

# DiffUMI: Training-Free Universal Model Inversion via Unconditional Diffusion for Face Recognition

Hanrui Wang<sup>1,\*</sup>, Shuo Wang<sup>2</sup>, Chun-Shien Lu<sup>3</sup>, and Isao Echizen<sup>1</sup>

<sup>1</sup>National Institute of Informatics, Japan; <sup>2</sup>Shanghai Jiao Tong University, China; <sup>3</sup>Academia Sinica, Taiwan

\*Corresponding Author

{hanrui\_wang, iechizen}@nii.ac.jp, wangshuosj@sjtu.edu.cn, lcs@iis.sinica.edu.tw

**Abstract**—Face recognition technology presents serious privacy risks due to its reliance on sensitive and immutable biometric data. To address these concerns, such systems typically convert raw facial images into embeddings, which are traditionally viewed as privacy-preserving. However, model inversion attacks challenge this assumption by reconstructing private facial images from embeddings, highlighting a critical vulnerability in face recognition systems. Most existing inversion methods require training a separate generator for each target model, making them computationally intensive. In this work, we introduce DiffUMI, a diffusion-based universal model inversion attack that requires no additional training. DiffUMI is the first approach to successfully leverage unconditional face generation without relying on model-specific generators. It surpasses state-of-the-art attacks by 15.5% and 9.82% in success rate on standard and privacy-preserving face recognition systems, respectively. Furthermore, we propose a novel use of out-of-domain detection (OODD), demonstrating for the first time that model inversion can differentiate between facial and non-facial embeddings using only the embedding space.

## 1. Introduction

Face recognition technology presents significant privacy risks, as it involves processing biometric data that is both sensitive and immutable if compromised. Modern systems address these concerns by leveraging feature embedding techniques, which enhance scalability, generalization to unknown identities, and retrieval efficiency [1]–[3]. These systems convert facial images into feature embeddings, which are stored in a database and compared using distance-based metrics such as cosine similarity or Euclidean distance for recognition. This kind of approaches was traditionally considered privacy-preserving, as it encodes raw biometric data [4]–[7], as shown in Fig. 1. However, model inversion attacks present a significant privacy threat by reconstructing facial images solely from feature embeddings. These reconstructions can facilitate further security breaches, including presentation attacks such as spoofing via photo, video replay, or 3D mask techniques [8]. As a result, model inversion attacks are essential for assessing the privacy vulnerabilities of face recognition systems [9].

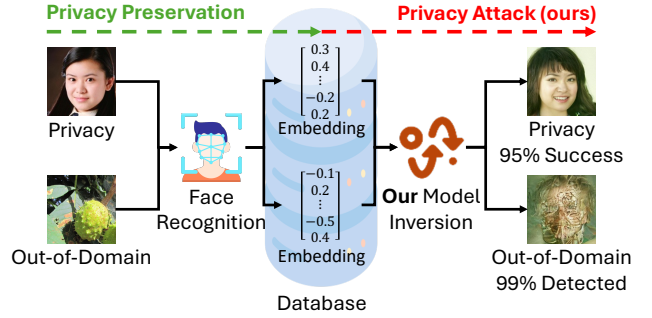


Figure 1. Our model inversion (DiffUMI) in privacy attacks [9] and out-of-domain detection (OODD) [10]–[13]. DiffUMI reconstructs images solely from embeddings in the database, traditionally considered privacy-preserving, achieving a 94.72% reconstruction success rate for facial inputs, with outputs closely resembling the target identity. In parallel, our OODD framework detects 98.9% of non-face inputs while maintaining a low 3.9% false positive rate for face inputs, demonstrating strong capability in distinguishing in-domain from out-of-domain data.

Nevertheless, existing model inversion attacks face several critical challenges. As summarized in Tab. 1, **most approaches are training-dependent, requiring the training or fine-tuning of a target-specific generator for each attack, which incurs significant computational costs** [15]–[27]. Additionally, embedding-based face recognition models are designed to generalize to unknown identities, making feature extraction an open-set task that accommodates a wide range of inputs, including those not seen during the training of the face recognition model or the generator used for inversion. This open-set nature necessitates that model inversion attacks generalize to an infinite number of identities, a requirement many existing methods fail to meet [14], [16], [19]–[23], [25], [28], [29], [31].

**Another major challenge is ensuring high visual fidelity in reconstructed images.** To effectively retrieve identity-related privacy, model inversion requires high-resolution reconstructions rich in visual attributes. A full headshot-style reconstruction, *a.k.a.* selfie, represents the optimal granularity for identity recovery [24], [26]–[28], [30], [31]. However, due to prohibitive training costs, most existing methods generate low-resolution reconstructions (*e.g.*,  $64 \times 64$  or grayscale images) and often focus only

TABLE 1. OVERVIEW OF RELATED WORKS ON MODEL INVERSION ATTACKS.

Method	Generator	Attack Cost			Task		Visual Fidelity		Code
		Training-Free	Input	W/B	Open-Set	OODD	Resolution	Selfie	
MIA [14] [2015]	None	✓	Label	Both	✗	✗	GrayScale	✗	✓
NbNet [15] [2018]	DeconvNet	✗	Embedding	Black	✓	✗	RGB160	✗	✓
Amplified-MIA [16] [2023]	DeconvNet	✗	Label	Black	✗	✗	GrayScale64	✗	✓
DSCasConv [17] [2024]*	DeconvNet	✗	Embedding	White	✓	✗	RGB112	✗	✓
DiBiGAN [18] [2020]	C-GAN	✗	Embedding	Both	✓	✗	RGB	✗	✗
GMI [19] [2020]	C-GAN	✗	Label	White	✗	✗	RGB64	✗	✓
$\alpha$ -GAN [20] [2022]	C-GAN	✗	Label	White	✗	✗	GrayScale	✗	✗
PLG-MI [21] [2023]	C-GAN	✗	Label	White	✗	✗	RGB64	✗	✓
LOKT [22] [2023]	C-GAN	✗	Label	Black	✗	✗	RGB128	✗	✓
ABE-MI [23] [2025]	C-GAN	✗	Label	Black	✗	✗	RGB128	✗	✗
ID3PM [24] [2023]	C-Diffusion	✗	Embedding	Black	✓	✗	RGB64	✓	✗
CDM [25] [2024]	C-Diffusion	✗	Label	Black	✗	✗	RGB64	✗	✗
Shahreza <i>et al.</i> [26] [2023]	StyleGAN	✗	Embedding	Both	✓	✗	RGB1024	✓	✓
Shahreza <i>et al.</i> [27] [2024]*	StyleGAN	✗	Embedding	Both	✓	✗	RGB1024	✓	✓
PPA [28] [2022]	StyleGAN	✓	Label	White	✗	✗	RGB1024	✓	✓
IF-GMI [29] [2024]	StyleGAN	✓	Label	White	✗	✗	RGB224	✗	✓
Dong <i>et al.</i> [30] [2023]	StyleGAN	✓	Embedding	Black	✓	✗	RGB1024	✓	✗
MAP <sup>2</sup> V [9] [2024]*	StyleGAN	✓	Embedding	Both	✓	✗	RGB192	✗	✓
PriDM [31] [2025]	DDPM	✓	Image	Black	✗	✗	RGB256	✓	✗
DiffUMI (ours)	DDPM	✓	Embedding	Both	✓	✓	RGB256	✓	✓

\* denotes benchmark methods used for empirical comparison, representing the latest exemplars of each strategy. Comparisons with closed-set attacks and PriDM [31] are excluded due to fundamental differences in assumptions: they depend on class labels or images, while our approach operates on target embeddings.

on the facial region, which may lack sufficient perceptual detail for accurate human identity recognition [14]–[25].

**Finally, while diffusion models have gained prominence in modern generative AI research [32], generative adversarial networks (GANs) [33]–[35] remain the dominant paradigm for model inversion.** This limited technological diversity restricts the application of state-of-the-art diffusion-driven generation techniques to model inversion. The most straightforward approach to diffusion-driven model inversion is to apply adversarial attacks (*e.g.*, APGD [36]) directly in the latent space. However, this naive strategy produces significant artifacts due to the sensitivity of unconditional diffusion models, leading to overfitting and failed privacy recovery (see Appendix A). Consequently, existing approaches rely on conditional diffusion (C-Diffusion) for facial image generation [24], [25], which are either unsuitable for open-set face recognition or require substantial computational resources (taking 1.5 to 2 days to train generators restricted to  $64 \times 64$  resolution).

**To address the aforementioned challenges, we propose DiffUMI, the first training-free, Diffusion-driven Universal Model Inversion attack against embedding-based face recognition models.** DiffUMI utilizes a fixed, pretrained denoising diffusion model [37] to unconditionally generate full headshot-style selfies, eliminating the need to train target-specific generators. Its universality stems from a consistent framework and generator capable of adapting to arbitrary identities and open-set face recognition models.

In terms of algorithm design, we hypothesize that if a reconstruction accurately matches the target identity without exhibiting adversarial artifacts, it successfully recovers private information. Thus, our objective is to maximize

the embedding similarity between the reconstruction and the target while minimizing perceptual artifacts. Contrary to intuition, we found that the choice of attack backbone (*e.g.*, APGD) is not the primary cause of artifacts. Instead, successful manipulation depends critically on three factors: reliable initialization of latent codes, fine-grained manipulation strategies, and mitigation of adversarial overfitting. To achieve these objectives, we first introduce an automated method for selecting reliable latent codes from randomly sampled Gaussian distributions. Second, we propose a ranked adversary strategy that performs fine-grained adversarial attacks [36], [38], guided by mathematically derived stopping criteria for optimizing the latent space in both white-box and black-box settings. To the best of our knowledge, our pipeline and algorithms are the first to effectively manipulate unconditional face generation.

Nevertheless, perfect artifact-free generation is inherently unattainable, and achieving it is not our goal. Instead, our objective is to recover the target identity from the embedding in a way that is visually recognizable to humans. The danger of adversarial artifacts lies in their potential to cause visually different images to be mapped to the same representation by the target model, leading to misleading matches. To eliminate the overfitting effect of these artifacts, which typically exhibit poor transferability, we evaluate privacy attacks using models different from the target. This evaluation protocol, widely adopted in prior work, assumes that if a reconstruction matches the target identity on models other than the one it was optimized against, it successfully recovers private information (*a.k.a.* transferability) [9], [17], [26], [27], [30].

In practice, we assess DiffUMI on two widely used face

datasets [39], [40] and four face recognition models [4]–[7], comparing its performance against three state-of-the-art benchmarks [9], [17], [27]. In addition to automatic evaluation using face recognition models, we conduct user studies in which human participants are asked to judge whether the reconstruction matches the target identity or whether the target can be identified using the reconstruction alone. The results demonstrate DiffUMI’s strong attack effectiveness, showing that an unconditional diffusion model combined with optimized adversarial search enables efficient, high-fidelity facial reconstruction. For example, when targeting the Labeled Faces in the Wild (LFW) dataset [40] and the ArcFace model [5] in the white-box setting, DiffUMI achieves Type I and Type II accuracies of 98.55% and 94.72%, outperforming benchmarks by 9.57% and 15.5%, respectively. It also successfully breaches privacy-preserving face recognition models [6], [7], designed to resist model inversion, achieving Type II accuracies of 84.42% to 92.87% across two datasets, exceeding benchmarks by 4.01% to 9.82%. These findings raise serious concerns about the effectiveness of current privacy-preserving techniques.

**Moreover, we introduce a novel application of Out-Of-Domain Detection (OODD), marking the first use of model inversion to distinguish non-face from face inputs based solely on embeddings [10]–[13].** As shown in Fig. 1, the open-set nature of face recognition models enables non-face inputs, such as those from ImageNet [41], to be processed into embeddings, which we define as out-of-domain inputs. These embeddings share the same dimensionality and numerical range as genuine facial images, making differentiation inherently challenging. Our OODD framework utilizes model inversion techniques, where reconstructions of out-of-domain inputs typically fail either by not resembling the target identity or by lacking discernible human facial features. These failure cases serve as key indicators for identifying potential out-of-domain inputs. Our OODD framework effectively detects 98.9% of non-face inputs [41] after embedding by the ArcFace model [5], with only a 3.9% error rate for genuine face inputs [39], [40].

The ability to distinguish real human face embeddings from out-of-domain inputs (*e.g.*, animals or synthetic faces) can enhance face recognition systems by mitigating spoofing, deepfake enrollment, and data poisoning risks [10]–[13]. It may support secure identity verification, clean large-scale face datasets, and ensure compliance with biometric standards. It may also aid in monitoring model robustness and controlling unintended use in applications like AR or photo filters. However, none of these works focus specifically on the face domain, embedding-level analysis, and open-set recognition, or leverage model inversion techniques.

**Our key contributions are summarized as follows:**

(i) We introduce DiffUMI, the first training-free, diffusion-driven universal model inversion attack against embedding-based face recognition models. (ii) We propose the first algorithm that effectively manipulates unconditional face generation by automatically selecting highly reliable latent codes and introducing a novel ranked adversary strategy.

(iii) We empirically establish DiffUMI as a state-of-the-art attack, revealing critical vulnerabilities in privacy-preserving face recognition systems and raising concerns about their ability to counter such threats. (iv) We introduce OODD, the first model inversion-based framework for distinguishing non-face inputs from face inputs based solely on feature embeddings.

## 2. Universal Model Inversion via Diffusion

This section introduces DiffUMI, outlining its objectives and framework in Sec. 2.1. We then formalize the threat model and problem definition in Sec. 2.2 and Sec. 2.3, respectively. Finally, we detail the algorithms for the three sequential steps in the framework, as presented in Sec. 2.4 through Sec. 2.6. To make this paper self-contained, we outline the preliminaries of DDPM [42], face recognition, D’Agostino’s  $K^2$  test [43]–[45], and MTCNN [46] in Appendix B. The step-by-step algorithm of DiffUMI and a comprehensive notation list are detailed in Appendix C.

### 2.1. Overview

DiffUMI is introduced as a framework to assess privacy risks of embedding-based face recognition models. A model is considered vulnerable if DiffUMI can reconstruct a facial image resembling the target identity using only its feature embedding. The primary goal of this privacy attack is to recover the target identity, rather than precisely reconstruct the original image from which the embedding was derived.

To facilitate a universally applicable attack, DiffUMI employs a three-step attack mechanism, as depicted in Fig. 2. The framework utilizes DDPM [37], a denoising diffusion probabilistic model, as the generator. Unlike training-dependent model inversion attacks that require a generator to be trained for each target model, our generator is independently pretrained for unconditional facial image synthesis from random Gaussian noise, rendering DiffUMI entirely training-free. As a result, DiffUMI facilitates effective attacks across diverse target identities and models without modifying any other components of the framework.

The core challenge addressed by DiffUMI is how to effectively manipulate unconditional diffusion generation. In the context of model inversion, this requires maximizing the embedding similarity between the target and the reconstruction while minimizing artifacts introduced by adversarial manipulation. While the similarity maximization is handled via a similarity-guided objective function (Sec. 2.3), the more difficult task of artifact minimization is tackled through our proposed three-step attack mechanism:

#### Step (a) (Preparation) – Latent Code Generation:

This phase independently generates a set of highly reliable latent codes, executed once and applicable for attacking any target. The reliability is ensured by a selection strategy combining D’Agostino’s  $K^2$  test [43]–[45] and MTCNN face detection [46]. The  $K^2$  test ensures the latent codes follow a normal distribution, improving statistical reliability, while MTCNN ensures that the generated images contain

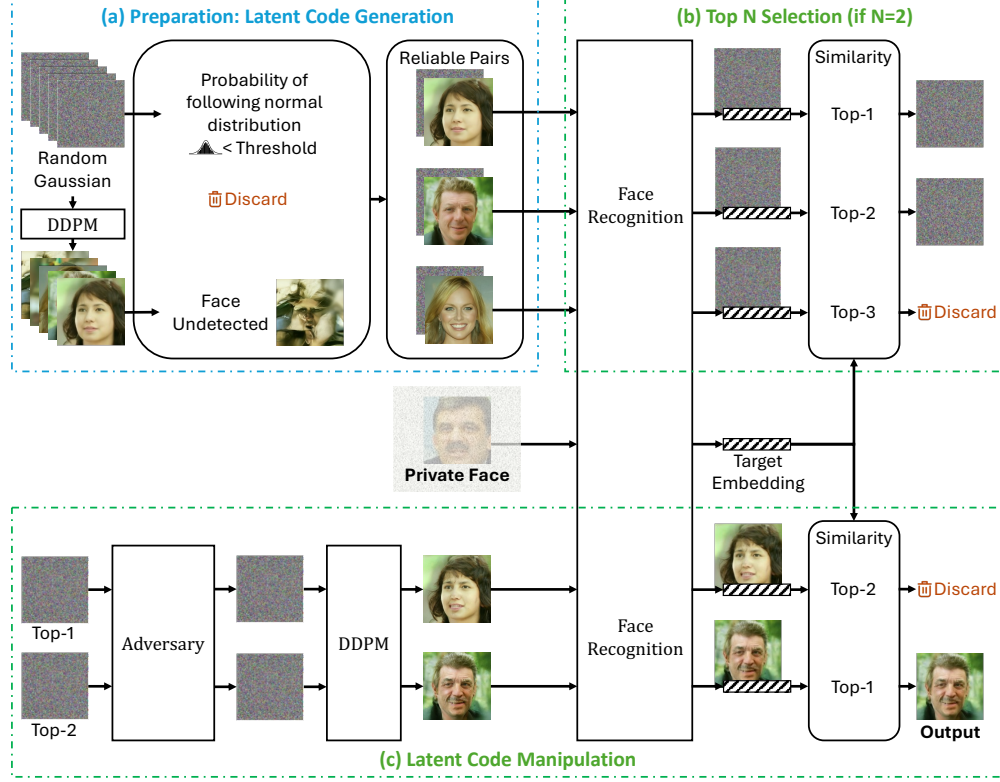


Figure 2. Framework of DiffUMI, reconstructing a facial image with the same identity of a private face, solely from its embedding. The generator, a denoising diffusion model (DDPM) [37], is pretrained independently, without prior knowledge of the private face or the target model. In Step (a), a set of highly reliable latent codes is generated, executed only once, and applicable for attacks on any target. In Step (b),  $N$  latent codes are selected based on the reconstruction embeddings most similar to the target embedding, serving as initialization for Step (c). In Step (c), these latent codes undergo adversarial refinement to progressively align the reconstructions with the target embedding. The final output is the reconstruction with the highest embedding similarity to the target. Notably, despite initial rankings, a higher-ranked latent code may yield suboptimal reconstruction after adversarial manipulation.

explicit facial features. Together, these criteria enhance the latent codes’ robustness against subsequent adversarial manipulations, minimizing distortions and artifacts.

**Step (b) – Top  $N$  Selection:** From the latent codes generated in Step (a), the top  $N$  are selected based on the embedding similarity between their reconstructions and the target embedding, as assessed by the target model. This ensures that the initial latent codes are well-aligned with the target identity, facilitating more effective adversarial manipulation in the next step.

**Step (c) – Latent Code Manipulation:** The selected latent codes from Step (b) undergo iterative refinement through adversarial manipulation to progressively align their reconstructions with the target identity. The final output is the reconstruction exhibiting the highest embedding similarity to the target. To enhance efficiency, we introduce a ranked adversary strategy that leverages the rankings from Step (b) to optimize the order of latent code manipulation. Additionally, this strategy allows for early termination once the attack objective is met, thereby reducing computational overhead while maintaining effectiveness.

## 2.2. Threat Model

The proposed DiffUMI framework functions as an attacker, evaluating the privacy vulnerabilities of a target face recognition model. Specifically, it assesses whether a facial image reconstructed solely from an embedding generated by the model can accurately resemble the identity of the original face associated with that embedding.

**Attack Knowledge:** The attacker’s knowledge is categorized based on access to the target model’s gradients, in addition to utilizing the embedding for the privacy attack:

- *White-box:* The attacker has access to both the target embedding and the model’s gradients.
- *Black-box:* The attacker only has access to the target embedding and can interact with the model via query-based feedback.

**Attack Objective:** The goal is to reconstruct a facial image that enables recognition of the target identity. To make this standard concrete and suitable for human evaluation, we define two criteria:

- The reconstructed image appears to be the same person as the target.

- The reconstructed image can help identify the target from a pool of identities.

However, human assessments require real-world user studies and are inherently subjective, varying across individuals. To provide a more objective measure, we also evaluate attack success using face recognition models. Since the attack relies solely on the target embedding and model, there is a risk of overfitting to adversarial artifacts rather than recovering genuine facial features. To mitigate this, we assess performance across multiple face recognition models beyond the target model. Formally, the attack objectives are:

- *During attack*: Maximize the similarity between the reconstructed image’s embedding and the target embedding in the target model (see Sec. 2.3).
- *During evaluation*: Ensure the reconstructed image is classified as the target identity by multiple models, even when compared against other images of the same person (excluding the one used for the target embedding) (see Sec. 4).

### 2.3. Problem Definition and Objective Function

Given a target face  $x^T$  and the embedding function of a target model  $F(\cdot)$ , DiffUMI seeks to reconstruct  $\hat{x} \approx x^T$  using only the feature embedding  $z^T = F(x^T)$ . However, in the context of privacy attacks, the goal is not for  $\hat{x}$  to be visually identical to  $x^T$ , but rather to share the same identity. Hence, the objective of DiffUMI is reformulated as:

$$F(\hat{x}) \approx F(x^T). \quad (1)$$

We define the reconstructed image  $\hat{x}$ , generated by applying a pretrained DDPM [37] as a generative function  $G(\cdot)$  to an initial Gaussian noise sample  $x_G$ :

$$\hat{x} = G(x_G). \quad (2)$$

The generator operates on a latent code  $x_G$  of size  $3 \times 256 \times 256$ , sampled from a Gaussian distribution. Thus, Eq. (1) is reformulated as:

$$F(\hat{x}) = F(G(x_G)) \approx F(x^T). \quad (3)$$

Randomly sampled latent codes generally do not satisfy Eq. (3). To address this, we introduce an adversarial attack to manipulate the latent code:

$$x'_G = x_G + \delta, \quad \text{s.t.} \quad \|\delta\|_p \leq \epsilon, \quad (4)$$

where  $\delta$  denotes the adversarial perturbation and  $\epsilon$  is the perturbation magnitude constrained by the  $L_p$ -norm. The objective is then formulated as:

$$F(G(x_G + \delta)) \approx F(x^T), \quad (5)$$

where  $x_G$  is drawn from a random Gaussian distribution.

Eq. (5) is satisfied when the similarity measure exceeds a predefined threshold  $\tau_F$ , indicating that the reconstructed image is classified as the same identity as the target.  $\tau_F$  is a parameter of face recognition models, set at the minimum equal error rate for standard face recognition tasks using

real facial images. While  $\tau_F$  serves as a criterion for evaluating attack success by verifying whether the reconstruction matches the target, it is not involved in attack optimization or required as attack knowledge. The objective function  $\mathcal{L}$  of DiffUMI is formulated as:

$$\begin{aligned} \hat{z} &= F(\hat{x}) = F(G(x_G + \delta)), \\ z^T &= F(x^T), \\ \mathcal{L} &= S(\hat{z}, z^T) = \frac{\hat{z} \cdot z^T}{\|\hat{z}\| \|z^T\|}, \end{aligned} \quad (6)$$

where  $S(\cdot, \cdot)$  denotes the function computing cosine similarity. DiffUMI aims to maximize  $\mathcal{L}$  by iteratively manipulating the latent code until:

$$\arg \max_{\hat{x}} \mathcal{L}. \quad (7)$$

### 2.4. Step (a) - Prepare: Latent Code Generation

We propose a two-stage approach for generating reliable latent codes as candidate initializations for DiffUMI (Fig. 15 of Appendix C). This strategy incorporates D’Agostino’s  $K^2$  test [43]–[45] to ensure that the selected latent codes conform to a normal distribution, referred to as Gaussian normality, and utilizes MTCNN [46] for face detection to guarantee that the generated latent codes produce discernible facial features.

The sequence of operations is deliberately structured: the  $K^2$  test is performed first, followed by face detection. This ordering is chosen because generating a new random Gaussian template incurs minimal computational cost, though a significant portion may fail the  $K^2$  test. Conversely, face detection is computationally more expensive, involving both image generation and verification. However, latent codes passing the  $K^2$  test are more likely to meet the face detection criterion, thereby optimizing efficiency.

This phase independently generates a set of reliable latent codes, which can be used for attacking any target. Since it is executed only once, the  $K^2$  test and face detection criteria can be applied rigorously to maximize the reliability of the generated latent codes.

**2.4.1. Normality Test via D’Agostino’s K-Square Statistic.** Reconstructing a facial image using DDPM requires a latent code of size  $3 \times 256 \times 256$  that follows a normal distribution (*i.e.*, Gaussian normality) [37]. The adherence of randomly generated latent codes to a normal distribution varies (Fig. 3) and generally correlates with reconstruction fidelity (higher is better). However, even initially high-normality codes degrade after manipulation (Eq. (4)), as demonstrated in Fig. 4. To mitigate this effect, we employ D’Agostino’s  $K^2$  test [43]–[45] to select latent codes with higher normality prior to manipulation, improving initial reconstruction quality and better preserving fidelity despite subsequent normality reduction.

Given a randomly generated latent code  $x_G$ , the  $K^2$  test function  $K(\cdot)$  quantifies deviations from normality based on skewness and kurtosis, producing a probability value:

$$p_K = K(x_G). \quad (8)$$



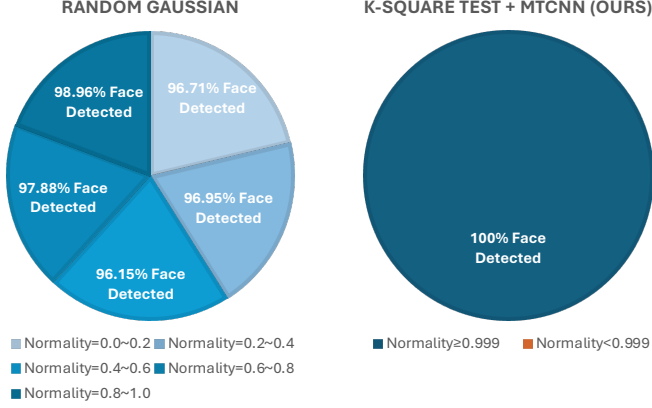


Figure 3. Reliability of randomly generated latent codes and the proposed strategy. The face detection confidence threshold is set to  $\tau_D = 0.99$ . As shown in the left subfigure, the Gaussian normality of randomly generated latent codes fluctuates, with higher normality generally leading to improved face detection rates, as indicated by darker regions. Our strategy guarantees 100% Gaussian normality, with  $p_K \geq 0.999$ , and consistently achieves a 100% face detection rate, with  $p_D \geq 0.99$ .

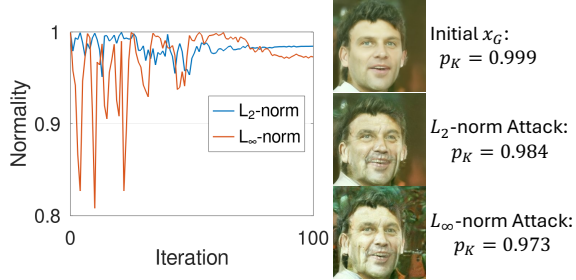


Figure 4. Degradation of Gaussian normality due to adversarial manipulation. The  $L_2$ -norm constrained adversary more effectively mitigates the decline in normality compared to the  $L_\infty$ -norm adversary, leading to enhanced reconstruction fidelity.

Latent codes are selected via a normality threshold  $\tau_K$ :

$$x_G \text{ is selected if } p_K \geq \tau_K, \quad (9)$$

where  $\tau_K$  is chosen sufficiently large as generating a random template remains computationally efficient. This selection process prioritizes latent codes closely adhering to Gaussian properties, enhancing robustness in subsequent steps.

**2.4.2. Face Detection via MTCNN.** We observe that randomly generated latent codes may fail to produce recognizable faces, as shown in Fig. 3 and Fig. 5. To ensure face presence in generated images, we utilize MTCNN [46], a deep learning-based face detection framework. Given a latent code  $x_G$ , we generate an image  $\hat{x}$  using DDPM  $G(\cdot)$ :

$$\hat{x} = G(x_G). \quad (10)$$

Next, we apply the detection function  $D(\cdot)$  to assess face presence, yielding a confidence score  $p_D$ :

$$p_D = D(\hat{x}). \quad (11)$$



Figure 5. Failed face generation using randomly generated latent codes, where MTCNN fails to detect a face, resulting in a detection confidence of  $p_D = 0$ .

Target	Initial	Final
Similarity ↑	0.2887	0.2831
Rank	Top-1	Top-2
		Top-2
		Top-1

Figure 6. An example where the initial latent code, resulting in the optimal initial reconstruction, fails to yield the best outcome following adversarial manipulation.

We define a threshold  $\tau_D$  for high-confidence face detection:

$$(x_G, \hat{x}) \text{ is selected if } p_D \geq \tau_D, \quad (12)$$

where  $\tau_D$  is set sufficiently high since most latent codes passing the  $K^2$  test also meet the face detection criterion. This ensures that only images with strong facial feature likelihoods are retained for further processing. The reconstructed facial image  $\hat{x}$  and  $x_G$  are stored together to enhance efficiency in the subsequent top  $N$  latent code selection step.

## 2.5. Step (b) - Top N Latent Code Selection

In Step (a), DiffUMI generates a set of  $V$  randomly sampled yet reliable latent codes and their corresponding reconstructions  $\{(x_{G_v}, \hat{x}_v)\}_{v=1}^V$ . Instead of processing all  $V$  candidates, it may select the one with the highest initial embedding similarity to the target, expecting lowest computational costs and distortions. However, the latent code yielding the best initial reconstruction may not always produce the optimal result after adversarial manipulation (Fig. 6). To address this, DiffUMI refines by choosing the top  $N$  latent codes from  $V$  (Fig. 16 of Appendix C), where increasing  $N$  improves model inversion performance but incurs higher computational overhead.

In particular, the  $V$  reliable pairs are input into the target model, which outputs  $V$  feature embeddings. Note that  $\hat{x}_v = G(x_{G_v})$ , previously computed and stored during Step (a) for face detection (Sec. 2.4.2), is used. Similarities, as described in Eq. (6), are calculated between these  $V$  embeddings and the target embedding. The top  $N$  embeddings, exhibiting the highest similarity, are retained for Step (c) as the initial latent codes for adversarial manipulation.

In the black-box setting, as outlined in Sec. 2.2, each of the  $(x_{G_v}, \hat{x}_v)$  pairs requires access to the target model, leading to a total of  $V$  queries in Step (b):

$$Q_{TopN} = V, \quad (13)$$

where  $V$  is the size of candidates from Step (a).

## 2.6. Step (c) - Latent Code Manipulation

As outlined in Sec. 2.3, a randomly sampled latent code rarely meets the attack objective of reconstructing a facial image that matches the target identity, even with the top-1 selection from Step (b). Therefore, refining the initial latent code is crucial to align the reconstructed face with the target. Existing methods manipulate latent codes starting from either the top-1 code (its limitation was discussed in Sec. 2.5) or a fusion of the top  $N$  selections [9]. However, we observe that fusion significantly reduces similarity compared to top-1 or top-2 selections. Moreover, relying on a single fused initialization may fail to ensure effective reconstruction, similar to the limitations of the top-1 approach. To address these issues, we propose the *Ranked Adversary* method, which retains the top  $N$  selection strategy while prioritizing latent codes with higher similarity for earlier adversarial manipulation. This approach includes early termination once the attack objective is met, optimizing computational efficiency without sacrificing effectiveness.

**2.6.1. Algorithm of Ranked Adversary.** The Ranked Adversary approach (Fig. 17 of Appendix C) begins the refinement process with the top-1 latent code  $x_{G_1}$ , applying an adversarial attack strategy to iteratively adjust  $x_{G_1}$  in pursuit of the objective defined in Eq. (7), guided by the objective function in Eq. (6). The process concludes when the similarity measure  $\mathcal{L}_1$  exceeds the predefined attack threshold  $\tau_A$ , indicating successful manipulation where the reconstructed image sufficiently matches the target. In such cases, the reconstructed image  $\hat{x}_1 = G(x_{G_1} + \delta_1)$  is produced as the final model inversion result. If  $\mathcal{L}_1$  remains below  $\tau_A$  after the maximum adversarial iterations  $t_{max}$ , the top-2 latent code  $x_{G_2}$  undergoes the same optimization process. If none of the top  $N$  latent codes achieve  $\mathcal{L}_n \geq \tau_A$ ,  $n = 1, \dots, N$ , the  $\hat{x}_n$  with the highest  $\mathcal{L}_n$  is selected as the final output.

As shown in Fig. 7, setting  $\tau_A = \tau_F$  may ensure successful target matching, but only on the target model. Increasing  $\tau_A$  beyond  $\tau_F$  enhances robustness, particularly on the test model. However, excessively high  $\tau_A$  or the absence of early stopping may lead to the attack proceeds beyond the point of achieving  $\mathcal{L} > 0.98$ , yielding negligible gains in optimization while exacerbating overfitting and diminishing generalization to the test model, as achieving such a threshold often forces adversarial manipulation to introduce artifacts that overfit the target model and exhaust all attack iterations. Therefore, we define  $\tau_A$  for attacking the specific model as the maximum embedding similarity achievable by real facial images in this model, which represents the best case where without any overfitting artifacts, the model can achieve the best similarity within the same identity. To

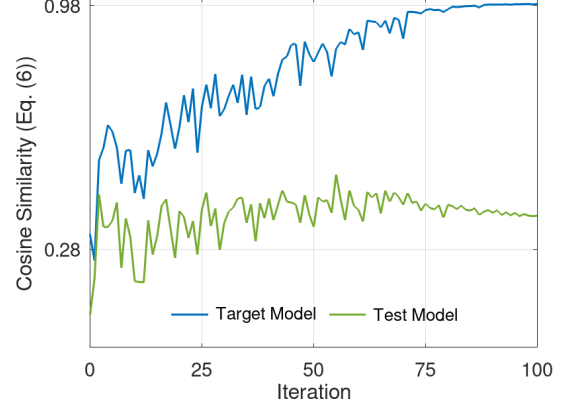


Figure 7. Overfitting and inefficiency in objective function maximization (Eq. (6)) without the proposed ranked adversary strategy. The target model (in blue), PartialFace [7], uses a predefined similarity threshold  $\tau_F = 0.28$  (minimum equal error rate), where embedding similarity above  $\tau_F$  indicates identity matching. The blue curve depicts the attack process with a maximum of 100 iterations and an attack threshold  $\tau_A = 1$ , effectively disabling the ranked adversary strategy due to the unattainable objective. The tick at 0.98 marks the highest embedding similarity typically observed between real facial images of the same identity (LFW [40]) under the PartialFace model, which we designate as the optimal  $\tau_A$  to terminate the adversary, thereby preventing overfitting and reducing computational cost. ArcFace [5] is used to evaluate overfitting. Here, test model validation is solely for performance assessment. It cannot be incorporated into objective optimization, as the attack operates solely on the target embedding, without access to the original image or its embedding of the test model.

compute this, we feed the entire face dataset  $\mathcal{X}_{real}$  (all real faces) into the target model, then compute the maximum similarity between any two images. The similarity between different identities is usually lower than that between the same identity. Thus, we define  $\tau_A$  as follows:

$$\tau_A = \max S(F(x^i), F(x^j)), \quad x^i, x^j \in \mathcal{X}_{real}, \quad (14)$$

where  $F(\cdot)$  is the embedding function of the target model, and  $S(\cdot, \cdot)$  is the cosine similarity function.

**2.6.2. Adversarial Attack.** Successful manipulation of unconditional diffusion generation does not require a new adversarial attack backbone, but instead relies on a fine-grained algorithm [38]. Ranked Adversary employs APGD [36] in the white-box setting and GreedyPixel [38] in the black-box setting. While various adversarial attack algorithms can be adapted for latent code manipulation within our framework, our empirical observation shows that only these fine-grained methods yield satisfactory reconstructions without introducing significant distortions or artifacts. This is due to the sensitivity of the diffusion model’s latent space, where even small perturbations can notably degrade reconstruction quality.

Additionally, we evaluated two alternative black-box attacks: Square attack [47], which uses the same  $L_2$ -norm constrained perturbation magnitude  $\epsilon$  as APGD, and BruSLe attack [48], a pixel-wise attack like GreedyPixel that enforces sparsity constraints rather than directly constraining  $\epsilon$ . However, both methods performed inferiorly compared to GreedyPixel, as detailed in Sec. 5.2.

**2.6.3. Query Efficiency in the Black-Box Setting.** In the black-box setting, the query cost for latent code manipulation arises from calculating the loss values in Eq. (6), which requires querying the target model to obtain feature embeddings of the reconstructed images. As a result, the total query cost is proportional to the number of iterations performed during adversarial manipulation. The query cost for optimizing a single latent code is given by:

$$Q_{Adv} = t_{max}, \quad (15)$$

where  $t_{max}$  represents the upper limit on the number of adversarial attack iterations.

Since the Ranked Adversary framework processes up to  $N$  selected latent codes, the total query cost for the latent code manipulation phase is bounded by:

$$t_{max} \leq Q_{Adv} \leq N \times t_{max}. \quad (16)$$

Including the query cost incurred during the top  $N$  latent code selection in Step (b) (Eq. (13)), the overall query complexity of DiffUMI is:

$$Q = Q_{TopN} + Q_{Adv}. \quad (17)$$

In practical black-box attack scenarios, a predefined query budget  $Q_{max}$  is often imposed as a hard constraint. Under this restriction, the maximum number of iterations per adversarial attack process must satisfy:

$$t_{max} = \lfloor \frac{Q_{max} - V}{N} \rfloor, \quad (18)$$

where  $\lfloor \cdot \rfloor$  is the floor function, returning the largest integer not exceeding the input, and  $V$  represents the size of latent code set in Step (a) used to select the top  $N$  latent codes.

### 3. Out-Of-Domain Detection

Deep learning models map both in-domain and out-of-domain inputs to the same feature space, making it challenging to distinguish between them based solely on embeddings. For example, in a face recognition system, both a human face and a non-face object (e.g., a cat) produce embeddings with identical dimensions and numerical ranges, despite their semantic differences. To address this, we leverage the high-fidelity reconstruction capabilities of our model inversion framework to develop an application of OOD that identifies out-of-domain inputs based on their embeddings. As illustrated in Fig. 8, two common failure cases arise in model inversion, sometimes concurrently. In-domain inputs, such as facial images, typically yield accurate, recognizable reconstructions. In contrast, out-of-domain inputs are more likely to exhibit at least one of these failure cases. Hence, we classify an input as out-of-domain if either failure is detected.

**Case 1: Reconstruction fails to match the target input across all test models except the target model.**

We define *Test Models* as those distinct from the target model. For instance, if the target model is FaceNet [4], alternative models like ArcFace, DCTDP, and PartialFace

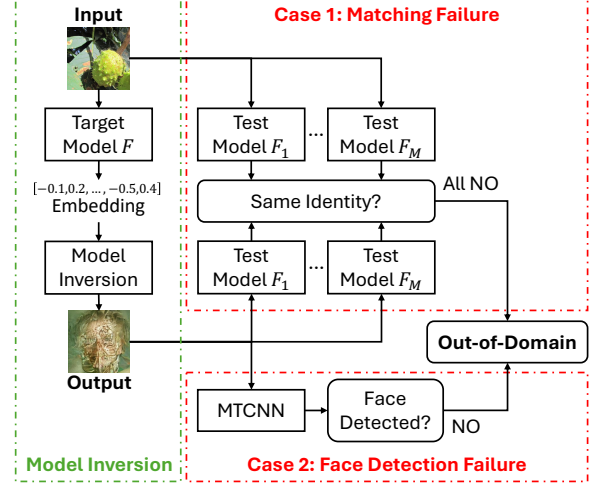


Figure 8. The proposed out-of-domain detection (OODD) framework defines two failure scenarios in model inversion, both indicating out-of-domain inputs: (i) **Matching Failure**, where the reconstructed image fails to match the target input across all test models except the target model, and (ii) **Face Detection Failure**, where the reconstructed image lacks identifiable facial features.

can serve as test models [5]–[7]. This failure case is assessed using test models because the reconstruction tends to overfit the target model, resulting in a successful match when evaluated on the target model alone.

Given an input image  $x$  and the embedding function  $F(\cdot)$  of the target model, we aim to identify whether  $x$  is an out-of-domain input using its embedding  $z = F(x)$ . First, we apply DiffUMI to obtain the reconstruction  $\hat{x}$ :

$$\hat{x} = \arg \max_{\hat{x}} S(F(\hat{x}), F(x)). \quad (19)$$

Next, using  $M$  test models  $F_1, \dots, F_M$ , we determine whether  $\hat{x}$  matches  $x$  in these models:

$$OOD_1 = \begin{cases} \text{True,} & \text{if } \forall m \in \{1, \dots, M\}, \\ & S(F_m(\hat{x}), F_m(x)) < \tau_{F_m}, \\ \text{False,} & \text{otherwise.} \end{cases} \quad (20)$$

where  $\tau_{F_m}$  is the predefined similarity threshold for the minimum equal error rate in face recognition.

Note that matching failure is defined only when the reconstruction fails to match the target across “all” test models (“ALL NO” in Fig. 8). Defining failure based on “any” mismatch, implying that an in-domain reconstruction should match the target across all test models, would be an overly idealized assumption, which consequently significantly increases the false detection of in-domain inputs as out-of-domain. For instance, our “all” strategy yields a 3.4% matching failure rate in detecting LFW [40] facial images on the FaceNet model [4], whereas the “any” criterion increases this error to 26.6%.

**Case 2: The reconstruction lacks a detectable face, even if it matches the target input.**

In this case, a successful match is likely attributable to adversarial artifacts rather than accurate identity recon-



TABLE 2. CONFIGURATION OF TARGET MODELS, WHITE-BOX DIFFUMI, AND OODD SETTINGS.

Face Recognition Model $F(\cdot)$	$\tau_F$	Latent Code Generation (ours)			Latent Code Manipulation (ours)					OODD (ours) $\tau_D^1$
		Volume $V$	$\tau_K$	$\tau_D^1$	Top $N$	$t_{max}$	$\tau_A$	Norm	$\epsilon$	
FaceNet [4]	0.40	1,000	0.999	0.999	3	100	0.99	$L_2$	25	0.9933
ArcFace [5]	0.23						0.99		35	
DCTDP [6]	0.26						0.98		35	
PartialFace [7]	0.28						0.98		35	

<sup>1</sup> A higher detection confidence threshold is applied during latent code generation to ensure reliable initialization, as generating a random Gaussian template is computationally efficient. All parameters refer to Tab. 14 in Appendix C for notation definitions.

struction. We utilize MTCNN  $D(\cdot)$  [46] with a detection threshold  $\tau_D$ :

$$OOD_2 = \begin{cases} \text{True,} & \text{if } p_D = D(\hat{x}) < \tau_D, \\ \text{False,} & \text{otherwise.} \end{cases} \quad (21)$$

Here,  $\tau_D$  may differ from the latent code generation threshold, as a higher  $\tau_D$  in OODD increases false positives by classifying in-domain inputs as out-of-domain.

The final OODD decision is determined by the logical disjunction of both criteria:  $OOD = OOD_1 \vee OOD_2$ , where  $\vee$  represents the logical disjunction (OR) operator.

## 4. Experimental Settings

**Face Recognition Models.** We assess model inversion attacks on two widely used models, FaceNet [4] and ArcFace [5], and two privacy-preserving models, DCTDP [6] and PartialFace [7], which are designed to mitigate inversion attacks. The decision thresholds for all models are set at the minimum equal error rate, as outlined in Tab. 2.

**Datasets.** We evaluate model inversion attacks and OODD using three datasets: LFW [40] and CelebA-HQ [39] for privacy attacks, and ImageNet [41], a non-face dataset, for OODD. Each dataset contains 1,000 samples. For LFW and CelebA-HQ, we select 10 images from each of 100 distinct identities for computing Type II accuracy. Datasets such as FFHQ [49], which lack identity annotations, are thus unsuitable. As our approach is training-free, large-scale datasets are unnecessary for evaluation.

LFW, which is independent of both face recognition model and generator training, serves as our primary evaluation dataset. In contrast, CelebA-HQ shares the training distribution of the generator but is not used to train any recognition models, enabling an assessment of whether such prior knowledge improves attack performance. The face recognition models themselves are trained on datasets distinct from both evaluation sets: FaceNet [4] and DCTDP [6] are trained on VGGFace2 [50], while ArcFace [5] and PartialFace [7] are trained on MS1MV2 [51].

**Benchmark.** We benchmark DiffUMI against MAP<sup>2</sup>V [9] in the white-box setting. As shown in Tab. 1, MAP<sup>2</sup>V is the most recent and most relevant state-of-the-art baseline, sharing our training-free setup and focus on open-set face recognition. While DiffUMI also outperforms training-dependent attacks [17], [27], as detailed in Appendix A, they are not our primary baseline due to their limitation that

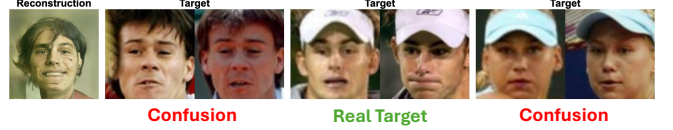


Figure 9. A challenging example of confusing choices in questionnaire design for identifying target identities from reconstructions. We provided two photos for each of three target options, with the order randomized. Some participants remarked that the two male choices (the first and second targets) looked too similar, making them appear to be the same person.

each target model requires a separately trained generator, resulting in substantial computational overhead.

**Experimental Setup.** The configurations for the white-box DiffUMI attack and OODD are provided in Tab. 2, while the settings for the black-box DiffUMI attack are outlined in Fig. 11. Specifically,  $\tau_F$  is the cosine similarity threshold in face recognition, optimized to minimize the equal error rate.  $\tau_K$  and  $\tau_D$  in latent code generation are set close to the upper bound (1.0) to ensure highly reliable initialization.  $\tau_A$  is defined in Eq. (14). For OODD,  $\tau_D$  is set to maintain a maximum false positive rate of 5% for in-domain inputs. In the ablation studies, we vary one parameter at a time while holding all others constant. Experiments are conducted on an NVIDIA A100 40GB GPU.

**Evaluation Protocols.** Model inversion attacks and OODD are executed directly on the target model, often leading to outputs that overfit this model. Consequently, exceptional performance on the target model may stem from adversarial artifacts rather than the accurate reconstruction of facial features, a scenario classified as a failure in Sec. 3. To thoroughly evaluate the effectiveness of privacy attacks and OODD, we adopt the following evaluation protocols.

**User Study:** Following the attack objectives defined in Sec. 2.2, we asked ten participants to evaluate three groups of images, specifically, to identify the target identities based on reconstructions and to judge whether each reconstruction depicts the same person as the target identity. Each evaluation included 60 images in total, drawn from two datasets. To make the questionnaire sufficiently challenging, we introduced confusing choices by selecting identities whose facial embeddings have a similarity of approximately  $80\% \times \tau_F$  to the target identity (as shown in Fig. 9), where a similarity of  $\geq \tau_F$  is considered a match.

**Type II Accuracy:** Mai *et al.* [15] introduced Type I and Type II accuracy as metrics for evaluating the effectiveness

of privacy attacks. In this study, we focus on Type II accuracy, which measures the similarity between reconstructed facial images and facial images from the target identity, but excluding the target facial image. This metric provides a more stringent evaluation by reducing the risk of overfitting to the target image, whose embedding serves as the reference during the attack process. Specifically, Type II accuracy is the rate at which the reconstructed image  $\hat{x}$  matches facial images  $x^{j \neq T}$  from the target identity but different from the target face  $x^T$ :

$$\text{Type II} = \frac{\sum_{i=1}^I \sum_{j=1}^J \mathbb{I}(S(F(\hat{x}_i), F(x_i^{j \neq T})) \geq \tau_F)}{I}, \quad (22)$$

where  $x_i^j \neq x_i^T$  but shares the same identity.  $\mathbb{I}(\cdot)$  and  $S(\cdot, \cdot)$  represent the indicator and cosine similarity functions, respectively.  $J$  denotes the number of other facial images associated with the identity of  $x_i^T$ ,  $I$  is the total number of attack samples, and  $\tau_F$  is the face recognition similarity threshold. A higher Type II accuracy reflects a stronger inversion attack from the attacker’s perspective and indicates greater vulnerability from the privacy protection standpoint.

For further validation, we also report Type I accuracy results in Appendix D, which reflect the strength of the inversion attack and highlight potential privacy vulnerabilities.

**OODD Rate:** The OODD rate quantifies the failure rate of model inversion, based on the two failure cases outlined in Sec. 3. For in-domain inputs, a lower OODD rate is preferable, indicating fewer inversion failures. Conversely, for out-of-domain inputs, a higher OODD rate is desirable, reflecting improved detection performance.

**Joint Evaluation Using Target and Test Models:** Test models, as defined in Sec. 3, refer to those that differ from the target model. Evaluating on these models helps mitigate overfitting to the target model. In this study, four face recognition models are utilized, with one designated as the target model and the remaining three as test models for each attack. However, testing exclusively on test models may introduce bias, as the test models vary for each target model, leading to an unfair comparison of model vulnerability to privacy risks. To address this, we adopt a joint evaluation approach that includes both the target and test models, ensuring consistent evaluation across all models. This approach mitigates the impact of overfitting to the target model alone.

Note that OODD is evaluated exclusively on test models, as detailed in Sec. 3. This is because out-of-domain inputs can only be matched with their reconstruction (similar to Type I accuracy), and such matching is always successful on the target model due to overfitting.

## 5. Performance

### 5.1. White-Box Model Inversion

We assess the vulnerability of face recognition models to privacy threats using DiffUMI in the white-box setting. As shown in Tab. 3, all four models, including privacy-preserving variants, fail to prevent privacy leakage. DiffUMI

TABLE 3. TYPE II ACCURACY (%) OF DIFFUMI ACROSS FOUR FACE RECOGNITION MODELS.

Dataset	Target Model	Test Model				Avg.
		FaceNet	ArcFace	DCTDP	PartialFace	
CelebA	FaceNet	96.93	80.12	84.19	77.67	84.73
	ArcFace	92.76	99.06	94.23	92.84	94.72
	DCTDP	87.43	89.62	96.52	86.61	90.05
	PartialFace	80.06	87.29	85.82	96.14	87.33
LFW	FaceNet	98.56	66.23	73.31	59.92	74.51
	ArcFace	91.03	99.64	95.88	92.33	94.72
	DCTDP	90.34	95.37	99.44	86.31	92.87
	PartialFace	75.39	85.44	77.99	98.84	84.42

Gray cells indicate cases where the target and test models are identical. Green and Red highlight the most and least secure models, respectively, based on the lowest and highest Type II accuracy.

FaceNet [4] offers the strongest privacy protection, surpassing even the privacy-preserving PartialFace [7] and DCTDP [6], while ArcFace [5] provides the weakest.

TABLE 4. USER STUDY RESULTS. TARGET MODEL IS THE PRIVACY-PRESERVING MODEL, PARTIALFACE [7].

Question	Dataset	Accuracy (%) $\uparrow$
Find target	CelebA	80.0 (16.0 / 20.0)
Find target	LFW	79.5 (15.9 / 20.0)
Same person?	LFW	73.0 (14.6 / 20.0)

achieves Type II accuracy between 74.51% and 94.72% across all models and datasets. Notably, the oldest standard model, FaceNet, demonstrates the highest resistance, highlighting the limitations of existing privacy-preserving techniques. Furthermore, the user study results in Tab. 4 reflect a high success rate of our privacy attack, as confirmed through direct human evaluation.

Compared to the benchmark, as shown in Tab. 5, DiffUMI consistently surpasses MAP<sup>2</sup>V in Type II accuracy and achieves similarity values closer to the attack threshold  $\tau_A$  across all scenarios. The “random” rows represent unguided diffusion model generations, which fail to match targets, emphasizing the necessity of a structured inversion approach. The visualization in Fig. 10 further highlights DiffUMI’s superiority, demonstrating higher identity recovery accuracy than MAP<sup>2</sup>V. Our approach produces full headshot-style reconstructions, achieving optimal granularity for identity recovery.

### 5.2. Black-Box Model Inversion

We evaluate DiffUMI in the black-box setting using various adversarial attack algorithms and compare its performance to the white-box counterpart. As shown in Fig. 11 (columns 3 vs. 10), black-box DiffUMI (GreedyPixel) achieves Type II accuracy slightly lower than its white-box version (APGD) but incurs substantially higher computational costs in queries. Moreover, Fig. 11 (columns 4–6) demonstrates that only fine-grained attacks like GreedyPixel

TABLE 5. PERFORMANCE OF OUR DIFFUMI ACROSS FOUR FACE RECOGNITION MODELS, COMPARED TO THE BENCHMARK ATTACK MAP<sup>2</sup>V [9].

Dataset	Attack	Target Model							
		FaceNet		ArcFace		DCTDP		PartialFace	
		Type II <sup>3</sup> (%) ↑	Similarity <sup>4</sup> ↑	Type II (%) ↑	Similarity ↑	Type II (%) ↑	Similarity ↑	Type II (%) ↑	Similarity ↑
CelebA	Original <sup>1</sup>	97.00	$\tau_A = 0.99$	99.09	$\tau_A = 0.99$	96.62	$\tau_A = 0.98$	95.65	$\tau_A = 0.98$
	Random <sup>2</sup>	4.50	0.0822	1.73	0.0304	4.74	0.0771	9.72	0.1404
	MAP <sup>2</sup> V	69.19	0.9248	84.61	0.8175	81.12	0.8135	80.84	0.7758
	Ours	<b>84.73</b>	<b>0.9920</b>	<b>94.72</b>	<b>0.9898</b>	<b>90.05</b>	<b>0.9818</b>	<b>87.33</b>	<b>0.9818</b>
LFW	Original <sup>1</sup>	98.60	$\tau_A = 0.99$	99.62	$\tau_A = 0.99$	99.47	$\tau_A = 0.98$	98.89	$\tau_A = 0.98$
	Random <sup>2</sup>	0.91	0.0154	0.06	0.0046	0.09	0.0129	0.54	0.0568
	MAP <sup>2</sup> V	73.14	0.9337	79.22	0.7723	83.05	0.7787	80.41	0.7636
	Ours	<b>74.51</b>	<b>0.9917</b>	<b>94.72</b>	<b>0.9834</b>	<b>92.87</b>	<b>0.9774</b>	<b>84.42</b>	<b>0.9794</b>

<sup>1</sup> Upper bound corresponding to true target faces.

<sup>2</sup> Lower bound referring to randomly generated facial images without a specific strategy.

<sup>3</sup> Average Type II accuracy across four test models [4]–[7], as per the joint evaluation using target and test models outlined in Sec. 4.

<sup>4</sup> Cosine embedding similarity between target and reconstruction, computed in the target model, with values closer to  $\tau_A$  indicating better performance.

**Bold** indicates the highest performance among attack methods, with our approach exceeding MAP<sup>2</sup>V attack by up to 15.54% in Type II accuracy.

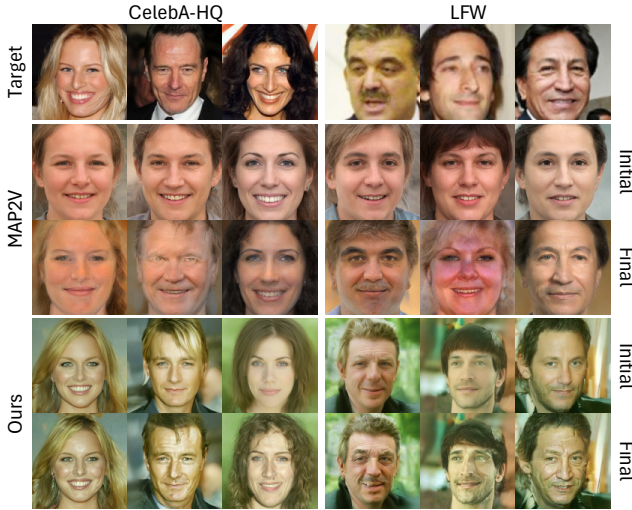


Figure 10. Illustrations of DiffUMI generation versus MAP<sup>2</sup>V [9] in the white-box setting. DiffUMI achieves superior identity recovery accuracy, producing full headshot-style reconstructions that maximize identity granularity. “Initial” denotes pre-manipulation generations. The target model in this figure is the privacy-preserving PartialFace [7].

effectively manipulate diffusion model latent codes, whereas coarser black-box attacks (e.g., Square [47] and BruSLe [48]) fail. Comparing columns 6–10, increasing GreedyPixel’s query budget and sparsity allowance enhances model inversion accuracy by modifying more pixels.

### 5.3. Out-Of-Domain Detection

We evaluate the proposed OOD framework on both face and non-face datasets across four models. As shown in Fig. 12, the model inversion outputs for out-of-domain inputs primarily fall into the two failure cases described in Sec. 3, each of which serves as an out-of-domain indicator within our framework. The results in Tab. 6 demonstrate a low detection rate for face datasets (LFW and CelebA-HQ)

TABLE 6. OUT-OF-DOMAIN DETECTION RATE (%) ACROSS MODELS AND DATASETS.

Data	FaceNet	ArcFace	DCTDP	PartialFace
CelebA ↓	2.3	4.4	4.9	1.7
LFW ↓	4.9	3.4	3.7	1.4
ImageNet ↑	91.3	98.9	95.5	95.2

Our OOD framework achieves high detection rates for out-of-domain inputs while maintaining low error for in-domain inputs.

and a high detection rate for non-face inputs (ImageNet), validating the effectiveness of our detection.

### 5.4. Prior Knowledge on Generator Training

We assess whether incorporating prior knowledge into the generator enhances reconstruction performance. To this end, we train a DDPM on CelebA-HQ and attack target identities also drawn from CelebA-HQ (with disjoint identities), ensuring that the generator and target images share the same distribution. Results are compared against attacks on the LFW dataset. As shown in Tab. 3, Tab. 4 and Tab. 6, it is surprising that performance varies inconsistently between CelebA-HQ and LFW across both privacy attacks and OOD tasks. These findings indicate that, in training-free, open-set model inversion, prior knowledge of the data distribution has minimal impact on reconstruction accuracy. Even with shared distributions, the generator cannot reliably synthesize identity-consistent images without a well-defined identity-guided mechanism.

## 6. Ablation Study

We conduct ablation studies on the proposed DiffUMI and OOD using the LFW [40] and ImageNet [41] datasets, with the privacy-preserving PartialFace [7] model as the target. Type II accuracy is averaged across all four face recognition models for joint evaluation, as outlined in Sec. 4.













Pre-Manipulation White-box Attack			Black-Box Attack						
1.Target	2.Top-1 Initial	3.APGD	4.Square	5.BruSLe	6.GreedyPixel	7.GreedyPixel	8.GreedyPixel	9.GreedyPixel	10.GreedyPixel
									
Magnitude $\epsilon$	N/A	L2-norm $\epsilon = 35$	L2-norm $\epsilon = 35$	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Sparsity		Unlimited	Unlimited	30%	10%	10%	10%	10%	Unlimited
Max. Queries	1,000	1,100	20,000	20,000	20,000	50,000	100,000	200,000	200,000
Type II (%) $\uparrow$	6.5	85.4	1.1	32.2	41.1	67.8	76.7	80.0	81.1

Figure 11. Performance of DiffUMI in the Black-Box Setting. Black-box DiffUMI achieves slight lower reconstruction accuracy to its white-box counterpart (columns 3 vs. 10) in terms of Type II accuracy. Among black-box attack methods (columns 4–6), the fine-grained GreedyPixel algorithm, which introduces adversarial patterns akin to white-box attacks, is the only viable approach for manipulating diffusion model latent codes. Comparing columns 6–10, increasing the attack budget (query or sparsity) enhances model inversion accuracy. The target model is the privacy-preserving PartialFace [7], and the test model is ArcFace [5].



Figure 12. Examples of two failure cases in the DiffUMI attack, which serve as indicators of OOD. The occurrence of either failure case signals an out-of-domain input: (i) **Matching Failure**, where the reconstructed image fails to align with the target input across all test models except the target model, and (ii) **Face Detection Failure**, where the reconstructed image lacks identifiable facial features. The target model in this figure is the privacy-preserving PartialFace [7].

In each study, we vary a single parameter while holding all others constant to isolate its impact on performance, with parameter settings provided in Tab. 2.

## 6.1. Model Inversion (DiffUMI)

**Reliability of Latent Codes.** We propose a latent code generation strategy in Sec. 2.4, combining D’Agostino’s  $K^2$  test with MTCNN to improve latent code reliability [43]–[46]. As shown in Tab. 7, our approach achieves the highest latent code reliability, leading to superior performance.

**Top  $N$  Selection.** As detailed in Sec. 2.5, DiffUMI does not necessitate processing all latent codes. While optimal initialization typically improves reconstruction accuracy, it does not always yield the best results after manipulation. Consequently, we adopt a top  $N$  selection strategy, where increasing  $N$  enhances reconstruction accuracy at the expense of higher computational cost. Based on our evaluation, we recommend  $N = 3$  as an optimal trade-off, achieving strong attack performance with manageable computational overhead, as shown in Tab. 8.

TABLE 7. PERFORMANCE ACROSS DIFFERENT LATENT CODE RELIABILITIES.

Strategy	Time (s) $\downarrow$	Type II (%) $\uparrow$	OODD (%) $\downarrow$
Random Gaussian	290	82.78	3.1
$K^2$ Test	293	82.47	2.4
$K^2$ Test + MTCNN	<b>256</b>	<b>84.42</b>	<b>1.4</b>

**Bold** indicates the superiority of our strategy.

TABLE 8. PERFORMANCE (%) FOR VARYING VALUES OF  $N$ .

Top $N$	Time (s) $\downarrow$	Type II (%) $\uparrow$	OODD (%) $\downarrow$
1	<b>189</b>	81.06	1.9
3	256	84.42	1.4
5	289	<b>85.21</b>	<b>0.9</b>

**Bold** indicates the highest performance, showing that increasing  $N$  enhances reconstruction accuracy but increases computational cost.

TABLE 9. PERFORMANCE (%) WITH AND WITHOUT THE RANKED ADVERSARY STRATEGY. “NO” INDICATES THAT ALL SELECTED  $N$  LATENT CODES ARE PROCESSED, AND THE RECONSTRUCTION WITH THE HIGHEST SIMILARITY IS SELECTED AS THE FINAL OUTPUT.

Ranked	Time (s) $\downarrow$	Type II (%) $\uparrow$	OODD (%) $\downarrow$
<b>✓</b>	<b>256</b>	84.42	1.4
<b>✗</b>	530	<b>85.31</b>	<b>1.0</b>

**Bold** indicates the highest performance, demonstrating that the proposed ranked adversary marginally reduces attack accuracy while significantly enhancing time efficiency.

**Ranked Adversary.** We propose the Ranked Adversary strategy in Sec. 2.6 to efficiently identify a satisfactory, rather than optimal, solution for model inversion. By prioritizing sequential initialization and allowing early termination once a predefined condition is met, this strategy reduces unnecessary computations. As shown in Tab. 9, it significantly lowers computational cost with only a minor trade-off in Type II accuracy and OODD performance, effectively balancing efficiency and attack performance.

**Attack Threshold.** We define the attack threshold  $\tau_A$  in Sec. 2.6.1 as the criterion for successful manipulation, where the reconstructed image achieves sufficient similarity to the

TABLE 10. PERFORMANCE (%) ACROSS VARYING ATTACK THRESHOLD  $\tau_A$ .

Threshold $\tau_A$	Time (s) ↓	Similarity ↑	Type II (%) ↑	OODD (%) ↓
0.97	<b>213</b>	0.9712	84.23	1.9
0.98	256	0.9794	84.42	<b>1.4</b>
0.99	442	<b>0.9861</b>	<b>84.98</b>	1.8

**Bold** indicates the highest performance, showing that increasing  $\tau_A$  enhances Type II accuracy but substantially raises computational cost and the risk of OODD.

TABLE 11. PERFORMANCE (%) UNDER VARYING PERTURBATION CONSTRAINTS.

Norm	Magnitude $\epsilon$	Time(s) ↓	Similarity ↑	Type II (%) ↑	OODD (%) ↓
$L_2$	30	455	0.9707	81.49	1.5
$L_2$	35	256	0.9794	84.42	<b>1.4</b>
$L_2$	40	<b>191</b>	<b>0.9816</b>	<b>86.11</b>	3.2
$L_\infty$	0.15	260	0.9791	75.28	3.1

**Bold** denotes the highest performance, indicating that the  $L_2$ -norm constraint yields the best results. Increasing  $\epsilon$  accelerates the attack and improves Type II accuracy, but also introduces more noise, increasing the risk of OODD.

target. This threshold also serves as the early termination criterion in the ranked adversary strategy.  $\tau_A$  must satisfy  $\tau_A \gg \tau_F$  for robustness. However, excessively high values of  $\tau_A$  (e.g.,  $\tau_A = 1$ ) lead to unnecessary computational cost and the risk of overfitting.

For the target model, PartialFace [7], used in this ablation study, the maximum embedding similarity achievable by real facial images is 0.98. As shown in Tab. 10,  $\tau_A < 0.98$  results in faster model inversion, whereas  $\tau_A > 0.98$  improves Type II accuracy at the cost of increased computational overhead. Additionally, higher  $\tau_A$  values raise the risk of being detected as out-of-domain due to greater distortion in the reconstructed images. These results demonstrate that our proposed strategy (Sec. 2.6.1), which defines  $\tau_A$  as the maximum embedding similarity achievable by real facial images in the target model, is optimal.

**Perturbation Constraint.** As discussed in Fig. 4, the  $L_2$ -norm constrained adversary better preserves normality compared to the  $L_\infty$ -norm adversary, leading to improved reconstruction fidelity. This is further validated by Tab. 11, where  $L_2$ -norm ( $\epsilon = 35$ ) and  $L_\infty$ -norm ( $\epsilon = 0.15$ ) achieve similar similarity values, yet the  $L_2$ -norm attack demonstrates significantly superior performance. Additionally, Tab. 11 and Fig. 13 show that while increasing the attack magnitude  $\epsilon$  enhances Type II accuracy and accelerates the attack, it also introduces more artifacts, raising the likelihood of detection as out-of-domain inputs.

## 6.2. Out-of-Domain Detection (OODD)

We propose OODD framework in Sec. 3, integrating two failure cases: matching failure and face detection failure. Both criteria must be considered together, as relying on a single strategy alone is insufficient for effective OODD. As shown in Tab. 12 (see red values), using only one strategy

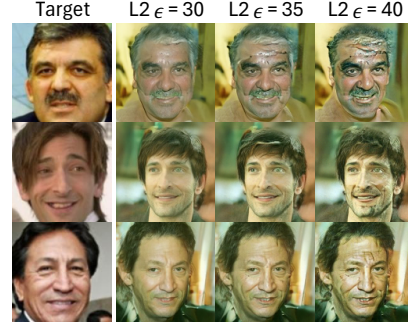


Figure 13. Larger perturbation magnitudes lead to increased distortion due to more pronounced noise and diminished naturalness in the generation.

TABLE 12. OODD RATE (%) WHEN JOINTLY CONSIDERING BOTH FAILURE CASES (OURS) VERSUS INDIVIDUAL STRATEGIES.

Target Model	Dataset	Matching Failure	Detection Failure	Ours
FaceNet	CelebA	0.6	1.7	2.3
	LFW	3.4	1.7	4.9
	ImageNet	<b>49.6</b>	<b>57.9</b>	91.3
ArcFace	CelebA	0.0	4.4	4.4
	LFW	0.2	3.2	3.4
	ImageNet	<b>1.6</b>	98.4	98.9
DCTDP	CelebA	0.0	4.9	4.9
	LFW	0.2	3.6	3.7
	ImageNet	<b>20.9</b>	<b>81.6</b>	95.5
PartialFace	CelebA	0.1	1.6	1.7
	LFW	0.8	0.6	1.4
	ImageNet	<b>55.3</b>	<b>57.0</b>	95.2

**Red** indicates instances of failed detection (insufficient detection rates), highlighting the superiority of our joint strategy over individual strategies.

results in an OODD rate of approximately 50% for out-of-domain inputs (i.e., ImageNet), highlighting its limitations.

## 7. Conclusion

In this paper, we introduce DiffUMI, the first approach to leverage diffusion models for unconditional facial image generation, achieving a training-free model inversion attack for open-set face recognition models. This eliminates the need for training separate generators for different models, addressing a key limitation of previous methods. Additionally, we propose OODD, a novel out-of-domain input detection framework that operates solely on feature embeddings. Our results demonstrate that DiffUMI achieves state-of-the-art performance in model inversion attacks, making it a powerful tool for evaluating the privacy vulnerabilities of face recognition models. Moreover, we are the first to systematically analyze the impact of latent code reliability on generation fidelity and propose an automated method for selecting highly reliable latent codes. We also provide new insights into the most effective adversarial attack strategies for manipulating latent space, rather than focusing on specific attack algorithms.



For future work, we aim to extend DiffUMI to broader domains, including classification models for datasets such as CIFAR-10, ImageNet, and medical imaging. Additionally, we intend to evaluate the potential defenses against our model inversion attack. We also plan to develop objective evaluation metrics to replace human assessment, as subjective judgments, such as “Do these two images belong to the same person?”, are inherently inconsistent. In extending OOD, we would like to investigate its potential applications in data poisoning detection and defense against malicious inputs. Additionally, we consider developing a novel OOD framework independent of model inversion to enhance its applicability and robustness.

## References

- [1] Microsoft, *Face API Reference*, 2024. Azure AI Services Documentation.
- [2] A. W. Services, *Amazon Rekognition Documentation*, 2024. AWS Documentation.
- [3] G. Cloud, *Cloud Vision Documentation*, 2024. Google Cloud Documentation.
- [4] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Boston, Massachusetts, USA), pp. 815–823, IEEE, 2015.
- [5] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Long Beach, CA, USA), pp. 4690–4699, IEEE, 2019.
- [6] J. Ji, H. Wang, Y. Huang, J. Wu, X. Xu, S. Ding, S. Zhang, L. Cao, and R. Ji, “Privacy-preserving face recognition with learnable privacy budgets in frequency domain,” in *European Conference on Computer Vision (ECCV)*, (Tel Aviv, Israel), pp. 475–491, Springer, 2022.
- [7] Y. Mi, Y. Huang, J. Ji, M. Zhao, J. Wu, X. Xu, S. Ding, and S. Zhou, “Privacy-preserving face recognition using random frequency components,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19673–19684, IEEE, 2023.
- [8] Z. Boulkenafet, J. Komulainen, L. Li, X. Feng, and A. Hadid, “Oulu-npu: A mobile face presentation attack database with real-world variations,” in *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*, pp. 612–618, IEEE, 2017.
- [9] H. Zhang, X. Dong, Y. Lai, Y. Zhou, X. Zhang, X. Lv, Z. Jin, and X. Li, “Validating privacy-preserving face recognition under a minimum assumption,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Seattle, Washington, United States), pp. 12205–12214, IEEE, 2024.
- [10] T. Jeong and H. Kim, “Ood-maml: Meta-learning for few-shot out-of-distribution detection and classification,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 33, pp. 3907–3916, 2020.
- [11] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, “Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10951–10960, 2020.
- [12] J. Yang, H. Wang, L. Feng, X. Yan, H. Zheng, W. Zhang, and Z. Liu, “Semantically coherent out-of-distribution detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8301–8309, 2021.
- [13] Y. Yu, S. Shin, S. Lee, C. Jun, and K. Lee, “Block selection method for using feature norm in out-of-distribution detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15701–15711, 2023.
- [14] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (ACM CCS)*, pp. 1322–1333, 2015.
- [15] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain, “On the reconstruction of face images from deep face templates,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1188–1202, 2018.
- [16] Z. Zhang, X. Wang, J. Huang, and S. Zhang, “Analysis and utilization of hidden information in model inversion attacks,” *IEEE Transactions on Information Forensics and Security*, 2023.
- [17] H. O. Shahreza, V. K. Hahn, and S. Marcel, “Vulnerability of state-of-the-art face recognition models to template inversion attack,” *IEEE Transactions on Information Forensics and Security*, 2024.
- [18] C. N. Duong, T.-D. Truong, K. Luu, K. G. Quach, H. Bui, and K. Roy, “Vec2face: Unveil human faces from their blackbox features in face recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6132–6141, 2020.
- [19] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 253–261, 2020.
- [20] M. Khosravy, K. Nakamura, Y. Hirose, N. Nitta, and N. Babaguchi, “Model inversion attack by integration of deep generative models: Privacy-sensitive face generation from a face recognition system,” *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 357–372, 2022.
- [21] X. Yuan, K. Chen, J. Zhang, W. Zhang, N. Yu, and Y. Zhang, “Pseudo label-guided model inversion attack via conditional generative adversarial network,” in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 37, pp. 3349–3357, 2023.
- [22] B.-N. Nguyen, K. Chandrasegaran, M. Abdollahzadeh, and N.-M. M. Cheung, “Label-only model inversion attacks via knowledge transfer,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 36, 2023.
- [23] J. Wu, C. Wan, H. Chen, Z. Zheng, and Y. Sun, “Label-only model inversion attacks: Adaptive boundary exclusion for limited queries,” *Neurocomputing*, p. 129902, 2025.
- [24] M. Kansy, A. Raël, G. Mignone, J. Naruniec, C. Schroers, M. Gross, and R. M. Weber, “Controllable inversion of black-box face recognition models via diffusion,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 3167–3177, 2023.
- [25] R. Liu, D. Wang, Y. Ren, Z. Wang, K. Guo, Q. Qin, and X. Liu, “Unstoppable attack: Label-only model inversion via conditional diffusion model,” *IEEE Transactions on Information Forensics and Security*, 2024.
- [26] H. Otroushi Shahreza and S. Marcel, “Face reconstruction from facial templates by learning latent space of a generator network,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 36, 2023.
- [27] H. O. Shahreza and S. Marcel, “Template inversion attack using synthetic face images against real face recognition systems,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2024.
- [28] L. Struppek, D. Hintersdorf, A. D. A. Correia, A. Adler, and K. Kersting, “Plug & play attacks: Towards robust and flexible model inversion attacks,” in *International Conference on Machine Learning (ICML)*, pp. 20522–20545, PMLR, 2022.
- [29] Y. Qiu, H. Fang, H. Yu, B. Chen, M. Qiu, and S.-T. Xia, “A closer look at gan priors: Exploiting intermediate features for enhanced model inversion attacks,” in *European Conference on Computer Vision (ECCV)*, pp. 109–126, Springer, 2024.
- [30] X. Dong, Z. Miao, L. Ma, J. Shen, Z. Jin, Z. Guo, and A. B. J. Teoh, “Reconstruct face from features based on genetic algorithm using gan generator as a distribution constraint,” *Computers & Security*, vol. 125, p. 103026, 2023.

- [31] S. Pang, Y. Rao, Z. Lu, H. Wang, Y. Zhou, and M. Xue, “Pridm: Effective and universal private data recovery via diffusion models,” *IEEE Transactions on Dependable and Secure Computing*, 2025.
- [32] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2022.
- [33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems (NIPS)*, vol. 27, 2014.
- [34] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in *International Conference on Learning Representations (ICLR)*, 2016.
- [35] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8110–8119, 2020.
- [36] F. Croce and M. Hein, “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks,” in *International Conference on Machine Learning (ICML)*, pp. 2206–2216, 2020.
- [37] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, “SDEdit: Guided image synthesis and editing with stochastic differential equations,” in *International Conference on Learning Representations (ICLR)*, 2022.
- [38] H. Wang, C.-C. Chang, C.-S. Lu, C. Leckie, and I. Echizen, “Greedy pixel: Fine-grained black-box adversarial attack via greedy algorithm,” *arXiv preprint arXiv:2501.14230*, 2025.
- [39] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, “Maskgan: Towards diverse and interactive facial image manipulation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [40] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” in *Workshop on Faces in ‘Real-Life’ Images: Detection, Alignment, and Recognition*, 2008.
- [41] I. G. Alex K, Ben Hamner, “Nips 2017: Defense against adversarial attack,” 2017.
- [42] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [43] R. B. D’Agostino, “Transformation to normality of the null distribution of  $g_1$ ,” *Biometrika*, vol. 57, no. 3, pp. 679–681, 1970.
- [44] R. D’agostino and E. S. Pearson, “Tests for departure from normality. empirical results for the distributions of  $b^2$  and  $\sqrt{b^1}$ ,” *Biometrika*, vol. 60, no. 3, pp. 613–622, 1973.
- [45] R. B. D’agostino, A. Belanger, and R. B. D’Agostino Jr, “A suggestion for using powerful and informative tests of normality,” *The American Statistician*, vol. 44, no. 4, pp. 316–321, 1990.
- [46] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.
- [47] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, “Square attack: a query-efficient black-box adversarial attack via random search,” in *European Conference on Computer Vision (ECCV)*, pp. 484–501, 2020.
- [48] Q. V. Vo, E. Abbasnejad, and D. Ranasinghe, “Brusleattack: Query-efficient score-based sparse adversarial attack,” in *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.
- [49] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4401–4410, 2019.

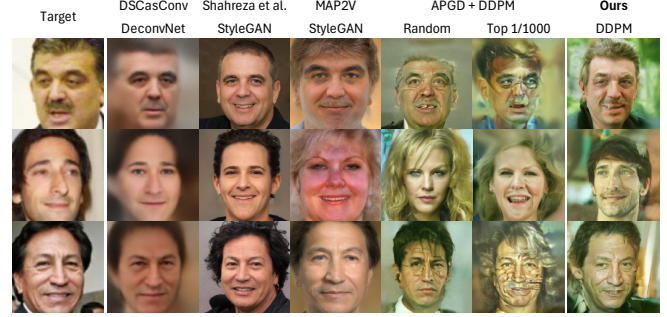


Figure 14. Visual fidelity comparison across attack methods shows that both Shahreza *et al.* [27] and our method produce high-resolution, headshot-style reconstructions with superior visual quality. In contrast, DSCasConv [17] yields the lowest fidelity, with reduced resolution, blurred features, and limited facial detail. Naively applying APGD to the DDPM latent space introduces severe visual artifacts, even when starting from the top-1 latent code out of 1,000 candidates.

- [50] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “Vg-gface2: A dataset for recognising faces across pose and age,” in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 67–74, IEEE, 2018.
- [51] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “Ms-celeb-1m: A dataset and benchmark for large-scale face recognition,” in *14th European Conference on Computer Vision (ECCV)*, pp. 87–102, Springer, 2016.
- [52] A. Krizhevsky, G. Hinton, *et al.*, “Learning multiple layers of features from tiny images,” tech. rep., University of Toronto, 2009.
- [53] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *International Conference on Learning Representations (ICLR)*, 2021.
- [54] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 34, pp. 8780–8794, 2021.

## Appendix A. Training-Dependent Attacks and Naive APGD

As shown in Tab. 1, MAP<sup>2</sup>V [9] is the most recent and relevant training-free baseline for open-set face recognition. We also compare with training-dependent attacks [17], [27] in Tab. 13 and Fig. 14, though they are not primary baselines due to a key limitation: each target model requires a separately trained generator, leading to high computational cost. We exclude closed-set face recognition attacks, as they rely on class labels rather than target embeddings and thus follow different assumptions.

As shown in Tab. 13, we evaluate two training-dependent attacks [17], [27], each using a generator trained specifically for the target model ArcFace [5]. The training (FFHQ [49]) and testing (LFW [40]) datasets are disjoint, ensuring an open-set setting. In contrast, both MAP<sup>2</sup>V [9] and our DiffUMI are training-free. Despite the advantage of training-dependent methods, DiffUMI performs comparably to DSCasConv [17] and significantly outperforms the others.

Despite DSCasConv’s good reconstruction accuracy, it faces substantial limitations. Specifically, it took 1.5 days of training on dual A100 80GB GPUs and is constrained to

TABLE 13. COMPARISON OF DIFFUMI WITH BENCHMARK ATTACK METHODS.

Attack	Generator	Training-Free	Dataset	Target Model	Test Model									
					FaceNet		ArcFace		DCTDP		PartialFace		Avg.	
					Type I	Type II	Type I	Type II	Type I	Type II	Type I	Type II	Type I	Type II
DSCasConv [17]	DeconvNet	✗	LFW	ArcFace	97.10	90.42	100.00	99.03	99.90	97.16	99.50	93.21	99.13	94.96
Shahreza <i>et al.</i> [27]	StyleGAN	✗			61.00	49.33	98.10	77.79	84.10	59.41	74.00	52.24	79.30	59.69
MAP <sup>2</sup> V [9]	StyleGAN	✓			67.60	60.40	100.00	99.42	97.10	83.78	91.20	73.27	88.98	79.22
APGD (Random)*	DDPM	✓			69.10	61.33	100.00	99.56	92.30	75.40	84.50	66.51	86.48	75.70
APGD (Top 1/1000)*	DDPM	✓			78.00	71.70	100.00	99.63	96.00	83.91	90.70	76.59	91.18	82.96
Ours	DDPM	✓			96.00	91.03	100.00	99.64	99.60	95.88	98.60	92.33	98.55	94.72

\*denotes direct application of APGD to the DDPM latent space without additional strategies. “Top 1/1000” refers to initialization from the best latent code among 1,000 random samples, while “Random” indicates no pre-selection (*i.e.*, Top 1/1). Gray cells indicate settings where the target and test models are the same. Our training-free attack matches the performance of the training-dependent DSCasConv, while outperforming other baselines.

generating  $112 \times 112$  resolution images for a specific target model. Moreover, DSCasConv produces the lowest visual fidelity (see Fig. 14), with blurred features and limited facial detail. These limitations highlight the value of training-free approaches and underscore the advantages of leveraging generative models such as GANs or diffusion models.

We also compare our method with a baseline that directly applies APGD to the latent space of diffusion models. As shown in Tab. 13 and Fig. 14, this naive strategy achieves near-perfect accuracy (nearly 100%) on the target model but suffers from severe adversarial artifacts and overfitting, resulting in 11.97%–29.7% lower Type II accuracy than our method on test models. These results underscore the novelty of our approach, which manipulates unconditional diffusion generation in a principled and transferable manner, rather than merely adapting existing adversarial methods.

## Appendix B. Preliminary

### B.1. Image Generation via DDPM

Denoising Diffusion Probabilistic Models (DDPM) [42] are generative models that refine images by reversing a diffusion process. Pretrained DDPMs have been applied to unconditional image generation of faces [37], CIFAR-10 [52], [53], and ImageNet [41], [54] from Gaussian distributions. In the forward diffusion process, Gaussian noise is progressively added to a data sample  $x_0$  over  $T$  time steps, generating a sequence  $\{x_t\}_{t=1}^T$ .

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)I), \quad (23)$$

where  $\alpha_t = 1 - \beta_t$  and  $\beta_t$  is a predefined noise variance schedule. The marginal distribution of  $x_t$  is given by:

$$q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I), \quad (24)$$

where  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ . The reverse process aims to denoise  $x_T \sim \mathcal{N}(0, I)$  back to a realistic image  $x_0$  using a learned model  $p_\theta$ :

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \quad (25)$$

For image generation, the pretrained DDPM learns a denoising function  $\epsilon_\theta$ , parameterized by a neural network:

$$\epsilon_\theta(x_t, t) \approx \epsilon \sim \mathcal{N}(0, I), \quad (26)$$

optimized through the objective:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{x_0, \epsilon, t} [\|\epsilon - \epsilon_\theta(x_t, t)\|^2]. \quad (27)$$

This noise-aware iterative refinement process enables high-fidelity image generation.

We define the final denoised image  $x_0$  as the reconstructed image  $\hat{x}$ , applying a pretrained DDPM as a generator  $G(\cdot)$  to an initial Gaussian noise sample  $x_G$ :

$$\hat{x} = G(x_G). \quad (28)$$

Here,  $x_G$  is a latent code sampled from a Gaussian distribution, which  $G(\cdot)$  refines into a high-fidelity image  $\hat{x}$ .

### B.2. Open-Set Face Recognition

Open-set face recognition systems [4]–[7] map facial images to embeddings and classify image pairs as the same identity if their similarity exceeds a threshold, typically chosen to balance false positives or minimize the equal error rate.

Let  $F(\cdot)$  be the embedding function of a face recognition model that maps a facial image  $x$  to a  $d$ -dimensional feature embedding  $z$ , encoding identity-specific attributes while ensuring robustness to variations in lighting, pose, and expression. Formally:

$$z = F(x), \quad z \in \mathbb{R}^d, \quad (29)$$

where  $d$  is determined by the model’s architecture and training process.

Feature embeddings facilitate identity verification or recognition by measuring similarity using metrics such as cosine similarity or Euclidean distance. Given two embeddings,  $z_1 = F(x_1)$  and  $z_2 = F(x_2)$ , their similarity score is computed as:

$$S(z_1, z_2) = \frac{z_1 \cdot z_2}{\|z_1\| \|z_2\|}, \quad (30)$$

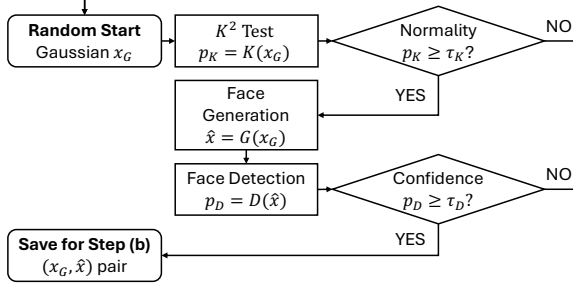


Figure 15. The algorithm of the proposed two-stage latent code generation process. Initially, D’Agostino’s  $K^2$  test  $K(\cdot)$  [43]–[45] evaluates a randomly generated latent code  $x_G$ , retaining those that exhibit Gaussian normality with a  $p_K$  value exceeding the threshold  $\tau_K$ . Subsequently, MTCNN  $D(\cdot)$  [46] assesses whether the latent code can generate a facial image  $\hat{x}$ , with face detection confidence  $p_D$  surpassing the threshold  $\tau_D$ .

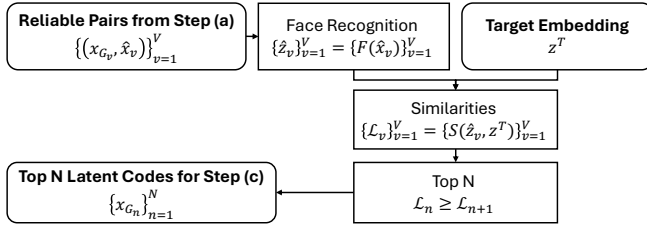


Figure 16. Algorithm for selecting the top  $N$  latent codes based on the highest embedding similarity between their reconstructions and the target.

for cosine similarity. If  $S(z_1, z_2) \geq \tau_F$ , the images are classified as belonging to the same identity. In this work,  $\tau_F$  denotes the predefined similarity threshold set at the minimum equal error rate.

### B.3. D’Agostino’s K-Square Test

Given a random latent code  $x_G$ , the  $K^2$  test function  $K(\cdot)$  [43]–[45] quantifies deviations from normality based on skewness and kurtosis, producing a probability value:

$$p_K = K(x_G) = 1 - \Psi_2(Y_1^2 + Y_2^2), \quad (31)$$

where  $Y_1$  and  $Y_2$  are the standardized skewness and kurtosis statistics:

$$Y_1 = \frac{b_1 - \mu_1}{\sigma_1}, \quad Y_2 = \frac{b_2 - \mu_2}{\sigma_2}. \quad (32)$$

Here,  $b_1$  and  $b_2$  represent the sample skewness and excess kurtosis, while  $\mu_1, \sigma_1, \mu_2$ , and  $\sigma_2$  denote their respective means and standard deviations under normality. The function  $\Psi_2(\cdot)$  is the cumulative distribution function of the chi-squared distribution with two degrees of freedom.

### B.4. Face Detection via MTCNN

Multi-Task Cascaded Convolutional Neural Networks (MTCNN) [46] uses a three-stage cascaded architecture: (i) The Proposal Network (P-Net) scans the image across multiple scales to generate candidate face regions. (ii) The

Refinement Network (R-Net) filters false positives and refines bounding boxes. (iii) The Output Network (O-Net) further refines detections and predicts five facial landmarks. The final detection confidence is obtained through:

$$p_D = D(\hat{x}) = f_{\text{O-Net}}(f_{\text{R-Net}}(f_{\text{P-Net}}(\hat{x}))). \quad (33)$$

## Appendix C. Algorithms of DiffUMI

The step-by-step algorithm of the proposed model inversion attack, DiffUMI, is detailed in Figs. 15 to 17. The key notations and their corresponding definitions are summarized in Tab. 14.

## Appendix D. More Results in Terms of Type I Accuracy

In Sec. 4, we introduce Type I and Type II accuracy as metrics for assessing the effectiveness of privacy attacks. This study primarily focuses on Type II accuracy, which quantifies the similarity between reconstructed facial images and images from the target identity, excluding the target image itself. Type II accuracy offers a more rigorous evaluation by mitigating the risk of overfitting to the target image, whose embedding is used as a reference during the attack. For further validation, we also report Type I accuracy results, which reflect the strength of the inversion attack and highlight potential privacy vulnerabilities. Specifically, Type I accuracy is the rate at which the reconstructed image  $\hat{x}$  matches the target face  $x^T$  in the feature space of the face recognition model  $F(\cdot)$ :

$$\text{Type I} = \frac{\sum_{i=1}^I \mathbb{I}(S(F(\hat{x}_i), F(x_i^T)) \geq \tau_F)}{I}, \quad (34)$$

where  $\mathbb{I}(\cdot)$  and  $S(\cdot, \cdot)$  represent the indicator and cosine similarity functions, respectively.  $I$  is the total number of attack samples, and  $\tau_F$  is the similarity threshold for face recognition.

The results in Tabs. 15 and 16 reinforce our main paper’s conclusion that all tested face recognition models are vulnerable to our privacy attack. Among them, FaceNet, the oldest standard model, exhibits the highest resistance, underscoring the limitations of existing privacy-preserving techniques. Our approach consistently outperforms the benchmark MAP<sup>2</sup>V attack [9], achieving higher Type I accuracy across all scenarios. Notably, Type I accuracy exceeds Type II accuracy by 3.83% to 15.79%, demonstrating that our primary evaluation metric is more rigorous and comprehensive.

TABLE 14. KEY NOTATIONS AND THEIR CORRESPONDING DEFINITIONS.

Notation	Definition	Remark	Reference
$F(\cdot)$	Embedding function (Face recognition)		
$x^T$	Target facial image	Unknown to Attackers	
$z^T$	Target embedding, transformed from the target face	Known to Attackers	
$\hat{x}$	Reconstructed image	Attack Output	
$\hat{z}$	Feature embedding of the reconstructed image		
$G(\cdot)$	Generative function (DDPM)	Pretrained	
$x_G$	Latent code, drawn from a random Gaussian distribution	Attack Input	Sec. 2.3
$x'_G$	Manipulated latent code		
$\delta$	Adversarial perturbations on the latent code		
$\ \cdot\ _p$	$L_p$ -norm		
$\epsilon$	Perturbation magnitude	Attack Setting	
$S(\cdot, \cdot)$	Similarity function (cosine)		
$\mathcal{L}$	Objective function		
$\tau_F$	Similarity decision threshold		
$K(\cdot)$	Gaussian normality test function ( $K^2$ test)		
$p_K$	Gaussian normality (the probability of following a normal distribution)		Sec. 2.4.1
$\tau_K$	Threshold of Gaussian normality	Attack Setting	
$D(\cdot)$	Face detection function (MTCNN)		
$p_D$	Face detection confidence score		Sec. 2.4.2
$\tau_D$	Threshold of detection confidence	Attack Setting	
$V$	Volume of reliable latent codes (Step (a))	Attack Setting	
$N$	Top $N$ selection (Step (b))	Attack Setting	Sec. 2.5
$Q$	Query Efficiency		
$t_{max}$	Maximum iterations per adversarial attack	Attack Setting	
$Q_{max}$	Maximum number of queries (only for black-box attacks)	Attack Setting	Sec. 2.6
$\tau_A$	Attack threshold (sufficient similarity)	Attack Setting	
$\lfloor \cdot \rfloor$	Floor function		
$\mathbb{I}(\cdot)$	Indicator function		
$I$	Total number of attack samples		Sec. 4
$x^{j \neq T}$	Facial images distinct from the target, associated with the same identity		
$J$	Total number of $x^{j \neq T}$ for each identity		

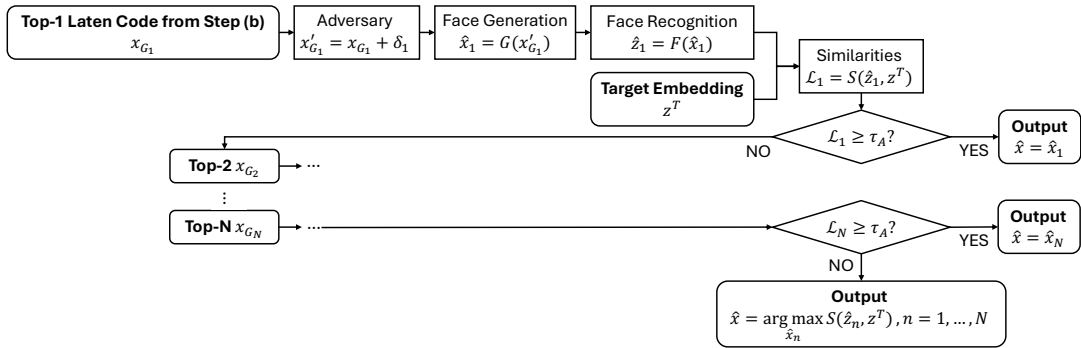


Figure 17. Algorithm for the proposed Ranked Adversary in latent code manipulation. This process sequentially optimizes the top  $N$  initial latent codes, ranked in Step (b), through adversarial manipulation to achieve the objective defined in Eq. (7) using the objective function in Eq. (6). The process concludes once  $\mathcal{L}_n \geq \tau_A$ ,  $n = 1, \dots, N$  for any manipulated latent code. If no code meets this criterion, the reconstruction with the highest  $\mathcal{L}_n$  is selected as the final output.



TABLE 15. TYPE I ACCURACY (%) OF DIFFUMI ACROSS FOUR FACE RECOGNITION MODELS.

Dataset	Target Model	Test Model				Avg.
		FaceNet	ArcFace	DCTDP	PartialFace	
CelebA	FaceNet	100.0	96.7	97.9	94.9	97.38
	ArcFace	99.9	100.0	100.0	99.9	99.95
	DCTDP	99.3	99.5	100.0	99.9	99.68
	PartialFace	97.7	99.2	99.6	100.0	99.13
LFW	FaceNet	100.0	87.0	93.4	80.8	90.30
	ArcFace	96.0	100.0	99.6	98.6	98.55
	DCTDP	96.7	99.7	100.0	97.8	98.55
	PartialFace	90.9	98.7	96.2	100.0	96.45

Gray cells indicate cases where the target and test models are identical. Green and Red highlight the most and least secure models, respectively, based on the lowest and highest Type I accuracy.

TABLE 16. TYPE I ACCURACY (%) OF OUR DIFFUMI ACROSS FOUR FACE RECOGNITION MODELS, COMPARED TO THE BENCHMARK ATTACK MAP<sup>2</sup>V [9].

Dataset	Attack	Target Model			
		FaceNet	ArcFace	DCTDP	PartialFace
CelebA	Original <sup>1</sup>	100.00	100.00	100.00	100.00
	Random <sup>2</sup>	4.40	1.30	4.20	10.10
	MAP <sup>2</sup> V	89.75	95.03	96.75	97.98
	Ours	<b>97.38</b>	<b>99.95</b>	<b>99.68</b>	<b>99.13</b>
LFW	Original <sup>1</sup>	100.00	100.00	100.00	100.00
	Random <sup>2</sup>	0.70	0.00	0.20	0.50
	MAP <sup>2</sup> V	90.05	88.98	92.50	90.54
	Ours	<b>90.30</b>	<b>98.55</b>	<b>98.55</b>	<b>96.45</b>

<sup>1</sup> Upper bound corresponding to true target faces.

<sup>2</sup> Lower bound referring to randomly generated facial images without a specific strategy.

**Bold** indicates the highest performance among attack methods, with our approach exceeding MAP<sup>2</sup>V attack by up to 9.57% in Type I accuracy.