FaceObfuscator: Defending Deep Learning-based Privacy Attacks with Gradient Descent-resistant Features in Face Recognition

Shuaifan Jin[†,≀] He Wang[†,≀] Zhibo Wang[†,≀,*] Feng Xiao[♮] Jiahui Hu[†,≀] Yuan He[‡]
Wenwen Zhang[‡] Zhongjie Ba[†,≀] Weijie Fang[♯] Shuhong Yuan[♯] Kui Ren[†,≀]

[†] *The State Key Laboratory of Blockchain and DataSecurity, Zhejiang University, P. R. China*
[≀] *School of Cyber Science and Technology, Zhejiang University, P. R. China* [‡] *Alibaba Group, P. R. China*
[♮] *Palo Alto Networks, USA* [♯] *Information Technology Center, Zhejiang University, P. R. China*

{shuaifanjin, wanghe_71, zhibowang}@zju.edu.cn, fxiao@paloaltonetworks.com, jiahuihu@zju.edu.cn
{heyuan.hy, karida.zww}@alibaba-inc.com, {zhongjieba, fangwj, shyuan, kuiren}@zju.edu.cn

Abstract

As face recognition is widely used in various security-sensitive scenarios, face privacy issues are receiving increasing attention. Recently, many face recognition works have focused on privacy preservation and converted the original images into protected facial features. However, our study reveals that emerging Deep Learning-based (DL-based) reconstruction attacks exhibit notable ability in learning and removing the protection patterns introduced by existing schemes and recovering the original facial images, thus posing a significant threat to face privacy. To address this threat, we introduce FaceObfuscator, a lightweight privacy-preserving face recognition system that first removes visual information that is non-crucial for face recognition from facial images via frequency domain and then generates obfuscated features interleaved in the feature space to resist gradient descent in DL-based reconstruction attacks. To minimize the loss in face recognition accuracy, obfuscated features with different identities are well-designed to be interleaved but non-duplicated in the feature space. This non-duplication ensures that FaceObfuscator can extract identity information from the obfuscated features for accurate face recognition. Extensive experimental results demonstrate that FaceObfuscator's privacy protection capability improves around 90% compared to existing privacy-preserving methods in two major leakage scenarios including channel leakage and database leakage, with a negligible 0.3% loss in face recognition accuracy. Our approach has also been evaluated in a real-world environment and protected more than 100K people's face data of a major university.

1 Introduction

Face recognition, a technology that utilizes the human face for biometric identification, is widely used in security-related scenarios. As facial information is a unique biometric trait of an individual, any potential leakage poses a significant risk. If malicious actors gain access to this facial information, they could potentially use it to impersonate the affected individuals and carry out unauthorized activities. Moreover, since the biometric profiles of individuals' facial characteristics are extremely hard to change, the hazards caused by the leakage will be long-lasting. Thus, the privacy issue of face recognition has received more and more attention in recent years. Various governments have enacted legislation pertaining to safeguarding facial privacy, such as the European Union's General Data Protection Regulation (GDPR [1]). Despite advancements, many commercial face recognition systems [2–4] still store sensitive facial features. These features are machine learning attributes extracted from facial images, and in some cases, the original photos are also stored. This practice significantly jeopardizes privacy, particularly in instances of database leaks. In such situations, confidential customer biometric data could potentially be extracted from facial features through a process known as a reconstruction attack [40, 43, 61]. Incidents of such leakages have occurred in the past, as exemplified by SenseNets' leakage of 2.5 million face data, Clearview AI's exposure of billions of photographs, and the U.S. Customs and Border Protection's (CBP) disclosure of an unspecified number of travelers' photos.

The root cause of such insecure privacy practices is twofold. Firstly, existing protection approaches often struggle to meet the practical requirements of real-world face recognition. For example, researchers design cryptography-based methods [26, 32, 42, 50] to encrypt facial features and even perform face recognition within the encrypted domain, thereby safeguarding facial privacy. Despite these cryptography-based methods taking practicality into account by minimizing overhead through various optimizations, their computation-intensive nature makes it difficult to scale to larger datasets. Consequently, as the volume of real-world production data increases, these methods incur greater computation and communication overhead [36, 41, 47].

Secondly, the recent advancements made by privacy adversaries often make it challenging for existing approaches to

[*] Zhibo Wang is the corresponding author.

Table 1: Experimental findings on the vulnerability of recent privacy-preserving methods. ● represents a good protection against attacks; ◐ represents a poor protection to attacks; ○ represents no protection against attacks.

Methods	Mapping Type (Images→Features)	Defend Against Privacy Attacks		Accuracy
		DL-based	DL-based (Adversarial Training)	
InstaHide [20]	Random	●	◐	0% – 10%
Cloak [41]	DL + Polynomial	–	○	80% – 90%
AdvFace [53]	DL + Polynomial	●	○	80% – 90%
PPFR-FD [52]	Orthogonal + Liner	◐	○	90% – 95%
DCTDP [24]	Orthogonal + Liner	○	○	90% – 95%
Duetface [39]	Orthogonal	○	○	90% – 95%

strike a balance between usability and effectiveness in protection. Because of the usability limitation of cryptography-based methods, there is a tendency to use more lightweight transformation-based methods. For instance, recent works [8, 20, 41, 53] use differential privacy or adversarial samples to generate noises and use such noises to perturb the facial features against privacy attacks. Other works [24, 39, 52] converted facial images into frequency features and removed several non-critical frequency channels, to protect facial privacy. However, our research finds that existing lightweight methods are vulnerable to state-of-the-art privacy attacks.

In particular, real-world attackers are very powerful given recent advances in DL-based reconstruction attacks [9, 12, 18, 35, 58, 61]. These attacks aim to learn the inverse mapping from facial features to facial images by training a reconstruction network with powerful fitting capabilities. Using the trained reconstruction network, the attacker can directly recover the facial image from the protected facial features. Our experiments across 8 widely-used datasets with visual and quantitative results demonstrate that previous privacy-preserving methods cannot provide comprehensive privacy protection when countering such a real-world scenario. Specifically, as summarized in Table 1, Cloak [41], AdvFace [53], PPFR-FD [52], DCTDP [24], Duetface [39] cannot defend against DL-based reconstruction attacks that use adversarial training. Although InstaHide [20] can partially protect face privacy, it suffers from a significant accuracy decline.

Hence, there is an urgent need to propose a new scheme that can provide strong privacy protection and guarantee the accuracy of face recognition at the same time, which leads to two mutually constraining challenges. *Challenge 1: How to disrupt the fitting ability of DL-based reconstruction attacks?* Previous privacy-preserving schemes [24, 39, 41, 52, 53] commonly mapped the same identity of facial images to the same categories of features, so the reconstruction network can always appropriate the corresponding inverse mapping (from features to images) via gradient descent to recover facial images. Therefore, it is a challenge to find a novel mapping that can essentially defend against reconstruction attacks. *Challenge 2: How to ensure the availability of face recognition at the same time?* If the fitting ability of the reconstruction network has already been disrupted successfully, this disruption may also have a huge interference on the face recognition network, resulting in serious degradation of face recognition accuracy, just as the result of InstaHide [20]. Therefore, it is challenging to preserve the information that can be used for face recognition while disturbing the fitting ability of the reconstruction network.

To address those challenges, we propose a lightweight privacy-preserving face recognition system, called FaceObfuscator, which has two key characteristics. *(1) FaceObfuscator can generate obfuscated features to prevent DL-based face reconstruction attacks from recovering facial images.* To this end, we first remove the visual information that is non-crucial for face recognition in frequency domain to weaken the reconstruction results. Then, we generate different candidate feature sets containing multiple obfuscated features for different facial features, and randomly select an obfuscated feature from the candidate feature set as a final facial feature, as shown in Fig. 2. Note that the obfuscated features in different candidate feature sets are interleaved with each other, which disrupts the essence of the reconstruction network, i.e., the mapping from features to images, so these features are gradient descent-resistant to the reconstruction network. Thus, the obfuscated features can defend against DL-based reconstruction attacks to achieve privacy protection. *(2) FaceObfuscator can still utilize the unique candidate feature sets of obfuscated features to maintain face recognition accuracy.* Since the obfuscated features in different candidate feature sets are simultaneously designed to be not duplicated, a specific candidate feature set could be found through an obfuscated feature. It is noted that the candidate feature set is generated from facial features and contains information about facial features. Therefore, identity information can be extracted from the obfuscated feature for face recognition. Extensive evaluation comparing 6 recent privacy-preserving methods on 8 widely-used datasets, shows FaceObfuscator's face privacy protection capability increases around 90% with a tolerable 0.3% accuracy sacrifice. Due to its practicability, our approach has also been evaluated in a real-world environment and protected more than 100K people's face data of a major university.

Our main contributions can be summarized as follows:

- We reveal the vulnerability of existing privacy-preserving face recognition schemes against DL-driven reconstruction attacks through extensive visual and quantitative experiments across 8 widely-used datasets.

- We propose a novel privacy-preserving face recognition system called FaceObfuscator, which generates interleaved but non-duplicated obfuscated features from their candidate feature sets. The obfuscated features are gradient descent-resistant to prevent face reconstruction attacks while containing identity information for accurate face recognition.

- FaceObfuscator serves as a lightweight privacy protection method that can be used for real-time face recognition. The time cost of FaceObfuscator from inputting an image to completing face recognition is comparable to the mainstream face recognition baseline without privacy protection capabilities, and is faster than most face privacy protection methods. The storage cost for its obfuscated features is notably lower, at 98KB per image, compared to the recent face recognition methods.

- FaceObfuscator has been used as one of the protection methods to guard the face privacy of over 100,000 individuals in a major university. Extensive experiments demonstrate that FaceObfuscator outperforms the state-of-the-art privacy-preserving face recognition methods in terms of superior privacy protection performance with a negligible face recognition accuracy loss.

2 Related Work

This section first overviews the related works on face reconstruction attacks, and then introduces recent privacy-preserving face recognition schemes and their vulnerabilities.

2.1 Face Reconstruction Attacks

In the early years, Mohanty et al. [43] utilized a linear approach to reconstruct facial images from face templates. Mignon et al. [40] used the RBF-regression in eigenspace to reconstruct facial images from their signatures. However, these traditional methods cannot cope with the features generated by the later emergence of deep neural networks or complex mappings.

Nowadays, mainstream face reconstruction attacks can be mainly divided into two kinds, i.e., optimizing-based and DL-based. The optimizing-based attacks [13, 45, 46] iteratively optimize input based on feedback from the face recognition system until the original image is recovered. However, these methods require numerous query operations and are heavily reliant on feedback from the face recognition system. Alternatively, a more practical strategy is the DL-based attack, whose objective is to establish an inverse mapping from features to images. Zhmoginov et al. [61] pioneered the utilization of DNNs to invert face embeddings into realistic images. Cole et al. [9] generated facial images from the facial features extracted by the face recognition network. Dosovitskiy et al. [12] and Mai et al. [35] utilized an up-convolutional neural network to reconstruct visual images from features. He et al. [18] enabled a malicious participant in collaborative inference to reconstruct the input of other participants. Moreover, some works [22, 28, 58] guided the inversion from labels or features to images by Generative Adversarial Networks. In this paper, *DL-based attacks are considered the primary threat due to their heightened attack capability and broader applicability.*

2.2 Privacy-preserving Face Recognition

In recent years, some privacy-preserving face recognition schemes have been proposed, which can be mainly divided into two categories, i.e., cryptography-based methods and non-cryptography methods.

The cryptography-based methods usually perform face recognition in encryption space to defend against privacy attacks, e.g., homomorphic encryption [14, 26], oblivious protocols [32], functional encryption [5], key-based CNN [34], random matrix [30], etc. *In fact, real-world face recognition systems rarely employ cryptography on the client side due to the contradiction between their weak computational power and huge computational overheads, which has a 1000x even 10000x slow down [41].*

Consequently, there is a growing preference for more lightweight transform-based methods. Some methods [8, 20, 20, 41, 53] transform facial features against reconstruction attacks by adding various types of noise. Chamikara et al. [8] used local differential privacy that adds perturbation to eigenfaces. Mireshghallah et al. [41] filtered out features that were less relevant to the target task by perturbation and subsequently protected them with a constant suppression value. [20] employed a one-time key to amalgamate different images for the protection of specific ones. Wang et al. [53] used adversarial noise to protect existing facial features. Recently, some methods [24, 39, 52] have achieved privacy protection by transforming images to the frequency domain. Wang et al. [52] shuffled and mixed up the channels in the frequency domain for privacy-preserving. Ji et al. [24] introduced differential privacy to the frequency domain. Mi et al. [39] removed some low-frequency channels in the frequency domain, which is essential for visualization. *However, our experiments reveal that all of these preferred lightweight methods are ineffective in achieving privacy preservation and maintaining face recognition accuracy at the same time when confronted with realistic DL-based reconstruction attacks*. This inefficiency stems from the regular mapping of images to features in previous perturbations and transformations. This mapping fails to prevent deep learning within the reconstruction network from appropriating its inverse mapping from features back to images, thus enabling the recovery of the facial images.

In summary, previous methods either involve impractical additional overhead or cannot maintain face recognition accuracy and protect face privacy at the same time.

3 Preliminary

This section first outlines typical face recognition systems and illustrates their susceptibility to privacy attacks. Subsequently, considering the weak capabilities of the attackers set in previous privacy-preserving works, we expound upon a more powerful and practical threat model applicable to nearly all face recognition systems, aiming to provide a more realistic

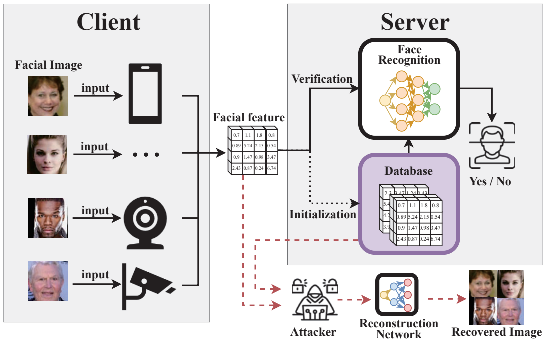


Figure 1: Typical face recognition systems and their potential attack surfaces.

assessment of the privacy-preserving capacities.

3.1 Typical Face Recognition System

Nowadays, mainstream face recognition systems, illustrated in Fig. 1, commonly adopt a client-server architecture. The client side collects and preprocesses facial images from users, and then transmits the preprocessed image to the server side. On the other hand, the server side utilizes its powerful computing power to execute the final face recognition process.

In order to further protect privacy, current privacy-preserving face recognition schemes usually convert facial images into facial features during the client's preprocessing stage to achieve privacy protection. Correspondingly, the database on the server side does not store the original image but the facial features. Though it is true that the facial features are already unrecognizable to humans, there remains a potential risk of privacy compromise due to the presence of the aforementioned DL-driven reconstruction attack, which aims to recover the original facial images.

3.2 Threat Model

Attack Scenarios. In this paper, we consider the server in face recognition to be trusted, but there exists a powerful external attacker whose goal is to recover facial images from facial features to invade users' privacy. It is worth noting that prior attempts at privacy preservation have often underestimated the capabilities of the attackers. For instance, chamikara et al. [8] employed traditional methods such as PCA-based models to reconstruct images and some works [24, 52] even directly considered inverse DCT operation which is the reverse operations of preprocessing as a white-box attack. Mireshghallah et al. [41] only considered attacks on the sensitive attributes of faces, ignoring attacks that can directly reconstruct the entire face. In addition, some works [24, 39, 52] did not adopt a strategy of adversarial training while training DL-based reconstruction network, leading to misjudgment of their scheme's ability to protect privacy.

In fact, attackers in the realistic scenario [7,17,23,31,38,49] can utilize deep learning-based reconstruction attacks and employ the adversarial training strategy, which can retrain the reconstruction network using pairs of protected facial features and facial images to learn its new mapping relationship for recovering facial images. This advanced attack methodology is able to breach established privacy-preserving techniques and obtain facial information, as summarized in Table 1. Therefore, we adopt such a powerful threat model to better accurately evaluate the efficacy of our privacy-preserving method.

Attacker's Knowledge. We assume the attacker has the following knowledge:

- **Leaked facial features** $\mathcal{X}' = \{x_1', x_2', \cdots, x_n'\}$: the attacker has intercepted the data in the **transmission channel** or gained access to the compromised **database**.
- **The client of the face recognition system** $\mathrm{C}(\cdot)$: the attacker can get the latest black-box client of the face recognition system by purchasing from the service provider.

Attacker's Strategy. With the mentioned knowledge, the attacker can obtain protected feature-image pairs $(\mathcal{X}, \mathcal{Y})$ from public face datasets $\mathcal{Y}$ through the latest client: $\mathcal{X} = C(\mathcal{Y})$, where $\mathcal{X}$ is the facial feature datasets. Then, attackers can train its corresponding reconstruction network $G(\cdot)$ with $(\mathcal{X}, \mathcal{Y})$, and finally feed the leaked facial features $\mathcal{X}'$ to the reconstruction network to recover the original images: $\mathcal{Y}_{target} = G(\mathcal{X}')$.

All DL-based face reconstruction attacks are based on the above strategy with adjustments of the training dataset, loss function, or network architecture. In this paper, we adopted two of the most prevalent and threatening types of DL-based reconstruction attacks, which use Deconvolutional Networks (DN) [9, 12, 18, 35, 57, 61] and Conditional Generative Adversarial Networks (cGAN) [22, 28, 58], respectively, to train reconstruction networks $G(\cdot)$.

4 Gradient Descent-resistant Face Protection

In this section, we present our novel Gradient Descent-Resistant system, called FaceObfuscator, for privacy-preserving face recognition. In Section 4.1, we provide an overview of our proposed system. In Section 4.2, we first investigate the role of frequency channels for face recognition. By retaining only the most critical channels for face recognition, we aim to reduce as much visual information as possible to weaken face reconstruction while preserving recognition accuracy. In Section 4.3, we reveal our innovative strategy developed to secure residual visual information by making it challenging for DL-based reconstruction networks to grasp it. We will elaborate on how we develop a mathematical model to generate an obfuscated feature that can be exploited for face recognition and how

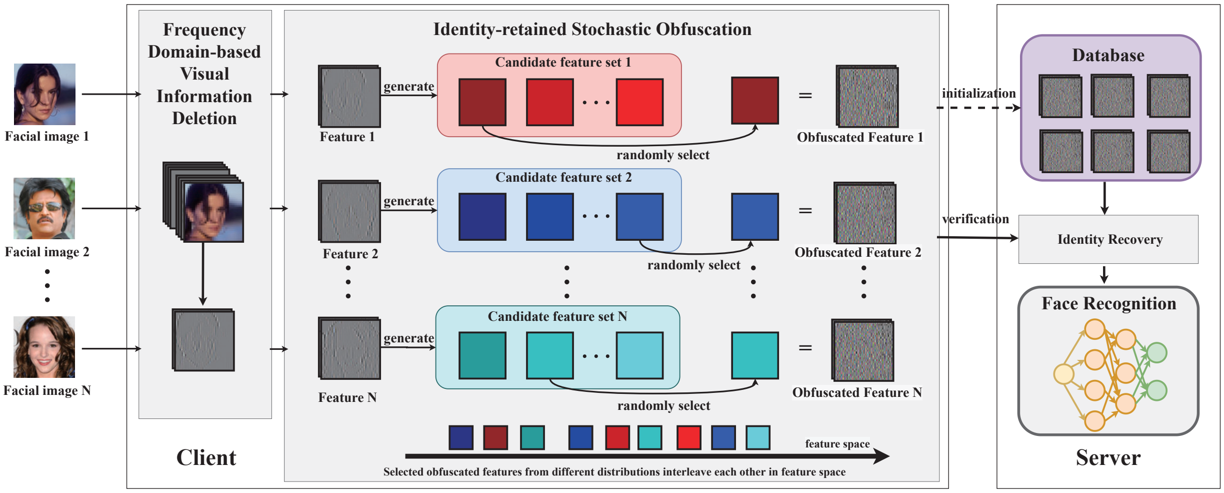
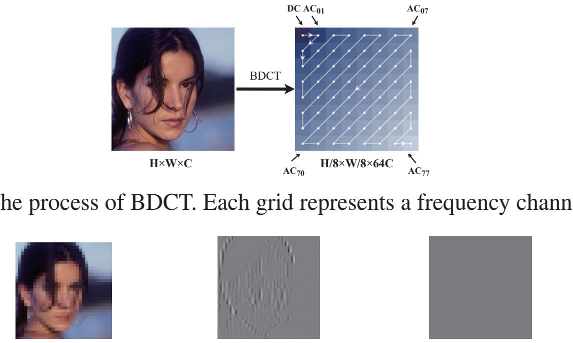


Figure 2: Overview of FaceObfuscator. On the client side, we first convert an original image into frequency channels and remove most of the frequency channels, then we use the remaining channels to generate a corresponding candidate feature set, and finally randomly select obfuscated features from this candidate feature set to resist privacy attacks. On the server side, the database stores obfuscated features, which will be processed by identity recovery before face recognition.



(a) The process of BDCT. Each grid represents a frequency channel.

(b) channel *DC* (c) channel AC_{01} (d) channel AC_{77}

Figure 3: After block discrete cosine transform (BDCT), different visual information in the image is arranged in a zigzag pattern from low to high frequency. (b) (c) (d) represent the visual information in frequency channel DC, AC_{01} and AC_{77}.

it successfully resists gradient descent to undermine the powerful DL-based face reconstruction capability.

4.1 Overview of FaceObfuscator

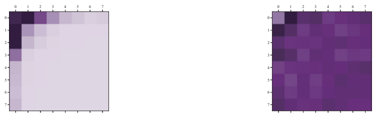
Our approach is based on an important observation of the neural network: the essence of DL-based face reconstruction attacks is to learn new mappings from protected features to original images for recovering facial images. Therefore, the core idea of FaceObfuscator is to disturb those mapping by resisting the gradient descent process in training reconstruction networks. To this end, we generate the obfuscated features by removing visual information non-critical for face recognition and introducing randomness to the features, which aims to make the obfuscated features of different facial images interleaved in the feature space so that the reconstruction network cannot distinguish them.

Fig. 2 shows the workflow of FaceObfuscator, which takes the facial images as the input of the client and generates the obfuscated features as the output of the client. The obfuscated features will be transmitted into the server and stored in the database of the server for face recognition to prevent potential face reconstruction attacks.

Specifically, FaceObfuscator has two components in the client (i.e., Frequency domain-based visual information deletion, Identity-retained stochastic obfuscation), and one component in the server (i.e. Identity recovery). On the client side, the client first feeds facial images into *Frequency domain-based Visual Information Deletion* to transform them into frequency-domain features, and then removes visual information non-critical for face recognition to weaken the reconstruction results. After that, through *Identity-retained Stochastic Obfuscation*, the client generates different candidate feature sets containing multiple obfuscated features for different facial features, and then randomly selects an obfuscated feature from the candidate feature set as a final facial feature. Note that the obfuscated features in different candidate feature sets are interleaved with each other, which disrupts the essence of the reconstruction network, i.e., the mapping from features to images, so these features are gradient descent-resistant to the reconstruction network. On the server side, the server stores obfuscated features transmitted from the client. Note that the obfuscated features in different candidate feature sets are designed to be not duplicated, so the server can find the candidate feature set of each obfuscated feature. Since the candidate feature set is generated from facial features and contains information about facial features, identity information can be extracted from the obfuscated feature through *Identity*

Table 2: Face recognition with a single frequency channel.

Channel	LFW	AgeDB-30	CALFW	CPLFW
DC	99.83	97.98	95.93	91.73
AC_{14}	99.50	95.82	95.08	87.37
AC_{37}	99.33	95.75	95.23	87.05
AC_{63}	99.57	97.02	95.65	90.08
AC_{77}	99.65	97.30	95.87	90.62



(a) w/o standardization (b) with standardization

Figure 4: The necessity of standardization for evaluating the importance of the channel for face recognition. (a) is the importance of frequency channels before standardizing the numerical values in frequency channels, whose results are inconsistent with the observation that each channel has visual information for face recognition; (b) is the re-assessed importance of frequency channels after standardizing, whose results are consistent with the above observation.

recovery for face recognition.

4.2 Frequency Domain-based Visual Information Deletion

We gain new insight on defending against DL-driven attacks from the unique perspective of the frequency domain, i.e., *most of the frequency channels can be removed with little loss of accuracy, which indeed contain visual information that is targeted by attacks*. Thus, we design our scheme to remove more than 90% of the channels to weaken the face visual reconstruction effectiveness (compared to previous schemes that only remove at most 50% of the frequency channels).

Next, we first explain the correlation between visual information and frequency channels (Section 4.2.1). Then, we introduce how we exploit the channel selection to achieve maximal visual information removal without compromising face recognition accuracy (Section 4.2.2).

4.2.1 Understanding Visual Information from the Frequency Domain

Because different frequency channels contain different visual information, as shown in Figs. 3b to 3d, previous works [24, 39, 52] indicated that it is possible to remove visual information to weaken the reconstruction results by partially removing frequency channels. So, following the standard Block Discrete Cosine Transform (BDCT) in JPEG compression [51], as shown in Fig. 3a, we first convert facial images to the frequency channels to separate out the visual information to be removed, say channel DC in Fig. 3b. However, previous works [24, 39, 52, 56] diverge in determining which channels should be retained to perform face recognition, i.e., most works [24, 52, 56] showed that low-frequency channels are much more crucial in face recognition, while one [39] showed another result that face recognition still works well even if removing such low-frequency channels.

To explore the real relationship between face recognition and visual information in frequency channels, we test the feasibility of using a single frequency channel for face recognition and observed that face recognition is possible with visual information from even a single frequency channel, as shown in Table 2, which is different from previous perception [24, 39, 52, 56]. This observation allows us to remove most of the frequency channels, rather than remove part of the frequency channels, to remove visual information against reconstruction attacks without compromising face recognition accuracy.

4.2.2 Identity-prioritized Frequency Channel Selection

Based on the above observation, our objective can be formulated as retaining only the most critical frequency channels for face recognition, where no further visual information can be precluded without compromising ID accuracy.

To further clarify the exact impact of each frequency channel on face recognition, we design an auxiliary network to measure different frequency channels' importance for face recognition. The auxiliary network assigns unconstrained weights directly to the initial input of the network, and these weights will be used to represent importance after training. Eventually, we reveal the true importance of each channel in face recognition by standardizing values in frequency channels, as shown in Fig. 4. Please refer to Appendix A for more details.

According to the importance of face recognition, we prioritize the removal of frequency channels with relatively low importance, thereby ensuring the retention of crucial information for face recognition. Subsequently, we succeed in retaining the channels that are the most critical for face recognition by iteratively removing channels while concurrently assessing their impact on face recognition. As shown in Fig. 5, only the retained 2 channels, i.e., channel AC_{01} and channel AC_{10}, are the most critical channels for face recognition, which ensures that there is no significant degradation in accuracy as well as weakening the face reconstruction results.

Notably, this approach can select the most important frequency domain channel based on different architecture facial recognition models, and flexibly adjust the number of reserved channels to ensure recognition accuracy while removing part of the visual information. In other words, this approach is scalable to different face recognition models.

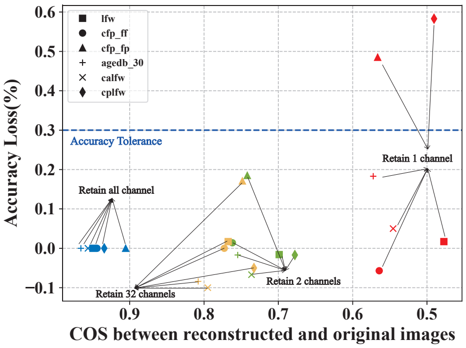


Figure 5: COS and Accuracy Loss with different numbers of channels retained on different datasets (the higher the COS value, the closer the reconstructed face is to the original face). When only 1 channel is retained, the loss of accuracy is too great to be acceptable (Accuracy loss exceeds the accuracy tolerance, say 0.3%); when many channels are retained, the reconstructed image is close to the original image. So retaining 2 channels can minimize the visual reconstruction effect while maintaining accuracy.

4.3 Identity-retained Stochastic Obfuscation

Since part of facial details may still be recovered from the retained frequency channels by DL-based face reconstruction attacks, it becomes imperative to further obfuscate the retained frequency channels to completely thwart potential privacy breaches.

Next, we introduce a novel algorithm to generate obfuscated features from their unique candidate feature sets, which are gradient descent-resistant to face reconstruction attacks (Section 4.3.1). Then, we formulate a corresponding algorithm for the server to utilize obfuscated features for face recognition (Section 4.3.2).

4.3.1 Gradient Descent-resistant Facial Feature Generation

The key to defending against reconstruction attacks is interfering with the gradient descent process of the reconstruction network. So, we generate obfuscated features that are gradient descent-resistant to reconstruction networks to provide strong privacy protection. This generation process involves two core steps. Step 1: For each original feature, we generate a unique candidate feature set by obfuscating the original feature by altering the sign direction for each element and value scale for each channel, where the sign direction represents the positive or negative of numerical values, and the value scale represents the magnitude of the numerical values. Step 2: We randomly select an obfuscated feature from the candidate feature set for each authentication. A step-by-step example is shown in Fig. 6 and the details of the algorithm are shown in algorithm 1.

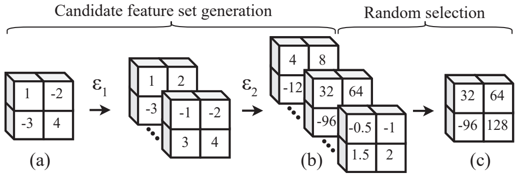


Figure 6: A step-by-step example of gradient descent-resistant facial feature generation. (a) is one of the frequency channels from the original feature, (b) is (a)'s corresponding candidate feature set, and (c) is (a)'s corresponding obfuscated feature.

Algorithm 1: Gradient descent-resistant facial feature obfuscation (client side).

```
input  : Facial features $f$ of size $h \times w \times c$
output : obfuscated features $e$ of size $h \times w \times c$
1  Initialize constant float $b$ from the server;
2  ***Preprocess***;
3  $a \leftarrow$ Self-Normalization$(f)$
     $= \{a_k \mid a_k = \frac{f_k - \mathrm{E}[f_k]}{\sqrt{\mathrm{Var}[f_k] + \beta}}, k \in [0, 1, 2, \cdots, c]\}$
4  ***Step 1: Generate*** candidate feature set $P$;
5  Obfuscate sign direction: $\varepsilon_1 = J_{h \times w \times c}$, all elements
   in $J$ are $(-1)^u$;
6  Obfuscate value scale: $\varepsilon_2 \leftarrow B_c = [b^v, b^v, \cdots, b^v_c]$;
7  Mask $\varepsilon \leftarrow \varepsilon_1 \odot \varepsilon_2$, all elements in $\varepsilon$ are $(-1)^u \times b^v$,
     where $\odot$ represents Hadamard product;
8  candidate feature set:
     $P(a, b) = a \odot \varepsilon = a \times (-1)^u \times b^v$
9  ***Step 2: Selecting an obfuscated feature from***
     candidate feature set $P$;
10 **for** $k \leftarrow 1$ **to** $c$ **do**
11     randomly select $e_k$ with $P(a_k, b)$, where $u$, $v$ is
         in integer range $[Z_{lower\ bound}, Z_{upper\ bound}]$;
12 obfuscated features $e \leftarrow \{e_1, \cdots, e_c\}$;
```

Generating different candidate feature sets from each facial feature (line 2 to line 8). Before resisting the gradient descent of the face reconstruction network, it's crucial to ensure the availability of obfuscated features for face recognition. Thus, we need to carefully design the position of the obfuscated features placed in the feature space. To this end, we design different candidate feature sets that contain multiple obfuscated features for different facial features, and *ensure the obfuscated features in different candidate feature sets are interleaved but not duplicated in the feature space*.

Such a design serves two purposes. The first purpose is to make it impossible for the reconstruction network to utilize the obfuscated features. Due to the obfuscated features of different facial images interleaved in the feature space, the obfuscated features from different candidate feature sets are not distinguishable to the reconstruction network. Therefore, the gradient descent of the reconstructed network will be resisted when the network tries to fit the mapping from obfuscated features to facial images. The second purpose is to preserve the identity information in the obfuscated features used for face recognition. The non-duplication of obfuscated features from different candidate feature sets ensures that there exists a many-to-one relationship between obfuscated features and candidate feature sets. Thus, a candidate feature set with a specific identity can be found from an obfuscated feature, allowing for the extraction of identity information within the obfuscated feature to perform face recognition accurately.

Specifically, for each facial feature (the remaining frequency channels after channel selection), the client first preprocesses the facial feature by self-normalization to make the features initially obfuscated, whose details are shown in Appendix B. Then, the client generates different candidate feature sets for different facial features, where each candidate feature set uniquely corresponds to a facial feature. The process to generate candidate feature sets can be expressed as $a \times (-1)^u \times b^v$, where a is the facial features after self-normalization, b is a constant value obtained from the server for initializing the client, $(-1)^u$ represents the obfuscation to facial features on the sign direction, b^v represents the obfuscation to facial features on the value scale, and u, v are discrete variables to be randomly selected, which ensures the features in different candidate feature sets are interleaved but not duplicated in the feature space.

Randomly selecting gradient descent-resistant features from their corresponding candidate feature sets (line 9 to line 12). After generating the corresponding candidate feature set for each feature, we can map a facial feature to a large number of obfuscated features. Since the obfuscated features from different identities are designed to be interleaved with each other, we can randomly select an obfuscated feature from the corresponding candidate feature set as the final facial feature for each authentication to resist gradient descent against reconstruction attacks. In this case, the mapping of original features to obfuscated features is a random mapping. Notably, we consider a single channel in each feature as the smallest unit of a. The values of u, v used for random selecting in each channel are randomly selected within an integer range, spanning from the minimum value $Z_{lower\ bound}$ to the maximum value $Z_{upper\ bound}$.

In the following, we will briefly demonstrate that obfuscated features can prevent face reconstruction network learning the mapping from features to facial images by resisting its gradient descent.

Formally, given a set of training samples $(\mathcal{X}, \mathcal{Y})$, where $\mathcal{Y}$ is a set of facial images, $\mathcal{X}$ is a set of corresponding facial features generated by $x = S(y)$, where S is the process of FaceObfuscator, x denotes a facial features, y denotes a facial image. The attacker's network can be defined as $G(x) = \hat{y}(x, \theta)$ with its corresponding loss function $l(\theta)$, where $\hat{y}$ is an estimate of the facial image y. Then, the process of gradient descent algorithm at θ_0 can be described as $\theta \leftarrow \theta_0 - \alpha \frac{\partial}{\partial \theta_0} l(\theta)$, where α denotes the learning rate of gradient descent, θ denotes the parameter to be learned in the neural network. Since the α is set by the attacker and unknown to us, we chose to implement a substantial manipulation on $\frac{\partial}{\partial \theta_0} l(\theta)$ to break the gradient descent algorithm. To simplify the demo, we will use the case of a single layer perceptron $\hat{y}(x, \theta) = \sum_{j=0}^{n} \theta^j x_j$ with an MSE loss function $l(\theta; x, y) = (\hat{y}(x, \theta) - y)^2$ as an example, where $y = [y_0]$, $x = [x_0, x_1, \cdots, x_n]^\top$, $\theta_0 = [\theta_0^0, \theta_0^1, \cdots, \theta_0^n]$. As single layer perceptron is the basis of neural networks, the derivation for multilayer networks with other structures can be derived from this in the same way. In this case, the gradient descent algorithm can be described as:

$$\frac{\partial}{\partial \theta_0} l(\theta) = \frac{\partial}{\partial \theta_0} (\hat{y}(x, \theta_0) - y)^2 = 2 \cdot \left(\sum_{j=0}^{n} \left(\theta_0^j \cdot x_j \right) - y_0 \right) \cdot x \tag{1}$$

As for our scheme, the core obfuscation process can be equated to put a random mask ε to the features, i.e., $e = \varepsilon \odot x$, where $\varepsilon = [\pm b^{v_0}, \pm b^{v_1}, \cdots, \pm b^{v_n}]^\top$, b is a constant value in the client given by the server, v_i is the randomly selected number range from the minimum value $Z_{lower\ bound}$ to the maximum value $Z_{upper\ bound}$. So the gradient descent after our obfuscation can be formulated as:

$$\frac{\partial}{\partial \theta_0} l(\theta) = 2 \cdot \left(\sum_{j=0}^{n} \left(\theta_0^j \varepsilon_j x_j \right) - y_0 \right) \cdot \varepsilon \odot x \tag{2}$$

Then, we use the Manhattan Distance between Eq. (2) and Eq. (1) to yield the degree of influence of obfuscation on the normal gradient descent:

$$\begin{aligned} & d_1 (Eq.\ (2), Eq.\ (1)) = \| Eq.\ (2) - Eq.\ (1) \|_1 \\ = & 2 \cdot \sum_{i=0}^{n} \left| \sum_{j=0}^{n} \left((\pm b^{v_i} b^{v_j} - 1) \theta_0^j x_i x_j \right) - (\pm b^{v_i} - 1) y_0 x_i \right| \end{aligned} \tag{3}$$

where $\theta_0^k y_i y_j$, $x_0 y_j$ are fixed values in the state of θ_0. As $\pm b^v$ is a stochastic value ranging from a very small value close to 0 to a very large value close to the upper limit of the floating point number, the distance of the obfuscated gradient varies considerably compared to the original gradient. To go a step further, when the reconstructed network performs gradient descent to fit the inverse mapping, the direction of the gradient descent is inconsistent at each iteration, which effectively resists the normal gradient descent to recover the facial images.

As for the security concern about the reversal of obfuscated features, two primary difficulties hinder attackers from recovering the original features. The first difficulty arises from the complexity of determining sign direction. As the direction of each element in the feature is entirely random, the resulting obfuscated feature for a given original feature will have $2^{h\times w\times c}$ possibilities, which is hard to reverse. The second difficulty is associated with the variability in the value scale. The value of each frequency channel in the feature also undergoes random changes, further expanding the possibilities of obfuscated feature by N^c, where N represents the number of choices within the integer range $\left[Z_{lower\ bound}, Z_{upper\ bound}\right]$. Consequently, the combination of these difficulties makes it nearly impossible for attackers to recover the original features successfully.

So far, the original facial features have been well obfuscated. Such obfuscated features will be transmitted to the server as output from the client and stored in the server's database for privacy protection.

4.3.2 Identity Recovery Based on Traceability of Candidate Feature Set

Algorithm 2: Identity recovery based on traceability of candidate feature set (server side).

```
  input  : Obfuscated features $e$ of size $h \times w \times c$
  output : Identity information $d$ of size $h \times w \times c$
1 Initializing $b$ preset with clients;
2 for $k \leftarrow 1$ to $c$ do
3     if $\max(|e_k|) > 0$ then
4         $r = round\ down(-\log_b \max(|e_k|))$;
5         $d_k = e_k \times b^{r+bias}$;
6     else if $\max(|e_k|) = 0$ then
7         $d_k = O_{h\times w}$;
8 Identity information $d \leftarrow \{d_1, \cdots, d_c\}$;
9 Individual ID $\leftarrow$ Face recognition($d$);
```

Given that each candidate feature set is generated from the corresponding facial feature, if we can ascertain the specific candidate feature set to which the obfuscated feature belongs, the extraction of identity information from the obfuscated features for face recognition becomes feasible. Thus, based on our designed non-duplicated candidate feature sets, we design an identity recovery method that cannot be learned by neural networks to trace back the candidate feature set corresponding to the obfuscated feature, as shown in algorithm 2. After that, the obfuscated features can be recovered to a specific feature in its corresponding unique candidate feature set and eventually utilized by face recognition on the server side.

Specifically, since Huang et al. [20] indicated that the sign direction factor $(-1)^u$ has minimal impact on the image classification tasks, i.e., face recognition in our case, we only offset the value scale factors b^v in the obfuscated features (line 2 to line 7) to extract the identity information, where each channel's maximum value is recalibrated to the range $\left[b^{bias}, b^{bias+1}\right)$ with the other values adjusted proportionally in multiples. For example, for facial features with different identities f_{Alice} and f_{Bob}, all obfuscated features of them will be uniquely transformed into final identities $f_{Alice} \times b^{\lambda}, f_{Bob} \times b^{\lambda}$ though algorithm 2, respectively, where f represents the facial feature of an individual before obfuscation, λ is the final value scale determined by $bias$. As a result, the final identity will be a specific obfuscated feature within the same interval range in different candidate feature sets. Such identities of different individuals are no longer interleaved with each other in the feature space, so they can be utilized by face recognition on the server side, as shown in Table 5 of Section 5.4.

Discussion on the value of b. In mathematics, we can choose b with any float value in range $(0,1)\cup(1,+\infty)$. The value of b will not impact the process of algorithm 2, since the obfuscated features of the same individual can always be unified to a specific feature in the range $\left[b^{bias}, b^{bias+1}\right)$, which is discriminative for face recognition. However, in practical deployment, the precision limitations of floating-point numbers impose constraints on b. For instance, considering a 32-bit floating-point number adhering to the IEEE 754 standard [27], the range of floating-point numbers is confined to $\pm[1,2)\times 2^{[-126,127]}$. Thus, the values of $a \times b^v$ must fall within this specified interval. In this case, when b is smaller, say $b = 2$, the range in which v can be selected is larger and the randomness is equally stronger. In addition, the value of b cannot be very close to zero because there may exist a floating-point precision loss problem during computing.

5 Experiments

To evaluate the efficacy of FaceObfuscator, we perform extensive evaluations of privacy protection and face recognition across diverse datasets. In the following sections, we will first introduce the experimental setup in Section 5.1, and discuss our evaluation results with three research questions:

- **RQ1**: How does our obfuscation impact face recognition accuracy? (Section 5.2)
- **RQ2**: Can FaceObfuscator effectively defend against various attacks to protect privacy? (Section 5.3)
- **RQ3**: How does each component contribute to the overall improvement of accuracy and privacy? (Section 5.4)

5.1 Experimental Setup

Datasets&Pre-prcocess. In the experiments, we use the MS-Celeb-1M [15], CelebA [33] as the train datasets and use the LFW [19], CFP-FF [48], CFP-FP [48], AgeDB-30 [44], CALFW [60], CPLFW [59], IJB-B [55], IJB-C [37] as the test datasets, whose details are shown in Appendix C.

Following common image pre-processes [10, 34, 53], we first randomly flip the images and resize them into 112×112. Then, we rescale the pixel value of images from $[0, 255]$ to $[0, 1]$. Finally, we normalize images with a mean of 0.5 and a variance of 0.5, whose values will be in $[-1, 1]$.

Models and Implementation Details. We will show the detailed setting of FaceObfuscator and DL-based face reconstruction attacks.

FaceObfuscator. Like most face recognition schemes, we choose Arcface [10] as the basis for the server side and choose Resnet50 [16] as its backbone. We use the SGD optimizer [6] with a learning rate of 0.1, momentum factor of 0.9, and weight decay of 5e-4. And we use a polyscheduler to dynamically adjust the learning rate according to a polynomial of given power 2. The b used to initialize the client is set to 2, and results for other b values are shown in Appendix D. All of the experiments of our scheme will be trained on MS-Celeb-1M [15] for 10 epochs. For easier deployment, we will turn the initial image into grayscale and upscale this image by 8 times when in the frequency domain scheme, at which point the image size after BDCT transformation aligns with the original and only a modification in the input channel for the face recognition network is needed.

DL-based attacks in threat model. In the experiments, we use the methods proposed by Mai et al. [35] and Isola et al. [22] as benchmark attacks of DN-based attacks and cGANs-based attacks respectively, which have proven to be effective and are also widely used by existing defense works [24,52,53]. Both of them are trained on the CelebA [33] dataset.

Methods for comparison. We compare our method with 8 different face recognition methods, specifically focusing on the last 6 methods that assert privacy protection capability.

Arcface [10] is the baseline method of face recognition for RGB images without privacy protection. **Arcface-FD [11]** is the baseline method of face recognition on the frequency domain without privacy protection. **InstaHide [20]** is a lightweight encryption-based method that incorporates the mix-up of k images, which we set to 2. **Cloak [41]** disturbs the input feature by a gradient-based perturbation model and we set its accuracy-privacy parameter to 1. **AdvFace [53]** proposes adversarial facial features obtained by adding adversarial latent noise to the original facial features to mislead the reconstruction network. **PPFR-FD [52]** is a privacy-preserving method in the frequency domain that disrupts the channel order and mixes them up, and face recognition is performed after the channels are reordered by energy. **DCTDP [24]** is a privacy-preserving face recognition with learnable privacy budgets of differential privacy, whose ε mean is set to 0.5. **Duetface [39]** is also a privacy-preserving method in the frequency domain that protects privacy by removing some of the low-frequency channels and deploying lightweight face recognition networks in the client to transfer additional auxiliary information to the server.

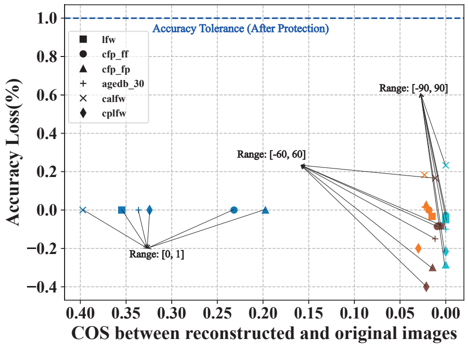


Figure 7: The effect of ranges of u, v on face reconstruction and face recognition.

Evaluation Metrics. We will use 5 metrics to evaluate the face privacy protection capabilities of different methods.

MSE, PSNR [25], SSIM [54] are three different methods for calculating the difference in pixels between two images. Specifically, a higher MSE, a lower PSNR, and a lower SSIM indicate a lower similarity between the reconstructed image and the original facial image, which implies a stronger defense. **COS** describes the cosine similarity between the identity vector of the reconstructed image and the identity vectors of the original image in the 512-dimensional facial feature space through another independent face recognition system. Specifically, a lower COS means a lower similarity between the reconstructed image and the original image. **SRRA [53] (success rate of replay attacks)** measures the success rate of replay attacks, which uses the reconstructed images from a certain face recognition system to cheat the same face recognition system for successful identity authentication. Specifically, a lower SRRA means a better privacy protection capability.

5.2 Accuracy of the Face Recognition (RQ1)

In this subsection, we will demonstrate what the impact of our FaceObfuscator on the accuracy of face recognition can be. We first discuss the effect of the range of u, v, i.e., $\left[z_{lower\ bound}, z_{upper\ bound}\right]$, on the accuracy, where u, v determine which obfuscated feature is selected from the candidate feature set. Then, we reassess the effect of the number of retaining frequency channels under our whole scheme, and finally compare our scheme with the state-of-the-art scheme.

The accuracy of face recognition remains almost unchanged as the range of values of u, v gradually expands. One factor that may affect the accuracy of face recognition is the upper and lower bounds of the perturbation magnitude. To this end, we gradually adjust the range of values of u, v from low to high, i.e., [0,1], [-30,30], [-60,60], [-90,90], to

Table 3: The performance of privacy protection methods in terms of face recognition accuracy. ● represents good protection against attacks; ◐ represents poor protection against attacks; ○ represents no protection against attacks; The yellow squares indicate flaws, say an accuracy loss over 3% compared to the baseline(Arcface) or poor protection capability, and The red squares indicates severe flaws, say an accuracy loss over 5% compared to the baseline(Arcface) or no protection capability.

Method	LFW (%)	CFP-FF (%)	CFP-FP (%)	AgeDB-30 (%)	CALFW (%)	CPLFW (%)	IJB-B (TPR@FPR) 10e-4 / 10e-5	IJB-C (TPR@FPR) 10e-4 / 10e-5	Channel Protection	Database Protection	Storage cost (KB/pic)	Time cost (ms/pic)
Arcface [10]	99.72	99.74	98.06	97.80	95.85	91.48	93.24 / 87.52	94.90 / 92.07	○	○	147	1.05
Arcface-FD [11]	**99.80**	**99.90**	98.50	97.97	95.97	92.08	**94.81 / 90.31**	**96.20 / 94.22**	○	○	9408	1.34
InstaHide [20]	96.05	92.43	86.89	74.60	76.97	74.62	0.71 / 0.17	0.70 / 0.19	◐	◐	147	0.98
Cloak [41]	99.50	99.69	96.29	96.20	95.17	88.38	86.45 / 67.37	88.99 / 77.61	○	○	147	**0.88**
AdvFace [53]	99.55	99.24	94.71	95.20	94.98	88.93	87.06 / 70.99	89.46 / 78.63	○	●	784	286.98
PPFR-FD [52]	99.82	99.76	98.33	98.00	95.87	91.97	94.08 / 89.40	95.67 / 93.57	○	○	1715	1.08
DCTDP [24]	99.83	99.81	**98.59**	**98.17**	96.00	92.00	94.64 / 90.62	96.11 / 94.21	○	○	9261	3.20
Duetface [39]	99.80	99.81	98.46	98.15	**96.12**	**92.33**	94.69 / 90.05	96.09 / 94.00	○	○	7938	4.40
Ours	99.68	99.71	97.57	97.65	95.83	91.33	92.93 / 88.48	94.57 / 92.17	●	●	**98**	1.18

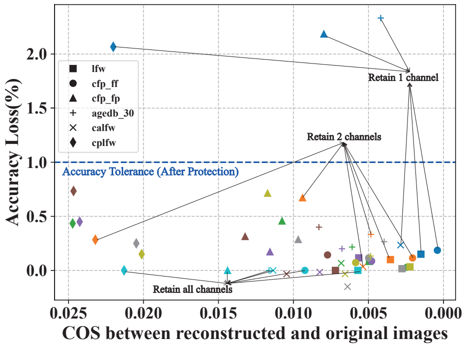


Figure 8: The effect of the number of retaining frequency channels on face reconstruction and face recognition.

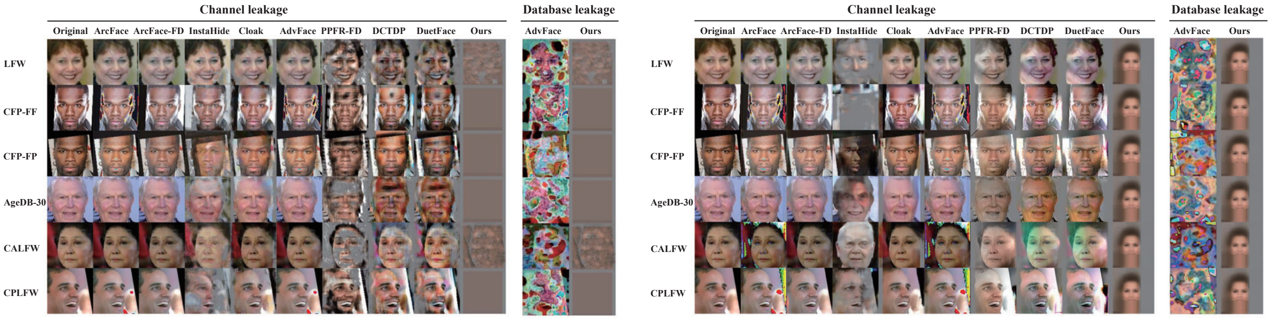
see how it affects privacy protection and face recognition accuracy respectively. Fig. 7 shows the trend of the accuracy and COS values (represent privacy-preserving capability) under different ranges, i.e., when the range of values of u, v gradually expands, the accuracy of face recognition remains almost unchanged. Therefore, we could theoretically choose a range as large as possible, as long as the value does not exceed the range of floating-point. Because the reconstructed network is no longer able to compute the gradient when the range reaches [-90, 90], we choose [-60, 60] in order to have a visual comparison with other schemes.

Channel AC_{01} and channel AC_{10} are still the most critical channels for face recognition. Another factor that can influence both face privacy protection capability and face recognition accuracy is the channels we retain in the face recognition system. So, we re-assess the importance of different frequency domain channels for face recognition under the whole scheme and prioritize the removal of channels with relatively low importance. Fig. 8 shows the accuracy and COS values when retaining a different number of channels. When the number of channels decreases from 64 to 2, there is no significant degradation in face recognition accuracy, whereas dropping to 1 channel results in a notable accuracy loss, as shown in the outlier in Fig. 8. Meanwhile, the lower the number of channels, the transmission and storage overheads required in face recognition, as well as the computational overheads, will be reduced accordingly. Therefore, combining the defending visual reconstruction effect, face recognition accuracy, and all kinds of overhead under different numbers of channels, our scheme only retains 2 channels, i.e., channel AC_{01} and channel AC_{10}, in Frequency domain-based visual information deletion. Notably, such fixed selection will not impair the protection effectiveness of our scheme. One reason for this is that the selected few channels only preserve little information that is useful for face reconstruction attacks. The other reason is that possibilities for obfuscated features derived from the same input still exceed 10^{7556} when only retaining 2 channels in our case, making recovery highly challenging.

FaceObfuscator has only a slight accuracy loss compared to the face recognition baseline. We compare the accuracy of face recognition with 2 baselines and 6 different privacy-preserving methods. As shown in Tab. 3, among the privacy protection schemes compared, the privacy-preserving schemes in the spatial domain, i.e., InstaHide, Cloak, and AdvFace, have a significant loss in accuracy, while the privacy-preserving schemes in the frequency domain, i.e., PPFR-FD, DCTDP, and Duetface, have less accuracy degradation but suffers from the disadvantage of tens of times higher communication and storage overhead. Most notably, none of the existing protection schemes in either the spatial or the frequency domain are able to provide privacy-preserving capabilities. In the case of providing outstanding privacy-preserving capability, our scheme has only a slight accuracy degradation compared to the baseline method Arcface-FD, and almost the same accuracy compared to the baseline method Arcface in the baseline, with minimal overhead.

5.3 Effectiveness of Privacy Protection (RQ2)

In this subsection, we will demonstrate the effectiveness of our scheme to defend against various attacks to protect privacy.



(a) DN-based attacks

(b) cGAN-based attacks

Figure 9: The facial images that are recovered by DN-based attacks (a) and cGAN-based attacks (b) from different face recognition schemes. Note that the reconstructed image is the same for all schemes except Advface, when the transmission channel leakage and the database leakage occur.

We compare FaceObfuscator's capability of privacy protection with the 2 baselines and the other 6 privacy-preserving methods by evaluating their performance against privacy attacks including 2 types of face reconstruction attacks (DN-based attacks and cGAN-based attacks), and their corresponding replay attacks in 2 different leakage scenarios (leakage from transmission channels and leakage from databases).

FaceObfuscator can effectively protect privacy in the scenario of transmission channel leakage. Protection schemes usually choose to protect the data prior to transmission to provide full-process protection capabilities, which also leads to the fact that an attacker can get the protected facial features and utilize adversarial training to obtain a mapping from the protected facial feature to the original image. Thus, once attackers capture the facial features transmitted in the transmission channel, they can use the trained generator to recover their corresponding original facial images.

We initially assessed the efficacy of safeguarding privacy against DL-based face reconstruction attacks. As for DN-based attacks, the channel leakage portion in Fig. 9a shows that the reconstruction images protected by our proposed FaceObfuscator are completely indistinguishable from each other, whereas the face details in the reconstruction images protected by the other privacy-preserving schemes are still clearly visible, leaking a large amount of facial information. As for cGAN-based attacks, as shown in the channel leakage portion of Fig. 9b, the original images of comparison schemes other than InstaHide [20] are completely recovered, while our scheme gets an average face rather than a face of a specific individual. Such an average face does not disclose personal privacy and is meaningless to the attacker.

Tab. 4 quantitatively shows that our method achieves higher MSE value, lower PSNR value, SSIM value, and COS value, which all indicate that our method achieves improved face privacy protection capabilities. Additionally, when attackers use reconstructed images to perform a replay attack, our scheme is considerably superior to the comparison schemes, where SRRA decreased from about 90% to about 0%, further proving the privacy protection capabilities of FaceObfuscator.

FreqObfuscator can effectively protect privacy in the scenario of database leakage. For most protection schemes, the server still stores the same protected features sent by the client instead of the "decrypted" features to cope with the requirement of achieving desensitized storage. Thus, when database leakage occurs, the protection effect of most schemes [20, 24, 39, 41, 52] is consistent with channel leakage portion in Figs. 9a and 9b.

However, Advface [53] achieved the aim of blocking adversarial training by only protecting facial features in the server. As shown in Table 4, although Advface [53] has some progress in privacy protection capability compared to other protection schemes, there is still a significant gap compared to our scheme quantitatively, especially in COS and SRRA. From the database leakage portion in Figs. 9a and 9b, we can see that the reconstructed image protected with advface may still reveal some outlines of faces, while the reconstructed image of our scheme reveals no private information at all.

FaceObfuscator can still protect privacy to some extent when facing white-box attacks. We also discuss scenarios of facing white-box attacks when the attacker knows all the details of the architecture and parameters of both the client and server. It can be seen from Fig. 10 that the defense effect of FaceObfuscator in the face of complete white-box attacks has declined slightly, but it is still far better than the defense effects of other solutions in the face of black-box attacks, as shown in Figs. 9a and 9b. It also indicates that service providers cannot reconstruct faces to violate privacy, even if they try to reconstruct facial images on the server. The reason that facial images can still be protected to some extent is twofold. First, the facial features only remain a small amount of frequency channels, retaining little visual information useful for reconstruction. Second, since the sign directions of all the elements in the features remain randomly obfuscated

Table 4: The performance of privacy protection methods in terms of MSE, PSNR, SSIM, COS against DN-based attacks, cGAN-based attacks and their corresponding replay attacks (SRRA), under 2 different leakage scenarios, i.e., leakage from transmission channels (C) and leakage from databases (D).

Metric	Method	Scenario	DN-based face reconstruction attack						cGAN-based face reconstruction attack					
			LFW	CFP-FF	CFP-FP	AgeDB-30	CALFW	CPLFW	LFW	CFP-FF	CFP-FP	AgeDB-30	CALFW	CPLFW
MSE↑	Arcface	C/D	0.002	0.008	0.022	0.001	0.002	0.002	0.004	0.013	0.027	0.002	0.003	0.003
	Arcface-FD	C/D	0.002	0.007	0.013	0.001	0.001	0.001	0.005	0.014	0.023	0.003	0.004	0.004
	InstaHide	C/D	0.025	0.040	0.052	0.046	0.033	0.033	0.026	0.043	0.052	0.047	0.031	0.031
	Cloak	C/D	0.007	0.018	0.025	0.008	0.008	0.007	0.009	0.019	0.025	0.009	0.009	0.009
	AdvFace	C	0.002	0.008	0.022	0.001	0.002	0.002	0.004	0.013	0.027	0.002	0.003	0.003
	AdvFace	D	0.047	0.072	0.090	0.048	0.049	0.051	0.064	0.090	0.106	0.068	0.069	0.070
	PPFR-FD	C/D	0.042	0.064	0.086	0.045	0.045	0.050	0.039	0.063	0.074	0.042	0.043	0.045
	DCTDP	C/D	0.041	0.063	0.076	0.044	0.045	0.048	0.045	0.063	0.076	0.046	0.048	0.049
	Duetface	C/D	0.040	0.062	0.082	0.041	0.043	0.046	0.038	0.057	0.065	0.040	0.041	0.042
	Ours	C/D	**0.057**	**0.098**	**0.115**	**0.074**	**0.068**	**0.067**	**0.073**	**0.106**	**0.120**	**0.090**	**0.083**	**0.083**
PSNR↓	Arcface	C/D	26.383	21.975	17.305	31.139	28.356	26.978	24.060	19.581	15.981	27.917	25.981	25.871
	Arcface-FD	C/D	27.368	22.266	19.329	31.671	29.228	28.961	23.980	19.361	16.685	26.722	25.044	24.958
	InstaHide	C/D	16.662	14.827	13.497	14.841	15.857	15.804	16.380	14.729	13.997	14.928	16.030	15.914
	Cloak	C/D	21.391	17.704	16.182	21.085	21.253	21.437	20.645	17.370	16.214	20.426	20.584	20.730
	AdvFace	C	26.383	21.975	17.305	31.139	28.356	26.978	24.060	19.581	15.981	27.917	25.981	25.871
	AdvFace	D	13.333	11.524	10.551	13.331	13.194	13.022	12.078	10.585	9.889	11.853	11.720	11.712
	PPFR-FD	C/D	14.101	12.298	11.120	13.966	13.958	13.658	14.722	12.256	11.577	14.350	14.478	14.261
	DCTDP	C/D	14.282	12.342	11.419	14.076	14.098	13.757	13.989	12.190	11.333	13.954	13.840	13.688
	Duetface	C/D	14.258	12.580	11.356	14.396	14.257	13.966	14.713	12.649	12.100	14.538	14.541	14.438
	Ours	C/D	**12.710**	**10.244**	**9.514**	**11.650**	**11.955**	**12.088**	**11.638**	**9.869**	**9.371**	**10.778**	**11.089**	**11.171**
SSIM↓	Arcface	C/D	0.968	0.852	0.672	0.974	0.971	0.964	0.923	0.785	0.639	0.956	0.946	0.944
	Arcface-FD	C/D	0.980	0.806	0.665	0.984	0.984	0.978	0.952	0.790	0.632	0.968	0.965	0.956
	InstaHide	C/D	0.703	0.562	0.456	0.595	0.677	0.673	0.697	0.575	0.475	0.602	0.681	0.689
	Cloak	C/D	0.832	0.702	0.591	0.824	0.833	0.831	0.843	0.720	0.602	0.829	0.838	0.845
	AdvFace	C	0.968	0.852	0.672	0.974	0.971	0.964	0.923	0.785	0.639	0.956	0.946	0.944
	AdvFace	D	0.424	0.363	0.321	0.415	0.414	0.403	0.395	0.322	0.287	0.380	0.381	0.378
	PPFR-FD	C/D	0.619	0.616	0.524	0.662	0.653	0.628	0.820	0.680	0.588	0.821	0.827	0.802
	DCTDP	C/D	0.710	0.662	0.569	0.727	0.732	0.692	0.800	0.677	0.579	0.804	0.802	0.782
	Duetface	C/D	0.677	0.653	0.560	0.723	0.716	0.677	0.819	0.687	0.592	0.825	0.827	0.801
	Ours	C/D	**0.471**	**0.313**	**0.279**	**0.412**	**0.434**	**0.479**	**0.391**	**0.261**	**0.248**	**0.340**	**0.364**	**0.405**
COS↓	Arcface	C/D	0.993	0.983	0.935	0.995	0.995	0.989	0.986	0.989	0.980	0.993	0.992	0.979
	Arcface-FD	C/D	0.993	0.989	0.979	0.991	0.993	0.990	0.979	0.983	0.964	0.985	0.987	0.970
	InstaHide	C/D	0.464	0.513	0.396	0.402	0.466	0.403	0.400	0.412	0.318	0.367	0.425	0.360
	Cloak	C/D	0.854	0.882	0.862	0.876	0.875	0.835	0.867	0.886	0.864	0.884	0.884	0.854
	AdvFace	C	0.993	0.983	0.935	0.995	0.995	0.989	0.986	0.989	0.980	0.993	0.992	0.979
	AdvFace	D	0.216	0.188	0.200	0.215	0.214	0.195	0.223	0.199	0.209	0.233	0.225	0.192
	PPFR-FD	C/D	0.629	0.710	0.682	0.685	0.671	0.619	0.915	0.922	0.906	0.938	0.932	0.889
	DCTDP	C/D	0.780	0.832	0.801	0.812	0.809	0.745	0.921	0.931	0.917	0.939	0.935	0.897
	Duetface	C/D	0.739	0.799	0.780	0.782	0.788	0.722	0.923	0.932	0.915	0.939	0.939	0.900
	Ours	C/D	**0.004**	**0.002**	**0.009**	**0.005**	**0.005**	**0.023**	**0.004**	**0.003**	**0.008**	**(0.0001)**	**(0.0004)**	**0.007**
SRRA↓	Arcface	C/D	99.37	97.63	94.54	95.90	91.67	85.93	99.50	99.60	96.83	96.40	92.27	86.27
	Arcface-FD	C/D	99.60	99.86	97.57	97.03	92.37	85.53	99.53	99.74	97.23	96.90	92.33	85.27
	InstaHide	C/D	74.43	65.57	75.94	46.87	53.80	53.93	53.73	39.60	58.91	37.10	38.10	42.30
	Cloak	C/D	99.10	99.11	93.71	93.60	89.80	80.50	97.87	98.14	93.06	92.73	88.23	80.23
	AdvFace	C	99.03	96.46	90.74	93.83	90.67	79.43	99.27	98.89	92.94	94.13	91.13	79.50
	AdvFace	D	30.93	22.57	53.14	34.17	30.67	42.67	13.17	11.20	40.57	20.37	16.17	16.17
	PPFR-FD	C/D	95.33	96.37	91.91	88.33	87.90	78.23	99.53	99.37	97.17	95.83	92.47	84.40
	DCTDP	C/D	98.30	98.60	96.37	94.06	90.40	82.10	99.53	99.51	97.71	96.33	92.27	84.17
	Duetface	C/D	96.83	98.71	93.20	91.37	90.13	81.20	99.50	99.69	96.29	96.20	92.63	85.57
	Ours	C/D	**0.00**	**0.00**	**1.31**	**0.13**	**0.06**	**6.00**	**0.03**	**0.03**	**1.31**	**0.07**	**0.03**	**6.20**

and unpredictable in the white-box scenario, it is difficult to recover the direction of elements to reverse the obfuscated features, which prevents the reconstruction of facial images.

Note that in realistic scenarios, we can use techniques such as code obfuscation to hide the operation of clients or redesign the candidate feature set of each feature to regain optimal protection.

Real-world Evaluation. Due to its practicability, we have submitted our method to the relevant authority at the university, which has carried out an evaluation of protection effectiveness and used it as one of the protection methods to protect the existing face data of over 100K individuals. The evaluation demonstrates that when the simulated attacker gets the facial features, they can neither obtain information about the

Table 5: Face recognition accuracy on the server side

Face recognition	LFW	IJB-B(TPR@FPR)	IJB-C(TPR@FPR)
w/o ID Recovery	50.00	0.01 / 0.00	0.02 / 0.00
ID Recovery	99.70	93.14 / 88.68	94.82 / 92.42

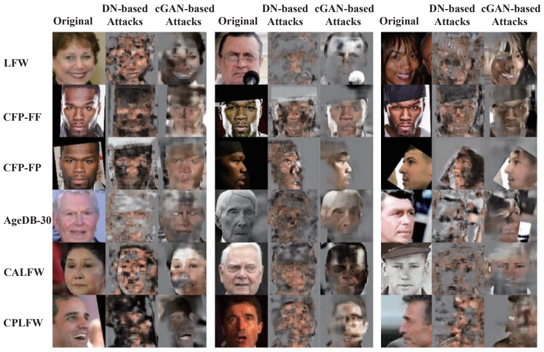


Figure 10: Reconstructed images generated by DN-based and cGAN-based from our scheme, when attackers know all the details of our scheme (white-box attack).

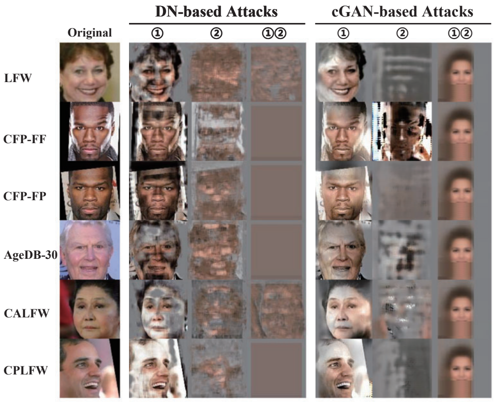


Figure 11: Reconstructed images generated by DN-based and cGAN-based from our scheme with different components in the client. ① represents frequency domain-based visual information deletion, ② represents identity-retained stochastic obfuscation.

original real-world faces nor use them to pass authentication systems. The quantitative results show that when exposed to DN-based attacks, the MSE, PSNR, and SSIM values are 0.084, 11.202, and 0.507, respectively. Similarly, when facing cGAN-based attacks, the corresponding values are 0.085, 11.285, and 0.493, achieving similar reconstruction results to Tab. 4. Furthermore, the COS between the original images and the images reconstructed from the facial features protected by FaceObfuscator are 0.201 and 0.227 against DN-based attacks and cGAN-based attacks respectively, far below the threshold that can be used for face reconstruction.

5.4 Ablation Study (RQ3)

In this subsection, we will demonstrate how each component on the client side and the server side contributes to the overall improvement of accuracy and privacy.

Frequency domain-based visual information deletion and identity-retained stochastic obfuscation on client side both contribute to privacy preservation. On the client side, we perform evaluations against face reconstruction attacks after the permutation of visual information deletion and stochastic obfuscation. From Fig. 11, we can see that visual information deletion reduces the visual information in reconstructed facial images, and stochastic obfuscation makes the reconstructed images almost unrecognized. Moreover, when they are used in combination, they provide perfect protection.

Identity recovery based on traceability of candidate feature set on the server side is essential to face recognition. On the server side, identity recovery based on the traceability of the candidate feature set is designed to maintain face recognition accuracy. To demonstrate its efficiency, we will perform face recognition without it. Table 5 shows that the accuracy of face recognition is right at 50% under datasets LFW, and the accuracy drops to near 0% under IJB after removing it. This means identity recovery based on the traceability of the candidate feature set is crucial for face recognition, without which face recognition simply cannot work.

6 Conclusions

In this work, we revealed that existing privacy-preserving face recognition schemes cannot defend against DL-based face reconstruction attacks. Then, we gave a new understanding, i.e., most frequency channels can be removed without compromising face recognition accuracy, which allows us to achieve privacy protection with lower storage costs and time costs. Eventually, we proposed, FaceObfuscator, a lightweight privacy-preserving face recognition system, that generates obfuscated features to achieve gradient descent-resistant against face reconstruction while maintaining the accuracy of the face recognition. Extensive experiments showed that FaceObfuscator significantly outperforms state-of-the-art methods in terms of privacy protection (90% improvement) with negligible 0.3% accuracy degradation.

Acknowledgments

This work was supported by National Key R&D Program of China (Grant No. 2021ZD0112803), National Natural Science Foundation of China (Grants No. 62122066, U20A20182) and Key R&D Program of Zhejiang (Grant No. 2022C01018, 2024C01164).

References

[1] https://gdpr-info.eu/issues/personal-data/.

[2] https://www.clearview.ai/.

[3] https://pursuit.unimelb.edu.au/articles/tiktok-captures-your-face.

[4] https://www.nytimes.com/2020/01/29/technology/facebook-privacy-lawsuit-earnings.html.

[5] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. *Cryptology ePrint Archive*, 2015.

[6] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.

[7] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pages 274–283, 2018.

[8] 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.

[9] Forrester Cole, David Belanger, Dilip Krishnan, Aaron Sarna, Inbar Mosseri, and William T Freeman. Synthesizing normalized faces from facial identity features. In *Proceedings of CVPR*, pages 3703–3712, 2017.

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of CVPR*, pages 4690–4699, 2019.

[11] Samuel Felipe dos Santos and Jurandy Almeida. Less is more: Accelerating faster neural networks straight from jpeg. In *CIARP*, pages 237–247, 2021.

[12] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *Proceedings of CVPR*, pages 4829–4837, 2016.

[13] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM CCS*, pages 1322–1333, 2015.

[14] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *EUROCRYPT 2011*, pages 129–148, 2011.

[15] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV 2016: 14th European Conference, Proceedings, Part III 14*, pages 87–102, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778, 2016.

[17] Xinlei He and Yang Zhang. Quantifying and mitigating privacy risks of contrastive learning. In *Proceedings of the 2021 ACM CCS*, pages 845–863, 2021.

[18] Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th ACSAC*, pages 148–162, 2019.

[19] 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.

[20] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. Instahide: Instance-hiding schemes for private distributed learning. In *ICML*, pages 4507–4518, 2020.

[21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of CVPR*, pages 1125–1134, 2017.

[23] Shubham Jain, Ana-Maria Crețu, and Yves-Alexandre de Montjoye. Adversarial detection avoidance attacks: Evaluating the robustness of perceptual hashing-based client-side scanning. In *31st USENIX Security*, pages 2317–2334, 2022.

[24] 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. In *ECCV 2022: 17th European Conference, Proceedings, Part XII*, pages 475–491, 2022.

[25] Don H Johnson. Signal-to-noise ratio. *Scholarpedia*, 1(12):2088, 2006.

[26] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. {GAZELLE}: A low latency framework for secure neural network inference. In *27th USENIX Security*, pages 1651–1669, 2018.

[27] William Kahan. Ieee standard 754 for binary floating-point arithmetic. *Lecture Notes on the Status of IEEE*, 754(94720-1776):11, 1996.

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of CVPR*, pages 4401–4410, 2019.

[29] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of CVPR*, pages 1931–1939, 2015.

[30] Xiaoyu Kou, Ziling Zhang, Yuelei Zhang, and Linlin Li. Efficient and privacy-preserving distributed face recognition scheme via facenet. In *ACM TURC*, pages 110–115, 2021.

[31] Huiying Li, Shawn Shan, Emily Wenger, Jiayun Zhang, Haitao Zheng, and Ben Y Zhao. Blacklight: Scalable defense for neural networks against {Query-Based}{Black-Box} attacks. In *31st USENIX Security*, pages 2117–2134, 2022.

[32] Jian Liu, Mika Juuti, Yao Lu, and Nadarajah Asokan. Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM CCS*, pages 619–631, 2017.

[33] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

[34] Guangcan Mai, Kai Cao, Xiangyuan Lan, and Pong C Yuen. Secureface: Face template protection. *IEEE TIFS*, 16:262–277, 2020.

[35] Guangcan Mai, Kai Cao, Pong C Yuen, and Anil K Jain. On the reconstruction of face images from deep face templates. *TPAMI*, 41(5):1188–1202, 2018.

[36] Peter Mattson, Vijay Janapa Reddi, Christine Cheng, Cody Coleman, Greg Diamos, David Kanter, Paulius Micikevicius, David Patterson, Guenther Schmuelling, Hanlin Tang, et al. Mlperf: An industry standard benchmark suite for machine learning performance. *IEEE Micro*, 40(2):8–16, 2020.

[37] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *2018 ICB*, pages 158–165, 2018.

[38] Shagufta Mehnaz, Sayanton V Dibbo, Roberta De Viti, Ehsanul Kabir, Björn B Brandenburg, Stefan Mangard, Ninghui Li, Elisa Bertino, Michael Backes, Emiliano De Cristofaro, et al. Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models. In *31st USENIX Security 22*, pages 4579–4596, 2022.

[39] 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 MM*, pages 6755–6764, 2022.

[40] Alexis Mignon and Frédéric Jurie. Reconstructing faces from their signatures using rbf regression. In *BMVC 2013*, pages 103–1, 2013.

[41] Fatemehsadat Mireshghallah, Mohammadkazem Taram, Ali Jalali, Ahmed Taha Taha Elthakeb, Dean Tullsen, and Hadi Esmaeilzadeh. Not all features are equal: Discovering essential features for preserving prediction privacy. In *Proceedings of the Web Conference 2021*, pages 669–680, 2021.

[42] Pratyush Mishra, Ryan Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, and Raluca Ada Popa. Delphi: a cryptographic inference service for neural networks. In *29th USENIX Security*, pages 2505–2522, 2020.

[43] Pranab Mohanty, Sudeep Sarkar, and Rangachar Kasturi. From scores to face templates: A model-based approach. *TPAMI*, 29(12):2065–2078, 2007.

[44] 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 CVPR workshops*, pages 51–59, 2017.

[45] Anton Razzhigaev, Klim Kireev, Edgar Kaziakhmedov, Nurislam Tursynbek, and Aleksandr Petiushko. Black-box face recovery from identity features. In *ECCV 2020 Workshops: Proceedings, Part V 16*, pages 462–475, 2020.

[46] Anton Razzhigaev, Klim Kireev, Igor Udovichenko, and Aleksandr Petiushko. Darker than black-box: Face reconstruction from similarity queries. *arXiv preprint arXiv:2106.14290*, 2021.

[47] Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou, et al. Mlperf inference benchmark. In *2020 ACM/IEEE 47th ISCA*, pages 446–459, 2020.

[48] 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 WACV*, pages 1–9, 2016.

[49] Shawn Shan, Wenxin Ding, Emily Wenger, Haitao Zheng, and Ben Y Zhao. Post-breach recovery: Protection against white-box adversarial examples for leaked dnn models. In *Proceedings of the 2022 ACM CCS*, pages 2611–2625, 2022.

[50] Sameer Wagh, Shruti Tople, Fabrice Benhamouda, Eyal Kushilevitz, Prateek Mittal, and Tal Rabin. Falcon: Honest-majority maliciously secure framework for private deep learning. *arXiv preprint arXiv:2004.02229*, 2020.

[51] Gregory K Wallace. The jpeg still picture compression standard. *IEEE TCE*, 38(1):xviii–xxxiv, 1992.

[52] Yinggui Wang, Jian Liu, Man Luo, Le Yang, and Li Wang. Privacy-preserving face recognition in the frequency domain. In *Proceedings of AAAI*, volume 36, pages 2558–2566, 2022.

[53] Zhibo Wang, He Wang, Shuaifan Jin, Wenwen Zhang, Jiahui Hu, Yan Wang, Peng Sun, Wei Yuan, Kaixin Liu, and Kui Ren. Privacy-preserving adversarial facial features. In *Proceedings of CVPR*, pages 8212–8221, 2023.

[54] 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.

[55] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *proceedings of CVPR workshops*, pages 90–98, 2017.

[56] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *Proceedings of CVPR*, pages 1740–1749, 2020.

[57] Ziqi Yang, Ee-Chien Chang, and Zhenkai Liang. Adversarial neural network inversion via auxiliary knowledge alignment. *arXiv preprint arXiv:1902.08552*, 2019.

[58] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of CVPR*, pages 253–261, 2020.

[59] Tianyue Zheng and Weihong Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5(7), 2018.

[60] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017.

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

A Evaluating the importance of different frequency channels for face recognition

Algorithm 3: Analysis of Channels' Importance

```
input  : Facial features $f$ of size $m \times n \times c$
output : Weight $\alpha_i$ of different channels
Initializing;
learnable weight $\alpha_i = 1$;
for $i \leftarrow 1$ to $c$ do
    if input is in RGB mode then
        $f_i = [Y_i, Cb_i, Cr_i]$;
        new feature $\leftarrow [\alpha_i Y_i, \alpha_i Cb_i, \alpha_i Cr_i]$;
    else if input is in Gray mode then
        new feature $\leftarrow \alpha_i \times f_i$;
Training;
for $j \leftarrow 0$ to $N$ steps do
    loss $\leftarrow$ Network(new feature);
    Update Network $\leftarrow$ loss;
    Update $\alpha_i$ $\leftarrow$ loss
```

This section shows the design of the auxiliary network in Section 4.2.2 and further elaborates on the necessity of standardization for evaluating the importance of frequency channels.

As shown in algorithm 3, we analyzed the importance of face recognition directly through the auxiliary network and found that the low-frequency channels play a much more important role than the high-frequency channels, shown in Fig. 12a, which is similar to HVS. However, as we systematically eliminated channels of higher importance, the roles of the channels with less importance previously became gradually apparent without significantly influencing the accuracy of face recognition, shown in Figs. 12b and 12c, which indicates the low-frequency channels inhibit the effect of the high-frequency channels on the face recognition network. Upon deeper analysis of the causes of the inhibition phenomenon, we note there is a large difference in the order of magnitude of the different frequency domain channels. Consequently, we hypothesized that it is the numerical magnitude of different frequency domain channels that affects the expression of importance. To validate this hypothesis, we carried out a reassessment after standardizing the numerical values of each channel to the same level. The results in Fig. 12d demonstrate

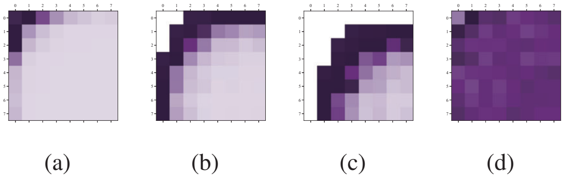


Figure 12: The importance of frequency channels for face recognition (the deeper the color is, the higher the importance is). (a) is the importance of frequency channels before standardizing the numerical values in frequency channels; (b) (c) are the re-assessed importance of frequency channels after removing the significant channels; (d) is the re-assessed importance of frequency channels after standardizing the numerical values of each channel.

the difference between different channels is not as disparate as the conclusions of previous works [24, 39, 52, 56]. Furthermore, experiments shown in Table 2, demonstrating that face recognition can be performed on a single channel regardless of high frequency or low frequency, is a strong proof of the correctness of our analysis.

B Self-normalization

To prevent reverse analysis of the numerical transformation process, we first need to make the facial features irreversible to the exact original values but still available for training neural networks. A practical approach is to normalize the data. However, the most prevalent normalization operation in deep learning, i.e., batch normalization [21], is sensitive to outliers in the batch, which normalizes the data by values in the entire batch. Such outliers are common in facial images due to factors such as shooting angle, lighting, and background. [52] tried to address this problem and enhance security by calculating the mean and variance of each image independently and normalizing each image separately. Nonetheless, this approach falls short when faced with substantial differences between different channels within a feature, compromising the maintenance of the feature's original characteristics.

Consequently, we design Self-Normalization to make each feature retain its own characteristics at the initial transformation, shown in line 3, where a represents the output of Self-Normalization, f_k represents the k^{th} channel in facial feature f, $\mathrm{E}(\cdot)$ is the mean of the elements, $\mathrm{Var}(\cdot)$ is the variance of the elements with Bessel's correction and β is a bias term to prevent infinity. In other words, each channel of each image in a batch has its own mean and variance, so normalization can be performed independently on channels.

C Details of datasets

MS-Celeb-1M [15] is a dataset consisting of 10M images of faces collected from the Internet. In this paper, we choose its third version, i.e., MS1Mv3 as the train set, which contains approximately 93K classes with more than 5.1M images. **CelebA [33]** is CelebFaces Attributes dataset, which contains 202,599 facial images from more than 10K identities. **LFW [19]** is a public benchmark for face verification, containing more than 13K images and 6K pairs to verify. **CFP-FF [48]** only compares the frontal faces of Celebrities in Frontal-Profile in the Wild, which contains 7K pairs to verify. **CFP-FP [48]** compares the frontal and profile faces of Celebrities in Frontal-Profile in the Wild, which also contains 7K pairs to verify. **AgeDB-30 [44]**, including 6K pairs, is the most challenging version with an age gap of 30 years in AgeDB. **CALFW [60]** is a renovation of LFW, which selects 3K positive face pairs with age gaps to add the aging process intra-class variance. **CPLFW [59]** is also a renovation of LFW, which selects 3K positive face pairs with pose difference to add pose variation to intra-class variance. **IJB-B [55]** contains 1845 subjects with 11,754 images, which are collected from the Internet and are totally unconstrained, with large variations in pose, illumination, image quality, etc. **IJB-C [37]** is an extension of the IJB-A [29] dataset with about 138K facial images.

Table 6: Face recognition under different values of b.

Value of b	Range of u and v	LFW	AgeDB-30	CALFW	CPLFW
1.8	$[-60, 60]$	99.62	97.73	96.00	91.38
2	$[-60, 60]$	99.68	97.65	95.83	91.33
10	$[-18, 18]$	99.65	97.33	95.77	90.97
100	$[-9, 9]$	99.67	97.53	96.05	90.92

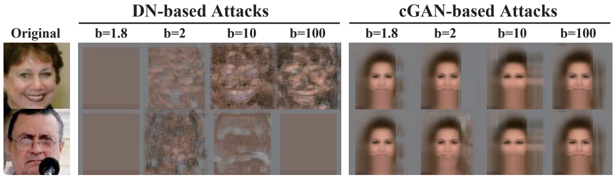


Figure 13: The facial images that are recovered by DN-based attacks and cGAN-based attacks with different values of b.

D The effect of FaceObfuscator under different values of b

To discuss the numerical stability concerns regarding other values of b, we evaluate the visual reconstruction and face recognition by taking values of b as 1.8, 2, 10, and 100. Additionally, due to the limitations in the representation of floating-point numbers mentioned in Section 4.3.2, different b correspond to different ranges of u and v. As shown in Table 6, the face recognition accuracy under different values of b is similar, proving that the value of b does not influence the stability of facial recognition. As shown in Fig. 13, all the reconstructed images are unrecognizable, indicating that different values of b also do not affect the effectiveness of privacy protection.