FaceGuard: A Self-Supervised Defense Against Adversarial Face Images

Debayan Deb, Xiaoming Liu, Anil K. Jain
Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, 48824
{debdebay, liuxm, jain}@cse.msu.edu

2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG) | 979-8-3503-4544-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/FG57933.2023.10042617

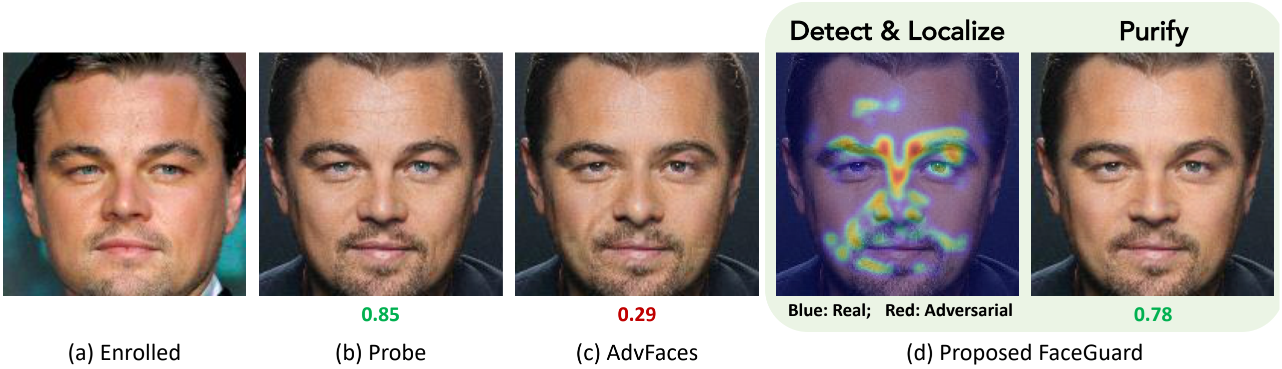


Fig. 1: Leonardo DiCaprio's real face photo (a) enrolled in the gallery and (b) his probe image[1]; (c) Adversarial probe synthesized by a state-of-the-art (SOTA) adversarial face generator, AdvFaces [13]; (d) Proposed adversarial defense framework, namely *FaceGuard* takes (c) as input, detects adversarial images, localizes perturbed regions, and outputs a "purified" face devoid of adversarial perturbations. A SOTA face recognition system, ArcFace, fails to match Leonardo's adversarial face (c) to (a), however, the purified face can successfully match to (a). Cosine similarity scores ($\in [-1, 1]$) obtained via ArcFace [14] are shown below the images. A score above **0.36** (threshold @ 0.1% False Accept Rate) indicates that two faces are of the same subject.

***Abstract*— Prevailing defense schemes against adversarial face images tend to overfit to the perturbations in the training set and fail to generalize to unseen adversarial attacks. We propose a new self-supervised adversarial defense framework, namely FaceGuard, that can automatically detect, localize, and purify a wide variety of adversarial faces without utilizing pre-computed adversarial training samples. During training, FaceGuard automatically synthesizes challenging and diverse adversarial attacks, enabling a classifier to learn to distinguish them from real faces. Concurrently, a purifier attempts to remove the adversarial perturbations in the image space. Experimental results on LFW, Celeb-A, and FFHQ datasets show that FaceGuard can achieve 99.81%, 98.73%, and 99.35% detection accuracies, respectively, on six unseen adversarial attack types. In addition, the proposed method can enhance the face recognition performance of ArcFace from 34.27% TAR @ 0.1% FAR under no defense to 77.46% TAR @ 0.1% FAR. Code, pre-trained models and dataset will be publicly available.**

I. INTRODUCTION

With the advent of deep learning and the availability of large datasets, Automated Face Recognition (AFR) systems have achieved impressive recognition rates [21]. The accuracy, usability, and touchless acquisition of state-of-the-art (SOTA) AFR systems have led to their ubiquitous adoption in a plethora of domains. However, this has also inadvertently sparked a community of attackers that dedicate their time and effort to manipulate faces either physically [31] or digitally [12], in order to evade AFR systems [11]. AFR systems have been shown to be vulnerable to adversarial attacks resulting from perturbing an input probe [10], [13], [15], [40]. Even when the amount of perturbation is imperceptible to the human eye, such adversarial attacks can degrade the face recognition performance of SOTA AFR systems [13]. With the growing dissemination of "fake news" and "deepfakes" [4], research groups and social media platforms alike are pushing towards generalizable defense against continuously evolving adversarial attacks.

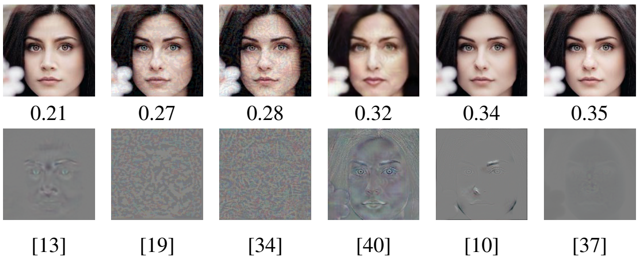


Fig. 2: *(Top Row)* Adversarial faces synthesized via 6 adversarial attacks used in our study. *(Bottom Row)* Corresponding adversarial perturbations (gray indicates no change from the input). Notice the diversity in the perturbations. ArcFace scores between adversarial image and the unaltered gallery image (not shown here) are given below each image. A score above **0.36** indicates that two faces are of the same subject. Zoom in for details.

A considerable amount of research has focused on synthesizing adversarial attacks [10], [13], [19], [27], [34], [37], [40]. Obfuscation attempts (faces are perturbed such that they cannot be identified as the attacker) are more effective [13], computationally efficient to synthesize [19], [34], and widely adopted [44] compared to impersonation attacks (perturbed faces can automatically match to a target subject). Similar

[1] https://bit.ly/2IkfSxk

979-8-3503-4544-5/23/$31.00 ©2023 IEEE

to prior defense efforts [16], [35], this paper focuses on defending against obfuscation attacks (see Fig. 1). Given an input probe image, $\mathbf{x}$, an adversarial generator has two requirements under the obfuscation scenario: (1) synthesize an adversarial face image, $\mathbf{x}_{adv} = \mathbf{x} + \delta$, such that SOTA AFR systems fail to match $\mathbf{x}_{adv}$ and $\mathbf{x}$, and (2) limit the magnitude of perturbation $||\delta||_p$ such that $\mathbf{x}_{adv}$ appears very similar to $\mathbf{x}$ to humans.

A number of approaches have been proposed to defend against adversarial attacks. Their major shortcoming is *generalizability* to unseen adversarial attacks. Adversarial face perturbations may vary significantly (see Fig. 2). For instance, gradient-based attacks, such as FGSM [34] and PGD [34], perturb every pixel in the face image, whereas, AdvFaces [13] and SemanticAdv [40] perturb only the salient facial regions, *e.g.*, eyes, nose, and mouth. On the other hand, GFLM [10] performs geometric warping to the face. Since the exact type of adversarial perturbation may not be known a priori, a defense system trained on a subset of adversarial attack types may have degraded performance on other unseen attacks.

To the best of our knowledge, we take the first step towards a complete defense against adversarial faces by integrating an adversarial face generator, a detector, and a purifier into a unified framework, namely *FaceGuard* (see Fig. 3). Robustness to unseen adversarial attacks is imparted via a stochastic generator that outputs diverse perturbations evading an AFR system, while a detector continuously learns to distinguish them from real faces. Concurrently, a purifier removes the adversarial perturbations from the synthesized image.

This work makes the following contributions:

- A new self-supervised framework, namely *FaceGuard*, for defending against adversarial face images. *FaceGuard* combines benefits of adversarial training, detection, and purification into a unified defense mechanism trained in an end-to-end manner.
- With the proposed diversity loss, a generator is regularized to produce stochastic and challenging adversarial faces. We show that the diversity in output perturbations is sufficient for improving *FaceGuard*'s robustness to unseen attacks compared to utilizing pre-computed training samples from known attacks.
- Synthesized adversarial faces aid the detector to learn a tight decision boundary around real faces. *FaceGuard*'s detector achieves SOTA detection accuracies of 99.81%, 98.73%, and 99.35% on 6 unseen attacks on LFW [22], Celeb-A [33], and FFHQ [25].
- As the generator trains, a purifier concurrently removes perturbations from the synthesized adversarial faces. With the proposed bonafide loss, the detector also guides purifier's training to ensure purified images are devoid of adversarial perturbations. At 0.1% False Accept Rate, *FaceGuard*'s purifier enhances the True Accept Rate of ArcFace [14] from 34.27% under no defense to 77.46%.

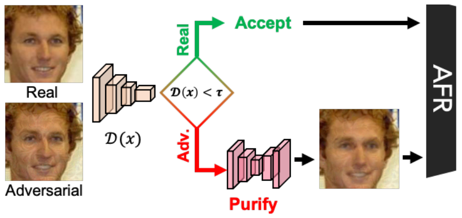


Fig. 3: *FaceGuard* employs a detector ($\mathcal{D}$) to compute an adversarial score. Scores below detection threshold (τ) passes the input to AFR, and high value invokes a purifier and sends the purified face to the AFR system.

II. Related Work

Defense Strategies. In literature, a common defense strategy, namely *robustness* is to re-train the classifier we wish to defend with adversarial examples [19], [24], [26], [34]. However, adversarial training has been shown to degrade classification accuracy on real (non-adversarial) images [46].

In order to prevent degradation in AFR performance, a large number of adversarial defense mechanisms are deployed as a pre-processing step, namely *detection*, which involves training a binary classifier to distinguish between real and adversarial examples [2], [16], [17], [35], [53]. The attacks considered in these studies [5], [7], [8] were initially proposed in the object recognition domain and they often fail to detect the attacks in a feature-extraction network setting, as in face recognition. Therefore, prevailing detectors against adversarial faces are demonstrated to be effective only in a highly constrained setting where the number of subjects is limited and fixed during training and testing [2], [16], [35].

Another pre-processing strategy, namely *purification*, involves automatically removing adversarial perturbations in the input image prior to passing them to a face matcher [36], [38], [42], [45]. However, without a dedicated adversarial detector, these defenses may end up "purifying" a real face image, resulting in high false reject rates.

In Tab. I, we summarize a few studies on adversarial defenses that are used as baselines in our work.

Adversarial Attacks:. Numerous adversarial attacks have been proposed in literature [9], [19], [34], [39], [48]. We evaluate *FaceGuard* on six unseen adversarial attacks that have high success rates in evading ArcFace [14]: FGSM [19], PGD [34], DeepFool [37], AdvFaces [13], GFLM [10], and SemanticAdv [40] (see Tab. II).

III. Limitations of State-of-the-Art Defenses

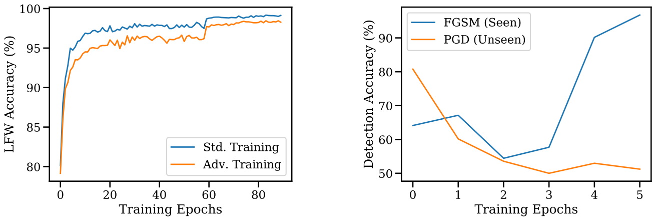
Robustness. Adversarial training is regarded as one of the most effective defense method [19], [30], [34] on small datasets including MNIST and CIFAR10. Adversarial training is formulated as [19], [34]:

$$\min_{\theta} \underset{(x,y)\sim\mathcal{P}_{data}}{\mathbb{E}} \left[\max_{\delta\in\Delta} \ell\left(f_\theta\left(x+\delta\right), y\right)\right], \qquad (1)$$

where $(x, y) \sim \mathcal{P}_{data}$ is the (image, label) joint distribution of data, $f_\theta(x)$ is the network parameterized by θ, and $\ell(f_\theta(x), y)$ is the loss function. Here, the network, f_θ is made robust by training with an adversarial noise (δ) that

	Study	Method	Dataset	Attacks	Self-Sup.
Robustness	Adv. Training [26] (2017)	Train with adv.	ImageNet	FGSM [19]	×
	RobGAN [30] (2019)	Train with generated adv.	CIFAR10, ImageNet	PGD [34]	×
	Feat. Denoising [49] (2019)	Custom network arch.	ImageNet	PGD [34]	×
	RoBal [47] (2021)	Long-Tailed Distribution	CIFAR10	PGD [34]	×
Detection	Gong *et al.* [17] (2017)	Binary CNN	MNIST, CIFAR10	FGSM [19]	×
	ODIN [28] (2018)	Out-of-distribution Detection	CIFAR10, ImageNet	OOD samples	×
	Massoli *et al.* [35] (2020)	MLP/LSTM on AFR Filters	VGGFace2 [6]	FGSM [19], C&W [9]	×
	Agarwal *et al.* [1] (2022)	Binary CNN	ImageNet	FGSM [19], DeepFool [37], PGD [34]	×
Purification	MagNet [36] (2017)	AE Purifier	MNIST, CIFAR10	FGSM [19], DeepFool [37], C&W [9]	×
	DefenseGAN [42] (2018)	GAN	MNIST, CIFAR10	FGSM [19], C&W [9]	×
	NRP [38] (2020)	AE Purifier	ImageNet	FGSM [19]	✓
	Yoon *et al.* [52] (2021)	Score-based Generative Models	MNIST, CIFAR10	PGD [34]	×
	FaceGuard (this study)	Adv. Generator + Detector + Purifier	LFW [22], Celeb-A [33], FFHQ [25]	FGSM [19], PGD [34], DeepFool [37], AdvFaces [13], GFLM [10], Semantic [40]	✓

TABLE I: Related work in adversarial defenses. Unlike majority of prior work, *FaceGuard* is self-supervised where no pre-computed adversarial examples are required for training.



(a) Adversarial Training [26]

(b) Detection [17]

Fig. 4: (a) Adversarial training degrades AFR performance of FaceNet matcher [43] on real faces in LFW dataset compared to standard training. (b) A binary classifier trained to distinguish between real faces and FGSM [19] attacks fails to detect unseen attack type, namely PGD [34].

maximally increases the classification loss. In other words, adversarial training involves training with the *strongest* adversarial attack.

The generalization of adversarial training has been in question [24], [30], [41], [46]. It was shown that adversarial training can significantly reduce classification accuracy on real examples [46]. In the context of face recognition, we illustrate this by training two face matchers on CASIA-WebFace: (i) FaceNet [43] trained via the standard training process, and (ii) FaceNet [43] by adversarial training (FGSM[2]). We then compute face recognition performance across training iterations on a separate testing dataset, LFW [22]. Fig. 4a shows that adversarial training drops the accuracy from 99.13% → 98.27%. We gain the following insight: adversarial training may degrade AFR performance on real faces.

Detection. Detection-based approaches employ a preprocessing step to "detect" whether an input face is real or adversarial [2], [17], [35]. A common approach is to utilize a binary classifier, $\mathcal{D}$, that maps a face image, $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$ to $\{0, 1\}$, where 0 indicates a real and 1 an adversarial face. We train a binary classifier to distinguish between real and FGSM attack samples in CASIA-WebFace [51]. In Fig. 4b, we evaluate its detection accuracy on FGSM and PGD samples in LFW [22].

We find that prevailing detection-based defense schemes overfit to the specific adversarial attacks utilized for training.

IV. FACEGUARD

Our defense aims to achieve robustness without sacrificing AFR performance on real face images. We posit that an adversarial defense trained alongside an adversarial generator in a *self-supervised* manner may improve robustness to unseen attacks. The main intuitions behind our defense mechanism are as follows:

- Since adversarial training may degrade AFR performance, we opt to obtain a robust adversarial *detector* and *purifier* to detect and purify adversarial attacks.
- Given that prevailing detection-based methods tend to overfit to known adversarial perturbations (see Supp.), a detector and purifier trained on *diverse* synthesized adversarial perturbations may be more robust to unseen attacks.
- Sufficient diversity in synthesized perturbations can guide the detector to learn a tighter boundary around real faces. In this case, the detector itself can serve as a powerful supervision for the purifier.
- Lastly, pixels involved in the purification process may serve to indicate adversarial regions in the input face.

A. *Adversarial Generator*

The generalizability of an adversarial detector and purifier relies on the quality of the synthesized adversarial face images output by *FaceGuard*'s adversarial generator. We propose an adversarial generator that continuously learns to synthesize challenging and diverse adversarial face images.

The generator, denoted as $\mathcal{G}$, takes an input real face image, $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$, and outputs an adversarial perturbation $\mathcal{G}(\mathbf{x}, \mathbf{z})$, where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ is a random latent vector. Inspired by prevailing adversarial attack generators [9], [13], [19], [34], [37], we treat the output perturbation $\mathcal{G}(\mathbf{x}, \mathbf{z})$ as an additive *perturbation mask*. The final adversarial face image, $\mathbf{x}_{adv}$, is given by $\mathbf{x}_{adv} = \mathbf{x} + \mathcal{G}(\mathbf{x}, \mathbf{z})$.

In an effort to impart generalizability to the detector and purifier, we emphasize the following requirements of $\mathcal{G}$:

- **Adversarial:** Perturbation $\mathcal{G}(\mathbf{x}, \mathbf{z})$, needs to be such that an AFR system cannot identify the adversarial face image $\mathbf{x}_{adv}$ as the same person as the input probe $\mathbf{x}$.

[2] With max perturbation hyperparameter as $\epsilon = 8/256$.

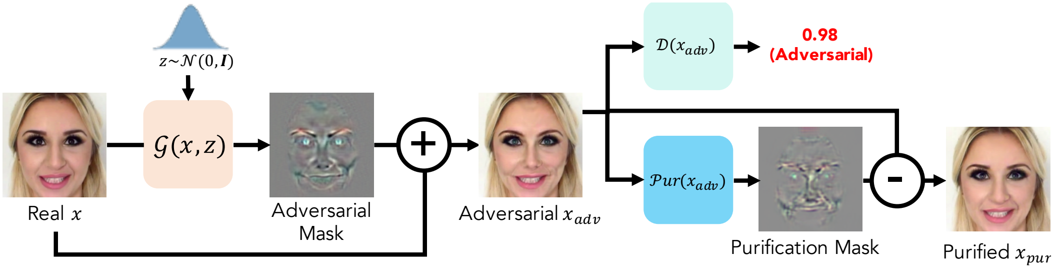


Fig. 5: Overview of training the proposed *FaceGuard* in a self-supervised manner. An *adversarial generator*, $\mathcal{G}$, continuously learns to synthesize challenging and diverse perturbations that evade a face matcher. At the same time, a *detector*, $\mathcal{D}$, learns to distinguish between the synthesized adversarial faces and real face images. Perturbations residing in the synthesized adversarial faces are removed via a *purifier*, $\mathcal{P}ur$.

- **Visually Realistic:** Perturbation $\mathcal{G}(\mathbf{x},\mathbf{z})$ should also be minimal such that $\mathbf{x}_{adv}$ appears as a legitimate face image of the subject in the input probe $\mathbf{x}$.
- **Stochastic:** For an input $\mathbf{x}$, we require diverse adversarial perturbations, $\mathcal{G}(\mathbf{x},\mathbf{z})$, for different latents $\mathbf{z}$.

For satisfying all of the above requirements, we propose multiple loss functions to train the generator.

Obfuscation Loss To ensure $\mathcal{G}(\mathbf{x},\mathbf{z})$ is indeed *adversarial*, we incorporate a white-box AFR system, $\mathcal{F}$, to supervise the generator. Given an input face, $\mathbf{x}$, the generator aims to output an adversarial face, $\mathbf{x}_{adv}=\mathbf{x}+\mathcal{G}(\mathbf{x},\mathbf{z})$ such that the face representations, $\mathcal{F}(\mathbf{x})$ and $\mathcal{F}(\mathbf{x}_{adv})$, do not match. In other words, the goal is to minimize the cosine similarity between the two face representations[3]:

$$\mathcal{L}_{obf}=\mathbb{E}_{\mathbf{x}}\left[\frac{\mathcal{F}(\mathbf{x})\cdot\mathcal{F}(\mathbf{x}_{adv})}{||\mathcal{F}(\mathbf{x})||\;||\mathcal{F}(\mathbf{x}_{adv})||}\right]. \tag{2}$$

Perturbation Loss With the identity loss alone, the generator may output perturbations with large magnitudes which will (a) be trivial for the detector to reject and (b) violate the visual realism requirement of x_{adv}. Therefore, we restrict the perturbations to be within $[-\epsilon,\epsilon]$ via a hinge loss:

$$\mathcal{L}_{pt}=\mathbb{E}_{\mathbf{x}}\left[\max\left(\epsilon,||\mathcal{G}(\mathbf{x},\mathbf{z})||_2\right)\right]. \tag{3}$$

Diversity Loss The above two losses jointly ensure that at each step, our generator learns to output challenging adversarial attacks. However, these attacks are deterministic; for an input image, we will obtain the same adversarial image. This may again lead to an inferior detector that overfits to a few deterministic perturbations seen during training. Motivated by studies on preventing mode collapse in GANs [50], we propose maximizing a diversity loss to promote stochastic perturbations per training iteration, i:

$$\mathcal{L}_{div}=-\frac{1}{N_{ite}}\sum_{i=1}^{N_{ite}}\frac{\left|\left|\mathcal{G}(\mathbf{x},\mathbf{z}_1)^{(i)}-\mathcal{G}(\mathbf{x},\mathbf{z}_2)^{(i)}\right|\right|_1}{||\mathbf{z}_1-\mathbf{z}_2||_1}, \tag{4}$$

where N_{ite} is the number of training iterations, $\mathcal{G}(\mathbf{x},\mathbf{z})^{(i)}$ is the perturbation output at iteration i, and $(\mathbf{z}_1,\mathbf{z}_2)$ are two i.i.d. samples from $\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})$. The diversity loss ensures that for two random latent vectors, $\mathbf{z}_1$ and $\mathbf{z}_2$, we will obtain two different perturbations $\mathcal{G}(\mathbf{x},\mathbf{z}_1)^{(i)}$ and $\mathcal{G}(\mathbf{x},\mathbf{z}_2)^{(i)}$.

GAN Loss Akin to prior work on GANs [18], [23], we introduce a discriminator to encourage perceptual realism of the adversarial images. The discriminator, Dsc, aims to distinguish between probes, $\mathbf{x}$, and synthesized faces $\mathbf{x}_{adv}$ via a GAN loss:

$$\mathcal{L}_{GAN}=\mathbb{E}_{\mathbf{x}}\left[\log Dsc(\mathbf{x})\right]+\mathbb{E}_{\mathbf{x}}[\log(1-Dsc(\mathbf{x}_{adv}))]. \tag{5}$$

B. Adversarial Detector

Similar to prevailing adversarial detectors, the proposed detector also learns a decision boundary between real and adversarial images [2], [17], [35]. A key difference, however, is that instead of utilizing pre-computed adversarial images from known attacks (*e.g.* FGSM and PGD) for training, the proposed detector learns to distinguish between real images and the *synthesized* set of diverse adversarial attacks output by the proposed adversarial generator in a self-supervised manner. This leads to the following advantage: *our proposed framework does not require a large collection of pre-computed adversarial face images for training.*

We utilize a binary CNN for distinguishing between real input probes, $\mathbf{x}$, and synthesized adversarial samples, $\mathbf{x}_{adv}$. The detector is trained with the Binary Cross-Entropy loss:

$$\mathcal{L}_{BCE}=\mathbb{E}_{\mathbf{x}}\left[\log\mathcal{D}(\mathbf{x})\right]+\mathbb{E}_{\mathbf{x}}\left[\log\left(1-\mathcal{D}(\mathbf{x}_{adv})\right)\right]. \tag{6}$$

C. Adversarial Purifier

The objective of the adversarial purifier is to recover the real face image $\mathbf{x}$ given an adversarial face $\mathbf{x}_{adv}$. We aim to automatically remove the adversarial perturbations by training a neural network $\mathcal{P}ur$, referred as an adversarial purifier.

The adversarial purification process can be viewed as an inverted procedure of adversarial image synthesis. Contrary to the obfuscation loss in the adversarial generator, we require that the purified image, $\mathbf{x}_{pur}$, successfully matches to the subject in the input probe $\mathbf{x}$. Note that this can be achieved via a *feature recovery loss*, which is the opposite to the obfuscation loss, *i.e.*, $\mathcal{L}_{fr}=-\mathcal{L}_{obf}$.

Note that an adversarial face image, $\mathbf{x}_{adv}=\mathbf{x}+\delta$, is metrically close to the real image, $\mathbf{x}$, in the input space. If we can estimate δ, then we can retrieve the real face image. Here, the perturbations can be predicted by a neural network, $\mathcal{P}ur$. In other words, retrieving the purified image, $\mathbf{x}_{pur}$ involves: (1) subtracting the perturbations from the

[3] For brevity, we denote $\mathbb{E}_{\mathbf{x}}\equiv\mathbb{E}_{\mathbf{x}\in\mathcal{P}_{data}}$.

Attacks	TAR (%) @ 0.1% FAR(↓)	SSIM(↑)
FGSM [19]	26.23	0.83 ± 0.24
PGD [34]	04.91	0.89 ± 0.12
DeepFool [37]	36.18	0.91 ± 0.09
AdvFaces [13]	00.17	0.89 ± 0.02
GFLM [10]	68.03	0.55 ± 0.14
SemanticAdv [40]	70.05	0.71 ± 0.21
No Attack	99.82	1.00 ± 0.00

TABLE II: Face recognition performance of ArcFace [14] under adversarial attack and average structural similarities (SSIM) between probe and adversarial images for obfuscation attacks on $485K$ genuine pairs in LFW [22].

adversarial image, $\mathbf{x}_{pur} = \mathbf{x}_{adv} - \mathcal{P}ur(\mathbf{x}_{adv})$ and (2) ensuring that the *purification mask*, $\mathcal{P}ur(\mathbf{x}_{adv})$, is small so that we do not alter the content of the face image by a large magnitude. Therefore, we propose a hybrid perceptual loss that (1) ensures $\mathbf{x}_{pur}$ is as close as possible to the real image, $\mathbf{x}$ via a ℓ_1 reconstruction loss and (2) a loss that minimizes the amount of alteration, $\mathcal{P}ur(\mathbf{x}_{adv})$:

$$\mathcal{L}_{perc} = \mathbb{E}_{\mathbf{x}} \left|\left|\mathbf{x}_{pur} - \mathbf{x}\right|\right|_1 + \left|\left|\mathcal{P}ur(\mathbf{x}_{adv})\right|\right|_2 . \quad (7)$$

Finally, we also incorporate our detector to guide the training of our purifier. Note that, due to the diversity in synthesized adversarial faces, the proposed detector learns a tight decision boundary around real faces. This can serve as a strong self-supervisory signal to the purifier for ensuring that the purified images belong to the real face distribution. Therefore, we also incorporate the detector as a discriminator for the purifier via the proposed bonafide loss:

$$\mathcal{L}_{bf} = \mathbb{E}_{\mathbf{x}} \left[\log \mathcal{D}(\mathbf{x}_{pur})\right] . \quad (8)$$

D. Training Framework

We train the entire *FaceGuard* framework in Fig. 5 in an end-to-end manner with the following objectives:

$$\min_{\mathcal{G}} \mathcal{L}_{\mathcal{G}} = \mathcal{L}_{GAN} + \lambda_{obf} \cdot \mathcal{L}_{obf} + \lambda_{pt} \cdot \mathcal{L}_{pt} - \lambda_{div} \cdot \mathcal{L}_{div},$$
$$\min_{\mathcal{D}} \mathcal{L}_{\mathcal{D}} = \mathcal{L}_{BCE},$$
$$\min_{\mathcal{P}ur} \mathcal{L}_{\mathcal{P}ur} = \lambda_{fr} \cdot \mathcal{L}_{fr} + \lambda_{perc} \cdot \mathcal{L}_{perc} + \lambda_{bf} \cdot \mathcal{L}_{bf}.$$

At each training iteration, the generator attempts to fool the discriminator by synthesizing visually realistic adversarial faces while the discriminator learns to distinguish between real and synthesized images. In the same iteration, an external critic network, namely detector $\mathcal{D}$, learns a decision boundary between real and synthesized adversarial samples. Concurrently, the purifier $\mathcal{P}ur$ learns to invert the adversarial synthesis process. Note: the generator is designed to specifically *fool* the discriminator but not necessarily the detector. We will show in our experiments that this crucial step prevents the detector from predicting $\mathcal{D}(\mathbf{x}) = 0.5$ for all $\mathbf{x}$ (see Tab. V).

V. EXPERIMENTAL RESULTS

A. Experimental Settings

Datasets. We train *FaceGuard* on real face images in CASIA-WebFace [51] dataset and then evaluate on real and adversarial faces synthesized for LFW [22], Celeb-A [33] and FFHQ [25] datasets. CASIA-WebFace [51] comprises of $494,414$ face images from $10,575$[4] different subjects. LFW [22] contains $13,233$ face images of $5,749$ subjects. Since we evaluate defenses under obfuscation attacks, we consider subjects with at least two face images[5]. After this filtering, $9,164$ face images of $1,680$ subjects in LFW remain. Results on CelebA and FFHQ are in Supp.

Implementation. The adversarial generator and purifier employ a convolutional encoder-decoder. The latent variable $\mathbf{z}$, a 128-dimensional feature vector, is fed as input to the generator through spatial padding and concatenation. The adversarial detector, a 4-layer binary CNN, is trained jointly with the generator and purifier. Empirically, we set $\lambda_{obf} = \lambda_{fr} = 10.0$, $\lambda_{pt} = \lambda_{perc} = 1.0$, $\lambda_{div} = 1.0$, $\lambda_{bf} = 1.0$ and $\epsilon = 3.0$. More details are provided in Supp.

Face Recognition Systems. Recall that the proposed defense utilizes a face matcher, $\mathcal{F}$, for training process of the generator. However, the deployed AFR system may not be known to the defense system a priori. Therefore, unlike prevailing defense mechanisms [2], [16], [35], we evaluate the effectiveness of the proposed defense on an AFR system *different* from $\mathcal{F}$. We highlight the effectiveness of our proposed defense: *FaceGuard is trained on FaceNet [43], while the adversarial attack test set is designed to evade ArcFace [14].* Obfuscation attempts perturb real probes into adversarial ones. Ideally, deployed AFR systems (say, ArcFace), should be able to match a genuine pair comprised of an adversarial probe and a real enrolled face of the same subject. Therefore, regardless of real or adversarial probe, we assume that genuine pairs should *always* match as ground truth. Tab. II provides AFR performance of ArcFace under 6 SOTA adversarial attacks for $484,514$ genuine pairs in LFW. It appears that some attacks, *e.g.*, AdvFaces [13], are effective in both low TAR and high SSIM, while some are less capable in both metrics.

B. Comparison with State-of-the-Art Defenses

SOTA Detectors. Our baselines include 9 SOTA detectors proposed both for general objects [17], [28], [29] and adversarial faces [2], [3], [16], [20], [35]. The detectors are trained on real and adversarial faces images synthesized via six adversarial generators for CASIA-WebFace [51]. Unlike all the baselines, *FaceGuard*'s detector does not utilize any pre-computed adversarial attack for training. We compute the classification accuracy for all methods on a dataset comprising of $9,164$ real images and $9,164$ adversarial face images per attack type in LFW.

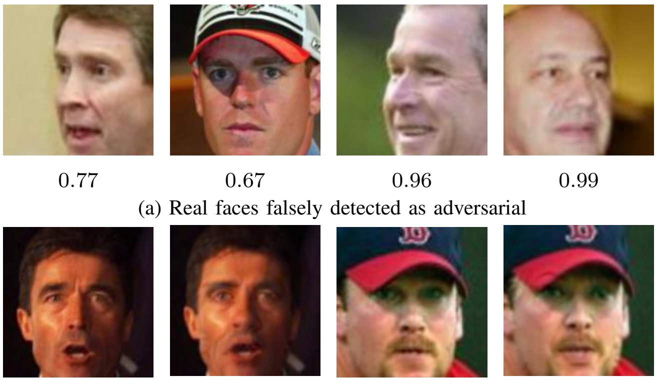
In Tab. III, we find that compared to the baselines, *FaceGuard* achieves the highest detection accuracy. Even when the 6 adversarial attack types are encountered in training, a binary CNN [17], still falls short compared to *FaceGuard*. This is likely because *FaceGuard* is trained on a diverse set of adversarial faces from the proposed generator. While the

[4] We removed 84 subjects in CASIA-WebFace that overlap with LFW.

[5] Obfuscation attempts only affect genuine pairs (two face images pertaining to the same subject).

	Detection Accuracy (%)	Year	FGSM [19]	PGD [34]	DpFl. [37]	AdvF. [13]	GFLM [10]	Smnt. [40]	Mean ± Std.
General	Gong *et al.* [17]	2017	98.94	97.91	95.87	92.69	**99.92**	**99.92**	97.54 ± 02.82
	ODIN [28]	2018	83.12	84.39	71.74	50.01	87.25	85.68	77.03 ± 14.34
	Steganalysis [29]	2019	88.76	89.34	75.97	54.30	58.99	78.62	74.33 ± 14.77
Face	UAP-D [2]	2018	61.32	74.33	56.78	51.11	65.33	76.78	64.28 ± 09.97
	SmartBox [16]	2018	58.79	62.53	51.32	54.87	50.97	62.14	56.77 ± 05.16
	Goswami *et al.* [20]	2019	84.56	91.32	89.75	76.51	52.97	81.12	79.37 ± 14.04
	Massoli *et al.* [35] (MLP)	2020	63.58	76.28	81.78	88.38	51.97	52.98	69.16 ± 15.29
	Massoli *et al.* [35] (LSTM)	2020	71.53	76.43	88.32	75.43	53.76	55.22	70.11 ± 13.35
	Agarwal *et al.* [3]	2020	94.44	95.38	91.19	74.32	51.68	87.03	87.03 ± 16.86
	Proposed FaceGuard	2022	**99.85**	**99.85**	**99.85**	**99.84**	99.61	99.85	**99.81 ± 00.10**

TABLE III: Detection accuracy of SOTA adversarial face detectors in classifying six adversarial attacks synthesized for the LFW dataset [22]. Detection threshold is set as 0.5 for all methods. All baseline methods require training on pre-computed adversarial attacks on CASIA-WebFace [51]. On the other hand, the proposed *FaceGuard* is self-guided and generates adversarial attacks on the fly. Hence, it can be regarded as a *black-box* defense system.



(a) Real faces falsely detected as adversarial

(b) Adversarial faces falsely detected as real

Fig. 6: Examples where the proposed *FaceGuard* fails to correctly detect (a) real faces and (b) adversarial faces. Detection scores $\in [0, 1]$ are given below each image, where 0 indicates real and 1 indicates adversarial face.

binary CNN has a small drop compared to FaceGuard in the seen attacks ($99.81\% \rightarrow 97.54\%$), it drops significantly on unseen adversarial attacks in testing (see Supp.).

Compared to hand-crafted features, such as PCA+SVM in UAP-D [2] and entropy detection in SmartBox [16], *FaceGuard* achieves superior detection results. Some baselines utilize AFR features for identifying adversarial inputs [20], [35]. We find that intermediate AFR features primarily represent the identity of the input face and do not appear to contain highly discriminative information for detecting adversarial faces.

Despite the robustness, *FaceGuard* misclassifies 28 out of $9,164$ real images in LFW [22] and falsely predicts 46 out of $54,984$ adversarial faces as real. From the latter, 44 are warped faces via GFLM [10] and the remaining two are synthesized via AdvFaces [13]. We find that *FaceGuard* tends to misclassify real faces under extreme poses and adversarial faces that are occluded (*e.g.*, hats) (see Fig. 6).

Comparison with Adversarial Training & Purifiers. We also compare with prevailing defenses designing robust face matchers [24], [26], [30] and purifiers [36], [38], [42]. We conduct a verification experiment by considering all possible genuine pairs (two faces belonging to the same subject) in LFW [22]. For one probe in a genuine pair, we craft six different adversarial probes (one per attack type). In total, there are $484,514$ real pairs and $\sim 3M$ adversarial pairs. For a fixed match threshold[6], we compute the True Accept Rate (TAR) of successfully matching two images in a real or adversarial pair in Tab. IV. In other words, TAR is defined here as the ratio of genuine pairs above the match threshold.

ArcFace without any adversarial defense system achieves 34.27% TAR at 0.1% FAR under attack. Adversarial training [24], [26], [30] inhibits the feature space of ArcFace, resulting in worse performance on both real and adversarial pairs. On the other hand, purification methods [36], [38], [42] can better retain face features in real pairs but their performance under attack is still undesirable.

Instead, the proposed *FaceGuard* defense system first detects whether an input face image is real or adversarial. If input faces are adversarial, they are further purified. From Tab. IV, we find that our defense system significantly outperforms SOTA baselines in protecting ArcFace [14] against attacks. Specifically, *FaceGuard*'s purifier enhances ArcFace's average TAR at 0.1% FAR under all six attacks (see Tab. II) from $34.27\% \rightarrow 77.46\%$. In addition, *FaceGuard* also maintains similar face recognition performance on real faces (TAR on real pairs drop from $99.82\% \rightarrow 99.81\%$).

Therefore, our proposed defense system ensures that benign users will not be incorrectly rejected while malicious attempts to evade the AFR system will be curbed.

C. Analysis of Our Approach

Quality of the Adversarial Generator. In Tab. V, we see that without the proposed adversarial generator ("Without $\mathcal{G}$"), *i.e.*, a detector trained on the six known attack types, suffers from high standard deviation. Instead, training a detector with a deterministic $\mathcal{G}$ ("Without $\mathcal{L}_{div}$"), leads to better generalization across attack types, since the detector still encounters variations in synthesized images as the generator learns to better generate adversarial faces. However, such a detector is still prone to overfitting to a few deterministic perturbations output by $\mathcal{G}$. Finally, *FaceGuard* with the diversity loss introduces diverse perturbations within and across training iterations (see Fig. 7).

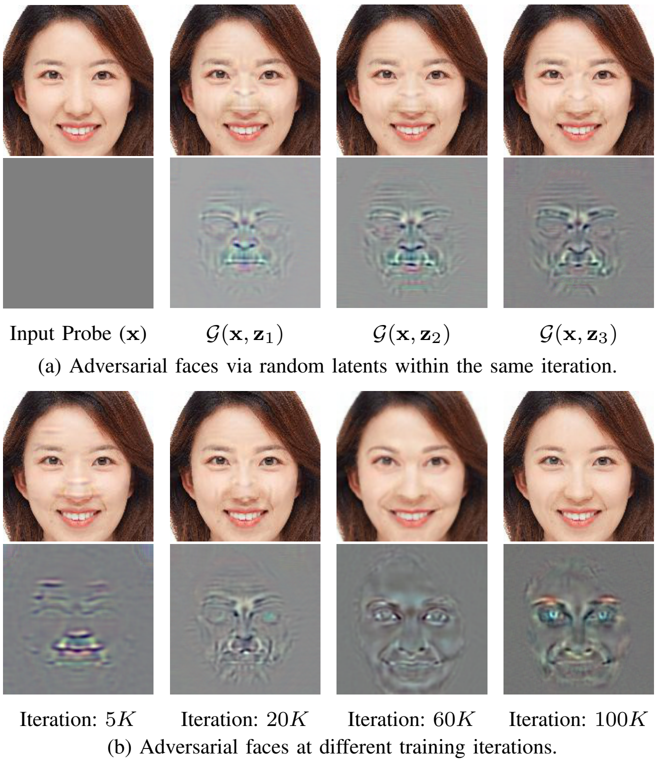
[6] We compute the threshold at 0.1% FAR on all possible image pairs in LFW, *e.g.*, threshold @ 0.1% FAR for ArcFace is set at 0.36.

Defenses	Year	Strategy	Real $485K$ pairs	Attacks $3M$ pairs
No-Defense	–	-	99.82	34.27
Adv. Training [26]	2017	Robustness	96.42	11.23
Rob-GAN [30]	2019	Robustness	91.35	13.89
Feat. Denoising [49]	2019	Robustness	87.61	17.97
L2L [24]	2019	Robustness	96.89	16.76
MagNet [36]	2017	Purification	94.47	38.32
DefenseGAN [42]	2018	Purification	96.78	39.21
Feat. Distillation [32]	2019	Purification	94.64	41.77
NRP [38]	2020	Purification	97.54	61.44
A-VAE [54]	2020	Purification	93.71	51.99
Proposed FaceGuard	2022	Purification	**99.81**	**77.46**

TABLE IV: AFR performance (TAR (%) @ 0.1% FAR) of ArcFace under no defense and when ArcFace is defended via proposed *FaceGuard* and prevailing robustness or purification techniques.

	Model	AdvFaces [13]	Mean ± Std.
Gen. $\mathcal{G}$	Without $\mathcal{G}$	91.72	97.12 ± 04.54
	Without $\mathcal{L}_{div}$	95.42	98.23 ± 01.33
	With $\mathcal{G}$ and $\mathcal{L}_{div}$	**99.84**	$\mathbf{99.81 \pm 00.10}$
Det. $\mathcal{D}$	$\mathcal{D}$ as Discriminator	50.00	75.25 ± 21.19
	$\mathcal{D}$ via Pre-Computed $\mathcal{G}$	52.01	69.37 ± 19.91
	$\mathcal{D}$ as Online Detector	**99.84**	$\mathbf{99.81 \pm 00.10}$

TABLE V: Ablating training schemes of the generator $\mathcal{G}$ and detector $\mathcal{D}$. All models are trained on CASIA-WebFace [51]. *(Col. 3)* We compute the detection accuracy in classifying real faces in LFW [22] and the most challenging adversarial attack in Tab. II, AdvFaces [13]. *(Col. 4)* The avg. and std. dev. of detection accuracy across all 6 adversarial attacks.


Input Probe ($\mathbf{x}$)
$\mathcal{G}(\mathbf{x}, \mathbf{z}_1)$
$\mathcal{G}(\mathbf{x}, \mathbf{z}_2)$
$\mathcal{G}(\mathbf{x}, \mathbf{z}_3)$

(a) Adversarial faces via random latents within the same iteration.

Iteration: $5K$
Iteration: $20K$
Iteration: $60K$
Iteration: $100K$

(b) Adversarial faces at different training iterations.

Fig. 7: Adversarial faces synthesized by *FaceGuard* during training. Note the diversity in perturbations (a) within and (b) across iterations.

Quality of the Adversarial Detector. The discriminator's task is similar to the detector; determine whether an input image is real or fake/adversarial. The key difference is that the generator is enforced to fool the discriminator, but not the detector. If we replace the discriminator with an adversarial detector, the generator continuously attempts to fool the detector by synthesizing images that are as close as possible to the real image distribution. By design, such a detector should converge to $Disc(\mathbf{x}) = 0.5$ for all $\mathbf{x}$ (real or adversarial). As we expect, in Tab. V, we cannot rely on predictions made by such a detector ("$\mathcal{D}$ as Discriminator"). We try another variant: we first train the generator $\mathcal{G}$ and then train a detector to distinguish between real and pre-computed attacks via $\mathcal{G}$ ("$\mathcal{D}$ via Pre-Computed $\mathcal{G}$"). As we expect, the proposed methodology of training the detector in an online fashion by utilizing the synthesized adversarial samples output by $\mathcal{G}$ at any given iteration leads to a significantly robust detector ("$\mathcal{D}$ as Online Detector"). This can likely be attributed to the fact that a detector trained on-line encounters a much larger variation as the generator trains alongside. "$\mathcal{D}$ via Pre-Computed $\mathcal{G}$" is exposed only to within-iteration variations (from random latent sampling), however, '$\mathcal{D}$ as Online Detector" encounters variations *both* within and across training iterations (see Fig. 7).

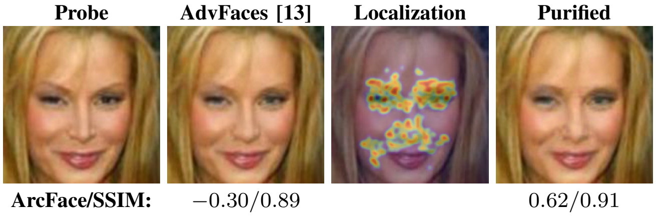
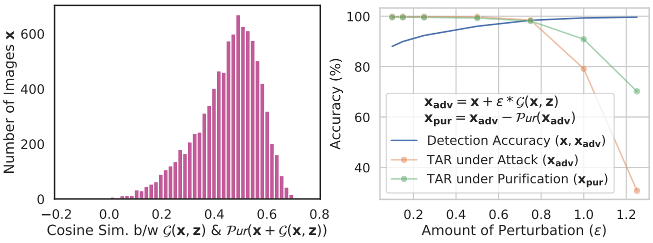


Fig. 8: *FaceGuard* successfully purifies the adversarial image (red regions indicate adversarial perturbations localized by our purification mask).



(a) (b)

Fig. 9: (a) *FaceGuard*'s purification is correlated with its adversarial synthesis process. (b) Trade-off between detection and purification with respect to perturbation magnitudes. With minimal perturbation, detection is challenging while purifier maintains AFR performance. Excessive perturbations lead to easier detection with greater challenge in purification.

Quality of the Adversarial Purifier. Recall that we enforced the purified image to be close to the real face via a reconstruction loss. Thus, the purification and perturbation masks should be similar. In Fig. 9a, we shows that the two masks are indeed correlated by plotting the Cosine similarity distribution ($\in [-1, 1]$) between $\mathcal{G}(\mathbf{x}, \mathbf{z})$ and $\mathcal{P}ur(\mathbf{x} + \mathcal{G}(\mathbf{x}, \mathbf{z}))$ for all $9,164$ images in LFW.

Therefore, pixels in $\mathbf{x}_{adv}$ involved in the purification process should correspond to those that cause the image to be adversarial in the first place. Fig. 8 highlights that perturbed regions can be automatically localized via constructing a heatmap out of $\mathcal{P}ur(\mathbf{x}_{adv})$. In Fig. 9b, we investigate the change in AFR performance (TAR (%) @ 0.1% FAR) of ArcFace under attack (synthesized adversarial faces via $\mathcal{G}(\mathbf{x}, \mathbf{z})$) when the amount of perturbation is varied. We find that (a) minimal perturbation is harder to detect but the purifier incurs minimal damage to the AFR, while, (b) excessive perturbations are easier to detect but increases the challenge in purification.

VI. CONCLUSIONS

With the introduction of sophisticated adversarial attacks on AFR systems, defense needs to be robust and generalizable. Without utilizing any pre-computed training samples from known adversarial attacks, the proposed *FaceGuard* achieved state-of-the-art detection performance against 6 different adversarial attacks. *FaceGuard*'s purifier also enhanced ArcFace's recognition performance under adversarial attacks. We are exploring whether an attention mechanism can further improve adversarial purification.

VII. ACKNOWLEDGMENTS

This work was partially supported by Facebook AI.

REFERENCES

[1] A. Agarwal, N. Ratha, M. Vatsa, and R. Singh. Exploring robustness connection between artificial and natural adversarial examples. In *CVPRW*, 2022.

[2] A. Agarwal, R. Singh, M. Vatsa, and N. Ratha. Are image-agnostic universal adversarial perturbations for face recognition difficult to detect? In *BTAS*, pages 1–7, 2018.

[3] A. Agarwal, R. Singh, M. Vatsa, and N. K. Ratha. Image transformation based defense against adversarial perturbation on deep learning models. *IEEE TDSC*, 2020.

[4] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li. Protecting world leaders against deep fakes. In *CVPR*, 2019.

[5] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security. *ICML*, 2018.

[6] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *FG*, 2018.

[7] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *AISEC*, 2017.

[8] N. Carlini and D. Wagner. Magnet and "efficient defenses against adversarial attacks" are not robust to adversarial examples. *arXiv:1711.08478*, 2017.

[9] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security & Privacy*, 2017.

[10] A. Dabouei, S. Soleymani, J. Dawson, and N. Nasrabadi. Fast geometrically-perturbed adversarial faces. In *WACV*, 2019.

[11] Daily Mail. Police arrest passenger who boarded plane in Hong Kong as an old man in flat cap and arrived in Canada a young Asian refugee. http://dailym.ai/2UBEcxO, 2011.

[12] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. Jain. On the detection of digital face manipulation. In *CVPR*, 2020.

[13] D. Deb, J. Zhang, and A. K. Jain. Advfaces: Adversarial face synthesis. In *IJCB*. IEEE, 2020.

[14] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[15] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, and J. Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*, pages 7714–7722, 2019.

[16] A. Goel, A. Singh, A. Agarwal, M. Vatsa, and R. Singh. Smartbox: Benchmarking adversarial detection and mitigation algorithms for face recognition. In *BTAS*, 2018.

[17] Z. Gong, W. Wang, and W.-S. Ku. Adversarial and clean data are not twins. *arXiv:1704.04960*, 2017.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv:1412.6572*, 2014.

[20] G. Goswami, A. Agarwal, N. Ratha, R. Singh, and M. Vatsa. Detecting and mitigating adversarial perturbations for robust face recognition. *IJCV*, 2019.

[21] P. Grother, M. Ngan, and K. Hanaoka. Ongoing face recognition vendor test (frvt). *NIST Interagency Report*, 2018.

[22] 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. Technical Report 07-49, UMass, 2007.

[23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[24] Y. Jang, T. Zhao, S. Hong, and H. Lee. Adversarial defense via learning to generate diverse attacks. In *ICCV*, 2019.

[25] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[26] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *ICLR*, 2017.

[27] D. Li, W. Wang, H. Fan, and J. Dong. Exploring adversarial fake images on face manifold. In *CVPR*, 2021.

[28] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *ICLR*, 2018.

[29] J. Liu, W. Zhang, Y. Zhang, D. Hou, Y. Liu, H. Zha, and N. Yu. Detection based defense against adversarial examples from the steganalysis point of view. In *CVPR*, 2019.

[30] X. Liu and C.-J. Hsieh. Rob-gan: Generator, discriminator, and adversarial attacker. In *CVPR*, 2019.

[31] Y. Liu, J. Stehouwer, and X. Liu. On disentangling spoof traces for generic face anti-spoofing. In *ECCV*, 2020.

[32] Z. Liu, Q. Liu, T. Liu, N. Xu, X. Lin, Y. Wang, and W. Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *CVPR*, 2019.

[33] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, December 2015.

[34] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv:1706.06083*, 2017.

[35] F. V. Massoli, F. Carrara, G. Amato, and F. Falchi. Detection of face recognition adversarial attacks. *CVIU*, page 103103, 2020.

[36] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In *ACM CCS*, 2017.

[37] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, pages 2574–2582, 2016.

[38] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli. A self-supervised approach for adversarial robustness. In *CVPR*, 2020.

[39] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *IEEE Symposium on Security & Privacy*, 2016.

[40] H. Qiu, C. Xiao, L. Yang, X. Yan, H. Lee, and B. Li. Semanticadv: Generating adversarial examples via attribute-conditional image editing. *arXiv:1906.07927*, 2019.

[41] A. Raghunathan, S. M. Xie, F. Yang, J. C. Duchi, and P. Liang. Adversarial training can hurt generalization. *arXiv:1906.06032*, 2019.

[42] P. Samangouei, M. Kabkab, and R. Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *ICLR*, 2018.

[43] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[44] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao. Fawkes: protecting privacy against unauthorized deep learning models. In *USENIX*, pages 1589–1604, 2020.

[45] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *ICLR*, 2017.

[46] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao. Is robustness the cost of accuracy? In *ECCV*, 2018.

[47] T. Wu, Z. Liu, Q. Huang, Y. Wang, and D. Lin. Adversarial robustness under long-tailed distribution. In *CVPR*, 2021.

[48] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks. *IJCAI*, 2018.

[49] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019.

[50] D. Yang, S. Hong, Y. Jang, T. Zhao, and H. Lee. Diversity-sensitive conditional generative adversarial networks. *ICLR*, 2019.

[51] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014.

[52] J. Yoon, S. J. Hwang, and J. Lee. Adversarial purification with score-based generative models. In *ICML*. PMLR, 2021.

[53] V. Zantedeschi, M.-I. Nicolae, and A. Rawat. Efficient defenses against adversarial attacks. In *ACM AISEC*, 2017.

[54] J. Zhou, C. Liang, and J. Chen. Manifold projection for adversarial defense on face recognition. In *ECCV*. Springer, 2020.