

796

Appendix

797

798

799	A Related Work	21
800	B Derivation of the Alignment-aware Training Objective	22
801	C Algorithmic Realizations of Gradient–Manifold Alignment Methods	22
802	D Experimental Setup and Implementation Details	23
803	D.1 Hard- and Software Details	23
804	D.2 Target Models	24
805	D.3 Evaluation Models	24
806	D.4 Attack Parameters	25
807	D.5 Evaluation Metrics	25
808	D.6 Experimental Details for Figure 3	26
809	D.7 Experimental Details for Figure 4	26
810	E Additional Experimental Results	28
811	E.1 Additional Empirical Validation of the Hypothesis	28
812	E.2 Additional Evaluations of Proposed Methods	30
813	E.3 Ablation Study	31
814	E.4 Visualization of Gradient Images	32
815	E.5 Visualization of Reconstructed Images	32
816	F Discussion	34

A Related Work

Model inversion attacks (MIAs) were first introduced by Fredrikson et al. [2014], who demonstrated the reconstruction of private data in simple regression tasks using shallow models. Their pioneering attack algorithm aimed to infer sensitive attributes, such as genetic markers, via input space optimization, assuming access to both the linear target model and auxiliary information. This work highlighted the privacy risks inherent in exposing model predictions. Building on this, Fredrikson et al. [2015] extended MIAs to shallow neural networks for reconstructing low-resolution grayscale face images. While effective for simple models, this method fails when applied to deep neural networks (DNNs) handling high-dimensional data, as reconstructions often lack semantic relevance.

To address these limitations, Zhang et al. [2020] introduced the two-stage *generative model inversion* approach, which leverages generative adversarial networks (GANs) [Goodfellow et al., 2014, Radford et al., 2016] to learn an image prior from public auxiliary datasets and constrains the attack optimization to the generator’s latent space. This breakthrough significantly improved the visual quality and semantic fidelity of reconstructed samples and has since fueled major advances in the field of MIAs, particularly for high-dimensional image data [Zhou et al., 2024]. Recent works can be categorized by the model inversion adversary’s access level: white-box, black-box, and label-only settings—each posing unique challenges and guiding corresponding defense developments.

In the *white-box* setting, where attackers have full access to the model architecture and weights, most works follow the generative model inversion framework. KEDMI [Chen et al., 2021] enhanced this by introducing an advanced discriminator that incorporates knowledge from the target model. VMI [Wang et al., 2021a] recast the problem as variational inference, using a Bayesian framework to balance diversity and fidelity. PPA [Struppek et al., 2022] further pushed the frontier by leveraging pre-trained StyleGAN generators and introducing the Poincaré loss to replace cross-entropy (CE) loss, addressing gradient vanishing issues in the inversion process. Similarly, Nguyen et al. [2023b] proposed the *logit maximization* (LOM) loss as an alternative to CE loss, alongside model augmentation techniques to mitigate overfitting. PLG-MI [Yuan et al., 2023] advanced MIAs by integrating a conditional GAN (cGAN) with max-margin loss and pseudo-label guidance, effectively decoupling class-specific search spaces and enhancing the exploitation of target model information. These methods primarily concentrate on either the initial training process of GANs or the optimization techniques used in the attacks. A Recent work PPDG-MI [Peng et al., 2024b] took a different direction by fine-tuning the GAN generator post-attack with reconstructed samples, narrowing the distribution gap between prior and private data distributions.

In the *black-box* setting, where attackers can only query the model, An et al. [2022] introduced a genetic search approach to replace gradient-based optimization, while RLB-MI [Han et al., 2023] framed the attack as a Markov decision process (MDP) and applied reinforcement learning to optimize the latent vector. In the *label-only* setting—the most restrictive scenario where only hard labels are accessible—Kahla et al. [2022] proposed the *boundary-repelling model inversion* (BREP-MI) method, which uses zeroth-Order Optimization method to approximate gradient descent and steer the search toward dense class regions. Inspired by transfer learning, Nguyen et al. [2023a] introduced *label-only via knowledge transfer* (LOKT), which uses a target model-assisted ACGAN (T-ACGAN) to effectively transform the label-only attack into a white-box setting.

Many studies have also focused on designing defense methods against generative MIAs. Since MIAs exploit the strong correlation between inputs and outputs for successful attacks, Wang et al. [2020] proposed augmenting the standard classification objective with a mutual information regularizer to penalize this correlation. However, this approach can significantly degrade the model’s predictive performance. To overcome this limitation, Peng et al. [2022] introduced bilateral dependency optimization (BiDO), which enhances the dependency between input features and latent representations while minimizing the dependency between representations and outputs [Peng et al., 2025]. Inspired by BiDO, Stealthy Shield Defense (SSD) [Zhuang et al., 2025] adopts an inference-time strategy that minimizes mutual information between input features and predictions while maximizing the mutual information between predictions and labels, providing an effective black-box defense. Additionally, Ho et al. [2024] proposed freezing the early layers of a pre-trained model and fine-tuning the remaining layers on private data to reduce vulnerability for reconstruction attacks. Struppek et al. [2024] further observed that negative label smoothing can also mitigate generative MIAs. In a recent work, Hao et al. [2024] examined the impact of model architecture on MIA robustness and found that residual connections can increase vulnerability to these attacks.

B Derivation of the Alignment-aware Training Objective

In this section, we provide a derivation of the Inequality (6), which serves as a relaxation used to obtain the final alignment-aware training objective in Eq. (7).

Lemma B.1. *Under the same notation as in Section 4, and assuming all gradient vectors $\nabla_{\mathbf{x}} f_i(\mathbf{x}; \boldsymbol{\theta})$ have equal norm, the following inequality holds:*

$$-\frac{\|\tilde{\mathbf{P}}_{\mathbf{x}} \sum_{i=1}^C \nabla_{\mathbf{x}} f_i(\mathbf{x}; \boldsymbol{\theta})\|}{\|\sum_{i=1}^C \nabla_{\mathbf{x}} f_i(\mathbf{x}; \boldsymbol{\theta})\|} \geq -\frac{1}{C} \sum_{i=1}^C \frac{\|\tilde{\mathbf{P}}_{\mathbf{x}} \nabla_{\mathbf{x}} f_i(\mathbf{x}; \boldsymbol{\theta})\|}{\|\nabla_{\mathbf{x}} f_i(\mathbf{x}; \boldsymbol{\theta})\|}. \quad (9)$$

Proof. Let $g_i := \nabla_{\mathbf{x}} f_i(\mathbf{x}; \boldsymbol{\theta})$ and assume $\|g_i\| = a > 0$ for all i . Put $g := \sum_{i=1}^C g_i$ and suppose $g \neq 0$. By linearity of the orthogonal projector $\tilde{\mathbf{P}}_{\mathbf{x}}$,

$$\tilde{\mathbf{P}}_{\mathbf{x}} g = \sum_{i=1}^C \tilde{\mathbf{P}}_{\mathbf{x}} g_i.$$

Applying the triangle inequality,

$$\|\tilde{\mathbf{P}}_{\mathbf{x}} g\| = \left\| \sum_{i=1}^C \tilde{\mathbf{P}}_{\mathbf{x}} g_i \right\| \leq \sum_{i=1}^C \|\tilde{\mathbf{P}}_{\mathbf{x}} g_i\|.$$

Dividing both sides by $\|g\|$ and multiplying by -1 reverses the inequality:

$$-\frac{\|\tilde{\mathbf{P}}_{\mathbf{x}} g\|}{\|g\|} \geq -\frac{\sum_{i=1}^C \|\tilde{\mathbf{P}}_{\mathbf{x}} g_i\|}{\|g\|}. \quad (10)$$

Since $\|g\| = \|\sum_{i=1}^C g_i\| \leq \sum_{i=1}^C \|g_i\| = Ca$, we have $\frac{1}{\|g\|} \geq \frac{1}{Ca}$. Substituting this bound into the right-hand side of Inequality (10) yields

$$-\frac{\sum_{i=1}^C \|\tilde{\mathbf{P}}_{\mathbf{x}} g_i\|}{\|g\|} \geq -\frac{\sum_{i=1}^C \|\tilde{\mathbf{P}}_{\mathbf{x}} g_i\|}{Ca} = -\frac{1}{C} \sum_{i=1}^C \frac{\|\tilde{\mathbf{P}}_{\mathbf{x}} g_i\|}{a}. \quad (11)$$

Combining Inequalities (10) and (11) and substituting $a = \|g_i\|$ gives

$$-\frac{\|\tilde{\mathbf{P}}_{\mathbf{x}} g\|}{\|g\|} \geq -\frac{1}{C} \sum_{i=1}^C \frac{\|\tilde{\mathbf{P}}_{\mathbf{x}} g_i\|}{\|g_i\|},$$

which is exactly Inequality (9). Equality holds iff both triangle inequalities above are tight, *i.e.*, (i) all vectors $\tilde{\mathbf{P}}_{\mathbf{x}} g_i$ are colinear and (ii) all g_i themselves are colinear. \square

C Algorithmic Realizations of Gradient–Manifold Alignment Methods

This section presents the algorithmic implementations of our proposed training objective for validating the hypothesis, as well as the training-free alignment approach designed to enhance gradient–manifold alignment and improve model inversion performance.

(1) Alignment-Aware Training. To validate our hypothesis that stronger alignment between loss gradients and the generator manifold leads to greater inversion vulnerability, we introduce a gradient–manifold alignment-aware training objective. This objective augments the standard classification loss with a geometric alignment term and can be optimized via standard backpropagation. The training procedure is detailed in Algorithm 1.

(2) Training-Free Alignment Promotion. Motivated by the above findings, we propose a training-free method that improves gradient–manifold alignment at inversion time. By averaging loss gradients over perturbed or transformed versions of the synthetic input, this approach denoises the gradient signal in a geometry-aware manner. The inference-time procedure is described in Algorithm 2.

Algorithm 1 Gradient–Manifold Alignment-Aware Training

Input: Classifier $f(\cdot; \theta)$, pre-trained VAE decoder \mathcal{D} , training set \mathcal{D}_{pri} , trade-off hyperparameter β , number of training steps T

Output: Updated target model parameters θ

```
1: for  $t = 1$  to  $T$  do
2:   Sample a minibatch  $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^B$  from  $\mathcal{D}_{\text{pri}}$ 
3:   for each  $(\mathbf{x}, y)$  in batch do
4:     Compute latent code:  $\mathbf{z} \leftarrow \mathcal{E}(\mathbf{x})$ 
5:     Compute Jacobian:  $J_{\mathcal{D}}(\mathbf{z}) = \frac{\partial \mathcal{D}}{\partial \mathbf{z}}$ 
6:     Compute SVD:  $J_{\mathcal{D}}(\mathbf{z}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ 
7:     Let  $\mathbf{U}_k$  be the first  $k$  columns of  $\mathbf{U}$ 
8:     Estimate projection matrix:  $\tilde{\mathbf{P}}_{\mathbf{x}} \leftarrow \mathbf{U}_k \mathbf{U}_k^\top$ 
9:     Compute softmax probabilities:  $p = \text{softmax}(f(\mathbf{x}; \theta))$ 
10:    Compute CE loss:  $\mathcal{L}_{\text{CE}} = -\log p_y$ 
11:    Compute input gradients of logits:  $\{\nabla_{\mathbf{x}} f_i(\mathbf{x}; \theta)\}_{i=1}^C$ 
12:    Compute gradient sum:  $\mathbf{g} = \sum_{i=1}^C \nabla_{\mathbf{x}} f_i(\mathbf{x}; \theta)$ 
13:    Compute alignment term:  $\mathcal{L}_{\text{align}}^{\text{geo}} \leftarrow \frac{\|\tilde{\mathbf{P}}_{\mathbf{x}} \mathbf{g}\|}{\|\mathbf{g}\|}$ 
14:    Compute final loss:  $\mathcal{L}_{\text{align}}(\theta) \leftarrow \mathcal{L}_{\text{CE}} - \beta \cdot \mathcal{L}_{\text{align}}^{\text{geo}}$ 
15:  end for
16:  Update  $\theta$  via backpropagation over average batch loss
17: end for
18: return  $\theta$ 
```

Algorithm 2 Training-Free Gradient–Manifold Alignment During Inversion

Input: Target model f , pre-trained generator G , inversion loss \mathcal{L} , initial latent code \mathbf{z} , number of inversion steps T , number of samples K , perturbation strength α , sampling strategy $\rho \in \{\text{PAA}, \text{TAA}\}$

Output: Recovered image $\hat{\mathbf{x}} = G(\mathbf{z})$

```
1: for  $t = 1$  to  $T$  do
2:    $\mathbf{x} \leftarrow G(\mathbf{z})$ 
3:   Initialize gradient buffer:  $\mathcal{G} \leftarrow \emptyset$ 
4:   for  $k = 1$  to  $K$  do
5:     if  $\rho = \text{PAA}$  then
6:       Compute noise scale:  $\sigma \leftarrow \alpha (\max(\mathbf{x}) - \min(\mathbf{x}))$ 
7:       Sample noise:  $\epsilon_k \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ 
8:        $\mathbf{x}_k \leftarrow \mathbf{x} + \epsilon_k$ 
9:     else if  $\rho = \text{TAA}$  then
10:      Sample transformation:  $\tau_k \sim \mathcal{T}$ 
11:       $\mathbf{x}_k \leftarrow \tau_k(\mathbf{x})$ 
12:    end if
13:    Compute loss gradient:  $\mathbf{g}_k \leftarrow \nabla_{\mathbf{x}_k} \mathcal{L}(\mathbf{x}_k)$ 
14:    Append to buffer:  $\mathcal{G} \leftarrow \mathcal{G} \cup \{\mathbf{g}_k\}$ 
15:  end for
16:  Compute averaged gradient:  $\tilde{\nabla} \mathcal{L}(\mathbf{x}) \leftarrow \frac{1}{K} \sum_{k=1}^K \mathbf{g}_k$ 
17:  Update latent code:  $\mathbf{z} \leftarrow \mathbf{z} - \eta J_G(\mathbf{z})^\top \tilde{\nabla} \mathcal{L}(\mathbf{x})$ 
18: end for
19: return  $\hat{\mathbf{x}} = G(\mathbf{z})$ 
```

900 D Experimental Setup and Implementation Details

901 D.1 Hard- and Software Details

902 All high-resolution MIA experiments using *Plug & Play Attacks* (PPA) were conducted on Oracle
903 Linux Server 8.9 with NVIDIA A100-80G GPUs, using CUDA 11.7, Python 3.9.18, and PyTorch

Table 3: A summary of experimental setups.

Setting	MIAs	Private Dataset	Public Dataset	Target Model	Evaluation Model
Low-resolution setting	GMI (LOMMA) / KEDMI (LOMMA) / PLG-MI	CelebA	CelebA / FFHQ	VGG16 / FaceNet (64)	FaceNet (112)
High-resolution setting	PPA	CelebA / FaceScrub	FFHQ	ResNet-18 / DenseNet-121 / ResNeSt-50	Inception-v3

904 1.13.1. Low-resolution facial recognition MIAs were run on Ubuntu 20.04.4 LTS with NVIDIA RTX
 905 3090 GPUs, under CUDA 11.6, Python 3.7.12, and PyTorch 1.13.1.

906 D.2 Target Models

907 **(1) Empirical Validation of the Hypothesis.** To validate our hypothesis, we conduct experiments
 908 on models pre-trained for a 1000-class classification task using 64×64 CelebA images. The
 909 model and training pipeline are based on the implementation provided at https://github.com/sutd-visual-computing-group/Re-thinking_MI. To compute alignment scores, we require
 910 estimates of the tangent space at each training point. These are obtained using a pre-trained VAE
 911 decoder, which maps latent representations back to the image space. For each training image, we
 912 compute the Jacobian of the decoder to extract the local tangent basis and pre-store it for downstream
 913 alignment computation. However, this procedure is memory-intensive. For example, estimating and
 914 storing tangent bases for approximately 2,700 training images from the first 100 classes of CelebA
 915 requires about 30 GB of disk space. Due to this storage constraint and the exploratory nature of the
 916 analysis, we restrict our investigation to a 100-class subset of the full dataset.

918 To obtain models trained on this 100-class subset, we first adapt the original 1000-class model by
 919 fine-tuning it on the corresponding subset. Fine-tuning is performed for 20 epochs using stochastic
 920 gradient descent with an initial learning rate of 10^{-2} , momentum of 0.9, weight decay of 10^{-4} , and
 921 batch size of 128. The learning rate is scheduled to decrease by a factor of 0.02 at epochs 10 and
 922 15. This procedure yields a 100-class vanilla model. Subsequently, to obtain models with varying
 923 levels of training-time gradient-manifold alignment, we continue fine-tuning the 100-class vanilla
 924 model for 30 additional epochs using our proposed alignment-aware training objective. The learning
 925 rate is fixed throughout this phase. To capture the evolution of training-time alignment scores, we
 926 save model checkpoints at intermediate epochs. These models serve as the basis for evaluating the
 927 correlation between alignment and model inversion vulnerability in later experiments.

928 **(2) Evaluation of Proposed Methods.** To evaluate our proposed methods, we adopt distinct training
 929 configurations for models at different image resolutions. For high-resolution inputs (224×224) from
 930 the CelebA and FaceScrub datasets, we follow the setup from Struppek et al. [2022]. Models are
 931 optimized using Adam [Kingma and Ba, 2015] with an initial learning rate of 10^{-3} , β parameters set
 932 to (0.9, 0.999), and a weight decay of 10^{-3} . Training runs for 100 epochs with a batch size of 128,
 933 and the learning rate is reduced by a factor of 0.1 at epochs 75 and 90. Input preprocessing includes
 934 normalization (mean and standard deviation both set to 0.5), followed by a sequence of augmentations:
 935 random cropping with a scale range of [0.85, 1.0] and fixed aspect ratio of 1.0, resizing to 224×224 ,
 936 and horizontal flipping with a probability of 0.5.

937 For low-resolution images (64×64) from CelebA, we follow the training protocol provided by
 938 https://github.com/sutd-visual-computing-group/Re-thinking_MI. Specifically, we
 939 use stochastic gradient descent (SGD) with an initial learning rate of 10^{-2} , momentum of 0.9, and
 940 weight decay of 10^{-4} . Models are trained for 100 epochs with a batch size of 64, and the learning
 941 rate is decayed by a factor of 0.1 at epochs 75 and 90.

942 D.3 Evaluation Models

943 For our PPA-based experiments, we follow the original implementation at <https://github.com/LukasStruppek/Plug-and-Play-Attacks> to train Inception-v3 evaluation models, using the
 944 training configurations specified in Struppek et al. [2022]. These models achieve test accuracies of
 945

96.53% on FaceScrub and 94.87% on CelebA. To compute K-nearest neighbor (KNN) distances, which serve as a similarity metric between reconstructed and true samples in facial recognition tasks, we adopt the pre-trained FaceNet model [Schroff et al., 2015], available at <https://github.com/timesler/facenet-pytorch>.

For experiments on target models trained on 64×64 resolution CelebA dataset, we use an evaluation model from https://github.com/sutd-visual-computing-group/Re-thinking_MI. This model is based on the face.evoLve architecture [Wang et al., 2021b] with a modified ResNet-50 backbone, and achieves a reported test accuracy of 95.88%. Details on the training procedure are available in Zhang et al. [2020].

D.4 Attack Parameters

High-Resolution Setting. In the high-resolution setting, we follow the *Plug & Play Attack* (PPA) method, which comprises three stages: (1) latent code pre-selection, (2) latent code optimization, and (3) result selection. During pre-selection, we sample 2000 latent codes per class and retain the top 100 candidates based on the target model’s response for both CelebA and FaceScrub datasets. In the optimization stage, we perform 70 iterations of gradient-based latent code updates per class. The final result selection stage is omitted in our implementation in order to include as many as samples for evaluation. We focus on the first 100 classes, generating 100 reconstructed samples per class.

As for the parameters of PAA strategy, we use Gaussian perturbations of standard deviation σ set to 5% of the synthesized images’ dynamic range. For parameters of TAA strategy, we apply three geometrically constrained transformations: random resized cropping with scale factors spanning $[0.8, 1.0]$ and aspect ratios limited to $[0.9, 1.1]$, horizontal flipping with probability $p = 0.5$, and random rotations within $\pm 5^\circ$ angular displacement.

Low-Resolution Setting. In the low-resolution setting, we target the first 100 classes from CelebA as the private dataset \mathcal{D}_{pri} and generate 100 samples per identity using CelebA, FFHQ and FaceScrub as auxiliary datasets \mathcal{D}_{aux} . For instantiations of AlignMI, we maintain identical PAA and TAA parameter configurations from the high-resolution setup unless explicitly stated. Implementation details differ slightly across MIAs. For GMI (LOMMA) using StyleGAN, we directly sample and optimize 100 latent codes for 100 steps with a batch size of 20, and set the PAA’s Gaussian noise standard deviation σ is set to 0.5% of the synthesized images’ dynamic range. For KEDMI (LOMMA) with DCGAN, we process 100 samples per identity through 200 optimization steps with a batch size of 100. For PLG-MI with a cGAN prior, the baseline includes a data augmentation pipeline comprising: random resized cropping to 64×64 with scale in $[0.8, 1.0]$ and fixed aspect ratio 1.0, color jittering with brightness and contrast set to ± 0.2 , random horizontal flips (probability 0.5), and rotations within $\pm 5^\circ$. In our PAA and TAA configurations, we omit this augmentation pipeline to isolate the effect of gradient–manifold alignment. Optimization for PLG-MI runs for 100 steps with a batch size of 20.

Due to the high computational cost of generative MIAs, we perform a single attack per target model. To reduce randomness, we generate at least 100 inversion samples per class across all configurations.

D.5 Evaluation Metrics

Attack Accuracy (Attack Acc). We employ an evaluation model (generally more robust and powerful than the target model) trained on the same dataset as the target model to verify whether reconstructed images correctly represent the target class, following the evaluation method of Zhang et al. [2020]. This metric serves as an automated proxy for human evaluation, assessing how well the reconstructed images capture the distinctive characteristics of the target class compared to other classes. The attack accuracy is computed as the percentage of predictions matching the target class, reporting both top-1 (Acc@1) and top-5 (Acc@5) accuracy scores.

K-Nearest Neighbors Distance (KNN Dist). KNN distance quantifies reconstruction quality through l_2 distance computation in a model’s feature embedding space, measuring the similarity between reconstructed images and their nearest original private training samples. This metric serves as a quantitative indicator of visual fidelity, where smaller distances correspond to higher similarity between generated and genuine training data. For high-resolution attacks in PPA [Struppek et al., 2022], we extract features from FaceNet’s penultimate layer [Schroff et al., 2015], while for low-resolution model inversion attacks, we use the evaluation model’s penultimate layer features.

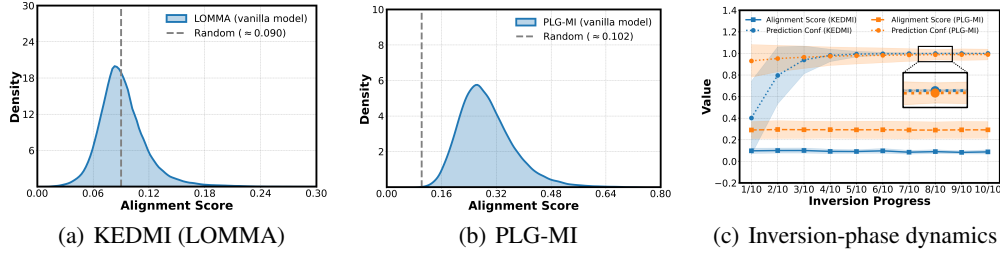


Figure 6: **Additional gradient-manifold alignment during inversion process.** (a) Alignment score distribution for KEDMI (LOMMA) using an inversion-specific GAN trained on CelebA. (b) Corresponding results for PLG-MI using a conditional GAN. (c) Evolution of mean alignment scores versus prediction confidence during inversion. Notably, while prediction confidence demonstrates monotonic improvement throughout the inversion process, gradient-manifold alignment in additional attack methods also remains stable and low, reinforcing the lack of correlation between confidence and gradient-manifold alignment.

D.6 Experimental Details for Figure 3

Low-Resolution Setting. In the low-resolution experiments, we adopt a DCGAN trained on CelebA as the generative prior. The latent space dimension of DCGAN is 100, corresponding to a random baseline alignment score of approximately 0.090. The target classifier is a VGG16 model trained on CelebA, and the inversion targets the first 25 classes, each containing 1,000 images. For Fig. 3(a), we run the inversion optimization for 1,200 steps and record the inversion-time alignment scores of the loss gradients every 10 steps for each reconstructed sample. The figure presents the distribution of all collected alignment scores. In Fig. 3(c), we further analyze temporal dynamics by averaging alignment scores across all classes at each step, illustrating how gradient-manifold alignment evolves during optimization.

Additionally, we evaluate gradient-manifold alignment during the inversion process for other attack methods, including KEDMI (LOMMA) and PLG-MI, in the low-resolution setting. The results are present in Fig. 6. Both methods leverage CelebA as the generative prior and target a VGG16 classifier trained on CelebA. Specifically for KEDMI, we adopt a DCGAN with latent space dimension of DCGAN 100, corresponding to a random baseline alignment score approximately 0.090. The inversion process targets the first 50 classes, each containing 500 images and proceeds 1,200 optimization steps. For PLG-MI, we use a conditional GAN (cGAN) with 128 latent dimensions, which corresponds to a random baseline alignment score approximately 0.102. The inversion process executes 100 optimization iterations targeting the first 100 classes, each containing 100 images.

Interestingly, the PLG-MI method exhibits higher inversion-time alignment scores than GMI (LOM) and KEDMI (LOM). This improvement can be attributed to its use of a conditional GAN, which incorporates label information throughout the inversion process. The stronger alignment may partially explain PLG-MI’s superior attack performance.

High-Resolution Setting. In the high-resolution experiments, we use a StyleGAN model trained on FFHQ as the generative prior. The latent space has dimension 512, yielding a random baseline alignment score of approximately 0.058. The target classifier is a ResNet18 model trained on CelebA, with inversion targeting the first 50 classes, each containing 50 images. For Fig. 3(b), inversion is run for 100 steps, with alignment scores recorded at 10 equally spaced intervals per reconstructed sample. The figure shows the distribution of the recorded scores. In Fig. 3(c), we track temporal alignment by averaging scores over all latent vectors at each interval, capturing how alignment develops throughout the inversion process.

D.7 Experimental Details for Figure 4

Tangent Space Estimation. To compute training-time alignment scores, we estimate the tangent space at each training sample using a pre-trained VAE from Stable Diffusion. Specifically, the VAE encoder maps an input image x of shape $64 \times 64 \times 3$ to a latent representation z of shape $8 \times 8 \times 4$, which is then decoded back to the image space by the VAE decoder. For each training image, we compute the Jacobian of the decoder to obtain the local tangent basis, resulting in a Jacobian matrix of

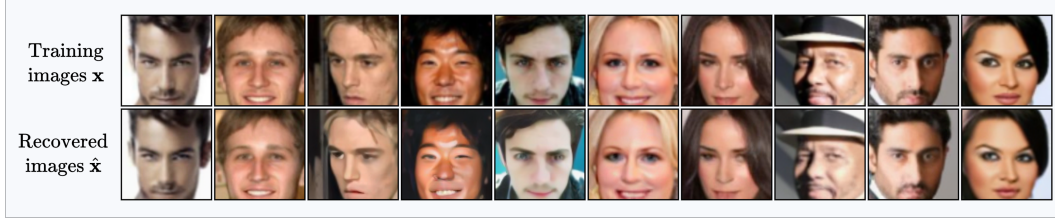


Figure 7: Original training samples (top row) and corresponding reconstructions (bottom row) from the pre-trained VAE used for tangent space estimation. The visual similarity confirms the VAE’s ability to approximate the natural image manifold reliably.

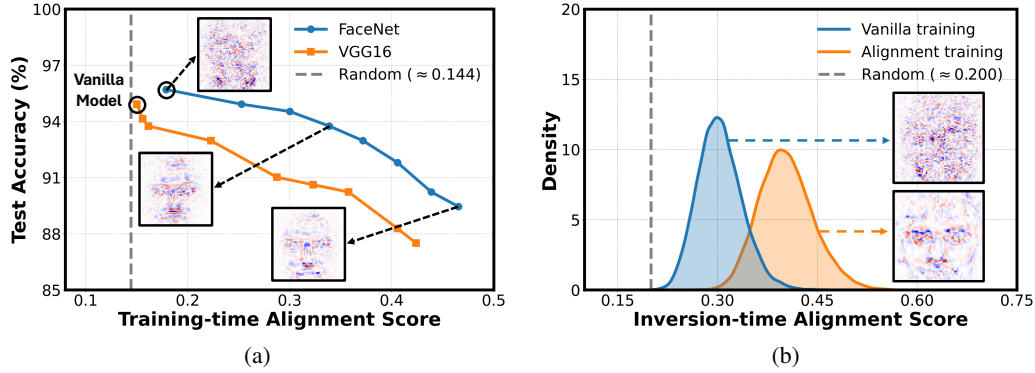


Figure 8: **Empirical evaluation of gradient-manifold alignment (enlarged version).** (a) Test accuracy vs. training-time alignment score (AS_{tr}) for models sampled during fine-tuning vanilla models with the alignment-aware training objective. Insets show input gradient visualizations for models with varying degrees of alignment. (b) Distribution of inversion-time alignment scores (AS_{inv}) for the vanilla model compared to the alignment-aware model.

1035 shape $12,288 \times 256$. This process is memory-intensive: for example, estimating and storing tangent
 1036 bases for approximately 2,700 training samples from the first 100 classes of CelebA consumes
 1037 roughly 30 GB of disk space. As shown in Fig. 7, the reconstructed images closely match the original
 1038 inputs, indicating that the pre-trained VAE, despite not being trained on the target dataset, offers a
 1039 reliable approximation of the natural image manifold.

1040 **Empirical evaluation of gradient-manifold alignment.** To empirically evaluate the trade-off
 1041 between test accuracy and training-time alignment score as shown in Fig. 4(a) (or Fig. 8(a)), we
 1042 conducted experiments using two 100-class target models: VGG16 and FaceNet. The training
 1043 procedures for these models followed the same specifications detailed in Appendix D.2. During
 1044 training, we saved intermediate model checkpoints at various epochs to capture the evolution of
 1045 model performance under our alignment-aware objective.

1046 For analyzing the distribution of inversion-time alignment scores presented in Fig. 4(b) (or Fig. 8(b)),
 1047 we select two 100-class FaceNet models as target models. The vanilla model achieves a test accuracy
 1048 of 96.53% with training-time alignment score $AS_{tr} = 0.175$, while the aligned model achieves a test
 1049 accuracy of 93.75% with $AS_{tr} = 0.339$. We use the GMI (LOM) attack method with StyleGAN as a
 1050 prior, targeting the first 25 classes and running the optimization for 100 steps with batch size 20 for
 1051 both the vanilla and aligned models.

1052 In Fig. 4(c), we extend our evaluation to 1000-class VGG16 models, following the same training
 1053 protocol as described in Appendix D.2. We save checkpoints at intermediate training epochs to obtain
 1054 models with varying test accuracies. The alignment scores AS_{tr} are recorded throughout the training
 1055 process. Additionally, we compute the alignment scores AS_{inv} using the GMI (LOM) attack with
 1056 StyleGAN, again targeting the first 25 classes and running the optimization for 100 steps.



Figure 9: **Training-time alignment progression with alignment-aware training.** Evolution of training-time alignment score (AS_{tr}) and gradient visualizations during fine-tuning of FaceNet using our alignment-aware objective. As alignment improves, loss gradients exhibit increasingly structured and semantically meaningful patterns. (Best viewed with zoom.)

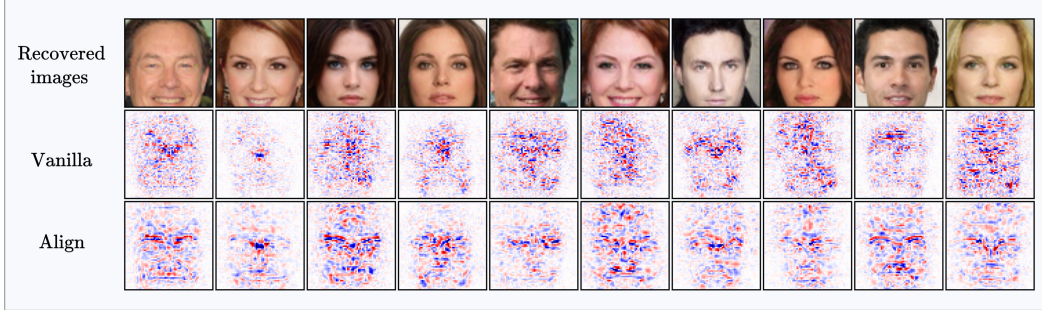


Figure 10: **Comparison of inversion-time loss gradients.** Visualization of loss gradients from the vanilla model (top) and the alignment-aware model (bottom). The alignment-aware model produces gradients that are sharper and more semantically aligned with facial structures, indicating stronger alignment with the generator manifold. (Best viewed with zoom.)

1057 E Additional Experimental Results

1058 E.1 Additional Empirical Validation of the Hypothesis

1059 We illustrate the fine-tuning progress of a FaceNet model optimized with our alignment-aware ob-
 1060 jective in Fig. 10. As fine-tuning proceeds, the training-time alignment score (AS_{tr}) consistently
 1061 increases, and corresponding gradient visualizations exhibit progressively clearer and more seman-
 1062 tically meaningful structures. This demonstrates the effectiveness of our alignment-aware training
 1063 strategy in promoting geometrically informative gradients.

1064 For comparison, Fig. 10 also presents inversion-time loss gradient images from both the vanilla and
 1065 alignment-aware models. The gradients from the alignment-aware model reveal clearer, semantically
 1066 meaningful structures, highlighting improved alignment with the underlying generator manifold.

1067 To further validate our hypothesis, we extend our experiments to include IR152 as the target model,
 1068 using the GMI (LOM) attack method. As shown in Fig. 11(a), the results are consistent with our
 1069 earlier findings in Fig. 4(a) (Sec. 6.2): as fine-tuning progresses, the training-time alignment score
 1070 (AS_{tr}) steadily increases, and corresponding gradient visualizations reveal increasingly semantically

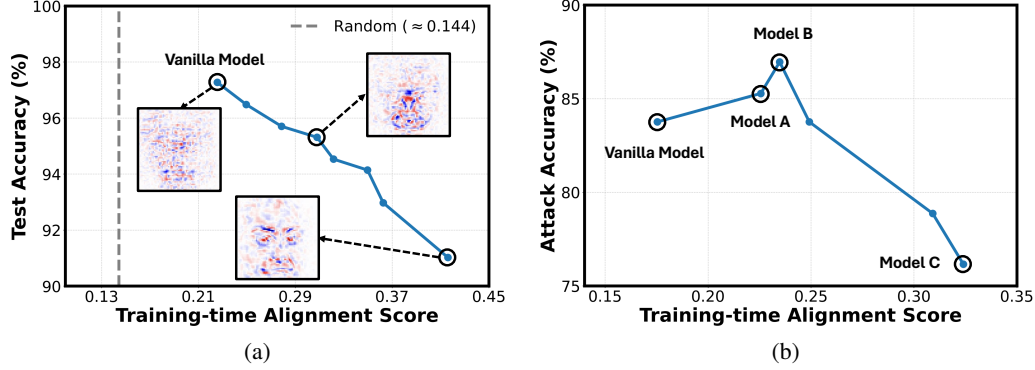


Figure 11: **Additional empirical evaluation of gradient-manifold alignment.** (a) Test accuracy vs. training-time alignment score (AS_{tr}) for IR152 models sampled during fine-tuning vanilla models with the alignment-aware training objective. Insets show input gradient visualizations for models with varying degrees of alignment. (b) MIA success on vanilla and alignment-aware IR152 models with different AS_{tr} .

Table 4: Comparison of inversion performance with white-box MIAs in the low-resolution setting. Target model $f = VGG16$ trained on $\mathcal{D}_{pri} = \text{CelebA}$. GANs are trained on $\mathcal{D}_{aux} = \text{CelebA}$ or FFHQ.

Method	CelebA				FFHQ			
	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow	Ratio \downarrow	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow	Ratio \downarrow
GMI (LOMMA)	74.56	90.40	1433.52	/	48.20	73.66	1623.42	/
+ PAA (K=50)	76.41 (+1.85)	91.77	1454.02 (+20.50)	2.45	48.77 (+0.57)	74.61	1610.53 (-12.89)	2.46
+ TAA (K=50)	82.99 (+8.43)	95.43	1401.38 (-32.14)	2.58	64.43 (+16.23)	86.48	1529.61 (-93.81)	2.62
KEDMI (LOMMA)	77.21	95.77	1290.24	/	41.50	70.62	1509.28	/
+ PAA	79.45 (+2.24)	96.02	1289.25 (-0.99)	18.59	42.20 (+0.70)	71.46	1474.45 (-34.83)	18.58
+ TAA	74.05 (-3.16)	95.09	1314.92 (+24.68)	19.71	41.99 (+0.49)	72.57	1500.89 (-8.39)	19.74

Table 5: Comparison of inversion performance with white-box MIAs in the low-resolution setting. Target model $f = \text{FaceNet}$ trained on $\mathcal{D}_{pri} = \text{CelebA}$. GANs are trained on $\mathcal{D}_{aux} = \text{CelebA}$ or FFHQ.

Method	CelebA				FFHQ			
	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow	Ratio \downarrow	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow	Ratio \downarrow
GMI (LOMMA)	79.21	89.94	1496.62	/	57.89	78.69	1607.35	/
+ PAA (ours)	80.24 (+1.03)	91.23	1474.16 (-22.46)	4.77	59.63 (+1.74)	80.57	1585.70 (-21.65)	4.75
+ TAA (ours)	91.90 (+12.69)	97.11	1407.35 (-89.27)	5.03	82.58 (+24.69)	93.31	1449.00 (-158.35)	5.05
KEDMI (LOMMA)	76.76	96.93	1278.26	/	58.40	86.21	1433.97	/
+ PAA (ours)	83.95 (+7.19)	97.95	1280.79 (+2.53)	19.09	49.47 (-8.93)	81.68	1458.88 (+24.91)	19.12
+ TAA (ours)	88.00 (+11.24)	98.89	1211.48 (-66.78)	15.39	62.89 (+4.49)	87.61	1438.19 (+4.22)	15.40

Table 6: Comparison of inversion performance with PLG-MI in the low-resolution setting. Target model $f = \text{FaceNet}$ trained on $\mathcal{D}_{pri} = \text{CelebA}$. GANs are trained on $\mathcal{D}_{aux} = \text{FaceScrub}$ or FFHQ.

Method	FaceScrub				FFHQ			
	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow	Ratio \downarrow	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow	Ratio \downarrow
PLG	32.06	58.17	1558.26	/	88.68	97.06	1267.12	/
+ PAA (ours)	29.93 (-2.13)	53.99	1557.11 (-1.15)	9.07	87.32 (-1.36)	96.37	1270.54 (+3.42)	9.04
+ TAA (ours)	35.99 (+3.93)	62.87	1539.27 (-18.99)	11.07	90.79 (+2.11)	97.56	1256.07 (-11.05)	11.07

1071 meaningful features. Notably, this rise in alignment is accompanied by a gradual decline in test
1072 accuracy, reaffirming the trade-off between alignment and generalization.

1073 Additionally, we evaluate model inversion performance across both vanilla and alignment-aware mod-
1074 els with varying levels of AS_{tr} . As shown in Fig. 11(b), the trend mirrors Fig. 5: MIA vulnerability
1075 increases with alignment up to a certain threshold, after which further increases in AS_{tr} reduce attack
1076 success. This characteristic inverted V-shaped relationship supports our hypothesis and demonstrates
1077 that the correlation between gradient-manifold alignment and inversion vulnerability holds across
1078 different model architectures.

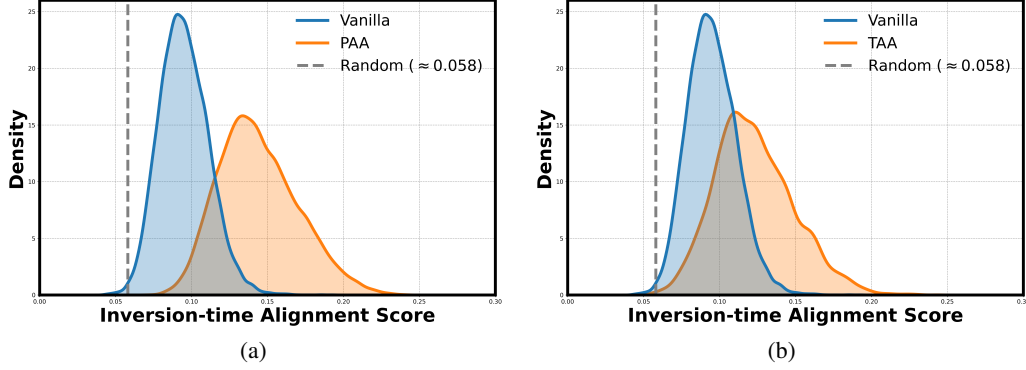


Figure 12: **Distribution of inversion-time alignment scores.** (a) Comparison between baseline and PAA method. (b) Comparison between baseline and TAA method. Each plot shows the distribution of alignment scores between the inversion-time loss gradients and the generator manifold. The measurement is performed using the PPA method with a StyleGAN generator trained on FFHQ, and the target model is a ResNet-18 trained on CelebA. Both PAA and TAA lead to a rightward shift in the score distribution, indicating stronger alignment with the generator manifold.

E.2 Additional Evaluations of Proposed Methods

Inversion-Time Alignment Score Comparison with PPA in High-Resolution Setting. Fig. 12 shows the distribution of inversion-time alignment scores for the baseline method and our training-free variants, PAA and TAA. These results are obtained using the PPA attack on a ResNet-18 model trained on CelebA, with a StyleGAN generator pre-trained on FFHQ. Both PAA and TAA significantly shift the alignment score distribution to the right compared to the vanilla baseline, indicating stronger alignment between the loss gradients and the generator manifold. This enhanced alignment aligns well with the improved gradient visualizations shown in Fig. 13.

Comparison with white-box MIAs in the low-resolution setting. In this experiment, we evaluate the performance of two target models, namely VGG16 and FaceNet, under three attack methods: GMI (LOMMA), KEDMI (LOMMA), and PLG-MI. Quantitative results are presented in Tabs. 4, 5, and 6. Overall, AlignMI consistently outperforms baseline methods in most setups, achieving gains in both attack accuracy and KNN distance across different auxiliary datasets. For example, when attacking VGG16 using GMI (LOMMA), PAA increases top-1 accuracy from 74.56% to 76.41% on CelebA, while TAA achieves an additional 8.43% improvement and reduces the KNN distance from 1433.52 to 1401.38. Similar trends are observed for KEDMI (LOMMA) and PLG-MI, demonstrating the broad effectiveness of our proposed techniques. However, we also observe occasional performance drops, particularly with PAA in certain KEDMI (LOMMA) and PLG-MI scenarios. This degradation likely arises from the poor visual quality of reconstructions produced by certain low-resolution attacks, especially under significant distribution shifts between the private and public auxiliary datasets. In such cases, additional perturbations further compromise image fidelity, diminishing the effectiveness of neighborhood sampling. As a result, the derived gradients become less informative, leading to occasional failures in inversion.

Comparisons under SOTA MIA defenses. Our evaluation focuses on the high-resolution setting, where we assess the effectiveness of our proposed training-free alignment enhancement methods, PAA and TAA, when integrated with state-of-the-art (SOTA) generative model inversion attacks against leading MIA defenses, including BiDO-HSIC [Peng et al., 2022], NegLS [Struppek et al., 2024], and TL-DMI [Ho et al., 2024]. The results, summarized in Tab. 7, show that both PAA and TAA improve inversion performance across all defense scenarios, with TAA consistently achieving the strongest results. All attacks are conducted using the *Plug & Play Attack* (PPA) method, targeting a ResNet-152 classifier trained on $\mathcal{D}_{\text{pri}} = \text{FaceScrub}$, with the generative prior provided by a StyleGAN model trained on $\mathcal{D}_{\text{aux}} = \text{FFHQ}$. Detailed results are shown in Tab. 7.

For the BiDO-HSIC defense, the baseline inversion performance drops significantly, with top-1 accuracy (Acc@1) of 35.11%, top-5 accuracy (Acc@5) of 59.14%, and KNN distance of 1.031. Integrating PAA yields moderate gains, raising Acc@1 to 39.06% and Acc@5 to 67.46%, while reducing the KNN distance to 0.975. In contrast, TAA achieves substantial improvements, boosting

Table 7: Model inversion performance against SOTA defense methods in high-resolution settings. Target model $f = \text{ResNet-152}$, trained on $\mathcal{D}_{\text{pri}} = \text{FaceScrub}$. GAN is pre-trained on $\mathcal{D}_{\text{aux}} = \text{FFHQ}$. **Bold** values indicate best performance under each defense.

Method	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow
No Defense	57.89	81.25	0.893
BiDO-HSIC	35.11	59.14	1.031
+ PAA	39.06 (+3.95)	67.46 (+8.32)	0.975 (-0.056)
+ TAA	62.58 (+27.47)	84.09 (+24.95)	0.855 (-0.176)
NegLS	8.40	23.50	1.309
+ PAA	8.62 (+0.22)	23.67 (+0.17)	1.303 (-0.006)
+ TAA	10.61 (+2.21)	27.31 (+3.81)	1.278 (-0.031)
TL-DMI	25.14	51.72	1.026
+ PAA	34.93 (+9.79)	63.66 (+11.94)	1.022 (-0.004)
+ TAA	47.80 (+22.66)	75.51 (+23.79)	0.971 (-0.055)

Table 8: Ablation study on PAA sample size K with $\alpha = 0.03$. Higher K improves results slightly, but gains saturate.

Method	K	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow
PPA	-	77.00	92.44	0.807
+ PAA	20	79.56	93.24	0.804
+ PAA	60	78.64	92.84	0.804
+ PAA	100	78.60	93.32	0.802
+ PAA	150	79.16	93.44	0.797

Table 9: Ablation study on PAA sample size K with $\alpha = 0.05$. Higher K improves results slightly, but gains saturate.

Method	K	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow
PPA	-	77.77	92.73	0.798
+ PAA	20	82.52	94.48	0.789
+ PAA	60	82.04	94.08	0.789
+ PAA	100	81.92	94.55	0.788
+ PAA	150	82.28	94.16	0.788

1115 Acc@1 to 62.58% and Acc@5 to 84.09%, alongside a sharper drop in KNN distance to 0.855. This
1116 suggests that TAA more effectively recovers semantically meaningful gradients that better align with
1117 the generator manifold.

1118 Under the stronger NegLS defense, which imposes stronger regularization and suppresses inversion
1119 more aggressively, the baseline Acc@1 is just 8.40%. Although this setting presents a more chal-
1120 lenging scenario, PAA still offers slight improvements, raising Acc@1 to 8.62% and reducing KNN
1121 distance from 1.309 to 1.303. TAA further improves Acc@1 to 10.61% and reduces KNN distance
1122 to 1.278. While the absolute gains are smaller due to the strength of the defense, the consistent
1123 improvements across all metrics indicate enhanced gradient informativeness.

1124 Finally, the TL-DMI defense, which involves partial model freezing during fine-tuning, the baseline
1125 attack achieves Acc@1 of 25.14%, Acc@5 of 51.72%, and KNN distance of 1.026. PAA improves
1126 Acc@1 to 34.93% and Acc@5 to 63.66%, slightly reducing the KNN distance to 1.022. TAA again
1127 shows superior performance, reaching Acc@1 of 47.80%, Acc@5 of 75.51%, and decreasing KNN
1128 distance to 0.971.

1129 Overall, across all three defenses, both PAA and TAA enhance inversion performance, with TAA
1130 consistently outperforming PAA in all metrics. These results highlight the generality and robustness of
1131 our alignment-enhancing framework. TAA, in particular, effectively boosts attack success rates while
1132 recovering reconstructions that are perceptually and semantically closer to the true data distribution,
1133 even under strong privacy-preserving defenses.

1134 E.3 Ablation Study

1135 In this subsection, we perform an ablation study to examine the sensitivity of our proposed *AlignMI*
1136 approach to two key hyperparameters: (1) the number of samples K used to compute the smoothed,
1137 alignment-enhanced gradients, and (2) the perturbation strength α used in the perturbation-averaged
1138 alignment (PAA) method. All experiments are conducted using a DenseNet-121 target model trained

Table 10: Ablation study on TAA sample size K . Higher K yields marginal gains.

Method	K	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow
PPA	-	77.77%	92.73%	0.798
+ TAA	20	87.64%	96.04%	0.748
+ TAA	60	88.28%	96.44%	0.746
+ TAA	100	88.44%	96.16%	0.745
+ TAA	150	88.16%	96.44%	0.745

Table 11: Ablation study on PAA perturbation scale α at fixed $K = 60$. Increasing α improves alignment but saturates.

Method	α	Acc@1 \uparrow	Acc@5 \uparrow	KNN Dist \downarrow
PAA	0.01	74.72%	91.80%	0.822
PAA	0.03	78.64%	92.84%	0.804
PAA	0.05	82.04%	94.08%	0.789
PAA	0.10	82.84%	94.48%	0.780
PAA	0.15	79.16%	93.44%	0.797

on the FaceScrub dataset at 224×224 resolution, with a StyleGAN generator pre-trained on FFHQ serving as the prior model.

Effect of Sample Number K in PAA. We first investigate the influence of the sample number K on PAA under two different perturbation strengths. As shown in Tabs. 8 and 9, we observe that increasing K has a limited effect on attack accuracy, which remains relatively stable across settings. However, the KNN distance continues to decrease slightly as K grows, indicating progressively finer reconstruction fidelity. These findings suggest that while larger K offers marginal improvements, even a relatively small sample number (*e.g.*, $K = 20$) is sufficient to achieve substantial gains over the baseline. This highlights the practicality of PAA in improving inversion performance with minimal computational overhead.

Effect of Sample Number K in TAA. We conduct a similar evaluation for the TAA method. As presented in Tab. 10, both attack accuracy and KNN distance improve as K increases, with performance gains tapering off beyond $K = 100$. Notably, TAA achieves strong results even with $K = 20$, outperforming the baseline by a significant margin. This again demonstrates that our training-free alignment promotion strategy enhances inversion performance effectively, even with limited sampling, thus making it computationally efficient.

Effect of Perturbation Strength α in PAA. Finally, we analyze the role of the perturbation strength α in PAA. As shown in Tab. 11, increasing α initially boosts both attack accuracy and KNN distance, with performance peaking around $\alpha = 0.1$. However, beyond this threshold (*e.g.*, $\alpha = 0.15$), both metrics begin to deteriorate, likely due to the perturbations introducing excessive noise that destabilizes the model’s prediction and results in unreliable gradients. This suggests that careful tuning of α is critical, and moderate values around 0.05 to 0.1 provide a favorable balance between denoising and preserving informative signals.

E.4 Visualization of Gradient Images

In this subsection, we qualitatively demonstrate that both PAA and TAA produce loss gradients that are better aligned with the generator manifold. Our analysis focuses on the high-resolution setting, which enables high-quality visualizations of gradient structures. Figs. 13, 14, and 15 present gradient visualizations from ResNet-18, DenseNet-121, and ResNeSt-50 models trained on CelebA. Each figure compares gradient maps produced by the baseline, PAA, and TAA methods, using GANs pre-trained on FFHQ. We also visualize the inversion-time loss gradient images for three attack methods in the low-resolution setting (see Fig. 16), as a complementary comparison to Fig. 1(b).

E.5 Visualization of Reconstructed Images

In this subsection, we present qualitative results of the baseline attack methods and our proposed AlignMI approach. High-resolution reconstructions are shown in Figs. 17 and 18. Fig. 17 compares reconstructed samples from the first ten classes using ResNet-18, DenseNet-121, and ResNeSt-50 trained on CelebA, with GANs pre-trained on FFHQ. Fig. 18 provides similar results for the same target models trained on FaceScrub, also using FFHQ-pretrained GANs.

In low-resolution setting, we evaluate reconstruction quality by comparing samples from the first ten classes generated by GMI (LOMMA) and KEDMI (LOMMA) attack methods. These experiments employ VGG16 and FaceNet trained on CelebA as target models, with GANs pre-trained on both CelebA and FFHQ datasets, as shown in Figs. 19, and 20 respectively. Additionally, we present PLG-MI reconstructions on FaceNet using GANs trained on FFHQ and FaceScrub datasets in Fig. 21.

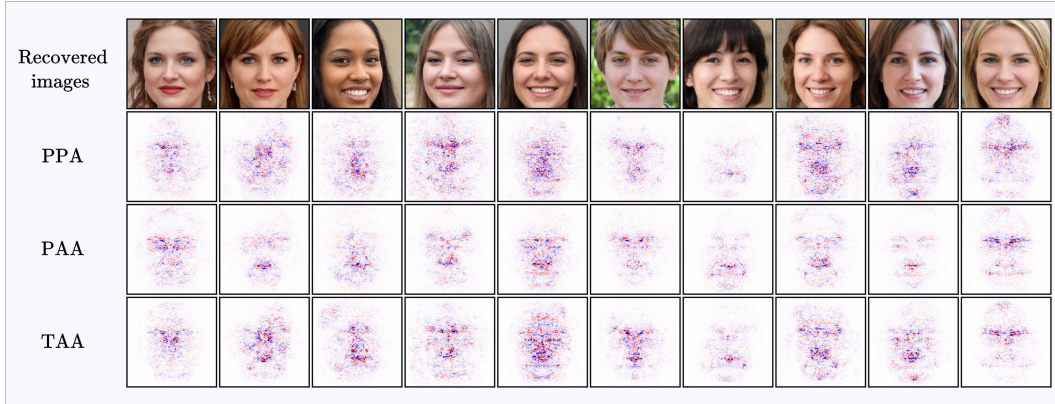


Figure 13: Visual comparison of inversion-time loss gradients for PPA in the high-resolution setting. We illustrate reconstructed samples for ten classes in $\mathcal{D}_{\text{pri}} = \text{CelebA}$ using GANs pre-trained on $\mathcal{D}_{\text{aux}} = \text{FFHQ}$. The target model is ResNet-18. (Best viewed with zoom.)

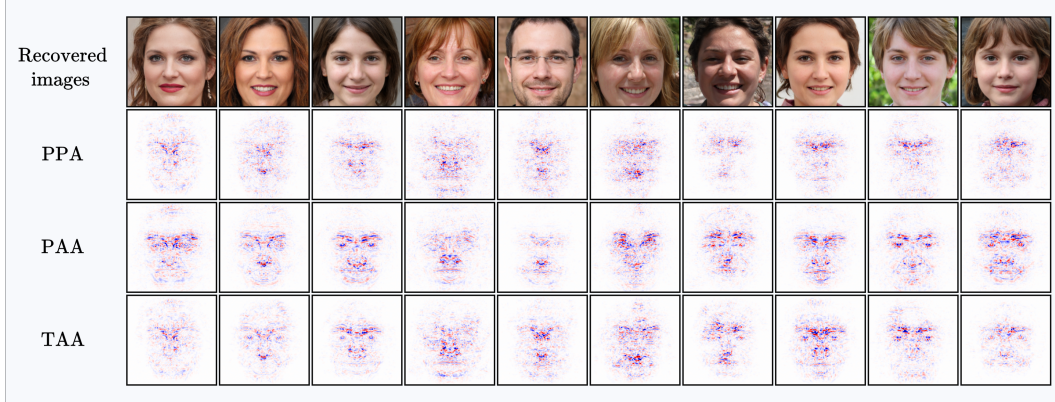


Figure 14: Visual comparison of inversion-time loss gradients for PPA in the high-resolution setting. We illustrate reconstructed samples for ten classes in $\mathcal{D}_{\text{pri}} = \text{CelebA}$ using GANs pre-trained on $\mathcal{D}_{\text{aux}} = \text{FFHQ}$. The target model is DenseNet-121. (Best viewed with zoom.)

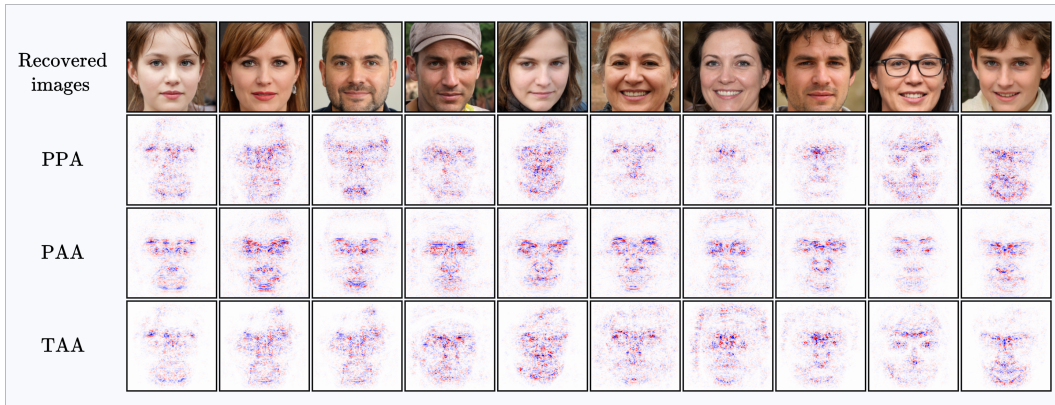


Figure 15: Visual comparison of inversion-time loss gradients for PPA in the high-resolution setting. We illustrate reconstructed samples for ten classes in $\mathcal{D}_{\text{pri}} = \text{CelebA}$ using GANs pre-trained on $\mathcal{D}_{\text{aux}} = \text{FFHQ}$. The target model is ResNeSt-50. (Best viewed with zoom.)

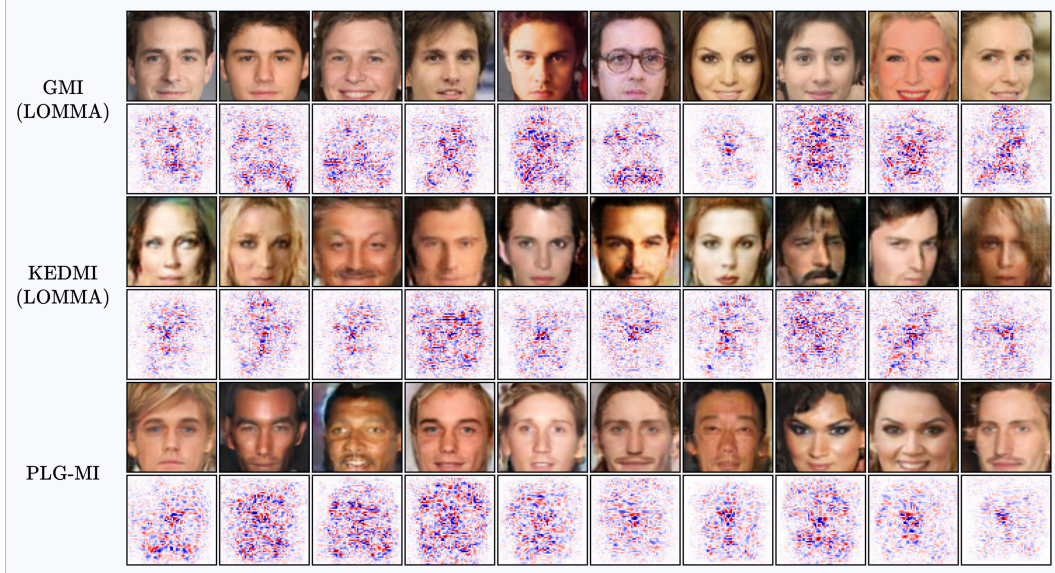


Figure 16: Visual of inversion-time loss gradients for three attack methods in the low-resolution setting. The target model is FaceNet. (Best viewed with zoom.)

F Discussion

Limitations. Although our experiments validate the proposed hypothesis in the low-resolution setting, gradient–manifold alignment-aware training is currently feasible only at this scale. We observe an empirical trade-off between alignment and predictive performance, suggesting that stronger alignment may come at the cost of generalization. However, due to computational limitations, we are unable to assess whether this trend persists in high-resolution settings. In particular, high-resolution inputs of size $224 \times 224 \times 3$ produce latent representations of size $28 \times 28 \times 4$ from the VAE encoder, resulting in a decoder Jacobian of size $150,528 \times 3136$. This is roughly 150 times larger than in the low-resolution case, rendering tangent space estimation computationally and memory intensive. Moreover, the underlying cause of the observed alignment–accuracy trade-off remains unclear and warrants further investigation in future work.

Broader Impacts. From a geometric standpoint, our analysis uncovers a previously overlooked dimension of model inversion vulnerability, complementing existing perspectives focused on predictive power. This insight sheds new light on the mechanisms behind privacy risks in machine learning models. From a broader societal perspective, the AlignMI approach, if misused, could increase the risk of exposing sensitive training data. Conversely, this geometric viewpoint also enables the development of principled defenses against generative MIAs. Specifically, reducing gradient-manifold alignment as a defense is a promising direction for future work.



ResNet-18



DenseNet-121



ResNeSt-50

Figure 17: Visual comparison in high-resolution settings. We illustrate reconstructed samples for the first ten classes in $\mathcal{D}_{\text{pri}} = \text{CelebA}$ using GANs pre-trained on $\mathcal{D}_{\text{aux}} = \text{FFHQ}$.



ResNet-18



DenseNet-121



ResNeSt-50

Figure 18: Visual comparison in high-resolution settings. We illustrate reconstructed samples for the first ten classes in $\mathcal{D}_{\text{pri}} = \text{FaceScrub}$ using GANs pre-trained on $\mathcal{D}_{\text{aux}} = \text{FFHQ}$.

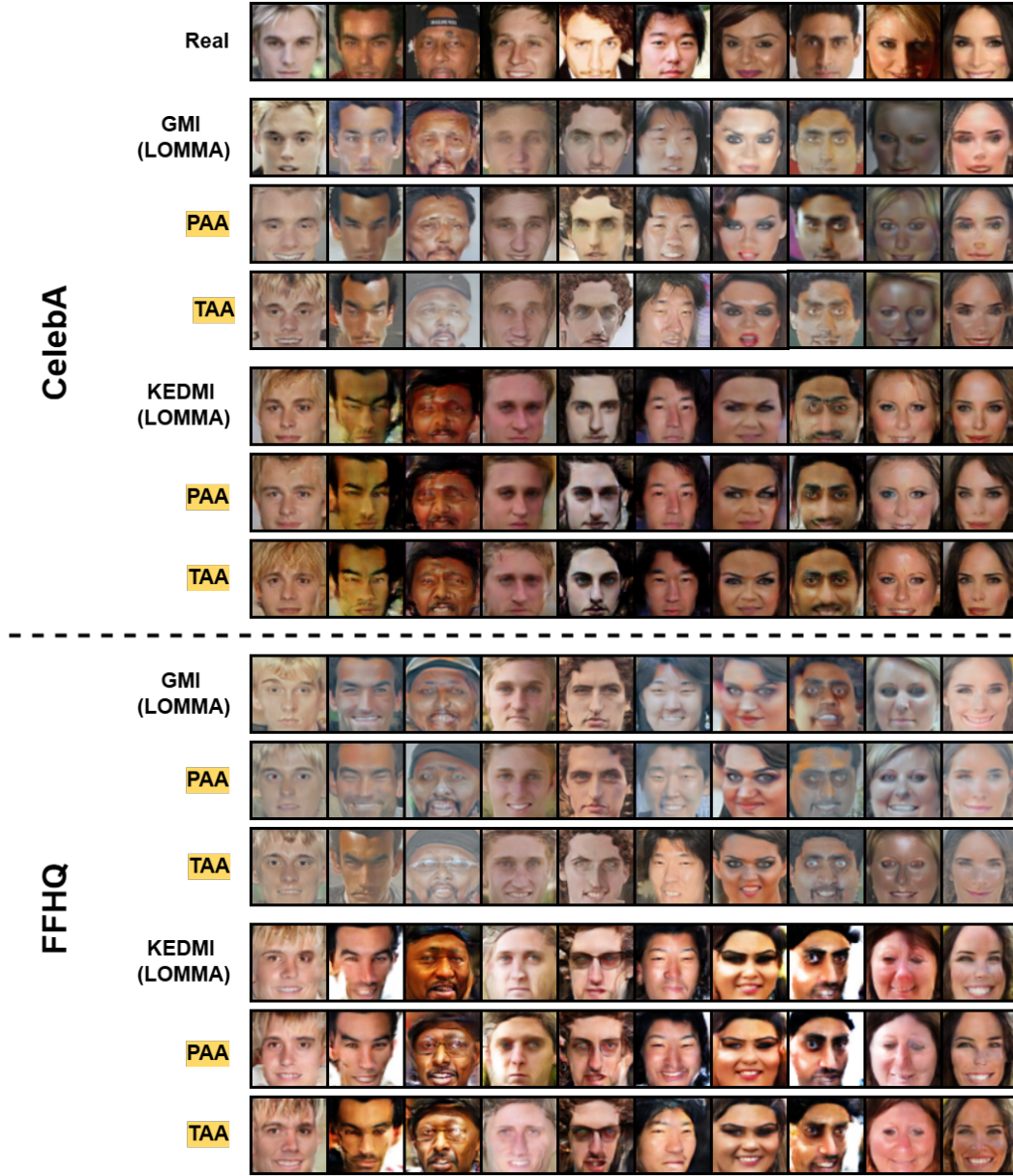


Figure 19: Visual comparison in low-resolutions settings. We illustrate reconstructed samples for the first ten classes in $\mathcal{D}_{\text{pri}} = \text{CelebA}$ using GANs trained from scratch on $\mathcal{D}_{\text{aux}} = \text{CelebA} / \text{FFHQ}$. The target model is VGG16.

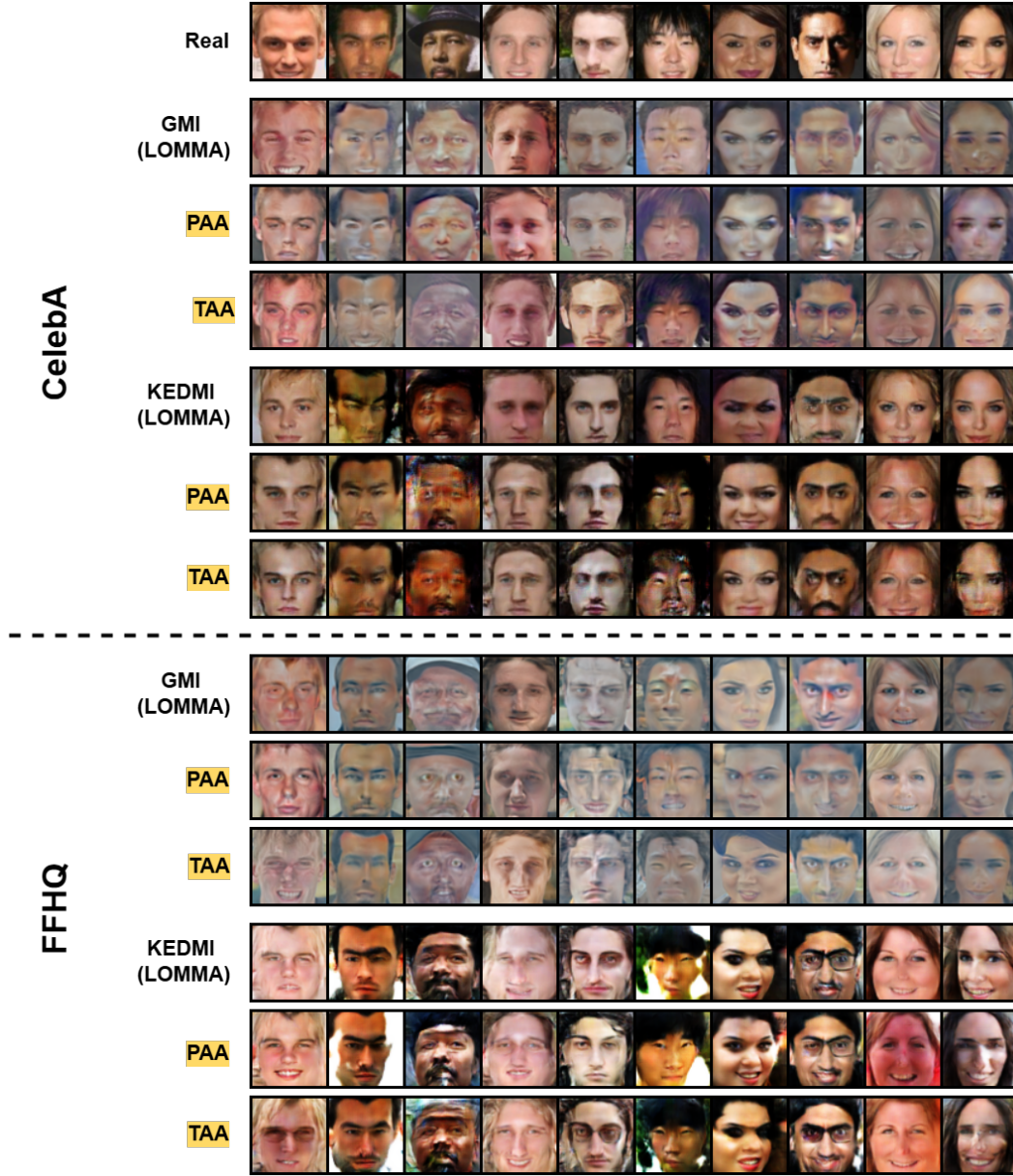


Figure 20: Visual comparison in low-resolutions settings. We illustrate reconstructed samples for the first ten classes in $\mathcal{D}_{\text{pri}} = \text{CelebA}$ using GANs trained from scratch on $\mathcal{D}_{\text{aux}} = \text{CelebA} / \text{FFHQ}$. The target model is FaceNet.

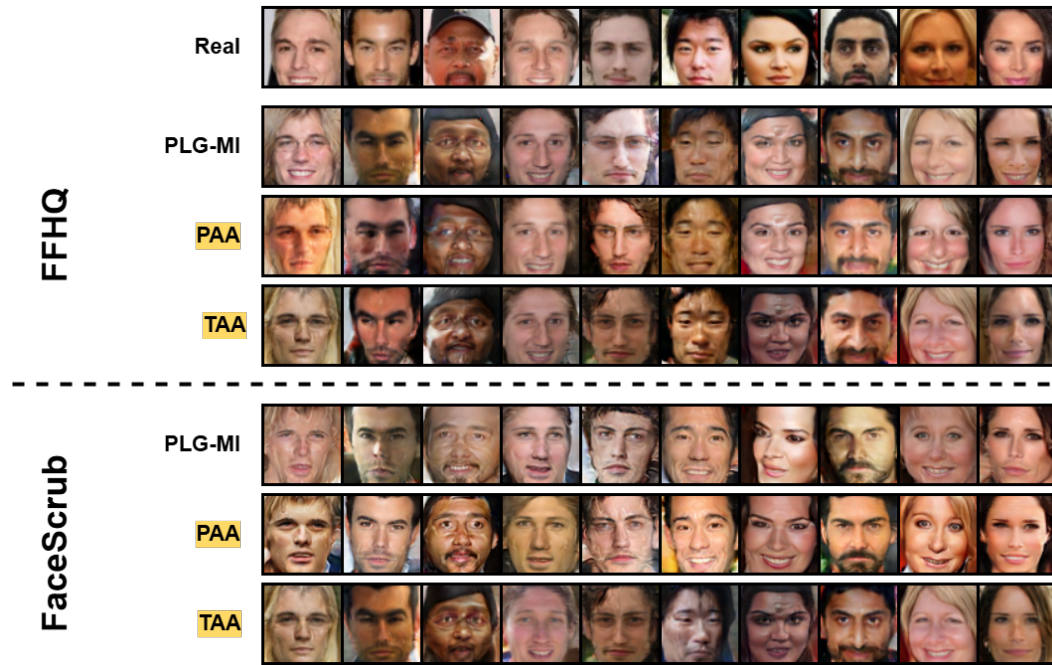


Figure 21: Visual comparison in low-resolutions settings. We illustrate reconstructed samples for the first ten classes in $\mathcal{D}_{\text{pri}} = \text{CelebA}$ using GANs trained from scratch on $\mathcal{D}_{\text{aux}} = \text{FFHQ} / \text{FaceScrub}$. The target model is FaceNet.