A. Additional Related Work

In this section, we give a more detailed literature review.

Beyond DDPM. Since the introduction of DDPMs, the field of diffusion models has seen significant advancements aimed at improving computational efficiency, sample quality, and applicability to various domains. Building upon DDPMs, **Latent Diffusion Models (LDMs)** [79] enhance computational efficiency by operating in a compressed latent space rather than directly in pixel space. Utilizing a pre-trained encoder-decoder architecture, LDMs reduce memory and computational requirements for high-resolution image synthesis while maintaining high-quality outputs. This approach has facilitated applications such as text-to-image generation, inpainting, and super-resolution. Notably, **Stable Diffusion**, based on the LDM framework, has popularized these models by providing an open-source and scalable implementation.

Another notable development is the **Denoising Diffusion Implicit Models (DDIMs)** [86], which introduces a non-Markovian diffusion process to accelerate sampling. DDIMs achieve faster generation times by reducing the number of required sampling steps without compromising output quality, addressing one of the primary limitations of earlier diffusion models.

Further advancements include the integration of classifier guidance and classifier-free guidance techniques [24]. These methods enhance controllability in the generation process, allowing for more precise adherence to desired attributes or conditions during sample generation.

Recent research has also explored the application of diffusion models beyond image generation. For instance, **Upsampling Diffusion Probabilistic Models (UDPMs)** [2] focus on upsampling tasks, generating high-resolution images from lower-resolution inputs. Additionally, diffusion models have been adapted for applications in audio generation, text synthesis, and even reinforcement learning scenarios [104].

These developments reflect the rapid evolution of diffusion models, expanding their capabilities and applications across various fields.

Diffusion Models as a Tool for FL Applications. FL has been widely adopted in domains like healthcare, research, finance, and mobile technologies [6, 15, 21, 45, 58, 74, 80, 84, 96]. A separate line of work applies diffusion models to facilitate specific FL tasks rather than federated training of the diffusion model itself. Sattarov et al .[82] developed FedTabDiff, which employs diffusion models to generate high-fidelity tabular data in FL settings without requiring centralized access to raw data. Other works explore how diffusion models can improve privacy-preserving FL through synthetic data augmentation or as generative priors in adversarial training. For example, FL frameworks incorporating diffusion models for one-shot learning and differential privacy constraints have been studied in [55].

Privacy in DMs. Recently, Seo et al. [83] developed an FL framework to train a consensus DM with privacy guarantees by introducing masks during the model update. In contrast to their method, we directly utilize the forward noise to achieve privacy guarantees, aiming to achieve improved privacy-utility trade-offs. In the non-federated setting (i.e., with a single dataset), there are several works [26, 30, 57, 59, 60, 97] that try to achieve privacy guarantees in diffusion models. For example, Dockhorn et al. [26] applied DP-SGD to the training process of diffusion models such that the generated samples are differentially private. Wang et al. [97] developed a training protocol that selectively uses noisy stochastic gradients only to train the final stages of de-noising, thus claiming to improve the privacy-utility trade-off of their trained diffusion models. However, since their method observes the actual noise used in forward diffusion, their approach does not ensure differential privacy guarantees. Beyond DMs, some previous work [5, 11] studied the privacy preservation in federated generative models based on GANs. In our work, we focus on the federated learning setting with DMs, and we aim to provide stronger local differential privacy guarantees.

B. Missing Details from Section 3

B.1. Denoising Diffusion Probabilistic Models (DDPMs)

Denoising Diffusion Probabilistic Models (DDPM) [39] are one of the most popular diffusion models. Their diffusion process $q(x_t|x_{t-1})$, which produces corrupted latents $\{x_t\}_{t\in[T]}$, is defined as a Markov chain that sequentially adds Gaussian noise to the data according to the noise schedule $\{\beta_t\}_{t\in[T]}$:

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I). \tag{10}$$

Algorithm 3 Training Procedure for DDPM (T-DDPM)

input Training dataset D, model parameters θ, time step T, noise scheduling parameters $\{\beta_t\}_{t=1}^{T}$

repeat
  Sample x_0 from D
  Sample $t \sim \text{Uniform}(\{1, \dots, T\})$
  Sample random noise $z \sim \mathcal{N}(0, I)$
  Set $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$
  Take gradient descent step on
    $\nabla_\theta \left\| z - z_\theta\left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} z, t\right) \right\|_2^2$
until converged

output denoiser z_θ

Given large enough T and appropriate noise schedule, the latent x_T nearly follows a standard Gaussian distribution. Furthermore, the diffusion process in equation 10 implies

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I), \tag{11}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$. Thus, we can directly sample an arbitrary latent x_t given x_0.

The reverse process $q(x_{t-1}|x_t)$ can be approximated by:

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_t; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{12}$$

where $\mu_\theta(x_t, t)$ is a trainable neural network with model parameter θ and $\Sigma_\theta(x_t, t)$ can be set to $\sigma_t^2 I$ [39, 71].

To learn p_θ, we can minimize the variational upper bound $\mathbb{E}_q[-\log p_\theta(x_{0:T})/q(x_{1:T}|x_0)]$ which is equivalent to minimizing the following sum of KL divergences [39, 85]

$$\mathbb{E}_q \left[\sum_{t>1} D_{\text{KL}}\big(q(x_{t-1}|x_t, x_0) || p_\theta(x_{t-1}|x_t)\big) \right], \tag{13}$$

where $q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I)$ and $\tilde{\mu}_t(x_t, x_0),\ \tilde{\beta}_t$ can be parameterized [39] as:

$$\tilde{\mu}_t(x_t, x_0) = \frac{1}{\sqrt{1 - \beta_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} z \right),\ \tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t,$$

where $z \sim \mathcal{N}(0, \mathbf{I})$.

Hence, Ho et al. [39] proposed to represent $\mu_\theta(x_t, t)$ in p_θ using a noise prediction network, i.e., denoiser, z_θ:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{1 - \beta_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} z_\theta(x_t, t) \right),$$

and the objective in equation 13 can be simplified to the following loss

$$\mathbb{E}_{t, x_0, z} \left[\|z - z_\theta(x_t, t)\|_2^2 \right],$$

where t is uniformly sampled from 1 to T. The detailed training procedure is summarized in Algorithm 3. After obtaining the denoiser z_θ, we can generate a sample by drawing $x_T \sim \mathcal{N}(0, I)$ and using the form of p_θ. The detailed sampling procedure is illustrated in Algorithm 4.

C. Hierarchical Denoising: Evidence from across Domains and a Case Study

A growing body of work views diffusion sampling as an inherently hierarchical, coarse-to-fine process and exploits this structure across a wide range of domains. In vision, several image restoration and generation methods explicitly separate early steps that recover global layout from later steps that sharpen fine details. For example, progressive deblurring schemes first recover low-frequency structure before they denoise high-frequency content, and coarse-to-fine diffusion transformers operate over multiple resolution levels to support high-quality image restoration and super-resolution [51, 56].

Algorithm 4 Sampling Procedure for DDPM (S-DDPM)

input Denoiser z_θ, time step T, noise scheduling parameters $\{\beta_t, \sigma_t\}_{t=1}^T$
1: Set $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$
2: Sample $x_T \sim \mathcal{N}(0, I)$
3: **for** $t = T, \ldots, 1$ **do**
4: Sample $z \sim \mathcal{N}(0, I)$ if $t > 1$ else $z = 0$
5: $x_{t-1} = \frac{1}{\sqrt{1-\beta_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(x_t, t)\right) + \sigma_t z$
6: **end for**
output x_0

In 3D geometry and scientific applications, hierarchical diffusion models generate coarse molecular scaffolds and then refine local atom-level configurations, or represent indoor scenes with latent trees that encode low-frequency geometry before they add higher-frequency object detail [76, 87]. Hierarchical music models use cascaded diffusion over symbolic sequences: they first sample long-range song structure and then fill in phrase-level detail that respects this scaffold [101]. In language, hierarchical diffusion language models operate over a multi-scale token vocabulary and interpret each denoising step as a prediction of the next semantic scale, moving from abstract high-level tokens to low-level surface forms [103].

Even when models keep the standard DDPM architecture unchanged, several works argue that the denoising process itself already behaves in a coarse-to-fine manner. These works use this structure to share early denoising computations across related prompts or to design noise schedules that prioritise semantically important scales [23, 25]. Our synthetic Gaussian mixture experiment in Figure 5 complements these domain-specific studies by providing a minimal, fully controlled setting in which the hierarchical structure of the learned denoising map becomes directly visible in data space. The experiment shows how the reverse diffusion dynamics recover cluster structure in a coarse-to-fine progression that mirrors the underlying hierarchical mixture. Taken together, these observations suggest that diffusion models do not denoise in an undifferentiated manner, but instead follow a structured trajectory that first resolves coarse semantic information and only later commits to fine-grained detail.

D. Additional Experiments

In this section, we provide more details of our experiments as well as additional results.

Model Architecture. For all datasets, we use U-Net with 8-layer ResNet blocks, where each block consists of 4 downsampling (3 convolutional layers followed by 1 attention-based block) and corresponding upsampling layers. For CIFAR-10, the U-Nets are conditioned on the class label with an embedding vector and for the Colorized MNIST and CelebA, we condition on two different labels, each with seperate embedding vectors. For the Colorized MNIST, each of the conditions reprrepresents color, and the class of each data sample, whereas for CelebA, the hair color and the gender is represented, respectively. For collaborative clients trained with a mixture of clipped and unclipped data, we add another condition on the same U-Net model as described above with an embedding vector for clipped or not. The purpose of this condition is to only guide the generation toward clipped or unclipped image, whereas the unclipped clear image is desired.
Training. All methods use a linear noise schedule with $T = 1000$. We train for $\{100, 300, 500, 700, 1000\}$ epochs and report the best results for each baseline. We fix the privacy budget at $\epsilon = 10, \delta = 10^{-5}$. To satisfy this budget, we set the clipping parameter C near the median data norm: $C = 15$ for CIFAR-10, $C = 10$ for Colorized MNIST, and $C = 35$ for CelebA. We then compute the required local diffusion depth t_0 via Theorem 5.1, yielding $t_0 = 740$, 690, and 850 respectively. All models are trained on a single NVIDIA A100 (80GB) using batch size 128 and AdamW with learning-rate decay.

D.1. Utility Evaluations

Results. Figure 7 and Figure 6 illustrate the results for different datasets. Recall that for our method, we choose $\epsilon = 10, \delta = 10^{-5}$. We can see from the results that our method is able to generate high-quality images, especially for the minority class. For example, on Colorized MNIST dataset, models trained without collaboration perform poorly on their minority classes, for instance, confusing 2 with 3, 8 with 0, in the bottom-right figure, whereas our personalized approach produces high-quality images even for underrepresented digits, demonstrating that the global model captures and transfers shared features across datasets. We also conducted downstream classification tasks using 1000 synthetic samples per class generate by our method and non-collaborative method. We train simple CNN models and a ResNet18 model using the synthetic samples

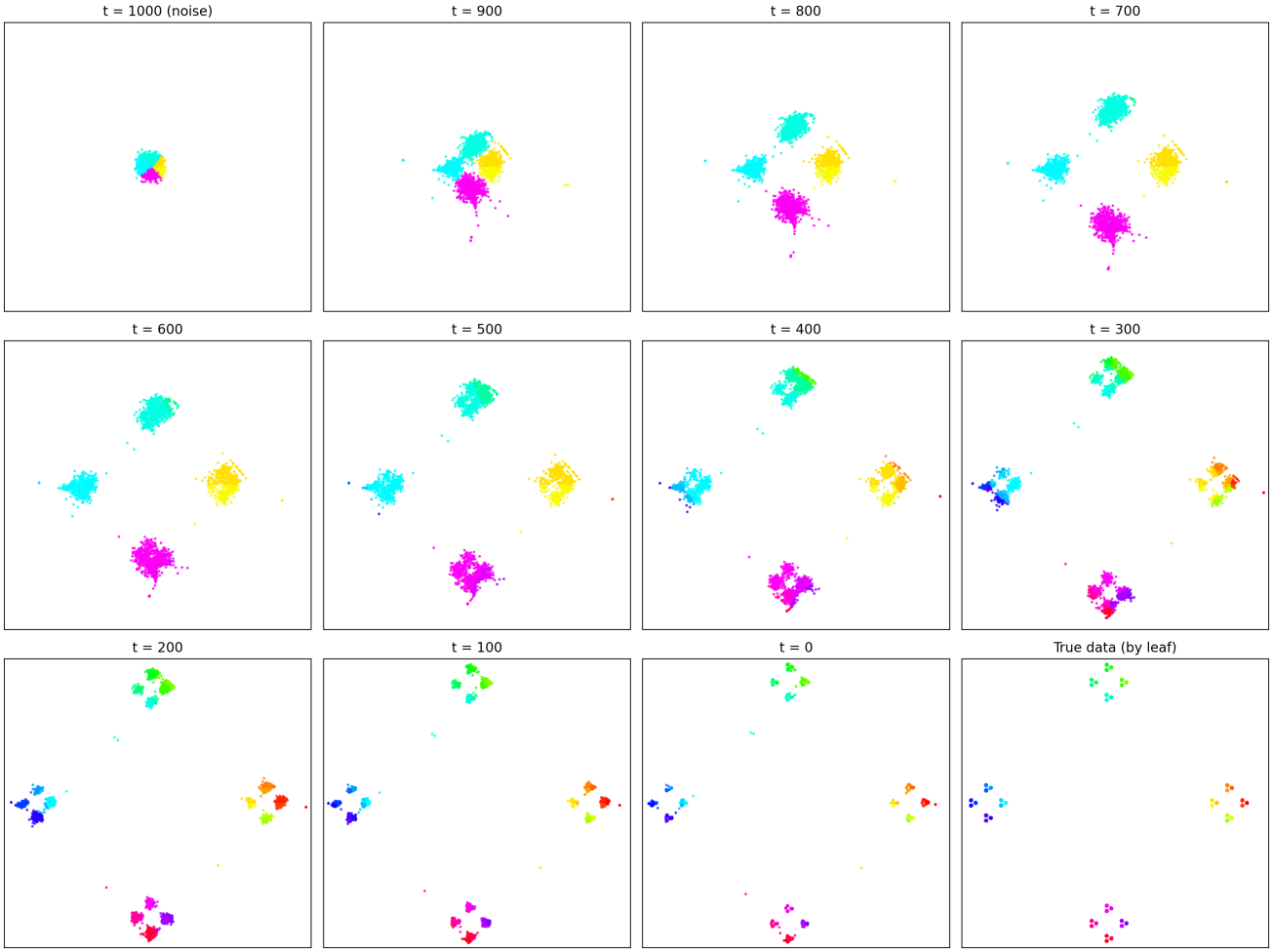

Figure 5. Hierarchical denoising in a synthetic Gaussian mixture. The bottom-right panel shows samples from a depth-3 hierarchical mixture of Gaussians in two dimensions, where each internal node in the tree splits its parent cluster into several child clusters at a finer spatial scale. The remaining panels, show samples from a DDPM trained on this distribution, stopped at intermediate denoising timesteps t (with pure Gaussian noise at the top left and $t = 0$ immediately to the left of the true data). As t decreases, the samples first collapse from isotropic noise into a single coarse blob, then separate into a small number of high-level clusters that correspond to top-level branches of the hierarchy, and finally split into the individual leaf Gaussians. This experiment demonstrates, in a setting where the ground-truth generative structure is known, that the reverse diffusion dynamics recover cluster structure in a coarse-to-fine fashion that mirrors the hierarchical mixture.

and test their classification accuracy on the test dataset. Table 4 reports the results for different methods. Our method significantly outperforms the non-collaborative baseline in test accuracy (over 10 runs). We also report the FID scores of the centralized private fine-tuning method, i.e., DP-LDM [60] as a reference (see Table 5). Note that centralized DP methods such as DP-LDM [60] and RAPID [41] cannot be directly applied in the FL setting and instead rely on public datasets for private fine-tuning, making direct comparisons less fair. Prior work [88] also notes that benchmarks like CIFAR for private fine-tuning methods can be misleading. Here, we include DP-LDM as a private reference. Our method outperforms DP-LDM in our settings.

D.2. Privacy Evaluations

Membership Inference Attacks. PIA [53] differentiates *member* and *non-member* samples by computing the discrepancy between the predicted noise at a given timestep t and a pseudo-groundtruth trajectory derived from the model's own output

Table 4. Classification accuracy for downstream tasks.

	Our		Non-collaborative	
Dataset	CNN	ResNet18	CNN	ResNet18
CIFAR10	57.4	63.6	55.4	61.8
CelebA	69.9	69.0	49.2	49.1

Table 5. FID scores for the centralized private fine-tuning method.

Public data →Private data	Major	Minor	Avg
EMNIST→MNIST	17.9	15.5	16.7
ImageNet→CIFAR10	44.0	36.5	38.4

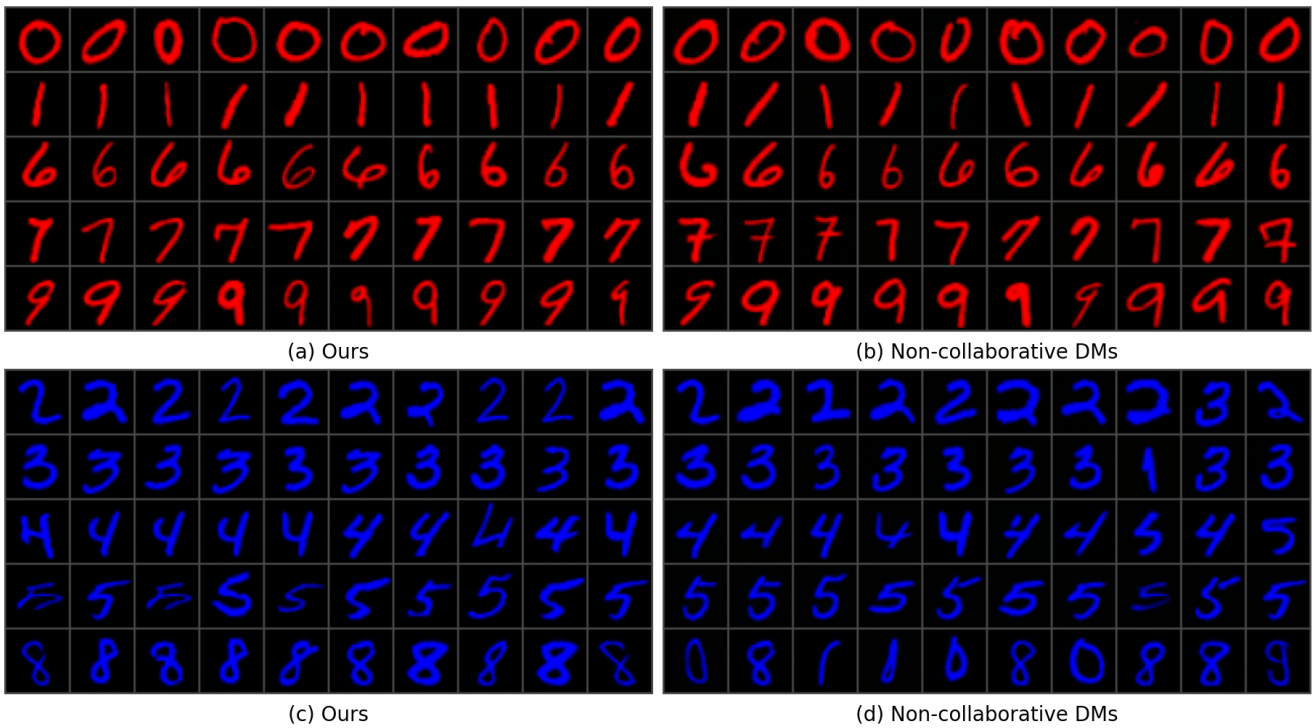

Figure 6. Colorized MNIST samples generated by different methods. (a), (b) correspond to samples generated for majority class group: red, 0,1,6,7,9. (c), (d) correspond to samples generated for minority class group: blue, 2,3,4,5,8. We report the results for the model trained by the first client.

at $t = 0$. Specifically, the attack metric for a given input sample x_0 is defined as:

$$R_{t,p} = \left\| z_\theta(x_0, 0) - z_\theta\left(\sqrt{\bar{a}_t} x_0 + \sqrt{1-\bar{a}_t}\, z_\theta(x_0, 0), t\right) \right\|_p,$$

where $z_\theta(x_0, 0)$ is the model's predicted noise at $t = 0$, and $\bar{\alpha}_t$ denotes the cumulative product of noise schedulers $\{\beta_t\}_{t=1}^T$ up to t. Samples with lower $R_{t,p}$ values are more likely to be in the training set.

We choose PIA as the evaluation attack for its low query complexity (only two queries required per sample) and superior performance in low-FPR settings. Following prior work, we set the attack timestep to $t = 200$, which balances discriminability and stability for discrete-time diffusion models, with 50% of the training set used as member samples and the remaining 50% treated as non-member samples.

Figure 7. CIFAR-10 samples generated by different methods. (a), (b) correspond to samples generated for majority class group. (c), (d) correspond to samples generated for minority class group. We report the results for the model trained with cluster 1 as majority class.

Table 6. MIA for different nonprivate models (centralized method). S1, S2, S3 correspond to the models trained for 300, 700, and 1000 epochs, respectively. 50% AUC/ASR means the random guess. Lower TPR@1% FPR indicates stronger privacy protection.

	CIFAR10	Colorized MNIST	CelebA
MODELS	AUC /ASR/TPR@1% FPR	AUC /ASR/TPR@1% FPR	AUC /ASR/TPR@1% FPR
S1	53.03/52.44/1.23	62.71/59.44/1.51	61.59/58.36/3.28
S2	64.79/60.91/2.53	97.71/93.96/52.52	90.15/82.12/31.36
S3	82.13/75.14/8.75	99.62/97.72/93.39	99.59/97.74/93.34

The PIA results for our method and the centralized method are summarized in Table 2 and Table 6. Across all datasets, the attack performance remains close to the random guess baseline (50%), indicating minimal privacy leakage and demonstrating the method's strong resistance to MIA. In contrast, standard nonprivate models (centralized method) suffer from greater privacy leakage than the proposed method.

Memorization. Given a generated sample w, let x_1 and x_2 be the nearest and second-nearest neighbors in the training set D under the ℓ_2 norm:

$$x_1 = \arg\min_{x \in D} \|w - x\|_2, \quad x_2 = \arg\min_{\substack{x \in D \\ x \neq x_1}} \|w - x\|_2.$$

We consider w to be a memorized sample if the following condition is satisfied:

$$\frac{\|w - x_1\|_2}{\|w - x_2\|_2} < \frac{1}{3}.$$

The threshold of $1/3$ is empirically determined based on its strong alignment with human judgment, and has been widely adopted in previous studies to identify memorized samples with high reliability [10, 34]. Compared with direct thresholds based on ℓ_2 or ℓ_∞ distances, this criterion offers a more balanced sensitivity to true memorization cases.

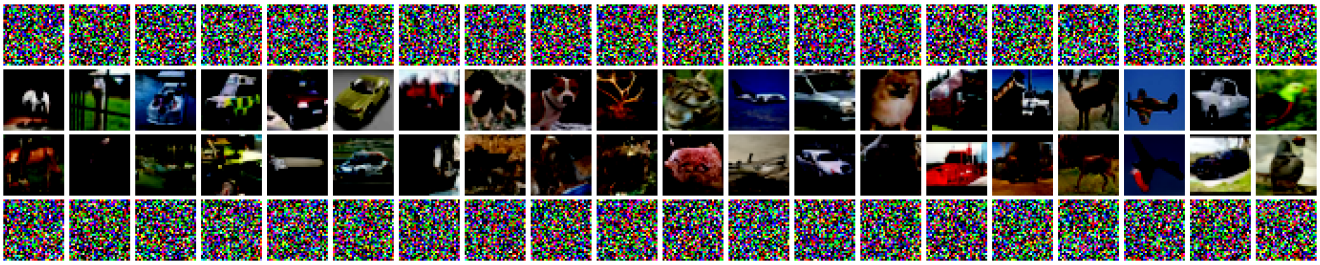

Figure 8. Reconstruction attack for CIFAR-10 dataset. First row: noisy data sent to the server. Second row: original data. Third row: reconstructed data from the noisy data using the pretrained attack model. Bottom row: reconstructed data from the noisy data using the global attack model.

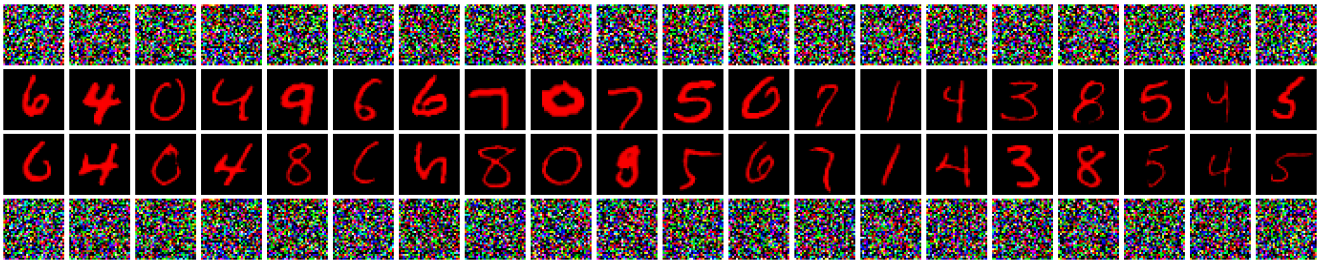

Figure 9. Reconstruction attack for Colorized MNIST dataset. First row: noisy data sent to the server. Second row: original data. Third row: reconstructed data from the noisy data using the pretrained attack model. Bottom row: reconstructed data from the noisy data using the global attack model.

Reconstruction Attack. Recall that we consider two types of attackers: (1) a global model attacker, which uses the same diffusion model trained collaboratively across clients, representing a threat with full parameter access to the DP model; and (2) a pretrained model attacker, which uses a denoiser pretrained on a similar dataset, representing an attacker with partial knowledge of the target data. For the CIFAR-10 dataset, the pretrained attack model is trained based on the STL-10 [20] dataset. For this dataset, we aim to reconstruct images except for the class Frog.

For the Colorized MNIST dataset, the pretrained attack model is trained on a dataset that consists of 1,000 red/blue samples for each digit in $\{2, 3, 4, 5, 8\}$ and 100 blue samples for each digit in $\{0, 1, 6, 7, 9\}$. For this dataset, we aim to reconstruct images with all red digits.

For the CelebA dataset, the pretrained attack model is trained on a dataset that consists of 5,000 female–black-hair, 5,000 female–non-black-hair, and 50 male–non-black-hair samples. For this dataset, we aim to reconstruct images with male and black hair.

Results. Figures 9, 8, 4 illustrate the reconstructed images for different methods. As we can see from the results, the reconstructed images are different from the original images.

E. Proof of Main Results

E.1. Proof of Theorem 5.1

We first introduce the definition of Rényi Differential Privacy (RDP) [68]. In our privacy analysis, we first use RDP to account for privacy loss and then translate the RDP guarantee to (ϵ, δ)-DP guarantee.

Definition E.1 (RDP)**.** *A randomized mechanism $\mathcal{A}$ satisfies (γ, ρ)-Rényi differential privacy with $\gamma > 1$ and $\rho > 0$ if for adjacent datasets $D, D' \in \mathcal{D}$ differing by one element, $D_\gamma\big(\mathcal{A}(D)||\mathcal{A}(D')\big) = \log \mathbb{E}_{\mathcal{A}(D')}\big(\mathcal{A}(D)/\mathcal{A}(D')\big)^\gamma/(1-\gamma) \leq \rho$.*

Given a privacy guarantee in terms of RDP, we can transfer it to (ϵ, δ)-DP using the following lemma [68].

Lemma E.2 (RDP to DP)**.** *If a randomized mechanism $\mathcal{A}$ satisfies (γ, ρ)-RDP, then $\mathcal{A}$ satisfies $(\rho + \log(1/\delta)/(\gamma - 1), \delta)$-DP for all $\delta \in (0, 1)$.*

To ensure the RDP, we need the following result for Gaussian mechanism [68].

Lemma E.3 (Gaussian Mechanism)**.** *Given a function q, the Gaussian Mechanism $\mathcal{A} = q(D) + z$, where $z \sim N(0, \sigma^2\mathbf{I})$, satisfies $(\gamma, \gamma S_2^2/(2\sigma^2))$-RDP, where S_2 is the ℓ_2-sensitivity of q and is defined as $S_2 = \sup_{D,D'} \|q(D) - q(D')\|_2$ for two adjacent datasets D, D' differing by one element.*

Now, we are ready to provide the privacy guarantees of our method.

Proof of Theorem 5.1. According to Algorithm 1, the training of the shared global denoiser z_w is based on the noisy dataset $\tilde{D} = \{\tilde{D}_m\}_{m\in[M]}$. For each data point $\tilde{x}_0^{i,m}$ in $\tilde{D}$, it is generated by adding random Gaussian noise to the original data as follows: $\tilde{x}_0^{i,m} = \sqrt{\bar{\alpha}_{t_0}} x_0^{i,m} + \sqrt{1-\bar{\alpha}_{t_0}} z$ (see line 7 in Algorithm 1), where $z \sim \mathcal{N}(0, \mathbf{I})$. Therefore, we only need to prove the privacy guarantee for $\sqrt{\bar{\alpha}_{t_0}} x_0^{i,m}$ under Gaussian mechanism. By Lemma E.3, we have that each data point in $\tilde{D}$ is $(\gamma, \gamma\tau)$-RDP with $\tau = 2\bar{\alpha}_{t_0} C^2/(1-\bar{\alpha}_t)$. According to Lemma E.2, it is (ϵ, δ)-DP with $\epsilon = \gamma\tau + \log(1/\delta)/(\gamma - 1)$. Therefore, we can choose $\gamma = 1 + \sqrt{\log(1/\delta)/\tau}$ to get the smallest $\epsilon = \tau + 2\sqrt{\log(1/\delta)\tau}$. By plugging the value of τ, it is $\left(\frac{2\bar{\alpha}_{t_0} C^2}{1-\bar{\alpha}_t} + C\sqrt{\frac{8\bar{\alpha}_{t_0}\log(1/\delta)}{1-\bar{\alpha}_{t_0}}}, \delta\right)$-DP. Since the guarantee is for each data point, we prove the same level of LDP for the creation of $\tilde{D}$. As a result, by the post processing property of differential privacy, the shared global denoiser z_w is $\left(\frac{2\bar{\alpha}_{t_0} C^2}{1-\bar{\alpha}_t} + C\sqrt{\frac{8\bar{\alpha}_{t_0}\log(1/\delta)}{1-\bar{\alpha}_{t_0}}}, \delta\right)$-LDP. □

E.2. Proof of Theorem 5.2

This section compares the 2-Wasserstein distance between the population distribution and the distributions learned by pure local training (i.e., no collaboration) and our personalized framework.

Pure Local Training. We first recall that due to Chen et al. [12], the limiting distribution (i.e., as the number of denoising steps approaches infinity) of the data generated on client m conditioned on label $y = k \in [K]$ is given by,

$$\lim_{T\to\infty} z_0^m | y = k \sim \mathcal{N}(\hat{\mu}_k^m, \hat{\Sigma}_k^m) \ ,$$

where

$$\hat{\mu}_k^m := \frac{1}{n_k^m} \sum_{i\in S_k^m} x_i^m \ ,$$

is the empirical mean of all the data-points with label k on client m denoted by the set S_k^m and

$$\hat{\Sigma}_k^m := \frac{1}{n_k^m} \sum_{i\in S_k^m} x_i^m (x_i^m)^T - \hat{\mu}_k^m (\hat{\mu}_k^m)^T \ ,$$

is the corresponding empirical co-variance matrix. One can already note that a major limitation of pure local training is that if there is a class k on client m, with very few samples, then the estimation of the target Gaussian could be very bad in the limit, especially when the number of samples is small. In particular, note that, we have

$$\mathbb{E}\left[\|\hat{\mu}_k^m - \mu_k\|_2^2\right] = \frac{1}{(n_k^m)^2} \sum_{i\in S_k^m} \mathbb{E}\left[\|x_i^m - \mu_k\|_2^2\right] = \frac{d}{n_k^m} \ , \tag{14}$$

where we use the fact that the data point for every class k on each client is sampled independently and identically from $\mathcal{N}(\mu_k, I_d)$, and the expectation is taken over the randomness of samples. Similarly, we have

$$\mathbb{E}\|\hat{\Sigma}_k^m - I_d\|_2^2 = O\left(\frac{d}{n_k^m}\right) \ .$$

In addition, we have with probability $1-\delta$,

$$\|\hat{\Sigma}_k^m - I_d\|_2^2 \le \frac{c_1 d \log(1/\delta)}{n_k^m} \ ,$$

where c_1 is some constant. Therefore, as long as $n_k^m \geq c_1 d \log(1/\delta)$, we have $\|\hat{\Sigma}_k^m - I_d\|_2 \leq 1$. In addition, we have

$$\|(\hat{\Sigma}_k^m)^{1/2} - I_d\|_2 \leq \frac{\|\hat{\Sigma}_k^m - I_d\|_2}{1 + \sqrt{1 - \|\hat{\Sigma}_k^m - I_d\|_2}} \leq \|\hat{\Sigma}_k^m - I_d\|_2 \ .$$

Therefore, we have the 2-Wasserstein distance as

$$\mathbb{E}\left[W_2^2\big(\mathcal{N}(\hat{\mu}_k^m, \hat{\Sigma}_k^m); \mathcal{N}(\mu_k, I_d)\big)\right] = \mathbb{E}\|\hat{\mu}_k^m - \mu_k\|_2^2 + \mathbb{E}\|(\hat{\Sigma}_k^m)^{1/2} - I_d\|_F^2 = O\left(\frac{d^2}{n_k^m}\right) \ .$$

Personalized Training. Recall that in the personalized setting, the clients each have a denoising model which is trained on the client dataset $\{S_k^m\}_{k\in[K]}$ for $m \in [M]$. Furthermore, the global model is trained by combining the noisy data of all the clients. In particular, we train the global model using the dataset $\tilde{S}_k := \left\{\{\tilde{x}_i^m\}_{i\in S_k^m}\right\}_{m\in[M]}$ for class $k \in [K]$, where we define the noisy data-point as

$$\tilde{x}_i^m := e^{-t_0} \cdot x_i^m + \sqrt{1 - e^{-2t_0}} \cdot \xi_i^m, \quad \forall k \in [K],\ m \in [M],\ i \in S_k^m \ .$$

Again, following the result of Chen et al. [12], we can conclude the limiting distribution of the sample generated by the global model for class k is given as,

$$\lim_{T\to\infty} z_0^g | y = k \sim \mathcal{N}(\tilde{\mu}_k, \tilde{\Sigma}_k) \ ,$$

where

$$\tilde{\mu}_k := \frac{1}{N_k} \sum_{m\in[M], i\in\tilde{S}_k^m} \tilde{x}_i^m \ ,$$

is the empirical mean of all the noisy data-points (denoting $N_k = \sum_{m\in[M]} n_k^m$) with label k and

$$\tilde{\Sigma}_k := \frac{1}{N_k} \sum_{m\in[M], i\in S_k^m} \tilde{x}_i^m (\tilde{x}_i^m)^T - \tilde{\mu}_k (\tilde{\mu}_k)^T \ ,$$

is the corresponding empirical covariance matrix. Each client will independently sample a z_0^g for a specific class k and then use its learned model—from pure local training—with only t_0 steps of de-noising to sample from the target Gaussian $\mathcal{N}(\mu_k, I)$. This is where we need to be careful; we can not use the off-the-shelf asymptotic result from Chen et al. [12] anymore, as the terminal noise distribution for each client is not standard Gaussian but $\mathcal{N}(\tilde{\mu}_k, \tilde{\Sigma}_k)$. Towards this end, we will directly compute the learned mean and variance for our personalized model, which is motivated by the proofs of Lemma 3 from Chen et al. [12].

We will focus on a client $m \in [M]$ and class $k \in [K]$. We assume the client's sample covariance matrix is $\hat{\Sigma}_k^m$ with the singular value decomposition as

$$\hat{\Sigma}_k^m = \sum_{i\in[d]} \lambda_{k,i}^m u_{k,i}^m (u_{k,i}^m)^T \ ,$$

where

$$\lambda_{\max} \geq \lambda_{k,1}^m \geq \lambda_{k,2}^m \geq \cdots \geq \lambda_{k,d}^m \geq \lambda_{\min} \ .$$

Note that, by the standard SubGaussian concentration [93], we have with probability $1-\delta$, $1 - c_2\sqrt{d\log(1/\delta)}/\sqrt{n_k^m} \leq \lambda_{\min} \leq \lambda_{\max} \leq 1 + c_2\sqrt{d\log(1/\delta)}/\sqrt{n_k^m}$ for some constant c_2. Therefore, as long as $n_k^m \geq 4c_2^2 d\log(1/\delta)$, we have the all the eigenvalues $\{\lambda_{k,i}^m\}_{i\in[d]}$ is in the range of $[0.5, 1.5]$.

Given the SVD and using the Euler-Maruyama discretization (c.f., Lemma 3, (30) in Chen et al. [12]), we can write a recursion for $\langle u_{k,i}^m, z_{jh}^m\rangle$ where z_{jh}^m is the partially de-noised sample for $h = \frac{t_0}{D}$ and $j \in [0, D]$ implying $jh \in [0, t_0]$. The recursion looks as follows for $j \in [0, D-1]$,

$$\langle u_{k,i}^m, z_{jh}^m\rangle$$

$$= \langle u^m_{k,i}, z^m_{(j+1)h} \rangle + \left(1 - \frac{1}{\sigma^2_{t_0-jh} + r^2_{t_0-jh}\lambda^m_{k,i}}\right) \langle u^m_{k,i}, z^m_{(j+1)h} h \rangle + \frac{r_{t_0-jh}}{\sigma^2_{t_0-jh} + r^2_{t_0-jh}\lambda^m_{k,i}} \langle u^m_{k,i}, \hat{\mu}^m_k h \rangle$$
$$= \left(1 + \left(1 - \frac{1}{\sigma^2_{t_0-jh} + r^2_{t_0-jh}\lambda^m_{k,i}}\right) h\right) \langle u^m_{k,i}, z^m_{(j+1)h} h \rangle + \frac{r_{t_0-jh}}{\sigma^2_{t_0-jh} + r^2_{t_0-jh}\lambda^m_{k,i}} \langle u^m_{k,i}, \hat{\mu}^m_k h \rangle \ .$$

Taking the conditional expectation above conditioned on all the data across machines X, as well as all the noise that is added for the global model $\tilde{X}$ and unrolling the recursion until $j = 0$ gives us the following,

$$\mathbb{E}\left[\langle u^m_{k,i}, z^m_0 \rangle | X, \tilde{X}\right]$$
$$= \prod_{j=0}^{D-1} \left(1 + \left(1 - \frac{1}{\sigma^2_{t_0-jh} + r^2_{t_0-jh}\lambda^m_{k,i}}\right) h\right) \mathbb{E}\left[\langle u^m_{k,i}, z^m_{Dh} \rangle | X, \tilde{X}\right]$$
$$+ \sum_{j=0}^{D-1} \frac{r_{t_0-jh}}{\sigma^2_{t_0-jh} + r^2_{t_0-jh}\lambda^m_{k,i}} \mathbb{E}\left[\langle u^m_{k,i}, \hat{\mu}^m_k \rangle | X, \tilde{X}\right] h \prod_{l=j+1}^{D-1} \left(1 + \left(1 - \frac{1}{\sigma^2_{t_0-lh} + r^2_{t_0-lh}\lambda^m_{k,i}}\right) h\right) \ .$$

Note that $u^m_{k,i}$ is measurable under the sigma algebra generated by the entire data across all machines and the randomness used to generate the noise data for the global model. On the other hand, $z^m_{Dh} = z^g_0$ depends on the extra randomness required to sampled from $\mathcal{N}(\tilde{\mu}_k, \tilde{\Sigma}_k)$. Thus, we get the following,

$$\mathbb{E}\left[\langle u^m_{k,i}, z^m_0 \rangle | X, \tilde{X}\right]$$
$$= \prod_{j=0}^{D-1} \left(1 + \left(1 - \frac{1}{\sigma^2_{t_0-jh} + r^2_{t_0-jh}\lambda^m_{k,i}}\right) h\right) \langle u^m_{k,i}, \tilde{\mu}_k \rangle$$
$$+ \sum_{j=0}^{D-1} \frac{r_{t_0-jh}}{\sigma^2_{t_0-jh} + r^2_{t_0-jh}\lambda^m_{k,i}} \langle u^m_{k,i}, \hat{\mu}^m_k \rangle h \prod_{l=j+1}^{D-1} \left(1 + \left(1 - \frac{1}{\sigma^2_{t_0-lh} + r^2_{t_0-lh}\lambda^m_{k,i}}\right) h\right) \ .$$

Now we will send $h \to 0$, and use the approximation $1 + hx \approx e^{hx}$ which allows us to convert the sums and products into appropriate integrals as follows,

$$\mathbb{E}\left[\langle u^m_{k,i}, z^m_0 \rangle | X, \tilde{X}\right]$$
$$= \exp \int_0^{t_0} \left(1 - \frac{1}{\sigma^2_{t_0-t} + r^2_{t_0-t}\lambda^m_{k,i}}\right) dt \langle u^m_{k,i}, \tilde{\mu}_k \rangle$$
$$+ \int_0^{t_0} \frac{r_{t_0-t}}{\sigma^2_{t_0-t} + r^2_{t_0-t}\lambda^m_{k,i}} \left(\exp \int_t^{t_0} \left(1 - \frac{1}{\sigma^2_{t_0-s} + r^2_{t_0-s}\lambda^m_{k,i}}\right) ds\right) \langle u^m_{k,i}, \hat{\mu}^m_k \rangle dt$$
$$= \exp\left(\frac{1}{2} \ln\left(\frac{\lambda^m_{k,i}}{1 + (\lambda^m_{k,i} - 1)e^{-2t_0}}\right)\right) \langle u^m_{k,i}, \tilde{\mu}_k \rangle$$
$$+ \int_0^{t_0} \frac{e^{t-t_0}}{1 + e^{2(t-t_0)}(\lambda^m_{k,i} - 1)} \left(\exp\left(\frac{1}{2} \ln\left(\frac{\lambda^m_{k,i}}{1 + (\lambda^m_{k,i} - 1)e^{2(t-t_0)}}\right)\right)\right) \langle u^m_{k,i}, \hat{\mu}^m_k \rangle dt$$
$$= \sqrt{\frac{\lambda^m_{k,i}}{1 + (\lambda^m_{k,i} - 1)e^{-2t_0}}} \langle u^m_{k,i}, \tilde{\mu}_k \rangle + \int_0^{t_0} \frac{e^{t-t_0}\sqrt{\lambda^m_{k,i}}}{\left(1 + e^{2(t-t_0)}(\lambda^m_{k,i} - 1)\right)^{3/2}} \langle u^m_{k,i}, \hat{\mu}^m_k \rangle dt$$
$$= \sqrt{\frac{\lambda^m_{k,i}}{1 + (\lambda^m_{k,i} - 1)e^{-2t_0}}} \langle u^m_{k,i}, \tilde{\mu}_k \rangle + \left(1 - \sqrt{\frac{\lambda^m_{k,i} e^{-2t_0}}{1 + (\lambda^m_{k,i} - 1)e^{-2t_0}}}\right) \langle u^m_{k,i}, \hat{\mu}^m_k \rangle \ .$$

Now multiplying this expression by $u_{k,i}^m$ and then summing both sides accross $i \in [d]$ gives us the following,

$$\begin{aligned}\mathbb{E}\left[z_0^m|X,\tilde{X}\right] &= \sum_{i\in[d]} \mathbb{E}\left[\langle u_{k,i}^m, z_0^m\rangle u_{k,i}^m|X,\tilde{X}\right] \\ &= \sum_{i\in[d]} \sqrt{\frac{\lambda_{k,i}^m}{1+(\lambda_{k,i}^m-1)e^{-2t_0}}}\langle u_{k,i}^m, \tilde{\mu}_k\rangle u_{k,i}^m \\ &\quad + \sum_{i\in[d]}\left(1-\sqrt{\frac{\lambda_{k,i}^m e^{-2t_0}}{1+(\lambda_{k,i}^m-1)e^{-2t_0}}}\right)\langle u_{k,i}^m, \hat{\mu}_k^m\rangle u_{k,i}^m \ ,\end{aligned}$$

where in the first line, we use the fact that $\{u_{k,i}^m\}_{i\in[d]}$ is an orthogonal basis of $\mathbb{R}^d$. We recall now that $\tilde{\mu}_k$ depends on two sources of randomness: of sampling the data, and of all the noise added to the training data of the global model. By plugging the expression of $\tilde{\mu}_k$, i.e.,

$$\tilde{\mu}_k = \frac{1}{N_k}\sum_{m\in[M], i\in S_k^m} e^{-t_0}\left(x_i^m + \sqrt{\frac{1-e^{-2t_0}}{e^{-2t_0}}}\cdot\xi_i^m\right)$$

into the above equation, we can get

$$\begin{aligned}&\mathbb{E}\left[z_0^m|X,\tilde{X}\right] \\ &= \sum_{i\in[d]}\sqrt{\frac{\lambda_{k,i}^m e^{-2t_0}}{1+(\lambda_{k,i}^m-1)e^{-2t_0}}}\langle u_{k,i}^m, \frac{1}{N_k}\sum_{m\in[M],i\in S_k^m} x_i^m\rangle u_{k,i}^m + \sum_{i\in[d]}\left(1-\sqrt{\frac{\lambda_{k,i}^m e^{-2t_0}}{1+(\lambda_{k,i}^m-1)e^{-2t_0}}}\right)\langle u_{k,i}^m, \hat{\mu}_k^m\rangle u_{k,i}^m \\ &\quad + \sum_{i\in[d]}\sqrt{\frac{\lambda_{k,i}^m e^{-2t_0}}{1+(\lambda_{k,i}^m-1)e^{-2t_0}}}\langle u_{k,i}^m, \frac{\gamma}{N_k}\sum_{m\in[M],i\in S_k^m}\xi_i^m\rangle u_{k,i}^m \\ &= \sum_{i\in[d]}\rho_i^m\langle u_{k,i}^m, \frac{1}{N_k}\sum_{m\in[M],i\in S_k^m}(x_i^m+\gamma\xi_i^m)\rangle u_{k,i}^m + \sum_{i\in[d]}(1-\rho_i^m)\langle u_{k,i}^m, \hat{\mu}_k^m\rangle u_{k,i}^m \ ,\end{aligned}$$

where $\gamma = \sqrt{\frac{1-e^{-2t_0}}{e^{-2t_0}}}$, $\rho_i^m = \sqrt{\frac{\lambda_{k,i}^m e^{-2t_0}}{1+(\lambda_{k,i}^m-1)e^{-2t_0}}}$. Recall that $\hat{\mu}_k = \frac{1}{N_k}\sum_{m\in[M],i\in S_k^m} x_i^m$, $\bar{\xi} = \frac{1}{N_k}\sum_{m\in[M],i\in S_k^m}\xi_i^m$, we have

$$\begin{aligned}\mathbb{E}\left[z_0^m|X,\tilde{X}\right] &= \sum_{i\in[d]}\rho_i^m u_{k,i}^m(u_{k,i}^m)^\top(\hat{\mu}_k+\gamma\bar{\xi}) + \sum_{i\in[d]}(1-\rho_i^m)u_{k,i}^m(u_{k,i}^m)^\top\hat{\mu}_k^m \\ &= A_k^m(\hat{\mu}_k+\gamma\bar{\xi}) + (I-A_k^m)\hat{\mu}_k^m \ ,\end{aligned} \tag{15}$$

where we have

$$A_k^m = U_k^m diag(\rho_1^m,\ldots,\rho_d^m)(U_k^m)^\top \ ,$$

where each column of U_k^m is $u_{k,i}^m$. Therefore, we have

$$\mu_{k,per}^m = \mathbb{E}\left[z_0^m|X\right] = A_k^m\hat{\mu}_k + (I-A_k^m)\hat{\mu}_k^m \ . \tag{16}$$

So we can bound the following

$$\begin{aligned}\left\|A_k^m\hat{\mu}_k + (I-A_k^m)\hat{\mu}_k^m - u_k\right\|_2 &\le \|A_k^m\|_2\|\hat{\mu}_k-u_k\|_2 + \|(I-A_k^m)\|_2\|\hat{\mu}_k^m-\mu_k\|_2 \\ &\le \sqrt{\frac{\lambda_{\max}}{\gamma^2+\lambda_{\max}}}\|\hat{\mu}_k-\mu_k\|_2 + \left(1-\sqrt{\frac{\lambda_{\min}}{\gamma^2+\lambda_{\min}}}\right)\|\hat{\mu}_k^m-\mu_k\|_2 \\ &\le \sqrt{\frac{3}{2\gamma^2+3}}\|\hat{\mu}_k-\mu_k\|_2 + \left(1-\sqrt{\frac{1}{2\gamma^2+1}}\right)\|\hat{\mu}_k^m-\mu_k\|_2 \ ,\end{aligned} \tag{17}$$

where the second line is due to the definition of A_k^m, ρ_i^m, and $\gamma^2(\rho_i^m)^2 = \lambda_i(1-(\rho_i^m)^2)$. The last line comes from the range of eigenvalues.

Following the similar argument, we can show that

$$Var\left[z_0^m|X,\tilde{X}\right] = \sum_{i\in[d]} \frac{\lambda_{k,i}^m}{1+(\lambda_{k,i}^m-1)e^{-2t_0}}(u_{k,i}^m)^\top \tilde{\Sigma}_k u_{k,i}^m \cdot u_{k,i}^m(u_{k,i}^m)^\top \ . \tag{18}$$

Recall that, we have

$$\begin{aligned}
\tilde{\Sigma}_k &= \frac{1}{N_k}\sum_{m\in[M],i\in S_k^m} \tilde{x}_i^m(\tilde{x}_i^m)^T - \tilde{\mu}_k(\tilde{\mu}_k)^T \\
&= \frac{e^{-2t_0}}{N_k}\sum_{m\in[M],i\in S_k^m} x_i^m(x_i^m)^T - e^{-2t_0}\hat{\mu}_k(\hat{\mu}_k)^T + \frac{1-e^{-2t_0}}{N_k}\sum_{m\in[M],i\in S_k^m}\xi_i^m(\xi_i^m)^T - (1-e^{-2t_0})\bar{\xi}\bar{\xi}^\top \\
&\quad + \frac{e^{-t_0}\sqrt{1-e^{-2t_0}}}{N_k}\sum_{m\in[M],i\in S_k^m}\left(x_i^m(\xi_i^m)^T + \xi_i^m(x_i^m)^T\right) - e^{-t_0}\sqrt{1-e^{-2t_0}}\left(\hat{\mu}_k\bar{\xi}^\top + \bar{\xi}\hat{\mu}_k^\top\right) \ ,
\end{aligned}$$

which can be further simplified as

$$\begin{aligned}
\tilde{\Sigma}_k &= \frac{e^{-2t_0}}{N_k}\sum_{m\in[M],i\in S_k^m}\left(x_i^m(x_i^m)^T - \hat{\mu}_k(\hat{\mu}_k)^T + \gamma^2\xi_i^m(\xi_i^m)^T - \gamma^2\bar{\xi}\bar{\xi}^\top\right) \\
&\quad + \frac{e^{-t_0}\sqrt{1-e^{-2t_0}}}{N_k}\sum_{m\in[M],i\in S_k^m}\left(x_i^m(\xi_i^m)^T + \xi_i^m(x_i^m)^T - \hat{\mu}_k\bar{\xi}^\top - \bar{\xi}\hat{\mu}_k^\top\right) \\
&= \frac{e^{-2t_0}}{N_k}\sum_{m\in[M],i\in S_k^m}\left(x_i^m(x_i^m)^T - \hat{\mu}_k(\hat{\mu}_k)^T + \gamma^2\xi_i^m(\xi_i^m)^T - \gamma^2\bar{\xi}\bar{\xi}^\top\right) + R \ ,
\end{aligned}$$

where we use R to denote the cross terms (this term will disappear in the following proofs). Therefore, according to equation 18, we have

$$\begin{aligned}
Var\left[z_0^m|X,\tilde{X}\right] &= \sum_{i\in[d]}(\rho_i^m)^2(u_{k,i}^m)^\top\left(\hat{\Sigma}_k + \gamma^2\hat{\Sigma}_\xi\right)u_{k,i}^m \cdot u_{k,i}^m(u_{k,i}^m)^\top \\
&\quad + \sum_{i\in[d]}(e^{t_0}\rho_i^m)^2(u_{k,i}^m)^\top R u_{k,i}^m \cdot u_{k,i}^m(u_{k,i}^m)^\top \ .
\end{aligned}$$

By law of total variance, we have

$$\begin{aligned}
Var\left[z_0^m|X\right] &= \mathbb{E}Var\left[z_0^m|X,\tilde{X}\right] + Var\mathbb{E}\left[z_0^m|X,\tilde{X}\right] \\
&= \sum_{i\in[d]}(\rho_i^m)^2(u_{k,i}^m)^\top\left(\hat{\Sigma}_k + \gamma^2\frac{N_k-1}{N_k}I\right)u_{k,i}^m \cdot u_{k,i}^m(u_{k,i}^m)^\top \\
&\quad + Var\left[\sum_{i\in[d]}\gamma\rho_i^m u_{k,i}^m(u_{k,i}^m)^\top\bar{\xi}\right] \\
&= \sum_{i\in[d]}(\rho_i^m)^2(u_{k,i}^m)^\top\left(\hat{\Sigma}_k + \gamma^2\frac{N_k-1}{N_k}I\right)u_{k,i}^m \cdot u_{k,i}^m(u_{k,i}^m)^\top \\
&\quad + \gamma^2\sum_{i\in[d]}(\rho_i^m)^2\frac{1}{N_k} \cdot u_{k,i}^m(u_{k,i}^m)^\top \\
&= \sum_{i\in[d]}(\rho_i^m)^2(u_{k,i}^m)^\top\hat{\Sigma}_k u_{k,i}^m \cdot u_{k,i}^m(u_{k,i}^m)^\top + \gamma^2\sum_{i\in[d]}(\rho_i^m)^2 u_{k,i}^m(u_{k,i}^m)^\top \ ,
\end{aligned}$$

where the second line uses the fact that ξ_i follows standard Gaussian, expectation of the sample covariance matrix, and equation 15.

We can further simplify it by plugging the value of ρ_i^m and γ

$$\begin{aligned}
Var\left[z_0^m|X\right] &= \sum_{i\in[d]} \frac{\lambda_i e^{-2t_0}}{1+(\lambda_i-1)e^{-2t_0}} (u_{k,i}^m)^\top \hat{\Sigma}_k u_{k,i}^m \cdot u_{k,i}^m (u_{k,i}^m)^\top + \sum_{i\in[d]} \frac{1-e^{-2t_0}}{1+(\lambda_i-1)e^{-2t_0}} \lambda_i u_{k,i}^m (u_{k,i}^m)^\top \\
&= \sum_{i\in[d]} \left(1 - \frac{1-e^{-2t_0}}{1+(\lambda_i-1)e^{-2t_0}}\right) (u_{k,i}^m)^\top \hat{\Sigma}_k u_{k,i}^m \cdot u_{k,i}^m (u_{k,i}^m)^\top + \sum_{i\in[d]} \frac{1-e^{-2t_0}}{1+(\lambda_i-1)e^{-2t_0}} \lambda_i u_{k,i}^m (u_{k,i}^m)^\top \\
&= \sum_{i\in[d]} (\rho_i^m)^2 (u_{k,i}^m)^\top \hat{\Sigma}_k u_{k,i}^m \cdot u_{k,i}^m (u_{k,i}^m)^\top + \sum_{i\in[d]} (1-(\rho_i^m)^2) \lambda_i u_{k,i}^m (u_{k,i}^m)^\top \\
&= (A_k^m)^2 \bar{\Sigma}_k^m + (I_d - (A_k^m)^2)\hat{\Sigma}_k^m \quad ,
\end{aligned}$$

where we have A_k^m as before to be

$$A_k^m = U_k^m diag(\rho_1^m, \ldots, \rho_d^m)(U_k^m)^\top \quad ,$$

and $\bar{\Sigma}_k^m$ as

$$\bar{\Sigma}_k^m = U_k^m diag\big((u_{k,1}^m)^\top \hat{\Sigma}_k u_{k,1}^m, \ldots, (u_{k,d}^m)^\top \hat{\Sigma}_k u_{k,d}^m\big)(U_k^m)^\top \quad .$$

Therefore, we have

$$\begin{aligned}
\left\|(A_k^m)^2\bar{\Sigma}_k^m + (I_d - (A_k^m)^2)\hat{\Sigma}_k^m - I_d\right\|_2 &\le \|(A_k^m)^2\|_2 \|\bar{\Sigma}_k^m - I_d\|_2 + \|(I_d - (A_k^m)^2)\|_2 \|\hat{\Sigma}_k^m - I_d\|_2 \\
&\le \frac{\lambda_{\max}}{\gamma^2+\lambda_{\max}} \|\bar{\Sigma}_k^m - I_d\|_2 + \left(1 - \frac{\lambda_{\min}}{\gamma^2+\lambda_{\min}}\right)\|\hat{\Sigma}_k^m - I_d\|_2 \\
&\le \frac{\lambda_{\max}}{\gamma^2+\lambda_{\max}} \|\hat{\Sigma}_k - I_d\|_2 + \left(1 - \frac{\lambda_{\min}}{\gamma^2+\lambda_{\min}}\right)\|\hat{\Sigma}_k^m - I_d\|_2 \quad ,
\end{aligned}$$

where the second line follows the same argument as in equation 17. The last line is due to the definition of $\bar{\Sigma}_k^m$. Note that we have

$$\mathbb{E}\|\hat{\Sigma}_k - I_d\|_2^2 = O\left(\frac{d}{N_k}\right) \quad ,$$

and as mentioned before, we have

$$\mathbb{E}\|\hat{\Sigma}_k^m - I_d\|_2^2 = O\left(\frac{d}{n_k^m}\right),$$

In addition, we have with probability $1-\delta$

$$\|\hat{\Sigma}_k - I_d\|_2 \le c_3 \sqrt{\frac{d\log(1/\delta)}{N_k}} \quad ,$$

and as mentioned before, with probability $1-\delta$, we have

$$\|\hat{\Sigma}_k^m - I_d\|_2 \le c_2 \sqrt{\frac{d\log(1/\delta)}{n_k^m}} \quad ,$$

where c_2, c_3 are some constants. Therefore, as long as $N_k \ge c_3 d\log(1/\delta)$, $n_k^m \ge c_2 d\log(1/\delta)$, we have both bounds to be less than 1. In addition, since we have $\Sigma_{k,per}^m = (A_k^m)^2\bar{\Sigma}_k^m + (I_d - (A_k^m)^2)\hat{\Sigma}_k^m$ to be the covariance matrix, we have

$$\|(\Sigma_{k,per}^m)^{1/2} - I_d\|_2 \le \frac{\|\Sigma_{per} - I_d\|_2}{1+\sqrt{1-\|\Sigma_{per}-I_d\|_2}}$$

$$
\begin{aligned}
&\leq \|\Sigma_{per} - I_d\|_2 \\
&\leq \frac{\lambda_{\max}}{\gamma^2 + \lambda_{\max}} \|\hat{\Sigma}_k - I_d\|_2 + \left(1 - \frac{\lambda_{\min}}{\gamma^2 + \lambda_{\min}}\right) \|\hat{\Sigma}_k^m - I_d\|_2 \\
&\leq \frac{3}{2\gamma^2 + 3} \|\hat{\Sigma}_k - I_d\|_2 + \left(1 - \frac{1}{2\gamma^2 + 1}\right) \|\hat{\Sigma}_k^m - I_d\|_2 \ . \qquad (19)
\end{aligned}
$$

Therefore, we have 2-Wasserstein distance as

$$
\begin{aligned}
&W_2^2\big(N(\mu_{k,per}^m, \Sigma_{k,per}^m); N(\mu_k, I_d)\big) \\
&\quad = \|\mu_{k,per}^m - \mu_k\|_2^2 + \|(\Sigma_{k,per}^m)^{1/2} - I_d\|_F^2 \\
&\quad \leq \frac{6}{2\gamma^2 + 3} \|\hat{\mu}_k - \mu_k\|_2^2 + 2\left(1 - \frac{1}{2\gamma^2 + 1}\right) \|\hat{\mu}_k^m - \mu_k\|_2^2 + \frac{6d}{2\gamma^2 + 3} \|\hat{\Sigma}_k - I_d\|_2^2 + 2d\left(1 - \frac{1}{2\gamma^2 + 1}\right) \|\hat{\Sigma}_k^m - I_d\|_2^2 \\
&\quad \leq 2\left(\frac{3}{2\gamma^2 + 3}\big(\|\hat{\mu}_k - \mu_k\|_2^2 + d\|\hat{\Sigma}_k - I_d\|_2^2\big) + \Big(1 - \frac{3}{2\gamma^2 + 3}\Big)\big(\|\hat{\mu}_k^m - \mu_k\|_2^2 + d\|\hat{\Sigma}_k^m - I_d\|_2^2\big)\right) \ .
\end{aligned}
$$

Taking expectation with respect to the samples, we have

$$
\mathbb{E}\left[W_2\big(N(\mu_{k,per}^m, \Sigma_{k,per}^m); N(\mu_k, I_d)\big)^2\right] = O\left(\rho(\gamma)\frac{d^2}{N_k} + \big(1 - \rho(\gamma)\big)\frac{d^2}{n_k^m}\right) \ ,
$$

where $\rho(\gamma) = \frac{3}{2\gamma^2+3}$ with $\gamma^2 = \frac{1-e^{-2t_0}}{e^{-2t_0}}$.

Finally, replace e^{-2t} with $\bar{\alpha}_t$ and in our case, and note the added noise variance $\sigma^2 = \gamma^2$, we can get

$$
\mathbb{E}\left[W_2^2\big(\mathcal{N}(\mu_{k,per}^m, \Sigma_{k,per}^m); \mathcal{N}(\mu_k, I_d)\big)\right] = O\left(\frac{2}{2 + 3\sigma^2} \cdot \frac{d^2}{N_k} + \frac{3\sigma^2}{2 + 3\sigma^2} \cdot \frac{d^2}{n_k^m}\right) \ ,
$$

which completes the proofs.

Let's now try to bound the following

$$
\begin{aligned}
&\mathbb{E}\left[W_2^2\big(\mathcal{N}(\hat{\mu}_k^m, \hat{\Sigma}_k^m); \mathcal{N}(\mu_k, I_d)\big) - W_2^2\big(\mathcal{N}(\hat{\mu}_k^m, \hat{\Sigma}_k^m); \mathcal{N}(\mu_k, I_d)\big)\right] \\
&\quad = \mathbb{E}\|\hat{\mu}_k^m - \mu_k\|_2^2 + \mathbb{E}\|(\hat{\Sigma}_k^m)^{1/2} - I_d\|_F^2 - \mathbb{E}\|\mu_{k,per}^m - \mu_k\|_2^2 - \mathbb{E}\|(\Sigma_{k,per}^m)^{1/2} - I_d\|_F^2 \ .
\end{aligned}
$$

We have

$$
\mathbb{E}\left[\|\hat{\mu}_k^m - \mu_k\|_2^2\right] = \frac{1}{(n_k^m)^2} \sum_{i \in S_k^m} \mathbb{E}\left[\|x_i^m - \mu_k\|_2^2\right] = \frac{d}{n_k^m} \ , \qquad (20)
$$

By Taylor expansion of square root of each eigenvalues, we have

$$
\|(\hat{\Sigma}_k^m)^{1/2} - I_d\|_F^2 = \frac{1}{4}\|\hat{\Sigma}_k^m - I_d\|_F^2 - \frac{1}{8}\sum_{i=1}^d (\lambda_{k,i}^m - 1)^3 + o(\sum_{i=1}^d (\lambda_{k,i}^m - 1)^3) \ ,
$$

which gives us

$$
\|(\hat{\Sigma}_k^m)^{1/2} - I_d\|_F^2 \geq \frac{1}{4}\|\hat{\Sigma}_k^m - I_d\|_F^2 - \frac{1}{8}\|\hat{\Sigma}_k^m - I_d\|_2 \cdot \|\hat{\Sigma}_k^m - I_d\|_F^2 + o(\sum_{i=1}^d (\lambda_{k,i}^m - 1)^3) \ .
$$

In addition, we have

$$
\mathbb{E}\left[\|\hat{\Sigma}_k^m - I_d\|_F^2\right] \geq \frac{d^2}{n_k^m} \ ,
$$

and

$$\|\hat{\Sigma}_k^m - I_d\|_2 = O\left(\frac{\sqrt{d\log(1/\delta)}}{\sqrt{n_k^m}}\right) .$$

Therefore, we have

$$\begin{aligned}\mathbb{E}\left[\|(\hat{\Sigma}_k^m)^{1/2} - I_d\|_F^2\right] &\geq \frac{d^2}{4n_k^m} - O\left(\frac{d^2}{n_k^m}\cdot\frac{d^{\frac{1}{2}}\log^{3/2}(1/\delta)}{(n_k^m)^{\frac{1}{2}}}\right) \\ &= \frac{d^2}{4n_k^m} - O\left(\frac{d^2}{n_k^m}\cdot r\right) ,\end{aligned} \tag{21}$$

where $r = \sqrt{d\log^{3/2}/n_k^m}$. Following the similar argument, we have

$$\|(\Sigma_{per})^{1/2} - I_d\|_F^2 \leq \frac{1}{4}\|\Sigma_{per} - I_d\|_F^2 + O(\|\Sigma_{per} - I_d\|_2 \cdot \|\Sigma_{per} - I_d\|_F^2) .$$

Therefore, we can get

$$\|(\Sigma_{per})^{1/2} - I_d\|_F^2 \leq \frac{d}{4}\|\Sigma_{per} - I_d\|_2^2 + O(d\|\Sigma_{per} - I_d\|_2^3) .$$

Note that we have

$$\begin{aligned}\mathbb{E}\left[\|\Sigma_{per} - I_d\|_2^2\right] &\leq \frac{9(1+c)}{(2\gamma^2+3)^2}\|\hat{\Sigma}_k - I_d\|_2^2 + (1+1/c)\left(\frac{4\gamma^4}{(2\gamma^2+1)^2}\right)\|\hat{\Sigma}_k^m - I_d\|_2^2 \\ &\leq \frac{9(1+c)}{(2\gamma^2+3)^2}\frac{d}{N_k} + (1+1/c)\left(\frac{4\gamma^4}{(2\gamma^2+1)^2}\right)\frac{d}{n_k^m} .\end{aligned}$$

Thus, we have

$$\mathbb{E}\left[\|(\Sigma_{per})^{1/2} - I_d\|_F^2\right] \leq \frac{9(1+c)}{4(2\gamma^2+3)^2}\frac{d^2}{N_k} + \frac{(1+1/c)\gamma^4}{(2\gamma^2+1)^2}\frac{d^2}{n_k^m} + O\left(\frac{d^2}{n_k^m}\cdot r\right) , \tag{22}$$

where $r = \sqrt{d\log^{3/2}/n_k^m}$. Therefore, combining equation 21 and equation 22, we have

$$\begin{aligned}&\mathbb{E}\left[\|(\hat{\Sigma}_k^m)^{1/2} - I_d\|_F^2\right] - \mathbb{E}\left[\|(\Sigma_{k,per}^m)^{1/2} - I_d\|_F^2\right] \\ &\geq \frac{d^2}{4n_k^m} - \frac{(1+1/c)\gamma^4}{(2\gamma^2+1)^2}\frac{d^2}{n_k^m} - \frac{9(1+c)}{4(2\gamma^2+3)^2}\frac{d^2}{N_k} - O\left(\frac{d^2}{n_k^m}\cdot r\right) \\ &\geq \frac{\gamma^2+\frac{1}{4}-\frac{\gamma^4}{c}}{(2\gamma^2+1)^2}\frac{d^2}{n_k^m} - \frac{\frac{9}{4}(1+c)}{(2\gamma^2+1)^2}\cdot\frac{n_k^m}{N_k}\cdot\frac{d^2}{n_k^m} - O\left(\frac{d^2}{n_k^m}\cdot r\right) .\end{aligned}$$

Therefore, let $c = 16\gamma^4$, and $N_k \geq 36(1+16\gamma^2)n_k^m$, we have

$$\mathbb{E}\left[\|(\hat{\Sigma}_k^m)^{1/2} - I_d\|_F^2\right] - \mathbb{E}\left[\|(\Sigma_{k,per}^m)^{1/2} - I_d\|_F^2\right] \tag{23}$$

$$\geq \frac{\gamma^2+\frac{1}{8}}{(2\gamma^2+1)^2}\frac{d^2}{n_k^m} - O\left(\frac{d^2}{n_k^m}\cdot r\right) , \tag{24}$$

where $r = \sqrt{d\log^{3/2}/n_k^m}$. On the other hand, according to equation 17, we have

$$\mathbb{E}\left[\|\mu_{per} - \mu_k\|_2^2\right] \leq \frac{(1+c')3}{2\gamma^2+3}\frac{d}{N_k} + \frac{2(1+1/c')\gamma^2}{2\gamma^2+1}\frac{d}{n_k^m}. \tag{25}$$

Combining equation 20 and equation 25, we have

$$\mathbb{E}\left[\|\hat{\mu}_k^m - \mu_k\|_2^2\right] - \mathbb{E}\left[\|\mu_{per} - \mu_k\|_2^2\right] \geq \left(\frac{1-\gamma^2/c'}{2\gamma^2+1} - \frac{3(1+c')}{2\gamma^2+1} \cdot \frac{n_k^m}{N_k}\right)\frac{d}{n_k^m} \ . \tag{26}$$

Therefore, let $c' = 4\gamma^2$ and $N_k \geq 12(11+4\gamma^2)n_k^m$, this term is larger than 0. As a result, combining equation 26 and equation 23, we have

$$\begin{aligned}
&\mathbb{E}\left[W_2^2\big(\mathcal{N}(\hat{\mu}_k^m, \hat{\Sigma}_k^m); \mathcal{N}(\mu_k, I_d)\big) - W_2^2\big(\mathcal{N}(\hat{\mu}_k^m, \hat{\Sigma}_k^m); \mathcal{N}(\mu_k, I_d)\big)\right] \\
&\geq \frac{\gamma^2 + \frac{1}{8}}{(2\gamma^2+1)^2}\frac{d^2}{n_k^m} - O\left(\frac{d^2}{n_k^m} \cdot r\right) \ ,
\end{aligned}$$

where $r = \sqrt{d\log^{3/2}/n_k^m}$, as long as $N_k = \Omega((1+\gamma^2)n_k^m)$. Finally, let $n_k^m = \Omega((\gamma^2+1)^2 d\log^{3/2}(1/\delta)$, we have

$$\begin{aligned}
&\mathbb{E}\left[W_2^2\big(\mathcal{N}(\hat{\mu}_k^m, \hat{\Sigma}_k^m); \mathcal{N}(\mu_k, I_d)\big) - W_2^2\big(\mathcal{N}(\hat{\mu}_k^m, \hat{\Sigma}_k^m); \mathcal{N}(\mu_k, I_d)\big)\right] \\
&\geq \frac{\gamma^2 + \frac{1}{8}}{2(2\gamma^2+1)^2}\frac{d^2}{n_k^m} \ ,
\end{aligned} \tag{27}$$

which completes the proof.