APPENDIX A
THREAT MODEL CLARIFICATIONS AND OTHER RELATED WORKS

A. Clarifications on the Threat Model

Three of the most prevailing attacks against user privacy is membership inference attack, attribute inference attack and model inversion attack. We compare Crafter with some most representative baselines in Table IX to further clarify the threat model in our scenario.

Crafter and Adv Learning [37] specifically defend against model inversion attackers that are interested in the victim user's unknown identity. The attacker only has access to some public images that are non-overlapping with the user's identity, and intend to unveil the user's appearance via reconstruction.

Disco [29] and TIPRDC [17] defend both model inversion and attribute inference attacks that aim to attain the users private attributes (eg. age and gender). Specifically, the attribute inference attacker trains an attribute classifer with an available training set. We change the sensitive attribute to 'identity' in our evaluation against inversion attacks.

Fawkes [28] and LowKey [11] also defend against attribute inference attacks, but the private attribute of interest is identity. However, their identity attribute privacy is significantly different from our identity privacy against inversion attack. In LowKey and Fawkes, the attacker already has access to some of the victim's private images, on which it trains an attribute inference (facial recognition) attacker model to infer the identity of the victim's other images. The defence is considered effective as long as the facial recognition attacker fails. In contrast, Crafter's attacker is interested in the appearance of an unknown victim, and the defence is robust only if the reconstructed image reveals no identity information percetually. Hence in Crafter, the facial recognition model merely serves as an oracle to evaluate the perceptual similarity between reconstructed and the raw images, but cannot be a real-world adversary model as private IDs of the training data are missing.

TABLE IX. COMPARISON BETWEEN CRAFTER AND EXISTING DEFENCES.

Defense	Threat model	Privacy	Utility
Crafter	Model inversion	Identity	Task oriented/agnostic
AdvLearn [37]			Task oriented
Disco [29]	Model inversion &	Attribute	Task oriented
TIPRDC [17]	Attribute inference		Task agnostic
Fawkes [28]	Attribute inference	Attribute	/
LowKey [11]	(Attribute=Identity)	(Identity)	

Besides those discussed in §VII, our work is also related to the following works.

B. Image Obfuscation

Another line of work protects users' private information by obfuscating their raw images. *InstaHide* [16] successfully preserves the visual privacy of raw images by mixing multiple images and randomly flipping the signs of the pixels. However, an advanced reconstruction attack [9] can recover nearly visually identical reconstructions to the private images from the obfuscated ones by *InstaHide*. Other work [31], [35], [4] adopt differential privacy (DP), which is a strong privacy guarantee for an individual's data against membership inference attacks and training data memorization. However, the DP guarantee is different from our identity privacy. For example, *DataLens* [31] leverages DP to generate synthetic DNN training data from users' private images, such that two datasets differing by one image sample have close probabilities to generate the same output DNN. Compared to DP, our goal is to prevent the attacker from inferring protected identity information of an image in DNN processing.

C. Feature Obfuscation

Another line of works obfuscates the private feature with other non-private ones to confuse the attacker. Dusmanu *et al.* [12] perform obfuscation by affine space embedding, and Chen *et al.* [10] extends the classic Differential Privacy mechanism to image features for obfuscation. However, the former [12] is designed for image description and visual positioning (e.g. SIFT descriptor [21]) rather than DNN processing, while the latter [10] is confined by the requirement of specifying the utility attributes in prior and thus can not fulfill unknown downstream computation tasks.

APPENDIX B
DETAILS OF WHITE-BOX ATTACK

The adversary aims to find through optimization an image $\hat{X}^*$ that best resembles the input X from the viewpoint of encoder representation [22] , *i.e.,* $\min_{\hat{X}} \|Enc(\hat{X}) - Enc(X)\|_2$. Finding $\hat{X}^*$ from scratch via gradient descent is extremely ill-posed or easily ends up at some visually meaningless local minimum. Therefore, the attackers can exploit public datasets (*e.g.,* facial images crawled from the Internet that are irrelevant to the private input X) to extract prior knowledge about general facial images. Specifically, the white-box attacker distills the public prior using the canonical Wasserstein-GAN structure [15] as proposed by Zhang [38], where a generator $G : \mathbb{R}^d \to \mathbb{R}^{w\times h}$ and discriminator $D : \mathbb{R}^{w\times h} \to \mathbb{R}$ are pretrained on public dataset $\mathcal{X}_{\text{pub}}$ that has no identity overlapping with users' private $\mathcal{X}_{\text{pvt}}$:

$$\min_G \max_D \ \mathbb{E}_{X'\in\mathcal{X}_{\text{pub}}}[D(X')] - \mathbb{E}_z[D \circ G(z)]. \tag{12}$$

Initiated from this public prior, searching for the best-suited image $\hat{X}^*$ can be transformed into gradient-based optimization on latent representation z, starting from some random z_0:

$$z^* = \arg\min_z \|Enc(X) - Enc \circ G(z)\|_2, \tag{13}$$

and the reconstructed image is $\hat{X}^* = G(z^*)$.

APPENDIX C
IMPLICIT GRADIENT COMPUTATION

Here we present the detailed computation of the implicit gradient $\frac{\partial \mathcal{L}_p^*(D, z^*(F_X))}{\partial F_X}$. For (F'_X, z') that satisfies 1) $\frac{\partial \mathcal{L}_{\text{inv}}(F_X, z)}{\partial z}\Big|_{F'_X, z'} = 0$ and 2) the Jacobian matrix of $\frac{\partial \mathcal{L}_{\text{inv}}}{\partial z}$,

$i.e.$ $\left[\frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial z}\right]$ is invertible, surrounding (F'_X, z') we can write $z^*(F_X)$ as a function of F_X s.t. $\frac{\partial \mathcal{L}_{\text{inv}}}{\partial z}\big|_{F_X, z^*(F_X)} = 0$ and

$$\frac{\partial z^*}{\partial F_X}\bigg|_{F'_X} = -\left[\frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial z}\right]^{-1} \times \frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial F_X}\bigg|_{F_X{}', z^*(F'_X)}. \quad (14)$$

Condition 1) $\frac{\partial \mathcal{L}_{\text{inv}}}{\partial z}\big|_{F'_X, z'} = 0$ can be satisfied as the first-order condition is met in optimizing $\mathcal{L}_{\text{inv}}$. As for condition 2), exactly inverting the Hessian in Eq. (14) introduces large computational overhead in high-dimensional space, and hence we adopt a tractable inverse Hessian approximation by Lemma 2:

$$\left[\frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial z}\right]^{-1} = \alpha \lim_{i \to \infty} \sum_{j=0}^{i} \left[I - \alpha \frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial z}\right]^j \quad (15)$$

where α is sufficiently small such that $|I - \alpha \frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial z}| < 1$. Putting them together, we have

$$\begin{aligned} \frac{\partial \mathcal{L}_p^*(D, z^*(F_X))}{\partial F_X} &= \frac{\mathcal{L}_p^*(D, z^*(F_X))}{\partial z^*(F_X)} \frac{\partial z^*(F_X)}{\partial F_X} \\ &= -\alpha \frac{\mathcal{L}_p^*(D, z^*)}{\partial z^*} \cdot \lim_{i \to \infty} \sum_{j=0}^{i} \left[I - \alpha \frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial z}\right]^j \cdot \frac{\partial^2 \mathcal{L}_{\text{inv}}}{\partial z \partial F_X}. \end{aligned} \quad (16)$$

APPENDIX D
PRIVACY GUARANTEE

A. The PII Guarantee of Crafter

We denote the discriminator class as $\mathcal{F}_D$, and use $\mathcal{H}_{F_X \times D}$ to represent the compositional function class of inverting a feature F_X into an image and feeding it into a discriminator. Then the distance between Crafter's PII and the optimal attainable PII is bounded by the Rademacher complexity $\mathcal{R}$ of $\mathcal{F}_D$ and $\mathcal{H}_{F_X \times D}$ as below.

Theorem 1 (PII guarantee on Crafter)**.** *Given dataset pair $(\mathcal{X}_{pvt}, \mathcal{X}_{pub})$, Crafter's setup, utility loss $\mathcal{L}_u = l$, we let the optimal attainable PII at l be ϵ^*. With probability at least $1 - 2\delta$ over the randomness of training samples, we have*

$$\begin{aligned} \epsilon - \epsilon^* \leq\ & 4\mathcal{R}(\mathcal{F}_D) + 4\mathcal{R}\left(\mathcal{H}_{F_X \times D}\right) \\ & + 2\left(Q_{\text{avg}} + Q\right)\sqrt{\frac{\log(1/\delta)}{2bs}}. \end{aligned} \quad (17)$$

The left-hand-side of Eq. (17) captures the performance gap between the Crafter generated feature and the ideal one at the same utility loss. Theorem 1 states that towards approximating a public prior distribution, Crafter generates feature that approaches the theoretically optimal privacy-utility tradeoff with a bounded distance.

Proof: First we make the following standard assumptions: the discriminator function class $\mathcal{F}_{\mathcal{D}}$ is defined on compact parameter sets and define $L_w(i), M_\omega(i)$ and B_{inv}:

$$\begin{aligned} D \in\ \mathcal{F}_{\mathcal{D}} := \{X : ||X|| \leq B_{\text{avg}}\} \mapsto \\ \mathbf{w}_d^\top \sigma_{d-1}\left(\mathbf{W}_{d-1}\sigma_{d-2}\left(\cdots \sigma_1\left(\mathbf{W}_1 \mathbf{x}\right)\right)\right) \in \mathbb{R}, \end{aligned} \quad (18)$$

where $\sigma_i(\cdot)$ is $L_w(i)$-Lipschitz continuous activation function of each discrinimator layer $i = 1, \ldots, d-1$, and W_i are parameter matrices satisfying $||W_i||_F \leq M_\omega(i)$. Similarly, we also assume that given feature F_X, the corresponding reconstructed images satisfy $||\hat{X}^*|| \leq B_{\text{inv}}$.

We define the following abbreviations for the ease of notation. Given feature F_X, the white-box attacker $\mathcal{A}$ starts the inversion in Eq. (3) from the random initial z_0. We denote the distribution of reconstructed images $G(z^*(F_X))$ as $p_{(F_X; z_0)}$, and denote the prior distrbution $G(z_r)$ as p_{avg}. In addition, we use $\hat{p}_{(F_X; z_0)}$ and $\hat{p}_{\text{avg}}$ to denote the empirical distributions of $p_{(F_X; z_0)}$ and p_{avg} over the samples.

By definition, from F_X^* satisfying ϵ-PII, we have

$$\epsilon = \text{EMD}(G(z^*(F_X^*))||G(z_r)) = d_{nn}(p_{(F_X^*; z_0)}, p_{\text{avg}}). \quad (19)$$

Recall that given X, F_X^* denotes the Crafter-generated feature via minimizing objective (8) over the distribution samples, i.e.

$$F_X^* = \arg\min_{F_X} \text{EMD}(\hat{p}_{(F_X^*; z_0)}||\hat{p}_{\text{avg}}) + \beta \cdot \mathcal{L}_u(F_X), \quad (20)$$

from which we have

$$F_X^* = \arg\min_{F_X \in \{F_X | \mathcal{L}_u(F_X) = l\}} d_{nn}(\hat{p}_{(F_X^*; z_0)}, \hat{p}_{\text{avg}}). \quad (21)$$

Also by definition, ϵ^* is the theoretical optimal attainable PII at utility loss l, so

$$\epsilon^* = \inf_{F_X \in \{F_X | \mathcal{L}_u(F_X) = l\}} d_{nn}(p_{(F_X; z_0)}, p_{\text{avg}}). \quad (22)$$

In the following proof, all features F_X are considered under the $\mathcal{L}_u(F_X) = l$ constraint, and we omit it for the ease of notation. To this point, we have

$$\begin{aligned} \epsilon - \epsilon^* =& d_{nn}(p_{\text{avg}}, p_{(F_X^*; z_0)}) - \inf_{F_X} d_{nn}(p_{\text{avg}}, p_{(F_X; z_0)}) \\ =& \underbrace{d_{nn}(p_{\text{avg}}, p_{(F_X^*; z_0)}) - d_{nn}(\hat{p}_{\text{avg}}, p_{(F_X^*; z_0)})}_{\text{(I)}} \\ & + \underbrace{\inf_{F_X} d_{nn}(\hat{p}_{\text{avg}}, p_{(F_X; z_0)}) - \inf_{F_X} d_{nn}(p_{\text{avg}}, p_{(F_X; z_0)})}_{\text{(II)}} \\ & + \underbrace{d_{nn}(\hat{p}_{\text{avg}}, p_{(F_X^*; z_0)}) - \inf_{F_X} d_{nn}(\hat{p}_{\text{avg}}, p_{(F_X; z_0)})}_{\text{(III)}}. \end{aligned} \quad (23)$$

We proceed to bound (I), (II) and (III) respectively. Throughout the proof we use this inequality: $\sup x - \sup y \leq \sup(x - y) \leq \sup|x - y|$, which we denote as (*).

Bound (I):.

$$\begin{aligned} \text{(I)} =& \sup_D [\mathbb{E}_{X_{\text{avg}} \sim p_{\text{avg}}} D(X_{\text{avg}}) - \mathbb{E}_{z_0 \sim p_{\text{avg}}} D \circ G(z^*(F_X; z_0)))] \\ & - \sup_D [\mathbb{E}_{X_{\text{avg}} \sim \hat{p}_{\text{avg}}} D(X_{\text{avg}}) - \mathbb{E}_{z_0 \sim p_{\text{avg}}} D \circ G(z^*(F_X; z_0)))] \\ \overset{(*)}{\leq}& \sup_D |\mathbb{E}_{X_{\text{avg}} \sim p_{\text{avg}}} D(X_{\text{avg}}) - \mathbb{E}_{X_{\text{avg}} \sim \hat{p}_{\text{avg}}} D(X_{\text{avg}})| \\ =& \underbrace{\sup_D |\mathbb{E}_{X_{\text{avg}} \sim p_{\text{avg}}} D(X_{\text{avg}}) - \frac{1}{bs} \textstyle\sum_{i=1}^{b} D(X_{\text{avg}}^{(i)})|}_{R(X_{\text{avg}}^{(1)}, \ldots, X_{\text{avg}}^{(b)})}. \end{aligned} \quad (24)$$

Bound (II): Let $\tilde{F_X} = \arg\min_{F_X} d_{nn}(p_{\text{avg}}, p_{(F_X,z_0)})$, and we obtain

$$\begin{aligned}(\text{II}) &\le d_{nn}(\hat{p}_{\text{avg}}, p_{(\tilde{F_X};z_0)}) - d_{nn}(p_{\text{avg}}, p_{(\tilde{F_X};z_0)}) \\ &\overset{(*)}{\le} \underbrace{\sup_D |\mathbb{E}_{X_{\text{avg}}\sim p_{\text{avg}}} D(X_{\text{avg}}) - \mathbb{E}_{X_{\text{avg}}\sim \hat{p}_{\text{avg}}} D(X_{\text{avg}})|}_{R(X_{\text{avg}}^{(1)},\ldots,X_{\text{avg}}^{(b)}) \quad \text{(same as Eq. (24))}}. \end{aligned} \tag{25}$$

Bound (III): Let $\hat{F_X} = \arg\min_{F_X} d_{nn}(\hat{p}_{\text{avg}}, p_{(F_X,z_0)})$

$$\begin{aligned}(\text{III}) &= d_{nn}(\hat{p}_{\text{avg}}, p_{(F_X^*;z_0)}) - d_{nn}(\hat{p}_{\text{avg}}, \hat{p}_{(F_X^*;z_0)}) \\ &\quad + d_{nn}(\hat{p}_{\text{avg}}, \hat{p}_{(F_X^*;z_0)}) - d_{nn}(\hat{p}_{\text{avg}}, p_{(\hat{F_X};z_0)}) \\ &\overset{(22)}{\le} d_{nn}(\hat{p}_{\text{avg}}, p_{(F_X^*;z_0)}) - d_{nn}(\hat{p}_{\text{avg}}, \hat{p}_{(F_X^*;z_0)}) \\ &\quad + d_{nn}(\hat{p}_{\text{avg}}, \hat{p}_{(\hat{F_X};z_0)}) - d_{nn}(\hat{p}_{\text{avg}}, p_{(\hat{F_X};z_0)}) \\ &\le \sup_D |\mathbb{E}_{z_0\sim p_{\text{avg}}} D\circ G(z(F_X^*;z_0)) - \mathbb{E}_{z_0\sim \hat{p}_{\text{avg}}} D\circ G(z(F_X^*;z_0))| \\ &\quad + \sup_D |\mathbb{E}_{z_0\sim p_{\text{avg}}} D\circ G(z(\hat{F_X};z_0)) - \mathbb{E}_{z_0\sim \hat{p}_{\text{avg}}} D\circ G(z(\hat{F_X};z_0))| \\ &\le 2\sup_{D,F_X} |\mathbb{E}_{z_0\sim p_{\text{avg}}} D\circ G(z(F_X;z_0)) - \mathbb{E}_{z_0\sim \hat{p}_{\text{avg}}} D\circ G(z(F_X;z_0))| \\ &= 2\underbrace{\sup_{D,F_X} |\mathbb{E}_{z_0\sim p_{\text{avg}}} D\circ G(z(F_X;z_0)) - \frac{1}{bs}\sum_{i=1}^{b} D\circ G(z(F_X;z_0^{(i)}))|}_{C(z_0^{(1)},\ldots,z_0^{(b)})}. \end{aligned} \tag{26}$$

To upper-bound $C(z_0^{(1)},\ldots,z_0^{(b)})$, we observe that $\forall z_0^{(1)},\ldots,z_0^{(1)},z_0^{(i)'}$,

$$\begin{aligned}&C(z_0^{(1)}\ldots,z_0^{(i)},\ldots,z_0^{(b)}) - C(z_0^{(1)}\ldots,z_0^{(i)'},\ldots,z_0^{(b)}) \\ &\overset{(*)}{\le} \sup_{D,F_X} |D\circ G(z(F_X;z_0^{(i)})) - D\circ G(z(F_X;z_0^{(i)'}))| \\ &\le 2Q/b \quad \text{(Cauchy-Schwarz inequality)} \end{aligned} \tag{27}$$

where

$$Q = B_{\text{inv}} \prod_{i=1}^{d-1} L_w(i) \prod_{i=1}^{d} M_w(i). \tag{28}$$

We apply the McDiarmid's inequality on Eq. (27) and obtain with probability at least $1-\delta$:

$$\begin{aligned}&C(z_0^{(1)}\ldots,z_0^{(i)},\ldots,z_0^{(b)}) \\ &\le \underbrace{\mathbb{E}_{z_0} C(z_0^{(1)}\ldots,z_0^{(i)},\ldots,z_0^{(b)})}_{\text{(a)}} + 2Q\sqrt{\tfrac{\log(1/\delta)}{2b}}. \end{aligned} \tag{29}$$

$$\begin{aligned}\text{(a)} =&\mathbb{E}_{z_0} \sup_D |\mathbb{E}_{\tilde{z_0}} \frac{1}{bs}\textstyle\sum_{i=1}^{b} D\circ G(z(F_X;\tilde{z_0}^{(i)})) \\ &- \frac{1}{bs}\textstyle\sum_{i=1}^{b} D\circ G(z(F_X;z_0^{(i)}))| \\ \le&\mathbb{E}_{z_0,\tilde{z_0}^{(i)}} \sup_D |\frac{1}{bs}\textstyle\sum_{i=1}^{b} D\circ G(z(F_X;\tilde{z_0}^{(i)})) \\ &- \frac{1}{bs}\textstyle\sum_{i=1}^{b} D\circ G(z(F_X;z_0^{(i)}))| \quad \text{(Jensen's inequality)} \\ =&\mathbb{E}_{\epsilon,z_0,\tilde{z_0}^{(i)}} \sup_D |\frac{1}{bs}\textstyle\sum_{i=1}^{b} \epsilon_i (D\circ G(z(F_X;\tilde{z_0}^{(i)})) \\ &- D\circ G(z(F_X;z_0^{(i)})))| \\ \le&\mathbb{E}_{\epsilon,z_0} \sup_D |\frac{1}{bs}\textstyle\sum_{i=1}^{b} \epsilon_i D\circ G(z(F_X;\tilde{z_0}^{(i)}))| \\ =&2\mathcal{R}(\mathcal{H}_{F_X\times D}). \end{aligned} \tag{30}$$

Therefore,

$$C(z_0^{(1)}\ldots,z_0^{(i)},\ldots,z_0^{(b)}) \le 2\mathcal{R}(\mathcal{H}_{F_X\times D}) + 2Q\sqrt{\tfrac{\log(1/\delta)}{2bs}} \tag{31}$$

where $Q = B_{\text{inv}} \prod_{i=1}^{d-1} L_w(i) \prod_{i=1}^{d} M_w(i)$. Similarly,

$$R(X_{\text{avg}}^{(1)},\ldots,X_{\text{avg}}^{(i)},\ldots,X_{\text{avg}}^{(b)}) \le 2\mathcal{R}(D) + 2Q_{\text{avg}}\sqrt{\tfrac{\log(1/\delta)}{2bs}} \tag{32}$$

where $Q_{\text{avg}} = B_{\text{avg}} \prod_{i=1}^{d-1} L_w(i) \prod_{i=1}^{d} M_w(i)$.

Summing bounds on (I) (II) and (III), we obtain

$$\begin{aligned}\epsilon - \epsilon^* &= d_{nn}(p_{\text{avg}}, p_{(F_X^*;z_0)}) - \inf_{F_X} d_{nn}(p_{\text{avg}}, p_{(F_X;z_0)}) \\ &\le 4\mathcal{R}(D) + 4\mathcal{R}(\mathcal{H}_{F_X\times D}) + 2(Q_{\text{avg}} + Q)\sqrt{\tfrac{\log(1/\delta)}{2bs}}. \end{aligned} \tag{33}$$

■

B. Validity of ϵ-PII

We show that the approximation error between the estimated and theoretical ϵ is bounded. We denote the estimated value as $\hat{\epsilon}$ and adopt the notations in Appendix D-A. Then by definition, $\hat{\epsilon} = d_{nn}(\hat{p}_{(F_X^*;z_0)}, \hat{p}_{\text{avg}})$, and the approximation error is

$$\begin{aligned}&\hat{\epsilon} - \epsilon \\ =&d_{nn}(\hat{p}_{(F_X^*;z_0)}, \hat{p}_{\text{avg}}) - d_{nn}(p_{(F_X^*;z_0)}, p_{\text{avg}}) \\ =&d_{nn}(\hat{p}_{(F_X^*;z_0)}, \hat{p}_{\text{avg}}) - d_{nn}(\hat{p}_{(F_X^*;z_0)}, p_{\text{avg}}) + \\ &d_{nn}(\hat{p}_{(F_X^*;z_0)}, p_{\text{avg}}) - d_{nn}(p_{(F_X^*;z_0)}, p_{\text{avg}}) \\ \overset{(26),(25)}{\le}& C(z_0^{(1)},\ldots,z_0^{(b)}) + R(X_{\text{avg}}^{(1)},\ldots,X_{\text{avg}}^{(b)}) \\ \overset{(31),(32)}{\le}& 2\mathcal{R}(\mathcal{H}_{F_X\times D}) + 2Q\sqrt{\tfrac{\log(1/\delta)}{2b}} + 2\mathcal{R}(D) + 2Q_{\text{avg}}\sqrt{\tfrac{\log(1/\delta)}{2b}} \end{aligned} \tag{34}$$

APPENDIX E
DESIGN IDEAS OF ADAPTIVE ATTACKS

As pointed out by Tramer et al. [30], no automated tool is able to comprehensively assess a protection's robustness. We thus explore three presently possible adaptive attacks **A1** to **A3** that attempt to target the protection's weakest links.

A1: Continue the optimization. A key defence part in Alg. 1 is the defender pitting against a simulated worst-case adversary. In the end of the algorithm, F_X^* is released to downstream tasks and will not be updated anymore. As a result, a stronger inversion attack that breaks the previous worst-case assumption may triumph the fixed defence. We design the adaptive adversary to proceed on minimizing a reconstruction loss $\mathcal{L}_{\text{attacker}}$ (the distance between inverted and original images), which is a consistent loss function. To evaluate our protection comprehensively, besides adjusting the white-box strategy G to optimize the reconstruction loss, we further update black-box Dec as a supplementary adaptive attack.

A white-box adaptive $\mathcal{A}_1$ tries to enhance its own generator G to match the protected feature F_X^* as below. It queries Crafter with its images $X \in \mathcal{X}_{\text{adv}}$, intercepts F_X^*, and reconstructs $G(z^*(F_X^*))$, where $z^*(F_X^*) = \arg\min_z \|F_X^* - Enc \circ G(z)\|_2$ is the best-reponse of the protected feature. Then white-box $\mathcal{A}_1$ updates G as

$$\min_G \|G(z^*(F_X^*)) - X\|_2. \tag{35}$$

Since the EMD between the reconstructed $G(z^*(F_X))$ and average faces $G(z_r)$ is minimized, it is equivalent to matching $G(z_r)$ with X. As random faces $G(z_r)$ and X are independent and identically distributed image samples, such update only weakens G.

Similarly, a black-box $\mathcal{A}_1$ updates its decoder Dec as

$$\min_{Dec} \|Dec(F_X^*) - X\|_2. \tag{36}$$

A2: Utilize different generators. Another key observation is that the defender's optimization relies on a specific simulated generator model G. One may think the defence overfits a particular G and is not as effective against other adaptive attacks using a different and possibly stronger generator models. Thereby we evaluate our scheme on generators of different structures and latent dimensions, including the advanced StyleGAN as proposed in [5].

A3: Average features over multiple queries. Crafter's protection relies soly on the perturbation on the original features. Therefore, we design this adaptive attack that targets the perturbation defence part.

APPENDIX F
ALGORITHM OF CRAFTER-Z

Alg. 3 shows the detailed algorithm of the Crafter-z method that evades implicit differentiation by directly optimizing latent vector z. It is the counterpart of Alg. 1.

In the algorithm, we initialize z as the best-response of $Enc(X)$ so that the corresponding feature representation starts off in the neighborhood of $Enc(X)$ to prevent ineffective utility loss update. z is manipulated to jeopardize discriminator's judgment between the generated distribution and attacker's prior. Meanwhile, the deviation in the feature space is restricted to prevent utility decline.

Algorithm 3 Crafter-z

Input: The same with Alg. 1.

Initialization: $z \leftarrow z^*(Enc(X))$, $z_r \leftarrow \texttt{randn}(bs, d)$
while z has not converged **do**
  for $t = 0, \ldots, n_{\text{critic}}$ **do**
    Sample $\{z^{(j)}\}_{j=1}^m$ a batch from z.
    Sample $\{z_r^{(j)}\}_{j=1}^m$ a batch from random z_r.
    $\mathcal{L}_p \leftarrow \frac{1}{m}\sum_{j=1}^m \left[D \circ G(z^{(j)}) - D \circ G(z_{\text{avg}}^{(j)})\right] + gp$
    $\omega \leftarrow \texttt{AdamOptimizer}(\nabla_D \mathcal{L}_p, D)$
  end for
  $\mathcal{L}_p \leftarrow \frac{1}{bs}\sum_{j=1}^{bs} -D \circ G(z^{(j)})$
  $v \leftarrow \beta \frac{\partial \mathcal{L}_u}{\partial z} + \frac{\partial \mathcal{L}_p}{\partial z}$
  $z \leftarrow \texttt{AdamOptimizer}(v, z, lr = zlr)$
end while

Output: $Enc \circ G(z^*)$

APPENDIX G
ATTACK HYPERPARAMETERS

We provide the hyperparameters for the white-box, black-box, hybrid white-box attacks and their adaptive versions.

For the model deployment scenario in §VI-B, white-box attacks perform 600 iterations of optimization for CelebA and LFW respectively. The latent vectors are of dimension 500 for CelebA and 3000 for LFW. In Crafter-z, the learning rate of latent vector is $lr_z = 0.005$ for both datasets. The hybrid white-box attack optimizes over 150 iterations on the image of LFW with a learning rate of $lr_X = 0.005$. Adaptive white-box attacks use G of white-box attacks as initial parameters, and update G using RMSProp optimizer on $\mathcal{X}_{\text{pub}}$ and the corresponding z_{inv} (calculated by inversion) pairs with $lr = 0.001$. Black-box attacks trains in total 500 epochs using Adam optimizer with default hyperparameters defined in PyTorch: $lr = 0.001$, $betas = (0.9, 0.999)$, $eps = 10^{-8}$, $weight\_decay = 0$. Adaptive black-box attacks use the black-box attack Dec as the initial point, and update 70 more epochs on the protected feature and image pairs. An Adam optimizer with default hyperparameters is adopted.

For the training scenario, white-box attacks perform 1000 iterations of optimization on latent vector of dimension 700 and $lr_z = 0.03$. The update of G of the adaptive white-box attacks uses a RMSProp optimizer with $lr = 0.001$. Hyperparameters of black-box and adaptive black-box attacks are the same with those in the inference scenario.

APPENDIX H
MODEL ARCHITECTURES

Table X, XI, XII, XIII respectively show the target networks, the discriminator, the generator, and the amortized network in the experiments.

TABLE X. ARCHITECTURE OF THE TARGET MODELS.

	ResNet18	VGG16
Image	3×64×64 (3×128×128)	3×64×64 (3×128×128)
Enc	conv 7×7, 64, stride 2	(3×3, 64) ×2, maxpool
	$\begin{bmatrix} 3\times 3, & 64 \\ 3\times 3, & 64 \end{bmatrix} \times 2$	(3×3, 128) ×2, maxpool
Feature	64×16×16 (64×32×32)	128×16×16 (128×32×32)
f	$\begin{bmatrix} 3\times 3, & 128 \\ 3\times 3, & 128 \end{bmatrix} \times 2$	(3×3, 256) ×3, maxpool
	$\begin{bmatrix} 3\times 3, & 256 \\ 3\times 3, & 256 \end{bmatrix} \times 2$	(3×3, 512) ×3, maxpool
	$\begin{bmatrix} 3\times 3, & 512 \\ 3\times 3, & 512 \end{bmatrix} \times 2$	(3×3, 512) ×3, maxpool
	avgPool, linear	avgPool, linear×3

TABLE XI. ARCHITECTURE OF DISCRIMINATOR MODELS.

D
Conv 5×5, 64, stride 2
Conv 5×5, 128, stride 2
Conv 5×5, 256, stride 2
Conv 5×5, 512, stride 2
Conv 4×4, 1, stride 1

APPENDIX I
INFERENCE RESULTS: FSIM VS. UTILITY

A. Against Black and White-Box Attacks

Figure 19 and 20 shows the FSIM-utility tradeoff against black-box and white-box attack respectively. Consistent with the conclusion in §VI-B, Crafter achieves the most desirable utility-privacy tradeoff.

B. Against Hybrid White-Box Attacks

Figure 21 shows the FSIM-utility tradeoff against hybrid white-box attack. Consistent with the conclusion in §VI-B, adversarial learning algorithm reaches an extremely low FSIM, through an adversarial-example-like way.

APPENDIX J
INFERENCE RESULTS: ADAPTIVE ATTACKS

Figure 22 and Figure 23 show respectively the changes of FSIM as the adaptive white-box attack and adaptive black-box attack training goes. Figure 24 is a supplement of Figure 12.

APPENDIX K
INFERENCE RESULTS: AZURE ACCURACY

Figure 25 shows the Azure Eval Acc and utility tradeoff. The tradeoff is consistent with those in Figure 6. Therefore, Crafter is indeed effective against white-box reconstruction attacks when evaluated by a commercial face verification API.

APPENDIX L
DEFECTS OF CRAFTER-Z

Crafter-z is not ideal in the sense that it usually offers poor tradeoffs which are difficult to manipulate. We first found out feasible ranges of β and lr_z to achieve acceptable privacy and utility in Crafter-z. Table XIV demonstrates the impact of β and lr_z of Crafter-z on CelebA. For fixed lr_z, changing β from 20 to 50 does not affect the Eval Acc and AUC much. But for fixed β, varying lr_z deliver a clear tradeoff. This drawback is more prominent on LFW with larger hyperparameter sets: marked yellow are tradeoff points of Crafter-z appearing in clusters of three. Points within each cluster share the same lr_z with $\beta = 5, 10, 20$. This phenomenon again illustrates that lr_z dominates the tradeoff rather than the expected β. In addition to poor tradeoff, this undesirable behavior is a second reason why Crafter-z is not preferred.

TABLE XII. ARCHITECTURE OF GENERATOR MODELS.

G_1	G_2
Linear(input dim, 64×256)	Linear(input dim, 64×128)
BatchNorm+ReLU	BatchNorm+ReLU
Reshape(64×4,8,8)	Reshape(64×8,4,4)
ConvTranspose(64×4, 64×2)	ConvTranspose(64×8, 64×4)
ConvTranspose(64×2, 64)	ConvTranspose(64×4, 64×2)
ConvTranspose(64, 3)	ConvTranspose(64×2, 64)
	ConvTranspose(64, 3)

TABLE XIII. ARCHITECTURE OF AMORTIZE NET, OMITTING BATCH NORM LAYERS AND LEAKY RELU AFTER EACH CONVOLUTION.

Amortize Net
Conv 4×4 128 stride 2
Conv 4×4 512 stride 2
Conv 4×4 1024 stride 2
Conv 2×2 512 stride 1
Conv 1×1 dim_z stride 1

TABLE XIV. CRAFTER-Z TRADEOFF ON CELEBA, INFERENCE SCENARIO.

Hyperparams		Eval Acc %		AUC
lr_z	β	white-box	black-box	mean AUC
0.0001	20	34.11	10.16	91.33
0.0001	50	35.94	10.94	91.01
0.0005	20	9.89	7.03	77.98
0.0005	50	7.81	3.13	81.22
0.001	20	5.73	2.34	71.89
0.001	50	2.86	2.34	73.15

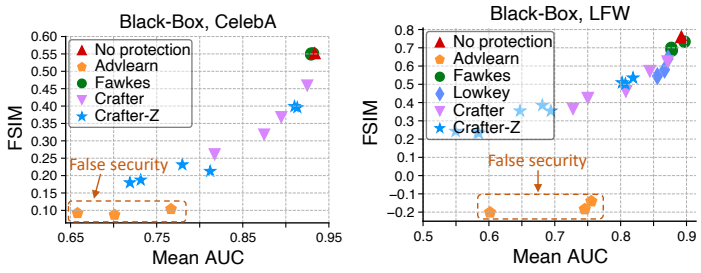
APPENDIX M
SPEED-UP WITH AMORTIZER

The goal of amortizer is to speed up the inversion in line 3 of Alg. 1 by establishing a mapping from feature space to latent space. It receives a batch of feature as input, and is expected to output a latent vector that approximates the real best-response computed via optimization. The white-box attacker uses the output latent vector of the amortizer as its initial point in the optimization, thereby cutting off the time expense.

Specifically, to train the amortizer, we generate some latent vector z following a random distribution, calculate the corresponding features $Enc(G(z))$ and optimize the amortizer $Amor$ as follows:

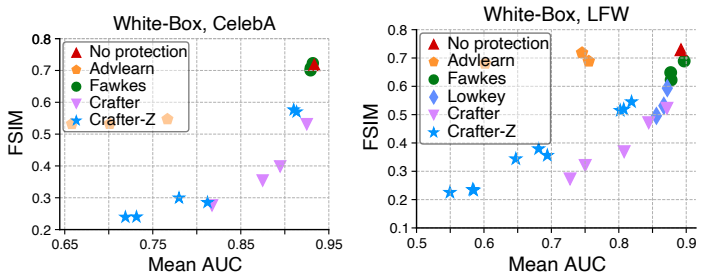
$$\min_{Amor} (z, Amor \circ Enc \circ G(z))). \tag{37}$$

The structure of amortizer we adopt is listed in Table XIII. We refer to a white-box inversion with the assistance of amortizer as the *amortized inversion*. Empirically, we verify that on the LFW dataset, it takes only 100 iterations for an amortized



(a) (b)

Fig. 19. Privacy-utility tradeoffs against black-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.



(a) (b)

Fig. 20. 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.

inversion to achieve the same level of reconstruction performance with that reached by 600 normal inversion iterations, thereby greatly reducing the computation overhead.

Appendix N
Visualization

Figure 26 shows the heatmap of CelebA. Figure 28, 29 are supplement visualizations for Figure 27 to illustrate the effectiveness of different protection schemes against various classes of attacks on LFW, $\lambda = 0.5, \beta = 4.5$. These visualization results further confirm the conclusion as stated in VI-C.

Figure 17 shows the reconstruction results under LowKey protection. Regardless of the protection mode, the reconstructed images resembles the original image, and the protection is ineffective.

Appendix O
Human Study Detail

TABLE XV. Human study statistics.

	A	B	C	D	'None'	Macro-F1 score
Recall	0.100	0.190	0.357	0.343	0.386	
Precision	0.115	0.114	0.347	0.363	0.248	
F1	0.107	0.143	0.352	0.353	0.302	0.251

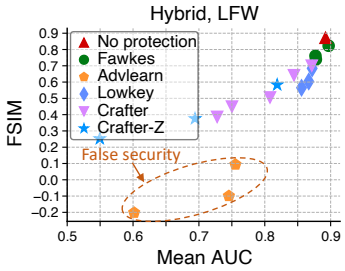


Fig. 21. FSIM-utility tradeoffs against hybrid white-box attacks.

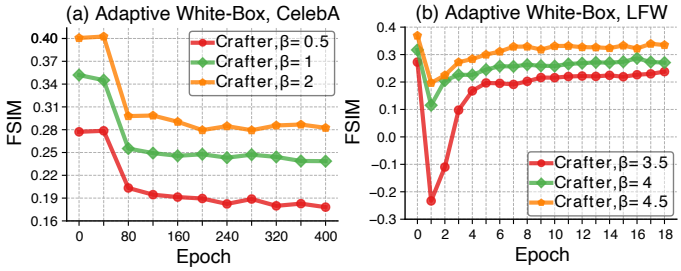


Fig. 22. FSIM changes along training epochs of adaptive white-box attack.

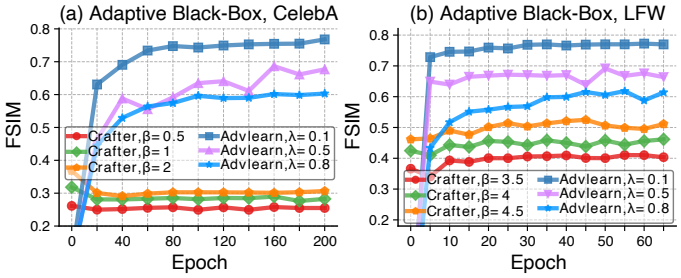


Fig. 23. FSIM changes along training epochs of adaptive black-box attack.

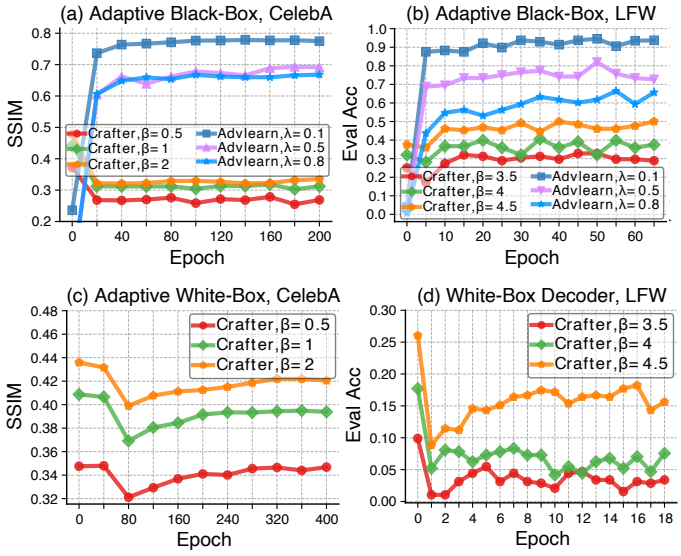


Fig. 24. Crafter and Adv Learning on CelebA and LFW against adaptive black/white box attacks, as a supplement of Figure 12.

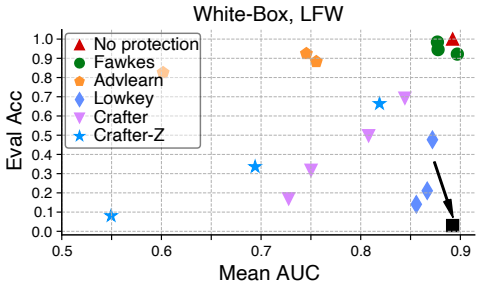


Fig. 25. Azure Eval Acc under white-box attacks, on LFW.

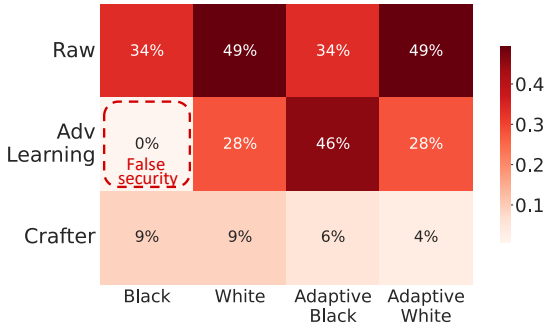


Fig. 26. Average Eval Acc of protection schemes on CelebA against different attacks, deployment scenario, averaged across $\beta \in \{0.5, 1, 2\}$.

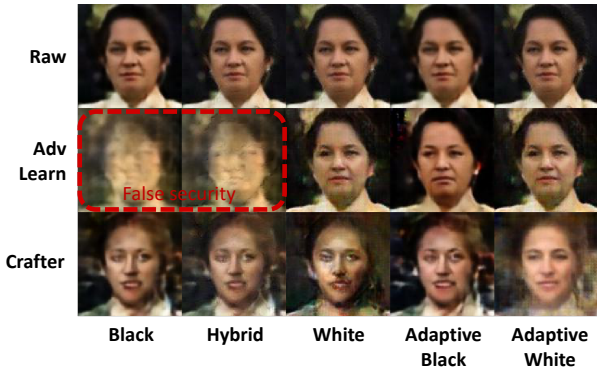


Fig. 27. A visualization effect of protection schemes against different attacks. $\lambda = 0.5, \beta = 4.5$.

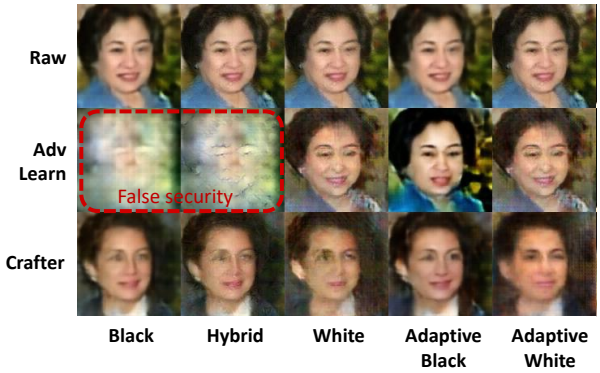


Fig. 28. Visualization results for different protections against different attacks.

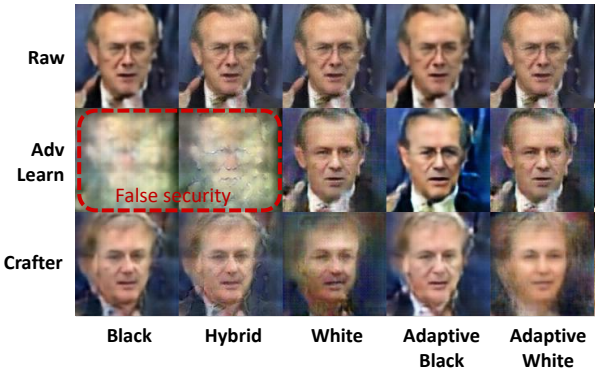


Fig. 29. More visualization results.