Exploring User-level Gradient Inversion with a Diffusion Prior

Zhuohang Li
Vanderbilt University
zhuohang.li@vanderbilt.edu

Andrew Lowy
University of Wisconsin-Madison
alowy@wisc.edu

Jing Liu
Mitsubishi Electric Research Laboratories
jiliu@merl.com

Toshiaki Koike-Akino
Mitsubishi Electric Research Laboratories
koike@merl.com

Bradley Malin
Vanderbilt University Medical Center
b.malin@vumc.org

Kieran Parsons
Mitsubishi Electric Research Laboratories
parsons@merl.com

Ye Wang
Mitsubishi Electric Research Laboratories
yewang@merl.com

Abstract

We explore user-level gradient inversion as a new attack surface in distributed learning. We first investigate existing attacks on their ability to make inferences about private information info beyond training data reconstruction. Motivated by the low reconstruction quality of existing methods, we propose a novel gradient inversion attack that applies a denoising diffusion model as a strong image prior in order to enhance recovery in the large batch setting. Unlike traditional attacks, which aim to reconstruct individual samples and suffer at large batch and image sizes, our approach instead aims to recover a representative image that captures the sensitive shared semantic information corresponding to the underlying user. Our experiments with face images demonstrate the ability of our methods to recover realistic facial images along with private user attributes.

1 Introduction

Computing and storing *gradients* is essential for training most types of modern deep-learning models, ranging from compact neural networks [1] designed for edge applications to massive foundational models [2] that are fine-tuned using gradient descent. In circumstances such as distributed training and federated learning [3], gradients are directly exchanged in place of raw training data to facilitate joint learning from large-scale distributed data.

Plainly sharing gradient information was long presumed to be safe. However, *gradient inversion*, a new type of privacy threat was recently demonstrated that calls this assumption into question. Zhu *et al.* [4] demonstrated that an adversary can recover the underlying private training data via an iterative gradient-matching optimization procedure. While plausible, this optimization problem becomes ill-posed and difficult to solve when the data is high-dimensional [5] and from a large batch size [6], or if local defenses are applied to obfuscate the gradients [7]. As a result, existing solutions based

International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023 (FL@FM-NeurIPS'23). This workshop does not have official proceedings and this paper is non-archival.

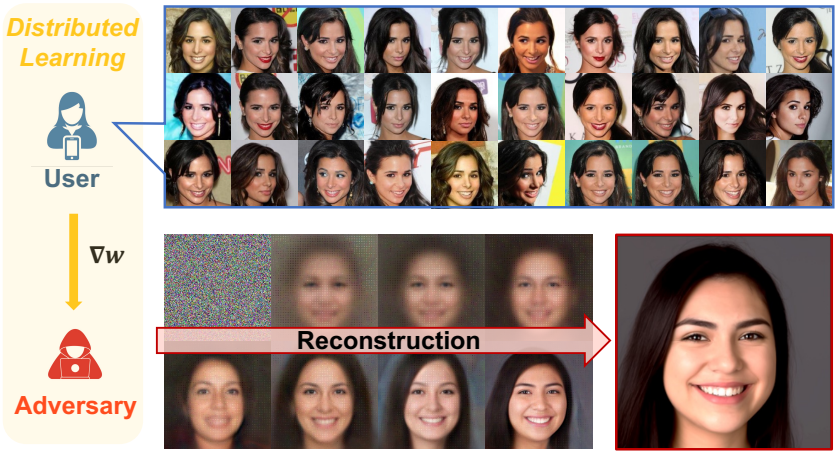


Figure 1: Illustration of *user-level* gradient inversion with diffusion prior. **Top**: user's private batch of 30 images. The measured pairwise facial similarity in the private batch ranges from 0.5217 to 0.9579. **Bottom**: reconstructed image from gradients using the proposed method. The average facial similarity to the original batch is 0.5565.

on total variation [5] or generative adversarial network (GAN) based image priors [8, 7] are prone to degradation in their reconstructed image quality with decreasing amounts of details as batch size and image resolution increase. Further, GAN-based approaches are also known to suffer from mode collapse [9], which can result in limited sample diversity and degrade overall attack performance.

In this work, we explore *user-level* gradient inversion as a new attack surface in distributed learning, beyond the risks of training data recovery. This is motivated by collaborative learning settings, where data across users may be neither independent nor identically distributed, and particularly when the data from each user may consistently belong to a distinct individual. Thus, while recovering individual data samples may be difficult, the potential inference of sensitive user-level characteristics may still pose a risk.

We empirically investigate the user-level privacy risks of existing gradient inversion methods, revealing their abilities to recover some level of private information albeit with low image quality. To improve reconstructed image quality and enable attacks against large batch sizes, we also propose a new type of user-level gradient inversion attack, utilizing a pre-trained diffusion model prior to assist the reconstruction of meaningful images. Specifically, in contrast to existing sample-level attacks that attempt to fully reconstruct every image in the target batch, we instead aim to synthesize a single representative image that captures the overall semantics of the private image batch, which makes the search space dimension invariant to the target batch size and thereby significantly reduces computational overheads and improves convergence stability.

Our method is motivated by the recent success in the research domain of using diffusion priors to solve inverse problems [10, 11, 12, 13]. However, most existing research focuses on typical image inverse problems, such as denoising and inpainting, which have well-defined linear forward operators. By contrast, the forward process in our problem involves the calculation of model gradients, which involves a complex series of non-linear operations. Moreover, the measurements in gradient inversion are the observed gradients, which contrasts with traditional inverse problems, where both the original and measured signals are in the image space, rendering many existing methods inapplicable. To address these challenges, we apply a plug-and-play diffusion prior technique [14] that formulates the inference of the posterior distribution as an optimization problem and propose a dynamic optimization scheme that focuses on recovering the high-level semantics at early stages and then fine-tune image details at the later stage. To the best of our knowledge, this is the first study to investigate the application of a diffusion model prior for improving gradient inversion.

Distinct from the prior literature, our evaluation extends beyond the conventional performance metrics for gradient inversion which measure only image-level similarity to inspect the recovery of certain private attributes that are more semantically meaningful and interpretable. Through experiments on the CelebA facial image dataset [15], we show that the proposed user-level gradient inversion method

can effectively recover high-level semantics of large-batches of images that carry important private information about the person, including gender, race, age, and facial identity, without relying on strong adversarial assumptions such as the BatchNorm statistics [16].

2 User-level Gradient Inversion with Diffusion Prior

2.1 Problem Formulation

We consider the typical supervised distributed learning setting with a set of clients C, where the goal is to find the optimal model parameters $\mathbf{w}$ of a neural network $f_{\mathbf{w}}$ such that the empirical risk is minimized, i.e., $\min_{\mathbf{w}} \sum_{c \in C} \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}_c} \mathcal{L}(f_{\mathbf{w}}(\mathbf{x}_{(i)}, \mathbf{y}_{(i)})$, where $\mathbf{x}_{(i)}, \mathbf{y}_{(i)}$ denotes the i-th image and corresponding label, respectively, $\mathcal{D}_c$ represents the client's local training dataset, and $\mathcal{L}$ is the loss function (e.g., cross-entropy). The most common practice is to optimize $\mathbf{w}$ through stochastic gradient descent. Specifically, at each training step, the client node samples a batch of B images $\{(\mathbf{x}_{(i)}, \mathbf{y}_{(i)})\}_{i=1}^{B}$ to compute the averaged gradients $\nabla \mathbf{w} = \frac{1}{B} \sum_{i=1}^{B} \nabla_{\mathbf{w}} \mathcal{L}(f_{\mathbf{w}}(\mathbf{x}_{(i)}), \mathbf{y}_{(i)})$, which are then transferred over the communication channel to the server (in the parameter server setting) or its peers (in the peer-to-peer setting). In the gradient inversion attack, the adversary observes the user's gradients $\nabla \mathbf{w}$ and global model weights $\mathbf{w}$ and tries to recover the original images and labels. When the adversary's access is limited to aggregated gradients (e.g., under secure aggregation), they may first initiate a disaggregation attack [17] to obtain gradients at the user level. Following prior work [18, 5, 6, 8, 7], we assume the labels can be analytically recovered from the gradient of the last layer. Therefore, the images can be reconstructed by solving the following optimization problem:

$$\hat{\mathbf{x}}^*_{(1)}, ..., \hat{\mathbf{x}}^*_{(B)} = \underset{\hat{\mathbf{x}}_{(1)}, ..., \hat{\mathbf{x}}_{(B)}}{\arg\min} \ \mathbf{d}(F(\hat{\mathbf{x}}_{(1)}, ..., \hat{\mathbf{x}}_{(B)}), \nabla \mathbf{w}), \tag{1}$$

where $\mathbf{d}$ is a distance metric (e.g., Euclidean squared distance [4] or cosine distance [5]), $\hat{\mathbf{x}}_{(1)}, ..., \hat{\mathbf{x}}_{(B)}$ is a batch of synthetic data, and $F(\hat{\mathbf{x}}_{(1)}, ..., \hat{\mathbf{x}}_{(B)}) = \frac{1}{B} \sum_{i=1}^{B} \nabla_{\mathbf{w}} \mathcal{L}(f_{\mathbf{w}}(\hat{\mathbf{x}}_{(i)}), \mathbf{y}_{(i)})$ is the corresponding gradient. This technique is referred to as *gradient matching* [4] in the literature and it has been empirically shown that as the distance in the gradient space reduces, the synthetic images will gradually recover the original images.

2.2 User-Level Gradient Inversion

2.2.1 Challenge of Sample-level Reconstruction

The gradient inversion attack in its canonical form (Eq. 1) suffers poor scalability in multiple aspects. First, due to the ill-posedness of the inverse problem, the returned synthetic image through naively minimizing the gradient matching loss may not be a natural image. These effects become more significant on deeper networks and higher-resolution images. One mitigation is to instead solve:

$$\hat{\mathbf{x}}^*_{(1)}, ..., \hat{\mathbf{x}}^*_{(B)} = \underset{\hat{\mathbf{x}}_{(1)}, ..., \hat{\mathbf{x}}_{(B)}}{\arg\min} \ \mathbf{d}(F(\hat{\mathbf{x}}_{(1)}, ..., \hat{\mathbf{x}}_{(B)}), \nabla \mathbf{w}) + \mathcal{R}_{\text{prior}}(\hat{\mathbf{x}}_{(1)}, ..., \hat{\mathbf{x}}_{(B)}), \tag{2}$$

where $\mathcal{R}_{\text{prior}}$ is an additional prior term (e.g., total variation [5]) introduced to regularize the synthetic image. A more recent line of work considers using the generator G from pre-trained GANs [8, 7] as a regularization and solve for low-dimensional latent vectors $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_B$ that minimizes the gradient matching loss, i.e.,

$$\hat{\mathbf{z}}^*_{(1)}, ..., \hat{\mathbf{z}}^*_{(B)} = \underset{\hat{\mathbf{z}}_{(1)}, ..., \hat{\mathbf{z}}_{(B)}}{\arg\min} \ \mathbf{d}(F(G(\hat{\mathbf{z}}_{(1)}), ..., G(\hat{\mathbf{z}}_{(B)})), \nabla \mathbf{w}). \tag{3}$$

Although this form enables the gradient inversion to generalize to larger images, it still does not address the other challenge that the search space scales linearly with the batch size B and the attack performance quickly degrades as B increases. As such, most existing gradient inversion attacks are confined to small-sized images with relatively small batch sizes (e.g., [8] considered a batch of 4 images rescaled to the size of 64×64px). The only exception is the work by Yin *et al.* [6], which is based on the strong assumption that the clients share additional BatchNorm statistics beyond regular gradients [16]. Another common assumption by prior work is that images in the batch are uncorrelated [19] or from different classes [6], which may not hold when learning across non-IID data. Extending gradient inversion attacks to practical batch sizes without such strong assumptions remains a challenging research problem [16].

2.2.2 From Sample-level to User-level Inversion

In the sample-level gradient inversion attacks as described above, the adversary observes the gradients $\nabla\mathbf{w}$ computed from a batch of B images, and the goal is to reconstruct every sample in the original batch, which is equivalent to estimating images to maximize $p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_B|\nabla\mathbf{w})$ [20]. However, in many practical distributed learning scenarios, such as cross-device federated learning on mobile devices [21, 22], data may often have strong local correlation (and may not be identical between users), i.e., local data samples are from the same user and thus have similar semantics. In these settings, the ultimate goal of a practical attack may not be image reconstruction, but rather recovery of private information about the user, such as demographics and identity. To account for this semantic consistency across local batches, we extend the Bayesian model of the data to introduce a latent encoding $\mathbf{h}_c$ that captures the user demographics that the adversary wishes to recover (e.g., gender, age, race, etc.). Joint recovery of the images and latent encoding, given a fixed $\nabla\mathbf{w}$, is guided by the probabilistic model

$$p(\mathbf{x}_{(1)}, ..., \mathbf{x}_{(B)}, \mathbf{h}_c|\nabla\mathbf{w}) \propto p(\mathbf{x}_{(1)}, ..., \mathbf{x}_{(B)}, \mathbf{h}_c, \nabla\mathbf{w}) \tag{4}$$

$$= p(\nabla\mathbf{w}|\mathbf{x}_{(1)}, ..., \mathbf{x}_{(B)})p(\mathbf{h}_c|\mathbf{x}_{(1)}, ..., \mathbf{x}_{(B)})p(\mathbf{x}_{(1)}, ..., \mathbf{x}_{(B)}), \tag{5}$$

where the equality is due to $\mathbf{h}_c \to (\mathbf{x}_{(1)}, ..., \mathbf{x}_{(B)}) \to \nabla\mathbf{w}$ forming a Markov chain, and which suggests the reconstruction optimization

$$\underset{\mathbf{x}_{(1)},...,\mathbf{x}_{(B)},\mathbf{h}_c}{\arg\max} \underbrace{\log p(\nabla\mathbf{w}|\mathbf{x}_{(1)}, ..., \mathbf{x}_{(B)})}_{\text{grad matching}} + \underbrace{\log p(\mathbf{h}_c|\mathbf{x}_{(1)}, ..., \mathbf{x}_{(B)})}_{\text{consistency}} + \underbrace{\log p(\mathbf{x}_{(1)}, ..., \mathbf{x}_{(B)})}_{\text{image prior}}. \tag{6}$$

Hence, the adversary should consider three terms, where the third term is the image prior, which will be discussed in the next section. The first term implies the recovered images should produce the observed gradient, which is achievable through gradient matching. The second term implies that the images should be consistent with the latent encoding, i.e., sharing semantics due to being from the same user.

Exploiting this semantic similarity, while obtaining computational advantages, the adversary can instead aim to recover a single representative image $\hat{\mathbf{x}}$ that captures the semantics of the original batch $\{\mathbf{x}_{(i)}\}_{i=1}^{B}$ by solving

$$\hat{\mathbf{x}}^* = \arg\min_{\hat{\mathbf{x}}} -\log p(\nabla\mathbf{w}|\mathcal{A}(\hat{\mathbf{x}})) - \log p(\hat{\mathbf{x}}), \tag{7}$$

where $\mathcal{A}(\hat{\mathbf{x}}) = \{\mathcal{T}(\hat{\mathbf{x}})\}_{i=1}^{B}$ is a batch of B augmented images produced from $\hat{\mathbf{x}}$ with random semantics-preserving image transformation(s) $\mathcal{T}$ (i.e., $p(\mathbf{h}_c|\mathcal{T}(\hat{\mathbf{x}})) \equiv p(\mathbf{h}_c|\hat{\mathbf{x}})$), and use $\arg\max_{\mathbf{h}_c} p(\mathbf{h}_c|\hat{\mathbf{x}})$ as the predicted value for $\mathbf{h}_c$. This approach has the benefits of simplifying the reconstruction of a potentially large batch to a single image, which substantially reduces the computational cost and memory consumption, accelerates the optimization, and thereby enables the attack to be extended to larger batch sizes. Despite this, in practice optimizing the first term in Eq. 7 drives the synthetic image $\hat{\mathbf{x}}$ towards the direction where its produced gradient approximates the observed gradient computed from a batch of images, which is likely to result in $\hat{\mathbf{x}}$ converging to an unrealistic noisy image. To obtain a natural image, we need to employ a strong prior to regularize $\hat{\mathbf{x}}$ during the optimization process, which we introduce next.

2.3 Diffusion Prior for Gradient Inversion

2.3.1 Denoising Diffusion Probabilistic Model as Prior

The denoising diffusion probabilistic model (DDPM) [23] is a generative model based on a Markov chain capturing a forward diffusion process that progressively adds Gaussian noise, given by

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\boldsymbol{I}), \tag{8}$$

where $\mathbf{x}_0 := \mathbf{x} \sim q$ represents a data sample from the distribution being modelled, and $\beta_t \in (0, 1)$ are fixed noise schedule hyper-parameters. This forward process can be summarized as

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}), \tag{9}$$

where $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, and $\alpha_t := 1 - \beta_t$. The forward process contains no trainable parameters, and DDPMs ultimately model the data distribution through a parameterized reverse process, given by

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \sigma_t^2 \boldsymbol{I}) \tag{10}$$

where $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \approx q(\mathbf{x}_T)$, σ_t is specified as a function of β_t, and

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) := \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right), \tag{11}$$

where $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is a neural network parameterized by $\boldsymbol{\theta}$ that is essentially trained to denoise, i.e., by recovering $\big((\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)/\sqrt{1 - \bar{\alpha}_t}\big)$, given samples $\mathbf{x}_t$ drawn from the forward process in Eq. 9.

The training of a DDPM (see [23] for further details) involves minimizing a variational upper bound of the negative log likelihood of the model,

$$\mathbb{E}_q[-\log p_{\boldsymbol{\theta}}(\mathbf{x}_0)] \leq \mathbb{E}_q \left[-\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] =: L. \tag{12}$$

With further derivation and discarding terms that are constant with respect to the trainable parameters (see [23]), this likelihood bound can be expressed as

$$\sum_t w_t \mathbb{E}_{\mathbf{x}_0 \sim q, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\big(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t\big) \right\|^2 \right], \tag{13}$$

where the weights w_t are a function of the noise schedule β_t.

Similar to the procedure of [14], we use Eq. 13 as an effective proxy for the model likelihood, by evaluating it with respect to $q(\mathbf{x}_0) = \delta(\mathbf{x}_0 - \hat{\mathbf{x}}_0)$, i.e., the delta distribution centered on the estimate variable $\hat{\mathbf{x}}_0$. Combining this into Eq. 7, we obtain an overall inversion strategy employing a pre-trained DDPM model as a prior, given by

$$\hat{\mathbf{x}}^* = \arg\min_{\hat{\mathbf{x}}} -\log p(\nabla \mathbf{w} | \mathcal{A}(\hat{\mathbf{x}})) - \sum_t w_t \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\big(\sqrt{\bar{\alpha}_t}\hat{\mathbf{x}} + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t\big) \right\|^2 \right]. \tag{14}$$

2.3.2 Optimization

In our implementation, we employ cosine distance [5] as a proxy for the gradient matching term in Eq. 14, which allows the overall optimization to be solved with standard gradient descent. However, due to the stochasticity of the reverse diffusion process, in practice the optimization becomes stochastic and may produce different estimates from different runs that could potentially deviate from the observed gradient. To ensure the production of reliable results that satisfy both the diffusion prior and the gradient constraint, ideally, the early stage of the optimization should coarsely explore the search space, while the later stage of the optimization should carefully fine-tune the estimated $\hat{\mathbf{x}}_0$ to steadily converge to a local maximum [14].

To this end, we make the following refinements to the optimization process. First, we observe that to achieve the best recovery, the gradient matching loss should be adjusted according to the phase of the optimization: in the early stage, we should first focus on recovering the correct high-level semantics such as gender and race; then the later stage we can extend to reconstruct detailed facial attributes. Therefore, instead of assigning equal weights to all gradients, we apply a sliding asymmetric Hamming window function to dynamically adjust the weight of the gradient of each layer. Second, instead of simultaneously optimizing for all t, we optimize according to a schedule that gradually anneals from T to a small value t_{min}. Third, similar to [23], we use constant weights w_t to simplify the optimization objective. To balance the two terms, we instead clip the gradient norm of the gradient matching loss dynamically according to the gradient norm of the diffusion prior loss term. The full algorithm and a detailed description of the procedure are included in the Appendix.

3 Experiments

Setup. We consider the binary classification task of attractiveness on the CelebA dataset [15]. The images are resized to 64×64 px and gradients are computed on a randomly initialized ResNet-18 with a default batch size of 30. To simulate cross-device FL settings, images are grouped according to user ID. The experiments use the first 50 users with ≥ 30 images as the evaluation set.

Table 1: Numerical comparison of the proposed method and baselines on the CelebA dataset.

(a) Image quality

	MSE↓	PSNR↑	LPIPS↓
Intra-user	0.4981	3.7879	0.4392
Inverting	0.5163	3.2927	0.7324
w/ disambiguation	0.3192	5.2633	0.7165
GIAS	0.5024	3.5402	0.6173
w/ disambiguation	0.2820	5.8809	0.5887
DiffULA	0.4210	4.6789	0.5039

(b) Semantics

	Face Detection Rate↑	Facial Similarity↑	Gender Accuracy↑	Race Accuracy↑	Age Error↓
Original Batch	0.95	0.7112	0.94	0.89	2.37
Inverting	0.03	0.3939	0.16	0.12	2.54
w/ ensemble	(0.22)	0.1275	0.44	0.48	4.91
GIAS	0.17	0.5986	0.18	0.84	2.64
w/ ensemble	(0.96)	0.6129	0.52	0.80	2.90
DiffULA	1.00	0.5740	0.71	0.29	3.61

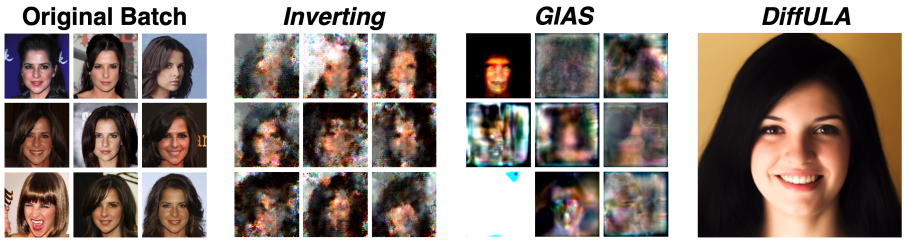


Figure 2: Visual comparison of reconstruction results from a batch of 30 images.

Baselines. We compare user-level attack with diffusion prior (*DiffULA*) with two baselines, namely, *inverting gradients* [5] based on total variation image prior and *GIAS* [8] based on GAN prior. We set hyperparameters by following original works [5, 8]. For GIAS, we use the state-of-art Diffusion-StyleGAN [24] pretrained on CelebA for generating 64×64 px images. We note that this is an unrealistically strong threat model, as the private evaluation images were included in the GAN training, however this can be viewed as a conservative upper bound on this baseline performance. For the diffusion prior, we use the DDPM weights from prior work [25], which are pre-trained on the FFHQ dataset [26] for generating 256×256 px images.

Evaluation Metrics. Following prior works [5, 6, 8, 7], we use image space Mean-Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Learned Perceptual Image Patch Similarity (LPIPS) [27] as image quality metrics. Additionally, we introduce the following metrics for measuring the semantic similarity between the original and reconstructed images: (1) *Face Detection Rate*: ratio of reconstructions with at least one image that can be successfully detected by an MTCNN [28] face detector; (2) *Gender Accuracy*: ratio of reconstructions with correct gender; (3) *Race Accuracy*: ratio of reconstructions with correct race; (4) *Age Error*: average error between the age of original images and the reconstructed images; (5) *Facial Similarity*: the similarity between the original and the reconstructed images measured in the embedding space. We use pre-trained neural networks provided in the DeepFace [29, 30] package for evaluating (1)-(5).

Image Quality. Table 1a presents the results on reconstructed image quality with respect to the original data. For reference, we also report the average pairwise (intra-user) score measured across the original private image batch. We observe that DiffULA achieves the best image quality scores compared to inverting and GIAS. However, as the optimization problem in Eq. 1 is permutation-invariant, the reconstructed images by sample-level attacks are arbitrarily ordered, which creates ambiguity for comparing pair-wise image quality. To disambiguate, we apply the Jonker-Volgenant algorithm [31] to solve for the optimal original-reconstructed image pairing that minimizes total MSE. Although the measured MSE and PSNR of inverting and GIAS improved significantly after disambiguation and even surpassed the intra-user reference, this result does not agree with the visual comparison shown in Figure 2 (see Appendix A for the full version). Overall, LPIPS is the most interpretable metric that is better aligned with human perception.

Semantics. Table 1b compares the performance in terms of facial semantics. We also report the results on the original batch as a reference for the experimental errors induced by the variation in the original images and the inaccuracies of the facial attribute predictors. We notice that inverting and GIAS may generate extremely distorted images that fail to be detected by face detectors. To account

for this effect, we employ a simple batch ensemble strategy to disregard images without a detected face within each batch for evaluation. When there is no single image with a detected face in the batch, we set the attack to random guessing. The results show that reconstructed images by inverting and GIAS have a very low sample-level face detection rate (3% and 17%). Applying the ensemble strategy, 22% of reconstructed batches by inverting and 96% by GIAS have at least a single image with a detectable face. Although DiffULA is able to reliably generate realistic facial images (100% detection rate), it shows no advantage over baselines in terms of recovering most facial attributes (except for gender). This indicates that the reconstructed images by inverting and GIAS might still contain useful cues for recovering private information despite their poor visual quality.

Computational Costs. We measure the computational times for running the attack on an Nvidia A100 GPU. Inverting gradients takes 23.2 minutes to run 24,000 optimization steps. GIAS takes 60.7 minutes to run 1,000 latent space search steps and 8,000 parameter space search steps. In comparison, DiffULA, which requires 4,000 optimization steps, takes only 23.3 minutes. This computational advantage enables DiffULA to be applied to larger batch sizes that were previously hard to achieve with traditional methods (e.g., see Appendix B for results with batch size of 100).

4 Conclusion

This work investigates user-level gradient inversion as a new attack surface in distributed learning. We first examine existing attacks in this new scenario and reveal the inefficacy of the existing method on practical batch sizes. We then explore employing diffusion prior for efficient reconstruction of representative images from large batches of images under the user-level assumption. Our study highlights the importance of designing pragmatic attacks that recover semantically meaningful private information under practical assumptions, as well as the need for better privacy metrics that are more aligned with human intuition, which are both interesting avenues for future research.

References

[1] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[2] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[3] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[4] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

[5] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[6] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

[7] Zhuohang Li, Jiaxin Zhang, Luyang Liu, and Jian Liu. Auditing privacy defenses in federated learning via generative gradient leakage. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10132–10142, 2022.

[8] Jinwoo Jeon, Kangwook Lee, Sewoong Oh, Jungseul Ok, et al. Gradient inversion with generative image prior. *Advances in neural information processing systems*, 34:29898–29908, 2021.

[9] Reza Bayat. A study on sample diversity in generative models: Gans vs. diffusion models. *International Conference on Learning Representations*, 2023.

[10] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005*, 2021.

[11] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in Neural Information Processing Systems*, 35:23593–23606, 2022.

[12] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022.

[13] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *International Conference on Learning Representations*, 2023.

[14] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35:14715–14728, 2022.

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision*, December 2015.

[16] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34:7232–7241, 2021.

[17] Maximilian Lam, Gu-Yeon Wei, David Brooks, Vijay Janapa Reddi, and Michael Mitzenmacher. Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix. In *International Conference on Machine Learning*, pages 5959–5968. PMLR, 2021.

[18] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[19] Sanjay Kariyappa, Chuan Guo, Kiwan Maeng, Wenjie Xiong, G Edward Suh, Moinuddin K Qureshi, and Hsien-Hsin S Lee. Cocktail party attack: Breaking aggregation-based privacy in federated learning using independent component analysis. In *International Conference on Machine Learning*, pages 15884–15899. PMLR, 2023.

[20] Mislav Balunović, Dimitar I Dimitrov, Robin Staab, and Martin Vechev. Bayesian framework for gradient leakage. *International Conference on Learning Representations*, 2022.

[21] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[22] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[24] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. *International Conference on Learning Representations*, 2023.

[25] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *International Conference on Learning Representations*, 2022.

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[27] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[28] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016.

[29] Sefik Ilkin Serengil and Alper Ozpinar. Lightface: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 23–27. IEEE, 2020.

[30] Sefik Ilkin Serengil and Alper Ozpinar. Hyperextended lightface: A facial attribute analysis framework. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–4. IEEE, 2021.

[31] David F Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

[32] Nicolas Pinto, Zak Stone, Todd Zickler, and David Cox. Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook. In *CVPR 2011 workshops*, pages 35–42. IEEE, 2011.

[33] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

A Visualization

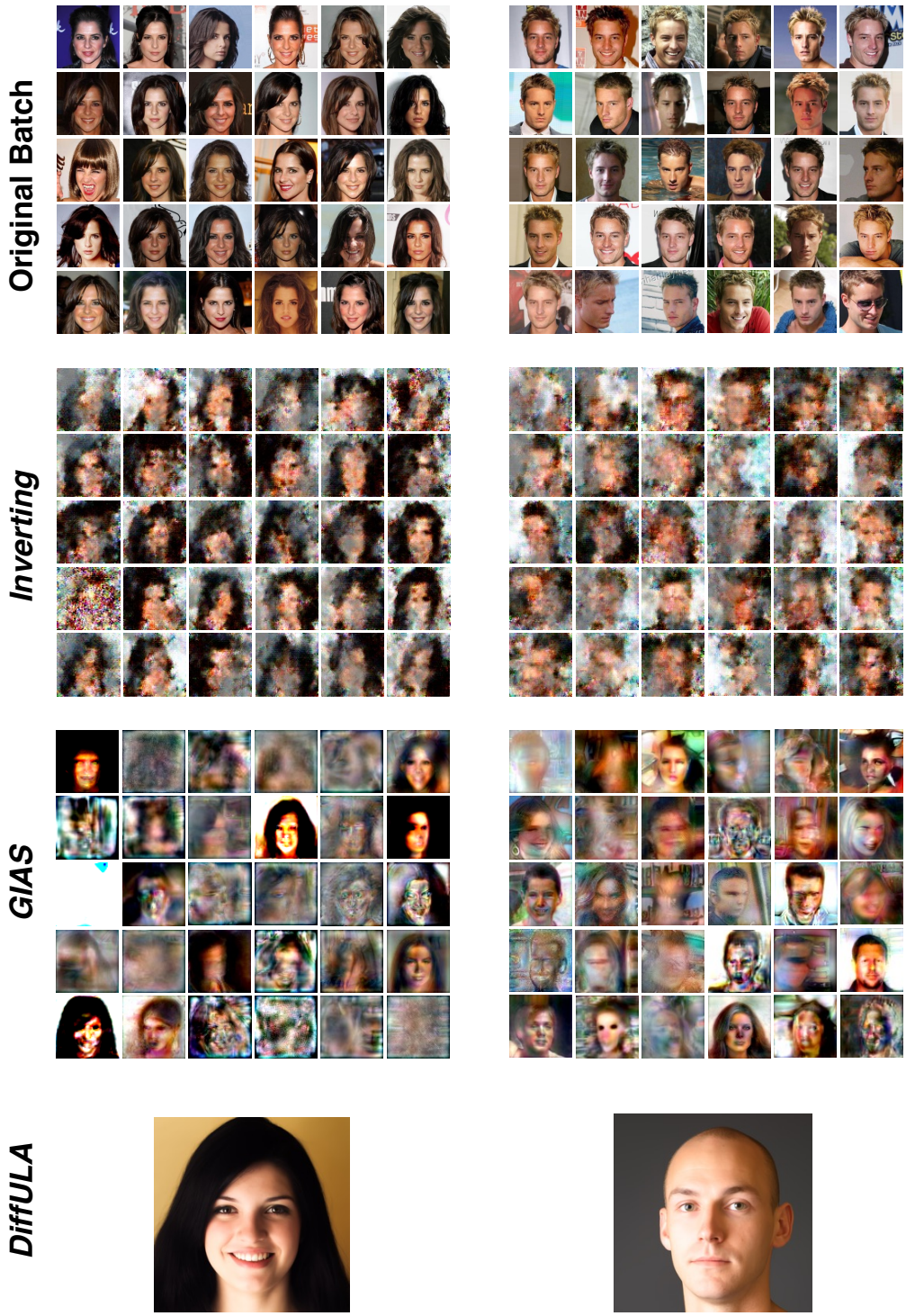


Figure 3: Visual comparison of reconstruction results (full batch).

Table 2: Numerical results with a batch size of 100.

	MSE↓	PSNR↑	LPIPS↓	Face Detection Rate↑	Facial Similarity↑	Gender Accuracy↑	Race Accuracy↑	Age Error↓
DiffULA	0.4028	3.1416	0.5101	1.00	0.5859	0.57	0.86	8.08

B Results on Large Batch Size

To explore the feasibility of reconstructing from larger batch sizes using DiffULA, we take the first 100 images (resized to 64×64 px) from the first 7 celebrities from the PubFig83 dataset [32]. Figure 4 provides a visualization of the reconstruction and the numerical scores are summarized in Table 2. We observe similar performance as with a batch size of 30.

C Implementation Details

We employ a cosine-modulated linear annealing time schedule with added uniform noise similar to [14]. As shown in Figure 5a, τ is scheduled to start from $1,000$ and gradually decreases to 500. We balance the effects of recovering the user's private image and generating a realistic facial image by clipping the norm of the gradient of the gradient matching term. The clipping scale factor ζ is scheduled to follow a cosine ramp down from 1.5 to 1.0. Additionally, as visualized in Figure 5b, we apply a window function to focus on the deeper layers and gradually introduce the shallower layers to the computation of the gradient-matching term as the optimization progresses.

We implemented $\mathcal{T}$ as a series of random image transformations, including Gaussian noise injection, random color jitter, random perspective transformation, and Gaussian blur. Each transformation has a probability of 0.5 to be applied for a given image. As the synthesized image from DDPM is of higher resolution, they are further resized before being fed into the learning model $f_{\mathbf{w}}$.

After optimization, we run some additional denoising steps on the reconstructed image using DDPM starting from $t^* = 200$ to further improve image quality. The complete procedure of the proposed attack is presented in Algorithm 1.

D Discussion

One inherent limitation of the proposed single-image-based, user-level gradient inversion is that it suffers from the "regression toward the mean" effect, that is, the best reconstruction of this strategy is to approximate the average of images in the batch. As such, its reconstruction performance highly depends on the image variance within the batch.

Many previous works have reported observing varying reconstruction performance across different images, i.e., some images are innately more susceptible to gradient inversion than others (e.g., Figure 3 in [5]). Currently, there is a lack of theoretical explanation for this phenomenon. The diffusion approach may perform poorly if the target image is not well-represented in the prior distribution $p_\theta(\mathbf{x}_0)$, which is a common limitation shared with all reconstruction methods relying on a generative prior. Additionally, in our experiments, we find that the utilization of diffusion prior may amplify this invariance due to the instability of reconstruction caused by the stochasticity of the diffusion process. For instance, as shown in Figure 8, we observe that in most of the failed cases, the reconstruction diverges when it approaches the end of the process. This is likely because the diffusion prior is prone to hallucinate when optimizing low t values where it attempts to add details to the image. This can also be seen from the progression of the facial similarity score of the reconstructed image during the optimization process, as shown in Figure 6. Adjusting the time schedule to only optimize for high t values may improve the accuracy of the inferred private attributes, but would result in generating lower-quality images, i.e., there is an inherent trade-off between the reconstruction of high-level semantics and image details.

Regarding evaluation, we note that the facial semantics analysis relies on deep learning models that are trained on different data distributions which may introduce noise to the derived predictions. For instance, the face embedding model is trained on VGG Face data, while the DDPM is trained on the

Algorithm 1: User-level gradient inversion with diffusion prior

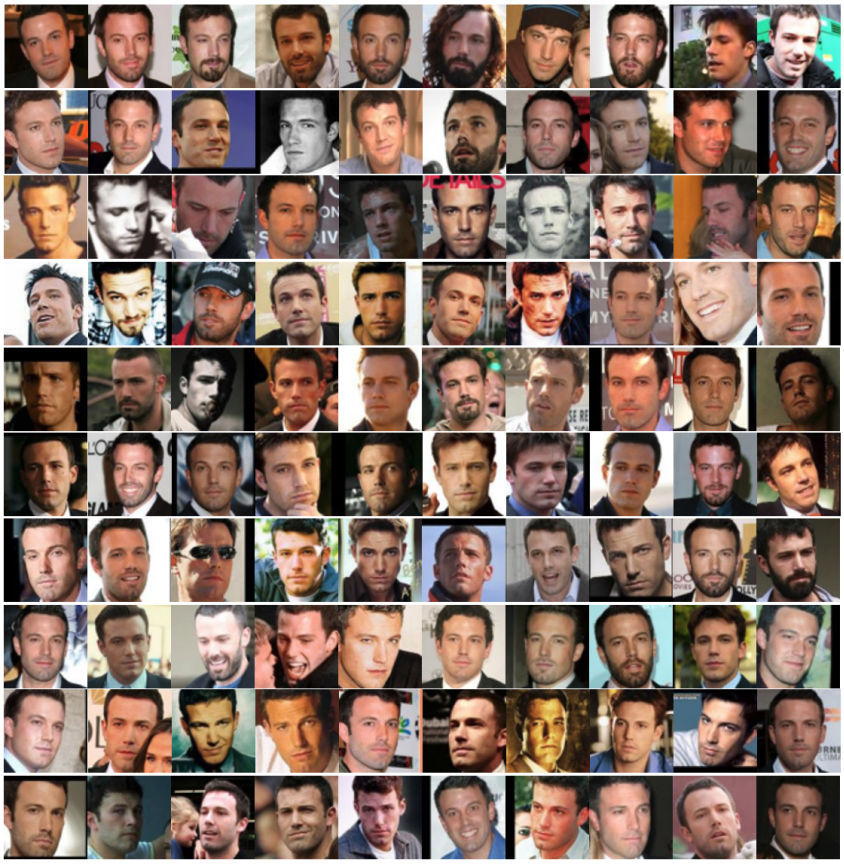
Input : observed gradient $\nabla \mathbf{w}$, model parameters $\mathbf{w}$, pretrained DDPM $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$
Parameter : number of steps S, image transformation(s) $\mathcal{T}$, time schedule $\{\tau_i\}_{i=1}^{S}$, gradient clipping schedule $\{\zeta_i\}_{i=1}^{S}$, denoising timestep t^*, learning rate η

Randomly initialize $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
for $i = 1, ..., S$ **do** `// Optimization steps`
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
  $\mathbf{x}_{\tau_i} \leftarrow \sqrt{\bar{\alpha}_{\tau_i}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{\tau_i}}\boldsymbol{\epsilon}$;
  $\mathbf{g}_{\text{p}} \leftarrow \nabla_{\mathbf{x}_0}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_{\tau_i}, \tau_i)||^2$; `/* Evaluate prior term */`
  $\mathcal{A}(\mathbf{x}_0) \leftarrow \{\mathcal{T}(\mathbf{x}_0)\}_{i=1}^{B}$;
  $\mathbf{g}_{\text{gm}} \leftarrow \nabla_{\mathbf{x}_0}\mathbf{d}\big(F(\mathcal{A}(\mathbf{x}_0)), \nabla\mathbf{w}\big)$; `/* Evaluate gradient matching term */`
  $\mathbf{g}_{\text{gm}} \leftarrow \mathbf{g}_{\text{gm}}/\max(1, \frac{||\mathbf{g}_{\text{gm}}||}{\zeta_{\tau_i}||\mathbf{g}_{\text{p}}||})$;
  $\mathbf{x}_0 \leftarrow \mathbf{x}_0 - \eta \cdot (\mathbf{g}_{\text{p}} + \mathbf{g}_{\text{gm}})$; `/* Update via gradient descent */`
end
$\mathbf{x}_{t^*} \leftarrow \mathbf{x}_0$;
for $t = t^*, ..., 1$ **do** `// Denoising steps`
  if $t > 1$ **then**
    Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
  else
    $\mathbf{z} \leftarrow \mathbf{0}$;
  end
  $\mathbf{x}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\big) + \sigma_t\mathbf{z}$;
end
$\hat{\mathbf{x}} \leftarrow \mathbf{x}_0$, $\mathbf{h}_c \leftarrow \arg\max p(\mathbf{h}_c|\mathbf{x}_0)$;
return reconstructed image $\hat{\mathbf{x}}$, inferred $\mathbf{h}_c$

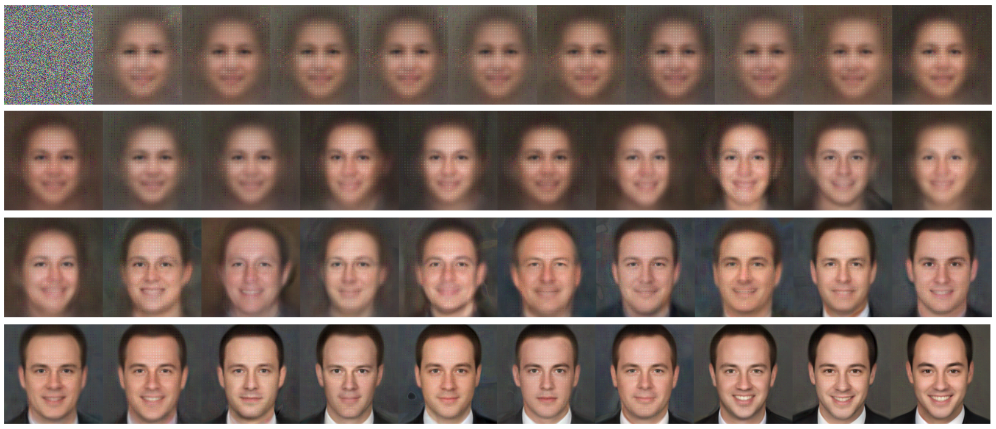
FFHQ dataset. In addition, the data selected for experiments are skewed, as shown in Figure 7, which may cause the results to be biased toward certain demographics.

E Alternative Formulation

There are several alternative ways to implement the user-level inversion as described in Eq. 6. For instance, apart from the single-image-based formulation explored in this paper, one may modify the sample-level approach to enforce the consistency of the latent encoding in the reconstructed images through the guidance of external classifiers. Alternatively, an adversary with access to the private data distribution may learn a model to directly predict $\mathbf{h}_c$ and further leverage the recovered $\mathbf{h}_c$ as a condition for synthesizing private images, i.e., model $p(\mathbf{h}_c|\nabla\mathbf{w})$ and then sample from $p(\mathbf{x}|\mathbf{h}_c) \propto p_{\boldsymbol{\theta}}(\mathbf{x})p(\mathbf{h}_c|\mathbf{x})$. Regarding the utilization of the diffusion prior, an alternative approach is to use guided sampling from diffusion with modified conditions [33, 13]. We leave explorations in these spaces as future work.



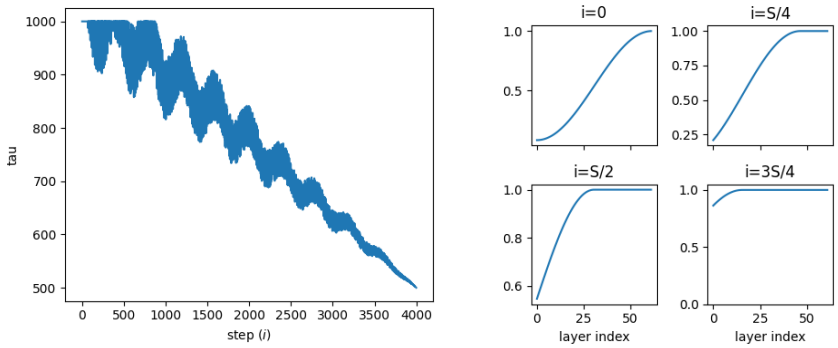
(a) Original batch of 100 images



(b) Reconstructed image of every 100 optimization steps

(c) Denoised image

Figure 4: Example of reconstructing from a batch of 100 images. The measured pairwise facial similarity in the private batch ranges from 0.4371 to 0.9528. The average facial similarity of the reconstructed image to the original batch is 0.7080.



(a) Time schedule (b) Window function

Figure 5: Time schedule and sliding window for optimization.

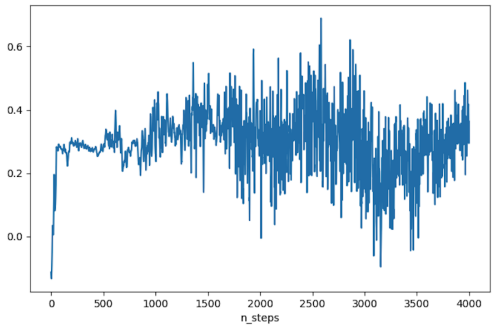


Figure 6: Facial similarity score per optimization step.

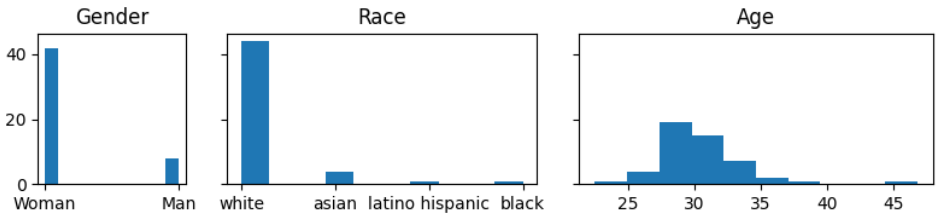


Figure 7: Demographics of the users involved in experiments.

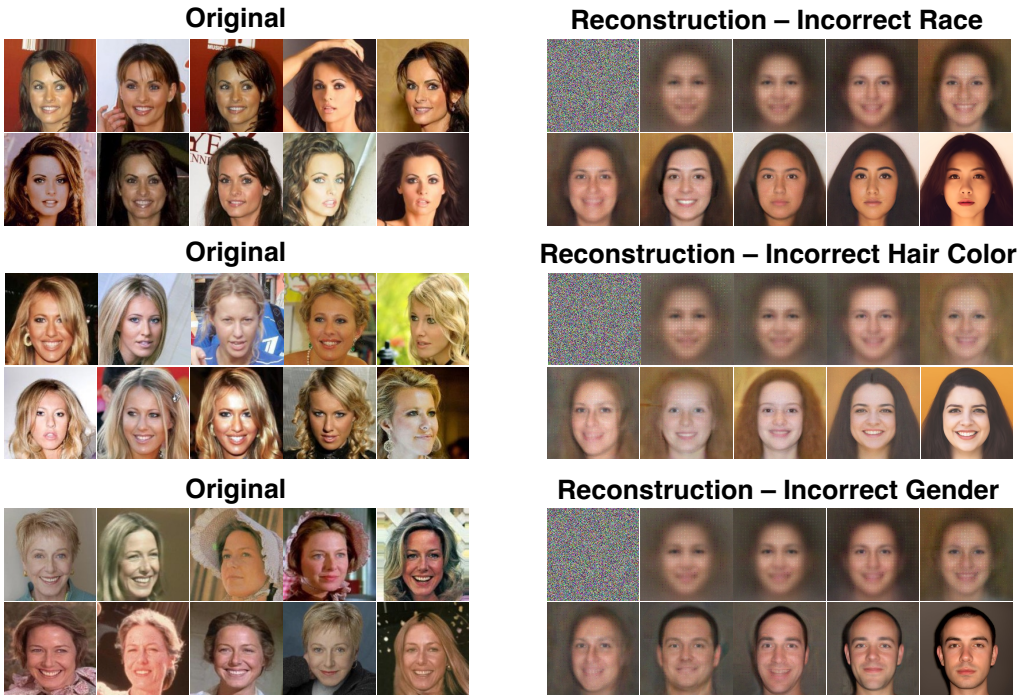


Figure 8: Reconstruction process of failed cases.