PRIVACY-PRESERVING PERSONALIZED FEDERATED LATENT DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling differentially private federated diffusion models to high-dimensional images requires overcoming two barriers: increasingly damaging privacy noise and rising training costs. While latent diffusion models offer a promising way to address these barriers, the properties of their latent representations that favor private collaborative learning remain underexplored. We propose Personalized Federated Training of Latent Diffusion Models (PF-LDM): a shared server-side latent diffusion model learns cross-client structure from privatized latent representations, while client models refine generated samples to recover fine-grained local details. Our theory and experiments link privacy–utility gains to the distribution of latent representations, rather than to their reduced dimensionality alone. Specifically, we prove that PF-LDM satisfies local differential privacy for each client and derive utility guarantees for learning a Gaussian mixture model. These guarantees and empirical findings show how collaboration, simpler structures (e.g., fewer mixture components), and more discriminative latent features improve learning. Extensive experiments on CelebA-HQ, FFHQ, and the medical imaging dataset CheXpert show that PF-LDM generates high-fidelity images and improves performance on underrepresented classes across clients while protecting client data privacy.

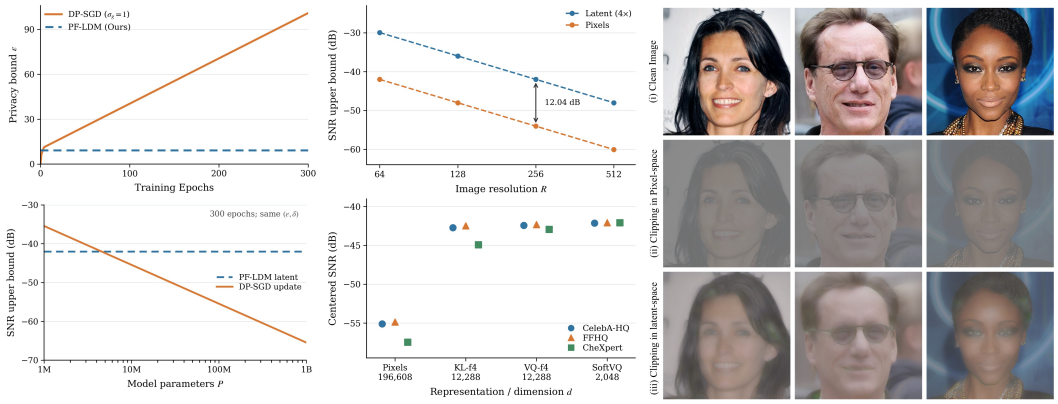
1 INTRODUCTION

Diffusion models (DMs) have transformed visual content creation (Ho et al., 2020; Lu et al., 2023), but their adoption in privacy-sensitive domains such as healthcare remains challenging. Training effective DMs requires representative data, yet institutions often cannot freely pool such data due to legal, ethical, and organizational constraints (GDPR, 2016; Cal. Legis. Serv., 2020; Price & Cohen, 2019; Epstein et al., 2023). While Federated learning (FL) (McMahan et al., 2017) offers a promising way to learn from these siloed datasets without sharing raw images, shared gradients or model parameters can still reveal sensitive information about training examples (Boenisch et al., 2023; Pichler et al., 2023; Geiping et al., 2020; Fowl et al., 2021; Melis et al., 2019). This privacy risk is particularly concerning for diffusion models, which can memorize training images and generate near-exact copies (Carlini et al., 2023; Gu et al., 2025; Somepalli et al., 2023). Therefore, generating synthetic images from sensitive data requires new privacy-preserving approaches beyond FL alone.

Differential privacy (DP) is a framework that offers formal guarantees for protecting training data (Dwork et al., 2006; Ponomareva et al., 2023) and has enabled the private training of large generative models (Sinha et al., 2025). Recent methods train DMs with DP guarantees but typically require centralized access to private data (Dockhorn et al., 2023; Li et al., 2024; Jiang et al., 2025; Liu et al., 2024). Moreover, these methods rely on differentially private stochastic gradient descent (DP-SGD) (Abadi et al., 2016), which privatizes gradients at each training step. As a result, privacy costs accumulate throughout training, computational demands grow with model and dataset size, and larger models can incur greater utility loss from privacy noise (see Figure 1(a)).

Patel et al. (2026) enabled private federated training of DMs by having clients clip images and add calibrated noise in pixel space to meet a target privacy guarantee before sharing them with the server. Clients release these privatized images only once, so further server training adds no extra privacy cost. However, pixel-space privatization has a dimension-dependent limit: for a fixed privacy budget, the upper bound on the signal-to-noise ratio (SNR) decreases inversely with image dimension (see Figure 1(b) and Appendix A). Moreover, clipping can distort the images even before noise is added

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) SNR comparison between DP-SGD and PF-LDM.

(b) SNR under pixel- and latent-space privatization

(c) Clipping-induced visual distortion in pixel and latent spaces.

Figure 1: Privacy and signal preservation under gradient, pixel, and latent perturbation. (a) The DP-SGD privacy bound increases with training duration at a fixed noise multiplier (top), whereas for a fixed privacy budget, its gradient SNR upper bound decreases with model size (bottom). PF-LDM decouples from such a limitation with a single privatized release. (b) PFDM's pixel-space privatization yields lower SNR as image dimension grows, while PF-LDM's compressed latents maintain higher SNR and (c) less clipping-induced visual distortion in the illustrated examples.

(see Figure 1(c)). As a result, in high dimensions, clipping and noise can obscure image features and reduce the fidelity of generated samples.

Latent diffusion models (LDMs) can alleviate this bottleneck by enabling training and privatization in a lower-dimensional space (Rombach et al., 2022), thereby reducing computational cost and overall privacy noise. However, autoencoders differ in what they preserve and in their latent distributions (Kingma & Welling, 2013; Razavi et al., 2019; Van Den Oord et al., 2017; Chen et al., 2025a;b). In particular, stronger compression may discard details needed for generation, so it need not improve utility. We therefore ask: *how can LDMs support private collaboration, and which properties of their representations, beyond dimensionality, improve utility?*

Our contributions. We answer this question through algorithm design, theory, and experiments:

- **Personalized Federated Latent Diffusion.** We introduce Personalized Federated Training of Latent Diffusion Models (PF-LDM), extending Patel et al. (2026)'s collaborative framework to latent space. Clients use a shared, fixed public encoder and release clipped, noisy latents once for the server to learn cross-client structure. Client models stay local, learn from clean latents, and refine server-generated samples with local details. PF-LDM supports personalized generation and representation comparisons under a common private collaboration protocol (Algorithms 1 and 2).
- **Formal Guarantees and Representation Structure.** We provide formal sample-level local differential privacy guarantees for PF-LDM (Theorem 4.1), with the privacy budget determined by the diffusion parameters and the degree of personalization. We further establish utility guarantees for learning Gaussian mixture models (Theorem 4.2), with the utility interpolating between non-collaborative training (the private extreme) and centralized training (completely non-private). Our result shows how lower-dimensional, structured latent spaces with fewer Gaussian mixture components can facilitate more effective diffusion model training. We also prove that collaboration improves learning when the global model more accurately approximates the noisy target than the noisy distribution implied by local-only learning. Our empirical analysis further suggests that more discriminative latent features can support better generation quality (Figure 3).
- **Empirical Evaluation.** Through experiments on CelebA-HQ, FFHQ, and CheXpert we show that PF-LDM can generate high-fidelity 256^2 images through collaboration while providing differential privacy guarantees for the shared models. Compared with non-collaborative training, PF-LDM improves generation quality for locally underrepresented groups (Table 1) and downstream classification on CelebA-HQ and FFHQ (Table 2). We further compare latent representations and denoiser architectures across datasets of varying sizes. In FFHQ privacy evaluations, the strongest membership inference attacks achieve near-chance AUROC against PF-LDM's shared models, versus near-perfect AUROC against non-private centralized baselines (Table 3).

2 PRELIMINARIES

Throughout the paper we use $[n]$ to denote the set $\{1, \ldots, n\}$.

Problem Setup. We consider M clients that collaboratively train personalized LDMs. Each client $m \in [M]$ holds a private image dataset $D_m = \{x^{i,m}\}_{i=1}^{n_m}$ drawn from a client-specific distribution q_0^m. Our goal is to learn models whose generated-image distributions $\{p^m\}_{m=1}^M$ approximate the corresponding client distributions, while preserving the privacy of each client's data. All clients share an autoencoder (AE), with encoder Enc and decoder Dec, pretrained on public data and kept fixed throughout training. The encoder maps each image to a latent representation $z_0^{i,m} = \mathrm{Enc}(x^{i,m})$, and the diffusion training is then performed in the latent space. We treat class labels as public and protect image contents.

Latent Representations. Autoencoders compress images into latent representations from which a decoder reconstructs them. Spatial AEs used in LDMs (Rombach et al., 2022) include KL-regularized models, which apply a weak Kullback–Leibler penalty toward a standard normal prior (Kingma & Welling, 2013), and VQ-regularized models, which use a learned codebook for vector quantization (Van Den Oord et al., 2017; Esser et al., 2021). Both produce spatial latents of shape $C \times (H/f) \times (W/f)$, where H, W are the image dimensions, C is the number of latent channels, and f is the spatial downsampling factor. Token-based models such as MAETok (Chen et al., 2025a) and SoftVQ-VAE (Chen et al., 2025b) instead use Vision Transformer backbones (Dosovitskiy, 2020) to encode images into compact sequences of latent tokens.

Latent Diffusion. A denoising diffusion probabilistic model (DDPM) learns to reverse a fixed Gaussian noising process (Ho et al., 2020). For a latent training distribution $p(z)$ and noise schedule $\{\beta_t\}_{t=1}^T$, define $\alpha_t \triangleq 1 - \beta_t$ and $\bar{\alpha}_t \triangleq \prod_{s=1}^t \alpha_s$. The forward process corrupts $z_0 \sim p(z)$ as

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1 - \bar{\alpha}_t}\, \eta, \qquad \eta \sim \mathcal{N}(0, I). \tag{1}$$

In PF-LDM, $p(z)$ describes clean representations for client denoisers and privatized representations for the server denoiser. We train a denoiser ξ_θ to predict the added noise by minimizing

$$L_{\mathrm{LDM}}(\theta) = \mathbb{E}_{z_0 \sim p(z),\, t \sim \mathrm{Unif}([T]),\, \eta \sim \mathcal{N}(0,I)} \left[\|\eta - \xi_\theta(z_t, t)\|_2^2 \right]. \tag{2}$$

An LDM sampler iteratively denoises Gaussian noise and decodes the resulting latent into an image (Rombach et al., 2022). We use the U-Net (Ho et al., 2020) and Diffusion Transformer (DiT) (Peebles & Xie, 2023) backbones as denoisers, omitting public class conditioning from the notation. Appendix C gives the training and sampling algorithms. Training noise alone does not ensure privacy, and PF-LDM privatizes representations before clients release them.

Differential Privacy. We use differential privacy (DP) (Dwork et al., 2006) to formalize privacy in federated diffusion models.

Definition 2.1 ((ϵ, δ)-DP)**.** A randomized mechanism $\mathcal{A}$ satisfies (ϵ, δ)-DP if, for any adjacent inputs u and u', and any measurable set O, $\mathbb{P}[\mathcal{A}(u) \in O] \leq e^\epsilon \mathbb{P}[\mathcal{A}(u') \in O] + \delta$.

The adjacency relation specifies the privacy model. In central DP, u and u' are adjacent datasets, typically differing in one record, and privacy is enforced by a trusted curator. In federated learning, central DP can protect either individual records (sample-level DP) or an entire client's dataset (client-level DP) (McMahan et al., 2018; Kairouz et al., 2021). Client-level DP protects a larger unit but usually requires more noise, which can reduce utility.

In this work, we focus on local DP (Kasiviswanathan et al., 2011), where u and u' are individual inputs and each client privatizes its data before communication. This removes the need for a trusted server: the server only receives privatized outputs. This is well-suited to cross-silo settings, where each silo may contain records from many individuals. Per-sample local DP implies sample-level central DP through post-processing and offers stronger protection at the same privacy budget by eliminating the need for a trusted server.

Autoencoder Configuration Choices. In this paper, we evaluate KL- and VQ-regularized AEs with downsampling factor $f = 4$, denoted KL-$f4$ and VQ-$f4$, alongside SoftVQ-VAE with 64 latent tokens of dimension 32. These configurations differ in regularization, latent structure, and compression. For the remainder of this work, we collectively refer to these models as AEs.

A detailed discussion of other related works is provided in Appendix B.

3 METHODOLOGY

PF-LDM combines a global denoiser trained on pooled privatized latents with local denoisers trained on each client's data. The global model lets clients benefit from data across institutions, while local models tailor generation to each client's distribution. Algorithm 1 summarizes training with a shared, fixed public AE. Using the same collaboration protocol lets us compare different AEs.

Algorithm 1 Personalized Federated Training of Latent Diffusion Models (PF-LDM)

input Client datasets $\{D_m\}_{m=1}^M$, public encoder Enc, initial denoiser parameters θ_g and $\{\theta_m\}_{m=1}^M$, clipping radius C_z, local timestep t_0, noise schedule $\{\beta_t\}_{t=1}^T$, learning rate λ_{lr}

Set $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$
on each client $m \in [M]$ **do**
Encode all images: $Z_m = \{z_0^{i,m} = \text{Enc}(x^{i,m})\}_{i=1}^{n_m}$
Form clipped variants: $\bar{z}_0^{i,m} = \text{CLIP}(z_0^{i,m}, C_z)$ for all $i \in [n_m]$
Form flagged training examples: $\mathcal{Z}_m = \{(z_0^{i,m}, 0), (\bar{z}_0^{i,m}, 1)\}_{i=1}^{n_m}$
$\xi_{\theta_m} \leftarrow \text{T-LDM}(\mathcal{Z}_m, \{\beta_t\}_{t=1}^T, \theta_m, \lambda_{\text{lr}})$
for $i \in [n_m]$ **do**
    $\tilde{z}_0^{i,m} = \sqrt{\bar{\alpha}_{t_0}}\, \bar{z}_0^{i,m} + \sqrt{1 - \bar{\alpha}_{t_0}}\, \eta^{i,m}$ with $\eta^{i,m} \sim \mathcal{N}(0, I)$
end for
Send once: $\tilde{Z}_m = \{\tilde{z}_0^{i,m}\}_{i=1}^{n_m}$
end on client
on server do
Pool the clients' privatized datasets: $\tilde{Z} = \{\tilde{Z}_m\}_{m=1}^M$
$\xi_{\theta_g} \leftarrow \text{T-LDM}(\tilde{Z}, \{\beta_t\}_{t=1}^T, \theta_g, \lambda_{\text{lr}})$
end on server

output Global denoiser ξ_{θ_g} and local denoisers $\{\xi_{\theta_m}\}_{m=1}^M$, retained by their respective clients

Local Denoiser Training. Each client m encodes its images into latents $Z_m = \{z_0^{i,m}\}_{i=1}^{n_m}$ and clips them to radius $C_z > 0$, obtaining $\bar{z}_0^{i,m} = \text{CLIP}(z_0^{i,m}, C_z)$, where

$$\text{CLIP}(z, C_z) = z \cdot \min\{1, C_z / \|z\|_2\}. \tag{3}$$

In our experiments, each client trains one conditional denoiser $\xi_{\theta_m}(z_t, t, b)$ on both variants, with $b = 0$ for original latents and $b = 1$ for clipped latents.[1] We train these denoisers with T-LDM, which adds Gaussian noise at sampled timesteps and trains the model to predict it (Algorithm 3 in Appendix C). We supply the clipping flag b as an additional conditioning input. Public class labels remain implicit in the main-text notation and routine calls.[2] These local denoisers and their training data remain on the clients.

Private Releases and Global Training. Each client privatizes every latent once by adding Gaussian noise to its clipped variant at timestep t_0:

$$\tilde{z}_0^{i,m} = \sqrt{\bar{\alpha}_{t_0}}\, \text{CLIP}(z_0^{i,m}, C_z) + \sqrt{1 - \bar{\alpha}_{t_0}}\, \eta^{i,m}. \tag{4}$$

Here, $\eta^{i,m} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$. The clipping radius C_z bounds sensitivity, while t_0 sets the noise level; together they determine the privacy bound in Theorem 4.1. A smaller radius can discard more information through clipping. At fixed C_z, a larger t_0 strengthens privacy but leaves less information in the released latents whereas a smaller t_0 releases more information but weakens privacy.

Clients send their privatized datasets $\tilde{Z}_m = \{\tilde{z}_0^{i,m}\}_{i=1}^{n_m}$ to the server, which pools them and trains the global denoiser ξ_{θ_g} using T-LDM. The privatized latents serve as the training examples to which the server adds further diffusion noise. The global model therefore learns to generate representations that resemble the privatized releases rather than the original clean latents. Server training reuses this fixed release and incurs no additional privacy cost, regardless of model size or optimization steps.

Collaborative Generation. Forward diffusion can obscure fine details while retaining broader structure at intermediate noise levels (Patel et al., 2026; Rissanen et al., 2023). PF-LDM therefore uses the global model to generate a starting latent $\tilde{z}_0$ informed by pooled data, which a local

[1] In our experiments, SoftVQ-VAE latents are not clipped. For this AE, we train on the original latents, use Z_m instead of $\mathcal{Z}_m$ in Algorithm 1, and omit the clipping-flag arguments during training and sampling. [2] In T-LDM, b^i denotes the full conditioning information, including the public class label and any clipping flag.

model refines. The client starts at $z_{t_0} = \tilde{z}_0$, matching the noise level used to privatize the clipped latents, and then denoises using the clipping flags $\{b_t\}_{t=1}^{t_0}$. For non-collaborative generation, the client instead denoises an initial Gaussian sample with its local model and $b = 0$. Both modes decode the final latent with the public decoder Dec. Note that, Algorithm 2 describes DDPM sampling (Ho et al., 2020); in some of our experiments, we also use DDIM (Song et al., 2021).

Algorithm 2 Sampling with PF-LDM (DDPM)

input Global denoiser ξ_{θ_g}, local denoiser $\xi_{\theta_m}(z, t, b)$, public decoder Dec, local timestep t_0, noise schedule $\{\beta_t\}_{t=1}^{T}$, collaborative clipping flags $\{b_t\}_{t=1}^{t_0}$, mode $\in \{\text{COLLABORATIVE}, \text{NON-COLLABORATIVE}\}$

Set $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, and $\sigma_t^2 = \beta_t(1 - \bar{\alpha}_{t-1})/(1 - \bar{\alpha}_t)$, with $\bar{\alpha}_0 = 1$
if mode = COLLABORATIVE **then**
  $\tilde{z}_0 \leftarrow$ S-LDM$(\xi_{\theta_g}, \{\beta_t\}_{t=1}^{T})$
  Set $z_{t_0} \leftarrow \tilde{z}_0$
  for $t = t_0, t_0 - 1, \ldots, 1$ **do**
    Draw $\eta_t \sim \mathcal{N}(0, I)$ independently if $t > 1$; otherwise set $\eta_t = 0$
    $z_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(z_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \xi_{\theta_m}(z_t, t, b_t) \right) + \sigma_t \eta_t$
  end for
else if mode = NON-COLLABORATIVE **then**
  $z_0 \leftarrow$ S-LDM$(\xi_{\theta_m}(\cdot, \cdot, 0), \{\beta_t\}_{t=1}^{T})$
end if
Decode the generated latent: $\hat{x} \leftarrow \text{Dec}(z_0)$

output Generated image $\hat{x}$

4 Main Results

4.1 Privacy Guarantees

We first establish the privacy guarantee of PF-LDM for the global denoiser, trained only on privatized latent representations. Client-specific denoisers remain local and are never shared.

Theorem 4.1 (Privacy Guarantee of PF-LDM)**.** *Consider Algorithm 1 applied to the training datasets $D = \{D_m\}_{m \in [M]}$, with local diffusion timestep t_0, total number of diffusion steps T, noise schedule $\{\beta_t\}_{t=1}^{T}$, and latent clipping radius C_z. Let $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$, $\sigma^2 = (1-\bar{\alpha}_{t_0})/\bar{\alpha}_{t_0}$. Then, for all $x \in D$ and any $\delta \in (0, 1)$, the output ξ_{θ_g} returned by Algorithm 1 satisfies*

$$\left(\frac{2C_z^2}{\sigma^2} + C_z \sqrt{\frac{8 \log(1/\delta)}{\sigma^2}}, \delta \right) \text{-local DP.}$$

Theorem 4.1 gives an explicit upper bound on the privacy parameter ϵ and $\sigma^2 = (1 - \bar{\alpha}_{t_0})/\bar{\alpha}_{t_0}$ is the effective Gaussian noise variance. Increasing t_0 increases σ^2 and tightens the privacy bound. Together, t_0 and C_z control the privacy–utility tradeoff. Since each latent is privatized and released once, subsequent server training incurs no additional privacy cost.

For example, with $T = 1000$, a linear noise schedule, and clipping radius $C_z = 35$, choosing $t_0 = 850$ yields an (ϵ, δ)-local DP guarantee with $\epsilon = 10$ and $\delta = 10^{-5}$. Unlike central DP with a trusted server, this guarantee protects the image content before it reaches the server. Although $\epsilon = 10$ may appear large, prior work shows that DP mechanisms can still mitigate practical membership inference attacks at relatively large budgets (Jayaraman et al., 2020; Lowy et al., 2024).

A key advantage of privatizing in latent space is that latent representations are much lower-dimensional than pixel-space images. This compression can support a substantially smaller clipping radius C_z than the corresponding pixel-space radius while preserving the essential semantic and perceptual information needed for generation. At fixed effective noise variance σ^2, reducing C_z lowers both terms in the privacy bound, yielding a smaller ϵ. Moreover, as illustrated in Figure 1, latent-space clipping can be less destructive than pixel-space clipping: decoded clipped latents retain meaningful image structure, whereas pixel-space clipping washes out fine visual details in these examples. Thus, latent-space privatization can improve the privacy–utility tradeoff not only by reducing effective sensitivity but also by preserving more useful information after clipping.

4.2 Utility Guarantees

To characterize the utility and role of latent representations, we model latent distributions using Gaussian mixture models (GMMs). Following Chen et al. (2025a), we use the number of mixture components to measure distributional complexity. We assume that all clients share a clean latent distribution in $\mathbb{R}^d$ consisting of K equally weighted Gaussian components: $q = K^{-1}\sum_{j=1}^{K}\mathcal{N}(\mu_j, I_d)$. The goal of PF-LDM is to train a shared global denoiser ξ_{θ_g} and private client-specific denoisers $\{\xi_{\theta_m}\}_{m\in[M]}$ whose composition defines personalized densities that approximate q.

We study a GMM-based instance of denoiser training in PF-LDM, using the DDPM objective (Ho et al., 2020) and the denoiser parameterization of Shah et al. (2023). In this setting, learning the component means at a suitable noise level determines the learned scores throughout the diffusion process. The detailed setup and its connection to standard DDPM are provided in Appendix G.

According to PF-LDM, client m trains its private denoiser ξ_{θ_m} on the clean latent dataset $Z_m = \{z_0^{i,m}\}_{i=1}^{n_m}$, with $z_0^{i,m} \sim q$. The shared global denoiser ξ_{θ_g} is trained on the pooled privatized latents $\tilde{Z} = \{\tilde{Z}_m\}_{m\in[M]}$, constructed according to equation 4. Following collaborative sampling in Algorithm 2, let $p^m_{\widetilde{D},D_m}$ denote the personalized latent density obtained by composing the learned global noisy distribution with the learned local model's ideal stochastic reverse transition from t_0 to zero (see Appendix G). The following theorem characterizes its KL approximation error.

Theorem 4.2 (Informal, see Theorem G.5)**.** *Suppose the means of q satisfy $\max_{j\in[K]}\|\mu_j\|_2 \le B$ and are sufficiently separated. Assume each client m holds n_m latent samples, each with ℓ_2 norm at most C_z. Under the same choices of t_0, $\{\beta_t\}$, and C_z as in Theorem 4.1,* PF-LDM *learns a composed density $p^m_{\widetilde{D},D_m}$ such that, for any fixed client $m \in [M]$, with high probability,*

$$\mathrm{KL}\Big(q \,\Big\|\, p^m_{\widetilde{D},D_m}\Big) \le \widetilde{\mathcal{O}}\left[K^4 d^5\left(\frac{B^8}{N(1+\sigma^2)} + \frac{B^{10}\sigma^2}{n_m(1+\sigma^2)}\right)\right],$$

where $N = \sum_{m=1}^{M} n_m$, $\sigma^2 = (1-\bar{\alpha}_{t_0})/\bar{\alpha}_{t_0}$, and $\widetilde{\mathcal{O}}$ suppresses logarithmic factors.

Theorem 4.2 suggests that, compact representations d with simpler mixture structure K can facilitate accurate distribution learning and potentially improve generation quality. It also captures the privacy–utility tradeoff through σ^2, controlled by t_0. The utility approaches the *non-collaborative* rate $\widetilde{\mathcal{O}}(K^4d^5/n_m)$ (see Appendix G) as $\sigma^2 \to \infty$, corresponding to increasingly strong privacy protection. As $\sigma^2 \to 0$, privacy protection diminishes and the utility approaches the *non-private centralized* rate $\widetilde{\mathcal{O}}(K^4d^5/N)$. Intermediate noise levels interpolate between these extremes. This suggests potentially substantial benefits from collaboration since N can greatly exceed n_m.

To characterize the benefit of collaboration, let $p^m_{D_m}$ denote the local-only learned density and $p_{\widetilde{D}}$ the learned global noisy density. Let q_{t_0} and $p^m_{D_m,t_0}$ be the forward-noised versions of q and $p^m_{D_m}$, respectively. Since N can greatly exceed n_m, pooling can reduce global estimation error enough to offset privatization. If the global model is more accurate at the handoff level, namely $\mathrm{KL}(q_{t_0}\|p_{\widetilde{D}}) < \mathrm{KL}(q_{t_0}\|p^m_{D_m,t_0})$, then composition with the same learned local model strictly improves clean-distribution accuracy (see the proof in Appendix G):

$$\mathrm{KL}\Big(q \,\Big\|\, p^m_{\widetilde{D},D_m}\Big) < \mathrm{KL}\big(q \,\big\|\, p^m_{D_m}\big)\,.$$

5 Experiments

5.1 Utility Evaluations

Data Preparation. We evaluate our method on three datasets, CelebA-HQ (Karras et al., 2018), FFHQ (Karras et al., 2019), and CheXpert (Irvin et al., 2019; Chambon et al., 2024) at 256^2 resolution. We partition the dataset into two disjoint client datasets with imbalanced class distributions. Each client is assigned a majority group, from which it receives most of its samples, and a minority group, from which it receives only a small number to model underrepresented local data. For CelebA-HQ and FFHQ, groups are defined by gender labels (male & female), and for CheXpert, groups are defined by age (< 60 & ≥ 60 years). Based on data availability, we select the per-client

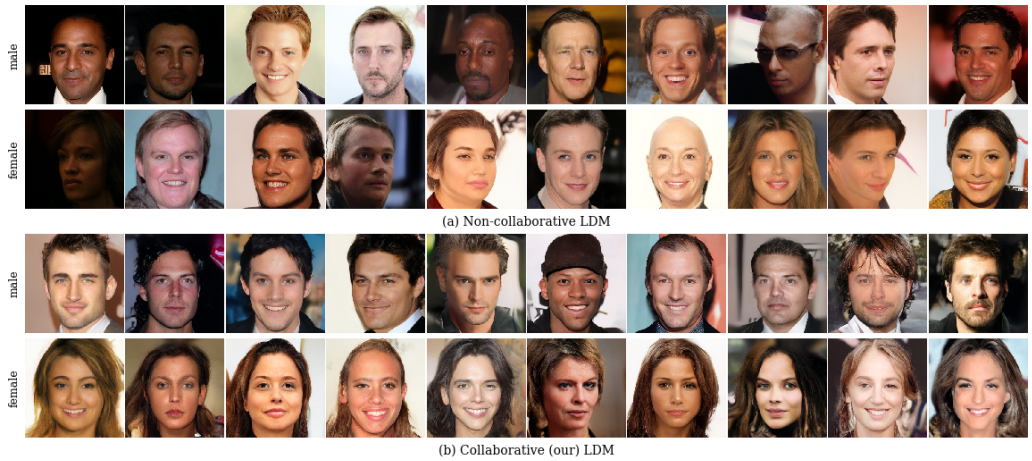


Figure 2: CelebA-HQ samples generated at 256^2 resolution. (a) Non-collaborative LDM samples show weaker minority-class (female) features. (b) PF-LDM collaboration improves generation across both majority and minority classes.

Table 1: FID scores across baselines and datasets. Lower is better.

Architecture	Baseline	CelebA-HQ FID (Maj/Min/Avg)	FFHQ FID (Maj/Min/Avg)	CheXpert FID (Maj/Min/Avg)
U-Net + KL-f4	NP	13.0659 / 11.7274 / 12.3967	8.5479 / 8.3941 / 8.4710	6.6533 / 6.9419 / 6.7976
	NC	15.1949 / 18.6017 / 16.8983	**10.0946** / 18.5595 / 14.3271	6.9846 / 8.7409 / 7.8627
	PF-LDM	**13.8089 / 16.4927 / 15.1508**	10.6924 / **15.9922 / 13.3423**	**6.4221 / 8.2418 / 7.3320**
U-Net + VQ-f4	NP	17.2868 / 13.7501 / 15.5184	8.2699 / 8.0414 / 8.1556	6.9634 / 6.4762 / 6.7198
	NC	17.8676 / 21.2057 / 19.5366	9.5294 / 17.2858 / 13.4076	6.7736 / 8.3863 / 7.5799
	PF-LDM	**15.9370 / 18.9965 / 17.4668**	**8.3923 / 15.9636 / 12.1780**	**5.9115 / 7.3607 / 6.6361**
DiT + SoftVQ-VAE	NP	16.9460 / 17.2961 / 17.1210	13.4380 / 14.6177 / 14.0279	11.6646 / 12.2773 / 11.9710
	NC	**19.2475** / 26.2116 / 22.7296	17.3684 / 41.1508 / 29.2596	**16.6921** / 18.6843 / **17.6882**
	PF-LDM	19.3425 / **25.7948 / 22.5686**	**14.8819 / 38.6082 / 26.7450**	16.9033 / **18.5300** / 17.7166

majority/minority sample counts as $5{,}000/50$, $15{,}000/150$, and $70{,}000/700$ for CelebA-HQ, FFHQ, and CheXpert, respectively (see Table 4 in Appendix D.1). This construction induces strong cross-client heterogeneity while limiting each client's access to its underrepresented group. The global denoiser is trained on the combined privatized latent releases from both clients, totaling $10{,}100$, $30{,}300$, and $141{,}400$ samples for the three datasets, respectively.

Modeling & Training Details. Following Rombach et al. (2022); Chen et al. (2025b), we use three different AEs to train LDMs: KL-$f4$, VQ-$f4$, and SoftVQ-VAE. The KL-$f4$ and VQ-$f4$ AEs are trained on OpenImages (Kuznetsova et al., 2020) dataset and the SoftVQ-VAE is trained on ImageNet (Deng et al., 2009) dataset at 256^2 resolution. All three AEs are publicly pretrained and never access private data during training, avoiding any privacy leakage. More details in Appendix D.2.

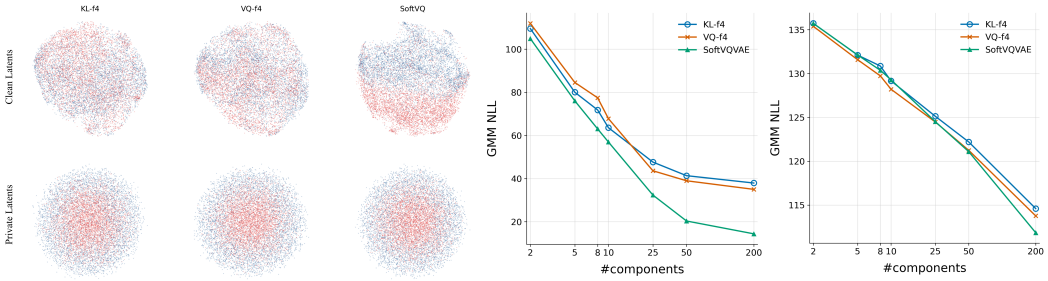
We use U-Net (Ronneberger et al., 2015) with KL-$f4$ and VQ-$f4$ AEs, and diffusion transformer (DiT) (Peebles & Xie, 2023) with SoftVQ-VAE as our main denoisers. All denoisers are class-conditioned on labels, and we assume the label information as public since our goal is to protect the contents of the image and not the designated labels. We train the denoisers on a linear noise schedule with $T = 1000$, and the privacy budget fixed at $\epsilon = 10, \delta = 10^{-5}$. This setting results in no clipping for the SoftVQ-VAE due to small latent norms.

Baselines. We consider the following baselines.

1. **Non-private Centralized Baseline (NP):** Non-private extreme; a single model is trained on all clients' pooled clean data without privacy protection.
2. **Non-Collaborative Baseline (NC):** Private extreme; each client trains independently on local data without sharing data or models.

Table 2: Classification accuracy and F1-score for downstream tasks.

Dataset	AE	CNN				ResNet			
		NC		PF-LDM		NC		PF-LDM	
		Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score
CelebA-HQ	KL-$f4$	71.223	71.089	**73.370**	**73.292**	59.020	56.955	**64.333**	**63.934**
	VQ-$f4$	62.943	62.836	**64.140**	**63.448**	63.340	61.646	**64.200**	**63.585**
	SoftVQ-VAE	60.747	59.850	**61.367**	**60.573**	82.830	82.417	**85.677**	**85.575**
FFHQ	KL-$f4$	74.997	74.731	**75.597**	**75.581**	55.230	50.839	**61.357**	**59.983**
	VQ-$f4$	58.760	57.487	**62.597**	**61.421**	74.140	74.102	**77.970**	**77.908**
	SoftVQ-VAE	69.163	67.330	**72.300**	**71.268**	72.863	71.220	**76.940**	**76.598**



(a) Latent-space Visualization

(b) GMM fitting on latent space (left: clean latents, right: private latents)

Figure 3: Latent-space analysis for clean and private FFHQ latents. (a) Visualization through UMAP showing discriminative structures. Latents are clustered with male and female class labels in red and blue. (b) GMM fitting across latent spaces shows the effect of varying the number of components.

3. **Collaborative (PF-LDM):** Collaborative and privacy-preserved baseline; clients share only privatized latents. Training and sampling follow the procedures in Section 3.

Results. We evaluate PF-LDM in terms of both generation quality and downstream task performance. For generation quality, we use Fréchet Inception Distance (FID) (Heusel et al., 2017) as the primary metric. CelebA-HQ and FFHQ use $25{,}000$ generated samples per majority/minority group ($50{,}000$ total), while CheXpert uses $15{,}000$ per group ($30{,}000$ total). The resulting FID scores are reported in Table 1. We further evaluate downstream classification accuracy and F1-score using $10{,}000$ generated samples for CelebA-HQ and FFHQ, as shown in Table 2. PF-LDM consistently improves minority-group FID and improves average FID in most settings. It also outperforms the NC baseline across all reported downstream classification accuracy and F1-score comparisons. Qualitative samples for CelebA-HQ are shown in Fig. 2 and FFHQ samples are provided in Fig. 4 in Appendix D.3.

What Favors Latent-space Collaboration? We examine latent-space properties that support collaboration from two perspectives. First, we assess whether discriminative structure survives privatization despite substantial Gaussian noise. We measure the class-balanced fraction of each sample's 30 nearest neighbors in a UMAP embedding that share its label. On FFHQ, the scores are 63.4% for KL-$f4$ and 66.1% for VQ-$f4$, both above the approximately 50% expected under random mixing. On CelebA-HQ, the ordering reverses, with 67.04% for KL-$f4$ and 66.63% for VQ-$f4$. These dataset-dependent rankings align with their relative FID performance, suggesting that stronger discriminative structure in privatized latents supports more effective global denoiser learning and, consequently, better contribution of coarse and global patterns during collaboration. UMAP visualizations are shown in Fig. 3(a).

Our second perspective is motivated by our utility analysis, which identifies mixture complexity as a key factor in the statistical difficulty of PF-LDM. To evaluate this effect, we fit GMMs with varying numbers of components and measure negative log-likelihood (NLL), where lower NLL indicates a better fit (Chen et al., 2025a). On FFHQ (Fig. 3(b)), privatized VQ-$f4$ achieves comparable NLL with fewer components than KL-$f4$, while their clean-latent behavior is similar. On CelebA-HQ (see Fig 5 in Appendix D.4), KL-$f4$ shows lower mixture complexity on clean latents, while the

Table 3: Membership inference attacks on FFHQ. For each target, we select the strongest attack family using labeled calibration data and report its mean performance over three attacker-fitting seeds on held-out test images. TPRs are attainable points on the empirical test ROC under the stated FPR cap. NP denotes the non-private baseline; PF-LDM denotes the shared server model. Larger values indicate greater privacy leakage.

Architecture	NP			PF-LDM		
	AUROC	TPR (%) @ 1% FPR	TPR (%) @ 0.1% FPR	AUROC	TPR (%) @ 1% FPR	TPR (%) @ 0.1% FPR
U-Net + KL-$f4$	0.9929	82.844	30.623	0.5012	0.825	0.171
U-Net + VQ-$f4$	0.9997	99.978	91.799	0.5003	1.119	0.099
DiT + SoftVQ-VAE	1.0000	100.000	100.000	0.4930	0.902	0.067
Chance reference	0.5000	1.000	0.100	0.5000	1.000	0.100

privatized representations behave similarly. In both cases, the representation with lower effective mixture complexity at one stage and comparable complexity at the other achieves better PF-LDM FID. Together with Theorem 4.2, this suggests that lower mixture complexity, and thus simpler structure, is a favorable property for private collaborative learning. SoftVQ-VAE's compact latents allow us to achieve the target privacy guarantee without clipping-induced distortion. However, its paired DiT denoiser benefits from large-scale training, whereas our datasets are considerably smaller than those used to demonstrate DiT's strong performance (Peebles & Xie, 2023).

5.2 Privacy Evaluations

We complement our formal privacy guarantee with an empirical membership-inference evaluation on FFHQ. The attacker receives a candidate image and tries to determine whether it helped train a denoiser. For each configuration, we compare the non-private centralized baseline (NP), trained on clean latents, with the PF-LDM shared server denoiser, trained on privatized latents from the same images at $(\epsilon, \delta) = (10, 10^{-5})$. The attacker has white-box access to the target denoiser and its noise schedule. It also knows the public encoder and gender label and has a separate pool of images with known membership labels. For PF-LDM, it can simulate fresh privatized latents from a candidate image but cannot see the uploads used in training.

We test whether membership leaves a trace in denoising behavior, using PIA and PIA-N (Kong et al., 2024), denoising-loss scores, and SecMI (Duan et al., 2023), or in parameter gradients, using GSA1/GSA2 (Pang et al., 2025). Across five timesteps, we evaluate ten learned attacks built from denoising scores or gradient features; we assess SecMI separately. For each target, we select the attack with the highest cross-validated AUROC on the labeled calibration pool, averaged over three fitting seeds, and report its mean performance across those seeds on disjoint test images. Table 3 shows strong membership leakage from NP, while the selected attacks against PF-LDM shared denoisers perform close to chance. Its TPRs are points on the empirical test ROC. Note that the audit does not assess the clients' local models. Appendix E gives the full protocols, results at calibration-fixed thresholds, and also reconstruction and source-identification attacks on privatized latent uploads.

6 Discussion

In this work, we introduced PF-LDM, a framework for federated LDM training with formal privacy guarantees. Our theoretical and empirical analyses identify latent discriminability and simpler structures as key properties that support effective private collaboration. We also demonstrate that PF-LDM achieves high-quality personalized generation while maintaining strong privacy protection. One limitation of our work is that client-specific models are not covered by the privacy guarantee and must remain local. Future work includes differentially private training of these models to enable sharing, evaluating compact tokenizers such as SoftVQ-VAE on larger datasets to assess their potential to avoid clipping-induced distortion at scale, and designing and pretraining latent representations tailored to private federated diffusion.

AI USE STATEMENT

In this work, we used generative AI tools to assist in reviewing the correctness of theoretical claims and in the writing and presentation of proofs. The research questions, methodological design, mathematical claims and derivations, experimental design, and interpretation of the main results were developed by the authors. Generative AI was not used to generate synthetic data, originate theoretical models or conceptual frameworks, or provide the core mathematical arguments underlying our theoretical results. Additionally, we used generative AI tools to assist with code development, plotting and refining figures, literature and information search, and writing refinement, including grammar, punctuation, and stylistic edits. All AI-assisted code, text, figures, theoretical arguments, and experimental materials were reviewed and verified by the authors. We take full responsibility for the final content of this work, including all claims and artifacts produced with the aid of generative AI.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Simeon Allmendinger, Domenique Zipperling, Lukas Struppek, and Niklas Kühl. Collafuse: Collaborative diffusion models. *Journal of Artificial Intelligence Research*, 86, 2026.

Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. In *International Conference on Learning Representations*, volume 2024, pp. 36916–36936, 2024.

Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. When the curious abandon honesty: Federated learning is not private. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, pp. 175–199. IEEE, 2023.

Cal. Legis. Serv. The California Privacy Rights Act of 2020, 2020. URL `https://oag.ca.gov/system/files/initiatives/pdfs/19-0021A1%20%28Consumer%20Privacy%20-%20Version%203%29_1.pdf`.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Pierre Chambon, Jean-Benoit Delbrouck, Thomas Sounack, Shih-Cheng Huang, Zhihong Chen, Maya Varma, Steven QH Truong, Chu The Chuong, and Curtis P Langlotz. Chexpert plus: Augmenting a large chest x-ray dataset with text radiology reports, patient demographics and additional image formats. *arXiv preprint arXiv:2405.19538*, 2024.

Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. In *Forty-second International Conference on Machine Learning*, 2025a.

Hao Chen, Ze Wang, Xiang Li, Ximeng Sun, Fangyi Chen, Jiang Liu, Jindong Wang, Bhiksha Raj, Zicheng Liu, and Emad Barsoum. Softvq-vae: Efficient 1-dimensional continuous tokenizer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28358–28370. IEEE, 2025b.

Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *International Conference on Machine Learning*, pp. 4735–4763. PMLR, 2023a.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=zyLVMgsZ0U_.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ZPpQk7FJXF.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In *International Conference on Machine Learning*, pp. 8717–8730. PMLR, 2023.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006.

Ziv Epstein, Aaron Hertzmann, Investigators of Human Creativity, Memo Akten, Hany Farid, Jessica Fjeld, Morgan R Frank, Matthew Groh, Laura Herman, Neil Leach, et al. Art and the science of generative ai. *Science*, 380(6650):1110–1111, 2023.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance), May 2016. URL http://data.europa.eu/eli/reg/2016/679/2016-05-04/eng. Legislative Body: OP_DATPRO.

Liam Fowl, Jonas Geiping, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. *arXiv preprint arXiv:2110.13057*, 2021.

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.

Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=D3DBqvSDbj.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 590–597, 2019.

Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

Tanqiu Jiang, Changjiang Li, Fenglong Ma, and Ting Wang. Rapid: Retrieval augmented training of differentially private diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 36547–36564, 2025.

Peter Kairouz and H Brendan McMahan. Advances and open problems in federated learning. *Foundations and trends in machine learning*, 14(1-2):1–210, 2021.

Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *International conference on machine learning*, pp. 5213–5225. PMLR, 2021.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hk99zCeAb.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 4396–4405. IEEE, 2019.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Fei Kong, Jinhao Duan, Heng Tao Shen, Xiaoshuang Shi, Xiaofeng Zhu, Kaidi Xu, et al. An efficient membership inference attack for the diffusion model by proximal initialization. In *International Conference on Learning Representations*, volume 2024, pp. 57876–57896, 2024.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

Kecen Li, Chen Gong, Zhixiang Li, Yuzhong Zhao, Xinwen Hou, and Tianhao Wang. {PrivImage}: Differentially private synthetic image generation using diffusion models with {Semantic-Aware} pretraining. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 4837–4854, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Michael F Liu, Saiyue Lyu, Margarita Vinaroz, and Mijung Park. Differentially private latent diffusion models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=AkdQ266kHj.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Andrew Lowy, Zhuohang Li, Jing Liu, Toshiaki Koike-Akino, Kieran Parsons, and Ye Wang. Why does differential privacy with large epsilon defend against practical membership inference attacks? *arXiv preprint arXiv:2402.09540*, 2024.

Yingzhou Lu, Lulu Chen, Yuanyuan Zhang, Minjie Shen, Huazheng Wang, Xiao Wang, Capucine van Rechem, Tianfan Fu, and Wenqi Wei. Machine learning for synthetic data generation: a review. *arXiv preprint arXiv:2302.04062*, 2023.

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*, pp. 1273–1282. Pmlr, 2017.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJ0hF1Z0b.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pp. 691–706. IEEE, 2019.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275. IEEE, 2017.

Yan Pang, Tianhao Wang, Xuhui Kang, Mengdi Huai, and Yang Zhang. White-box membership inference attacks against diffusion models. *Proceedings on Privacy Enhancing Technologies*, 2025(2):398–415, April 2025. ISSN 2299-0984. doi: 10.56553/popets-2025-0068. URL http://dx.doi.org/10.56553/popets-2025-0068.

Kumar Kshitij Patel, Bingqing Jiang, AFM Kabir, Weitong Zhang, Difan Zou, and Lingxiao Wang. Personalized federated training of diffusion models with privacy guarantees. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31790–31801, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Georg Pichler, Marco Romanelli, Leonardo Rey Vega, and Pablo Piantanida. Perfectly accurate membership inference by a dishonest central server in federated learning. *IEEE Transactions on Dependable and Secure Computing*, 21(4):4290–4296, 2023.

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.

W Nicholson Price and I Glenn Cohen. Privacy in the age of medical big data. *Nature medicine*, 25 (1):37–43, 2019.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4PJUBT9f2Ol.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Kulin Shah, Sitan Chen, and Adam Klivans. Learning mixtures of gaussians using the ddpm objective. *Advances in Neural Information Processing Systems*, 36:19636–19649, 2023.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017. doi: 10.1109/SP.2017.41. URL https://www.ieee-security.org/TC/SP2017/papers/313.pdf.

Amer Sinha, Thomas Mesnard, Ryan McKenna, Daogao Liu, Christopher A Choquette-Choo, Yangsibo Huang, Da Yu, George Kaissis, Zachary Charles, Ruibo Liu, et al. Vaultgemma: A differentially private gemma model. *arXiv preprint arXiv:2510.15001*, 2025.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Fiona Victoria Stanley Jothiraj and Afra Mashhadi. Phoenix: A federated generative diffusion model. In *Companion Proceedings of the ACM Web Conference 2024*, pp. 1568–1577, 2024.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

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

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026.

A SNR Upper Bound under Privatization

Here, we show that the SNR upper bound depends on the dimension of the data for the PFDM privatization mechanism (Patel et al., 2026).

Consider a d-dimensional pixel-space representation $x \in \mathbb{R}^d$, clipped to an ℓ_2-radius C_x:

$$\bar{x} = x \cdot \min\left(1, \frac{C_x}{\|x\|_2}\right), \qquad \|\bar{x}\|_2 \leq C_x. \tag{5}$$

The clipped representation is privatized with isotropic Gaussian noise:

$$\tilde{x} = \bar{x} + \sigma C_x \eta, \qquad \eta \sim \mathcal{N}(0, I_d), \tag{6}$$

where σ denotes the noise multiplier. We define the total energy signal-to-noise ratio as

$$\mathrm{SNR} = \frac{\|\bar{x}\|_2^2}{\mathbb{E}\|\sigma C_x \eta\|_2^2}. \tag{7}$$

Clipping bounds the total signal energy as

$$\|\bar{x}\|_2^2 \leq C_x^2, \tag{8}$$

while the expected Gaussian noise energy is

$$\mathbb{E}\|\sigma C_x \eta\|_2^2 = d\sigma^2 C_x^2. \tag{9}$$

Therefore,

$$\mathrm{SNR} \leq \frac{C_x^2}{d\sigma^2 C_x^2} = \frac{1}{d\sigma^2}. \tag{10}$$

Thus, for a fixed privacy noise multiplier σ, the maximum attainable total SNR decreases inversely with the representation dimension d. This yields a dimension-dependent privacy–utility bottleneck: clipping bounds the total signal energy independently of d, while the aggregate Gaussian noise energy grows linearly with d.

B Related Works

Perceptual Compression in Diffusion Modeling. (Rombach et al., 2022) introduced perceptual compression in their original LDM work to reduce the computational overhead of diffusion modeling. They mainly considered two types of regularization for training such compressors, or autoencoders (AEs): KL regularization and VQ regularization. KL-regularized AEs impose a weak KL penalty toward a standard normal distribution on the learned latent space, while VQ-regularized AEs use vector quantization and can be interpreted as VQGAN-style compressors. In both cases, the resulting latent has a shape of $C \times (H/f) \times (W/f)$, where f denotes the spatial downsampling factor. While this latent autoencoding paradigm remains widely used and was later adopted by diffusion transformers (DiTs) (Peebles & Xie, 2023), a separate line of recent work explores image tokenization, such as MAETok (Chen et al., 2025a) and SoftVQ-VAE (Chen et al., 2025b). These tokenizers use Vision Transformer (ViT)-based backbones to encode images into 1-D token sequences rather than the spatial latent grids produced by KL- and VQ-regularized AEs. Another line explores representation autoencoders (RAEs) (Zheng et al., 2026), which use pretrained encoders for stronger representations. However, their limited dimensionality reduction makes them unsuitable for this work. In this work, we use KL- and VQ-regularized AEs with a compression factor of $f = 4$, together with a SoftVQ-VAE tokenizer producing 64 latent tokens of dimension 32. For the remainder of this work, we denote these models as KL-$f4$, VQ-$f4$, and SoftVQ-VAE, respectively, and collectively refer to them as AEs.

Image Generation with Latent Diffusion Models. Rombach et al. (2022) introduced latent diffusion models (LDMs) as an efficient alternative to pixel-space diffusion modeling of DDPM (Ho et al., 2020). Their method largely follows the DDPM formulation but performs the diffusion process in a compressed latent space rather than directly in pixel space, while using the same UNet-based backbone (Ronneberger et al., 2015). Subsequent works have explored transformer-based architectures (Vaswani et al., 2017), such as the Diffusion Transformer (DiT) (Peebles & Xie, 2023), for improved scalability and performance, as well as flow-matching formulations for generative modeling

Algorithm 3 Training a Latent Diffusion Model (T-LDM)

input Latent dataset $Z = \{(z_0^i, b^i)\}_{i=1}^N$, where b^i contains the public class label and, for local models where applicable, the clipping flag; noise schedule $\{\beta_t\}_{t=1}^T$, initial denoiser parameters θ, learning rate λ_{lr}

1: Set $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$
2: **for each training iteration do**
3: Sample a mini-batch: $\{(z_0^i, b^i)\}_{i \in \mathcal{B}}$ from Z
4: For each $i \in \mathcal{B}$, independently draw $t_i \sim \text{Unif}([T])$ and $\eta_i \sim \mathcal{N}(0, I)$
5: Form noisy latents: $z_{t_i}^i = \sqrt{\bar{\alpha}_{t_i}}\, z_0^i + \sqrt{1 - \bar{\alpha}_{t_i}}\, \eta_i$
6: Compute the noise-prediction loss:

$$\mathcal{L}_{\text{LDM}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left\| \eta_i - \xi_\theta(z_{t_i}^i, t_i, b^i) \right\|_2^2$$

7: Update: $\theta \leftarrow \theta - \lambda_{\text{lr}} \nabla_\theta \mathcal{L}_{\text{LDM}}(\theta)$
8: **end for**

output Trained latent denoiser ξ_θ

Algorithm 4 Sampling from a Latent Diffusion Model (S-LDM; DDPM)

input Trained denoiser ξ_θ with the desired sampling conditions fixed, noise schedule $\{\beta_t\}_{t=1}^T$

1: Set $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, and $\sigma_t^2 = \beta_t(1 - \bar{\alpha}_{t-1})/(1 - \bar{\alpha}_t)$, with $\bar{\alpha}_0 = 1$
2: Sample initial latent: $\hat{z}_T \sim \mathcal{N}(0, I)$
3: **for** $t = T, T-1, \ldots, 1$ **do**
4: Predict noise: $\hat{\eta}_t = \xi_\theta(\hat{z}_t, t)$
5: Compute the reverse mean:

$$\mu_\theta(\hat{z}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\hat{z}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\, \hat{\eta}_t \right)$$

6: Draw $\eta_t \sim \mathcal{N}(0, I)$ independently if $t > 1$; otherwise set $\eta_t = 0$
7: Compute previous latent: $\hat{z}_{t-1} \leftarrow \mu_\theta(\hat{z}_t, t) + \sigma_t \eta_t$
8: **end for**

output Generated latent sample $\hat{z}_0$

(Lipman et al., 2023; Liu et al., 2023). In this work, we primarily use DDPM-based diffusion training with both UNet and DiT architectures. Our approach builds on the LDM framework (Rombach et al., 2022) and modern latent tokenization methods such as SoftVQ-VAE (Chen et al., 2025b), but differs in its core objective: we study privacy-preserving LDM training under heterogeneous federated settings.

Federated Learning in Diffusion Models. Existing approaches in FL in diffusion modeling largely follow FedAvg-style training, where locally trained model updates are periodically aggregated at a central server. Recent works further demonstrate the benefits of federated diffusion: Stanley Jothiraj & Mashhadi (2024) show that effective collaboration does not require client data to be individually IID, highlighting the value of cross-client diversity, while Allmendinger et al. (2026) show that collaboration can reduce the computational burden on individual clients. While these approaches collaborate through repeated model-update aggregation, our method requires only a single server-client communication during training and instead performs collaboration across diffusion timesteps. This eliminates the repeated communication rounds required by typical FedAvg-style training while simultaneously enabling privacy-preserving collaboration.

Differential Privacy in Diffusion Generative Models. Recent works have addressed memorization in diffusion models by providing formal differential privacy (DP) guarantees. Most rely on DP-SGD (Abadi et al., 2016), which clips per-sample gradients and injects Gaussian noise during optimization. Methods such as DPDM (Dockhorn et al., 2023), DP-LDM (Liu et al., 2024), RAPID (Jiang et al., 2025), and PrivImage (Li et al., 2024) adopt this mechanism while differing in which parameters or stages of training are optimized privately. However, gradient clipping and noise injection can substantially degrade optimization, particularly for large generative models. Moreover, because privacy loss accumulates across optimization steps, longer training generally incurs a larger privacy cost for a fixed noise level. This is especially restrictive for modern diffusion models, which often require extensive optimization. In contrast, our method privatizes the latent representations



(a) Non-collaborative LDM

(b) Collaborative (our) LDM

Figure 4: FFHQ samples generated at 256^2 resolution. (a) Non-collaborative LDM samples show weaker minority-class (female) features. (b) PF-LDM collaboration improves generation across both majority and minority classes.

Table 4: Sample counts for both clients and their combined privatized releases. Groups 1 and 2 correspond to male and female samples for CelebA-HQ and FFHQ, and ages < 60 and ≥ 60 for CheXpert, respectively. Majority and minority groups are reversed between clients.

	Client 1		Client 2		
Dataset	Group 1	Group 2	Group 1	Group 2	Global total
CelebA-HQ	5,000	50	50	5,000	10,100
FFHQ	15,000	150	150	15,000	30,300
CheXpert	70,000	700	700	70,000	141,400

Table 5: Input and latent dimensions of the evaluated autoencoders.

	VQ-$f4$	KL-$f4$	SoftVQ-VAE
Input size	256×256	256×256	256×256
Latent size	64×64	64×64	64 tokens
Downsampling Factor	4	4	–
Latent shape	$64 \times 64 \times 3$	$64 \times 64 \times 3$	64×32

communicated to the server, making its privacy guarantee independent of the number of optimization steps used to train the diffusion models.

C Additional Preliminaries for Section 2

Denoising Latent Diffusion Models (LDMs). Image generation with LDMs (Rombach et al., 2022) follows a two-stage procedure. First, an image compressor maps images from pixel space to a lower-dimensional latent space. Given an image dataset $X = \{(x_0^i, b^i)\}_{i=1}^N$, an autoencoder with encoder Enc and decoder Dec maps each image to $z_0^i = \mathrm{Enc}(x_0^i)$, forming the latent dataset $Z = \{(z_0^i, b^i)\}_{i=1}^N$, where b^i denotes the class condition. In the second stage, a denoiser learns the latent distribution rather than directly modeling images in pixel space. This involves a forward diffusion process and a reverse diffusion process. In the forward process, a clean latent z_0 is perturbed as

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1 - \bar{\alpha}_t}\, \eta, \qquad \eta \sim \mathcal{N}(0, I), \tag{11}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$ is determined by the noise schedule $\{\beta_t\}_{t=1}^T$. During training, a mini-batch is sampled from Z, together with timesteps $t_i \sim \mathrm{Unif}([T])$ and Gaussian noise $\eta_i \sim \mathcal{N}(0, I)$ (line 3 in Algorithm 3). Following Ho et al. (2020); Rombach et al. (2022), the class-conditional

Table 6: Training cost and FID score compared across different settings. FID is calculated for $30{,}000$ samples on FFHQ with the NP baseline.

Settings	GPU-hours (100 epochs)	Time to converge	FID score ↓
DiT + KL-$f4$	25.25	144–156 h	11.3065
DiT + VQ-$f4$	27.18	144–156 h	13.9988
U-Net + KL-$f4$	15.62	47 h	9.1629
U-Net + VQ-$f4$	15.62	47 h	8.7512
DiT + SoftVQ-VAE	3.04	46.5 h	14.6785

denoiser ξ_θ is trained to predict the injected noise with the objective

$$L_{\mathrm{LDM}}(\theta) = \mathbb{E}_{(z_0,b)\sim p(z,b),\, t\sim\mathrm{Unif}([T]),\, \eta\sim\mathcal{N}(0,I)} \left[\|\eta - \xi_\theta(z_t, t, b)\|_2^2 \right], \tag{12}$$

as detailed in lines 4–6 of Algorithm 3.

To generate images, sampling follows the reverse diffusion process in latent space conditioned on a class label b. Starting from Gaussian noise $\hat{z}_T \sim \mathcal{N}(0, I)$ (line 2 in Algorithm 4), the denoiser ξ_θ iteratively predicts the noise component as $\hat{\eta}_t = \xi_\theta(\hat{z}_t, t, b)$. Following the DDPM noise parameterization (Ho et al., 2020), this prediction is used to parameterize the reverse mean $\mu_\theta(\hat{z}_t, t, b)$ rather than directly predicting $\hat{z}_{t-1}$ (lines 4–5). The previous latent is then sampled by adding Gaussian noise at intermediate timesteps and setting the noise to zero at the final step (lines 6–7). After all reverse steps, the generated latent $\hat{z}_0$ is decoded back to image space as $\hat{x} = \mathrm{Dec}(\hat{z}_0)$.

D Missing Experimental Details from Section 5

D.1 Data Preparation.

We consider a two-client distributed training setting where each have underrepresented class distribution following (Kairouz & McMahan, 2021) setting. For each of the clients, we construct the majority of samples from one group and a small minority from the other, while the reverse is done for the other client. Each client privatizes their data using the method described in Section 3 before sending it to the server. The server combines data from each client before training. Table 4 shows the exact number of samples we use for each of the clients and the global model training.

D.2 Modeling & Training Details.

Choice of Autoencoders. Table 5 presents the AE-specific details. The $z-$shape is the dimension the denoisers observe during training. As can be seen, the SoftVQ-VAE has the lowest dimension, indicating the most compressed latent among all three.

Choice of Denoisers. We use U-Net backbones for KL-$f4$ and VQ-$f4$, which produce spatial $64 \times 64 \times 3$ latents, and DiT for SoftVQ-VAE, which produces token sequences. Convolutional U-Nets are naturally suited to spatial latent grids, whereas tokenized representations are better matched to transformer-based denoisers. DiTs, however, typically require more data and optimization to reach strong performance. To assess this trade-off, we compare training efficiency and convergence on FFHQ under the non-private (NP) baseline in Table 6. At our dataset scale, U-Net with KL-$f4$ or VQ-$f4$ achieves better FID and converges faster than DiT with the same AEs. In contrast, DiT with SoftVQ-VAE trains substantially faster due to its much smaller latent representation. Based on these observations, we adopt U-Net for KL-$f4$/VQ-$f4$ and DiT for SoftVQ-VAE.

Selection of Privacy Parameters. For our privacy-preserved communication between the clients and the server, we use $C_z = 35$ and $t_0 = 850$ for all KL-$f4$ and VQ-$f4$ latents following Algorithm 1. For SoftVQ-VAE, we use the parameters $C_z = 48$ and $t_0 = 886$. Both of these settings result in the same privacy bound of $\epsilon = 10, \delta = 10^{-5}$. However, because of the highly compressed latent behavior of SoftVQ-VAE, the latent norms are very small, and all SoftVQ-VAE latents for all datasets are less than 48. Hence, according to Algorithm 1, our training settings result in no clipping at all for SoftVQ-VAE latents without compromising privacy protection.

Table 7: Training and privacy configurations for CelebA-HQ.

Setting	**U-Net + KL-$f4$**	**U-Net + VQ-$f4$**	**DiT-SL + SoftVQ-VAE**
Beta schedule	Linear	Linear	Linear
Beta start/end	$10^{-4} \rightarrow 0.02015$	$10^{-4} \rightarrow 0.02015$	$10^{-4} \rightarrow 0.02015$
C_z	35	35	48
t_0	850	850	886
ϵ	9.98246	9.98246	9.95618
δ	10^{-5}	10^{-5}	10^{-5}
Effective learning rate	6.4×10^{-5}	6.4×10^{-5}	10^{-4}
LR schedule	Constant	Constant	Constant
Global batch size	32	32	512
Latent normalization	Scale 0.150881	Scale 1.0	$(z - 0.0009714666)/0.07259091$
Denoiser channel multipliers	[1,2,3,4]	[1,2,3,4]	N/A
Residual blocks	2 per resolution	2 per resolution	16 transformer blocks
Attention resolution	[8,4,2]	[8,4,2]	All tokens
Conditioning	Cross-attention	Cross-attention	AdaLN
Conditioning dimension	512	512	1024
Parameters (M)	414.715	414.715	305.816

Table 8: Training and privacy configurations for FFHQ.

Setting	**U-Net + KL-$f4$**	**U-Net + VQ-$f4$**	**DiT-SL + SoftVQ-VAE**
Beta schedule	Linear	Linear	Linear
Beta start/end	$10^{-4} \rightarrow 0.02015$	$10^{-4} \rightarrow 0.02015$	$10^{-4} \rightarrow 0.02015$
C_z	35	35	48
t_0	850	850	886
ϵ	9.98246	9.98246	9.95618
δ	10^{-5}	10^{-5}	10^{-5}
Effective learning rate	8.4×10^{-5}	8.4×10^{-5}	10^{-4}
LR schedule	Constant	Constant	Constant
Global batch size	42	42	512
Latent normalization	Scale 0.16072381	Scale 1.0	$(z - 0.0009714666)/0.07259091$
Denoiser channel multipliers	[1,2,3,4]	[1,2,3,4]	N/A
Residual blocks	2 per resolution	2 per resolution	16 transformer blocks
Attention resolution	[8,4,2]	[8,4,2]	All tokens
Conditioning	Cross-attention	Cross-attention	AdaLN
Conditioning dimension	512	512	1024
Parameters (M)	414.715	414.715	305.816

Training Hyperparameters. In this section, we provide detailed training hyperparameters and denoiser model configurations in Tables 7, 8, and 9.

Details on Downstream Task Evaluation. For downstream evaluation, we use $10{,}000$ generated images per method, with $5{,}000$ samples from each group. We consider CNN and ResNet architectures with varying capacities and learning rates of 3×10^{-4}, 1×10^{-3}, and 3×10^{-3}, and select the best configuration for each baseline. Each configuration is run with three random seeds, and results are reported as the mean. Accuracy and F1-score are evaluated on a held-out test set of $10{,}000$ samples for each of the FFHQ and CelebA-HQ datasets.

D.3 Sample Generation

The generation process follows Algorithm 2 in Section 3. The sampling configurations used for evaluation are summarized in Table 10. Each model is trained for a fixed number of epochs, and the checkpoint achieving the best FID is selected for reporting. Qualitative samples for CelebA-HQ and FFHQ are shown in Fig. 2 and Fig. 4, respectively.

Table 9: Training and privacy configurations for CheXpert.

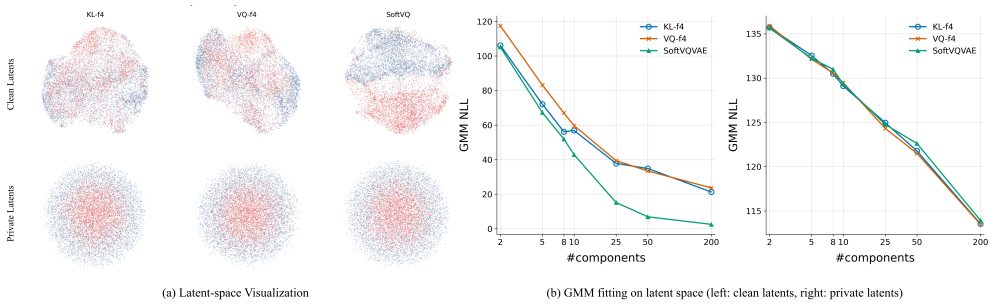
Setting	**U-Net + KL**	**U-Net + VQ**	**DiT-L + SoftVQ**
Beta schedule	Linear	Linear	Linear
Beta start/end	$10^{-4} \rightarrow 0.02015$	$10^{-4} \rightarrow 0.02015$	$10^{-4} \rightarrow 0.02015$
C_z	35	35	48
t_0	850	850	886
ϵ	9.98246	9.98246	9.95618
δ	10^{-5}	10^{-5}	10^{-5}
Effective learning rate	6.4×10^{-5}	6.4×10^{-5}	10^{-4}
LR schedule	Constant	Constant	Constant
Global batch size	32	32	512
Latent normalization	Scale 0.1468593103	Scale 1.0	$(z - 0.0009714666)/0.07259091$
Denoiser channel multipliers	[1,2,3,4]	[1,2,3,4]	N/A
Residual blocks	2 per resolution	2 per resolution	24 transformer blocks
Attention resolution	[8,4,2]	[8,4,2]	All tokens
Conditioning	Cross-attention	Cross-attention	AdaLN
Conditioning dimension	512	512	1024
Parameters (M)	539.774	539.774	456.932

Table 10: Sampling configurations used for the results in Table 1. Here, e denotes the training epoch, while e_g and e_ℓ denote the global and local model epochs, respectively.

Architecture	**Baseline**	**CelebA-HQ**	**FFHQ**	**CheXpert**
U-Net + KL-$f4$	NP	$e = 300$ DDIM-150	$e = 200$ DDIM-200	$e = 300$ DDIM-200
	NC	$e = 300$ DDIM-150	$e = 300$ DDIM-200	$e = 400$ DDIM-200
	PF-LDM	$e_g/e_\ell = 500/200$ DDIM-50 $\rightarrow$ DDPM-850	$e_g/e_\ell = 400/200$ DDIM-50 $\rightarrow$ DDPM-850	$e_g/e_\ell = 100/250$ DDIM-10 $\rightarrow$ DDPM-850
U-Net + VQ-$f4$	NP	$e = 200$ DDIM-150	$e = 300$ DDIM-200	$e = 300$ DDPM-1000
	NC	$e = 300$ DDIM-150	$e = 400$ DDIM-200	$e = 200$ DDPM-1000
	PF-LDM	$e_g/e_\ell = 200/100$ DDIM-75 $\rightarrow$ DDPM-850	$e_g/e_\ell = 200/200$ DDIM-50 $\rightarrow$ DDPM-850	$e_g/e_\ell = 400/150$ DDIM-50 $\rightarrow$ DDPM-850
DiT + SoftVQ-VAE	NP	$e = 700$ DDIM-400	$e = 1500$ DDIM-200	$e = 500$ DDIM-200
	NC	$e = 800$ DDIM-400	$e = 1500$ DDIM-200	$e = 300$ DDIM-200
	PF-LDM	$e_g/e_\ell = 1400/800$ DDIM-50 $\rightarrow$ DDPM-886	$e_g/e_\ell = 1500/1500$ DDIM-50 $\rightarrow$ DDPM-886	$e_g/e_\ell = 600/300$ DDIM-50 $\rightarrow$ DDIM-200

D.4 Latent Space Analysis.

To generate UMAP visualizations, we sample $10{,}000$ random images, $5{,}000$ from each of the groups (male and female) from the CelebA-HQ and FFHQ datasets. The samples were encoded using each AE and flattened before projecting them using a seeded Euclidean UMAP with 30 neighbors. The same $10{,}000$ samples were used for the GMM fitting as well. Particularly, the flattened latents were first reduced in dimension to avoid the curse of dimensionality using 100-dimensional UMAPs. Later, the latent data were normalized and fitted using a GMM to return the negative log-likelihood



(a) Latent-space Visualization

(b) GMM fitting on latent space (left: clean latents, right: private latents)

Figure 5: Latent-space analysis for clean and private CelebA-HQ latents. (a) Visualization through UMAP. Top and bottom rows indicate clean and private latents, both clustered with male and female class labels as red and blue. (b) GMM fitting on CelebA-HQ latents on different latent spaces.

losses (NLL). Each GMM was trained for 500 epochs on its respective latent data with a batch size of 256 on a single A100 80GB GPU. Fig. 5 shows the resulting visualizations for CelebA-HQ, whereas the FFHQ visualizations are presented in Fig 3.

E More Details on Privacy Attacks

We expand the FFHQ privacy evaluation in Section 5.2. Membership attacks ask whether an image trained a denoiser; reconstruction and source-identification attacks ask what one privatized latent upload reveals about its source.

E.1 Attack goals and information

Each attack predicts a hidden quantity from the information specified below. The evaluator uses the hidden answer only to score the prediction.

Membership inference. The attacker has white-box access to the frozen non-private centralized (NP) denoiser or PF-LDM shared denoiser, including its parameters, gradients, and noise schedule. It knows the public encoder and gender label and receives a separate calibration pool with known membership labels. Given a candidate image x and gender c, it predicts whether x trained the target. For PF-LDM, it can simulate fresh uploads but cannot observe the original training uploads. Client refinement models remain outside this evaluation. We train no shadow models (surrogate denoisers with known training membership) (Shokri et al., 2017) and do not use LiRA (Carlini et al., 2022).

Reconstruction. The attacker knows the public encoder, decoder, release rule, and gender. It learns from disjoint auxiliary image–upload pairs, then predicts an unknown test image from one upload. It sees neither the test image nor its clean latent and uses no denoiser or membership labels.

Candidate identification. The attacker knows the public release rule and receives one upload and an unordered set of exact, same-gender candidate latents containing its source. With equal candidate priors, it predicts which latent produced the upload; gallery order and the evaluator's distractor-selection rule are unavailable. For KL, an exact latent includes the encoder posterior draw, which provides more information than the image alone. This task measures source association, not training membership.

E.2 Data, targets, and candidate inputs

Data. We use the two-client FFHQ training split in Appendix D.1 and Table 4. NP and PF-LDM train on the same 30,300 images, using clean and privatized latents, respectively. We select 10,000 gender-balanced nonmembers from the same filtered FFHQ pool; none trains either target. We divide members and nonmembers into 20% calibration and 80% test pools, stratified by client and

gender (Table 11). All member images train the target denoisers, including those assigned to the attack test pool.

Table 11: Membership-candidate split. Calibration and test pools are disjoint; all member images train the target denoiser.

Source	Metadata gender	Total	Attack calibration	Attack test
Client 1: members	Male	15,000	3,000	12,000
Client 1: members	Female	150	30	120
Client 2: members	Male	150	30	120
Client 2: members	Female	15,000	3,000	12,000
Nonmembers	Male	5,000	1,000	4,000
Nonmembers	Female	5,000	1,000	4,000
All members		30,300	6,060	24,240
All nonmembers		10,000	2,000	8,000

A disjoint reconstruction pool contains 5,323 images: 4,258 auxiliary training images and 1,065 validation images, including 206 and 52 male images, respectively. Thus this auxiliary pool is predominantly female.

Targets and encoders. We attack the U-Net/KL-f4, U-Net/VQ-f4, and DiT/SoftVQ-VAE denoisers used in the FFHQ generation experiments (Appendix D.3 and Table 10), abbreviated KL, VQ, and SoftVQ below. We hold their exponential-moving-average weights fixed. NP and PF-LDM differ in diffusion schedules and, for U-Nets, training duration, so the comparison does not isolate the effect of privatization.

We use the encoders and normalization in Tables 5 and 8. The attacker knows the latent centering and scaling parameters. We map 256×256 RGB images to $[-1, 1]$ without data augmentation. We write $z = E(x)$ in denoiser coordinates; E includes encoder normalization, and D reverses it before decoding. KL samples its encoder posterior. VQ uses continuous pre-quantization latents with quantization in the decoder; SoftVQ uses deterministic soft-quantized latents.

Latent uploads. For the release in Algorithm 1, write $C = C_z$ and $a_{\mathrm{rel}} = \bar{\alpha}_{t_0}$. The client clips z and adds Gaussian noise:

$$\mathrm{clip}_C(z) = z \min\{1, C/\|z\|_2\}, \quad \mathrm{clip}_C(0) = 0, \tag{13}$$

$$Y = \mu(z) + \sigma\eta, \qquad \mu(z) = \sqrt{a_{\mathrm{rel}}}\,\mathrm{clip}_C(z), \quad \sigma^2 = 1 - a_{\mathrm{rel}}, \quad \eta \sim \mathcal{N}(0, I_d). \tag{14}$$

We use the $(\epsilon, \delta) = (10, 10^{-5})$ settings in Table 8, under replace-one adjacency as in Theorem 4.1. Here d counts latent entries and σ^2 is the variance in upload coordinates; rescaling by $1/\sqrt{a_{\mathrm{rel}}}$ gives the theorem's variance $(1 - a_{\mathrm{rel}})/a_{\mathrm{rel}}$.

Schedules and attack inputs. Let f_θ denote the noise predictor ξ_θ in Appendix C, conditioned on gender c. We index timesteps from 0 to 999 and set $a_t = \prod_{k=0}^{t}(1 - \beta_k) = \bar{\alpha}_{t+1}$, so $a_0 = 1 - \beta_0$. For NP U-Nets, $\sqrt{\beta_t}$ increases linearly from $\sqrt{0.0015}$ to $\sqrt{0.0195}$; NP DiT uses a linear β_t schedule from 10^{-4} to 0.02. PF-LDM uses Table 8. The release step t_0 corresponds to zero-based index $t_0 - 1$ and is distinct from the attack timestep.

Membership attacks use $u = z$ for NP and a fresh simulated $u = Y$ for PF-LDM, identically for members and nonmembers. We average scores or feature vectors over four candidate inputs: four fixed posterior draws per KL image and independent release noise for PF-LDM. Deterministic NP encoders yield the same latent each time. Thus each denoiser receives candidates in its training input domain.

E.3 Membership scores

We probe $\mathcal{T} = \{50, 100, 200, 400, 700\}$. Smaller scalar scores indicate stronger evidence of membership. Fixed scores require no membership labels; selecting timesteps or fitting classifiers uses the labeled calibration pool.

Denoising loss. For $\xi \sim \mathcal{N}(0, I_d)$ and $u_t = \sqrt{a_t}u + \sqrt{1-a_t}\xi$, define

$$L_t(u;\xi) = d^{-1}\|f_\theta(u_t, t, c) - \xi\|_2^2. \tag{15}$$

We average two noise draws per input and timestep. Besides fixed and calibration-selected timesteps, we evaluate the mean across $\mathcal{T}$.

PIA and PIA-N. Proximal initialization (Kong et al., 2024) replaces random noise with the model's initial prediction. With $e = f_\theta(u, 0, c)$ and $v_t(e) = \sqrt{a_t}u + \sqrt{1-a_t}e$, our PIA score is

$$P_t(u) = \sum_{j=1}^{d}[e_j - f_\theta(v_t(e), t, c)_j]^4. \tag{16}$$

It measures consistency between predictions at two timesteps. PIA-N replaces both occurrences of e with $\widetilde{e} = \sqrt{2/\pi}\, e/(d^{-1}\sum_j |e_j|)$, matching standard Gaussian noise's mean absolute magnitude.

SecMI. SecMI (Duan et al., 2023) measures the error of a deterministic DDIM forward-and-return step (Song et al., 2021). Define

$$\widehat{u}_0(v, t) = [v - \sqrt{1-a_t}f_\theta(v, t, c)]/\sqrt{a_t}, \tag{17}$$

$$D_{t\to s}(v) = \sqrt{a_s}\widehat{u}_0(v, t) + \sqrt{1-a_s}f_\theta(v, t, c). \tag{18}$$

Starting from $v_0 = u$, we invert along $0, 10, \ldots, 700$ and score

$$S_t(u) = \|D_{t+10\to t}(D_{t\to t+10}(v_t)) - v_t\|_2^2, \qquad t \in \mathcal{T}. \tag{19}$$

The maps do not clip predicted latents; the return step measures error without changing the inversion path.

GSA gradient features. GSA (Pang et al., 2025) uses the gradients that a candidate induces in each parameter tensor θ_j. Let $g_j(t) = \nabla_{\theta_j} L_t(u;\xi_t)$, with one noise draw per timestep. For $p \in \{1, 2\}$, we evaluate

$$h_j^{(1,p)} = \left\| \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} g_j(t) \right\|_2^p, \qquad h_j^{(2,p)} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \|g_j(t)\|_2^p. \tag{20}$$

GSA1 averages gradients before taking their norms; GSA2 averages the norms. The four variants are GSA1/GSA2 with L_2 or L_2^2 features. We average feature vectors over candidate inputs after taking these norms, then fit a membership classifier. The target parameters never change.

E.4 Calibration and evaluation

Fitting and selecting attacks. Of the 6,060 calibration members and 2,000 nonmembers, we use 4,545/1,500 to fit attackers and tune hyperparameters by three-fold stratified cross-validation (CV), reserving 1,515/500 for threshold choice. Ten learned families compete: logistic regression on the five denoising losses or all 15 PIA/PIA-N/loss scores, and logistic regression or XGBoost on each of the four GSA variants. We weight the two membership classes equally, take logs of scalar scores, and apply $\log(1+v)$ to GSA features. Logistic models standardize features within each CV fold and tune L2 regularization over $C_{\mathrm{LR}} \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. Boosted trees use 300 trees, learning rate 0.05, and CV-selected depth in $\{2, 4, 6\}$. We negate classifier logits or margins so smaller scores favor membership.

For each target, we select the family with the highest mean CV AUROC over three attacker-fitting seeds, then report its mean test performance over those fits. All metrics therefore describe one family chosen without test labels. For the scalar results in Table 13, we select each timestep by AUROC on the full calibration pool. Every test image is disjoint from calibration.

Metrics and thresholds. A score s predicts membership when $s \leq \tau$. TPR is the fraction of members accepted; FPR is the fraction of nonmembers accepted. Sweeping τ traces the ROC of a fixed score. AUROC is the fraction of member/nonmember pairs ranked correctly, with half credit for ties; chance is 0.5. We report

$$\widehat{\mathrm{TPR}}_{\alpha} = \max_{\tau:\, \widehat{\mathrm{FPR}}(\tau) \leq \alpha} \widehat{\mathrm{TPR}}(\tau), \qquad \alpha \in \{0.01, 0.001\}, \tag{21}$$

with tied scores entering together and no interpolation. These are descriptive points on the labeled test ROC. The caps allow 80 and eight false positives among 8,000 test nonmembers, respectively.

Separately, we freeze the largest observed threshold on the reserved calibration pool that meets each FPR cap, using $-\infty$ if none qualifies, and report both test TPR and test FPR. The 500 reserved nonmembers permit zero calibration false positives at the 0.1% cap; this does not guarantee the same population or test FPR. Test labels do not select these frozen thresholds, and we never use them to reverse score orientation.

Uncertainty and subgroups. We form 95% percentile-bootstrap intervals from 1,000 test-image resamples, stratified by membership, client, and gender and shared across fitting or noise repeats. These pointwise intervals condition on the targets, split, selected families, and fitted attackers; perfect separation can yield degenerate intervals. Subgroup analyses retain the pooled classifier and thresholds, comparing each client's members with same-gender nonmembers. Each subgroup has 120 or 12,000 members and 4,000 nonmembers; other clients' members never count as nonmembers. Robustness checks reuse the same test images and do not constitute an independent benchmark or correct for adaptive experiment selection.

E.5 Membership results and robustness checks

Table 12 identifies the selected families and extends the main comparison with uncertainty intervals. Calibration selects gradient attacks for NP U-Nets, while the 15-score classifier separates NP DiT's test members perfectly. The selected PF-LDM attacks remain near chance. Scalar attacks generally trail the gradient attacks on NP U-Nets but also nearly separate NP DiT members (Table 13).

Table 12: Calibration-selected attacks: mean over three fitting seeds and conditional 95% bootstrap intervals. LR denotes logistic regression; XGB denotes boosted trees. PF denotes the shared PF-LDM denoiser.

Target	Selected feature / learner	AUROC	$\mathrm{TPR}_{1\%}$ (%)	$\mathrm{TPR}_{0.1\%}$ (%)
KL / NP	GSA1-L_2 / LR	0.9929 [0.9919, 0.9938]	82.84 [80.07, 86.45]	30.62 [23.29, 51.02]
KL / PF	GSA2-L_2^2 / LR	0.5012 [0.4941, 0.5083]	0.83 [0.67, 1.04]	0.17 [0.10, 0.27]
VQ / NP	GSA1-L_2 / LR	0.9997 [0.9995, 0.9999]	99.98 [99.96, 99.99]	91.80 [77.06, 99.25]
VQ / PF	GSA1-L_2^2 / XGB	0.5003 [0.4926, 0.5065]	1.12 [0.89, 1.31]	0.10 [0.07, 0.17]
SoftVQ / NP	15-score / LR	1.0000 [1.0000, 1.0000]	100.00 [100.00, 100.00]	100.00 [100.00, 100.00]
SoftVQ / PF	GSA1-L_2^2 / XGB	0.4930 [0.4859, 0.4994]	0.90 [0.71, 1.17]	0.07 [0.03, 0.17]

Subgroups and fixed thresholds. The pooled classifiers show similar patterns across clients and genders (Table 14), with much smaller minority samples. Calibration-fixed thresholds can realize different test FPRs (Table 15): KL/NP reaches 0.958% test FPR at the nominal 0.1% calibration cap. Its corresponding TPR therefore does not measure detection at 0.1% test FPR.

Fitting variability and calibration size. Table 16 reports variation across three attacker fits and refits using 25%, 50%, or 100% of each membership–gender calibration stratum, preserving the 75%/25% fitting/threshold split. Reduced-budget runs use one common fitting seed and retain the family selected at full budget; they measure refitting sensitivity, not attack selection with fewer known membership labels. The denoisers remain fixed.

Table 13: Scalar attacks: test AUROC and timestep selected by AUROC on the full calibration pool. These scores are separate from the ten learned candidates.

Target	PIA	PIA-N	Denoising loss	SecMI
KL / NP	0.6198 $t = 100$	0.5243 $t = 200$	0.5807 $t = 200$	0.5892 $t = 200$
KL / PF	0.5042 $t = 50$	0.4949 $t = 100$	0.5053 $t = 200$	0.4996 $t = 100$
VQ / NP	0.8059 $t = 200$	0.5312 $t = 50$	0.7136 $t = 100$	0.7630 $t = 50$
VQ / PF	0.5019 $t = 50$	0.4969 $t = 200$	0.5053 $t = 200$	0.4972 $t = 200$
SoftVQ / NP	1.0000 $t = 100$	0.9419 $t = 100$	1.0000 $t = 50$	0.9999 $t = 50$
SoftVQ / PF	0.5010 $t = 100$	0.4975 $t = 100$	0.5050 $t = 700$	0.4977 $t = 50$

Table 14: Client–gender AUROC using the pooled selected attackers, averaged over three fits. Client 1 is mostly male; client 2 is mostly female. Each minority subgroup has only 120 test members.

Target	Client 1 / male	Client 1 / female	Client 2 / male	Client 2 / female
KL / NP	0.9932	0.9920	0.9956	0.9925
KL / PF	0.4999	0.4947	0.5089	0.5016
VQ / NP	0.9995	0.9999	0.9994	0.9999
VQ / PF	0.5041	0.5046	0.5190	0.4962
SoftVQ / NP	1.0000	1.0000	1.0000	1.0000
SoftVQ / PF	0.4904	0.5065	0.4921	0.4956

Table 15: Thresholds fixed using calibration data: mean test TPR and FPR over three fits. All entries are percentages; the realized test FPR can exceed the calibration cap.

	Calibration FPR cap: 1%		Calibration FPR cap: 0.1%	
Target	Test TPR (%)	Test FPR (%)	Test TPR (%)	Test FPR (%)
KL / NP	89.827	1.629	81.680	0.958
KL / PF	1.419	1.579	0.220	0.163
VQ / NP	99.982	1.413	99.824	0.446
VQ / PF	0.726	0.700	0.099	0.075
SoftVQ / NP	100.000	0.900	100.000	0.062
SoftVQ / PF	1.081	1.083	0.160	0.204

Table 16: Fitting variability and reduced calibration budgets. Standard deviations use three fits (pp: percentage points). Budget comparisons use one common fitting seed and retain the family selected at full budget.

	Fit-seed standard deviation			Calibration fraction: AUROC		
Target	AUROC	$\text{TPR}_{1\%}$ (pp)	$\text{TPR}_{0.1\%}$ (pp)	25%	50%	100%
KL / NP	0.00008	0.252	3.508	0.9888	0.9908	0.9928
KL / PF	0.00081	0.079	0.024	0.4999	0.4967	0.5017
VQ / NP	0.00002	0.005	1.079	0.9994	0.9996	0.9997
VQ / PF	0.00112	0.010	0.032	0.4941	0.4969	0.5009
SoftVQ / NP	0.00000	0.000	0.000	1.0000	1.0000	1.0000
SoftVQ / PF	0.00183	0.072	0.036	0.5019	0.5019	0.4949

Fresh noise and additional queries. We recompute PIA, PIA-N, loss, and GSA features for three independent release/attack-noise repeats, holding KL posterior draws, images, denoisers, and the selected family fixed. Each repeat refits the attacker with a common fitting seed. Table 17(a) shows stable AUROCs but greater variation in KL/NP's low-FPR TPR. We also average each image's features across the three repeats and fit one classifier, retuning its hyperparameter by calibration CV. This triples the query budget and improves NP detection (Table 17(b)). PF-LDM AUROCs

remain near chance, although near-chance AUROC does not imply zero advantage at every FPR. All additional PF-LDM inputs are fresh simulated releases of the candidate, not historical training uploads.

Table 17: Noise sensitivity and query budget. (a) Mean ± standard deviation over three independent noise repeats, each using four candidate inputs per image and the same fitting seed. (b) One attacker fit to features averaged across all three repeats, using three times the query budget. Both retain the calibration-selected family. Brackets give conditional 95% bootstrap intervals for the mean in (a) and the single fit in (b). TPRs are percentages; their standard deviations are percentage points.

(a) Independent noise repeats

Target	AUROC	$\text{TPR}_{1\%}$ (%)	$\text{TPR}_{0.1\%}$ (%)
KL / NP	0.99262 ± 0.00004 [0.99169, 0.99357]	82.92 ± 1.28 [79.46, 85.99]	31.411 ± 7.040 [15.199, 45.011]
KL / PF	0.49831 ± 0.00632 [0.49403, 0.50251]	1.02 ± 0.20 [0.90, 1.21]	0.138 ± 0.038 [0.084, 0.186]
VQ / NP	0.99970 ± 0.00002 [0.99946, 0.99987]	99.97 ± 0.02 [99.94, 99.98]	93.519 ± 0.259 [85.607, 97.802]
VQ / PF	0.50107 ± 0.00375 [0.49689, 0.50533]	1.05 ± 0.11 [0.90, 1.16]	0.120 ± 0.037 [0.070, 0.175]
SoftVQ / NP	1.00000 ± 0.00000 [1.00000, 1.00000]	100.00 ± 0.00 [100.00, 100.00]	100.000 ± 0.000 [100.000, 100.000]
SoftVQ / PF	0.50056 ± 0.00196 [0.49624, 0.50500]	1.00 ± 0.03 [0.85, 1.17]	0.092 ± 0.016 [0.060, 0.172]

(b) Features averaged across repeats

Target	AUROC	$\text{TPR}_{1\%}$ (%)	$\text{TPR}_{0.1\%}$ (%)
KL / NP	0.9946 [0.9937, 0.9955]	87.26 [83.31, 90.01]	42.859 [18.432, 61.511]
KL / PF	0.5043 [0.4970, 0.5114]	1.08 [0.76, 1.33]	0.190 [0.062, 0.293]
VQ / NP	0.9997 [0.9995, 0.9999]	99.99 [99.98, 100.00]	93.981 [82.681, 99.249]
VQ / PF	0.4990 [0.4918, 0.5060]	1.02 [0.80, 1.32]	0.186 [0.107, 0.289]
SoftVQ / NP	1.0000 [1.0000, 1.0000]	100.00 [100.00, 100.00]	100.000 [100.000, 100.000]
SoftVQ / PF	0.4982 [0.4907, 0.5048]	0.78 [0.60, 1.00]	0.120 [0.074, 0.182]

E.6 Reconstructing an image from one upload

Attacker and training. Using the auxiliary data in Appendix E.2, we learn

$$\widehat{z} = m_c + sF_\psi(Y/r, c), \qquad \widehat{x} = D(\widehat{z}), \tag{22}$$

where m_c is the auxiliary gender-specific latent mean, s is the a single RMS scale computed over all auxiliary-training latent residual entries, and r scales the input by its expected root-mean-square magnitude. All statistics use only auxiliary training data. We use a residual convolutional network for spatial KL/VQ latents and a residual token/channel mixing network for SoftVQ, both of width 64.

We train for up to 40 epochs with AdamW (learning rate 10^{-3}, weight decay 10^{-4}, batch size 16), minimizing normalized latent MSE with equal total weight for each gender. Training uses fresh release noise and, for KL, cycles through the four posterior draws. We select the epoch with the lowest gender-balanced validation latent MSE, using fixed validation noise and posterior draws. The test images do not guide training.

Controls and metrics. We train separate reconstructors for

$$Y_q = \sqrt{a_{\text{rel}}}\,\text{clip}_C(z) + q\sqrt{1 - a_{\text{rel}}}\eta, \qquad q \in \{1, 1/4, 0\}, \tag{23}$$

and for clean latents $Y_{\text{clean}} = z$. Only $q = 1$ uses the private release settings; $q = 1/4$ quarters the noise standard deviation. All conditions share architecture, training budget, and 2,048 test im-

ages (1,024 per gender), with paired latent/noise draws. These controls change the observed latent, without querying or changing a denoiser.

The gender prior predicts $D(m_c)$ without observing an upload. We clip decoded RGB outputs to $[0, 1]$ and report pixel MSE, mean per-image PSNR $= -10 \log_{10}(\mathrm{MSE})$, SSIM (Wang et al., 2004), and AlexNet LPIPS (Zhang et al., 2018). Lower MSE/LPIPS and higher PSNR/SSIM indicate closer reconstructions.

Table 18: Reconstruction on 2,048 paired test images. Each estimator trains on its stated observation condition; the prior uses only gender. These attacks query no denoiser.

Encoder	Attacker input / estimator	MSE ↓	PSNR ↑	SSIM ↑	LPIPS ↓
KL	Gender latent-mean prior	0.06821	12.100	0.4278	0.7713
	Actual upload	0.06518	12.228	0.4293	0.7780
	Quarter noise std.	0.05291	13.022	0.4444	0.7514
	No noise; clipped	0.00077	31.696	0.9070	0.0174
	Clean latent	0.00071	32.029	0.9091	0.0168
VQ	Gender latent-mean prior	0.06522	12.198	0.4333	0.8284
	Actual upload	0.06475	12.212	0.4337	0.8298
	Quarter noise std.	0.05866	12.612	0.4423	0.8146
	No noise; clipped	0.00079	31.536	0.9055	0.0200
	Clean latent	0.00073	31.892	0.9081	0.0183
SoftVQ	Gender latent-mean prior	0.06682	12.135	0.3603	0.5144
	Actual upload	0.06645	12.147	0.3590	0.5136
	Quarter noise std.	0.06543	12.204	0.3620	0.5125
	No noise; clipped	0.00293	25.812	0.7367	0.0581
	Clean latent	0.00293	25.823	0.7367	0.0581

At the actual release noise, reconstruction gains little over the gender prior. Removing noise while retaining clipping raises PSNR by 19.47, 19.32, and 13.67 dB for KL, VQ, and SoftVQ, respectively. The controls show that the reconstructor can recover detail from more informative latents; they do not establish that it is the strongest possible reconstructor.

E.7 Identifying the source of a latent upload

For an unordered gallery of exact latents $\{z_1, \ldots, z_K\}$, candidate j produces an upload with distribution $\mathcal{N}(\mu(z_j), \sigma^2 I_d)$ by equation 14. Under equal priors, the Bayes rule selects the closest release mean:

$$\widehat{J} = \arg \min_{1 \le j \le K} \|Y - \mu(z_j)\|_2^2. \tag{24}$$

The attacker fits no parameters and queries no denoiser; exact ties receive uniform fractional credit.

For each of the 2,048 reconstruction queries, we form nested galleries of size $K = 2, 10, 100$ using distinct same-gender distractors from the full 32,240-image test pool, under the access assumptions in Appendix E.1. Each gallery size uses the same upload. Intervals use 1,000 image bootstraps stratified by gender, conditional on the galleries and uploads.

Table 19: Exact-latent source identification: accuracy (%) and conditional 95% bootstrap intervals over 2,048 queries. All candidates share the public gender; chance accuracy is $100/K\%$.

Encoder	$K = 2$	$K = 10$	$K = 100$
KL-f4	71.24 [69.24, 73.15]	27.15 [25.29, 28.96]	6.20 [5.18, 7.23]
VQ-f4	72.17 [70.26, 74.02]	29.30 [27.39, 31.20]	6.49 [5.42, 7.57]
SoftVQ-VAE	60.79 [58.50, 63.09]	16.11 [14.55, 17.77]	1.95 [1.37, 2.54]
Chance	50.00	10.00	1.00

Accuracy exceeds chance across encoders and gallery sizes. This result demonstrates association with known exact latents; it does not measure identification of an arbitrary unknown person or training membership.

Binary check. For two candidates with release means μ_A and μ_B, the equal-prior Bayes accuracy is $\Phi(\|\mu_B - \mu_A\|_2/(2\sigma))$, where Φ is the standard normal CDF. A separate same-gender paired evaluation agrees closely with this prediction for all three encoders.

These experiments concern the shared denoisers and individual latent uploads under the stated attacker information. Near-chance membership results do not establish differential privacy or rule out stronger attacks; source identification with exact candidate knowledge does not by itself contradict Theorem 4.1.

F Proof of Theorem 4.1

We first recall the notion of Rényi differential privacy (RDP) (Mironov, 2017), which is convenient for analyzing the Gaussian mechanism. We will derive an RDP guarantee for the privatization step and then convert it to an (ϵ, δ)-DP guarantee.

Definition F.1 (Rényi Differential Privacy)**.** A randomized mechanism $\mathcal{A}$ satisfies (γ, ρ)-Rényi differential privacy, for $\gamma > 1$ and $\rho > 0$, if for any adjacent datasets $D, D' \in \mathcal{D}$ that differ in one element,

$$D_\gamma\big(\mathcal{A}(D) \,\|\, \mathcal{A}(D')\big) = \frac{1}{\gamma - 1} \log \mathbb{E}_{o \sim \mathcal{A}(D')} \left[\left(\frac{\mathbb{P}[\mathcal{A}(D) = o]}{\mathbb{P}[\mathcal{A}(D') = o]} \right)^\gamma \right] \leq \rho.$$

The following standard result converts RDP to approximate DP.

Lemma F.2 (RDP to DP (Mironov, 2017))**.** *If a randomized mechanism $\mathcal{A}$ satisfies (γ, ρ)-RDP, then for any $\delta \in (0, 1)$, $\mathcal{A}$ satisfies*

$$\left(\rho + \frac{\log(1/\delta)}{\gamma - 1}, \delta \right) \text{-DP.}$$

We also use the RDP guarantee of the Gaussian mechanism.

Lemma F.3 (Gaussian Mechanism (Mironov, 2017))**.** *Let q be a function with ℓ_2-sensitivity*

$$S_2 = \sup_{D, D'} \|q(D) - q(D')\|_2,$$

where the supremum is over adjacent datasets D, D'. Then the Gaussian mechanism

$$\mathcal{A}(D) = q(D) + z, \qquad z \sim \mathcal{N}(0, \sigma^2 I),$$

satisfies

$$\left(\gamma, \frac{\gamma S_2^2}{2\sigma^2} \right) \text{-RDP.}$$

Proof of Theorem 4.1. The shared global denoiser is trained only on the privatized latent dataset

$$\widetilde{Z} = \{\widetilde{Z}_m\}_{m \in [M]}.$$

For each client m and sample i, the privatized latent is generated as

$$\widetilde{z}_0^{i,m} = \sqrt{\bar{\alpha}_{t_0}}\, z_0^{i,m} + \sqrt{1 - \bar{\alpha}_{t_0}}\, \xi, \qquad \xi \sim \mathcal{N}(0, I).$$

Thus, the privatization step is a Gaussian mechanism applied to the clipped latent representation. Assuming $\|z_0^{i,m}\|_2 \leq C_z$, the sensitivity of the deterministic part

$$q(z_0) = \sqrt{\bar{\alpha}_{t_0}}\, z_0$$

is bounded by

$$S_2 \leq \sup_{\|z\|_2, \|z'\|_2 \leq C_z} \sqrt{\bar{\alpha}_{t_0}} \|z - z'\|_2 \leq 2\sqrt{\bar{\alpha}_{t_0}}\, C_z.$$

The Gaussian noise variance is

$$\sigma^2 = 1 - \bar{\alpha}_{t_0}.$$

By Lemma F.3, each privatized latent sample satisfies

$$\left(\gamma, \frac{\gamma S_2^2}{2\sigma^2}\right)\text{-RDP} = \left(\gamma, \gamma \cdot \frac{2\bar{\alpha}_{t_0} C^2}{1-\bar{\alpha}_{t_0}}\right)\text{-RDP}.$$

Let

$$\tau = \frac{2\bar{\alpha}_{t_0} C^2}{1-\bar{\alpha}_{t_0}}.$$

Then each privatized latent is $(\gamma, \gamma\tau)$-RDP. Applying Lemma F.2, we obtain an (ϵ, δ)-DP guarantee with

$$\epsilon = \gamma\tau + \frac{\log(1/\delta)}{\gamma - 1}.$$

Optimizing over $\gamma > 1$ gives

$$\gamma = 1 + \sqrt{\frac{\log(1/\delta)}{\tau}},$$

and therefore

$$\epsilon = \tau + 2\sqrt{\tau \log(1/\delta)}.$$

Substituting the value of τ, each privatized latent sample satisfies

$$\left(\frac{2\bar{\alpha}_{t_0} C^2}{1-\bar{\alpha}_{t_0}} + C\sqrt{\frac{8\bar{\alpha}_{t_0}\log(1/\delta)}{1-\bar{\alpha}_{t_0}}}, \delta\right)\text{-DP}.$$

Since this randomization is applied locally to each sample before communication, the construction of $\widetilde{Z}$ satisfies the same per-sample local DP guarantee. Finally, the shared global denoiser is trained only as a function of $\widetilde{Z}$. Therefore, by the post-processing property of differential privacy, the learned shared denoiser satisfies

$$\left(\frac{2\bar{\alpha}_{t_0} C_z^2}{1-\bar{\alpha}_{t_0}} + C_z\sqrt{\frac{8\bar{\alpha}_{t_0}\log(1/\delta)}{1-\bar{\alpha}_{t_0}}}, \delta\right)\text{-local DP}.$$

This proves the theorem. □

G Proof of Theorem 4.2

In this section, we provide the formal statement of Theorem 4.2 and the corresponding proofs. We first introduce a more detailed problem setup. We consider the latent data distribution to be a GMM with K equally weighted Gaussians and is shared by all clients:

$$p_0 = \frac{1}{K}\sum_{j=1}^{K} \mathcal{N}(\mu_j, I_d). \tag{25}$$

Each client m has n_l independent samples from p_0, and the total number of the pooled privatized data is N.

Let

$$A = \bar{\alpha}_{t_0} = \prod_{s=1}^{t_0}(1-\beta_s) \in (0,1), \qquad a = \sqrt{A}, \qquad b = 1 - A, \qquad \sigma^2 = \frac{1-A}{A}. \tag{26}$$

Each data point in the pooled privatized dataset is independently perturbed through the Gaussian transition $Q_A(y \mid x) = \mathcal{N}(y; ax, bI_d)$. Given component j, the perturbed mean is $a\mu_j$ and the covariance is $a^2 I_d + bI_d = I_d$. Hence, the global training distribution is

$$r_A = p_0 Q_A = \frac{1}{K}\sum_{j=1}^{K} \mathcal{N}(a\mu_j, I_d). \tag{27}$$

Both the global and local learning problems therefore have known equal component weights and identity covariance. The unknown parameters are the clean means $\{\mu_j\}$ locally and the noisy means $\{a\mu_j\}$ globally.

DDPM learning objective. For the distribution $r \in \{p_0, r_A\}$, let $r_u = P_u r$ be the distribution of

$$Z_u = e^{-u} Z_0 + \sqrt{1 - e^{-2u}}\, E, \qquad Z_0 \sim r, \quad E \sim \mathcal{N}(0, I_d). \tag{28}$$

Following Shah et al. (2023), we consider the following score-matching objective at a noise level $u > 0$:

$$\mathcal{L}_{\mathrm{SM}}(\theta; r, u) = \mathbb{E}_{Z_u \sim r_u}\left[\|\xi_\theta(Z_u, u) - \nabla \log r_u(Z_u)\|_2^2 \right], \tag{29}$$

where ξ_θ is the denoising score model and $\nabla \log r_u$ is the oracle score. Let $v_u = 1 - e^{-2u}$, we have the corresponding DDPM denoising objective as

$$\mathcal{L}_{\mathrm{DSM}}(\theta; r, u) = \mathbb{E}_{Z_0, E}\left[\left\| \xi_\theta(Z_u, u) + \frac{E}{\sqrt{v_u}} \right\|_2^2 \right]. \tag{30}$$

These two objectives satisfy

$$\mathcal{L}_{\mathrm{DSM}}(\theta; r, u) = \mathcal{L}_{\mathrm{SM}}(\theta; r, u) + C_{r,u}, \tag{31}$$

where $C_{r,u}$ is independent of θ. Let the noise predictor $\epsilon_\theta(z, u) = -\sqrt{v_u}\, \xi_\theta(z, u)$,

$$\mathbb{E}_{Z_0, E} \|E - \epsilon_\theta(Z_u, u)\|_2^2 = v_u \mathcal{L}_{\mathrm{DSM}}(\theta; r, u). \tag{32}$$

Thus, the objective corresponds to the loss in Algorithm 3.

GMM score parameterization. Write the training target as $r = K^{-1} \sum_{j=1}^K \mathcal{N}(c_j^\star, I_d)$, where $c_j^\star = \mu_j$ locally and $c_j^\star = a\mu_j$ globally. Under this GMM model, the oracle score takes the form

$$\nabla \log r_u(z) = -z + \sum_{j=1}^K w_j^\star(z, u) c_{j,u}^\star, \tag{33}$$

$$w_j^\star(z, u) = \frac{\exp\!\big(-\|z - c_{j,u}^\star\|_2^2/2\big)}{\sum_{i=1}^K \exp\!\big(-\|z - c_{i,u}^\star\|_2^2/2\big)}, \qquad c_{j,u}^\star = e^{-u} c_j^\star. \tag{34}$$

Following Shah et al. (2023), we use the corresponding parameterized denoiser

$$\xi_\theta(z, u) = -z + \sum_{j=1}^K w_{\theta,j}(z, u) c_{j,u}, \tag{35}$$

$$w_{\theta,j}(z, u) = \frac{\exp\!\big(-\|z - c_{j,u}\|_2^2/2\big)}{\sum_{i=1}^K \exp(-\|z - c_{i,u}\|_2^2/2)}, \qquad c_{j,u} = e^{-u} c_j, \tag{36}$$

where $\theta = (c_1, \ldots, c_K) \in (\mathbb{R}^d)^K$ contains the trainable means. These parameters define the learned GMM

$$h_\theta = \frac{1}{K} \sum_{j=1}^K \mathcal{N}(c_j, I_d), \qquad \xi_\theta(\cdot, u) = \nabla \log(P_u h_\theta). \tag{37}$$

The true parameter vectors are $\theta_l^\star = (\mu_1, \ldots, \mu_K)$ and $\theta_g^\star = (a\mu_1, \ldots, a\mu_K)$ for the local and global learners, respectively. Learning the component means at a suitable noise level therefore determines the learned scores throughout the diffusion process. We use the selected-noise training procedure of Shah et al. (2023) under the conditions specified in the parameter-learning lemma below.

Learned models and composition. Let the learned local and global distributions to be

$$f = \frac{1}{K} \sum_{j=1}^K \mathcal{N}(\widehat{\mu}_j, I_d), \qquad g = \frac{1}{K} \sum_{j=1}^K \mathcal{N}(\widehat{\nu}_j, I_d). \tag{38}$$

For the global model, $\widehat{\nu}_j$ estimates $a\mu_j$. Let $f_A = fQ_A$ and the true and learned reverses to be

$$R_A(dx \mid y) = \frac{p_0(x) Q_A(y \mid x)}{r_A(y)}\, dx, \qquad \widehat{R}_A(dx \mid y) = \frac{f(x) Q_A(y \mid x)}{f_A(y)}\, dx. \tag{39}$$

The learned collaborative density is

$$\widehat{p}_{\mathrm{PF}}(x) := \int g(y) \frac{f(x) Q_A(y \mid x)}{f_A(y)}\, dy, \qquad \widehat{p}_{\mathrm{PF}} = g\widehat{R}_A = p^m_{\widehat{D},D_m}. \tag{40}$$

This follows the global–local structure of Algorithm 2. The density is generated by exact sampling from g followed by the learned local stochastic reverse conditional.

Here, the local *ideal stochastic reverse transition* is the exact conditional $\widehat{R}_A(\cdot \mid y)$ induced by the learned local density f and the Gaussian forward channel Q_A in equation 39. It maps an input at the handoff level $A = \bar{\alpha}_{t_0}$ to a random clean latent. The ideal composition draws $Y \sim g$ and then $X \mid Y = y \sim \widehat{R}_A(\cdot \mid y)$, giving the output density $\widehat{p}_{\mathrm{PF}}$. "Ideal" refers to exact sampling from the learned models. Both stages admit direct sampling in the GMM setting. Approximation errors of numerical diffusion samplers are not analyzed.

Assumptions. We consider three common assumptions: data separability, warm-start initialization, and a bounded mean for the target distribution (Chen et al., 2023b;a; Benton et al., 2024).

Assumption G.1 (Component separation (Shah et al., 2023))**.** For GMM in equation 25, let $\Delta = \min_{i \neq j} \|\mu_i - \mu_j\|_2$, we assume the following holds for a sufficiently large constant C_{sep}:

$$\sqrt{A}\,\Delta \geq C_{\mathrm{sep}} \sqrt{\log(\min(K, d))}. \tag{41}$$

Assumption G.2 (Warm-start initialization (Shah et al., 2023))**.** For each component, we have local and global initializations $\mu^{(0)}_{l,j}, \mu^{(0)}_{g,j}$ satisfy

$$\max_j \|\mu^{(0)}_{l,j} - \mu_j\|_2 \leq c_{\mathrm{init}} \sqrt{\log(\min(K, d))}, \tag{42}$$

$$\max_j \|\nu^{(0)}_{g,j} - a\mu_j\|_2 \leq c_{\mathrm{init}} \sqrt{\log(\min(K, d))}, \tag{43}$$

where $c_{\mathrm{init}} > 0$ is a sufficiently small constant.

Assumption G.3 (Bounded true means)**.** For GMM in equation 25, we have

$$\max_{j \in [K]} \|\mu_j\|_2 \leq B. \tag{44}$$

Our proof is based on the following result in Shah et al. (2023), which provides the estimation error of mean parameter learning for DDPM with gradient descent.

Lemma G.4 (Adapted from Shah et al. (2023))**.** *Let q be the GMM defined in equation 25. Suppose its means satisfy Assumptions G.1–G.3. If we choose ϵ_n, noise scale μ satisfy*

$$\varepsilon_n = c\sqrt{\frac{K^4 d^5 B^6}{n}} > \frac{1}{\textit{poly(d)}}, \qquad u = \Theta(\varepsilon_n) \tag{45}$$

with some constant c. Then, gradient descent on the DDPM objective at noise scale μ with $H \geq \Theta(\log(\varepsilon_n^{-1} \log d))$ iterations and n samples outputs means $\widehat{\theta} = (\widehat{c}_1, \ldots, \widehat{c}_K)$ satisfying, with high probability,

$$\max_{j \in [K]} \|\widehat{c}_j - c^\star_j\|_2^2 \leq \varepsilon_n^2 = \mathcal{O}\left(\frac{K^4 d^5 B^6}{n}\right). \tag{46}$$

Moreover, the learned density $h_{\widehat{\theta}} = K^{-1} \sum_j \mathcal{N}(\widehat{c}_j, I_d)$ satisfies

$$\mathrm{KL}(r \| h_{\widehat{\theta}}) \leq \frac{1}{2K} \sum_{j=1}^K \|\widehat{c}_j - c^\star_j\|_2^2 \leq \frac{\varepsilon_n^2}{2}. \tag{47}$$

Local and global learning errors. Apply Lemma G.4 to the local target p_0 with $n = n_l$ and $c^\star_j = \mu_j$, and to the global target r_A with $n = N$ and $c^\star_j = a\mu_j$. We have the local and global errors e_l and e_g as:

$$e_l^2 := \max_j \|\widehat{\mu}^m_j - \mu_j\|_2^2 \leq \widetilde{\mathcal{O}}\left(\frac{K^4 d^5 B^6}{n_l}\right), \tag{48}$$

$$e_g^2 := \max_j \|\widehat{\nu}_j - a\mu_j\|_2^2 \leq \widetilde{\mathcal{O}}\left(\frac{K^4 d^5 B^6}{N}\right). \tag{49}$$

Thus we have

$$\mathrm{KL}(p_0 \| f_m) \le \frac{e_l^2}{2}, \qquad \mathrm{KL}(r_A \| g) \le \frac{e_g^2}{2}. \tag{50}$$

Now we are ready to provide the formal theorem and its corresponding proofs.

Theorem G.5 (KL approximation of the composed PF-LDM model)**.** *Suppose Assumptions G.1–G.3 hold, and assume each client m holds n_l latent samples from p_0, each with ℓ_2 norm at most C_z and the global holds N samples generated through equation 4. Under the same choices of t_0, $\{\beta_t\}$, and C_z as in Theorem 4.1, train the local and global denoisers by gradient descent on the DDPM denoising objective and GMM score parameterization. Under the same conditions in Lemma G.4, the learned composed density $\widehat{p}_{\mathrm{PF}} = g\widehat{R}_A$ defined in equation 40 satisfies*

$$\mathrm{KL}(p_0 \,\|\, \widehat{p}_{\mathrm{PF}}) \le \widetilde{\mathcal{O}}\left[K^4 d^5 \left(\frac{B^8 \bar{\alpha}_{t_0}}{N} + \frac{B^{10}(1-\bar{\alpha}_{t_0})}{n_l}\right)\right]. \tag{51}$$

Here $\bar{\alpha}_{t_0} = (1+\sigma^2)^{-1}$ and $\widetilde{\mathcal{O}}$ suppresses logarithmic factors.

Proof. Note that under the same choices of t_0, $\{\beta_t\}$, and C_z as in Theorem 4.1, there will be no clipping for global noisy samples. In the following discussion, we let $\mathcal{E}(A) = \mathrm{KL}(p_0 \| \widehat{p}_{\mathrm{PF}})$.

Step 1: Local conditional error. For a given learned local model f, define

$$\mathcal{C}_l(A) = \mathbb{E}_{Y \sim r_A} D(Y), \text{ where } D(y) = \mathrm{KL}\Big(R_A(\cdot \mid y) \,\Big\|\, \widehat{R}_A(\cdot \mid y)\Big).$$

This quantity measures the learned local transition's error when initialized from the true noisy distribution. Applying the KL chain rule to the joint densities $p_0(x)Q_A(y \mid x)$ and $f(x)Q_A(y \mid x)$, we have

$$\mathcal{C}_l(A) = \mathrm{KL}(p_0 \| f) - \mathrm{KL}(r_A \| f_A) \le \frac{L}{2}, \tag{52}$$

where $L = K^{-1}\sum_j \|\widehat{\mu}_j - \mu_j\|_2^2$.

To obtain an explicit dependence on A, let $p_u = P_u p_0$, $f_u = P_u f$, $s_u = \nabla \log p_u$, and $\widehat{s}_u = \nabla \log f_u$. For $c = e^{-u}$, let

$$\Delta_j = \widehat{\mu}_j - \mu_j, \qquad \ell_j(x) = c\mu_j^\top x - \frac{c^2}{2}\|\mu_j\|_2^2,$$

with $\widehat{\ell}_j$ defined analogously. The true and learned mixture weights $w(x) = \mathrm{softmax}(\ell(x))$ and $\widehat{w}(x) = \mathrm{softmax}(\widehat{\ell}(x))$. Using the GMM score formula, $e_l = \max_j \|\Delta_j\|_2$, and $\max_j\{\|\mu_j\|_2, \|\widehat{\mu}_j\|_2\} \le R$, we obtain

$$\begin{aligned}\|\widehat{s}_u(x) - s_u(x)\|_2 &\le ce_l + cR\|\widehat{w}(x) - w(x)\|_1 \\ &\le ce_l + 2cR\|\widehat{\ell}(x) - \ell(x)\|_\infty,\end{aligned} \tag{53}$$

where the second inequality uses $\|\mathrm{softmax}(v) - \mathrm{softmax}(w)\|_1 \le 2\|v - w\|_\infty$.

Denote a sample from p_u by $X = c\mu_I + Z$, where I is uniform on $[K]$ and $Z \sim \mathcal{N}(0, I_d)$ is independent of I. Then

$$\widehat{\ell}_j(X) - \ell_j(X) = c\Delta_j^\top Z + c^2 \Delta_j^\top \left(\mu_I - \frac{\mu_j + \widehat{\mu}_j}{2}\right).$$

Consequently, with $M_\Delta = \max_j |\Delta_j^\top Z|$,

$$\|\widehat{\ell}(X) - \ell(X)\|_\infty \le cM_\Delta + 2c^2 R e_l.$$

For $e_l > 0$, each projection $\Delta_j^\top Z$ is Gaussian with variance at most e_l^2. Thus, a union bound therefore gives, for $v \ge 0$,

$$\Pr(M_\Delta^2 > v) \le \min\left\{1,\, 2K\exp\left(-\frac{v}{2e_l^2}\right)\right\}.$$

Integrating this tail bound yields

$$\mathbb{E}M_\Delta^2 = \int_0^\infty \Pr(M_\Delta^2 > v)\, dv \le 2e_l^2[\log(2K)+1].$$

Applying $(x+y)^2 \le 2x^2 + 2y^2$ twice to equation 53 and substituting the above bounds, we obtain

$$\begin{aligned}\mathbb{E}_{p_u}\|\widehat{s}_u - s_u\|_2^2 &\le 2c^2 e_l^2 + 16c^4R^2\mathbb{E}M_\Delta^2 + 64c^6R^4e_l^2 \\ &\le \big[2c^2 + 32c^4R^2[\log(2K)+1] + 64c^6R^4\big]e_l^2 \\ &\le \Lambda_R e^{-2u} e_l^2, \end{aligned} \tag{54}$$

where the last inequality uses $c = e^{-u} \le 1$.

We next relate this score discrepancy to $\mathcal{C}_l(A)$. Both forward marginals satisfy $\partial_u h = \Delta_x h + \nabla_x \cdot (xh)$. Writing $\rho_u = p_u/f_u$, differentiation and integration by parts give

$$\begin{aligned}\frac{d}{du}\,\mathrm{KL}(p_u\|f_u) &= \int (\partial_u p_u)\log\rho_u\,dx - \int \rho_u\,\partial_u f_u\,dx \\ &= -\int (\nabla p_u + x p_u)\cdot\nabla\log\rho_u\,dx + \int \nabla\rho_u\cdot(\nabla f_u + x f_u)\,dx \\ &= -\int p_u\|\nabla\log\rho_u\|_2^2\,dx \\ &= -\mathbb{E}_{p_u}\|s_u - \widehat{s}_u\|_2^2.\end{aligned}$$

All spatial integrals are over $\mathbb{R}^d$. The Gaussian tails and polynomial growth of the relevant log-density ratios and derivatives justify differentiation under the integral and the vanishing boundary terms.

Set $u_0 = -\frac{1}{2}\log A$. Since P_0 is the identity and $e^{-2u_0} = A$, the endpoint marginals are $p_{u=0} = p_0$, $f_{u=0} = f$, $p_{u_0} = r_A$, and $f_{u_0} = f_A$. Combining equation 52 with the preceding derivative identity gives

$$\begin{aligned}\mathcal{C}_l(A) &= \mathrm{KL}(p_0\|f) - \mathrm{KL}(p_{u_0}\|f_{u_0}) \\ &= -\int_0^{u_0} \frac{d}{du}\,\mathrm{KL}(p_u\|f_u)\,du \\ &= \int_0^{u_0} \mathbb{E}_{p_u}\|s_u - \widehat{s}_u\|_2^2\,du.\end{aligned} \tag{55}$$

Applying equation 54, we conclude

$$\mathcal{C}_l(A) \le \Lambda_R e_l^2 \int_0^{u_0} e^{-2u}\,du = \frac{\Lambda_R}{2}(1-A)e_l^2.$$

Together with equation 52, this yields

$$\mathcal{C}_l(A) \le \frac{1}{2}\min\{L, \Lambda_R(1-A)e_l^2\}. \tag{56}$$

Step 2: Effect of global input error on local refinement. Step 1 controls the learned local transition at the same input as the true transition. In our composition, however, the input comes from the learned global model and may differ from a true noisy sample. For arbitrary inputs $y, z \in \mathbb{R}^d$ at the same handoff level A, we therefore bound $\mathrm{KL}(R_A(\cdot\mid y)\|\widehat{R}_A(\cdot\mid z))$ in terms of the same-input local error

$$D(y) = \mathrm{KL}\Big(R_A(\cdot\mid y)\,\Big\|\,\widehat{R}_A(\cdot\mid y)\Big)$$

and the input discrepancy $\|y-z\|_2$. Recall that $a = \sqrt{A}$, $b = 1-A > 0$, and $\max_j \|\widehat{\mu}_j\|_2 \le R$.

Posterior representation and moment bound. Under the learned local model, let J denote the mixture index. Conditionally on $J = j$, $X \sim \mathcal{N}(\widehat{\mu}_j, I_d)$ and $Y = aX + \sqrt{b}\,E$, with independent $E \sim$

$\mathcal{N}(0, I_d)$. Thus, $\mathrm{Var}(Y \mid J = j) = I_d$ and $\mathrm{Cov}(X, Y \mid J = j) = aI_d$. Gaussian conditioning and Bayes' rule give

$$\widehat{R}_A(\cdot \mid y) = \sum_{j=1}^{K} \widehat{w}_j(y)\mathcal{N}(ay + b\widehat{\mu}_j, bI_d), \quad \widehat{w}_j(y) = \frac{\exp\bigl(-\|y - a\widehat{\mu}_j\|_2^2/2\bigr)}{\sum_{i=1}^{K} \exp(-\|y - a\widehat{\mu}_i\|_2^2/2)}. \tag{57}$$

For fixed y, a conditional draw can therefore be written as

$$X = ay + b\widehat{\mu}_J + \sqrt{b}\, H, \qquad J \sim \mathrm{Categorical}(\widehat{w}(y)), \qquad H \sim \mathcal{N}(0, I_d),$$

where H is independent of J. Define $\overline{\mu}_y = \sum_j \widehat{w}_j(y)\widehat{\mu}_j$ and $m_y = \mathbb{E}_{\widehat{R}_A(\cdot|y)} X = ay + b\overline{\mu}_y$. Then

$$X - m_y = b(\widehat{\mu}_J - \overline{\mu}_y) + \sqrt{b}\, H.$$

For any $h \in \mathbb{R}^d$, the scalar $bh^\top \widehat{\mu}_J$ lies in an interval of length at most $2bR\|h\|_2$. Hoeffding's moment bound therefore yields

$$\log \mathbb{E}_{J|y} \exp\bigl(bh^\top(\widehat{\mu}_J - \overline{\mu}_y)\bigr) \le \frac{(2bR\|h\|_2)^2}{8} = \frac{b^2R^2}{2}\|h\|_2^2.$$

The independent Gaussian contribution satisfies $\log \mathbb{E}e^{\sqrt{b}\, h^\top H} = b\|h\|_2^2/2$. Combining these two bounds gives

$$\begin{aligned} \log \mathbb{E}_{\widehat{R}_A(\cdot|y)} e^{h^\top(X - m_y)} &= \log \mathbb{E}_{J|y} e^{bh^\top(\widehat{\mu}_J - \overline{\mu}_y)} + \frac{b}{2}\|h\|_2^2 \\ &\le \frac{v_R}{2}\|h\|_2^2, \end{aligned} \tag{58}$$

where $v_R = b + b^2R^2$.

Changing the input is an exponential tilt. By the definition of its reverse conditional,

$$\frac{\widehat{R}_A(x \mid z)}{\widehat{R}_A(x \mid y)} = \frac{Q_A(z \mid x)}{Q_A(y \mid x)} \frac{f_A(y)}{f_A(z)}.$$

For the Gaussian forward channel,

$$\begin{aligned} \log \frac{Q_A(z \mid x)}{Q_A(y \mid x)} &= -\frac{\|z - ax\|_2^2 - \|y - ax\|_2^2}{2b} \\ &= \frac{a}{b}(z - y)^\top x - \frac{\|z\|_2^2 - \|y\|_2^2}{2b}. \end{aligned}$$

All factors independent of x are absorbed into the normalizing constant. Consequently,

$$\widehat{R}_A(dx \mid z) = \frac{e^{h^\top x}\widehat{R}_A(dx \mid y)}{\mathbb{E}_{\widehat{R}_A(\cdot|y)} e^{h^\top X}}, \qquad h = \frac{a(z - y)}{b}. \tag{59}$$

Thus, the input discrepancy determines an exponential reweighting of the learned posterior. We next control the resulting change in KL.

KL bound under an exponential tilt. Consider probability measures P, Q with finite first moments and $D = \mathrm{KL}(P\|Q) < \infty$. Suppose Q satisfies the centered moment bound

$$\log \mathbb{E}_Q e^{h^\top(X - \mathbb{E}_Q X)} \le \frac{v}{2}\|h\|_2^2 \qquad \text{for every } h \in \mathbb{R}^d,$$

where $v > 0$. Define its exponential tilt by $Q_h(dx) = e^{h^\top x}Q(dx)/\mathbb{E}_Q e^{h^\top X}$.

The entropy variational inequality $\mathbb{E}_P F \le D + \log \mathbb{E}_Q e^F$, applied with $F = sw^\top(X - \mathbb{E}_Q X)$, gives

$$sw^\top(\mathbb{E}_P X - \mathbb{E}_Q X) \le D + \frac{vs^2}{2}$$

for every unit vector w and $s > 0$. Dividing by s, optimizing over it, and taking the supremum over w yield

$$\|\mathbb{E}_P X - \mathbb{E}_Q X\|_2 \le \inf_{s>0} \left\{ \frac{D}{s} + \frac{vs}{2} \right\} = \sqrt{2vD}.$$

Using the definition of Q_h and centering its log-moment term, we obtain

$$\begin{aligned} \mathrm{KL}(P\|Q_h) &= D + \log \mathbb{E}_Q e^{h^\top X} - h^\top \mathbb{E}_P X \\ &= D + \log \mathbb{E}_Q e^{h^\top (X - \mathbb{E}_Q X)} + h^\top (\mathbb{E}_Q X - \mathbb{E}_P X) \\ &\le D + \frac{v}{2}\|h\|_2^2 + \sqrt{2vD}\|h\|_2 \\ &= \left(\sqrt{D} + \sqrt{v/2}\|h\|_2\right)^2. \end{aligned} \tag{60}$$

The conditional Gaussian mixtures considered here have finite first moments, so this calculation applies.

Finally, take

$$P = R_A(\cdot \mid y), \qquad Q = \widehat{R}_A(\cdot \mid y), \qquad Q_h = \widehat{R}_A(\cdot \mid z).$$

Then $D = D(y)$, $v = v_R = b + b^2R^2$ by equation 58, and $h = a(z - y)/b$ by equation 59. In particular,

$$\sqrt{\frac{v_R}{2}}\|h\|_2 = \sqrt{\frac{A}{2}\left(\frac{1}{1-A} + R^2\right)}\|y - z\|_2.$$

Taking square roots in equation 60 therefore gives

$$\sqrt{\mathrm{KL}\Big(R_A(\cdot \mid y) \,\Big\|\, \widehat{R}_A(\cdot \mid z)\Big)} \le \sqrt{D(y)} + \lambda_R(A)\|y - z\|_2, \tag{61}$$

where $\lambda_R(A) = \sqrt{\frac{A}{2}\left(\frac{1}{1-A} + R^2\right)}$. The first term is the same-input local error controlled in Step 1, while the second accounts for changing the noisy input. The next step averages this pointwise bound over a coupling of the true and learned global input distributions.

Step 3: Composition with the global model. To apply Step 2, couple the true and learned global inputs as $Y = a\mu_I + Z$ and $V = \widehat{\nu}_I + Z$, where $I \sim \mathrm{Unif}([K])$ and $Z \sim \mathcal{N}(0, I_d)$ are independent. Writing π for their joint law, we have

$$Y \sim r_A, \qquad V \sim g, \qquad \mathbb{E}_\pi\|Y - V\|_2^2 = G, \qquad \mathrm{KL}(r_A\|g) \le G/2, \tag{62}$$

where the equality follows from the shared Gaussian noise, and the inequality follows by comparing corresponding Gaussian components.

The joint laws $\pi(dy, dv)R_A(dx \mid y)$ and $\pi(dy, dv)\widehat{R}_A(dx \mid v)$ have clean marginals p_0 and $\widehat{p}_{\mathrm{PF}}$, respectively. Since their input marginal π is identical, the KL chain rule and marginalization give

$$\mathcal{E}(A) \le \mathbb{E}_\pi \,\mathrm{KL}\Big(R_A(\cdot \mid Y) \,\Big\|\, \widehat{R}_A(\cdot \mid V)\Big). \tag{63}$$

Applying equation 61 from Step 2 and bounding the cross term by Cauchy–Schwarz yields

$$\begin{aligned} \sqrt{\mathcal{E}(A)} &\le \left[\mathbb{E}\left(\sqrt{D(Y)} + \lambda_R(A)\|Y - V\|_2\right)^2\right]^{1/2} \\ &\le \sqrt{\mathbb{E}D(Y)} + \lambda_R(A)\sqrt{\mathbb{E}\|Y - V\|_2^2} \\ &= \sqrt{\mathcal{C}_l(A)} + \lambda_R(A)\sqrt{G}, \end{aligned} \tag{64}$$

where the last equality uses $Y \sim r_A$ and equation 62.

To obtain a bound that remains finite as A goes to 1, compare instead the joint laws $r_A(dy)R_A(dx \mid y)$ and $g(dy)\widehat{R}_A(dx \mid y)$. The same chain-rule argument gives

$$\mathcal{E}(A) \le \mathrm{KL}(r_A\|g) + \mathcal{C}_l(A) \le G/2 + \mathcal{C}_l(A),$$

and hence $\sqrt{\mathcal{E}(A)} \le \sqrt{\mathcal{C}_l(A)} + \sqrt{G/2}$. Combining this with equation 64, we obtain

$$\begin{aligned}\sqrt{\mathcal{E}(A)} &\le \sqrt{\mathcal{C}_l(A)} + \min\{1/\sqrt{2}, \lambda_R(A)\}\sqrt{G} \\ &\le \sqrt{\mathcal{C}_l(A)} + C_R\sqrt{AG}, \end{aligned} \tag{65}$$

where $C_R^2 = 1 + R^2/2$. This is because for $A \le 1/2$,

$$\lambda_R(A)^2 = \frac{A}{2}\left(\frac{1}{1-A} + R^2\right) \le C_R^2 A,$$

while for $A \ge 1/2$, $1/2 \le A \le C_R^2 A$.

Finally, squaring equation 65 on both side, using $(x+y)^2 \le 2x^2 + 2y^2$, and applying the local-error bound equation 56 give

$$\begin{aligned}\mathcal{E}(A) &\le 2\mathcal{C}_l(A) + 2C_R^2 AG \\ &\le (2 + R^2)AG + \min\{L, \Lambda_R(1-A)e_l^2\}.\end{aligned}$$

Step 4: Apply the learning lemma. Recall that Lemma G.4 yields the local and global parameter error rates. Suppose $e_l \le \varepsilon_{n_l} \le 1 \le B$, so $\max_j \|\widehat{\mu}_j\|_2 \le B + e_l \le 2B$. Therefore, take $R = 2B$ in the previous bound, and plug in $G \le e_g^2$, $L \le e_l^2$, we have

$$\mathcal{E}(A) \le \widetilde{\mathcal{O}}\left[K^4 d^5 B^6 \left(\frac{(2+4B^2)A}{N} + \frac{\Lambda_{2B}(1-A)}{n_l}\right)\right]. \tag{66}$$

Since $B \ge 1$, $B^6(2+4B^2) \le 6B^8$ and $B^6\Lambda_{2B} = \mathcal{O}(B^{10}[1 + \log(2K)])$. Absorbing logarithmic factors proves equation 51. □

Benefit of collaboration. We compare the composed density with the local-only density, keeping the learned local model fixed. According to Step 3, we have

$$\mathrm{KL}(p_0 \| \widehat{p}_{\mathrm{PF}}) \le \mathrm{KL}(r_A \| g) + \mathcal{C}_l(A).$$

Substituting the identity from equation 52, we obtain

$$\begin{aligned}\mathrm{KL}(p_0 \| \widehat{p}_{\mathrm{PF}}) &\le \mathrm{KL}(p_0 \| f) + \mathrm{KL}(r_A \| g) - \mathrm{KL}(r_A \| f_A) \\ &< \mathrm{KL}(p_0 \| f),\end{aligned}$$

where the strict inequality follows from the condition that the global model is more accurate at the handoff level $\mathrm{KL}(r_A \| g) < \mathrm{KL}(r_A \| f_A)$.