Crafter: Facial Feature Crafting against Inversion-based Identity Theft on Deep Models

Shiming Wang[1], Zhe Ji[1], Liyao Xiang[1†], Hao Zhang[1], Xinbing Wang[1], Chenghu Zhou[2], Bo Li[3]
[1]Shanghai Jiao Tong University, [2]Chinese Academy of Science, [3]Hong Kong University of Science and Technology
[1]{my16wsm, ji_zhe, xiangliyao08, mypeach, xwang8}@sjtu.edu.cn, [2]zhouchsjtu@gmail.com, [3]bli@cse.ust.hk

***Abstract*—With the increased capabilities at the edge (e.g., mobile device) and more stringent privacy requirement, it becomes a recent trend for deep learning-enabled applications to pre-process sensitive raw data at the edge and transmit the features to the backend cloud for further processing. A typical application is to run machine learning (ML) services on facial images collected from different individuals. To prevent identity theft, conventional methods commonly rely on an adversarial game-based approach to shed the identity information from the feature. However, such methods can not defend against adaptive attacks, in which an attacker takes a countermove against a known defence strategy.**

We propose Crafter, a feature crafting mechanism deployed at the edge, to protect the identity information from adaptive model inversion attacks while ensuring the ML tasks are properly carried out in the cloud. The key defence strategy is to mislead the attacker to a non-private prior from which the attacker gains little about the private identity. In this case, the crafted features act like poison training samples for attackers with adaptive model updates. Experimental results indicate that Crafter successfully defends both basic and possible adaptive attacks, which can not be achieved by state-of-the-art adversarial game-based methods.

I. Introduction

Deep learning demonstrates impressive performance in many applications, owing to the complicated structures of learning models and massive crowdsourced data. Since local processing is often infeasible, the edge, e.g., mobile devices, collect the sensitive individual data, encode and transmit it to the untrusted cloud for further processing by learning models. Facial image data raises the most concern as it is highly sensitive and susceptible to identity theft. Hence it is a critical issue to remove the sensitive identity information from the encoded features while accomplishing cloud learning tasks. Examples could be makeup recommendations based on users' facial attributes or training facial expression detection model on crowdsourced images, where identity information needs to be protected while preserving useful features.

Inversion attacks [22], [38], [14] can invert the private input pixel by pixel from the features, leading to identity leakage. The perception of identity involves not only small reconstruction distortion but also high-level semantic information. Thereby we propose *identity perceptual privacy against inversion attacks*, which is more complicated than reconstruction distortion-based defence [37] and attribute inference-based defence [11], [28], [17], [33], [18]. For instance, although [11], [28] prevent recognition models from inferring the identity attribute, their imperceptible perturbation fails to evade visual detection and still compromises privacy at the image level.

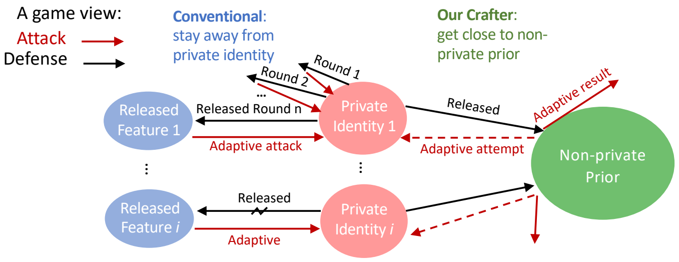


Fig. 1. Conventional methods adopt a stay-away approach where the defence strategy is easily overwhelmed by an adaptive attacker step; our Crafter takes a get-close approach where the crafted features act like poison training samples to the adversary, disrupting the training of adaptive attackers.

This motivates us to consider feature manipulation at the edge, so that transmitted features maintain high utility for the cloud ML task while protecting identity perceptual privacy. A naive solution is to train the learning model end to end to shed the identity information from the feature, which inevitably introduces a race between the defence party and the adversary [37], [17], [33], [18], [29]. In the race, the defence party pushes the feature away from the regime of private identity perception as in the left part of Fig 1. The attacker could almost always overwhelm the defence party since the defence strategy is fixed and known upon the feature release, especially in the presence of adaptive attacks.

We aim to overcome this seemingly endless tit-for-tat between the defence party and the adversary by leveraging a non-private prior to bound the adversary's perceptual gain on private identities. Since the defender's strategy is fixed, its optimal move is to stay close to the adversary's prior (or a non-private prior in practice), thereby limiting the attacker's gain from the defender's move. As shown in the right part of Fig 1, the adaptive attacker takes a countermove by mapping the non-private prior to different and independent identities, which disrupts the potential adaptive training of the attacker model. In addition, we assume the defender plays against a worst-case adversary, *i.e.,* a white-box attacker that obtains not only the defender's strategy but also all model weights to derive its move. If such an omniscient attacker can be defended, we

[†] Corresponding author.

Network and Distributed System Security (NDSS) Symposium 2024
26 February - 1 March 2024, San Diego, CA, USA
ISBN 1-891562-93-2
https://dx.doi.org/10.14722/ndss.2024.23326
www.ndss-symposium.org

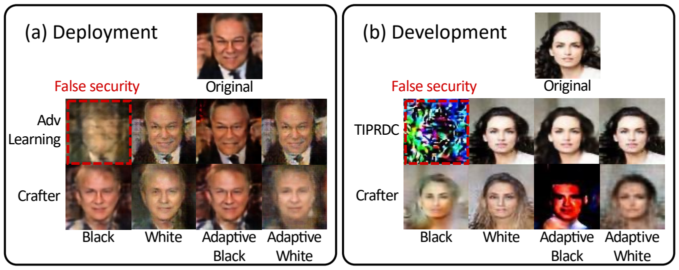


Fig. 2. Inversion results of existing defences and Crafter against multiple attacks. Crafter demonstrates robustness against both basic and possible adaptive attacks, while the baselines are not adaptive attacker-proof.

have reasons to believe that the protection scheme is robust against other real-world attackers.

To this end, we propose Crafter, a facial feature crafting mechanism against inversion attacks on deep learning models. Given the original model, the framework perturbs the intermediate features to trick attackers into reconstructing non-private facial images while keeping the perturbation under a threshold to accomplish cloud tasks with high accuracies. What distinguishes Crafter from previous work is that it chooses not to *erase* [25], [17], nor to *obfuscate* [12], [10] the concerned private attribute in feature representations, but to *draw* the feature *close to* a non-private prior perceptually. The crafted features act as poison training samples to the inversion attacker, as they are close to the original features but drastically different in identity perception on the reconstructed image space. Since neither image level distortion nor attribute level accuracy alone is sufficient to quantify Crafter's identity perceptual privacy, we propose a holistic privacy index, *perceptual inversion indistinguishability*, as a distributional distance from the inversion attacker's prior to posterior view on the reconstructed images. We show through analysis and experiments that it is a valid privacy notion for Crafter's defence on facial images.

Unlike the adversarial game-based methods, we demonstrate through analysis and experiments that Crafter consistently prevails over adaptive attackers that are specifically designed for Crafter. In addition, Crafter is able to decouple from the cloud learning tasks by taking advantage of the high-dimensional feature space and the robustness of deep models against minor input perturbations. It allows the feature to achieve the privacy goal with slight distortion, therefore not affecting the cloud inference or training performance. As the feature is expressed implicitly in our objective, we resort to the implicit function theorem to resolve the optimization challenge. Experimental results under various settings show that Crafter successfully defends black- and white-box attacks and their adaptive versions (Figure 2), outperforming the state-of-the-art yet fulfilling the cloud tasks with high accuracies.

In summary, our key contributions are as follows:

- We propose Crafter, a facial feature crafting approach that prevents identity leakage through inversion attacks, and is robust against possible adaptive attacks.
- We formulate the privacy of interest with perceptual inversion indistinguishability, a distributional distance between the attacker's posterior and prior beliefs on the reconstructed image space, and show that Crafter achieves approximately optimal privacy-utility tradeoff.
- Crafter is open-sourced and easy to deploy as a plug-in to the edge-cloud computing framework, without any change in the backend models. **Code is available in the repository [3].**

II. Preliminaries

The Earth Mover's distance (EMD) is a classic measure of inter-distribution distance, defined as $\mathrm{EMD}(P||Q) = \inf_{\gamma\in\Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$. P, Q denote distributions and $\Pi(P,Q)$ is the set of all joint distributions whose marginals are P and Q. The infimum of the expectation is easy to compute for discrete tabular data, but it is intractable to traverse all joints of high-dimensional image distributions or continuous feature distributions. Hence we leverage its dual form:

Definition 1 (KR duality of the Earth Mover's distance)**.** *For distributions P and Q, and a 1-Lipschitz continuous function f, the EMD between the distributions is*

$$\mathrm{EMD}(P||Q) = \sup_{||f||_L\le 1} \mathbb{E}_{x\sim P}[f(x)] - \mathbb{E}_{x\sim Q}[f(x)]. \quad (1)$$

In practice, we optimize a discriminator network D to approximate the supremum on function f. This is a common practice in line with works on Wasserstein-GAN [6], [15]. To encourage D to be 1-Lipschitz, i.e. D has gradient with norm at most 1 everywhere, Gulrajani et al. [15] enforce a gradient penalty term g_p on the norm and adds it to the original EMD as a soft constraint: $g_p = \mathop{\mathbb{E}}_{\hat{x}\sim P_{\hat{x}}}\left[(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1)^2\right]$, and $\hat{x} = \epsilon x_1 + (1-\epsilon)x_2 \sim P_{\hat{x}}$ is an interpolation of the two distributions where $x_1 \sim P$ and $x_2 \sim Q$.

We further show the key lemmas in Implicit Differentiation.

Lemma 1 (Cauchy, Implicit Function Theorem)**.** *For a function $f(x,y) : \mathbb{R}^{n+m} \to \mathbb{R}^m$, if some (a,b) satisfies 1) $f(a,b) = 0$ and 2) the Jacobian matrix $J_{f,y}(a,b) = \left[\frac{\partial f_i}{\partial y_j}(a,b)\right]$ is invertible, then surrounding (a,b) there exist $U \subset \mathbb{R}^n$ and a unique continuously differentiable function $g : U \to \mathbb{R}^m$ that $g(a) = b$ and $f(x,g(x)) = 0, \forall x \in U$. In addition, $\frac{\partial g}{x}(x) = -\left[J_{f,\mathbf{y}}(x,g(x))\right]^{-1}\left[\frac{\partial f}{\partial x}(x,g(x))\right]$.*

Lemma 2 (Lorraine [20], Neumann Inverse Approximation)**.** *For a matrix A and a sufficiently small scale α s.t. $|I - \alpha A| < 1$, the following series holds and converges: $A^{-1} = \alpha \lim_{i\to\infty} \sum_{j=0}^{i} [I-\alpha A]^j$.*

III. Problem Setting and Threat Model

A. Problem Setting

We focus on the typical edge cloud computing scenario, where the cloud provides a crowdsourcing service that requires facial data from users to perform a joint machine learning (ML) task. Any raw data or feature transmission that potentially exposes user identity is forbidden due to privacy concerns, and the cloud is responsible for providing a data collecting service to protect user privacy while accomplishing the crowdsourcing tasks. We present two concrete examples for both *model deployment* and *model development* tasks to facilitate understanding. For model deployment, the service runs a trained facial attribute-based makeup recommender to which users upload their face shots. The recommeder analyzes

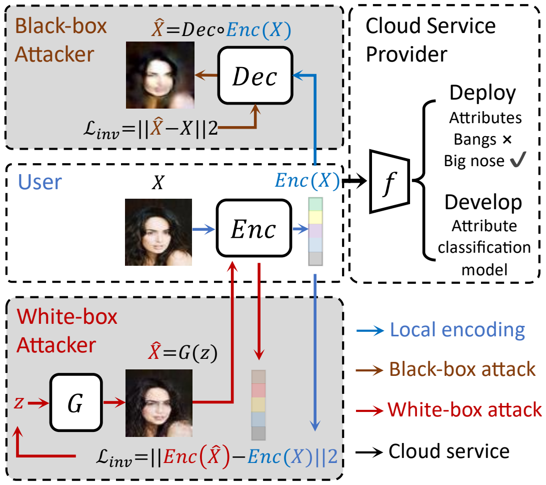


Fig. 3. Problem overview. Users (blue) release locally encoded feature $Enc(X)$ of private image X to complete computation tasks (black). Attackers intercept the released feature and attempt to reconstruct original private input through either black-box attack (brown) or white-box attack (red).

the customers' facial characteristics but should not record their identities. For model development, the service collects facial images from volunteers to train a facial expression detection model without distinguishing their identities.

To hide the identity information, the service deploys a local pre-processor to users and allow them to encode their images into features before transmission. An encoder Enc extracts features from private images $X \in \mathcal{X}_{\text{pvt}}$ and sends the features to the downstream computation task denoted by f. We formally show the problem setting in the lighted area of Figure 3. The pipeline consists of an *offline* and an *online* stage involving three parties: *user*, *cloud service provider*, and *adversary*. In the deployment scenario, the service trains $Enc \circ f$ end to end offline, and distributes Enc to the user while keeping f at the cloud. Hence Enc and f are fixed beforehand, and features $Enc(X)$ are fed into f for online prediction tasks. In the development scenario, the service releases Enc as a general feature extractor to the user offline and collects $Enc(X)$, on which it runs the online training task for model f which is not fixed apriori.

B. Threat Model

Unfortunately, transmitting features still faces serious privacy risks. We focus on non-targeted feature inversion attacks aiming to recover raw faces of unknown identities from the features. An inside-attacker in the untrusted cloud or a man-in-the-middle attacker receives the locally encoded raw feature $Enc(X)$ and reconstructs image $\hat{X}$. If $\hat{X}$ highly resembles the original face X as shown in Fig 3, the adversary would acquire the appearance of the unknown user which clearly leaks user privacy: the adversary may link to an external database and decode the user's identity.

Our threat model is different from those in certain existing defenses for user privacy. Instead of concrete attributes such as hair style and nose shape considered in [25], [17], [29], our adversary takes interest in user's identity, which is a semantic information particularly hard to isolate and remove from images. In addition, we target inversion attacks that aim to unveil the initially undisclosed visage of a victim. We do not consider membership inference attacks as in [31], or attribute inference attacks as in [11], [28] where the adversary already obtains some of the victim's private facial images and attempts to infer the identity of its other faces with recognition models. Detailed clarifications are in our online Appendix A-A [3]. To highlight this, we formally define the privacy of interest as *identity perceptual privacy against inversion attack*. We omit "against inversion attack" in the remainder of the paper for brevity.

Identity perceptual privacy describes the extent to which an inverted image is perceived as the true private identity by an attacker, similar to a human observer's interpretation. Obviously, this perceptual privacy cannot be simply characterized by the pixel-level distortion between $\hat{X}$ and X (e.g., SSIM), or the attribute inference accuracy (e.g., whether a recognition model extracts the correct ID from $\hat{X}$). For example, if $\hat{X}$ and X are two images of the same identity taken under different light and angles, their SSIM is low but the recognition accuracy is still high; conversely, $\hat{X}$ can be visually similar to X while evading facial recognition models, i.e. low accuracy but high SSIM. Typically, a high level of identity perceptual privacy suggests both large image-wise distortion and low accuracy of facial recognition models. To specify how an inversion attacker perceives an unknown identity, we instantiate reconstruction attacks with representative methods as in Fig 3, categorized into basic and adaptive attacks according to whether an adversary adjusts its strategy to a given defence.

Basic attacks are generally categorized as either black-box or white-box. A *black-box adversary* queries the user's local encoder Enc for unlimited times with images of public identities crawled from the Internet , denoted as $\mathcal{X}_{\text{pub}}$. It constructs a shadow decoder Dec as the inverse of Enc and trains the decoder as below [23]:

$$\min_{Dec} \mathbb{E}_{X' \in \mathcal{X}_{\text{pub}}} \|Dec(Enc(X')) - X'\|_2. \tag{2}$$

We denote the reconstructed image as $\hat{X}^* = Dec(Enc(X))$.

In contrast, a *white-box adversary* has unrestricted access to Enc and its parameters. The adversary first trains a Wasserstein-GAN to distill public prior knowledge of general facial images from public datasets $\mathcal{X}_{\text{pub}}$ [38]. Given latent vectors z_r following random distributions, the pretrained generator G is able to generate realistic-looking facial images with no particular private identity, referred to as *average faces*. Initiated from this public prior, the adversary reconstructs a target image via gradient-based optimization on latent representation z, starting from some random z_0:

$$z^* = \arg\min_{z} \|Enc(X) - Enc \circ G(z)\|_2, \tag{3}$$

and the reconstructed image is $\hat{X}^* = G(z^*)$. We refer to it as a canonical white-box attacker, and z^* as the *best-response* latent representation of image X's feature. To avoid confusion, we use z_{org} to denote the best-response of the raw feature $Enc(X)$. An et. al [5] further propose a more advanced white-box reconstruction with potentially higher fidelity using StyleGAN instead of the canonical WGAN.

Alternatively, a white-box adversary can initialize its optimization with a black-box decoder, referred to as a *hybrid* *white-box adversary*. Upon receiving $Enc(X)$, the attacker initializes the image as output of the pre-trained black-box Dec and runs pixel-level optimization to minimize feature loss:

$$\min_{\hat{X}} \|Enc(X) - Enc(\hat{X})\|_2, \ \hat{X}_{\text{init}} = Dec(Enc(X)). \quad (4)$$

Notice that white-box access is a realistic assumption in our scenario. As discussed in §III-A, the cloud distributes Enc to users. Thus an inside attacker naturally has white-box access to Enc, and a man-in-the-middle adversary can disguise itself as a benign user and acquire the parameters.

Adaptive attacks. Apart from the basic attacks discussed above, adversaries can specifically adjust their strategies to target a given defense. As an example, the adversary may update Dec and G with protected features F_X. Intuitively, such adaptive attacks can be stronger as they leverage the protection strategy and try to bypass it. We consider all possible adaptive approaches under our edge cloud scenario. Detailed definitions of the adaptive attacks are deferred to §V-A, after we introduce our protection mechanism.

Attacker's knowledge. We assume the attackers, basic or adaptive, have access to any public datasets crawled from the Internet, and any Dec and G models. There is no constraint on the adversary's reconstruction approach. The attacker is free to choose between decoder-based (Dec), GAN-based (G) and more advanced StyleGAN-based methods. The adversary is assumed to have no access to the private images of the unknown identity, i.e., $\mathcal{X}_{\text{pub}}$ and $\mathcal{X}_{\text{pvt}}$ has no identity overlapping. Intuitively, if the adversary already acquires multiple faces of a victim, the harm is already done and it is meaningless to prevent another image from exposure. Hence the adversary cannot launch an attribute inference attack on the features directly: it can only train the identity classifier on $\mathcal{X}_{\text{pub}}$ which does not effectively recognize the identities of $\mathcal{X}_{\text{pvt}}$. We will further extend the attacker's capability in §VI-F.

Defender's knowledge. We assume the defender has knowledge of the white-box attacker's reconstruction loss function $\mathcal{L}_{\text{inv}}$, which is the L_2 feature distortion between the original and the reconstructed as in most inversion attacks. It is later verified that our defence using $\mathcal{L}_{\text{inv}}$ is also effective against black-box and other attacks. The defender also has full access to a trained G provided by trusted third parties, which can be any open platform offering widely-acknowledged pretrained generators. To be noted, the defender knows nothing about the adversary's reconstruction model, indicating the adversary does not necessarily use G in its attack.

IV. Methodology

In face of the privacy threats, the cloud provides a user encoder to let users craft features with the following goals:

- *Privacy*: given features, the attacker reconstructs images that reveal little private identity information.
- *Utility*: the crafted features should complete the subsequent ML tasks with high performance.
- *Robust against adaptive attacks*: once features are released, attackers cannot bypass the defense via adaptive updates.

Towards these goals, we illustrate our design choices in §IV-A and our approach of privacy-preserving feature representation construction in §IV-B. The key idea is to iteratively craft a feature F_X to bring the adversary's posterior close to a non-private prior (minimizing $\mathcal{L}_p$) while restricting feature perturbation (minimizing $\mathcal{L}_u$). To capture the privacy leakage, we give a theoretical privacy guarantee for our method in §IV-D. Finally, we discuss the advantage of Crafter in real-world cases in §V. Notations used are listed in Table I.

TABLE I. Notations.

Notation	**Definition**
	DNN models:
Enc	The local encoder under attack.
G	Pretrained generator; input: latent vectors; output: images.
D	Discriminator that attempts to distinguish reconstructed images from attacker's prior.
	Variables:
X	Private input images.
F_X	Protected feature representation of X.
z_r	Latent random vector.
z_{org}	Best-response latent vector of raw feature $Enc(X)$.
$z^*(F_X)$	Best-response latent vector of feature F_X.
$G(z_r)$	Prior belief.
$G(z^*(F_X))$	Posterior belief given F_X.
	Loss functions:
$\mathcal{L}_p$	Privacy loss, EM distance between the distributions of reconstructed and prior images.
$\mathcal{L}_u$	Utility loss, deviation from the protected feature to the original feature.
$\mathcal{L}_{\text{inv}}$	Reconstruction loss, deviation from features of reconstructed images to original features.

A. Design Choices

We make the following design choices according to the goals above:

Feature-manipulation protection. We manipulate the locally encoded feature representation $Enc(x)$ before release so that the attacker fails to extract private information from what it intercepts. The reason that we do not perform image-level manipulation [11], [28] is that previous image perturbation either fails to prevent white-box attack, or is visually identifiable (See §VI). Further, we do not replace the local encoder Enc with other models since Enc has been pre-trained to fit the downstream tasks. A simple replacement may fail to meet the inference or training requirement.

Protection against white-box attacks. In our setting, $Enc(\cdot)$ is deployed by the service beforehand and thus can be acquired by the adversary. Our scheme should fight against a white-box attacker which is typically stronger than a black-box attacker. A defence method that withstands the stronger white-box attacker suffices to transfer well to the weaker black-box inversion attacks.

Exploiting non-private prior. To achieve the privacy goal, previous work formulates a multi-player game: the defender maximizes the difference between information revealed by released features and the private raw images; the attacker updates itself simultaneously against defender's strategy [17], [37]. We refer to the type of strategy as the *stay-away* (from the original) approach. However, there is no guarantee that the attacker strategy is worst-case at the end of the optimization. Hence the attacker can proceed the adversarial game given a fixed defence strategy and eventually undermines the protection.

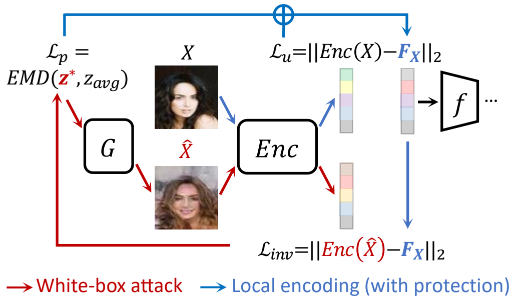

$\mathcal{L}_p$ = EMD(z^*, z_{avg})
X
$\mathcal{L}_u$=||$Enc(X)$−F_X||$_2$
G
$\hat{X}$
Enc
f
...
$\mathcal{L}_{inv}$=||$Enc(\hat{X})$−F_X||$_2$
White-box attack
Local encoding (with protection)

Fig. 4. Overview of our feature crafting scheme against inversion attack. Attacker (red) obtains a best-response latent vector z^* of protected feature F_X by minimizing inversion loss $\mathcal{L}_{\text{inv}}$. Defender (blue) manipulates F_X to balance privacy $\mathcal{L}_{\text{p}}$ of reconstructed z^* and utility $\mathcal{L}_{\text{u}}$ for computation tasks.

In contrast to the stay-away approach, we choose a *get-close* method to draw the released features close to a public prior on the reconstructed image space. Notice that we make no assumption on the defender's knowledge about any particular adversary's prior: any public prior will suffice as long as it does not overlap users' private information. Ideally, the exposed features do not enhance the attacker's knowledge. They act like poison training samples to the inversion attacker, as they are close to the raw data on the feature space but drastically different on the inverted image space. Hence the attacker will only corrupt its model if it adaptively updates its strategy based on these poison samples, thus breaking the adversaial game.

Distributional distance as privacy loss. To better quantify what an attacker perceives from the reconstruction, we follow the well-defined 'perceptual quality' formulation in signal restoration [8], [32], [7] and adopt distributional distance as our privacy-preserving loss. In image signal restoration, perceptual quality refers to how much an output signal $\hat{X}$ is perceived by humans as a realistic sample. It quantifies the perception of 'naturalness' with the distance between the distribution of output signals and the distribution of natural signals, eg. EMD [7], [8]. Similarly, we quantify the attacker's perception of identity using the EMD between the possible reconstruction distribution and the non-private average face distribution. Just as a smaller EMD in signal restoration indicates the output signal is perceptually closer to realistic images, a smaller EMD in our scenario implies that the reconstruction is more likely to be perceived as non-private images and thus higher identity perceptual privacy.

B. Privacy-Preserving Feature Crafting

Our privacy-preserving feature construction scheme embeds carefully-crafted perturbations in the feature representation before releasing it to complete the computation tasks. Such a perturbation is crafted to disrupt the attacker's reconstruction ability via feature collision [27], misleading the attacker's view of the private input to some non-private prior, and is kept small to maximally retain the utility of the downstream tasks.

We show the design overview in Figure 4. Given private images X, the defender simulates a white-box attacker's reconstruction result $\hat{X}^* = G(z^*)$ by minimizing the inversion loss $\mathcal{L}_{\text{inv}}$. $\hat{X}^*$ is used to measure the adversary's perception of certain identity. To prevent the identity leakage from F_X, the defender brings the distribution of reconstructed images close to that of the average faces, which are not associated with any private identity (*i.e.*, minimizing $\mathcal{L}_p$). Meanwhile, the perturbation on feature should be limited so that the computation tasks are not disrupted (*i.e.*, minimizing $\mathcal{L}_u$). We will demonstrate each part in the following.

Privacy protection. Following the design choice, we describe attacker's identity perception by the EMD between the distributions of attacker's inverted images and prior belief $G(z_r)$. The *privacy loss* is:

$$\mathcal{L}_p(z^*(F_X)) = \text{EMD}(G(z^*(F_X))||G(z_r)), \tag{5}$$

where $z^*(F_X)$ is the best-response latent representation of F_X that minimizes the white-box inversion loss $\mathcal{L}_{\text{inv}}(F_X, z) = \|F_X - Enc \circ G\,(z)\|_2$ which replaces $Enc(x)$ in Eq. (3) with F_X. As discussed in §IV-A, $\mathcal{L}_p$ represents the enhancement of attacker's knowledge given F_X compared to its prior belief of general facial images. With this loss minimized, the attacker is tricked into generating close-to-average faces and perceiving non-private identities, thereby achieving a successful defence.

Utility preservation. We restrict F_X's deviation from the original feature $Enc(X)$ to prevent severe drops in downstream utility. The *utility loss* is:

$$\mathcal{L}_u(F_X) = \|F_X - Enc(X)\|_2\,. \tag{6}$$

Notice that the utility loss does not concern the downstream model f. Given an encoder that already functions well for f (either pretrained in the deployment scenario or generally provided in the development scenario), f is expected to be robust under minor deviation from its original input $Enc(X)$, and intuitively the larger the feature deviation, the larger the impact on utility. This independence of f decouples privacy goals from the downstream model, and thus admits unknown computation tasks.

Overall objective. Given a target network $Enc \circ f$, a private input $X \in \mathcal{X}_{\text{pvt}}$ and a generator G trained with public images $\mathcal{X}_{\text{pub}}$, the high complexity and nonlinearity of Enc and G makes it possible to find a feature F_X^* that collides with the original feature $Enc(X)$, while its best-response reconstructed image $G(z^*)$ approximates to the average face distribution $G(z_r)$ in image space. The overall goal is to seek an ideal spot in the utility-privacy tradeoff. Hence the optimization objective is such that

$$\begin{gathered}\min_{F_X} \mathcal{L}_p(z^*) \text{ where } z^* = \arg\min_z \mathcal{L}_{\text{inv}}(F_X, z), \\ \text{subject to } \mathcal{L}_u(F_X) \leq l.\end{gathered} \tag{7}$$

We adopt its Lagrange dual form, and transform the minimization of EMD (Eq. 5)) which has no closed-form solution to a minimax game as the canonical Wasserstein-GAN. This formulation utilizes neural network, so we use 'neural net distance' $d_{nn}(\cdot,\cdot)$ with EMD interchangeably. Thereby we formulate our protection scheme as:

$$\begin{aligned}&\min_{F_X} \max_{|D|_L \leq 1} \mathcal{L}_p(D, z^*(F_X)) + \beta \cdot \mathcal{L}_u(F_X) \text{ where} \\ &\qquad z^*(F_X) = \arg\min_z \mathcal{L}_{\text{inv}}(F_X, z), \\ &\mathcal{L}_p(D, z^*(F_X)) = \mathbb{E}_{z_r}[D \circ G(z_r)] - \mathbb{E}_{z^*}[D \circ G(z^*(F_X))], \\ &\mathcal{L}_u(F_X) = \|F_X - Enc(X)\|_2\,, \\ &\mathcal{L}_{\text{inv}}(F_X, z) = \|F_X - Enc \circ G\,(z)\|_2\,.\end{aligned} \tag{8}$$

The discriminator D introduced here attempts to distinguish the reconstructed image from the average face. It is different from the pretrained D in the white-box attack (Eq. (12) in the appendix [3]). Also notice that the pretrained G is fixed, and it is the feature F_X that competes with the discriminator D.

The above formulation is a nested optimization. Solving it with gradient-based optimizers is challenging as one must differentiate through the best-response latent vector z^* as a function of the feature F_X. To address this problem, we propose a method based on the Implicit Function Theorem (IFT) to compute the privacy loss gradient with respect to F_X.

Optimization via IFT. We show how to solve Eq. (8) to seek an optimized F_X^*. $\frac{\partial \mathcal{L}_p(D,z^*)}{\partial D}$ and $\frac{\partial \mathcal{L}_u(F_X)}{\partial F_X}$ are both *direct gradients*, and can be directly computed. The bottleneck lies in the *indirect gradient* $\frac{\partial \mathcal{L}_p(D,z^*)}{\partial F_X}$, since z^* changes in each iteration with respect to the protected feature. $\frac{\partial z^*(F_X)}{F_X}$ is difficult to obtain as z^* is determined by optimizing $\mathcal{L}_{\text{inv}}(F_X, z)$. We thus resort to the IFT (Lemma 1) and compute the indirect gradient $\frac{\partial \mathcal{L}_p(D,z^*(F_X))}{\partial F_X}$ as

$$-\alpha \frac{\mathcal{L}_p(D,z^*)}{\partial z^*} \cdot \lim_{i \to \infty} \sum_{j=0}^{i} \left[I - \alpha \frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial z} \right]^j \cdot \frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial F_X}$$

Detailed derivations are in Appendix C [3]. Having tackled the implicit differentiation, we are ready to solve Eq. (8).

C. Algorithm of Crafter

We outline our scheme in Alg. 1. Alg. 2 is adopted from [20] for computing the indirect gradient $\frac{\partial \mathcal{L}_p(D,z^*)}{\partial F_X}$. In Alg. 1, the simulated attacker intercepts the feature representations of a batch of private images $X \in \mathbb{R}^{b \times (w \times h)}$, and computes the corresponding $z^* \in \mathbb{R}^{b \times d}$. Each $z^{*(j)}, j \in \{1, \cdots, b\}$ generates a reconstructed image $G(z^{*(j)}) \in \mathbb{R}^{w \times h}$, which can be considered as a sample from the distribution of reconstructed images, rather than the distribution of pixels in the image. The discriminator respectively samples m images from the reconstructed images and the average images, trying to distinguish the two groups of data at each iteration of optimization. Unlike the canonical WGAN where a generator directly competes with the discriminator, our generator G is pretrained on public images and fixed during the process. It is the feature F_X that strives to confuse the discriminator. We follow the tradition in WGAN that the discriminator undergoes multiple training steps (lines 5 to 8) for each update of F_X.

To sum up, our feature crafting system operates in the following two phases.

Offline: G and Enc preparation. The trusted party collects a public image dataset $\mathcal{X}_{\text{pub}}$ on which it trains a WGAN following Eq.(12), and releases the trained generator G. G takes in random latent vectors and outputs realistic-looking facial images. The user receives G from any trusted party and the local encoder model Enc from cloud depending on the utility tasks. For a deployment task, f is also readily deployed on cloud. For a development task, features from users are crowdsourced for training new models on cloud.

Algorithm 1 Crafter

```
Input: Target network $Enc$, generator $G$, a batch of private images $X$ of batch size $b$, minibatch size $m$, latent vector dimension $d$, training iterations of the discriminator per feature update $n_{\text{critic}}$, tradeoff scale $\beta$.
 1: Initialization: $F_X \leftarrow Enc(X)$, $z_{\text{avg}} \leftarrow \texttt{randn}(b,d)$ $z_r \leftarrow \texttt{randn}(b,d)$
 2: while $F_X$ has not converged do
 3:     $z^* = \arg\min_z \mathcal{L}_{\text{inv}}(F_X, z)$
 4:     for $t = 0, \ldots, n_{\text{critic}}$ do
 5:         Sample $\{z^{*(j)}\}_{j=1}^m$ a minibatch from inverted $z^*$.
 6:         Sample $\{z_r^{(j)}\}_{j=1}^m$ a minibatch from random $z_r$ .
 7:         $\mathcal{L}_p \leftarrow \frac{1}{m} \sum_{j=1}^m \left[ D_\omega \circ G(z^{*(j)}) - D_\omega \circ G(z_{\text{avg}}^{(j)}) \right] + g_p$
 8:         $\omega \leftarrow \texttt{AdamOptimizer}(\nabla_\omega \mathcal{L}_p, \omega)$
 9:     end for
10:     $\mathcal{L}_p \leftarrow \frac{1}{bs} \sum_{j=1}^b -D_\omega \circ G(z^{*(j)})$
11:     $v_1 \leftarrow \texttt{approxInverseHVP}(\frac{\partial \mathcal{L}_p}{\partial z^*}, \frac{\partial \mathcal{L}_{\text{inv}}}{\partial z^*})$
12:     $v_2 \leftarrow \beta \frac{\partial \mathcal{L}_u}{\partial F_X} - \texttt{grad}(\frac{\partial \mathcal{L}_{\text{inv}}}{\partial z^*}, F_X, \texttt{grad\_outputs} = v_1)$
13:     $F_X \leftarrow \texttt{AdamOptimizer}(v_2, F_X, lr = flr)$
14: end while
15: $F_X^* \leftarrow F_X$
Output: Crafted feature $F_X^*$.
```

Algorithm 2 approxInverseHVP($\frac{\partial \mathcal{L}_p}{\partial z}, \frac{\partial \mathcal{L}_{\text{inv}}}{\partial z}$).
Experiments used the default values $\alpha = 0.001$, $si = 150$

```
Initialization: $p \leftarrow \frac{\partial \mathcal{L}_p}{\partial z}$
for $j = 0, \ldots, i$ do
    $v \leftarrow v - \alpha \cdot \texttt{grad}(\frac{\partial \mathcal{L}_{\text{inv}}}{\partial z}, z, \texttt{grad\_outputs} = v)$
    $p \leftarrow p + v$
end for
Output: $\alpha p$
```

Online: feature crafting and task completion. After determining Enc and G, the user runs Alg.1 to construct F_X of its private images and sends F_X to the cloud. The server receives F_X. If it undertakes a deployment task, the features go through model f to return a prediction result. Otherwise, the server collects features as training data to train a target model.

D. ϵ-Perceptual Inversion Indistinguishability

We formally define ϵ-perceptual inversion indistinguishability (PII) which is inspired by the concepts of differential privacy and t-closeness. The PII is directly defined on the EMD loss which indicates how a simulated white-box attacker perceives identities from reconstructions. Specifically, let $\mathcal{X}_{\text{pub}}$ be the public dataset with no private identity involved. For any feature F_X, let $G \circ F_X$ denote the inverted distribution using white-box G, i.e. $G \circ F_X = G(z^*(F_X))$. We then have:

Definition 2 (ϵ-Perceptual Inversion Indistinguishability)**.** *A feature crafting system M is ϵ-perceptual inversion indistinguishable (ϵ-PII) on the private image $\mathcal{X}_{pvt}$ if*

$$\text{EMD}(G \circ M(\mathcal{X}_{pvt}) || G \circ M(\mathcal{X}_{pub})) \leq \epsilon \tag{9}$$

The interpretation of Def. 2 is that the smaller the privacy upperbound ϵ, the closer M is to ideal privacy perceptually against the inversion attack, and thus the less identity leakage.

Note that PII is not associated with any realistic attacker but only a public generator G serving as a simulated privacy indicator — in fact the specific choice of G would affect only the value of ϵ, but not the qualitative relationship between ϵ and the realistic defence capability. That is, as long as G is well-trained (eg. provided by a trusted third party and capable of reliable inversions), the defence strength against realistic attacks will increase as ϵ decreases.

Crafter meets the definition with $G \circ M(\mathcal{X}_{\text{pvt}})$ being $G(z^*(F_X^*))$ for $X \in \mathcal{X}_{\text{pvt}}$, and $G \circ M(\mathcal{X}_{\text{pub}})$ being the non-private prior $G(z_r)$. Hence as Crafter optimizes the feature F_X towards minimizing EMD$(G(z^*(F_X))||G(z_r))$ in Eq. (5), it minimizes the left-hand-side of Eq. (9), thereby satisfying Def. 2 with a smaller ϵ indicating stronger perceptual privacy.

Validity of ϵ-PII. We show ϵ-PII is a valid indistinguishability index by providing 1) discussion on avoiding potential limintation of PII; 2) experimental consistency of perceptual privacy in §VI-B. A potential limitation of the ϵ-PII formulation is that EMD may not measure the inter-distribution distance precisely when the stability difference between the users' reconstructed $M(\mathcal{X}_{\text{pvt}})$ and the public prior $M(\mathcal{X}_{\text{pub}})$ is too large. That is, the ϵ-value might not reflect the true dissimilarity between the distributions. To reduce the stability difference, Crafter normalizes the public faces according to the pixel mean and variance of users' private images so that the reconstructed images are likely to be pixel-level-similar to the public faces $G(z_r)$ in Def. 2.

Connection to other privacy concepts. We notice that ϵ-PII is related to the conventional ϵ-differential privacy and t-closeness definitions. We analyze their relations as below.

Definition 3 (Adjacent datasets)**.** *Two datasets D and D' are adjacent if they differ in the existence of a single user's data.*

Definition 4 (ϵ-Differential Privacy)**.** *Let $D_\infty(P||Q)$ denotes the max divergence between distributions P and Q. A randomized mechanism M is ϵ-differentially private (ϵ-DP) if its distribution over any two adjacent datasets D and D' satisfies $D_\infty(M(D)||M(D')) \leq \epsilon$.*

DP and PII share the same intuition: bounding the impact that the private data's presence has on the mechanism outputs. The key differences lie in their definitions of paired datasets, the dependency of ϵ values, and the choices of divergence function. DP considers a pair of adjacent dataset (D, D') differing on a single user's private data record, while the $\mathcal{X}_{\text{pvt}}$ and $\mathcal{X}_{\text{pub}}$ considered in PII differ on the existence of the private identities, which aligns with our goal of quantifying identity leakage. Second, ϵ in DP is a privacy upper bound on any adjacent sets, focusing on the worst-case privacy of the mechanism itself on any possible data record. In contrast, PII is contingent on the given private data $\mathcal{X}_{\text{pvt}}$, emphasizing the privacy of the user data under the mechanism's protection. PII is also dependent on the public generator G quantitatively but not qualitatively as shown by our experiments. Finally, PII adopts EMD rather than the max divergence D_∞ for the inter-distribution divergence, as the EMD is closely related to human perception rather than the theoretical worst case.

Next we compare our proposed ϵ-PII to t-closeness, which is conventionally applied to structural tabular data.

Definition 5 (Equivalence class)**.** *Let quasi-identifiers be attributes whose values when taken together can potentially identify an individual in an anonymized data table. An equivalence class of a data table is a set of records that have the same values for the quasi-identifiers.*

Definition 6 (The t-closeness principle)**.** *An equivalence class has t-closeness if the EMD between the distribution of a sensitive attribute S in this class and the distribution of the attribute in the whole table is no more than a threshold t.*

PII and t-closeness both limit the knowledge gain between the prior and posterior view of the attacker (or observer in the tabular data context). We list their counterparts for comparison in Table II. The major differences are as follows. The entity that t-closeness aims to protect is an equivalence class in the data table, while for ϵ-PII is the private raw images as a whole. The private information at risk for t-closeness is a sensitive attribute (eg. a disease), while for PII it is the reconstructed images. Finally, the public prior information assumed by t-closeness is the whole datatable distribution, whereas in PII it is the public images with no identity overlapping.

TABLE II. COMPARISON OF t-CLOSENESS AND ϵ-PII.

Privacy notion	t-closeness	ϵ-PII
Protected entity Attacker's goal	Equivalence class Sensitive attribute S	Private raw images Reconstructed images
Prior	Distribution of S in the whole datatable	Distribution of reconstructed public images
Posterior	Distribution of S in the equivalence class	Distribution of reconstructed private images

Finally, we provide a theoretical upper bound on ϵ of Crafter in Appendix D-A [3]. The theoretical result indicates that at the same utility loss, Crafter offers an ϵ within a bounded distance to the infimum ϵ, thereby achieving an approximate optimal privacy-utility tradeoff.

V. Discussion

We answer the following questions in this section: *Does Crafter remain robust against an adaptive attacker? Why does Crafter use implicit optimization?*

A. Robustness against Adaptive Attacks

We explore three possible adaptive attacks against Crafter. Details of the design idea are in Appendix E [3]. Experimental results are in §VI-C.

A1: Continue the optimization. Once the protection is completed, the feature F_X^* is released and is fixed ever since. We design A1 that continues to optimize its attack model against the protection. Specifically for Crafter, A1 queries Crafter with its own images X, intercepts the corresponding protected features F_X^*, and obtains the reconstructed image $\hat{X} = G(z^*(F_X^*))$ (white-box A1) or $\hat{X} = Dec(F_X^*)$ (black-box A1). Then it updates G or Dec to minimize the reconstruction loss $\mathcal{L}_{\text{attacker}} = \|\hat{X} - X\|_2$. The same can be done for other adversarial game-based defences [37], [17]. We establish the family of adversarial game-based defences, and show why they are vulnerable to A1 while Crafter remains secure.

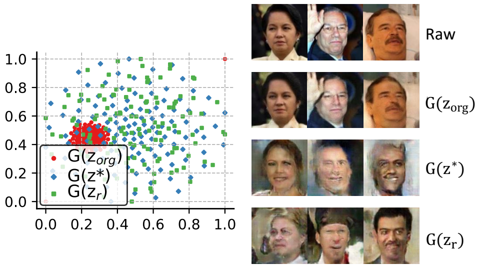


Fig. 5. Visualization of the best-response z_{org} of the raw feature (red), z^* of our crafted feature (blue), and z_r the attacker's prior (green). Our framework shifts the unprotected posterior belief towards attacker's prior belief.

Definition 7 (The family of adversarial game-based protection.)**.** *Given a defencer $\mathcal{P}$ with strategy x_1 and an attacker $\mathcal{A}$ with strategy x_2, a game-based protection framework is:*

$$\text{Find } x^* = (x_1^*, x_2{}^*) \text{ s.t.} \tag{10}$$
$$x_1^* = \arg\min_{x_1} \mathcal{L}_{\text{privacy}}(x_1, x_2^*) + \beta \cdot \mathcal{L}_{\text{utility}}(x_1)$$
$$x_2^* = \arg\min_{x_2} \mathcal{L}_{\text{attacker}}(x_1^*, x_2, X_{\text{test}}).$$

Table III describes our and previous defence [37], [17], [29] under the above framework. $\mathcal{A}$ aims to minimize its loss $\mathcal{L}_{\text{attacker}}$ on the private test set X_{test}, which is not available to $\mathcal{A}$, so in practice $\mathcal{A}$ will use X_{train} instead. Previous work commonly adopt an update approach which converges at $(\hat{x}_1, \hat{x}_2)$, and claim the protection successful as $\hat{x}_1$ has an advantage over $\hat{x}_2$. However, none of them guarantees the adversary $\hat{x}_2$ at convergence is the worst-case. An adaptive $\mathcal{A}_1$ can thus continue to optimize x_2 and reduce the attacker loss on X_{train}, which is the opposite of the privacy loss under the *stay-away* approach. Therefore, the adaptive attacker successfully undermines the defence and transfers well to X_{test}.

Our framework is free of this worry. We argue that our simulated $\mathcal{A}$ without any adversarial update is stronger than any potential A1 adaptive attacks $\mathcal{A}_1$. Under the *get-close* approach, $\mathcal{A}$ is misled to reconstruct images close to random $G(z_r)$, as shown in Figure 5. $\mathcal{A}_1$ essentially matches $G(z_r)$ with X, which are independent and identically distributed image samples. Establishing correspondence between independent samples only results in larger $\mathcal{L}_{\text{attacker}}$ than $\mathcal{A}$ and weakens $\mathcal{A}_1$. The same holds for black-box adaptive attacks. Hence the simulated attacker $\mathcal{A}$ is stronger than any $\mathcal{A}_1$, and the attacker is discouraged from adversarial updates. If $\mathcal{P}$ has an advantage over $\mathcal{A}$, it is robust against A1 adaptive attacks.

A2: Utilize different generators. Crafter's optimization relies on a specific simulated generator model G. This adaptive adversary uses generator models that are different from and possibly stronger than G. Specifically, we evaluate our scheme on generators of different structures and latent dimensions, including the more advanced StyleGAN as proposed in [5], and show through experiments (§VI-C) that Crafter is robust against different generators. This is because Crafter transfers well to different A2 adaptive attacks, as long as the simulated attack used in training can reliably extract private identity information through reconstruction. Hence, when choosing the simulated G in Eq. (8), a generator model with fair reconstruction performance on Enc is sufficient for qualified defence across different A2 adaptive attacks. A2 attacks may also leverage G trained with different public datasets. However, we empirically discover that the attack is usually the strongest when the adversary uses the same public dataset for its WGAN, so we show the worst-case results and only present results using different public sets for StyleGAN as in [5].

A3: Average features over multiple queries. This adaptive adversary waits for a user to query Crafter on the same batch of image multiple times, and averages over the multiple protected features before reconstruction. Privacy of Crafter is solely accomplished by feature perturbation. As discussed in §III, a user can run defence on the same batch of image multiple times and ends up with different feature perturbations because of the randomness of z_r in Eq. (8) during each iteration. As a result, the perturbation of each query may offset each other, and averaging over the queries may remove the perturbation.

We show in §VI-C that this averaging strategy indeed undermines the defence. However, by shuffling user's batch of data each time feeding into Crafter, we ensure robustness against A3 adaptive attacks. We implement shuffling as an inherent part of the encoder in Crafter, so that no additional abnormal query detection is required for the user. The intuition is that A3 works only when image batches of different queries are identical, including the order of images. If the attack averages the features of two batches containing the same images but in shuffled orders, features of different images are mismatched. Perturbations can be hardly removed if the input batches contain a sufficient number of images. If a batch merely has a few images, Crafter records the features the first time the batch is fed, and reuses the features afterwards to prevent averaging.

False security means a protection lacks important robustness evaluations against comprehensive adaptive attacks [30]. A protection must be effective against realistic adaptive attacks to be of practical use. The adaptive attacks in this section could be launched without additional assumptions, and are all easy to implement in real applications. Therefore, if a protection fails to defend adaptive attacks, privacy against basic attacks is meaningless and is merely a false security. No matter how strong the privacy is, the adversary can always breach the security with a simple adaptive approach.

B. Implicit Optimization

One may question the necessity of using implicit optimization in §IV-B. Indeed, we can evade the indirect gradient computation if we perform optimization on the latent representations instead of manipulating features. Specifically, in §IV-B, we control the latent z instead, and the protected feature is a function of z: $F_X = Enc \circ G(z)$. The optimization on latent z thus becomes:

$$\min_{z} \max_{|D|_L \leq 1} \mathcal{L}_p(D, z) + \beta \cdot \mathcal{L}_u(z) \text{ where} \tag{11}$$
$$\mathcal{L}_p(D, z) = \mathbb{E}_{z_r}[D \circ G(z_r)] - \mathbb{E}_z[D \circ G(z)],$$
$$\mathcal{L}_u(z) = \left\| Enc \circ G(z) - Enc(X) \right\|_2.$$

Since F_X can be computed by forwarding z through NNs, we can apply gradient-based optimizers on z and D. The algorithm can be found in Appendix F [3]. We refer to this alternative as *Crafter-z*. Although it evades implicit differentiation, it delivers poor privacy-utility tradeoff empirically, mostly because the

TABLE III. LIST OF GAME-BASED DEFENCE. CE DENOTES THE CROSS-ENTROPY LOSS. C CLASSIFIES FEATURES TO DIFFERENT PRIVATE ATTRIBUTES u. $I(;)$ MEANS MUTUAL INFORMATION.

Name	x_1 of $\mathcal{P}$	x_2 of $\mathcal{A}$	$\mathcal{L}_{\text{privacy}}$	$\mathcal{L}_{\text{utility}}$	$\mathcal{L}_{\text{attacker}}, X \in X_{\text{train}}$
Adv Learn [37]	*Enc, f*	*Dec*	$-\lVert Dec \circ Enc(X) - X\rVert_2$	$CE(f \circ Enc(X), Y)$	$\lVert Dec \circ Enc(X) - X\rVert_2$
Disco [29]	*Enc, f, Pruner*	*Dec*	$-\lVert Dec \circ Pruner \circ Enc(X) - X\rVert$	$CE(f \circ Pruner \circ Enc(X), Y)$	$\lVert Dec \circ Pruner \circ Enc(X) - X\rVert$
TIPRDC [17]	*Enc*	*C*	$-CE(C \circ Enc(X), u)$	$I(Enc(X); X)$	$CE(C \circ Enc(X), u)$
Ours	F_X	G	$\text{EMD}(G(z^*(F_X)), G(z_r))$	$\lVert F_X - Enc(X)\rVert_2$	$\lVert G(z^*(F_X)) - X\rVert_2$

tradeoff is better to manipulate in the feature space (F_X) rather than the latent space (z). We will elaborate on this in §VI-B.

VI. EVALUATION

We aim to answer the following questions in this section:

Q1: Is Crafter effective against white-box attacks? Does Crafter transfer well against black-box attacks? How well does Crafter maintain downstream utility?
Q2: Is Crafter robust against the three adaptive attacks?
Q3: What is the advantage of using implicit optimization?
Q4: Does Crafter introduce large runtime overhead?

A. Setup

Implementation. We implement Crafter with `PyTorch 1.10.0` and run all experiments on NVIDIA GeForce RTX 3090 GPU. We first act as the trusted party to train G on $\mathcal{X}_{\text{pub}}$. The cloud trains Enc and f end to end on $\mathcal{X}_{\text{pub}}$ in the deployment scenario, or leverages a general feature extractor Enc in the development scenario. Users collect Enc and G to generate crafted features of $\mathcal{X}_{\text{test}}$, which accomplish the subsequent downstream tasks.

Datasets. We use the widely-adopted CelebA [34], LFW [19] and VGGFace2 for training and testing Crafter. CelebA is labeled with 40 binary facial attributes, and is split into 200K images for public set $\mathcal{X}_{\text{pub}}$, 17K images for private train set $\mathcal{X}_{\text{train}}$ and 4K for private test set $\mathcal{X}_{\text{test}}$. The input dimension of each image is 64×64. For LFW, we choose 10 independent binary facial attributes and split the dataset into 10K images for $\mathcal{X}_{\text{pub}}$, 2K for private train $\mathcal{X}_{\text{train}}$ and the rest 1K for private test $\mathcal{X}_{\text{test}}$. We crop and resize each image to 128×128. The public $\mathcal{X}_{\text{pub}}$ has no identity overlapping with the private sets, while $\mathcal{X}_{\text{train}}$ and $\mathcal{X}_{\text{test}}$ is a 4:1 (2:1) split for each private identity's images in CelebA (LFW). For VGGFace2, we crop and rezie images to 112×112, and perform 2:1 train-test split. Note that on each dataset, $\mathcal{X}_{\text{train}}$ is used in baselines, or to train the oracle evaluating networks, not by Crafter. We consider 'identity' as the private attribute to be protected whereas the cloud tasks are 40 facial attributes classification, 10 attributes classification, and a 5-class hair color classification, for CelebA, LFW, VGGFace2, respectively.

Models. For the target models under attack, we use the classic image processing DNNs (ResNet18, VGG16 and ResNet50) as $Enc \circ f$ in the deployment and development scenarios. Enc is chosen as the first few layers of the models. For D in Crafter and in the white-box attacker model, a CNN model is adopted. We prepare three generator models — G_1, G_2 and StyleGAN ([5]) — for white-box attacks, and a decoder Dec for black-box attacks. G_1, G_2, Dec are composed of stacks of `ConvTranspose2D` layers. For StyleGAN, $\mathcal{X}_{\text{pub}}$ is from CelebA, and $\mathcal{X}_{\text{test}}$ is from VGGFace2 following the design in [5]; for other models, $\mathcal{X}_{\text{pub}}$ and $\mathcal{X}_{\text{test}}$ are from the same dataset. For the evaluating networks, we adopt ResNet152 for CelebA, Facenet [26] for LFW, and Azure Face API [2] for VGGFace2. Detailed architecture of the networks is in Appendix H [3].

Metrics. We use the mean AUC as the utility metric to evaluate the performance of cloud tasks. For privacy, we simulate white-box, black-box, hybrid white-box and adaptive attacks to reconstruct images from intercepted features. Hyperparameters of the attacks are in Appendix G [3]. The empirical identity perceptual privacy of each defence is evaluated against the attacks by the following metrics:

- *Evaluation Accuracy (Eval Acc).* We use a face verification model as well as the Microsoft Azure Face API [2] as the evaluating networks, trained on $\mathcal{X}_{\text{train}}$. The evaluation accuracy is the identification accuracy on the reconstructed private test images.
- *Feature Similarity (FSIM).* We feed the inverted and raw private $\mathcal{X}_{\text{test}}$ into the evaluating network, extract the penultimate layer outputs and calculate their cosine similarity.
- *SSIM* evaluates the resemblance between the reconstructed images and the original ones on a pixel level. It is a supplement of the semantic security metrics above.
- *Human study.* We conduct a human study to further quantify Crafter's privacy performance following the design in [5].

Different from simply applying the evaluation accuracy of a specified private attribute, our combination of privacy metrics goes beyond attribute-level or pixel-level privacy, but evaluates identity privacy, as mentioned in §III-B.

Baselines. We compare Crafter with state-of-the-art privacy-preserving approaches against inversion attacks. *Adv Learning* [37] and *Disco* [29] fall under the model deployment scenario. *TIPRDC* [17] falls under the development scenario. *Fawkes* [11] and *LowKey* [28] are image-manipulation-based defences against attribute inference attack, and we adapt them to our threat model. We implement Crafter-z as a supplementary baseline as discussed in §V-B.

- *Adv Learning* [37] presents an adversarial game-based approach by pitting the encoder *Enc* and downstream *f* against a black-box decoder Dec trained on $\mathcal{X}_{\text{pub}}$. The tradeoff hyperparameter $\lambda \in \{0.1, 0.5, 0.8\}$. It is a task-oriented protection, which requires prespecified utility tasks and downstream network f. Therefore, this is a baseline under the model development scenario.
- *Disco* [29] takes the same adversarial game-based approach as Adv Learning [37], but it further introduces a pruner to mask privacy-leaking feature channels. We set its tradeoff parameter as $\lambda \in \{0.2, 0.6, 0.8\}$. It is also a baseline under the model deployment scenario.
- *TIPRDC* aims to protect private attribute (ID in our setting) through adversarially training *Enc* on $\mathcal{X}_{\text{train}}$ to minimize the

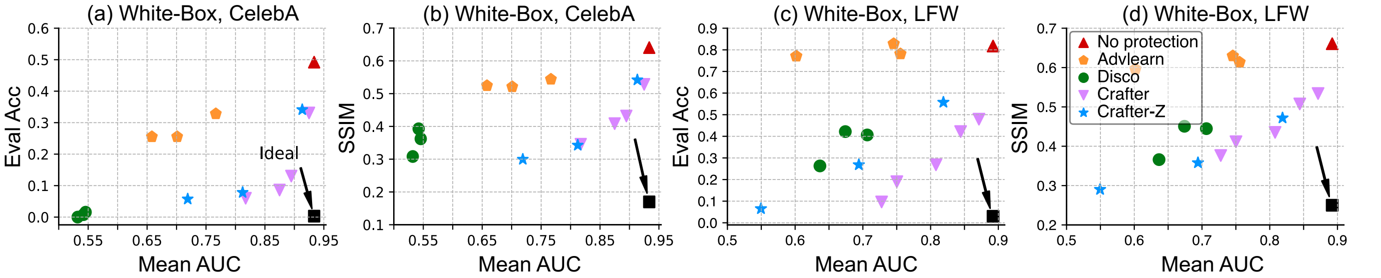


Fig. 6. Privacy-utility tradeoffs against white-box attacks. On CelebA, $\beta \in \{0.5, 1, 2, 10\}$ for Crafter, and $\beta \in \{20, 50\}$ for Crafter-z. On LFW, $\beta \in \{3.5, 4, 4.5, 6, 7\}$ for Crafter, and $\beta \in \{5, 10, 20\}$ for Crafter-z. Subfigures share the same legend. For Adv learning, $\lambda \in \{0.1, 0.5, 0.8\}$. For Disco, $\lambda \in \{0.2, 0.6, 0.8\}$. The black square denotes the ideal tradeoff point.

mutual information between the feature and the input. The tradeoff hyperparameter are chosen as $\lambda \in \{0.1, 0.5, 0.8\}$. It is a task-independent protection, which generates feature representataion of raw input images for unknown downstream tasks. Hence it falls into the category of baselines under the model development scenario.

- *Fawkes* & *LowKey* methods [11], [28] perturb the input images under Fawkes mode {low, mid, high} (or iterations $\in$ {50, 75, 100} for LowKey) to mislead an ensemble of ID classifiers. Their original design is to release the perturbed training images as poison samples for the adversarial ID classifiers. To fairly compare with Crafter, we replay their approach on the private test set to see how it preserves the input privacy.

B. Defence against Basic Attacks

Model deployment scene. We show the Crafter's performance in comparison with baselines in the deployment scenario. For fair and integral comparison, we report results under a variety of utility-privacy tradeoffs by tuning the hyperparameter β. A discussion on β can be found in the end of this section. The ideal tradeoff is a point with a high AUC and a low privacy loss metric value, as depicted by the black square in each figure.

White-box attacks. As Figure 6 shows, Crafter gives the best tradeoff against white-box attacks in almost all cases, as it is closest to the lower-right corner (**Q1**). We leave FSIM-utility plots to Appendix I-A [3] as FSIM is mostly consistent with Eval Acc. In Figure 6(a), $\beta = 1$ reduces the Eval Acc of the unprotected from 49.22% to as low as 8.59%, while downstream AUC drops a mere 0.05. In contrast, Crafter-z offers a less satisfactory tradeoff under the white-box attack. For example, on a 5.7% reconstructed Eval Acc, the average AUC of Crafter-z drops to 0.72, supporting the use of implicit optimization other than direct optimization (**Q3**). Among the three baselines, Adv Learning fails to defend against white-box attacks on both datasets, *e.g.,* on CelebA, the Eval Acc is still high around 30% while it sacrifices 0.20 AUC. Disco improves the tradeoff upon Adv Learning on LFW, but is still inferior to Crafter as in Figure 6(c)(d). On CelebA, Disco achieves strong privacy but low utility akin to random guessing (around 0.55 AUC) for all tradeoff parameter λ values (Figure 6(a)(b)). We test Crafter and the baselines on Microsoft Azure Face and obtain similar tradeoffs. The results are in Appendix K [3].

Black-box attacks. We test Crafter and baselines against the black-box decoder Dec trained on the feature-image pairs of the public data. Comparison between Fig 6 and Fig 7 illustrates that white-box attackers are generally stronger than black-box ones, *e.g.,* the 50% Eval Acc in Figure 6(a) is higher than the 34% in Figure 7(a). And Crafter achieves a comparable or lower value of Eval Acc or SSIM against black-box attacks, and obtains satisfactory utility-privacy tradeoff (**Q1**). On CelebA, Figure 7 show that Crafter-z achieves a comparable tradeoff to Crafter, but is not a proper alternative to the IFT-based Crafter, as the tradeoffs are mostly manipulated through the learning rate lr_z other than β in Crafter-z (**Q3**, further discussion in Appendix L [3]).

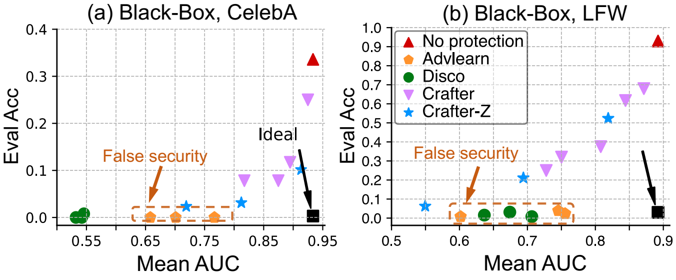


Fig. 7. Privacy-utility tradeoffs against black-box attacks, deployment scenario. For Crafter, $\beta \in \{0.5, 1, 2, 10\}$ on CelebA and $\beta \in \{3.5, 4, 4.5, 6, 7\}$ on LFW. For Crafter-z, as β cannot properly trade off privacy and utility, we manipulate the learning rate instead: $lr_z \in \{0.0001, 0.0005, 0.001\}$ on CelebA and $lr_z \in \{0.0001, 0.001, 0.01\}$ on LFW.

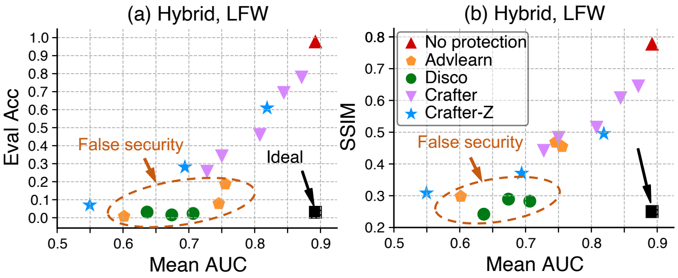


Fig. 8. Privacy-utility tradeoffs against hybrid attacks, deployment scenario.

Hybrid white-box attacks. Since white-box inversion starts off from some random z which can affect optimization, it does not triumph black-box attackers pervasively. This is exactly the case for LFW: without any protection, the EvalAcc of white-box attacks is 81.77%, while that of black-box is high as 92.97%. Hence we initiates a hybrid white-box attack which starts from the output of the black-box attack. The hybrid one successfully outperforms with an EvalAcc of 97.65%. Figure 8 shows that Crafter gives better tradeoffs when pitting against this hybrid attacker than other baselines except for Adv Learning and Disco.

False security. Adv Learning and Disco display much higher robustness against black-box and hybrid attacks than Crafter, shown by Figure 7 and 8. However, we point out that it is an unreliable **false security** discussed in §V-A, as an A1 adaptive attack could easily break them. §VI-C will show that Crafter is in fact superior to the baselines against black-box and hybrid attacks.

Model development scene. We compare the performance of Crafter with TIPRDC on CelebA against white-box and black-box attacks, and report the privacy-utility tradeoffs across βs. The cloud task is to train a 40-facial-attribute classifier given features of $\mathcal{X}_{\mathrm{train}}$, with its performance evaluated by the mean AUC on $\mathcal{X}_{\mathrm{pub}}$. As in Figure 9, Crafter gives a better tradeoff against both white- and black-box attacks than TIPRDC. At $\beta = 1.0$, Crafter reduces the EvalAcc of the unprotected from 43% to 10%, while AUC drops a mere 0.02. Moreover, privacy loss metrics of black-box attacks is close to white-box attacks, confirming the transferability of Crafter against attacks.

In contrast, TIPRDC fails to preserve privacy against attacks as all choices of λ lead to an Eval Acc above 40%. For a fair analysis, we directly report evaluation results against adaptive attacks. As we analyze, TIPRDC exhibits good privacy performance in [17] on the binary sensitive attribute (*e.g.,* gender), which is easy to isolate from the insensitive semantic information. Our setting requires a higher level of semantic privacy, to preserve the identity information that depend on the general appearance and is hard to separate from the input. Hence erasing such private information is contradictory to TIPRDC's utility goal of maximally preserving the semantic information of raw inputs. The conflict goal of utility and privacy disrupts the TIPRDC encoder, yielding features containing ample original information yet fails in the downstream tasks. This explains why the Eval Acc of TIPRDC is even higher than that of the unprotected feature.

A naive baseline in this scenario is to perform the training task using only $\mathcal{X}_{\text{pub}}$, which provides perfect privacy as no private images are involved. However, it may exhibit unsatisfactory utility due to potentially imbalanced utility labels for the unknown training tasks. We simulate an imbalanced $\mathcal{X}_{\text{pub}}$ by randomly choosing 1 among 40 binary utility attributes in CelebA and removing the images with the attribute labeled '0'. The utility AUC of model trained on $\mathcal{X}_{\text{pub}}$ alone drops to 0.76, while that of $\mathcal{X}_{\text{pub}} \cup \mathcal{X}_{\text{pvt}}$ can reach 0.82. Hence users' $\mathcal{X}_{\text{pvt}}$ is needed to augment $\mathcal{X}_{\text{pub}}$ to improve model performance.

Impacts of hyperparamter β. We take a closer look at how β gauges the tradeoff between privacy and utility. Regradless of the attacker type, data points of Crafter in Fig 6, 7, 8 from left to right correspond to $\beta = 0.5, 1, 2, 10$ on CelebA and $\beta = 3.5, 4, 4.5, 6, 7$ on LFW under the deployment scenario. A decreasing β allows a less informative image to be reconstructed but undermines the cloud utility. The same is true for the development scenario in Figure 9. Hence Crafter's tradeoff is easy to manipulate by a single coefficient β.

Validity of ϵ-PII. We show that ϵ-PII successfully reflects Crafter's empirical perceptual inversion privacy, supporting its validity as a perceptual privacy index for Crafter. The exact EMD between the $G(z^*(F_X^*))$ distribution and the $G(z_r)$ distribution is intractable to compute, so we compute its empirical approximation from the distribution samples with the POT solver [13], and scale it down by the image size for simplicity. We show that this approximation error is bounded in Appendix D-B [3]. Figure 10 shows Crafter's result on LFW and CelebA under the model deployment and development scenario against different inversion attacks with different access. As ϵ increases, all three empirical privacy metrics (Eval Acc, SSIM and FSIM) increase collaboratively regardless of the attacker instantiation, indicating weaker empirical identity perceptual privacy against inversion attacks.

C. Defence against Adaptive Attacks

A1: Continue the optimization. We claim in §V that Adv Learning and Disco do not provide any worst-case guarantee of their simulated adversary and is vulnerable against adaptive attacks, while Crafter successfully prevents adaptive attacks under the model deployment scenario. Figure 11 and Figure 12 respectively show how Crafter defends against adaptive white-box and black-box attacks through iterations of update. Basic attacks correspond to epoch 0 in each figure. In Figure 12, line plots of Adv Learning starts at an Eval Acc and SSIM much lower than Crafter, indicating stronger privacy protection against basic attacks. As the black-box Dec proceeds to update itself on $\mathcal{X}_{\mathrm{pub}}$, the privacy loss metrics drastically increases and ends up much higher than Crafter. Disco exhibits a similar performance: on LFW, its Eval Acc increases from 0 to 0.47, 0.60 and 0.63 for tradeoff parameter $\lambda = 0.8, 0.6$ and 0.2 respectively. We omit the adaptive privacy evaluations of Disco on CelebA due to its impractical utility. In contrast, for Crafter, the Eval Acc and SSIM drop to or maintain at the same level with the basic black-box attacks. Similarly for the model development scenario, TIPRDC also fails to defend against A1 adaptive attacks. Figure 13 shows that Eval Acc of TIPRDC increases more than 15% in 70 epochs, while that of Crafter decreases slightly from the basic attacks. The experimental results support our claim (**Q2**).

We summarize how different schemes defend against attacks of varied strength in Figure 14 and the corresponding visualization is in Figure 27 with more in Appendix N [3]. Figure 14(a) reports the average Eval Acc of the raw feature, Adv Learning and Crafter against five attacks on LFW in the deployment scene. It is averaged across $\beta \in \{3.5, 4, 4.5\}$ for Crafter and $\lambda \in \{0.1, 0.5, 0.8\}$ for Adv Learning. Crafter exhibits robust privacy performance against all attacks, while Adv Learning is robust only against basic black-box and hybrid attacks. Figure 14(b) draws a similar conclusion in comparison with TIPRDC in the training scene.

As pointed out by Tramer et. al [30], building a non-robust defence that prevents a particular attack is of little value (the "no-free-lunch-theorem"). The final Eval Acc that evaluates the protection strength should be the maximal value among all basic and adaptive attacks under all adversary access, so Crafter is superior to the baselines, as shown in Figure 14.

A2: Utilize different generators. We show Crafter remains effective across attack models with different structures and z dimensions. Specifically, Crafter uses generator G_1 and z dimension of 500, whereas the adaptive attackers employ

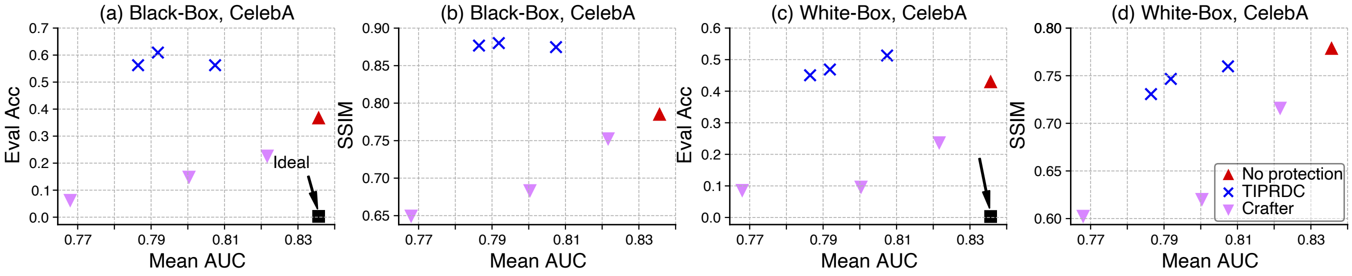
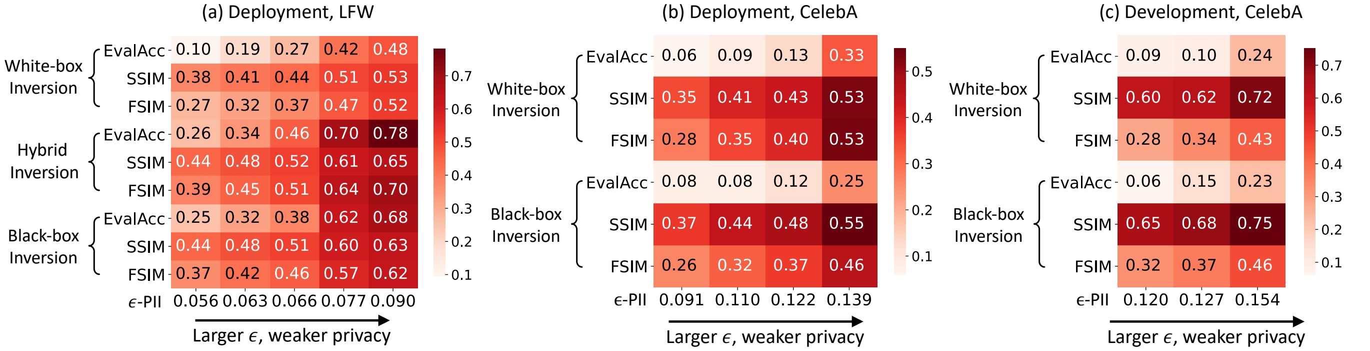


Fig. 9. Crafter and TIPRDC on CelebA against white/black-box attacks, deployment scenario. $\beta \in \{0.5, 1, 5\}$ for Crafter, and $\lambda \in \{0.1, 0.5, 0.8\}$ for TIPRDC.



(a) Deployment, LFW

White-box Inversion	EvalAcc	0.10	0.19	0.27	0.42	0.48
	SSIM	0.38	0.41	0.44	0.51	0.53
	FSIM	0.27	0.32	0.37	0.47	0.52
Hybrid Inversion	EvalAcc	0.26	0.34	0.46	0.70	0.78
	SSIM	0.44	0.48	0.52	0.61	0.65
	FSIM	0.39	0.45	0.51	0.64	0.70
Black-box Inversion	EvalAcc	0.25	0.32	0.38	0.62	0.68
	SSIM	0.44	0.48	0.51	0.60	0.63
	FSIM	0.37	0.42	0.46	0.57	0.62
	ϵ-PII	0.056	0.063	0.066	0.077	0.090

(b) Deployment, CelebA

White-box Inversion	EvalAcc	0.06	0.09	0.13	0.33
	SSIM	0.35	0.41	0.43	0.53
	FSIM	0.28	0.35	0.40	0.53
Black-box Inversion	EvalAcc	0.08	0.08	0.12	0.25
	SSIM	0.37	0.44	0.48	0.55
	FSIM	0.26	0.32	0.37	0.46
	ϵ-PII	0.091	0.110	0.122	0.139

(c) Development, CelebA

White-box Inversion	EvalAcc	0.09	0.10	0.24
	SSIM	0.60	0.62	0.72
	FSIM	0.28	0.34	0.43
Black-box Inversion	EvalAcc	0.06	0.15	0.23
	SSIM	0.65	0.68	0.75
	FSIM	0.32	0.37	0.46
	ϵ-PII	0.120	0.127	0.154

Fig. 10. Crafter's PII on LFW and CelebA against different attacks, deployment and development scenario. A darker color indicates weaker empirical privacy.

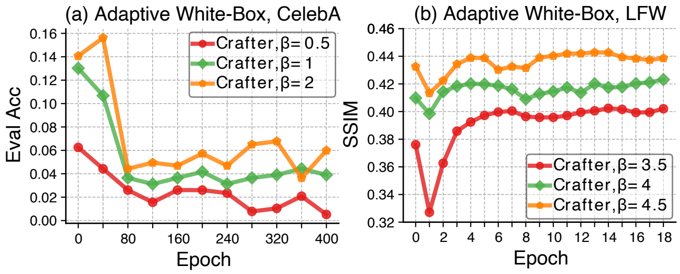


Fig. 11. Crafter on CelebA and LFW against A1 adaptive white-box attacks, deployment scenario.

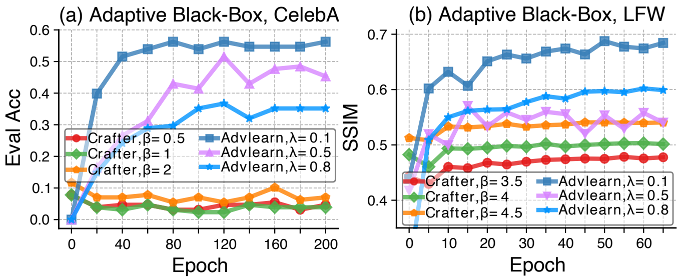


Fig. 12. Crafter and Adv Learning on CelebA and LFW against A1 adaptive black-box attacks, deployment scenario.

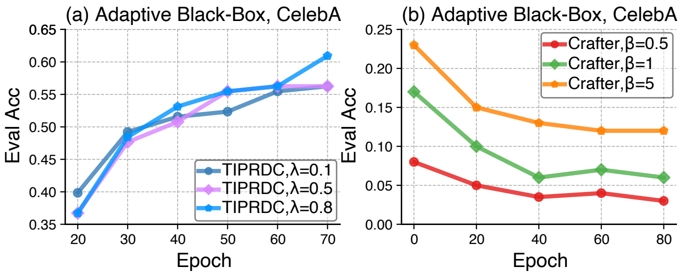
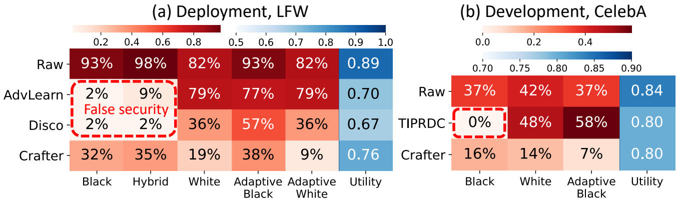


Fig. 13. TIPRDC and Crafter on CelebA against black-box A1 adaptive attacks, development scenario.



(a) Deployment, LFW

	Black	Hybrid	White	Adaptive Black	Adaptive White	Utility
Raw	93%	98%	82%	93%	82%	0.89
AdvLearn	2%	9%	79%	77%	79%	0.70
Disco	2%	2%	36%	57%	36%	0.67
Crafter	32%	35%	19%	38%	9%	0.76

(b) Development, CelebA

	Black	White	Adaptive Black	Utility
Raw	37%	42%	37%	0.84
TIPRDC	0%	48%	58%	0.80
Crafter	16%	14%	7%	0.80

Fig. 14. Average Eval Acc and utility AUC of defences on LFW and CelebA against different attacks, deployment and development scenario. A darker red color indicates less robustness against attacks, and a darker blue indicates better utility. The dashed line indicates false security.

generators G_1 and G_2 across 5 different latent dimensions in the attacks. Figure 15 shows that Crafter's Eval Acc on CelebA at $\beta = 1$ against different attacker models fluctuates within a threshold of 3%, and FSIM differs no more than 0.01. We also evaluate Crafter against an adaptive attack with a more advanced model StyleGAN. Under Crafter protection, the attack achieves 0 success rate on CelebA, and does not beat basic attack on VGGFace2 either (Table IV). The 'mean' entry reports the average SSIM of reconstructed images across the dataset, while the 'worst' entry reports the largest SSIM. Hence Crafter is effective against A2 adaptive attacks (**Q2**).

A3: Average features over multiple queries. We simulate an A3 adaptive attack that queries Crafter 5 times with the same batch of 64 images, and computes their mean as an averaged feature. The drastic increase from 'Basic' to 'Averaging' in Table V shows that without shuffling, crafted perturbations on each query indeed offset each other weakening the defence.

TABLE IV. SSIM of Crafter on VGGFace2 against basic (WGAN) attack and A2 adaptive attack (StyleGAN).

β	Basic attack (WGAN)		A2 attack (StyleGAN)		Utility
	mean	worst	mean	worst	
2	0.27	0.46	0.2	0.3	0.5
5	0.29	0.5	0.22	0.28	0.59
7	0.32	0.5	0.27	0.38	0.6875

TABLE V. Eval Acc of Crafter on CelebA against A3 adaptive attack. Batch shuffling is an essential user requirement to defend against averaging.

β	0.5	1	2
Basic attack	5.98%	8.59%	13.02%
A3 attack (Averaging)	10.16%	20.31%	26.56%
A3 attack against shuffling defence	0%	0%	0.52%

However, with a simple shuffling within the batch, the attack success rate is reduced to almost 0%. Therefore, to ensure robustness against A3 adaptive attack, we implement input shuffling as an inferent part of Crafter. We acknowledge that passing A1, A2 and A3 does not guarantee Crafter bullet-proof. It is possible that more advanced attacks in the future may corrupt the current defence.

D. Image-manipulation Defences

The image-manipulation defences are not strictly comparable with feature-manipulation schemes, but LowKey [11] and Fawkes [28] share the same goal of private information concealing with Crafter. Hence we adopt the two schemes on private test sets to evade identity classification models. Specifically, poisoned images by their schemes are fed to *Enc* and the produced features are being attacked. Figure 16 shows that the encoding and reconstruction procedure strips away Fawkes perturbation on images, leading to privacy loss metrics as poor as the raw feature. LowKey on LFW attains a comparable Eval Acc with a higher AUC than Crafter in Figure 16. However, upon a closer look, the SSIM of reconstructed LowKey images are still as high as unprotected ones, meaning the attacker is able to reconstruct an image with high confidence although a facial verification model fails to predict its true identity. Evani et. al [24] shows that even such an advantage over verification models can be overcome through robust training. The visualization (Figure 17) also supports the conclusion that such image-manipulation protection is ineffective against reconstruction attack. The black-box attack shows similar results.

As an alternative, the poisoned images can be generated to evade *Enc*. It achieves high privacy but disrupts data utility: the AUC drops to 0.52. This is because drawing the user's protected feature close to that of another independent individual (Fawkes) or far away from the original feature (LowKey) causes large feature deviations, thus not preserving utility however small the image perturbation is. Therefore, existing image-manipulation defences fail under our edge-cloud computing scenario.

E. User study and Running Time

Human study. Besides using the evaluating network as an oracle, we conduct a human study to further evaluate if the inverted images under attacks can be recognized by human.

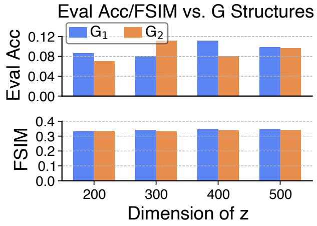


Fig. 15. Crafter on CelebA against basic attack (G_1, $z_{dim} = 500$) and A2 adaptive attacks ($G_1, z_{dim} \in \{200, 300, 400\}$) and ($G_2$).

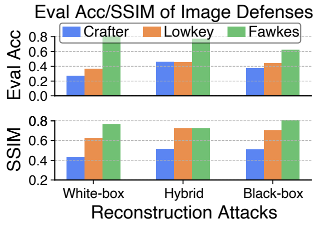


Fig. 16. LowKey, Fawkes and Crafter on LFW against white-box, hybrid and black-box reconstruction attacks.

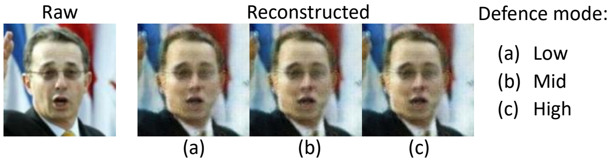


Fig. 17. Reconstructed images under LowKey protection. LowKey fails to preserve pixel-level privacy regardless of the defence mode.

A sample question is in Figure 18, where participants are given one reconstructed LFW image under Crafter's protection with $\beta = 4.5$ (marked with an asterisk). At most one of the options belongs to the same identity as the protected image, and participants may choose "None above" if they believe options A-E do not cover the correct answer. Hence each option has an equivalent rate of 20% to be correct. Please refer to Appendix O [3] for more details of the study protocol.

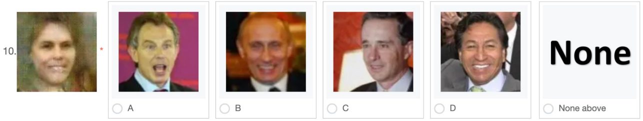


Fig. 18. A sample question from the human study poll.

We eventually harvest 35 valid feedbacks each from a different individual as shown in Table VI. The Macro-F1 measure is 0.251, which is close to that of a random guess (0.200). Thus Crafter is effective by human evaluation.

TABLE VI. Human study results.

35 participants		Participant's choice				
		A	B	C	D	'None'
Ground Truth	A	7	11	11	17	24
	B	17	8	21	6	18
	C	15	4	25	11	15
	D	7	10	4	24	25
	'None'	15	9	11	8	27

Running time. Crafter runs upon every incoming batch, and it is unrealistic to benchmark it against schemes involving network training (*e.g.,* Adv Learning, TIPRDC). Hence, we compare Crafter's running time with a similar setting in LowKey, where the 128×128 LFW image is crafted. The results are shown in Table VII that Crafter's running time is comparable with existing ones (**Q4**). In Crafter, the feature crafting iteration no. is 500 and the `approxInverseHVP` iteration no. is 150. We observe the runtime bottleneck is

TABLE VII. RUNNING TIME COMPARISON BETWEEN CRAFTER AND LOWKEY WITH VARIED FEATURE/IMAGE SIZES, ON LFW.

Defense	Feature (Image) Size	Runtime(s)/Image
Crafter	(64, 16, 16)	27.15
	(128, 16, 16)	30.82
	(64, 32, 32)	53.71
LowKey	(3, 128, 128)	87.45

TABLE VIII. EVAL ACC OF CRAFTER AGAINST WHITE-BOX ATTACK W/WO PRIVATE IMAGE EXPOSURE, AND W/WO THE ORIGINAL FEATURE EXPOSURE, ON CELEBA.

β	0.5	1	2
Crafter	5.98%	8.59%	13.02%
With $\mathcal{X}_{\text{pvt}}$	6.77%	10.93%	13.28%
With $Enc(X)$	9.38%	17.97%	21.09%

the inversion and `approxInverseHVP` steps (lines 3 and 11 in Alg. 1). Thus we speedup the inversion by training an amortizer offline (see details in Appendix M [3]). Improving the speed of the latter is left to future work, which now takes up 71.5% of the total running time.

F. Limitations

We discuss Crafter's limitations to further specify its usage.

Private image exposure. Ideally, private images are not publicly available, but we discuss how the accidental exposure of private images would affect Crafter. We expose 10% of each individual's private images to the white-box attacker and retrain its generator to simulate a realistic exposure. Comparions between entry 'Crafter' and 'With $\mathcal{X}_{\text{pvt}}$' in Table VIII shows the Eval Acc of Crafter on CelebA against white-box attacks with and without private image exposure. There is a slight increase of 2% on Eval Acc, but the impact is minor overall.

Exposure of original features. It is also a threat if the attacker happens to intercept the original features $Enc(X)$ of a set of private images, based on which it trains a feature-to-identity classification network, and predict the ID of private F_X directly. The two groups of private images are different but having identity overlaps. We simulate the case by letting the attacker collect $(Enc(\mathcal{X}_{\text{train}}), ID)$ pairs to train an ID classifier, and infer the ID by feeding in $F_{\mathcal{X}_{\text{test}}}$. The evaluating accuracy is given in Table VIII, which is inferior to Crafter's original performance. Hence it is important not to reveal the corresponding identity label of private features $Enc(X)$ for the effectiveness of Crafter. We consider this requirement attainable, as users are not motivated to share the private identity label anywhere in our problem setting.

VII. RELATED WORKS

We focus on prior works preserving input privacy in deep neural networks (DNNs), especially when inputs are images.

A. Image Perturbation.

To prevent the identity of an image from being disclosed in DNN processing, de-identification is proposed to alter the raw image. One technique to achieve de-identification is adversarial image perturbation, where visually insignificant perturbation is crafted to disrupt the prediction result of an ensemble of identity classifiers [11], [28]. Although these approaches are effective against state-of-the-art auto-recognition models such as Microsoft Azure Face API [2] and Amazon Rekognition [1], they fail to preserve visual privacy, i.e. the perturbed image is close to the original one and any semantic information could be exposed. Zheng *et al.* [39] prevents GAN-inversion-based facial image manipulation with imperceptible image perturbation that maximizes the distance between the original and pertubed images in the latent and feature space. This is a scenario opposite to that of Crafter: [39] pushes the protected image away from the original in feature space to protect privacy, while Crafter draws the feature close to the original to preserve utility. Wu *et al.* [36] protects visual privacy by a transformer network *DAPter* that generates images with low image entropy while minimizing inference loss to preserve image utility. However, it is restricted to specified learning tasks and does not defend inversion attacks. Different from image perturbation, we choose to preserve privacy by injecting perturbations to features.

B. Feature Perturbation.

An alternative to image perturbation/obfuscation is to send an encoded feature of the corresponding input to serve the downstream tasks instead of raw images. The feature, on one hand, carries much information of the input images, on the other, prevents direct revealing of raw inputs. To prevent adversaries from reconstructing inputs or inferring private attributes from features, several works propose to simulate a game between the attacker and protector with conflicting privacy goals: the protector fights against the worst-case attacker by producing features which the attacker would fail to invert. Li *et al.* [18], Xiao *et al.* [37], Singh *et al.* [29] and Wang *et al.* [33] propose to learn perturbed features with adversarial networks and only upload those features for downstream DNN tasks. They either minimize the resemblance of reconstructed images to original ones [37], [18], or minimize the mutual information between the obfuscated feature and the raw input [33]. While sharing a similar idea of preserving utility by minimizing inference accuracy loss, the aforementioned works all suffer the drawback of requiring specified inference tasks in the adversarial training. Being task-independent, TIPRDC [17] and Decouple [25] lift the constraint by maximizing the mutual information between features and raw inputs to preserve utility, and thus require no knowledge of the learning task. However, they demand the private information be specifically defined with labels, i.e., a man with or without a beard. Our method is designed to defend against identity theft, in which the private attribute is challenging to isolate from the raw input.

The above adversarial-networks-based solutions share a common defect: there is no guarantee for the convergence point of the adversarial game, rendering them vulnerable to adaptive attackers in §IV-A. In addition, they all involve retraining the backend model on the cloud, which is typically costly. In comparison, our framework is robust against adaptive attackers, and requires no change in the cloud backend.

VIII. CONCLUSION

We present Crafter, a facial feature crafting system against inversion attacks in deep learning models. We define privacy as the distributional distances between the attacker's posterior and prior views on the input facial images given the feature. As feature is implicitly expressed in the privacy-utility joint optimization objective, we take an IFT-based approach to solve the problem. Analysis and experimental results support that Crafter successfully defends against a variety of attacks with little computation accuracy loss.

Acknowledgment

The authors would like to thank the reviewers for their constructive comments. This work was supported in part by National Key R&D Program of China under Grant 2021ZD0112801, NSF China (62272306, 62032020, 62136006), a RGC RIF grant under the contract R6021-20, RGC CRF grants under the contracts C7004-22G and C1029-22G, and RGC GRF grants under the contracts 16209120, 16200221 and 16207922. Authors would like to appreciate the Student Innovation Center of SJTU for providing GPUs. Shiming Wang personally thanks Ruiming Lu for feedbacks on early versions of this paper.

References

[1] "Amazon rekognition face verification api." https://aws.amazon.com/rekognition/.

[2] "Microsoft azure face api." https://azure.microsoft.com/en-us/services/cognitive-services/face/.

[3] "Our crafter." https://github.com/ShimingWang98/Facial_Feature_Crafting_against_Inversion_based_Identity_Theft/tree/main.

[4] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc.of ACM SIGSAC*, 2016.

[5] S. An, G. Tao, Q. Xu, Y. Liu, G. Shen, Y. Yao, J. Xu, and X. Zhang, "Mirror: Model inversion for deep learning network with high fidelity," in *Proc. of NDSS*, 2022.

[6] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc.of ICML*.

[7] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6228–6237.

[8] ——, "Rethinking lossy compression: The rate-distortion-perception tradeoff," in *International Conference on Machine Learning*. PMLR, 2019, pp. 675–685.

[9] N. Carlini, S. Deng, S. Garg, S. Jha, S. Mahloujifar, M. Mahmoody, A. Thakurta, and F. Tramèr, "Is private learning possible with instance encoding?" in *Proc.of 2021 IEEE S&P*, 2021.

[10] J. Chen, L. Chen, C. Yu, and C. Lu, "Perceptual indistinguishability-net (pi-net): Facial image obfuscation with manipulable semantics," in *Proc.of CVPR*, 2021.

[11] V. Cherepanova, M. Goldblum, H. Foley, S. Duan, J. Dickerson, G. Taylor, and T. Goldstein, "Lowkey: leveraging adversarial attacks to protect social media users from facial recognition," in *Proc.of ICLR*, 2021.

[12] M. Dusmanu, J. L. Schönberger, S. N. Sinha, and M. Pollefeys, "Privacy-preserving image features via adversarial affine subspace embeddings." in *Proc.of CVPR*, 2020.

[13] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer, "Pot: Python optimal transport," in *Journal of Machine Learning Research*, vol. 22, no. 78, 2021, pp. 1–8.

[14] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. of ACM SIGSAC*, 2015.

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Proc.of NIPS*, 2017.

[16] Y. Huang, Z. Song, K. Li, and S. Arora, "Instahide: Instance-hiding schemes for private distributed learning," in *Proc.of ICML*, 2020.

[17] A. Li, Y. Duan, H. Yang, Y. Chen, and J. Yang, "Tiprdc: task-independent privacy-respecting data crowdsourcing framework for deep learning with anonymized intermediate representations," in *Proc.of ACM SIGKDD*, 2020.

[18] A. Li, J. Guo, H. Yang, F. D. Salim, and Y. Chen, "Deepobfuscator: Obfuscating intermediate representations with privacy-preserving adversarial learning on smartphones," in *Proc.of IOTDI*, 2021.

[19] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc.of CVPR*, 2015.

[20] J. Lorraine, P. Vicol, and D. Duvenaud, "Optimizing millions of hyperparameters by implicit differentiation," in *Proc.of AISTATS*, 2020.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *Proc.of INT J COMPUT VISION*, 2004.

[22] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc.of cs.CV*, 2014.

[23] F. M. Orekondy T, Schiele B, "Knockoff nets: Stealing functionality of black-box models," in *Proc. of CVPR*, 2019.

[24] T. F. Radiya-Dixit E, Sanghyun Hong, "Data poisoning won't save you from facial recognition," 2022.

[25] J. Ragan-Kelley, J. Lehtinen, J. Chen, M. Doggett, and F. Durand, "Decoupled sampling for graphics pipelines," in *ACM Transactions on Graphics (TOG)*, vol. 30, no. 3. ACM New York, NY, USA, 2011, pp. 1–17.

[26] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc.of CVPR*, 2015.

[27] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proc.of NIPS*, 2018.

[28] S. Shan, E. Wenger, J. Zhang, H. Li, H. Zheng, and B. Y. Zhao, "Fawkes: Protecting privacy against unauthorized deep learning models," in *Proc.of USENIX Security*, 2020.

[29] A. Singh, A. Chopra, E. Garza, E. Zhang, P. Vepakomma, V. Sharma, and R. Raskar, "Disco: Dynamic and invariant sensitive channel obfuscation for deep neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 125–12 135.

[30] B. W. Tramer F, Carlini N, "On adaptive attacks to adversarial example defenses," in *Proc.of NIPS*, 2020.

[31] B. Wang, F. Wu, Y. Long, L. Rimanic, C. Zhang, and B. Li, "Datalens: Scalable privacy preserving training via gradient compression and aggregation," in *Proc.of ACM SIGSAC*, 2021.

[32] T. Wang, Y. Zhang, Y. Fan, J. Wang, and Q. Chen, "High-fidelity gan inversion for image attribute editing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 11 379–11 388.

[33] T. Wang, Y. Zhang, and R. Jia, "Improving robustness to model inversion attacks via mutual information regularization." in *Proc.of AAAI*, 2020.

[34] Z. Wang, S. Chang, Y. Yang, D. Liu, and T. S. Huang, "Studying very low resolution recognition using deep networks," in *Proc.of CVPR*, 2016.

[35] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," in *IEEE Transactions on Information Forensics and Security*, 2020.

[36] H. Wu, X. Tian, Y. Gong, X. Su, M. Li, and F. Xu, "Dapter: Preventing user data abuse in deep learning inferenceservices," in *Proc.of WWW*, 2021.

[37] T. Xiao, Y.-H. Tsai, K. Sohn, M. Chandraker, and M.-H. Yang, "Adversarial learning of privacy-preserving and task-oriented representations," in *Proc.of AAAI*, 2020.

[38] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *Proc.of cs.LG*, 2020.

[39] L. Zheng, Y. Ning, A. Salem, M. Backes, M. Fritz, and Z. Yang, "Unganable: Defending against gan-based face manipulation," in *Proc.of USENIX Security*, 2023.