Privacy-preserving Adversarial Facial Features

Zhibo Wang[†,ʖ,*], He Wang[†], Shuaifan Jin[†], Wenwen Zhang[‡], Jiahui Hu[†], Yan Wang[‡]
Peng Sun[♯], Wei Yuan[♮], Kaixin Liu[♮], Kui Ren[†]
[†]School of Cyber Science and Technology, Zhejiang University, P. R. China
[ʖ]ZJU-Hangzhou Global Scientific and Technological Innovation Center [‡]Alibaba Group, P. R. China
[♯]College of Computer Science and Electronic Engineering, Hunan University, P. R. China
[♮]School of Cyber Science and Engineering, Wuhan University, P. R. China
{zhibowang, wanghe_71, shuaifanjin}@zju.edu.cn, karida.zww@alibaba-inc.com, jiahuihu@zju.edu.cn
wy84378@alibaba-inc.com, psun@hnu.edu.cn, {wyuan, kxliu777}@whu.edu.cn, kuiren@zju.edu.cn

Abstract

Face recognition service providers protect face privacy by extracting compact and discriminative facial features (representations) from images, and storing the facial features for real-time recognition. However, such features can still be exploited to recover the appearance of the original face by building a reconstruction network. Although several privacy-preserving methods have been proposed, the enhancement of face privacy protection is at the expense of accuracy degradation. In this paper, we propose an adversarial features-based face privacy protection (AdvFace) approach to generate privacy-preserving adversarial features, which can disrupt the mapping from adversarial features to facial images to defend against reconstruction attacks. To this end, we design a shadow model which simulates the attackers' behavior to capture the mapping function from facial features to images and generate adversarial latent noise to disrupt the mapping. The adversarial features rather than the original features are stored in the server's database to prevent leaked features from exposing facial information. Moreover, the AdvFace requires no changes to the face recognition network and can be implemented as a privacy-enhancing plugin in deployed face recognition systems. Extensive experimental results demonstrate that AdvFace outperforms the state-of-the-art face privacy-preserving methods in defending against reconstruction attacks while maintaining face recognition accuracy.

1. Introduction

Face recognition is a way of identifying an individual's identity using their face, which has been widely used in many security-sensitive applications. Undoubtedly, biometric facial images are private and discriminative information to each person that should be protected. Recently, much attention has been paid to privacy protection, such as the General Data Protection Regulation, making the preservation of face privacy increasingly important. In order to avoid direct leakage of facial images, mainstream face recognition systems usually adopt a client-server mode that extracts features from facial images with a feature extractor on the client side and stores the facial features rather than facial images on the server side for future online identification. As facial features suppress the visual information of faces, face privacy protection can be realized to some extent.

However, recent studies showed that it is possible to reconstruct original images from facial features, which is called reconstruction attack, including optimization-based [9, 29] and learning-based reconstruction attacks [7, 13, 23, 37]. The former gradually adjusts the pixels of the input image to make the output of the feature extractor as close as possible to a particular feature until the facial image (the input image) corresponding to this feature is reconstructed [9, 29]. The latter trains a feature-image decoder with a deconvolutional neural network (D-CNN) to reconstruct images directly from facial features [7, 13, 23, 37]. These studies imply that existing face recognition systems suffer from severe privacy threats once the features in their database were leaked. Therefore, it is essential to provide approaches to prevent facial features from being reconstructed.

Several approaches have been proposed to protect face privacy. [1, 10, 18, 22] transform the features into the encrypted space and perform face recognition based on the cryptographic primitives and security protocols, which however bear prohibitive computation and communication costs for face recognition systems. [3, 24] utilize differential privacy to protect face privacy by perturbing features with noises, which however suffers from a significant accuracy drop in face recognition. [19, 34] proposed adversarial

*Zhibo Wang is the corresponding author.

training-based methods that retrain the main task network (e.g., gender classification from facial images) using adversarial training between the reconstruction network and the main task network to generate the privacy-preserving features directly. However, [19] demonstrated that facial features learned from adversarial training significantly compromise accuracy when dealing with face recognition tasks. Recently, several frequency domain-based methods [15,25] were proposed, which transform raw images into the frequency domain and remove features' critical channels used for visualization to protect face privacy. However, [15] struggles with the trade-off between accuracy and privacy protection and our experimental results demonstrate that [25] actually cannot resist powerful reconstruction attacks. In addition, both the adversarial learning-based and the frequency domain-based methods require retraining the face recognition network, which is not applicable to deployed face recognition systems.

In this paper, we aim to propose a novel approach to generate privacy-preserving facial features which are able to thwart reconstruction attacks as well as maintain satisfactory recognition accuracy. Undoubtedly, it is non-trivial to realize this objective. The first challenge is *how to defend against reconstruction attacks under the black-box setting*. An attacker may utilize different reconstruction networks, which are unknown to the face recognition systems in advance, to reconstruct images from facial features. How to enable the generated facial features to defend against such unknown and different reconstruction attacks is very challenging. The second challenge is *how to disrupt the visual information embedded in facial features while keeping the recognition accuracy*. Since visual information is somewhat critical to face recognition, disrupting visual information may incur a reduction in recognition accuracy. The last challenge is *how to generate privacy-preserving features without changing the face recognition network*. Once a face recognition network is deployed, it would be expensive to retrain the network and redeploy it to millions of clients. Therefore, a plug-in module is more welcomed for the deployed face recognition systems.

To address the above challenges, we propose an *adversarial features-based face privacy protection (AdvFace)* method, which generates the privacy-preserving adversarial features against reconstruction attacks. The intuition of AdvFace is to disrupt the mapping from features to facial images by obfuscating features with adversarial latent noise to maximize the difference between the original images and the reconstructed images from the features. To this end, we train a shadow model to simulate the behavior of the reconstruction attacks to obtain the reconstruction loss which denotes the quality of the reconstructed images. Thereafter, we maximize the reconstruction loss to generate the adversarial features by iteratively adding the adversarial latent noise to features along the direction of the gradient (loss w.r.t. the targeted feature). Moreover, to ensure face recognition accuracy, the magnitude of adversarial latent noise would be constrained during the optimization.

Our main contributions are summarized as follows:

- We propose a novel facial privacy-preserving method (namely AdvFace), which can generate privacy-preserving adversarial features against unknown reconstruction attacks while maintaining face recognition accuracy. Moreover, AdvFace requires no changes to the deployed face recognition model and thus can be integrated as a plug-in privacy-enhancing module into face recognition systems.
- We unveil the rationale of the reconstruction attack and build a shadow model to simulate the behavior of the reconstruction attacks and generate adversarial features, which can disrupt the mapping from features to facial images by maximizing the reconstruction loss of the shadow model.
- Extensive experimental results demonstrate that our proposed AdvFace outperforms the state-of-the-art facial privacy-preserving methods in terms of superior privacy protection performance while only incurring negligible face recognition accuracy loss. Moreover, the transferability of AdvFace is validated. That is, it can effectively resist different reconstruction networks.

2. Related work

This section provides an overview of related works on face reconstruction attacks and face privacy protection.

2.1. Face Reconstruction Attacks

In earlier works, Mignon et al. [26] used the radial basis function regression to reconstruct faces from their features. Mohanty et al. [27] used the inverse of the affine transformation model, which simulates the face recognition system, to reconstruct facial images. However, However, regression and affine transformation-based methods are no longer applicable when facial features are extracted by complex deep neural networks. Fredrikson et al. [9] performed the face reconstruction by solving an optimization problem, which aims to generate the reconstructed images that minimize the distance between the targeted features and features from the reconstructed images. Similarly, Razzhigaev et al. [29] followed the same optimization objective and transformed the problem into the linear space of 2D Gaussian functions for higher efficiency. However, these optimization-based reconstruction attacks incur large computation costs even for reconstructing only one facial image. Hence, some recent works [4, 7, 13, 23, 37] used the reconstruction network trained by a large number of (image, feature) pairs to map the features back to the facial images. In this paper, given

their powerful attacking performance with moderate attacking costs, we choose the reconstruction network-based attacks as our defense target.

2.2. Face Privacy Protection

Several face privacy protection methods have recently been proposed, which can be divided into four categories. The encryption-based methods, such as homomorphic encryption [10], matrix encryption [18], functional encryption [1], and randomized CNN with user-specific keys [22], encrypted facial images or features, and then performed face recognition in the encrypted space to protect face privacy. However, the high computational overhead of exchanging and processing data is not suitable for face recognition networks that already consume a lot of computational resources. In [3, 24], differential privacy-based methods were incorporated to add carefully crafted noises to features or images to prevent leaking information that can distinguish faces. However, these methods suffer from a significant accuracy drop in face recognition tasks. The adversarial training-based methods [19, 34] strengthened face privacy by directly generating a facial feature that could resist reconstruction attacks through the adversarial training between the reconstruction network and the feature extractor. However, researchers in [19] revealed that the accuracy of face recognition decreased significantly (from 99.97% to 30.38%) when dealing with the driver identity recognition task. Researchers in [15, 25] proposed face privacy protection methods based on the frequency domain segmentation, which transforms the facial images into the frequency features and removes parts of the features that are important for image reconstruction but secondary to face recognition to thwart the reconstruction attacks while ensuring the accuracy of face recognition. However, [15] struggled with the tradeoff between accuracy and privacy protection and our experimental results demonstrate that [25] cannot resist powerful reconstruction attacks. In addition, the frequency domain-based methods require retraining and redeploying the face recognition network leading to significantly increased costs. In summary, the aforementioned methods are either incapable of defending against reconstruction attacks while maintaining recognition accuracy or incurring large computation overhead or redeployment costs.

3. Preliminary

In this section, we first introduce a typical architecture of face recognition systems and then present a realistic threat model considered in this paper.

3.1. Face Recognition Systems

For face protection, existing face recognition systems usually use the client-server architecture [8, 17]. As shown in Fig. 1, the face recognition network is partitioned and deployed as two sequential modules, i.e., the feature extractor

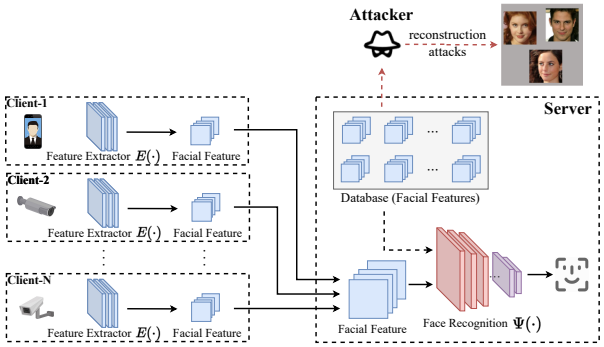

Figure 1. The typical architecture of face recognition systems.

$E(\cdot)$ (the front layers of the network) on the client side and the remaining layers $\Psi(\cdot)$ on the server side. In particular, each client uses $E(\cdot)$ to extract features from facial images and submits them to the server, where $\Psi(\cdot)$ is employed to recognize the identities of the received facial features.

Rather than original facial images, the server stores facial features that do not visually disclose facial information. However, as previously mentioned, once stolen from the server's database, the facial features can still be exploited to reconstruct facial images via the reconstruction attack. Therefore, effective face privacy-preserving methods that can protect the original facial images from being reconstructed from the facial features, are highly desired.

3.2. Threat Model

In this paper, we consider that the server in the face recognition system is trusted. However, there may exist external attackers, who tend to steal the facial features from the server's database and launch reconstruction attacks to obtain clients' original facial images.

Attacker's Knowledge: We consider that the attacker is powerful, and it has the following knowledge:

- Facial features: The attacker can obtain the facial features stored in the server's database.
- The Feature Extractor: The attacker can access the clients' black-box feature extractor of the face recognition model. Note that this can be easily achieved by purchasing a client device from the face recognition service provider.

This powerful attacker also makes it difficult for us to design an effective facial privacy-preserving approach.

Attacker's Strategy: Following existing reconstruction attacks [4, 7, 13, 23, 37], we consider that the attacker tends to reconstruct facial images from facial features by building a reconstruction network, denoted by $R(\cdot)$. The attacker can train $R(\cdot)$ by minimizing the reconstruction loss function $\mathcal{L}_R$, which is defined as the L_1-norm distance between the original and reconstructed images. Formally, we have:

$$\mathcal{L}_R(Z, X) = \sum_{i=1}^{N} \|x_i - R(z_i)\|_1 , \quad (1)$$

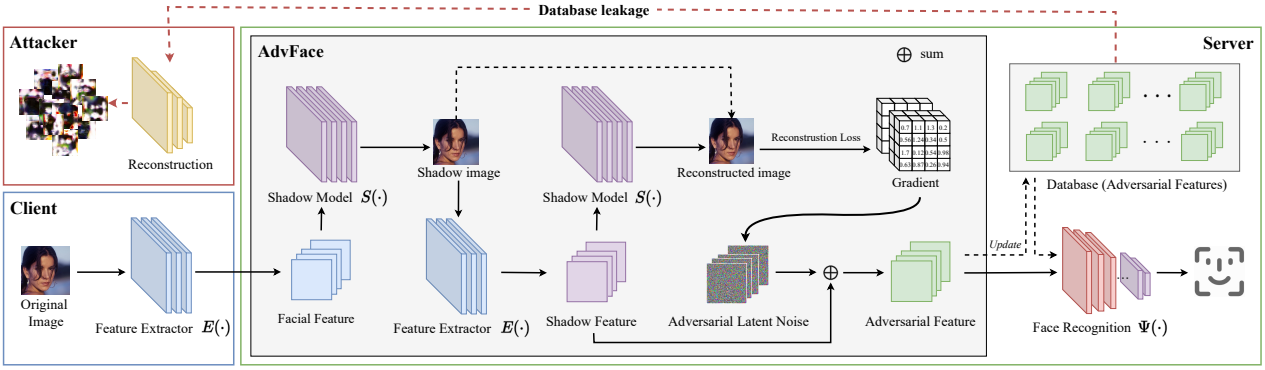

Figure 2. The pipeline of AdvFace in the deployed face recognition system. In the client, the original image is converted to a facial feature which will be uploaded to the server. In the server, the facial feature is reconstructed to a shadow image by the shadow model built by the server, and the shadow feature is extracted and further converted to the reconstructed image. Finally, the generation of the adversarial feature depends on the loss of the shadow image and the reconstructed image. The server stores adversarial features rather than original facial features uploaded from the client against reconstruction attacks.

where x_i denotes a facial image in a public face dataset $X = \{x_1, \ldots, x_N\}$ and $Z = \{z_1, \ldots, z_N\}$ (where $z_i = E(x_i)$) represents the corresponding features extracted by the feature extractor. Here, N represents the total number of facial images in the public dataset.

With the trained reconstruction network $R(\cdot)$, the attacker can easily reconstruct facial images from the facial features via only one step of forward propagation on $R(\cdot)$.

4. Adversarial Features Based Face Protection

In this section, we first provide an overview of AdvFace and then elaborate on the design of the shadow model design and the adversarial features generation. Finally, we present the application scenarios of AdvFace.

4.1. Overview of AdvFace

The key to the success of reconstruction attacks is that they can learn the mapping from features to images by optimizing Eq. (1). Although attackers may use different reconstruction network structures, similar mappings could be learned given the image-feature pairs extracted from the same feature extractor. Therefore, our basic idea is to learn the mapping function by building a shadow model of the reconstruction network and then disrupt the mapping from features to facial images along the opposite direction of training the reconstruction network. To this end, we craft the adversarial features with the adversarial latent noise by solving a constrained optimization problem, which aims to maximize the difference between the original images and the reconstructed images from the adversarial features through the shadow model.

Fig. 2 shows the workflow of AdvFace, which takes the original facial features uploaded from a client as the input and outputs the corresponding adversarial features that will be stored in the server's database for future online face recognition. It is worth noting that the original facial features would not be stored in the database. Thus, when the database is breached, only these adversarial features would be leaked, which could prevent the attacker from reconstructing the facial images.

Specifically, AdvFace consists of the following phases: 1) converting the original facial features uploaded from the client to shadow images with the shadow model; 2) extracting shadow features from shadow images with the feature extractor; 3) reverting the shadow features to the reconstructed images by the same shadow model, and calculating the gradients of $\mathcal{L}_S$ w.r.t. shadow features, which indicates the direction where $\mathcal{L}_R$ increases most quickly; 4) generating adversarial latent noise by gradient ascent; 5) perturbing the shadow features with the adversarial latent noise to generate the adversarial features, which will be stored in the database of the server for future face recognition.

4.2. Shadow Model Building

Despite being exploited by attackers to reconstruct facial images, the visual information contained in facial features is essential for face recognition. This, however, has been largely neglected in existing obfuscation-based privacy protection methods, which usually choose to distort facial features indiscriminately. Consequently, these methods sacrifice face recognition accuracy for privacy protection. To strike a desirable balance between privacy protection and face recognition, we need to craft adversarial features that maximize the reconstruction loss while minimizing the face recognition accuracy loss. Specifically, for a well-trained face recognition network, one feasible way of mitigating recognition accuracy loss is to maximize the reconstruction loss while enforcing a constraint on the magnitude of the disturbance on the features.

In particular, AdvFace aims to find an L_p-norm bounded noise δ to distort features such that the reconstruction loss $\mathcal{L}_R$ is maximized, which is formulated as the following constrained optimization problem:

$$\arg\max_{\delta} \|R(z+\delta)-x\|_1, \quad \text{s.t. } \|\delta\|_p < \xi, \tag{2}$$

where x is an original facial image, z represents the facial features extracted from x, δ denotes the adversarial latent noise, and ξ represents the noise bound. Intuitively, the optimization problem (2) can be easily solved by adding noises along the direction of the gradient of $\mathcal{L}_R$.

However, it is rather challenging or even impossible to directly solve (2) as the reconstruction network employed by the attacker is unknown beforehand. Since different reconstruction networks learn a similar mapping from facial features to images, given the image-feature pairs extracted by the same feature extractor, the solution to problem (2) is actually reconstruction model-agnostic. Hence, we advocate building a powerful shadow model $S(\cdot)$ at the server, which can be any reconstruction network, to learn the corresponding mapping and compute the reconstruction loss (then the corresponding gradients). Similar to (1), we can train the shadow model $S(\cdot)$ on a public face dataset by minimizing the following loss function:

$$\mathcal{L}_S(Z,X) = \sum_{i=1}^{N} \|x_i - S(z_i)\|_1 . \tag{3}$$

However, without accessing the original facial images, the server can not derive the reconstruction loss regarding the stored facial features. To address this issue, as shown in Fig. 2, we first convert the features submitted by the client to shadow images through the shadow model and extract shadow features using the feature extractor. Then, we can compute the reconstruction loss and gradient based on the shadow images and images reconstructed from shadow features. The above process can be formally represented as

$$\tilde{x} = S(z), \tilde{z} = E(\tilde{x}), \tag{4}$$

where $\tilde{x}$ is the shadow image reconstructed by the shadow model from the facial feature z submitted by the client, and $\tilde{z}$ is the shadow feature extracted from the shadow image $\tilde{x}$ using the feature extractor.

Then, we can calculate the gradient $grad(S, \tilde{z}+\delta, \tilde{x})$ of the reconstruction loss of the shadow model w.r.t. the perturbation δ as follows

$$grad(S, \tilde{z}+\delta, \tilde{x}) = \nabla_\delta \|S(\tilde{z}+\delta) - \tilde{x}\|_1, \tag{5}$$

where $\tilde{z}+\delta$ denotes the adversarial feature with δ being initialized to zero. With noise added, the attacker can not recover the shadow image $\tilde{x}$ from the adversarial feature. Since $\tilde{x}$ is highly similar to the original image x, it is also hard for the attacker to reconstruct the original image.

Furthermore, considering that the facial images used for training and those encountered after deploying the face recognition network can be quite different, we update the parameters of the batch normalization (BN) layer in the shadow model. Specifically, unlike a typical BN process, which normalizes inputs during the inference stage using the parameters learned from the training dataset, we compute the mean μ and variance σ based on the transmitted testing feature batches to ensure that the noise added to each feature is more appropriate.

4.3. Adversarial Features Generation

To generate the adversarial features, we inject the adversarial latent noise δ into the shadow features under the guidance of $grad(S, \tilde{z}+\delta, \tilde{x})$ as in (5). Generally, we can use any gradient-based methods to generate adversarial features [6, 11, 21] to break the mapping from the features to the original facial images such that the face recognition system can defend against the reconstruction attack. In this paper, we choose the Project Gradient Descent (PGD) [21], which iteratively adds noises along the gradient direction while restricting the perturbation range in each iteration. Specifically, the generation of adversarial features can be formulated as:

$$\begin{gathered} z_{t+1} = z_t + \alpha \cdot sign(grad(S, z_t, \tilde{x})),\ z_0 = \tilde{z}, \\ \text{s.t. } \|z_{t+1} - z_t\| < \varepsilon, \end{gathered} \tag{6}$$

where α controls the magnitude of noise and ε restricts the noise level added in each iteration. Here, $sign(\cdot)$ is an element-wise function that outputs 1 for positive gradient values, -1 for negative gradient values, and 0 for 0.

Starting from $\tilde{z}$, we update the adversarial feature by iteratively adding noises following (6). With the function $sign(\cdot)$, the noise added in each iteration is not exactly along the direction of the gradient but an approximate one. This helps alleviate the negative influences of some extreme samples, contributing to enhanced robustness of adversarial features.

4.4. Discussions of Application Scenarios

We would like to strengthen that our proposed AdvFace can protect facial privacy without changing the face recognition networks. Thus, AdvFace can be easily integrated into deployed face recognition systems as a plug-in privacy-enhancing module.

Moreover, AdvFace can work in both online and offline modes. Specifically, the server of a face recognition system can employ AdvFace to generate adversarial features from original facial features in real-time, i.e., online mode. However, with noises added, the adversarial features inevitably incur a slight decrease in face recognition accuracy. To tackle this issue, the server can use the adversarial features and labels stored in the database to retrain the face recognition network on the server side, which is the offline mode. The server-side face recognition network can learn sufficient information about the adversarial features through the offline mode, thereby improving face recognition accuracy.

Note that the offline mode does not involve any changes to the adversarial features themselves. Thus, the privacy protection performance will not be compromised. Furthermore, the generated adversarial features can be packaged into privacy-preserving datasets for data sharing and reused for training other face recognition networks.

5. Experimental Evaluation

In this section, we conduct extensive experiments on various datasets and models to evaluate the performance of AdvFace in terms of face recognition accuracy, attack resistance, and transferability.

5.1. Experimental Setup

5.1.1 Datasets

We use the following datasets in our experiments.

- *CASIA-WebFace* [35] contains 490K facial images from more than 10k different individuals.
- *CelebA* [20] contains 202K facial images from more than 10k celebrities.
- *LFW* [14] contains 13K facial images from 5.7K different identities and 6K face pairs (i.e., two facial images) for evaluation.
- *CFP-FP* [32] contains 7K images from 500 identities and 7K face pairs for evaluation.
- *AgeDB-30* [28] contains more than 12K images from 570 identities and 6K face pairs for evaluation.

Following the preprocessing operation adopted in prior works [5, 22], we crop all facial images with the multi-task convolutional neural network (MTCNN) [36], which can detect faces and facial landmarks in images. Besides, for each cropped image, we resize it to 160 $\times$ 160 for a fair comparison. Moreover, each pixel in RGB format (i.e., [0,255]) is normalized to [0,1] before being fed into the neural network.

5.1.2 Models and Implementation Details

Face Recognition Model: We employ the FaceNet [31] with a pre-trained Inception-ResNet-v1 [12] as the backbone for face recognition. We select the first three convolutional layers of the backbone as the feature extractor $E(\cdot)$ deployed on the client-side, while the remaining layers are deployed on the server-side. We fine-tune the classifier of FaceNet for 50 epochs with the backbone frozen using the CASIA-WebFace dataset, and then fine-tune the entire network for 50 epochs. We use the Adam optimizer [16] with a scheduler, where the period and multiplicative factor of learning rate decay are set to 1 and 0.94. The entire model is trained with triplet loss [2] and cross-entropy loss.

Face Reconstruction Model: As summarized in the appendix, three types of reconstruction networks can be employed by the attacker. Specifically, the URec is built based on the U-net [30] architecture and the ResRec is implemented with the ResNet [12] architecture. The TransRec is an exactly mirrored reconstruction network by performing a layer-to-layer reversion. All reconstruction networks are trained on the CelebA dataset.

AdvFace Model: To implement AdvFace, we also train three types of shadow models $S(\cdot)$, including URec, ResRec, and TransRec on the CASIA-WebFace dataset. We adopt the Adam optimizer with a learning rate of 1e-4. In the PGD process, we implement 40 iterations with $\alpha = 0.2$. Unless otherwise specified, the noise bound is $\varepsilon = 0.2$.

5.1.3 Baseline Defense Methods

We compare AdvFace with the following three widely used face privacy protection methods.

Random Perturbation: This method iteratively adds randomly generated noises to the original facial features. To ensure a fair comparison, the total number of iterations is 40 and the noise bound in each iteration is 0.2.

Differential Privacy (DP) [19]: This method adds noises generated from the Laplace mechanism to the original facial features. The privacy budget is set to 1.0 to ensure the same noise bound as our method.

DuetFace [25]: This is the latest face privacy protection method based on frequency domain segmentation. It splits the frequency channels into two parts according to their importance for visualization and mainly use the non-crucial part for transmission and face recognition.

5.1.4 Evaluation Metrics

To evaluate the performance of privacy protection methods against reconstruction attacks, we use SSIM [33], PSNR, and MSE to measure the quality of reconstructed images. Specifically, a larger MSE or a smaller SSIM and PSNR, indicates a lower similarity between the reconstructed image and the original facial image, which implies a stronger defense. For replay attacks which use the reconstructed images to cheat the face recognition system, we use the success rate of replay attacks (SRRA) to measure the performance of the protected features. A lower SRRA indicates that the protected facial feature has a stronger defense ability against the replay attack. Moreover, we characterize the utility of face recognition using the accuracy (ACC) of identifying whether two face features (from face pairs in LFW, CFP-FP, and AgeDB-30) belong to one person.

5.2. Trade-off between Privacy and Utility

We first evaluate the effectiveness of AdvFace in terms of the trade-off between face privacy protection and face recognition accuracy. To this end, we perturb the features with different noise bounds, i.e., ε varies from 0.00 to 0.30 with a step size of 0.05, where $\varepsilon = 0.00$ indicates no protection for the shadow features. We conduct experiments on three datasets, i.e., LFW, AgeDB-30, and CFP-FP, where

Table 1. Performance comparison results among privacy protection methods in terms of SSIM, PSNR, MSE, and SRRA.

	LFW				CFP-FP				AgeDB-30			
Methods	SSIM↓	PSNR↓	MSE↑	SRRA↓	SSIM↓	PSNR↓	MSE↑	SRRA↓	SSIM↓	PSNR↓	MSE↑	SRRA↓
Unprotected	0.93	27.87	0.002	97.40%	0.83	22.89	0.006	89.71%	0.87	23.96	0.005	84.53%
Random	0.90	22.81	0.005	94.73%	0.79	20.73	0.009	87.26%	0.86	21.68	0.007	77.47%
DP	0.90	23.12	0.005	93.97%	0.79	20.89	0.009	84.94%	0.86	21.86	0.007	78.07%
DuetFace	0.85	20.92	0.009	95.17%	0.66	14.38	0.043	70.23%	0.76	14.65	0.040	87.27%
Ours	0.28	6.97	0.206	4.03%	0.23	5.98	0.261	18.43%	0.24	5.85	0.269	16.67%

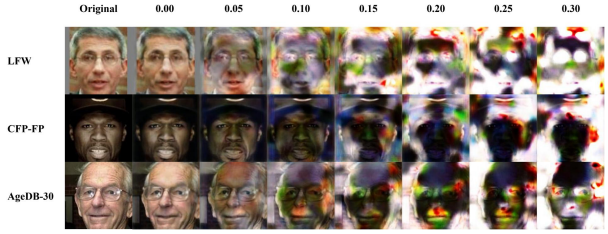

Figure 3. Reconstructed images from adversarial features with different noise bounds on datasets LFW, CFP-FP, and AgeDB-30.

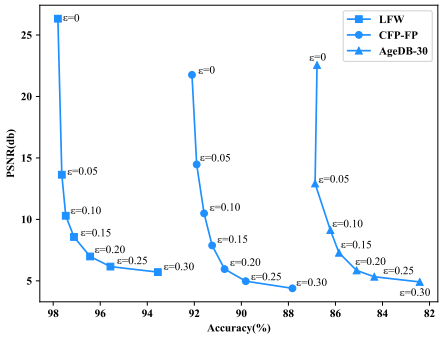

Figure 4. Performance of AdvFace in terms of the trade-off between PSNR and Accuracy with different noise bounds.

the ResRec is used as the reconstruction network and the shadow model in AdvFace.

Fig. 3 shows that the strength of privacy protection is proportional to ε. That is, a larger ε always provides a stronger privacy protection. Moreover, we can see that face privacy can be well protected when $\varepsilon \geq 0.2$. Fig. 4 shows the accuracy and PSNR values under different ε. We can see that the accuracy decreases as ε increases, which is opposite to the changing trend of privacy protection (i.e., decreasing PSNR means improving privacy protection). The bottom left of Fig. 4 presents a good performance on both face privacy protection and face recognition accuracy. It can be seen that the point of $\varepsilon = 0.2$ is closest to the bottom left, and thus we set ε as 0.2 for the following evaluation.

5.3. Defense against Privacy Attacks

We now compare AdvFace with baselines to evaluate its defense performance against privacy attacks (including reconstruction attacks and replay attacks) on the datasets of LFW, CFP-FP, and AgeDB-30. Both the reconstruction network and the shadow model adopt the ResRec architecture.

Defense against Reconstruction Attacks: Fig. 5 shows the

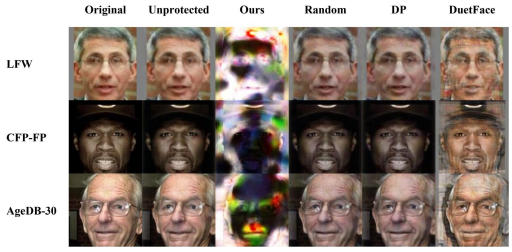

Figure 5. Reconstructed images from facial features generated by different privacy protection methods.

reconstructed images from facial features protected by different methods. As shown in the third column, the reconstructed images from the generated adversarial features by our proposed AdvFace are hard to distinguish, while those protected by other methods (columns 4-6) undergo much information leakage about the original images, which can identify the person. Tab. 1 summarizes the average values of SSIM, PSNR, and MSE of the reconstructed images and original images. We can see that our method always has a lower SSIM/PSNR and a higher MSE, which demonstrates that our AdvFace outperforms other protection methods on the defense performance against the reconstruction attacks.

Defense against Replay Attacks: These attacks input the reconstructed facial image to the face recognition system for malicious face authentication, where we use the SRRA to measure the defense effectiveness. Tab. 1 shows the outstanding performance of AdvFace in preventing attackers from launching replay attacks. Specifically, the value of SRRA significantly decreases (from 97.40% to 4.03%) after using AdvFace for privacy protection. In contrast, other protection methods fail to resist the replay attacks, with the SRRA being 94.73%, 93.97%, and 95.17% for Random Perturbation, DP, and DuetFace, respectively.

5.4. Accuracy of Face Recognition

In this subsection, we compare AdvFace with baseline methods to evaluate its performance in face recognition accuracy. As mentioned in Sec. 4.4, AdvFace can work in the online mode or offline mode according to whether the face recognition network is retrained or not. Note that, unless specifically marked as offline mode, the AdvFace works in the online mode.

As shown in Tab. 2, the online AdvFace integrated into the unprotected methods causes a small drop in accuracy (i.e., 1.7%, 2.57%, and 2.47% lower than the unprotected

Table 2. The performance of privacy protection methods in terms of face recognition accuracy (note that Ours (online) represents that AdvFace is employed as a plug-and-play privacy-enhancing module and Ours (offline) indicates that we use the adversarial features to retrain the downstream face recognition network to further improve the accuracy).

Mehods	LFW	CFP-FP	AgeDB-30
Unprotected	98.13%	93.16%	87.57%
Random	97.20%	91.67%	86.60%
DP	96.27%	90.84%	85.12%
DuetFace	98.02%	84.37%	87.10%
Ours(online)	96.43%	90.59%	85.10%
Ours(offline)	97.78%	92.04%	86.35%

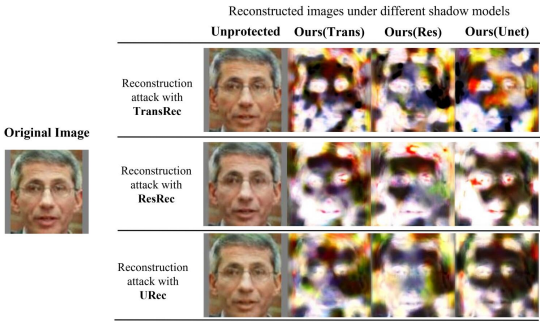

Figure 6. Transferability of AdvFace on defending against reconstruction attacks. Ours(Trans), Ours(Res), and Ours(Unet) represent that we adopt the transpose, resnet, and unet network as the shadow model, respectively. TransRec, ResRec, and URec represent that the transpose, resnet, and unet network is employed as the reconstruction network, respectively.

method on the three datasets). However, we would like to clarify that such a slight decrease in accuracy is acceptable given the outstanding privacy protection performance of AdvFace. Furthermore, when AdvFace works in the offline mode, it achieves comparable accuracy with other protection methods. For instance, the accuracy of offline AdvFace is only 0.35%, 1.12%, and 1.22% lower than the unprotected benchmark.

5.5. Transferability of AdvFace

This subsection investigates the impact of the shadow model structure on AdvFace and the transferability of adversarial features generated by AdvFace. To this end, we first evaluate the performance of AdvFace on face recognition accuracy with different shadow model structures. Then, we evaluate the performance of AdvFace with different shadow model structures regarding the defense effectiveness against different reconstruction networks.

Tab. 3 shows that similar face recognition accuracies are achieved by AdvFace with different shadow models, implying that the shadow model structure has only a slight influence on the performance of AdvFace. Thus, AdvFace is easy to implement.

In Fig. 6, we show the facial images reconstructed from the adversarial features by three different reconstruction networks. In Tab. 4, we quantitatively describe the average quality of reconstructed images with SSIM and PSNR on three datasets. We can see the defense effectiveness of AdvFace is maintained when encountering different attack networks, which validates the transferability of the adversarial features generated by AdvFace. Specifically, the adversarial features generated by AdvFace (based on any shadow model) can defend against different reconstruction attacks.

Table 3. The performance of AdvFace in the face recognition accuracy with different shadow models.

Protect Method	LFW	CFP-FP	AgeDB-30
Ours(Trans)	96.53%	90.77%	85.13%
Ours(Res)	96.43%	90.59%	85.10%
Ours(Unet)	96.42%	90.13%	84.58%

Table 4. The performance of AdvFace in terms of SSIM and PSNR under different reconstruction networks.

Metric	Protect Method	Attack Method	LFW	CFP-FP	AgeDB-30
SSIM	Ours(transpose)	TransRec	0.20	0.16	0.19
		ResRec	0.27	0.20	0.23
		UReC	0.26	0.20	0.22
	Ours(resnet)	TransRec	0.23	0.19	0.21
		ResRec	0.28	0.23	0.24
		UReC	0.29	0.24	0.24
	Ours(Unet)	TransRec	0.24	0.20	0.23
		ResRec	0.27	0.21	0.23
		UReC	0.28	0.23	0.23
PSNR	Ours(transpose)	TransRec	6.93	7.40	5.92
		ResRec	6.84	5.96	5.59
		UReC	7.33	6.81	6.14
	Ours(resnet)	TransRec	6.90	7.01	5.92
		ResRec	6.97	5.98	5.85
		UReC	7.47	6.70	6.20
	Ours(Unet)	TransRec	6.73	7.36	5.88
		ResRec	6.57	6.17	5.48
		UReC	7.01	6.94	5.95

6. Conclusions

In this work, we proposed an adversarial features-based face privacy protection (AdvFace) method to generate privacy-preserving adversarial features against the reconstruction attack while maintaining face recognition accuracy. Extensive experimental results show the superior performance of AdvFace in defending against reconstruction attacks compared to those state-of-the-art methods. At the same time, AdvFace can be easily integrated into deployed face recognition systems as a plug-in privacy-enhancing module. Moreover, the experiments also validate that AdvFace can achieve a desirable tradeoff between accuracy and utility and generate adversarial features with excellent transferability, promising its practicality and applicability.

Acknowledgments

This work was supported by Key R&D Program of Zhejiang (Grant No. 2022C01018), National Natural Science Foundation of China (Grants No. 62122066, U20A20182, 61872274, 62102337) and National Key R&D Program of China (Grant No. 2021ZD0112803).

References

[1] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In *IACR International Workshop on Public Key Cryptography*, pages 733–751. Springer, 2015. 1, 3

[2] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Bmvc*, volume 1, page 3, 2016. 6

[3] Mahawaga Arachchige Pathum Chamikara, Peter Bertok, Ibrahim Khalil, Dongxi Liu, and Seyit Camtepe. Privacy preserving face recognition utilizing differential privacy. *Computers & Security*, 97:101951, 2020. 1, 3

[4] Forrester Cole, David Belanger, Dilip Krishnan, Aaron Sarna, Inbar Mosseri, and William T Freeman. Synthesizing normalized faces from facial identity features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3703–3712, 2017. 2, 3

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 6

[6] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 5

[7] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837, 2016. 1, 2, 3

[8] Amir Erfan Eshratifar, Mohammad Saeed Abrishami, and Massoud Pedram. Jointdnn: An efficient training and inference engine for intelligent mobile cloud computing services. *IEEE Transactions on Mobile Computing*, 20(2):565–576, 2019. 3

[9] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015. 1, 2

[10] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 129–148. Springer, 2011. 1, 3

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[13] Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 148–162, 2019. 1, 2, 3

[14] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008. 6

[15] Jiazhen Ji, Huan Wang, Yuge Huang, Jiaxiang Wu, Xingkun Xu, Shouhong Ding, ShengChuan Zhang, Liujuan Cao, and Rongrong Ji. Privacy-preserving face recognition with learnable privacy budgets in frequency domain. *arXiv preprint arXiv:2207.07316*, 2022. 2, 3

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[17] Jong Hwan Ko, Taesik Na, Mohammad Faisal Amir, and Saibal Mukhopadhyay. Edge-host partitioning of deep neural networks with feature space encoding for resource-constrained internet-of-things platforms. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2018. 3

[18] Xiaoyu Kou, Ziling Zhang, Yuelei Zhang, and Linlin Li. Efficient and privacy-preserving distributed face recognition scheme via facenet. In *ACM Turing Award Celebration Conference-China (ACM TURC 2021)*, pages 110–115, 2021. 1, 3

[19] Ang Li, Jiayi Guo, Huanrui Yang, Flora D Salim, and Yiran Chen. Deepobfuscator: Obfuscating intermediate representations with privacy-preserving adversarial learning on smartphones. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, pages 28–39, 2021. 1, 2, 3, 6

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 6

[21] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 5

[22] Guangcan Mai, Kai Cao, Xiangyuan Lan, and Pong C Yuen. Secureface: Face template protection. *IEEE Transactions on Information Forensics and Security*, 16:262–277, 2020. 1, 3, 6

[23] Guangcan Mai, Kai Cao, Pong C Yuen, and Anil K Jain. On the reconstruction of face images from deep face templates. *IEEE transactions on pattern analysis and machine intelligence*, 41(5):1188–1202, 2018. 1, 2, 3

[24] Yunlong Mao, Jinghao Feng, Fengyuan Xu, and Sheng Zhong. A privacy-preserving deep learning approach for face recognition with edge computing. In *HotEdge*, 2018. 1, 3

[25] Yuxi Mi, Yuge Huang, Jiazhen Ji, Hongquan Liu, Xingkun Xu, Shouhong Ding, and Shuigeng Zhou. Duetface: Collaborative privacy-preserving face recognition via channel splitting in the frequency domain. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6755–6764, 2022. 2, 3, 6

[26] Alexis Mignon and Frédéric Jurie. Reconstructing faces from their signatures using rbf regression. In *British Machine Vision Conference 2013*, pages 103–1, 2013. 2

[27] Pranab Mohanty, Sudeep Sarkar, and Rangachar Kasturi. From scores to face templates: A model-based approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(12):2065–2078, 2007. 2

[28] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 51–59, 2017. 6

[29] Anton Razzhigaev, Klim Kireev, Edgar Kaziakhmedov, Nurislam Tursynbek, and Aleksandr Petiushko. Black-box face recovery from identity features. In *European Conference on Computer Vision*, pages 462–475. Springer, 2020. 1, 2

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6

[31] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 6

[32] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–9. IEEE, 2016. 6

[33] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[34] Taihong Xiao, Yi-Hsuan Tsai, Kihyuk Sohn, Manmohan Chandraker, and Ming-Hsuan Yang. Adversarial learning of privacy-preserving and task-oriented representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12434–12441, 2020. 1, 3

[35] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 6

[36] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016. 6

[37] Andrey Zhmoginov and Mark Sandler. Inverting face embeddings with convolutional neural networks. *arXiv preprint arXiv:1606.04189*, 2016. 1, 2, 3