International Journal of Computational Intelligence Systems (2023) 16:196
https://doi.org/10.1007/s44196-023-00376-6

RESEARCH ARTICLE

Semantic Adversarial Attacks on Face Recognition Through Significant Attributes

Yasmeen M. Khedr[1,2] · Yifeng Xiong[1] · Kun He[1]

Received: 5 September 2023 / Accepted: 24 November 2023

Abstract
Face recognition systems are susceptible to adversarial attacks, where adversarial facial images are generated without awareness of the intrinsic attributes of the images in existing works. They change only a single attribute indiscriminately. To this end, we propose a new Semantic Adversarial Attack using StarGAN (SAA-StarGAN), which manipulates the facial attributes that are significant for each image. Specifically, we apply the cosine similarity or probability score to predict the most significant attributes. In the probability score method, we train the face verification model to perform an attribute prediction task to get a class probability score for each attribute. Then, we calculate the degree of change in the probability value in an image before and after altering the attribute. Therefore, we perform the prediction process and then alter either one or more of the most significant facial attributes under white-box or black-box settings. Experimental results illustrate that SAA-StarGAN outperforms transformation-based, gradient-based, stealthy-based, and patch-based attacks under impersonation and dodging attacks. Besides, our method achieves high attack success rates on various models in the black-box setting. In the end, the experiments confirm that the prediction of the most important attributes significantly impacts the success of adversarial attacks in both white-box and black-box settings and could improve the transferability of the generated adversarial examples.

Keywords Adversarial examples · Image-to-image translation · Face verification · Feature fusion · Black-box attack · Attack transferability

1 Introduction

Face Recognition (FR) [1] is a vital computer vision task widely used in solving authentication problems. FR is categorized into Face Identification (FI) and Face Verification (FV). FV determines whether a pair of face images belong to the same identity [2], while FI compares the input image with a gallery set of face images and then classifies the image as an identity. Over the past decades, FV has achieved significant achievements in various applications, such as mobile payment, military, finance, surveillance security, and border control. However, Szegedy et al. [3] have found that Deep Neural Networks (DNNs) are susceptible to adversarial examples. Adversarial examples are tiny perturbations that remain imperceptible to human vision, which are added to benign images to mislead DNN models. Then, many other studies confirm the vulnerability of DNNs to adversarial examples [4–6]. Adversarial examples are categorized into gradient-based [4, 7–9], input transformation-based [10–14], and unrestricted perturbations [15, 16]. Attacks vary by goals and assumptions, with white-box and black-box settings being the main distinctions based on the attacker's knowledge on target model. White-box assumes full knowledge of the model, while the black-box only has access to inputs and outputs [17]. This paper focuses on generating unrestricted perturbations under the white-box and black-box attacks.

Adversarial studies for face recognition models are growing interest due to the importance of FR in the real world [17–21]. The crafting of adversarial face images depends on manipulating the facial content such as face synthesis, face

✉ Kun He
brooklet60@hust.edu.cn

Yasmeen M. Khedr
yasmeenkhedr@hust.edu.cn

Yifeng Xiong
xiongyf@hust.edu.cn

1 School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China

2 Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt

Published online: 19 December 2023

attribute manipulation, and expression swap [18]. Some studies use face attribute manipulation for different goals. Rozsa et al. [22, 23] mislead facial attribute recognition. Also, Mirjalili and Ross [24] modify the face image for a gender classifier. Recently, methods based on Generative Adversarial Networks (GANs) have appeared that are used to manipulate facial attribute images, such as StarGAN [25], STGAN [26], and AttGAN [27]. Joshi et al. [28] use AttGAN to generate semantic attacks to deceive gender classifiers. These studies are limited to classification problems instead of facial recognition. Meanwhile, Qiu et al. [18] craft adversarial examples of misleading FR by changing the attribute individually and checking whether the generated image is adversarial until they find an adversarial example or fail after attempting the change. However, they craft an adversarial face image by indiscriminately distorting facial attributes without being aware of the significant facial attributes in each image. Hu et al. [21] propose Adversarial Makeup Transfer GAN (AMT-GAN) to generate adversarial face images. Still, it tends to produce high-quality images of females due to an imbalance of gender in the training dataset. Xiao et al. [29] propose a generative adversarial patch (GenAP) attack method to craft transferable adversarial patches. However, GenAP generates unnatural adversarial face images that arouse suspicion. These studies handle the attack on face recognition. Still, they either suffer from the weakness of transferability to black-box models due to changing the face attribute randomly, exhibit bias on genders due to the imbalance of data, or make the generated face fill with awkward noises.

Our work aims to mislead face recognition models depending on changing the significant facial attributes. Thereby, we propose a new attack method called the Semantic Adversarial Attack using StarGAN (SAA-StarGAN), which effectively and easily crafts semantic adversarial examples besides improving the attack transferability significantly by tampering with significant facial attributes for each input image. These attributes are supposed to affect the decisions of different FV models, leading to deceiving the FV models and enhancing the adversarial transferability. In the white-box setting, we predict the most significant attributes for each input image using either the cosine similarity (CS) or the probability score (PS) based on the Target Face Verification (TFV) model. Then, we change one or multiple via the StarGAN model in the feature space. The Attention Feature Fusion (AFF) method is used to fuse the features of inconsistent semantics to generate a realistic image and produce β different values used for interpolation. In the black-box setting, SAA-StarGAN depends on predicting the most important attributes through the cosine similarity (CS) method. Based on the prediction step, these attributes are modified on the input image according to their arrangement by making an iterative loop to alter them sequentially until reaching the adversarial face images.

The empirical results confirm that predicting the most significant attributes (that will be changed first) plays an essential role in successful attacks. Our SAA-StarGAN method outperforms other methods significantly on the attack success rate in the black-box setting. Also, it preserves high attack success rates in the white-box setting for both impersonation and dodging attacks. Our method provides perceptually realistic images that maintain the source image identity to avoid confusing human perception. We also analyze the attention map of the TFV model that is attacked by our adversarial face images using gradient-weighted class activation. As a result, our method focuses on trivial rather than prominent features.

Our main contributions are summarized as follows:

- We propose a novel method called Semantic Adversarial Attack using StarGAN (SAA-StarGAN) that enhances the transferability of adversarial face images by tampering with critical facial attributes for the input image.
- SAA-StarGAN generates semantic adversarial face images easily and effectively by predicting the most significant facial attributes using two techniques, cosine similarity or probability score, for impersonation and dodging attacks.
- We propose modifications on SAA-StarGAN to depend only on the output of the target model in a black-box setting by applying a linear search to find the optimal value of the interpolation coefficient that affects the generated face images.
- The empirical results confirm that predicting the most significant attributes (which will be changed first) plays a vital role in a successful attack. Our SAA-StarGAN method outperforms the baselines considerably on the white-box attack success rate and black-box adversarial transferability. Also, it provides perceptually realistic images that maintain the source image identity to avoid confusing human perception.

2 Related Work

We introduce the related work of generating adversarial examples in image classification. After that, we focus on recent studies crafting adversarial examples on FR models.

2.1 Adversarial Attacks on Images

Many adversarial example generation methods have been proposed to mislead different image classification models.

For white-box attacks, some studies focus on generating restricted adversarial examples by adding perturbations to the pixels of the input images. The L-BFGS method is an early method designed to mislead image classification tasks [3]. Iterative adversarial attack methods, such as the Basic Iterative Method (BIM) [7] and Projected Gradient Descent (PGD) [30], refine the adversarial perturbations of smaller magnitude iteratively. This process improves attack success rates compared to a Fast Gradient Sign Method (FGSM) [4] that uses the gradients of the neural network to generate adversarial examples with a one-step gradient update. Thereafter, momentum [8, 31] is incorporated into the iterative FGSM to boost the attack transferability.

For black-box attacks, one interesting direction is to improve the transfer-based attacks. The Diverse Input Method (DIM) [10] enhances the diversity of input images by applying random transformations to the input images with probability p. The Translation-Invariant Method (TIM) [11] achieves a translation-invariant attack that convolves the gradients on the untranslated images with a pre-defined kernel matrix. Many follow-up works [9, 32–34] seek to improve the transferability of adversarial examples. Another family of black-box attack methods is based on score-based or decision-based attacks. Some work only obtains the hard-label predictions [35, 36], and others get the predicted probability by the target model [36]. Liu et al. [37] propose an attack model based on the confidence probabilities to select the important words in the text for misleading a text classification model.

2.2 Adversarial Attacks on Face Recognition

Recently, many adversarial attacks have been proposed for attacking face recognition models. One line of study focuses on changing the facial appearance of input images by adding small perturbations in a specific region to be imperceptible to human eyes. Deb et al. [20] propose an automated adversarial face method called AdvFaces that generates minimal perturbations in the salient facial regions via GANs. Zhu et al. [38] hide the attack information by the makeup effect to attack the eye regions only. Other works exploit GANs to generate high-quality images. Hu et al. [21] synthesize adversarial face images with makeup from reference images to improve transferable attacks. On the other hand, some works are based on manipulating facial attributes. Rozsa et al. [23] propose the Fast Flipping Attribute (FFA) technique, which found that the robustness of DNNs against adversarial attacks varies highly between facial attributes. Kakizaki and Yoshida [19] use image translation techniques to generate unrestricted adversarial examples by translating the source image into any desired facial appearance with large perturbations. Qiu et al. [18] introduce SemanticAdv, which can generate unrestricted adversarial examples by altering a single facial attribute.

For patch-based methods are easy to perceive by humans and have weak transferability. These attacks can be generated by a variety of different tools, such as adding some adversarial face accessories [39], printed adversarial hat [40], 3D printed masks [41], natural makeup [42], and adversarial patches [43].

3 Methodology

In this section, we first review the problem formulation according to adversarial examples on FR. Then, we present our methods to craft diverse semantic adversarial face images using the proposed SAA-StarGAN method in white-box and black-box settings. Finally, we present a preliminary method called Random Attributes Selection in the simplest scenarios in the white-box setting as a baseline method for comparison. From this method, we found the motivation to study the impact of diverse attributes and predict the most significance.

3.1 Problem Formulation

Face Verification system (FV) compares the input with another face image to decide whether they belong to the same identity. Given a face verification system FV and an input face image x, the main goal of our work is generating an adversarial face image x_{adv} similar to the original image x but misleads the FV model, i.e., $FV(x_{adv}) = y' \neq y$, where y is the corresponding ground truth label.

On the other hand, adversarial attacks are divided into two types: dodging and impersonation attacks [17]. The dodging attack (untargeted attack) is developed to fool the target model, such that the output is a random identity excluding the original one. In contrast, the impersonation attack (targeted attack) misleads the target model by recognizing the adversarial face image as a specified target identity. In general, the impersonation attack is more difficult than the dodging attack. This work focuses on achieving both dodging and impersonation attacks. For the dodging attack, we generate an adversarial face image x_{adv} from x, and x_{adv} is identified as not having the same identity as $FV(x_{adv}) \neq y$. For an impersonation attack, we seek to make the model recognize x_{adv} as the same identity of another given image, such that $FV(x_{adv}) = y_{tgt}$.

3.2 Semantic Adversarial Attack

To craft semantic adversarial facial images, we propose an efficient attack method based on changing the most significant attribute by the face editing method, StarGAN. A StarGAN model [25] consists of a single generator G and

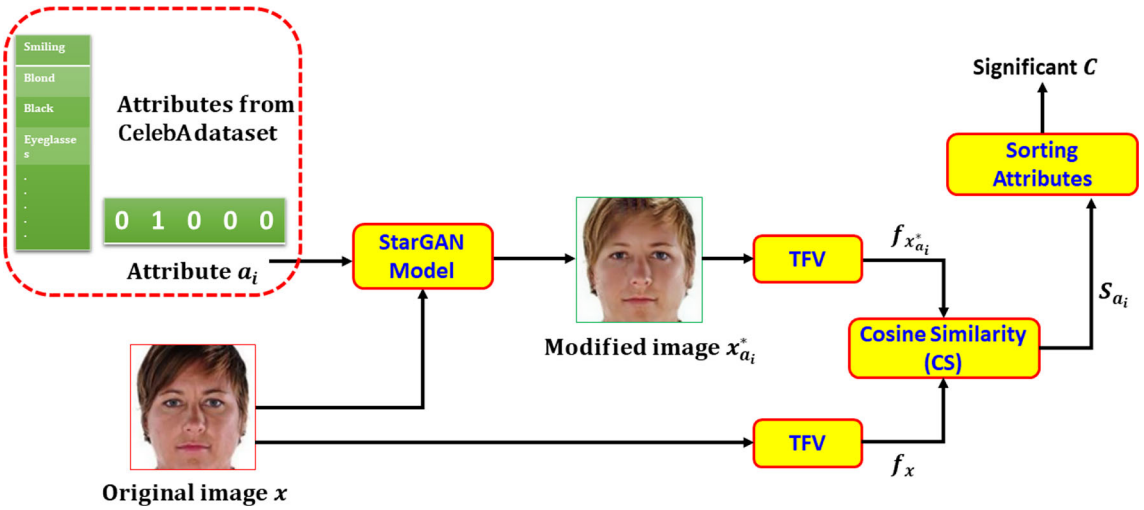


Fig. 1 Prediction of the most significant attributes by a CS technique

a discriminator D that work together to achieve translation across different domains. For instance, it converts images between different human faces, such as changing the hair color of an image x, called input attribute, from black to blond. By taking a face image x as the input and one particular attribute value $a = (a_1, a_2, ..., a_K)$, where K indicates the number of attributes and a_i denotes the appearance of the i^{th} attribute that represents $\{0, 1\}$; the StarGAN model synthesizes a new face image with the modified attribute.

Our work first predicts each input image's most significant facial attributes. Then, we generate an adversarial face image in the intermediate layers instead of the output layers by manipulating the most significant facial attribute. The motivation for this step is to boost internal feature distortion, leading to improved performance. Consequently, our method could easily craft semantic adversarial facial images in white-box and black-box settings to achieve impersonation and dodge attacks. Besides, it enhances adversarial transferability for black-box attacks.

The proposed framework mainly includes two steps for *a white-box attack*: (1) the most significant attributes prediction for each input image; (2) an adversarial face image generation by modifying one or multiple most significant attributes by StarGAN.

3.2.1 Significant Attribute Prediction

The awareness of the image's intrinsic attributes plays a critical role in dominating the decision of different FV models. Therefore, we propose two methods, the cosine similarity (CS) or the probability score (PS), to detect the significant attributes. These methods are indicated as SAA-StarGAN-CS and SAA-StarGAN-PS, respectively. For our attack method, we apply CS or PS to predict the most significant attributes based on the StarGAN model and then compare their results. Consequently, we re-train the G of StarGAN to use in the significant attribute prediction step.

Algorithm 1 Determine the most significant attributes by cosine similarity

Require: Original image x, attributes $\{a\}$, generator G of StarGAN, target face verification model TFV
Ensure: The most significant attributes C
1: **for** each a_i in attributes $\{a\}$ **do**
2: change a_i using G
3: $x^*_{a_i} \leftarrow G(x, a_i)$
4: $f_{x^*} \leftarrow TFV(x)$, $f_{x^*_{a_i}} \leftarrow TFV(x^*_{a_i})$
5: $S_{a_i} \leftarrow CS(f_x, f_{x^*_{a_i}})$
6: **end for**
7: $C \leftarrow$ Sort $\{a\}$ according to each S_{a_i} in the ascending order
8: Output the most significant attributes $C = (c_1, c_2, c_3, ..., c_K)$

Cosine Similarity (CS): Cosine similarity is a technique to measure the similarity between two face images to decide whether belonging to the same person or not. Unlike our work, we leverage from this point to get the degree of similarity between the input image before and after changing attributes. We observe that changing the attribute on the input image affects the degree of similarity, leading to changing the decision of different face recognition models. Therefore, we utilize the TFV models [44, 45] to obtain the important attributes by computing the cosine similarity between the output features of the TFV model. Algorithm 1 illustrates the predicted most significant attributes using the CS technique. According to a StarGAN model [25], we give the original image with the modified attribute to synthesize a modified face image. Therefore, we use a StarGAN model to generate various synthesized face images according to the original image. To discover the impact of diverse

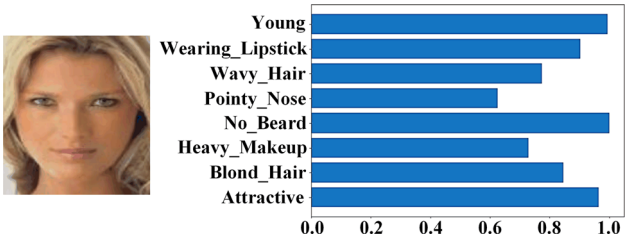


Fig. 2 Output from the attribute prediction model using FaceNet

attributes, we change each attribute a_i of the original face image x to get the synthesis image $x^*_{a_i}$. Here, image attributes $a = (a_1, a_2, a_3, ..., a_K)$, a_i indicate the ith attribute, and K is the total number of attributes. Then, we extract the synthesis image features $f_{x^*_{a_i}}$ and the original image features f_x by the TFV model. After that, we calculate their cosine similarity to get S_{a_i}, which refers to the degree of change in the output of TFV by changing attribute a_i. The accuracy of the TFV model in predicting the attributes is due to the difference between the original and synthesized face images. Finally, to detect the most important attributes $C = (c_1, c_2, c_3, ..., c_K)$, we sort the attributes in a in ascending order according to the similarity score values, where c_i indicates the i^{th} sorted attribute. Here, we consider that the less the cosine similarity, the increase the significance of the attribute on impacting. Figure 1 illustrates the steps to predict the most significant attributes by a CS technique. Also, the cosine similarity, as illustrated in Eq. (1), is a way to measure the similarity between two vectors, ranging from 0 to 1

$$S_{a_i} = CS(f_x, f_{x^*_{a_i}}) = \frac{f_x \cdot f_{x^*_{a_i}}}{\| f_x \| \cdot \| f_{x^*_{a_i}} \|}. \tag{1}$$

Probability Score (PS): The basic idea of the probability score method is to use a TFV model as the *Attribute Prediction model (Att-Pred)* to predict the important attributes of the input. Therefore, we train a TFV model for the attribute prediction task to have the class probability score (PS) for each attribute a, as shown in Fig. 2. For our work, an *attribute prediction task* is defined as a classification problem despite being a regression problem.

For training an Att-Pred model, we first freeze the weights of the early layers of a TFV model. After that, we train the top layers for the attribute prediction model task. Specifically, we choose BCEWithLogitsLoss to discriminate the single element of the attribute vector, which combines a sigmoid layer and the binary cross entropy loss in one class. This loss creates a criterion to measure an error between the truth vector $a = [a_1, a_2, a_3,...]$ and the output attribute vector $a' = [a'_1, a'_2, a'_3,...]$ where each element in the two vectors indicates the attribute class. Finally, we evaluate FaceNet as an Att-Pred model, where the effectiveness of our method

Algorithm 2 Determine the most significant attributes by probability score

Require: Original image x, attributes $\{a\}$, generator G of StarGAN, attribute prediction model f
Ensure: The most significant attributes $\{C\}$
1: Use attribute prediction model to predict attributes
2: **for** each a'_i in attributes $\{a\}$ **do**
3: $P_{a'_i}(x) \leftarrow f(x)$
4: Change a'_i using G
5: $x^*_{a'_i} \leftarrow G(x, a'_i)$
6: $P_{a'_i}(x^*_{a'_i}) \leftarrow f(x^*_{a'_i})$ ▷ Compute the probability for each attribute
7: Compute $\Delta P_{a'_i}$ by Eq. (2)
8: **end for**
9: $C \leftarrow$ Sort $\{a\}$ using $\Delta P_{a'_i}$ in descending order
10: Output the most significant attributes $C = (c_1, c_2, c_3, ..., c_K)$

depends on a good prediction model that can accurately predict facial attributes. Therefore, we present the accuracy of an Att-Pred model in Table 1. In our attack method, we use a StarGAN model to change the modified attribute a'_i to get the synthesized image $x^*_{a'_i}$. After that, we use an Att-Pred model for each modified attribute a'_i to obtain the class probability score (PS). Then, we need to calculate the probability value $P_{a'_i}$ of a'_i for x and x^* to get the degree of change since each attribute in the input face image has a different impact on the final decision. Here, $\Delta P_{a'_i}$ indicates the degree of change in the probability value in an image x before and after changing a'_i

$$\Delta P_{a'_i} = P_{a'_i}(x) - P_{a'_i}(x^*_{a'_i}), \tag{2}$$

where

$$x_{a_i} = x_{a_1} x_{a_2} ... x_{a_i} ... x_{a_k},$$
$$x^*_{a'_i} = x^*_{a_1} x^*_{a_2} ... x^*_{a'_i} ... x^*_{a_k}.$$

Here, $x^*_{a'_i}$ indicates the generated image after changing modified attribute a'_i, and P_{a_i} represents the probability score for modified a'_i for an image. Finally, we sort the attributes in descending order using $\Delta P_{a'_i}$ to obtain the most significant attributes C to help us craft the adversarial example x_{adv}. These attributes represent the best attack effect. The steps for the PS technique are illustrated in Algorithm 2.

3.2.2 Adversarial Face Image Generation

The second step focuses on generating the adversarial face images x_{adv}, as illustrated in Fig. 3. To craft a x_{adv}, we apply two kinds of perturbations using single or multiple significant attributes. For the CS technique, SAA-StarGAN-CS and SAA-StarGAN-CS-M indicate the methods used for single and multiple attributes, respectively. While SAA-StarGAN-PS and SAA-StarGAN-PS-M refer to the corresponding

Table 1 Accuracy (Acc.) of each facial attribute using a FaceNet model as an Att-Pred model

Attributes	Acc. (%)	Attributes	Acc. (%)	Attributes	Acc. (%)	Attributes	Acc. (%)
5 Shadow	94.0	Bushy Eyebrows	92.0	Mustache	96.0	Eyeglasses	100.0
Arched eyebrows	82.0	Gray Hair	98.0	Chubby	95.0	Double Chin	96.0
Bags under eyes	83.0	Goatee	96.0	Gray Hair	98.0	Heavy Makeup	92.0
Bald	99.0	High Cheekbones	86.0	Male	99.0	Mouth Slightly Open	91.0
Bangs	95.0	Narrow Eyes	88.0	No Beard	95.0	Oval Face	74.0
Big lips	76.0	Pale Skin	96.0	Pointy Nose	76.0	Receding Hairline	95.0
Big nose	83.0	Rosy Checks	94.0	Sideburns	96.0	Smiling	92.0
Black hair	86.0	Straight Hair	80.0	Wavy Hair	77.0	Wearing Earrings	85.0
Blond hair	94.0	Wearing Hat	99.0	Wearing Lipstick	93.0	Wearing Necklace	87.0
Wearing necktie	94.0	Blurry	96.0	Young	88.0	Attractive	80.0

methods in the PS technique. Finally, we study the effect of these perturbations on attacking the face images regarding the quality of the adversarial example and to what extent they mislead the model. For a single attribute, after we get the most significant attributes $C = (c_1, c_2, c_3, ..., c_K)$ from the prediction step, we use the generator G of StarGAN, which is already trained on facial attributes in the previous step. Here, G is composed of an encoder (G_E) and a decoder (G_D), as shown in Eq. (3). The (G_E) takes an input image x and the single significant attribute c_1, and then, we extract features in intermediate layers. After that, (G_D) takes the features as input and outputs the synthesized image.

For more technical details, Fig. 3 illustrates an SAA-StarGAN framework. First, we use x as an input to G_E with significant attribute c_1. This G_E shares the same architecture for the first half of StarGAN G [25]. Then, we extract the output features from different layers as a $conv$ layer f^*_{conv} and a residual block layer f^*_{res} of an encoder, as shown in Eqs. (4) and (5). This process is because we exploit an Attention Feature Fusion (AFF) method [46], which uses attention mechanisms to learn how to combine features from different layers. Dai et al. [46] prove that the addition and concatenation methods are not the best choice for combining features. Therefore, we use an AFF method to obtain the fused feature as an input to the decoder G_D. AFF is a framework combining features from different layers based on the Multi-Scale channel attention (MS) module. In our work, we follow the settings of an AFF method in a short skip connection scenario to overcome the inconsistent semantics among the input features and generate a more realistic image. Because this process helps gradients to flow more effectively during training, improving the overall performance [47]. More details about the AFF framework, including the MS module, are presented in [46]

$$G = G_E \circ G_D, \tag{3}$$

$$f^*_{\text{conv}} = G_E(x, c, \text{conv\_layer}), \tag{4}$$

$$f^*_{\text{res}} = G_E(x, c, \text{res\_layer}). \tag{5}$$

To obtain the fusion weights β, we calculate them from the attention weights generated by the MS module in AFF. As shown in Eq. (6), β is fusion weights where $\beta \in [0, 1]$. As a result, we update the value of β until the model is misleading, as shown in Eq. (7) to get the better-fused feature f^*. Finally, G_D takes a fused feature f^* as input and gets a synthesized face image x^* as an output, as shown in Eq. (8)

$$\beta = MS(f^*_{\text{conv}}, f^*_{\text{res}}), \tag{6}$$

$$f^* = \beta \cdot f^*_{\text{conv}} + (1 - \beta) \cdot f^*_{\text{res}}, \tag{7}$$

$$x^* = G_D(f^*). \tag{8}$$

Our work obtains the adversary x_{adv} by modifying a significant attribute c_1 for each image through feature-level interpolation. To achieve an impersonation attack, we use the L_2 loss function to minimize the distance between a face embedding of the generated adversary x_{adv} and a target image x_{tgt}, as shown in Eq. (9)

$$x_{adv} = Min_{x^*} \parallel TFV(x^*) - TFV(x_{tgt}) \parallel_2^2 . \tag{9}$$

The adversarial face image is crafted for dodging the attack by maximizing the distance between x_{adv} and the original image x in the feature space, as shown in the following:

$$x_{adv} = Max_{x^*} \parallel TFV(x^*) - TFV(x) \parallel_2^2 . \tag{10}$$

For multiple attributes, similarly, we use the same G of StarGAN, which has been used to modify a single attribute, but in this step, to change multiple significant attributes C in the feature space. In our experiments, we change two significant attributes, c_1 and c_2, represented as one-hot vectors. As a result, we obtain the synthesized image with a bigger change. After that, we study the effect of this change on attacking the face images.

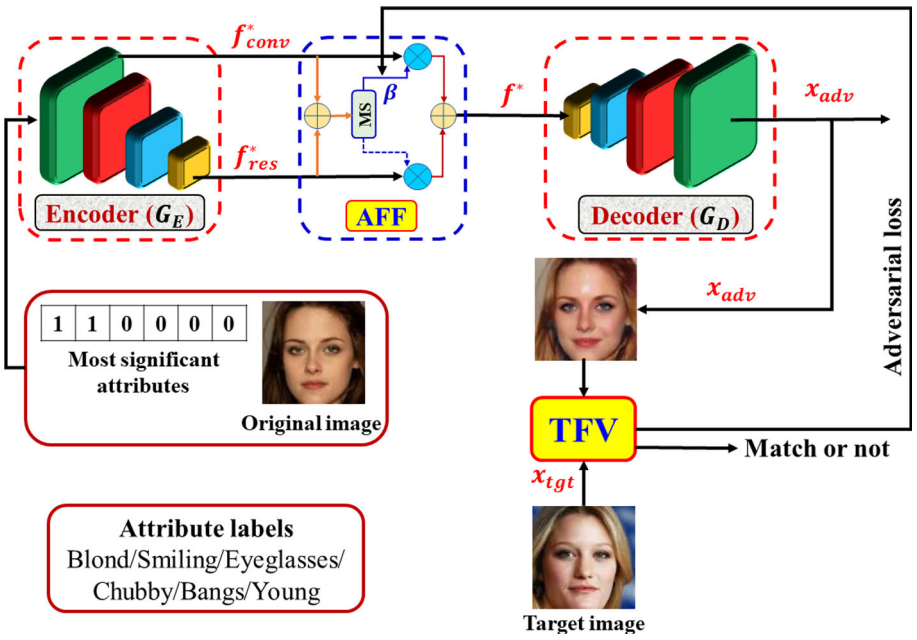


Fig. 3 The SAA-StarGAN attack framework. The original image and most significant attributes are fed into Encoder (G_E) to change these attributes and extract their features from different layers. The Attention Feature Fusion (AFF) framework is then used to generate β and perform fusion between features. After that, the fused features are sent to the decoder (G_D) to obtain the synthesis image. Finally, the Target Face Verification model (TFV) receives both the synthesis image and target identity to calculate the adversarial loss and optimize the β value at the feature level

3.2.3 SAA-StarGAN in Black-Box Setting

A black-box attack differs from a white-box attack as the adversary has no access to the model's gradient or parameters. Therefore, we propose modifications for our proposed method, SAA-StarGAN, to depend only on the target model's output in the black-box setting. To generate a successful attack, we need two steps: (1) predicting the most significant attributes of the target model. This step is similar to that in the white-box setting, where we depend on a CS method to predict the attributes. (2) Doing a linear search to find the largest γ value that affects the generated face image by changing the most significant attributes with γ for each face image until the output misleads the model.

Algorithm 3 illustrates in detail a black-box attack method. The main objective is to mislead the FV model into dodging or impersonation attacks. However, we also need the generated adversarial face image to be visually and semantically similar to the original for human perception. Therefore, we should first find the most important attributes that contribute the most to the prediction results and then modify them slightly by controlling the semantic similarity. We use the CS method to predict the important attributes. Then, we change the ordered most significant attributes in C for each input face image in an iterative loop. This step helps us achieve the goal of slightly changing an original face image and measuring the effect of the attributes on results. To change the most significant attributes C, we use G_E from a StarGAN model to get features. However, we make an iterative loop to alter the attributes according to the most important to the least until the adversarial condition is met. Besides, we use a bi-linear interpolation with a variable γ value to generate the fused features $f^*_{\gamma_i}$. Therefore, we create a vector γ that consists of 100 random values drawn from [0, 1]. This vector is applied to study the effect of each value γ in Eq. (11). Then, we do a linear search to find the optimal γ_{opt} value according to the change in the confidence score. The γ_{opt} is a random value from γ that is computed for each input image separately. Therefore, the optimal γ_{opt} value is not considered a fixed value that can be calculated once and used afterward to craft the adversarial face images. After that, the values are arranged according to the score change, and then, we select the optimal γ_{opt} value for an impersonation attack that decreases the distance between the facial embeddings of both the generated face image and a target face image. In contrast, we select the optimal γ_{opt} value for the dodging attack that maximizes the distance between the face embedding of the generated face image and the input face image. Finally, we substitute the optimal γ_{opt} value to generate the fused feature, which is then fed to a decoder to get the synthesized image x^* by the following equations:

$$f^* = \gamma_{opt} \cdot f^*_1 + (1 - \gamma_{opt}) \cdot f^*_{c_i}, \tag{11}$$

$$x^* = G_D(f^*). \tag{12}$$

To guarantee that a generated image preserves semantic similarity from the original face image, we need to measure the semantic similarity sim between a generated face image and an input face image to filter out the face images that are not realistic and control their quality. The adversarial face image is found when an adversarial criterion is achieved and the semantic similarity is above a threshold th ($sim > th$). Moreover, we use the L_2 loss function to measure the distance between two identity embeddings from the FV model. For an

Algorithm 3 The SAA-StarGAN for Black-Box Attack

Require: Original image x, original attribute c_{ori}, most significant attributes $\{c\}$, encoder G_E, decoder G_D, target face verification model TFV, target image x_{tgt}, L_2 distance function $dist$, threshold_sim th

Ensure: Adversarial example x_{adv}

1: $f_1^* \leftarrow G_E\ (x, c_{ori})$
2: $\gamma \leftarrow$ create an array in the range [0,1]
3: **for** c_i in most significant attributes C **do**
4: $\quad f_{c_i}^* \leftarrow G_E\ (x, c_i)$
5: $\quad$ **for** γ_i in γ **do**
6: $\quad\quad f_{\gamma_i}^* = \gamma_i \cdot f_1^* + (1\text{- }\gamma_i) \cdot f_{c_i}^*$
7: $\quad\quad x_{\gamma_i}^* = G_D\ (f_{\gamma_i}^*)$
8: $\quad\quad d_{\gamma_i} \leftarrow dist\ (TFV(x_{tgt}, x_{\gamma_i}^*)$
9: $\quad\quad score.insert(d_{\gamma_i}, \gamma_i\ )$
10: $\quad$ **end for**
11: $\quad$ **if** impersonation attack **then**
12: $\quad\quad \gamma_{opt} = argmax_\gamma\ (score)$
13: $\quad$ **else if** dodging attack **then**
14: $\quad\quad \gamma_{opt} = argmin_\gamma\ (score)$
15: $\quad$ **end if**
16: $\quad$ Substitute γ_{opt} in f^*
17: $\quad x^* = G_D\ (f^*)$
18: $\quad$ **if** $sim(x, x^*) \leq th$ **then**
19: $\quad\quad$ Return None
20: $\quad$ **end if**
21: $\quad$ **if** $dist(TFV(x^*, x_{tgt}) \leq T)$ **then**
22: $\quad\quad x_{adv} \leftarrow x^*$
23: $\quad\quad$ break
24: $\quad$ **else**
25: $\quad\quad x \leftarrow x^*$
26: $\quad\quad c_i \leftarrow$ add the next attribute from C
27: $\quad$ **end if**
28: $\quad$ **if** $dist(TFV(x^*, x) \geq T)$ **then**
29: $\quad\quad x_{adv} \leftarrow x^*$
30: $\quad\quad$ break
31: $\quad$ **else**
32: $\quad\quad x \leftarrow x^*$
33: $\quad\quad c_i \leftarrow$ add the next attribute from C
34: $\quad$ **end if**
35: **end for**

impersonation attack, we minimize the distance between the face embedding of x_{adv} and x_{tgt}, as shown in Eq. (9). For a dodging attack, we maximize the distance between the face embedding of x_{adv} and x, as shown in Eq. (10). If not, the above steps will repeat to add the next significant attribute in the ordered significant attributes until we find an adversarial example (Lines 18 - 34). Details of the implementation can be found at https://github.com/Code5432/Adv.

3.3 Random Attribute Selection-Based Attack

This subsection presents a preliminary method, the Random Selection, as a baseline method to compare with our SAA-StarGAN method. We generate adversarial face images based on randomly selecting a set of attributes. These attributes are changed using a StarGAN model. Therefore, we make two copies of the generator from StarGAN and re-train them separately using two sets of attributes representing each attribute a as a one-hot vector. The first generator, G_1, is trained by the first set, S_1, including black hair color, blond hair color, heavy makeup, gender, and pale skin. The second generator, G_2, is trained by the second set, S_2, which consists of smiling, mouth slightly open, bangs, eyeglasses, and young. As a result, the overall Random Selection system involves three components, namely G_1, G_2, and the TFV model.

First, G_1 takes an original image x with a specific attribute a_{s_1} from S_1 to generate image synthesis x_1^* with dimensions H, W, and L for height, width, and channels, respectively. The second image synthesis x_2^* is then obtained using the translated image x_1^* as the input to G_2 with another attribute a_{s_2} from S_2, as shown in Eqs. (13) and (14). Note that the two attributes are chosen randomly from sets S_1 and S_2. Through permutations between the attributes from the previous sets, we change two attributes using a StarGAN model. Finally, interpolation is applied between the pair of images produced from G_1 and G_2 to generate an adversary x_{adv}. We assign the interpolation parameter by a tensor α, whose element values are in [0, 1], with the size $H \times W \times L$ that matches with the size of x_1^*. Each value of α within the specified size lies in the range [0,1], as shown in Eq. (15) [48], and this value is updated until a stopping condition is met. According to the presented procedure, 25 adversarial face images are generated for each x due to the permutation between different attributes from the two sets

$$x_1^* = G_1(x, a_{s_1}), \tag{13}$$

$$x_2^* = G_2(G_1(x, a_{s_1}), a_{s_2}), \tag{14}$$

$$x_{adv} = \alpha \cdot x_1^* + (1 - \alpha) \cdot x_2^*. \tag{15}$$

4 Experimental Setup

In this section, We provide an experimental setup, including the dataset, targeted models, baselines, evaluation metrics, and implementation details of our method.

4.1 Dataset

In our work, we select the CelebA dataset [49]. The CelebA dataset is the most popular face recognition task, with 202,599 face images with 40 facial attributes and 10,177 identities. We use facial attributes to train a StarGAN model in our work. Moreover, we randomly chose 5000 different identities to generate the adversarial face images. Image values are normalized between 0 and 1.

4.2 Target Face Verification Models

To evaluate the effectiveness of the proposed SAA-StarGAN, we choose ten state-of-the-art FV models containing differ-

ent model architectures and training loss functions. We use two of them as white-box TFV models: FaceNet [44] and ArcFace [45]. Besides, we use two other publicly available models, SphereFace [50] and CosFace [51], for evaluation. Also, we select different trained models under different backbones and loss functions such as ResNet-101 [52], IResNet50 [45, 47], MobileFace [53], and ShuffleNet V2 [54] to demonstrate the effectiveness of our SAA-StarGAN on different models.

4.3 Baselines

To evaluate our attack method, we compare our method with four types of attacks: gradient-based, transformation-based, patch-based, and stealthy-based. For gradient-based methods, we choose the typical one-step attack method of FGSM [4], BIM [7], PGD [30], and MI-FGSM [8]. In addition, we apply DIM [10] and TIM [11] in transformation-based methods. PGD, MI-FGSM, DIM, and TIM are famous for their strong attack ability. On the other hand, we perform face attack methods such as Sticker and Face Mask attacks [41] as the representative of patch-based methods. Finally, we select the Random Selection method presented in Sect. 3.3 and SemanticAdv [18] for stealthy-based methods. SemanticAdv is the most comparable method to ours, which modifies a face attribute to generate adversarial face images. For attacking settings, we apply five facial attributes for G_1: hair color (black-blond), heavy makeup, gender, and pale skin, while for G_2, smiling, mouth slightly open, bangs, eyeglasses, and young attributes in a Random Selection method. Conversely, we set a maximum perturbation $\epsilon = 8/255$, step size = 1.6, decay factor $\mu = 1$, transformation probability $p = 0.5$ for DIM, and the Gaussian kernel with size 7×7 for TIM.

4.4 Evaluation Metrics

We use several evaluation matrices to estimate our attack effectiveness over different baselines. We select the attack success rate (ASR) to evaluate the adversarial face images crafted by SAA-StarGAN.

ASR used for impersonation attacks is computed as follows:

$$\text{success\_rate} = \frac{(\#\text{ImagePairs}(x_{adv}, x_{tgt}) \geq T_s)}{(\#\text{TotalImagePairs})}, \tag{16}$$

where ImagePairs consists of an adversarial face image generated by SAA-StarGAN and the matched target face. ASR used for dodging attacks is computed as follows:

$$\text{success\_rate} = \frac{(\#\text{ImagePairs}(x_{adv}, x) < T_s)}{(\#\text{TotalImagePairs})}, \tag{17}$$

where ImagePairs consists of an adversarial face image generated by SAA-StarGAN and the input face image. Also, to measure the similarity between images, we use a cosine similarity and a threshold T @ 0.1 % FPR for each TFV model. Finally, we compute the Mean Square Error (MSE) [17] and the Structural Similarity Index Measure (SSIM) [55] to evaluate the quality between the original face images x and the adversarial face images x_{adv} for different attacks.

4.5 Implementation Details

The generator of a StarGAN model [25] is the dependent architecture in our attack method to craft high-quality semantic facial images. We re-train StarGAN on facial attributes to use the trained model in the significant attributes prediction step. Then, we use an encoder from a StarGAN generator, consisting of the first half of the layers for the architecture of StarGAN, to take an input image and the modified attribute. In contrast, a decoder is the second part of the architecture. In our attack method, we use the Adam optimizer [56] with a fixed learning rate of 0.05 and up to 300 epochs for all experiments. SAA-StarGAN is implemented using PyTorch v1.7.0. All experiments are conducted on a single Titan X GPU to generate adversarial face images. All selected face images pass through an MTCNN detector [57] to detect the face image and align images for an entire image.

5 Experiments

To validate the effectiveness of SAA-StarGAN, we empirically evaluate our adversarial face images and illustrate that SAA-StarGAN achieves higher attack success rates against different FV models. On the other hand, SAA-StarGAN boosts adversarial transferability with high efficiency. The black-box setting demonstrates that SAA-StarGAN achieves high attack success rates significantly and shows the importance of significant attribute prediction. Besides, SAA-StarGAN can generate realistic and diverse adversarial face images. Tables 2 and Table 3 list the ASRs of various methods, using FaceNet and ArcFace models to generate adversarial examples, respectively, for impersonation and dodging attacks.

5.1 Comparison for White-Box Attacks

To validate the efficacy of our attack method, we first compare SAA-StarGAN with the baselines in the white-box setting. For a fair comparison, we train these baseline methods using FaceNet and ArcFace models on the CelebA dataset with T @ 0.1% FPR for impersonation and dodging attacks. Then, we compute the ASR for each method. The third column in Table 2 compares our SAA-StarGAN and the baselines in

the white-box setting on a FaceNet model for an impersonation attack in (a) and a dodging attack in (b). All variants of the proposed SAA-StarGAN method have reached a nearly 100% ASR, confirming that SAA-StarGAN can mislead the TFV models successfully. The fourth column in Table 3 compares our SAA-StarGAN and the baselines in the white-box setting on an ArcFace model for the impersonation attack in (a) and a dodging attack in (b). For all the cases, our SAA-StarGAN can effectively craft adversarial face images to fool TFV models in the white-box setting.

5.2 Comparison for Black-Box Attacks

Most face recognition systems do not allow access to any internal information of the neural networks. In the black-box setting, the weights and network architectures are not included in the training process. Therefore, it is essential to evaluate the vulnerability of FRs in both the transfer-based attack setting and the attack based on score confidence.

5.2.1 Transferability Analysis

The transferability across different models is one of the most important properties of adversarial examples. To demonstrate the adversarial face image transferability generated by SAA-StarGAN under two white-box models, we construct a dataset from successful adversaries crafted by a FaceNet model. Then, we evaluate an ASR on nine different TFV models. In contrast, we also use adversarial images generated by an ArcFace model to assess the ASR. We can observe from Table 2 that for impersonation attacks, our SAA-StarGAN achieves a high ASR under different experimental conditions compared to various baselines. This is attributed to its dependence on selecting the significant attribute for each input image given to TFV. The ASR of our SAA-StarGAN is significantly higher than other gradient-based methods by 23.4% on SphereFace. In contrast, the ASR of the Sticker and Face mask attacks has not exceeded 17%. Transformation-based methods achieve better transferability than gradient-based and patch-based methods. For the Random Selection and SemanticAdv methods, the ASR of our method is still higher by a clear margin. Although the Random Selection and SemanticAdv methods craft the adversaries by modifying the facial attributes, they only depend on changing random attributes or a single fixed attribute. We can see that the SAA-StarGAN-CS transferability for a single attribute surpasses that of other methods in all cases. For example, attacking a ShuffleNet V2 model using adversarial face images crafted by FaceNet by the SAA-StarGAN-CS achieves the best ASR of 54.30%, outperforming the baseline attacks by 17.1–43.1%. Finally, we conclude that our SAA-StarGAN method outperforms others on all models besides maintaining a high ASR on the white-box setting.

Similarly, our attack outperforms the other baselines significantly in dodging attacks, as presented in Table 2. We can see that SAA-StarGAN achieves a high ASR of 77.2% on SphereFace. In addition, our method beats transformation-based, patch-based, gradient-based, and stealthy-based methods by 3.4, 14.1, 16.1, and 37.7%, respectively. We observe that the TIM method exceeds our method on CosFace and IR50-SphereFace by 0.6 and 1.2%, respectively. We also observe that the TIM method exceeds our method on CosFace and IR50-SphereFace by 0.6 and 1.2%, respectively. However, our method still surpasses the other baselines against different TFV models. Therefore, we infer that transformation-based methods are effective for transferability under dodging attacks after our SAA-StarGAN.

Moreover, we analyze the attack performance of the generated face images crafted by ArcFace against different models, whether under impersonation or dodging attacks. In Table 3, our SAA-StarGAN significantly improves the transferability of adversarial face images over the baseline attacks. The main reason is that SAA-StarGAN depends on manipulating the most important attributes that affect the decision. As clear from Table 3, SAA-StarGAN-CS outperforms the baselines of transformation-based, gradient-based, stealthy-based, and patch-based by 2.6, 11.0, 31.0, and 66.0%, respectively, under the impersonation attack, against IR50-CosFace model. Except the TIM method exceeds our SAA-StarGAN by 0.4% against a MobileFace model. To dodge an attack, the adversarial face images crafted by SAA-StarGAN-CS against the MobileFace model surpass transformation-based, gradient-based, stealthy-based, and patch-based by 0.1, 18.8, 13.1, and 20.1%, respectively. Finally, we observe that the models within the same backbone have good transferability. Therefore, the generated adversarial face images crafted on the ArcFace model against IR50-Softmax, IR50-CosFace, and IR50-SphereFace have high transferability. In addition, ShuffleNet V2 and MobileFace have light weights, which are easily attacked by the adversarial face images generated on ArcFace. In contrast, the FaceNet model and ResNet-101 are challenging to transfer to other models, where they are trained on IncepetionResNetV2 and ResNet-101 based on the softmax loss, respectively. Besides, the main advantage of SAA-StarGAN-CS is that it uses different TFV models easily and efficiently to calculate the cosine similarity-based-important attributes. In this paper, we have analyzed different techniques to predict the significant attributes and evaluate them affecting attacking FR tasks.

5.2.2 Comparison Based on Score Confidence

We also evaluate the performance of SAA-StarGAN of black-box attacks based on score confidence directly [58] to empirically demonstrate that the proposed method in the black-box setting can generate adversarial images that mislead differ-

Table 2 Transferability of the adversarial examples generated by SAA-StarGAN and the baselines against black-box models. We use adversarial examples generated on a FaceNet model to attack nine different TFV models for impersonation & dodging attacks

Method	Targeted model	FaceNet	ArcFace	ResNet-101	CosFace	SphereFace	MobileFace	ShuffleNet V2	IR50-Softmax	IR50-CosFace	IR50-SphereFace
(a) Impersonation attack											
Gradient-based	FGSM	88.3[a]	07.2	02.4	11.2	15.2	09.6	13.1	13.4	07.2	09.3
	BIM	90.0[a]	11.8	07.8	18.1	18.7	17.6	18.2	13.9	08.7	10.8
	PGD	99.5[a]	12.1	10.3	21.7	23.7	20.4	22.9	16.7	14.8	18.6
	MI-FGSM	99.7[a]	12.4	11.2	28.1	29.6	22.3	28.6	18.8	17.9	20.4
Transformation-based	DIM	96.6[a]	19.5	14.5	32.6	36.2	29.2	36.1	22.6	20.2	26.4
	TIM	97.7[a]	20.6	16.3	34.8	40.8	30.6	37.2	25.6	22.0	32.8
Stealthy-based	SemanticAdv	99.8[a]	32.4	09.9	22.1	29.0	28.1	35.7	16.2	19.5	14.7
	Random Selection	99.7[a]	31.6	09.5	19.9	25.1	27.7	31.4	17.2	29.5	16.2
Patch-based	Sticker	100.0[a]	09.7	07.4	10.4	11.5	10.7	11.2	9.9	10.5	11.1
	Face Mask	95.4[a]	10.2	09.6	13.7	17.0	11.9	15.7	12.1	11.4	13.0
Ours	SAA-StarGAN-CS	100.0[a]	**46.8**	**31.5**	**40.3**	**53.0**	**48.2**	**54.3**	**37.8**	35.5	**36.2**
	SAA-StarGAN-PS	100.0[a]	45.5	31.3	39.6	50.0	48.1	53.2	34.5	**36.5**	35.3
	SAA-StarGAN-CS-M	100.0[a]	44.3	29.0	39.9	48.7	46.7	52.0	34.9	34.8	36.0
	SAA-StarGAN-PS-M	100.0[a]	46.2	30.5	40.2	50.6	47.0	53.3	34.3	35.7	34.4
(b) Dodging attack											
Gradient-based	FGSM	92.6[a]	24.9	12.5	19.5	28.3	24.6	27.0	27.6	19.6	20.4
	BIM	97.3[a]	26.1	18.5	30.1	30.7	28.1	31.3	28.3	18.7	23.7
	PGD	100.0[a]	50.0	27.3	62.4	60.6	49.9	62.0	41.5	37.6	42.9
	MI-FGSM	100.0[a]	51.1	28.0	60.3	61.1	50.1	62.8	43.2	42.2	48.4
Transformation-based	DIM	99.0[a]	53.2	32.2	63.9	70.2	51.4	65.0	45.1	43.2	53.8
	TIM	99.3[a]	55.4	34.0	**65.7**	73.8	52.6	69.6	46.7	44.8	**60.4**
Stealthy-based	SemanticAdv	100.0[a]	45.2	20.4	41.0	39.5	31.0	48.1	32.4	33.1	27.6
	Random Selection	100.0[a]	43.2	20.0	40.2	36.2	30.8	42.9	36.2	47.2	28.0
Patch-based	Sticker	100.0[a]	21.4	9.8	22.1	23.5	27.4	26.4	19.9	21.4	23.4
	Face Mask	100.0[a]	55.6	31.2	63.5	63.1	52.1	63.4	40.1	38.4	43.7
Ours	SAA-StarGAN-CS	100.0[a]	**65.3**	**44.6**	65.1	**77.2**	**53.6**	**76.8**	**49.5**	**53.4**	55.4
	SAA-StarGAN-PS	100.0[a]	62.4	43.5	61.4	71.8	53.8	69.4	47.3	49.5	54.3
	SAA-StarGAN-CS-M	100.0[a]	51.5	41.1	61.2	61.4	54.9	63.2	42.8	47.6	59.2
	SAA-StarGAN-PS-M	100.0[a]	54.7	42.8	64.4	66.76	54.5	66.0	47.2	48.9	51.0

Values represent the attack success rate (%). The numbers marked in bold represent the best attack success rate

[a]White-box attacks

Table 3 Transferability of the adversarial examples generated by SAA-StarGAN and the baselines against black-box models. We use adversarial examples generated on an ArcFace model to attack nine different TFV models for impersonation and dodging attacks

Method	Targeted model	FaceNet	ArcFace	ResNet-101	CosFace	SphereFace	MobileFace	ShuffleNet V2	IR50-Softmax	IR50-CosFace	IR50-SphereFace
(a) Impersonation attack											
Gradient-based	FGSM	15.2	96.8[a]	09.9	19.9	21.4	35.7	29.8	20.1	30.0	23.3
	BIM	18.0	100.0[a]	14.6	22.6	25.3	39.1	32.7	25.0	33.0	29.5
	PGD	24.9	99.9[a]	23.4	41.2	45.5	52.1	47.6	43.7	58.9	43.0
	MI-FGSM	25.0	100.0[a]	22.9	47.9	55.4	65.1	55.3	50.4	69.5	50.9
Transformation-based	DIM	33.8	100.0[a]	30.7	51.7	60.0	73.1	62.2	55.1	73.4	55.6
	TIM	36.8	100.0[a]	34.2	53.5	61.9	**74.7**	64.5	57.3	77.9	57.1
Stealthy-based	SemanticAdv	30.4	95.8[a]	17.5	32.6	45.6	51.5	47.3	37.5	45.0	30.7
	Random Selection	30.6	97.8[a]	19.4	33.4	38.9	50.1	48.7	38.0	49.5	27.3
Patch-based	Sticker	18.7	100.0[a]	8.9	14.0	10.1	21.4	20.9	15.8	12.4	13.2
	Face Mask	19.4	100.0[a]	9.3	16.5	11.4	22.6	21.9	17.0	14.5	13.9
Ours	SAA-StarGAN-CS	**37.2**	99.7[a]	**27.6**	**54.2**	**62.1**	74.3	**65.6**	**65.9**	**80.5**	**58.5**
	SAA-StarGAN-PS	36.7	99.8[a]	26.6	52.0	61.5	73.5	64.1	64.7	78.1	57.5
	SAA-StarGAN-CS-M	34.2	99.1[a]	24.0	53.0	59.4	67.1	57.0	61.9	73.0	52.9
	SAA-StarGAN-PS-M	35.9	99.2[a]	26.3	53.7	61.5	68.5	59.1	64.4	76.6	56.4
(b) Dodging attack											
Gradient-based	FGSM	34.1	99.0[a]	32.7	46.7	44.7	66.1	62.7	49.2	54.3	45.5
	BIM	35.6	100.0[a]	35.8	51.5	49.0	67.2	66.5	51.0	57.4	49.9
	PGD	47.7	99.9[a]	39.2	63.2	70.6	72.6	66.9	64.9	67.7	55.5
	MI-FGSM	48.2	100.0[a]	41.5	66.0	73.0	75.8	70.0	66.8	69.8	56.7
Transformation-based	DIM	51.2	100.0[a]	46.8	75.4	88.1	93.2	88.8	74.8	87.0	67.0
	TIM	52.1	100.0[a]	47.2	**77.5**	**90.1**	94.5	**91.3**	76.1	88.9	68.7
Stealthy-based	SemanticAdv	44.4	99.9[a]	37.4	53.2	64.6	80.5	76.3	66.5	64.0	52.7
	Random Selection	45.6	99.8[a]	38.9	54.8	60.3	81.5	77.1	69.4	61.9	53.6
Patch-based	Sticker	21.3	100.0[a]	25.4	27.8	30.1	40.2	38.4	24.7	26.8	27.4
	Face Mask	45.0	100.0[a]	42.4	64.7	70.0	74.5	70.5	68.7	69.1	58.4
Ours	SAA-StarGAN-CS	**53.2**	100.0[a]	**48.0**	75.6	86.4	**94.6**	89.1	**77.1**	**89.3**	**69.9**
	SAA-StarGAN-PS	52.4	100.0[a]	47.2	73.3	83.8	93.9	88.7	76.0	88.1	64.8
	SAA-StarGAN-CS-M	49.2	100.0[a]	44.6	72.4	80.4	91.1	86.0	70.9	72.1	61.9
	SAA-StarGAN-PS-M	50.9	100.0[a]	46.6	72.7	81.5	92.5	87.1	73.4	75.6	65.4

Values represent the attack success rate (%). The numbers marked in bold represent the best attack success rate

[a] White-box attacks

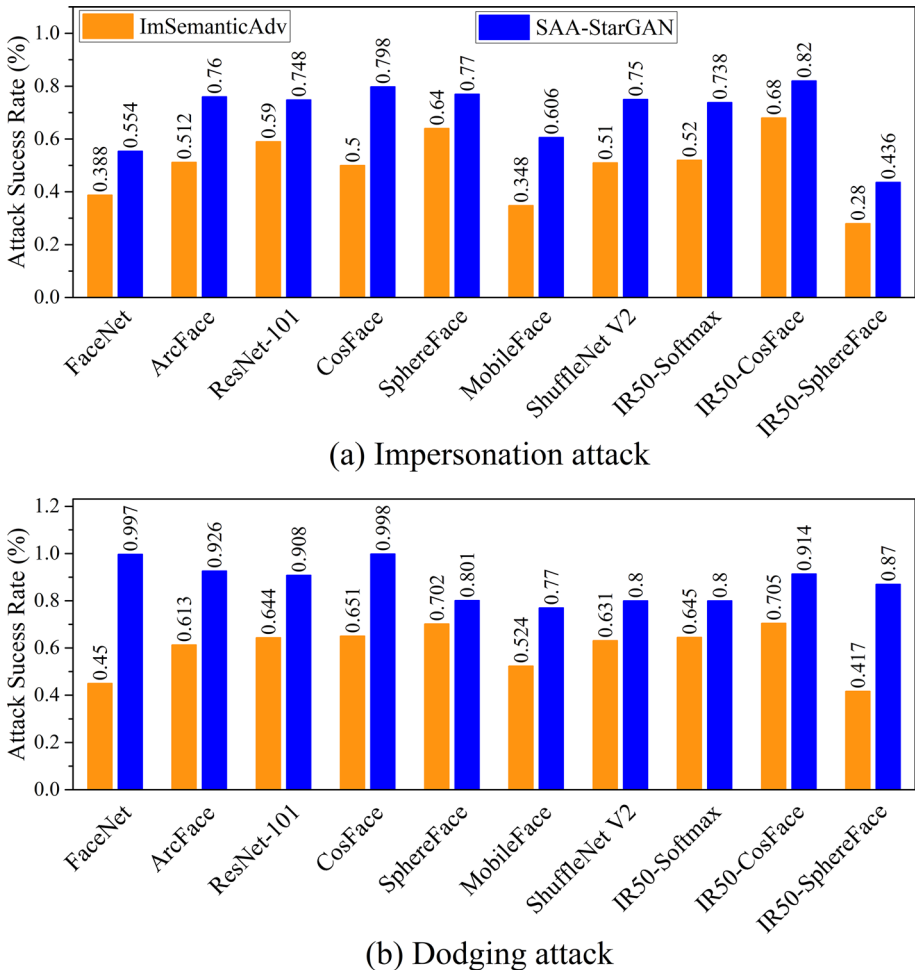


Fig. 4 The average attack success rate (%) for black-box attacks. SAA-StarGAN outperforms ImSemanticAdv by a clear margin

ent FV models. We have improved the SemanticAdv method (denoted as ImSemanticAdv) to add multiple attributes in the black-box setting based on the random attribute selection. We compare SAA-StarGAN with ImSemanticAdv, which follows our method's procedure when changing attributes. This method is performed by making a loop to change the random attributes until reaching the adversary randomly. The main results of black-box attacks based on score confidence for the impersonation and dodging are shown in Fig. 4. We can observe that SAA-StarGAN outperforms ImSemanticAdv across all different models by a large margin for impersonation and dodging black-box attacks. Compared with ImSemanticAdv, SAA-StarGAN improves the average ASR by 13–29.8% and 9.9–54.7% under impersonation and dodging attacks, respectively. Therefore, predicting the most important attributes of each image significantly affects the attack's performance.

5.3 Evaluations on Cosine Similarity

To emphasize and clarify the effectiveness of the proposed SAA-StarGAN method in improving the transferability against different models. We use adversarial face images crafted on FaceNet from an attack method against the SphereFace model under the impersonation attack. Besides, we exploit the generated adversarial face images on FaceNet for the dodging attack with the original images. Then, we measure the cosine similarity scores before and after the attack to see the improvement percentage for each attack method against the SphereFace model. In Fig. 5, we observe that most scores between the generated face and target faces of the SAA-StarGAN method fall above T_s @ 0.1% FPR, demonstrating that the generated face images can be falsely accepted by 53, 50, 48.7, and 50.6% in an impersonation attack. In contrast to the baseline methods, a few scores fall above T_s. For the dodging attack, the scores fall below T_s @ 0.1% FPR by 77.2, 71.8, 61.4, and 66.7%, demonstrating that the SphereFace model can falsely reject the image pairs. We conclude that the SAA-StarGAN significantly outperforms the baseline attacks to improve the transferability of black-box models.

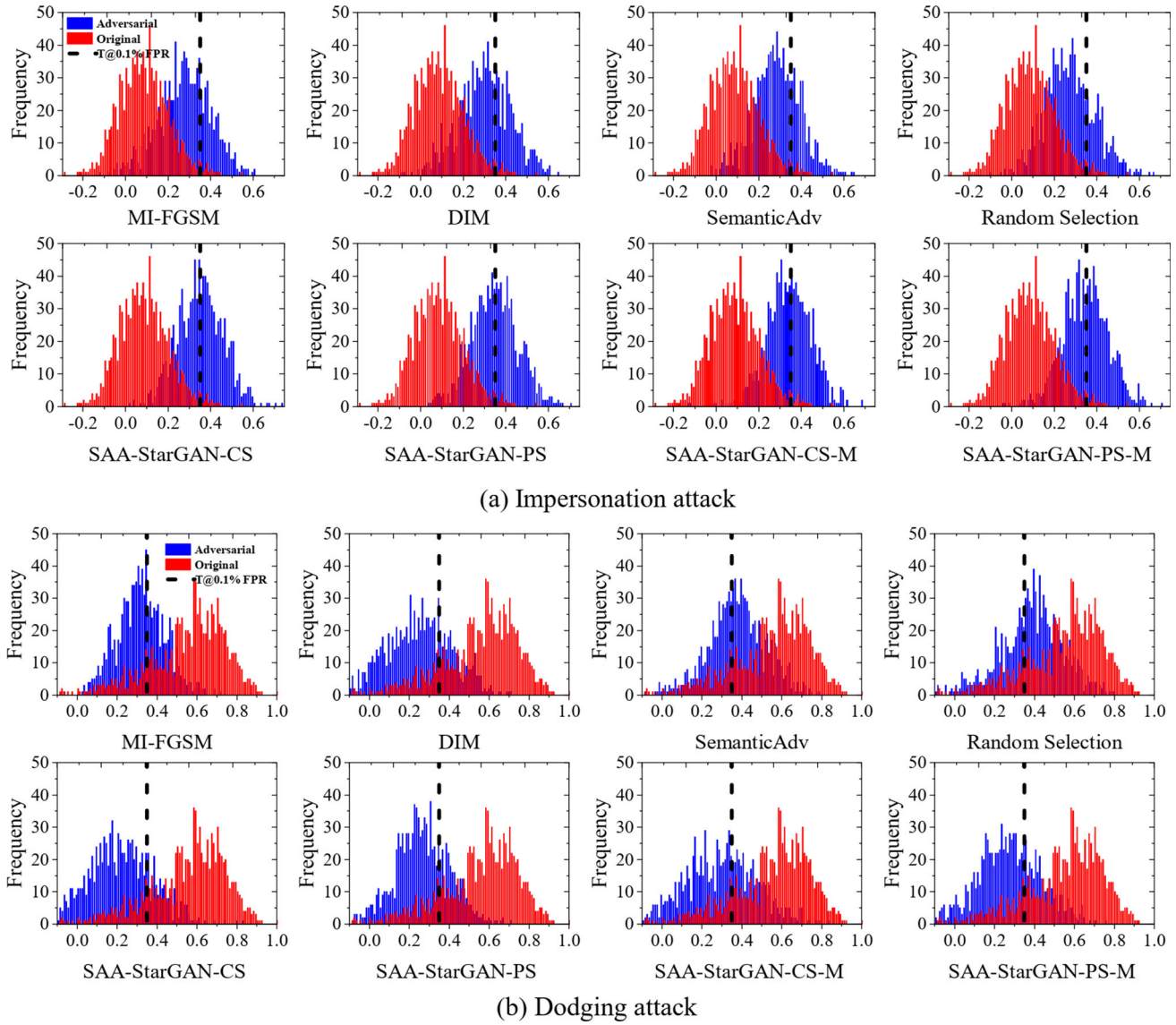
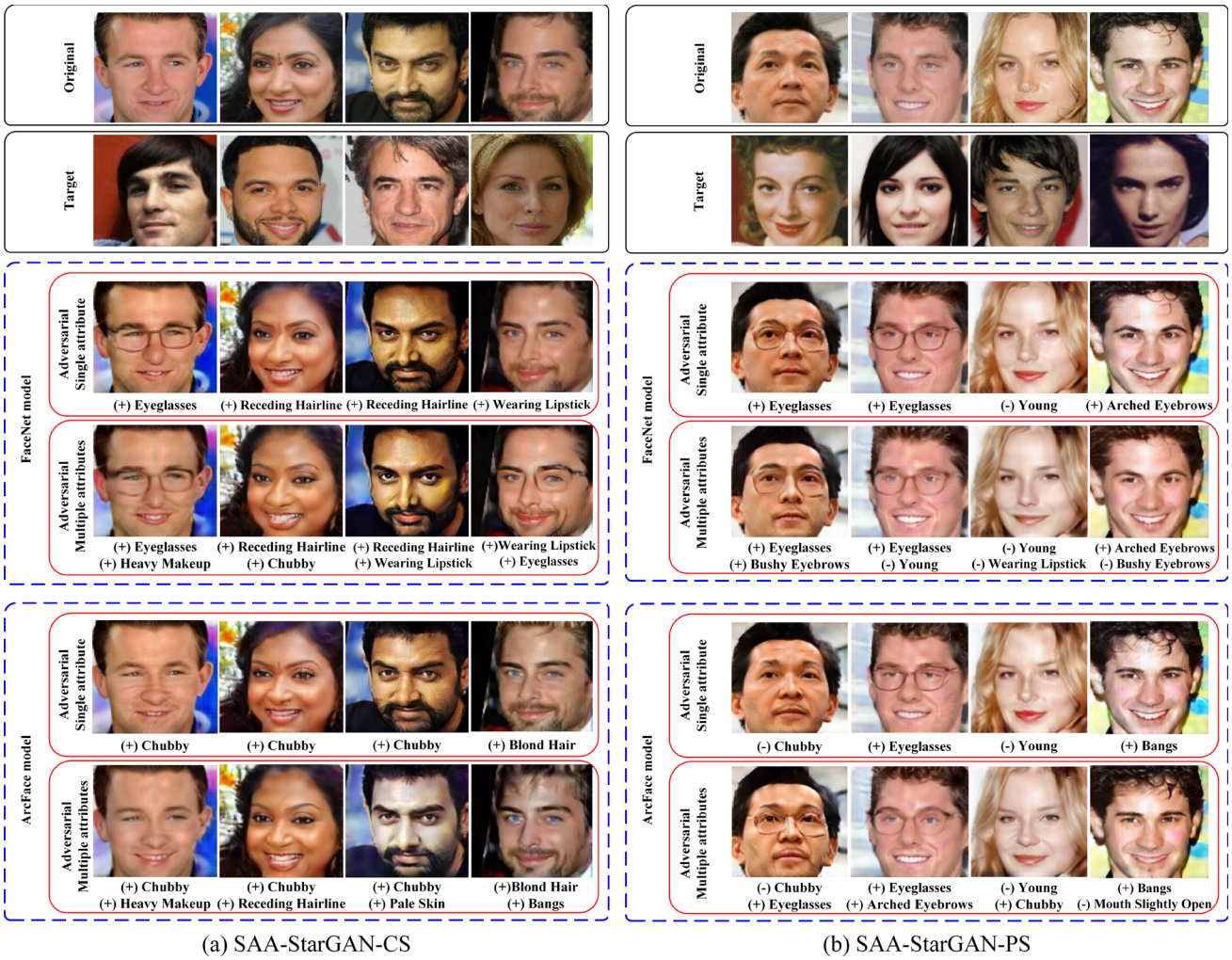


Fig. 5 Shift in cosine similarity scores for SphereFace before and after adversarial attacks generated by SAA-StarGAN and the baselines on FaceNet. The shift of SAA-StarGAN is much more distinct as compared to the baselines

5.4 Visualization for SAA-StarGAN

Adversarial face images aim to mislead the FV models without fooling humans. Consequently, we illustrate samples of adversarial face images generated by SAA-StarGAN-CS and SAA-StarGAN-PS on FaceNet and ArcFace in the white-box setting. As illustrated in Fig. 6, the first two rows illustrate the original image and the target image, while the third and fourth rows present the generated adversarial face images on FaceNet for single and multiple attributes, respectively. The last two rows represent the generated adversarial face images on ArcFace for single and multiple attributes, respectively. According to the label of images, sign (+) indicates the addition of an attribute, while sign (−) denotes removing an attribute. We can see that the proposed method produces realistic images with slight changes. Figure 7 presents a set of adversarial face images generated on FaceNet based on the score confidence in the black-box setting. We can see that our proposed method generates realistic and diverse images except for a few images. On the other hand, Fig. 8 illustrates the comparison between SAA-StarGAN and some of the baselines on the FaceNet model. Although two attributes are applied to the proposed SAA-StarGAN-CS and SAA-StarGAN-PS methods, the realism of the generated adversarial face images is close enough to those produced by SemanticAdv with a single attribute. In addition, the adversarial face images generated by MI-FGSM, PGD, DIM, and TIM are unclear compared to our SAA-StarGAN.



(a) SAA-StarGAN-CS (b) SAA-StarGAN-PS

Fig. 6 Adversarial face images generated by SAA-StarGAN in the white-box setting

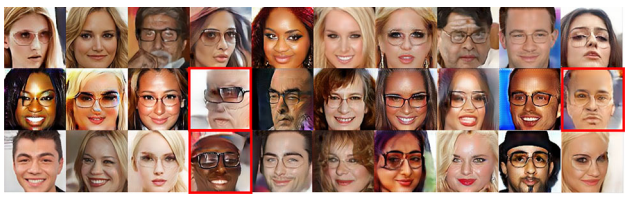


Fig. 7 Adversarial face images generated by SAA-StarGAN in the black-box setting based on the score confidence. The figure illustrates the realistic face images, and the red borders indicate a few unrealistic adversarial face images generated by SAA-StarGAN

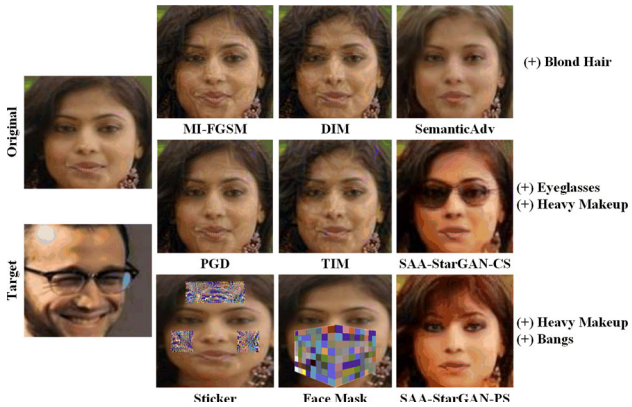


Fig. 8 The adversarial face images are generated on FaceNet by different methods. SAA-StarGAN can generate high-quality adversarial face images as compared to the baselines

For the Sticker and Face Mask attacks, the face images generated are unrealistic and misleading to the human eye. As known, the main objective of crafting an adversarial face image is to deceive TFV models while avoiding affecting human perception in FR. We can conclude that this goal has been successfully achieved in the current study, as the presented faces are realistic and can be easily identified.

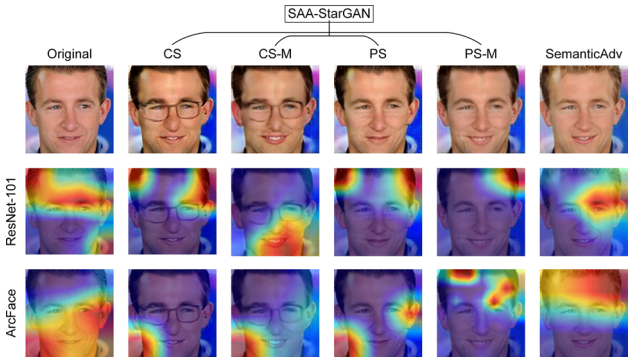


Fig. 9 Grad-CAM attention visualization for ResNet-101 and ArcFace models being attacked with adversarial face images generated on the FaceNet model. This Figure compares our SAA-StarGAN based on the important facial attributes and SemanticAdv based on indiscriminate attributes

5.5 Visualization of the Attention

In this subsection, we apply the Gradient-weighted Class Activation Mapping (Grad-CAM) [59], an attention visualization technique, to find the discriminating regions in the image according to a TFV model. We use it to show the attention of the ArcFace and ResNet-101 models for both the original and the generated adversarial face images on the FaceNet model. In Fig. 9, the first column shows an original image, while the second-to-fifth columns illustrate SAA-StarGAN with four variants according to the CS and PS techniques. Finally, SemanticAdv displays in the sixth column. The Grad-CAM attention map shows that our SAA-StarGAN focuses on the trivial regions instead of the prominent regions for both ResNet-101 and ArcFace models. Thus, SAA-StarGAN could improve the attack transferability significantly. In contrast, the Grad-Cam attention map shows the generated face images by the SemanticAdv method that focuses on the most prominent regions in a face image. The reason is that our method depends on altering the significant attribute for each input image according to the used TFV. On the contrary, SemanticAdv has changed a random attribute, where this attribute considers fixed on all images. Therefore, SemanticAdv is weaker than our method in transferability. The prediction step for the most significant attributes is critical in improving the attack transferability.

5.6 Similarity to the Original Image

It is evident in the literature that most methods succeed in generating adversarial face images. Therefore, we aim to measure the quality between the adversarial face and original images through MSE and SSIM. MSE measures the absolute errors that depend strongly on the image intensity scaling, so the lowest value is the best. However, it poorly correlates with human perception of the visual system. SSIM overcomes this issue, which is used for measuring the similarity between two images by predicting the perceived quality of images. Therefore, the highest value is the best.

We compare the adversarial face images crafted by SAA-StarGAN-CS and SAA-StarGAN-PS methods for single and multiple attributes with the baseline methods. We select 2000 original images and 2000 adversarial face images for different attack methods for evaluation. After that, we calculated the MSE and SSIM for each method separately, as shown in Table 4. SAA-StarGAN-PS and SAA-StarGAN-CS for a single attribute show the largest values of SSIM compared to

Table 4 MSE and SSIM to compare the original and adversarial face images generated on FaceNet and ArcFace models by our SAA-StarGAN and the different baselines

Attacks	FaceNet		ArcFace	
	MSE (↓)	SSIM (↑)	MSE (↓)	SSIM(↑)
FGSM	0.024	0.708	0.026	0.705
BIM	0.026	0.707	0.025	0.709
PGD	0.025	0.725	0.025	0.718
MI-FGSM	0.027	0.720	0.028	0.710
DIM	0.018	0.810	0.018	0.809
TIM	0.017	0.807	0.018	0.808
SemanticAdv	0.023	0.813	0.023	0.813
Random Selection	0.034	0.727	0.034	0.726
Sticker	0.958	0.009	0.959	0.008
Face Mask	0.968	0.007	0.969	0.006
SAA-StarGAN-CS	0.022	0.822	0.020	0.821
SAA-StarGAN-PS	**0.015**[a]	**0.854**	**0.016**[a]	**0.846**
SAA-StarGAN-CS-M	0.040	0.729	0.036	0.735
SAA-StarGAN-PS-M	0.026	0.792	0.026	0.784

Best values appear in bold
[a] indicates the best values of MSE

others. Besides, they have the lowest values of MSE, which gives them preference and makes them more suitable for generating adversarial face images. The main reason is that our method depends on changing a single most significant attribute. This change leads to slightly modifying the adversarial face image but with a high effect in misleading the different models. We can observe that SAA-StarGAN-PS-M and SAA-StarGAN-CS-M have a lower structural similarity than the single attribute, because the adversarial image has a greater change.

On the other hand, SemanticAdv has a high structural similarity value, because it changes one attribute but is random. The random selection method randomly applies two attributes; therefore, it is lower than the SemanticAdv method in structural similarity. In contrast, Sticker and Face Mask methods on FaceNet or ArcFace model show the highest absolute error. These methods mainly depend on pixel level and grid level, which add large perturbations to the face image, covering approximately 20–30% area of the face image. We can see that any adversarial face images generated by FGSM, BIM, PGD, MI-FGSM, DIM, and TIM are expected to be perceptible to human eyes. Generally, SAA-StarGAN-PS and SAA-StarGAN-CS for a single attribute are most similar to the original images and have the lowest error. Therefore, it is recommended to use our proposed method to craft the adversarial face images.

6 Conclusion

This work proposed a new attack method termed SAA-StarGAN for crafting semantic adversarial face images in both white-box and black-box settings. It prioritizes predicting vital facial attributes for each input, manipulating a few vital attributes to make the model promote the trivial features in the face images. In a white-box setting, SAA-StarGAN first predicts the significant attributes of each input image via the cosine similarity or probability score. Then, we use a StarGAN model to alter the most important attributes. For a black-box setting, we make a loop to change the ordered significant attributes until the output is an adversary. As the result, the proposed SAA-StarGAN exhibited significantly higher transferability than the state-of-the-art face attack methods due to the step of important attribute prediction for each input image. Besides, our method produced high-quality images evaluated with SSIM. Yet, we required further metrics to assess the naturalness of face images compared to others. In future work, we will generate adversarial face images to attack FR in the physical domain.

Acknowledgements The authors thank the National Natural Science Foundation of China (U22B2017, 62076105) for supporting this work.

Author Contributions Methodology: YMK; formal analysis and investigation: YMK and KH; writing—original draft preparation: YMK; Writing—review and editing: YMK, YX, and KH; supervision and funding: KH.

Funding This work is supported by National Natural Science Foundation of China (Grant Nos. U22B2017 and 62076105).

Data Availability Data will be made available on request.

Declarations

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics Approval and Consent to Participate Not applicable.

Consent for Publication No individual images were used. All images are from the CelebA dataset, which is a public dataset.

References

1. Wang, M., Deng, W.: Deep face recognition: a survey. Neurocomputing **429**, 215–244 (2021)
2. Hou, J., Wang, Z., Li, Y.: A network for makeup face verification based upon deep learning. In: 2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC), pp. 123–127. Beijing, China (2020)
3. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: 2nd International Conference on Learning Representations (ICLR), Banff, Canada (2014)
4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations (ICLR), San Diego, USA (2015)
5. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. San Jose, USA (2017)
6. Rasheed, B., Khattak, A.M., Khan, A., Protasov, S.I., Ahmad, M.: Boosting adversarial training using robust selective data augmentation. Int. J. Comput. Intell. Syst. **16**(1), 89 (2023)
7. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: 5th International Conference on Learning Representations (ICLR), Toulon, France (2017)
8. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: 2018 IEEE/CVF Confer-

ence on Computer Vision and Pattern Recognition (CVPR), pp. 9185–9193. Salt Lake, USA (2018)
9. Wang, X., He, K.: Enhancing the transferability of adversarial attacks through variance tuning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1924–1933. Nashville, USA(2021)
10. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L.: Improving transferability of adversarial examples with input diversity. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, USA, pp. 2730–2739 (2019)
11. Dong, Y., Pang, T., Su, H., Zhu, J.: Evading defenses to transferable adversarial examples by translation-invariant attacks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4312–4321. Long Beach, USA (2019)
12. Wang, X., He, X., Wang, J., He, K.: Admix: enhancing the transferability of adversarial attacks. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16138–16147. Montreal, Canada (2021)
13. Wang, X., Huang, C., Cheng, H.: Improving transferability of adversarial examples with powerful affine-shear transformation attack. Comput. Stand. Interfaces **84**, 103693 (2023)
14. Duan, Y., Zou, J., Zhou, X., Zhang, W., Zhang, J., Pan, Z.: Enhancing transferability of adversarial examples via rotation-invariant attacks. IET Comput. Vis. **16**(1), 1–11 (2022)
15. Song, Y., Shu, R., Kushman, N., Ermon, S.: Constructing unrestricted adversarial examples with generative models. In: International Conference on Neural Information Processing Systems (NIPS), pp. 8322–8333 (2018)
16. Wang, X., He, K., Song, C., Wang, L., Hopcroft, J.E.: AT-GAN: an adversarial generator model for non-constrained adversarial examples. CoRR arXiv:1904.07793 (2019)
17. Dong, Y., Su, H., Wu, B., Li, Z., Liu, W., Zhang, T., Zhu, J.: Efficient decision-based black-box adversarial attacks on face recognition. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7714–7722. Long Beach, USA (2019)
18. Qiu, H., Xiao, C., Yang, L., Yan, X., Lee, H., Li, B.: Semanticadv: generating adversarial examples via attribute-conditioned image editing. In: 6th European Conference on Computer Vision (ECCV), pp. 19–37. Glasgow, UK (2020)
19. Kakizaki, K., Yoshida, K.: Adversarial image translation: unrestricted adversarial examples in face recognition systems. In: Proceedings of 34th AAAI Conference on Artificial Intelligence, pp. 6–13. New York, USA (2020)
20. Deb, D., Zhang, J., Jain, A.K.: Advfaces: adversarial face synthesis. In: 2020 IEEE International Joint Conference on Biometrics (IJCB), pp. 1–10. Houston, USA (2020)
21. Hu, S., Liu, X., Zhang, Y., Li, M., Zhang, L.Y., Jin, H., Wu, L.: Protecting facial privacy: generating adversarial identity masks via style-robust makeup transfer. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15014–15023. New Orleans, USA (2022)
22. Rozsa, A., Günther, M., Rudd, E.M., Boult, T.E.: Are facial attributes adversarially robust? In: 23rd International Conference on Pattern Recognition (ICPR), pp. 3121–3127. Cancun, Mexico (2016)
23. Rozsa, A., Günther, M., Rudd, E.M., Boult, T.E.: Facial attributes: accuracy and adversarial robustness. Pattern Recognit. Lett. **124**, 100–108 (2019)
24. Mirjalili, V., Ross, A.: Soft biometric privacy: retaining biometric utility of face images while perturbing gender. In: 2017 IEEE International Joint Conference on Biometrics (IJCB), pp. 564–573. Denver, USA (2017)
25. Choi, Y., Choi, M., Kim, M., Ha, J., Kim, S., Choo, J.: Stargan: unified generative adversarial networks for multi-domain image-to-image translation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), USA, pp. 8789–8797. Salt Lake (2018)
26. Liu, M., Ding, Y., Xia, M., Liu, X., Ding, E., Zuo, W., Wen, S.: STGAN: a unified selective transfer network for arbitrary image attribute editing. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3673–3682. Long Beach, USA (2019)
27. He, Z., Zuo, W., Kan, M., Shan, S., Chen, X.: AttGAN: facial attribute editing by only changing what you want. IEEE Trans. Image Process. **28**(11), 5464–5478 (2019)
28. Joshi, A., Mukherjee, A., Sarkar, S., Hegde, C.: Semantic adversarial attacks: parametric transformations that fool deep classifiers. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4772–4782. Seoul, Korea (South) (2019)
29. Xiao, Z., Gao, X., Fu, C., Dong, Y., Gao, W., Zhang, X., Zhou, J., Zhu, J.: Improving transferability of adversarial patches on face recognition with generative models. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11845–11854. Nashville, USA (2021)
30. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations (ICLR), Vancouver, Canada (2018)
31. Lin, J., Song, C., He, K., Wang, L., Hopcroft, J.E.: Nesterov accelerated gradient and scale invariance for adversarial attacks. In: 8th International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia (2020)
32. Wang, X., Lin, J., Hu, H., Wang, J., He, K.: Boosting adversarial transferability through enhanced momentum. In: 32nd British Machine Vision Conference (BMVC), Online, p. 272 (2021)
33. Byun, J., Cho, S., Kwon, M., Kim, H., Kim, C.: Improving the transferability of targeted adversarial examples through object-based diverse input. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15223–15232. New Orleans, USA (2022)
34. Li, C., Yao, W., Wang, H., Jiang, T.: Adaptive momentum variance for attention-guided sparse adversarial attacks. Pattern Recognit. **133**, 108979 (2023)
35. Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: reliable attacks against black-box machine learning models. In: 6th International Conference on Learning Representations (ICLR), Vancouver, Canada (2018)
36. Ilyas, A., Engstrom, L., Athalye, A., Lin, J.: Black-box adversarial attacks with limited queries and information. In: Proceedings of the 35th International Conference on Machine Learning (ICML), vol. 80, pp. 2142–2151. Stockholm, Sweden (2018)
37. Liu, J., Jin, H., Xu, G., Lin, M., Wu, T., Nour, M.K.A., Alenezi, F., Alhudhaif, A., Polat, K.: Aliasing black box adversarial attack with joint self-attention distribution and confidence probability. Expert Syst. Appl. **214**, 119110 (2023)
38. Zhu, Z., Lu, Y., Chiang, C.: Generating adversarial examples by makeup attacks on face recognition. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 2516–2520. Taipei, Taiwan (2019)
39. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: A general framework for adversarial examples with objectives. ACM Trans. Priv. Secur. **22**(3), 16–11630 (2019)
40. Komkov, S., Petiushko, A.: Advhat: real-world adversarial attack on arcface face id system. In: 25th International Conference on Pattern Recognition (ICPR), pp. 819–826. Milan, Italy (2021)
41. Tong, L., Chen, Z., Ni, J., Cheng, W., Song, D., Chen, H., Vorobeychik, Y.: FACESEC: a fine-grained robustness evaluation framework for face recognition systems. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13249–13258. Nashville, USA (2021)

42. Guetta, N., Shabtai, A., Singh, I., Momiyama, S., Elovici, Y.: Dodging attack using carefully crafted natural makeup. CoRR arXiv:2109.06467 (2021)
43. Ryu, G., Park, H., Choi, D.: Adversarial attacks by attaching noise markers on the face against deep face recognition. J. Inf. Secur. Appl. **60**, 102874 (2021)
44. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815–823. Boston, USA (2015)
45. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: additive angular margin loss for deep face recognition. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4690–4699. Long Beach, USA (2019)
46. Dai, Y., Gieseke, F., Oehmcke, S., Wu, Y., Barnard, K.: Attentional feature fusion. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 3559–3568. Waikoloa, USA (2021)
47. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. Las Vegas, USA (2016)
48. Ak, K.E., Kassim, A.A., Lim, J., Tham, J.Y.: Attribute manipulation generative adversarial networks for fashion images. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10540–10549. Seoul, Korea (South) (2019)
49. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: IEEE International Conference on Computer Vision (ICCV), pp. 3730–3738 (2015)
50. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: deep hypersphere embedding for face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6738–6746 (2017)
51. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: large margin cosine loss for deep face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)), pp. 5265–5274 (2018)
52. Ranjan, R., Castillo, C.D., Chellappa, R.: L2-constrained softmax loss for discriminative face verification. CoRR arXiv:1703.09507 (2017)
53. Chen, S., Liu, Y., Gao, X., Han, Z.: Mobilefacenets: efficient cnns for accurate real-time face verification on mobile devices. In: Biometric Recognition - 13th Chinese Conference (CCBR), vol. 10996, pp. 428–438. Urumqi, China (2018)
54. Ma, N., Zhang, X., Zheng, H., Sun, J.: ShuffleNet V2: practical guidelines for efficient CNN architecture design. In: European Conference on Computer Vision (ECCV), pp. 122–138 (2018)
55. Horé, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: International Conference on Pattern Recognition (ICPR), pp. 2366–2369 (2010)
56. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
57. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. **23**(10), 1499–1503 (2016)
58. Li, J., Ji, S., Du, T., Li, B., Wang, T.: TextBugger: generating adversarial text against real-world applications. In: 26th Annual Network and Distributed System Security Symposium (NDSS), San Diego, USA (2019)
59. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 618–626. Venice, Italy (2017)