the face adversarial attack. Fig. 6 shows some visual examples generated by these two methods. It is noticeable that the adversarial patches generated by SMAP have the features

TABLE VIII

THE SUCCESS RATES OF DIFFERENT PATCH ATTACK METHODS ON THE COMMERCIAL FACE RECOGNITION SYSTEMS. THE ADVERSARIAL PATCHES ARE GENERATED ON ARCFACE, COSFACE, SFACE+ AND SFACER RESPECTIVELY, AND THEN ARE EVALUATED ON COMMERCIAL MODELS. HIGHER SUCCESS RATES ARE IN BOLD

Model	Method	CelebA-HQ		LFW		Average
		TCloud	Face++	TCloud	Face++	
ArcFace	GenAP [16]	0.1400	0.1567	0.2033	0.1333	0.1583
	ROA [27]	0.1533	0.3733	0.2233	0.3567	0.2766
	Patch-Fool [35]	0.0700	0.1533	0.0800	0.0966	0.1000
	Patch-RL [37]	0.0367	0.2100	0.0433	0.1233	0.1033
	SMAP (ours)	**0.2100**	**0.3933**	**0.2433**	**0.3967**	**0.3108**
CosFace	GenAP [16]	0.1833	0.1200	0.2133	0.1667	0.1708
	ROA [27]	0.1767	0.3633	0.2167	0.3300	0.2716
	Patch-Fool [35]	0.0567	0.1100	0.0733	0.1133	0.0883
	Patch-RL [37]	0.0400	0.1667	0.0633	0.1200	0.0975
	SMAP (ours)	**0.2100**	**0.3933**	**0.2433**	**0.3433**	**0.2974**
SFace+	GenAP [16]	0.1533	0.1900	0.1967	0.1700	0.1775
	ROA [27]	0.1833	0.4133	0.1800	0.4167	0.2983
	Patch-Fool [35]	0.0567	0.1900	0.0766	0.1100	0.1083
	Patch-RL [37]	0.0367	0.2033	0.0500	0.1367	0.1067
	SMAP (ours)	**0.2267**	**0.4467**	**0.2233**	**0.4433**	**0.3350**
SFaceR	GenAP [16]	0.1533	0.1167	0.1933	0.1333	0.1491
	ROA [27]	0.1833	0.3500	0.2167	0.3200	0.2675
	Patch-Fool [35]	0.0500	0.1300	0.0533	0.0933	0.0816
	Patch-RL [37]	0.0333	0.1533	0.0333	0.1233	0.0858
	SMAP (ours)	**0.1967**	**0.3700**	**0.2467**	**0.3567**	**0.2925**

TABLE IX

THE SUCCESS RATES OF DIFFERENT PATCH ATTACK METHODS ON THE COMMERCIAL FACE RECOGNITION SYSTEMS. THE ADVERSARIAL PATCHES ARE GENERATED ON ARCFACE. HIGHER SUCCESS RATES ARE IN BOLD

Method	TCloud	Face++	Average
CelebA-HQ			
Adv-Glasses [20]	0.0100	0.0433	0.0267
Adv-Hat [21]	0.0100	0.0300	0.0200
SMAP (ours)	**0.2100**	**0.3933**	**0.3016**
LFW			
Adv-Glasses [20]	0.0067	0.0300	0.0183
Adv-Hat [21]	0.0100	0.0100	0.0100
SMAP (ours)	**0.2433**	**0.3967**	**0.3200**

TABLE X

THE COMPARISON OF SUCCESS RATES WITH TNTS ON VGGFACE MODEL. CONF IS THE OUTPUT CONFIDENCE OF THE SOFTMAX FUNCTION. *Unlimited* MEANS THAT CONFIDENCE TO BE CONSIDERED A SUCCESS IS NOT LIMITED. HIGHER SUCCESS RATES ARE IN BOLD

	TnTs [12]	SMAP (ours)
Conf $>$ 0.9	0.9067	**1.0000**
Unlimited	0.9667	**1.0000**

of human faces and are located in critical areas of the face, such as the eyes and nose. This is the reason why our approach is more effective. Compared to Fig. 4, due to the difference in loss function, the perturbation magnitude on classification-based face recognition is greater while adversarial texture generated on similarity-based face recognition is more similar to the target identity.

E. Ablation Study

In this section, the ablation studies on patch initialization, location and size are performed to demonstrate the effects of these components.

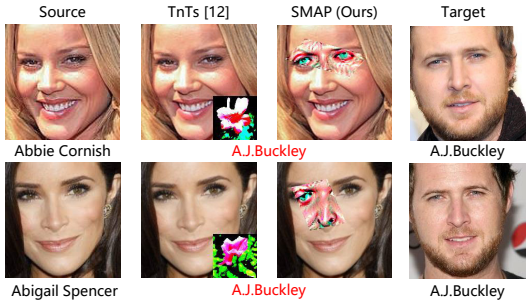

Fig. 6. Examples generated by adversarial patch methods. The second and third columns are the adversarial images generated by TnTs and SMAP, respectively. The black text below the image denotes the real identity, and the red text is the false identity after the attack.

1) Patch Initializetion: Three patch initialization methods are applied, including filling the patch with 0 (Zero), initializing the patch with the face image of the source identity (Source) and initializing the patch with the face image of the target identity (Target). The results are shown in Tab. XI. It can be seen that Target is beneficial for obtaining better results, especially on the face identification task. It indicates that the adversarial image generated by the Source scheme still retains the features of the source identity, resulting in a higher similarity between the adversarial image and the image of the source identity rather than that of the target identity. And the facial features of the adversarial image generated by the Zero scheme are destroyed, resulting in a low success rate.

2) Patch Location: The experiments to explore the importance of patch location are conducted. First, the gradient heatmap processed by calculating the absolute value and the patch location selected by PLSS is visualized. As shown in Fig. 7, it is observed that four face recognition models pay more attention to the nose and right eye, and different face recognition models generally focus on the same regions of the face. Therefore, perturbing this region will undoubtedly

TABLE XI

THE SUCCESS RATES OF PATCH ATTACK METHODS WITH DIFFERENT INITIALIZATION METHODS (ZERO, SOURCE AND TARGET) UNDER FACE VERIFICATION AND IDENTIFICATION TASK. THE ADVERSARIAL PATCHES ARE GENERATED ON ARCFACE AND COSFACE RESPECTIVELY. THE SURROGATE MODELS ARE LISTED IN THE ROWS, AND THE TEST MODELS ARE IN THE COLUMNS. * INDICATES WHITE-BOX ATTACKS. HIGHER SUCCESS RATES ARE IN BOLD

Model	Initialization	CelebA-HQ				LFW				Average (black-box)
		ArcFace	CosFace	SFace+	SFaceR	ArcFace	CosFace	SFace+	SFaceR	
Face Verification										
ArcFace	Zero	0.8767*	0.1600	0.1567	0.0333	0.9300*	0.1000	0.1367	0.0800	0.1111
	Source	**0.9067***	0.3567	0.3433	0.2300	0.9667*	0.2767	0.3367	0.1767	0.2867
	Target	0.9000*	**0.3700**	**0.4367**	**0.2600**	**0.9800***	**0.3133**	**0.5733**	**0.2800**	**0.3722**
CosFace	Zero	0.2733	0.7867*	0.1233	0.1567	0.2233	0.9333*	0.1733	0.1167	0.1777
	Source	**0.5300**	0.8233*	0.4000	**0.3133**	**0.5167**	0.9767*	0.4100	0.2833	0.4088
	Target	0.5033	**0.9133***	**0.4567**	0.2867	0.4467	**0.9867***	**0.5133**	**0.3200**	**0.4211**
Face Identification										
ArcFace	Zero	0.7300*	0.0633	0.0677	0.0633	0.7433*	0.0133	0.0400	0.0300	0.0461
	Source	0.8267*	0.1300	0.1533	0.1300	0.8267*	0.0667	0.0833	0.0567	0.1033
	Target	**0.8633***	**0.2633**	**0.3400**	**0.2300**	**0.9500***	**0.1500**	**0.3767**	**0.1667**	**0.2544**
CosFace	Zero	0.1067	0.8000*	0.0500	0.1000	0.0400	0.7767*	0.0533	0.0600	0.0683
	Source	0.2267	0.8667*	0.1633	0.1900	0.1200	0.8800*	0.1067	0.1000	0.1511
	Target	**0.3533**	**0.8867***	**0.3500**	**0.2533**	**0.2300**	**0.9633***	**0.3067**	**0.2133**	**0.2844**

TABLE XII

THE SUCCESS RATES OF PATCH ATTACK METHODS WITH DIFFERENT LOCATION SCHEMES (RANDOM, SEARCH, TRANSLATION AND SMAP) UNDER FACE VERIFICATION AND IDENTIFICATION TASK. THE ADVERSARIAL PATCHES ARE GENERATED ON ARCFACE AND COSFACE RESPECTIVELY. THE SURROGATE MODELS ARE LISTED IN THE ROWS, AND THE TEST MODELS ARE IN THE COLUMNS. * INDICATES WHITE-BOX ATTACKS. HIGHER SUCCESS RATES ARE IN BOLD

Model	Location	CelebA-HQ				LFW				Average (black-box)
		ArcFace	CosFace	SFace+	SFaceR	ArcFace	CosFace	SFace+	SFaceR	
Face Verification										
ArcFace	Random	0.6900*	0.1233	0.2100	0.1033	0.8200*	0.0867	0.2233	0.0867	0.1388
	Search	0.9067*	0.3800	0.4700	0.2867	0.9867*	0.3067	0.4967	0.2533	0.3655
	Translation	0.9067*	0.4000	0.5100	0.3200	**0.9900***	0.3233	0.5133	**0.2800**	0.3911
	SMAP	**0.9133***	**0.4500**	**0.5400**	**0.3567**	0.9867*	**0.3533**	**0.5500**	0.2700	**0.4200**
CosFace	Random	0.1933	0.6700*	0.2100	0.1433	0.1867	0.8067*	0.2500	0.1167	0.1833
	Search	0.4933	0.9033*	0.4500	0.3467	0.4300	**0.9833***	0.4633	0.2800	0.4105
	Translation	0.4900	**0.9133***	0.5033	0.3400	0.4533	0.9800*	0.5300	**0.3267**	0.4405
	SMAP	**0.5467**	0.9100*	**0.5700**	**0.4167**	**0.4567**	0.9733*	**0.5367**	0.3167	**0.4739**
Face Identification										
ArcFace	Random	0.5300*	0.0400	0.1133	0.0467	0.5700*	0.0167	0.1033	0.0333	0.0588
	Search	**0.8867***	0.2533	0.3600	0.2400	0.9600*	0.1533	0.3400	0.1500	0.2494
	Translation	0.8800*	0.2900	0.3833	0.2667	**0.9667***	0.1533	0.3567	0.1700	0.2700
	SMAP	0.8767*	**0.3000**	**0.4367**	**0.2767**	0.9433*	**0.1633**	**0.3800**	**0.2033**	**0.2933**
CosFace	Random	0.0833	0.5267*	0.0933	0.0767	0.0633	0.6233*	0.1067	0.0633	0.0811
	Search	0.3433	0.8967*	0.3400	0.2867	0.2233	**0.9467***	0.3067	0.2033	0.2838
	Translation	0.3500	**0.9067***	0.3767	0.2767	0.2067	**0.9467***	0.3333	**0.2300**	0.2955
	SMAP	**0.3733**	0.8933*	**0.4400**	**0.3400**	**0.2433**	0.9400*	**0.3467**	0.2233	**0.3277**

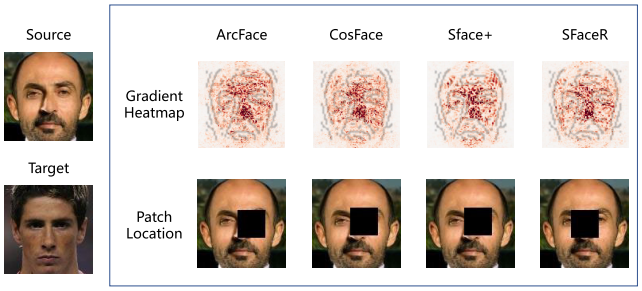

Fig. 7. Examples of PLSS for four face recognition models (ArcFace, CosFace, SFace+, SFaceR) under the impersonation attack task. The left column is the original image. The first row in the box is the gradient heatmap, and the second row is the patch location selected by PLSS.

increase the success rate and transferability, which is why the PLSS is designed.

Tab. XII shows the results of four location schemes considered to optimize the patch, including random location (Random), search-based key location at the beginning of the attack (Search), dynamic optimal location (Translation) and SMAP. The patch location in Search is found by PLSS. The difference between Translation and SMAP is that in Translation the patch shape is fixed, while in SMAP the deformation parameters are updated during the iteration. It can be seen that attacks with a random patch location are almost unsuccessful for black-box models. The attack success rate increases significantly when a crucial patch location is selected at the beginning. Moreover, the performance of Translation and SMAP is improved after combining the patch update scheme, which shows the effectiveness of dynamically changing the patch position and shape during iterations. For example, Random achieves the average black-box success rate of 18.33% on face verification task and 8.11% on face identification task using ArcFace surrogate model. By using Translation and SMAP scheme, the average success rates is increased by 25.72% and 29.06% on face verification task, and by 21.44% and 24.66% on face identification task. It can be noticed that compared to Search and Translation, SMAP

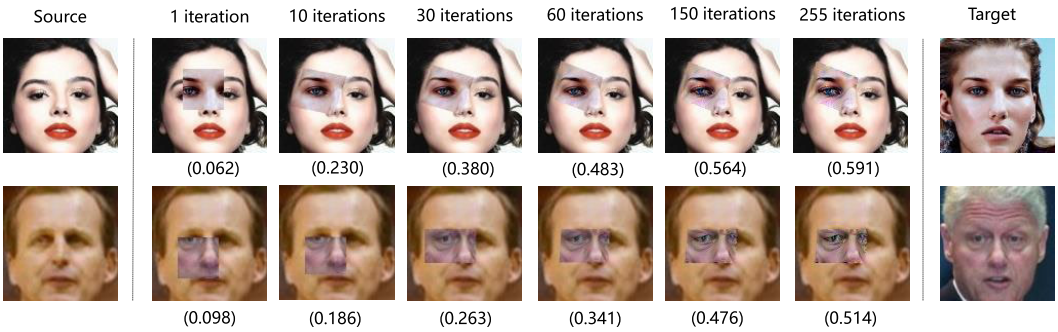

Fig. 8. The optimization process of examples at different stages which generated by the proposed method against ArcFace. The first row is from the CelebA-HQ and the second is from the LFW. The text above images denotes the number of iterations, and the text below is the similarity score.

TABLE XIII

THE SUCCESS RATES OF PATCH ATTACK METHODS WITH DIFFERENT PATCH SIZE UNDER FACE VERIFICATION AND IDENTIFICATION TASK. THE ADVERSARIAL PATCHES ARE GENERATED ON ARCFACE AND COSFACE RESPECTIVELY. THE SURROGATE MODELS ARE LISTED IN THE ROWS, AND THE TEST MODELS ARE IN THE COLUMNS. * INDICATES WHITE-BOX ATTACKS. HIGHER SUCCESS RATES ARE IN BOLD

Model	Size	CelebA-HQ				LFW				Average
		ArcFace	CosFace	SFace+	SFaceR	ArcFace	CosFace	SFace+	SFaceR	(black-box)
Face Verification										
ArcFace	Small (9.7%)	0.8233*	0.1667	0.2833	0.1333	0.8400*	0.0800	0.2467	0.0800	0.1650
	Large (14.3%)	0.9000*	0.3700	0.4367	0.2600	0.9800*	0.3133	0.5733	0.2800	0.3722
CosFace	Small (9.7%)	0.2600	0.7833*	0.2667	0.1633	0.1933	0.8233*	0.2700	0.1300	0.2138
	Large (14.3%)	0.5033	0.9133*	0.4567	0.2867	0.4467	0.9867*	0.5133	0.3200	0.4211
Face Identification										
ArcFace	Small (9.7%)	0.6267*	0.0633	0.1367	0.0567	0.5200*	0.0133	0.0800	0.0133	0.0605
	Large (14.3%)	0.8633*	0.2633	0.3400	0.2300	0.9500*	0.1500	0.3767	0.1667	0.2544
CosFace	Small (9.7%)	0.0933	0.6300*	0.1500	0.0967	0.0467	0.5533*	0.0833	0.0300	0.0833
	Large (14.3%)	0.3533	0.8867*	0.3500	0.2533	0.2300	0.9633*	0.3067	0.2133	0.2844

performs poorly in the white-box setting, especially against CosFace. The reason can be attributed to the fact that Search and Translation focus on one or two influencing factors and are prone to overfitting. In contrast, the joint optimization of texture, position and shape has a more balanced optimization objective, which mitigates this phenomenon. SMAP gives up the extreme white-box success rate, but gets better transferability.

3) Patch Size: The impact of patch area size on the success rate is investigated. Two area patches were tested, consisting of 9.7% image pixels (Small) and 14.3% image pixels (Large), respectively. The results are illustrated in Tab. XIII. It can be seen that when the area is increased by 4.6%, the average black-box success rate of face verification task almost doubles. The success rate of face identification task grows more dramatically, especially on LFW. It demonstrates the importance of patch size for success and the vulnerability of the face recognition models for large area patches.

F. Visualization

Fig. 8 illustrates the joint optimization process of texture, position and shape of the adversarial patches. It can be seen that after the optimal position is selected in the first iteration according to the PLSS, the position and shape of the patch continue to be optimized as the iterations proceed. Since PLSS searches with the fixed size and step, some positions will be missed. The subsequent dynamic optimization will fix this drawback. The process is gradual, with no sudden and dramatic changes in position and shape. The schedule of the update step will guarantee the stability of the optimization.

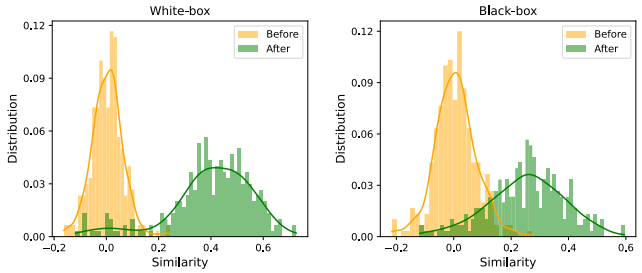

Fig. 9. (a) The similarity distribution of image pairs before and after SMAP attack in the white-box setting. (b) The similarity distribution of image pairs before and after SMAP attack in the black-box setting.

Face recognition systems use similarity to determine whether a pair of faces matches. Therefore, the proposed method aims to improve the similarity between the adversarial face and the target face by injecting a patch. To highlight the impact of the patch, the change in similarity before and after the SMAP attack is visualized. The histograms of the similarity distribution of 300 pairs of face images before and after SMAP attack are plotted in Fig. 9. Specifically, the source face and the adversarial face generated on ArcFace are respectively fed into ArcFace to calculate the similarity with the face of the target identity in the white-box setting, and the distribution density of these sampled pairs before and after SMAP attack is shown in Fig. 9a. Then they are fed into CosFace to calculate the similarity in the black-box setting, and the distribution density before and after SMAP attack is shown in Fig. 9b. It can be seen that before the attack, the similarity is concentrated outside the acceptance threshold

due to identity mismatch. After the attack, the adversarial faces confuse the model's decisions so that the distribution is shifted. In addition, the variance of similarity in black-box is greater than it in white-box which illustrates the difficulty of black-box attacks.

VI. Conclusion

The development of adversarial attack raises security concerns for DNN-based face recognition. To evaluate the vulnerability of face recognition models, we delve into threatening impersonation attacks. In this paper, a Spatial Mutable Adversarial Patch method is proposed that can dynamically change the position and shape of the patch during the iteration. Firstly, the Patch Location Selection Scheme is introduced to find a critical patch location. Then the differentiable affine transformation and the sampling mechanism are applied to update patch perturbation and the affine matrix based on the gradients simultaneously. By doing this, the weakness of the fixed shape and position of the patch is overcome. The experimental results on the SOTA face recognition models and commercial models validate that the proposed method could effectively improve transferability. Adversarial patch attack can be broadly divided into research on position and shape and research on natural-like texture. The former focuses more on the impact of location and shape on the patch, while the latter pays more attention to concealed and seemingly harmless adversarial patches. In our future work, we will continue to delve into the former, aiming at 3D, more aggressive position and shape, and smaller attack area, to make adversarial patch easier to implement in the physical world. It will further promote the development of secure face recognition systems in real-world scenarios.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[2] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 3104–3112.

[4] A. Vaswani et al., "Attention is all you need," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[5] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4685–4694.

[6] H. Wang et al., "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 5265–5274.

[7] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. 2nd Int. Conf. Learn. Represent.*, Banff, AB, Canada, Apr. 2014, pp. 1–10.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–11.

[9] H. Zhang, Y. Avrithis, T. Furon, and L. Amsaleg, "Walking on the edge: Fast, low-distortion adversarial examples," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 701–713, 2021.

[10] B. Bonnet, T. Furon, and P. Bas, "Generating adversarial images in quantized domains," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 373–385, 2022.

[11] Y. Zhong and W. Deng, "Towards transferable adversarial attack against deep face recognition," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1452–1466, 2021.

[12] B. G. Doan, M. Xue, S. Ma, E. Abbasnejad, and D. C. Ranasinghe, "TnT attacks! Universal naturalistic adversarial patches against deep neural network systems," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 3816–3830, 2022.

[13] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Dallas, TX, USA, Nov. 2017, pp. 15–26.

[14] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, Apr. 2018, pp. 1–12.

[15] X.-C. Li, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Decision-based adversarial attack with frequency mixup," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1038–1052, 2022.

[16] Z. Xiao et al., "Improving transferability of adversarial patches on face recognition with generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11840–11849.

[17] T. B. Brown, D. Mane, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," in *Proc. 31st Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–6.

[18] D. Karmon, D. Zoran, and Y. Goldberg, "LaVAN: Localized and visible adversarial noise," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, Stockholm, Sweden, 2018, pp. 2512–2520.

[19] Z. Hu, S. Huang, X. Zhu, F. Sun, B. Zhang, and X. Hu, "Adversarial texture for fooling person detectors in the physical world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New Orleans, LA, USA, Jun. 2022, pp. 13297–13306.

[20] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 1528–1540.

[21] S. Komkov and A. Petiushko, "AdvHat: Real-world adversarial attack on ArcFace face ID system," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 819–826.

[22] X. Yang, F. Wei, H. Zhang, and J. Zhu, "Design and interpretation of universal adversarial patches in face detection," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)* (Lecture Notes in Computer Science), vol. 12362. Glasgow, U.K.: Springer, Aug. 2020, pp. 174–191.

[23] A. Chindaudom, P. Siritanawan, K. Sumongkayothin, and K. Kotani, "AdversarialQR: An adversarial patch in QR code format," in *Proc. Joint 9th Int. Conf. Informat., Electron. Vis. (ICIEV) 4th Int. Conf. Imag., Vis. Pattern Recognit. (icIVPR)*, Aug. 2020, pp. 1–6.

[24] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2921–2929.

[25] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 618–626.

[26] A. Liu et al., "Perceptual-sensitive GAN for generating adversarial patches," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Jan. 2019, pp. 1028–1035.

[27] T. Wu, L. Tong, and Y. Vorobeychik, "Defending against physically realizable attacks on image classification," in *Proc. 8th Int. Conf. Learn. Represent.*, Addis Ababa, Ethiopia, Apr. 2020, pp. 1–26.

[28] S. Rao, D. Stutz, and B. Schiele, "Adversarial training against location-optimized adversarial patches," in *Computer Vision—ECCV Workshops* (Lecture Notes in Computer Science), vol. 12539. Glasgow, U.K.: Springer, Aug. 2020, 2020, pp. 429–448.

[29] X. Li and S. Ji, "Generative dynamic patch attack," in *Proc. 32nd Brit. Mach. Vis. Conf.*, 2021, p. 156.

[30] Z. Chen, B. Li, S. Wu, J. Xu, S. Ding, and W. Zhang, "Shape matters: Deformable patch attack," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)* (Lecture Notes in Computer Science), vol. 13664. Tel Aviv, Israel: Springer, Oct. 2022, pp. 529–548.

[31] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, 2018, pp. 1–28.

[32] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 39–57.

[33] Z. Wu, S. Lim, L. S. Davis, and T. Goldstein, "Making an invisibility cloak: Real world adversarial attacks on object detectors," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)* (Lecture Notes in Computer Science), vol. 12349. Glasgow, U.K.: Springer, Aug. 2020, pp. 1–17.
[34] D. Song et al., "Physical adversarial examples for object detectors," in *Proc. 12th USENIX Workshop Offensive Technol.*, Baltimore, MD, USA, 2018, pp. 1–10.
[35] Y. Fu, S. Zhang, S. Wu, C. Wan, and Y. Lin, "Patch-fool: Are vision transformers always robust against adversarial perturbations?" in *Proc. 10th Int. Conf. Learn. Represent.*, 2022, pp. 1–18.
[36] Y. Dong et al., "Efficient decision-based black-box adversarial attacks on face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 7706–7714.
[37] X. Wei, Y. Guo, J. Yu, and B. Zhang, "Simultaneously optimizing perturbations and positions for black-box adversarial patch attacks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 9041–9054, Jul. 2023.
[38] B. Yin et al., "Adv-Makeup: A new imperceptible and transferable attack on face recognition," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1252–1258.
[39] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, 2017, pp. 1–14.
[40] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 2017–2025.
[41] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep, 07-49, Oct. 2007. [Online]. Available: https://vis-www.cs.umass.edu/lfw/
[42] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, 2018, pp. 1–26.
[43] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vis. Conf.*, X. Xie, M. W. Jones, and G. K. L. Tam, Eds. Swansea, U.K., 2015, p. 41.
[44] W. Liu et al., "Learning towards minimum hyperspherical energy," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2018, pp. 6225–6236.
[45] W. Liu, Y. Wen, B. Raj, R. Singh, and A. Weller, "SphereFace revived: Unifying hyperspherical face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2458–2474, Feb. 2023.
[46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, 2015, pp. 1–14.
[47] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 815–823.
[48] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–12.
[49] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.
[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

Haotian Ma received the B.S. degree from Xidian University, Xi'an, China, in 2022. He is currently pursuing the M.S. degree with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include adversarial example and artificial intelligence security.

Ke Xu (Member, IEEE) received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2019. He is currently an Associate Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include action recognition, gait recognition, and abnormal events detection.

Xinghao Jiang (Senior Member, IEEE) received the Ph.D. degree in electronic science and technology from Zhejiang University, Hangzhou, China, in 2003. He was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, from 2011 to 2012. He is currently a Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include multimedia security and image retrieval, intelligent information processing, cyber information security, information hiding, and watermarking.

Zeyu Zhao received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2020, where he is currently pursuing the Ph.D. degree with the School of Electronic Information and Electrical Engineering. His research interests include adversarial example and artificial intelligence security.

Tanfeng Sun (Senior Member, IEEE) received the Ph.D. degree in information and communication system from Jilin University, Changchun, China, in 2003. He was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, from 2012 to 2013. He is currently a Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. He is also a Researcher with the National Engineering Laboratory for Information Content Analysis Technology, Shanghai. His research interests include videos content recognition and understanding with AI, digital forensics on video, and image forgery.