# Diffusion Federated Dataset

**Seok-Ju Hahn**[1]
Argonne National Laboratory
hahns@anl.gov

**Junghye Lee**[2]
Seoul National University
junghye@snu.ac.kr

## Abstract

Diffusion models have demonstrated decent generation quality, yet their deployment in federated learning scenarios remains challenging. Due to data heterogeneity and a large number of parameters, conventional parameter averaging schemes often fail to achieve stable collaborative training of diffusion models. We reframe collaborative synthetic data generation as a cooperative sampling procedure from a mixture of decentralized distributions, each encoded by a pre-trained local diffusion model. This leverages the connection between diffusion and energy-based models, which readily supports compositional generation thereof. Consequently, we can directly obtain refined synthetic dataset, optionally with differential privacy guarantee, even without exchanging diffusion model parameters. Our framework reduces communication overhead while maintaining the generation quality, realized through an unadjusted Langevin algorithm with a convergence guarantee.

## 1 Introduction

Federated learning (FL [1]) enables clients (i.e., data owners) to collaboratively train a statistical model by exchanging locally updated parameters with a central server over iterative communication rounds, thereby preserving data privacy. While this *model-centric FL* paradigm is well-established, sharing public data can substantially enhance FL performance by mitigating statistical heterogeneity arising from non-independent and identically distributed (non-IID) local data distributions [1–4]. For instance, public or synthetic datasets can homogenize disparate local distributions and serve as direct signals for server-side pretraining. This facilitates client-side transfer learning or data augmentation, both improve the overall utility of FL.

While these *data-centric FL scheme* offers clear advantages over purely model-centric approaches, there remain challenges. First, curating public datasets is often infeasible in a real-world FL system. Although generation of synthetic data via the collaborative training of generative models is a viable alternative, it is challenging in FL settings. For example, generative adversarial networks (GANs [5]) suffer from training instabilities and suboptimal sample quality, which are exacerbated in FL by statistical heterogeneity [6–8]. Even advanced diffusion models [9–12] incur substantial computational and communication overheads due to their large parameter sizes and fine-grained optimization requirements. Thus, effective synthetic data generation methods for FL constitute a critical yet underexplored research area. Specifically in cross-silo FL settings, clients often have a limited number of samples (e.g., hospitals or enterprises with small datasets). In this sample-limited condition, generating synthetic data becomes critical as it can complement scarce and disparate local dataset with high-fidelity synthetic samples. Thereby, this directly addresses the statistical heterogeneity problem in federated environments.

In this work, we redefine federated synthetic data generation as a *collaborative sampling process* from a mixture of heterogeneous and inaccessible local distributions. By modeling each local distribution with a client-side diffusion model, we enable efficient compositional sampling from them, leveraging

Code is available at: https://github.com/vaseline555/DfD

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

energy-based interpretations of diffusion models [13] and the mixture-of-experts paradigm [14]. The sampling is embarrassingly simple, through the unadjusted Langevin algorithm (ULA [15, 16]). Building on these, we introduce DfD (Diffusion-federated Dataset), a cooperative inference framework that generates synthetic data through *sampling directly from a mixture distribution*, eschewing traditional model averaging. DfD advances federated synthetic data generation as follows:

- We propose a novel view on federated synthetic data generation as cooperative sampling from individually trained diffusion models, without necessitating the exchange of model parameters.
- Through energy-based parameterization and compatibility of ULA with the diffusion reverse process, we refine the connections between diffusion models and energy-based models (EBMs [17]). We also derive the optimal step size and non-asymptotic distributional convergence for DfD.
- We empirically validate fidelity and utility of synthetic dataset from DfD under non-IID conditions, optionally with formal privacy guarantees, addressing key needs in cross-silo FL scenarios.

## 2 Related Works

**Synthetic Data in FL.** FL often struggles with slow convergence when the client's local private data sets differ significantly, a common challenge known as statistical heterogeneity or the non-IID problem [1, 4]. This issue is critical in that the central server cannot directly access or adjust these heterogeneous local datasets to align their disparate optimization trajectories. Most prior work has addressed this through *model-centric* approaches, such as local update regularization [18–21], modified central aggregation schemes [22–27], or personalization [28–30].

While effective, a complementary *data-centric* perspective still remains underexplored. These include sharing additional server-side public data [2, 31–33], using indiscernible auxiliary representations [34–40], or leveraging a generative model to obtain plausible synthetic data [41–50]. These provide clients with a proxy for global distribution, which directly mitigates the non-IID problem and improves convergence [51]. Notably, as studied in [2], sharing only a small portion of public data can significantly boost FL performance, though acquiring such data is nontrivial in practice.

Hence, synthetic data is widely used with generative models in e.g., healthcare [52–57]. However, current synthetic data generation methods in FL, including real-world applications, mostly resort to GANs [58] (optionally with privacy guarantee [59–62]), which suffer from subpar generation quality and optimization instability due to their adversarial training scheme (e.g., mode collapse [6–8]).

**Diffusion Models in FL.** Diffusion models [11, 12], such as Denoising Diffusion Probabilistic Models (DDPMs [9]), have offered a superior generation quality training stability, compared to other generative models, e.g., GANs. [10]. Although promising, their adoption in FL is challenging and sometimes even prohibitive due to high computational costs and large model sizes. Thus, current methods suffer from significant communication overhead [63], poor scalability to high-resolution data [64], and even require retraining of local models [65] or data sharing [66] due to non-IID problem. In addition, the inherent loss design of diffusion models, which depend on multiple time-steps, also requires frequent parameter exchanges during training, making them difficult to adopt in FL [66, 67].

Our framework detours by directly generating samples from an inaccessible mixture of heterogeneous local distributions, encoded by locally-trained diffusion models. This is rooted in exploiting the connection of diffusion models to energy-based models (EBMs [17]), which estimate unnormalized probability densities through their gradients with respect to inputs (i.e., *scores* [68]). This intriguing connection enables easy compositional sampling, which can be viewed as sampling from a mixture-of-experts [14], even without accessing model parameters. As a result, DfD offers an efficient and scalable solution for adopting diffusion models in federated synthetic data generation.

## 3 Preliminaries

### 3.1 Diffusion Models

Diffusion models aim to encode data distribution $p_{\text{data}}(\boldsymbol{x})$ by learning transition from noise-perturbed data $\{\boldsymbol{x}_t\}_{t=1}^T$, where $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$, into its clean original counterpart, $\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}) \equiv q(\boldsymbol{x}_0)$ through paired forward and backward processes. Specifically, Gaussian diffusion defines a Markov

chain joint distribution $q(\boldsymbol{x}_0, ..., \boldsymbol{x}_T) = q(\boldsymbol{x}_0) \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$, where the *forward process* is defined by incrementally adding Gaussian noise over $t \in [T]$ as $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}_d)$. Note that $d$ is the data dimension, $0 < \beta_t \leq 1$ and $\alpha_t = 1 - \beta_t$ are noise constants. The *reverse process*, typically parameterized by a deep network with $\boldsymbol{\theta}$, approximates $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, in order to progressively denoise from the Gaussian noise $\boldsymbol{x}_T$ into the original data $\boldsymbol{x}_0$. With sufficiently small $\beta_t$, each transition of reverse process approximately follows Gaussian [11]. This allows:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}\Big(\boldsymbol{x}_{t-1}; \underbrace{\frac{1}{\sqrt{\alpha_t}}\Big(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\Big)}_{=:\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}, \tilde{\beta}_t \mathbf{I}_d\Big), \tag{1}$$

where $\bar{\alpha}_t, \tilde{\beta}_t$ are some transformations of $\beta_t, \forall t \in [T]$, following the configurations of [9] (see also Appendix C.1).

Eventually, the parameterized deep network needs to predict $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ as a mapping $\boldsymbol{\epsilon}_{\boldsymbol{\theta}} : \mathbb{R}^d \times [T] \to \mathbb{R}^d$. Note that diffusion models ensure the analytic conversion from the original to the perturbed data at any timestep $t \in [T]$ [9]:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d). \tag{2}$$

Using this property, we can optimize with composite loss [9] as $\mathcal{L}(\boldsymbol{\theta}) = \sum_{t=1}^{T} \mathcal{L}(\boldsymbol{\theta}, t)$, where

$$\mathcal{L}(\boldsymbol{\theta}, t) = \mathbb{E}_{\boldsymbol{x}_0 \sim p_{\text{data}}(\boldsymbol{x}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right]. \tag{3}$$

By minimizing this objective, diffusion models are capable of generating high-quality samples by constructing $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ from their prediction $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ and progressively denoising from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$ to $\boldsymbol{x}_0$ over $t = T-1, ..., 1$, using Eq. (1). We refer to Appendix A for detailed derivations.

### 3.2 Energy-based Interpretation of Diffusion Models

Diffusion models have an intriguing connection with EBMs [12, 13]. EBMs [17] define an unnormalized probability density as:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{\exp\left(-\lambda f_{\boldsymbol{\theta}}(\boldsymbol{x})\right)}{Z_{\boldsymbol{\theta}}}, \tag{4}$$

where EBMs forgo modeling of the normalizing constant $Z_{\boldsymbol{\theta}} = \int_{\boldsymbol{x} \in \mathcal{X}} \exp\left(-\lambda f_{\boldsymbol{\theta}}(\boldsymbol{x})\right) \mathrm{d}\boldsymbol{x}$. We define $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}$ as an *energy function* with parameter $\boldsymbol{\theta} \in \mathbb{R}^p$, scale factor $\lambda \in \mathbb{R}^+$ and $\nabla_{\boldsymbol{x}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = -\lambda \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x})$ as a *score*. Note that we have $d \ll p$ if we choose deep networks, which are typically overparameterized.

The abstention of modeling normalizing constant prevents exact likelihood computation. To address the issues that arise from this design, *denoising score matching* [69] has been proposed to minimize the Fisher divergence between the model's score and that of a noise-perturbed data distribution, i.e., $q(\boldsymbol{x}_\sigma) = \int_{\boldsymbol{x} \in \mathcal{X}} q_\sigma(\boldsymbol{x}_\sigma|\boldsymbol{x}) p_{\text{data}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$. Note here that $\sigma$ is a noise variance and the perturbation is given as $\boldsymbol{x}_\sigma = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$. Building on these, the denoising score matching objective is:

$$\mathcal{L}(\boldsymbol{\theta}, \sigma) = \mathbb{E}_{\boldsymbol{x}_\sigma \sim q(\boldsymbol{x}_\sigma|\boldsymbol{x}), \boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})} \left[\|\nabla_{\boldsymbol{x}_\sigma} \log q(\boldsymbol{x}_\sigma|\boldsymbol{x}) - \nabla_{\boldsymbol{x}_\sigma} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_\sigma)\|^2\right]. \tag{5}$$

This is equivalent (up to a constant) to:

$$\sigma^2 \mathcal{L}(\boldsymbol{\theta}, \sigma) = \mathbb{E}_{\boldsymbol{x}_\sigma \sim q(\boldsymbol{x}_\sigma|\boldsymbol{x}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)} \left[\|\boldsymbol{\epsilon} - \sigma\lambda \nabla_{\boldsymbol{x}_\sigma} f_{\boldsymbol{\theta}}(\boldsymbol{x}_\sigma)\|^2\right]. \tag{6}$$

Interestingly, the objective of diffusion models in Eq. (3) aligns with the scaled objective above [13], with following connection (along with replacing $\sigma$ into $\sigma_t$):

$$\nabla_{\boldsymbol{x}_{\sigma_t}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_{\sigma_t}) = -\lambda \nabla_{\boldsymbol{x}_{\sigma_t}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_{\sigma_t}) \equiv -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sigma_t}. \tag{7}$$

Note that this connection to EBMs can be further concretized for diffusion models by a specific choice of the *energy-based parameterization* introduced in following Section 3.4. It should also be noted that this explicit connection allows using a sampler for diffusion models, e.g., ULA. We defer all detailed derivations in this section to Appendix B.

### 3.3 Federated Synthetic Data Generation by Sampling from a Mixture Distributions

The ULA follows a discretized Langevin diffusion process [15] and enables sampling from a target distribution $p(\boldsymbol{x})$ with its score $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$, by iteratively updating from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$ using:

$$\boldsymbol{x}_{t-1} = \boldsymbol{x}_t + \eta_t \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) + \sqrt{2\eta_t}\boldsymbol{z}_t, \quad \boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d), \tag{8}$$

where $\eta_t \geq 0$ is a step size, and it ensures $\boldsymbol{x}_0 \sim p(\boldsymbol{x})$ [70]. We denote the notation of decreasing timesteps as $t = T, ..., 1$ for the compatibility with the diffusion reverse process.

In FL setup, we have $K$ clients each having private dataset $\mathcal{D}_i$. Then, a target distribution is naturally defined as a mixture of local distributions: $p^\star(\boldsymbol{x}) = \sum_{i=1}^K w_i p_i(\boldsymbol{x})$, where $p_i(\boldsymbol{x})$ represents unknown local distribution of $\mathcal{D}_i$ from $i$-th client and $w_i \geq 0$ is a mixing coefficient satisfying $\sum_{i=1}^K w_i = 1$ (e.g., $w_i = 1/K$ if uniform weighting). To generate samples from the mixture of local distributions, what we need to estimate the *global score* $\nabla_{\boldsymbol{x}} \log p^\star(\boldsymbol{x})$ defined as follows:

$$\nabla_{\boldsymbol{x}} \log p^\star(\boldsymbol{x}) = \sum\nolimits_{i=1}^K \tilde{w}_i \nabla_{\boldsymbol{x}} \log p_{\boldsymbol{\theta}_i}(\boldsymbol{x}), \quad \tilde{w}_i = \frac{w_i \exp(-\lambda f_{\boldsymbol{\theta}_i}(\boldsymbol{x}))}{\sum_{j=1}^K w_j \exp(-\lambda f_{\boldsymbol{\theta}_j}(\boldsymbol{x}))}, \tag{9}$$

where $\tilde{w}_i$ is derived from $p_{\boldsymbol{\theta}_i}(\boldsymbol{x}) \propto \exp\left(-\lambda f_{\boldsymbol{\theta}_i}(\boldsymbol{x})\right)$ due to Eq. (4). Note that it directly supports embarrassingly parallel computation across clients, aligning well with FL settings. In detail, the estimation of the global score $\nabla_{\boldsymbol{x}} \log p^\star(\boldsymbol{x})$ is available as long as we have both i) *local scores* $\nabla_{\boldsymbol{x}} \log p_{\boldsymbol{\theta}_i}(\boldsymbol{x})$ and ii) *energies* (unnormalized density values) $\exp(-\lambda f_{\boldsymbol{\theta}_i}(\boldsymbol{x}))$ of each client.

However, diffusion models do not explicitly provide $f_{\boldsymbol{\theta}_i}(\boldsymbol{x})$ in its inherent design. This can be easily addressed using energy-based parameterization described in the following section.

### 3.4 Energy-based Parameterization of Diffusion Models

To implement ULA to directly sample from a mixture of local distributions, we should estimate the energies $p_{\boldsymbol{\theta}_i}(\boldsymbol{x}) \propto \exp(-\lambda f_{\boldsymbol{\theta}_i}(\boldsymbol{x}))$ for $\tilde{w}_i$ in Eq. (9). Since diffusion models lack explicit density function, prior arts proposed to approximate them by defining $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ using an energy-based $\ell_2$ parameterization trick [13, 71]. (We refer to Section D of [13] for details on other tricks)

**Definition 3.1** (energy-based $\ell_2$ parameterization [13])**.** The energy function of a diffusion model is approximated as $f_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{1}{2}\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2$, where $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is a prediction of a diffusion model.

Having this energy function, we can now define scores of diffusion models and obtain the global score in Eq. (9), accordingly. Unfortunately, this parameterization requires a modification in training of diffusion models, and this often yields subpar generation quality [13, 71].

## 4 Proposed Method

### 4.1 Refined Energy-based Parameterization

To detour the modification in training, we start from the notion of *well-trained diffusion models*.

**Definition 4.1** (Well-trained diffusion model)**.** A diffusion model is well-trained if, through minimization of the objective in Eq. (3), its noise prediction satisfies $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$.

**Remark 4.2.** Note that this captures the empirical observation that sufficiently trained diffusion models accurately predict added noise. In addition, thanks to $\boldsymbol{\epsilon} = (\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0)/\sqrt{1-\bar{\alpha}_t}$ from Eq. (2), a well-trained diffusion model readily satisfies that $\nabla_{\boldsymbol{x}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \nabla_{\boldsymbol{x}_t}\boldsymbol{\epsilon} = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\mathbf{I}_d$.

With these, we can now approximate the score of well-trained diffusion models as follows:

$$\begin{aligned}
\nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) &= -\lambda \nabla_{\boldsymbol{x}_t} f_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \\
&= -\lambda \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)^\intercal \nabla_{\boldsymbol{x}_t} \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \\
&\approx -\frac{\lambda}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t),
\end{aligned} \tag{10}$$

where the first equality is a direct result from Eq. (4), the second equality is due to Definition 3.1, and the last approximation is from Definition 4.1. Notably, this matches Eq. (7) if $\sigma_t = \sqrt{1-\bar{\alpha}_t}/\lambda$. To

summarize, the *refined* energy-based $\ell_2$ reparameterization provides an unnormalized density and a score of well-trained diffusion models as:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \propto \exp\left(-\frac{\lambda}{2}\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right), \quad \nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = -\frac{\lambda}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \tag{11}$$

for all timesteps $t \in [T]$. With these, *no modification to the training* of diffusion models is required.

### 4.2 DfD: Cooperative Diffusion Models Inference Framework for Synthetic Dataset

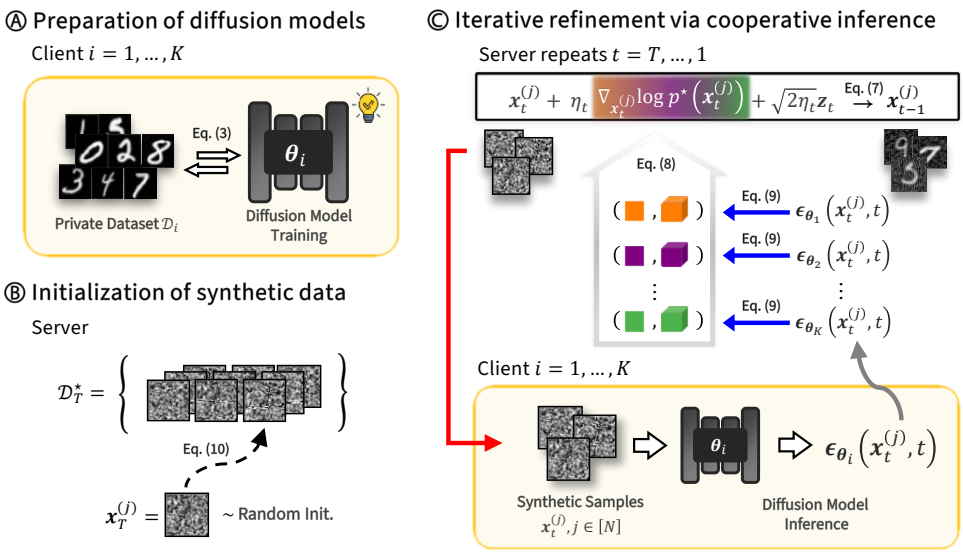


Figure 1: **Overview of DfD.** Ⓐ Clients independently train diffusion models to be well-trained with Eq. (3). Ⓑ The server randomly initializes synthetic dataset per Eq. (12). Ⓒ The server requests (➡) inference on synthetic dataset to all clients, receives (➡) predictions $\boldsymbol{\epsilon}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_t^{(j)}, t), \forall i \in [K], j \in [N]$, transforms (➡) into energies (■, ■, ■) and scores (■, ■, ■) using Eq. (11), composes into global scores using Eq. (9), and refines synthetic dataset using ULA in Eq. (8) over $T$ steps.

Our proposed framework, DfD, generates synthetic data samples directly from a mixture of local distributions encoded by local diffusion models, independently trained on private and non-IID client datasets. An overview of the framework is provided in Figure 1 and the overall procedure of DfD (for the case of unconditional generation) is described in Algorithm 1.

A key innovation of DfD is its ability to leverage locally trained diffusion models, avoiding repetitive local updates along with the exchange of model parameters. This is achieved by exchanging the predictions of well-trained diffusion models instead, and the models are prepared by each client before cooperative inference begins. These predictions are iteratively collected and transformed into energies and scores at the server, to construct a global score in Eq. (9).

Ⓐ **Preparation of diffusion models.** Each client $i$ trains its own diffusion model on its private dataset $\mathcal{D}_i$ by minimizing Eq. (3), to obtain a *well-trained* model as in Definition 4.1. The model can be unconditional, predicting $\boldsymbol{\epsilon}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_t, t)$, or conditional on label $\boldsymbol{y}$ (e.g., attributes or classes), predicting $\boldsymbol{\epsilon}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_t, \boldsymbol{y}, t)$. Note that the dimension of predictions is equal to that of inputs, which is significantly smaller than the model parameter size. In addition, local pre-training can occur asynchronously, and clients may optionally apply differential privacy (DP) mechanisms [72].

Ⓑ **Initialization of synthetic data.** The central server randomly initializes $N$ synthetic data samples $\mathcal{D}_T^\star = \{\boldsymbol{x}_T^{(j)}\}_{j=1}^N$ (or $\mathcal{D}_T^\star = \{(\boldsymbol{x}_T^{(j)}, \boldsymbol{y}^{(j)})\}_{j=1}^N$) as:

$$\boldsymbol{x}_T^{(j)} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d), \qquad \text{(if conditional)} \quad \boldsymbol{y}^{(j)} \sim \text{Categorical}\left(C^{-1}\mathbf{1}_C\right), \tag{12}$$

**Algorithm 1** DfD: Cooperative Diffusion Models Inference Framework for Synthetic Dataset

**Require:** number of clients $K$, synthetic dataset size $N$, communication rounds $T$
**Procedure:**
  **All clients** $i \in [K]$ prepare a well-trained diffusion model $\boldsymbol{\theta}_i \in \mathbb{R}^p$ using $\mathcal{D}_i$ with Eq. (3).
  Server initializes $N$ samples in Eq. (12) to have $\mathcal{D}_T^\star$.
  **for** $t = T, ..., 1$ **the server**
    Requests inference to all clients in parallel on $\mathcal{D}_t^\star$.
    Receives predictions $\{\boldsymbol{\epsilon}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_t^{(j)}, t) \in \mathbb{R}^d \mid i \in [K], j \in [N]\}$.
    Transforms predictions into energies and scores with Eq. (11).
    Computes global scores for all samples with Eq. (9).
    Updates synthetic dataset into $\mathcal{D}_{t-1}^\star$ using ULA in Eq. (8).
  **end for**
**Return:** $\mathcal{D}_0^\star$

where $C$ is the number of conditions (e.g., classes, attributes) encoded by labels. The synthetic dataset size $N$ is determined based on communication constraints, where $N$ can be set much smaller than the required parameter size of diffusion models, e.g., $N \ll \max_i \dim(\boldsymbol{\theta}_i)$.

**Ⓒ Iterative refinement via cooperative inference.** For each communication round $t = T, ..., 1$, the central server sends the current synthetic dataset to all clients and requests predictions from their diffusion models. With these predictions, the server computes energies and scores of each client using Eq. (11). The server then constructs global scores using Eq. (9) and refines the server-side synthetic dataset using ULA, as in Eq. (8). Note that it can be extended to the conditional case by simply incorporating $\boldsymbol{y}^{(j)}$ in this step. At the end, the server obtains a refined synthetic dataset, $\mathcal{D}_0^\star$.

### 4.3 Theoretical Analysis

The ULA is the main workhorse of DfD as it relies on energy-based parameterization to sample from a mixture of local distributions using global scores in Eq. (9). Hence, we must carefully select the step size, denoted by $\eta_t$, to ensure that the DfD correctly settles at the target mixture distribution. We theoretically derive the step size guidance in two steps: ⓐ verification of the compatibility of ULA with diffusion reverse process, and ⓑ analysis of non-asymptotic convergence behavior of ULA to the target distribution in KL divergence [70]. We defer all proofs in Appendix C.

**ⓐ Compatibility of ULA with diffusion reverse process.** DfD resort to diffusion models as main components. Thus, we begin with the successful diffusion reverse process and transplant its key success factor into the ULA to ensure compatibility. Interestingly, we find that *non-expansiveness* w.r.t. $\ell_2$-norm is inherently encoded in the diffusion reverse process, and perceive it as a key factor.

**Lemma 4.3** (Non-expansiveness of diffusion reverse process)**.** *The diffusion reverse process in Eq. (1) preserves the squared $\ell_2$-norm of resulting iterates to be non-expansive, i.e., $\mathbb{E}[\|\boldsymbol{x}_{t-1}\|_2^2] \leq \mathbb{E}[\|\boldsymbol{x}_t\|_2^2]$.*

Next, we proved that this property can be similarly induced for ULA under following conditions. This gives an explicit guidance for the choice of scale factor $\lambda$ in Eq. (4), which is used for the construction of energies and scores in Eq. (9).

**Lemma 4.4** (Non-expansiveness condition of ULA)**.** *ULA satisfies the non-expansiveness w.r.t. squared $\ell_2$-norm as $\mathbb{E}[\|\boldsymbol{x}_{t-1}\|_2^2] \leq \mathbb{E}[\|\boldsymbol{x}_t\|_2^2]$, for well-trained diffusion models with energy-based $\ell_2$ parameterization, if and only if $\eta_t \in [0, \frac{1}{2}]$ and $\lambda = 2$.*

**ⓑ Non-asymptotic convergence of ULA.** Though previous work heuristically adopted the naive resemblance of ULA with the diffusion reverse process to set the step size (i.e., simply setting $\eta_t = \beta_t$ while ignoring the scaling factor $\frac{1}{\sqrt{\alpha_t}}$) [71], this approach has no theoretical justification. Thus, we theoretically derive a ULA step size and the convergence guarantee toward a target mixture distribution under KL divergence, with acceptable assumptions provided in Appendix C.

**Theorem 4.5** (Convergence guarantee of DfD)**.** *Let $\tilde{p}_t$ be the evolving distribution of $\boldsymbol{x}_t \in \mathbb{R}^d$ from ULA and $p_t$ be the mixture of distributions encoded by diffusion models. For $\delta \geq \frac{18d\varsigma\eta_T(1-\bar{\alpha}_{T-t})}{\upsilon}$ and $\rho \in (0, \sqrt{3}/6)$, the iterates $\boldsymbol{x}_{T-t} \sim \tilde{p}_{T-t}^\star$ guarantee $\mathrm{D}_{\mathrm{KL}}(\tilde{p}_{T-t} \parallel p_{T-t}) < \delta$ after $t \geq \frac{1-\bar{\alpha}_{T-t}}{\upsilon} \log\left(\frac{2\,\mathrm{D}_{\mathrm{KL}}(\tilde{p}_T \| p_T)}{\delta}\right)$ steps with a step size $\eta_{T-t} \leq \min\left\{\frac{\upsilon\delta}{18d\varsigma(1-\bar{\alpha}_{T-t})}, \frac{\rho(1-\bar{\alpha}_{T-t})^p}{2^p}\right\}$.*

Note that this ensures DfD can sample from a mixture of inaccessible and heterogeneous distributions in a finite number of steps, *without access* to the local dataset $\mathcal{D}_i$ and the local model parameters $\boldsymbol{\theta}_i$.

### 4.4 Privacy Guarantee

**Definition 4.6** ($(\epsilon, \delta)$-DP [72])**.** A mechanism $\mathcal{M}$ satisfies $(\epsilon, \delta)$-DP if, for any two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ differing in one record, and for any output set $S$, $\Pr[\mathcal{M}(\mathcal{D}) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(\mathcal{D}') \in S] + \delta$, where $\epsilon > 0$ is the privacy budget and $\delta \geq 0$ is the failure probability.

The communicated signals in DfD are client predictions $\boldsymbol{\epsilon}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_t^{(j)}, t)$ from a diffusion model trained on a private dataset $\mathcal{D}_i$. In FL, we typically use DP mechanism to protect sensitive information. Intriguingly, DfD can inherit DP guarantee as long as each client $i$ already trained its own diffusion model $\boldsymbol{\theta}_i$ to achieve $(\epsilon_i, \delta_i)$-DP, e.g., using DP-SGD [73].

**Theorem 4.7** (DP guarantee of DfD)**.** *Assume all client datasets $\mathcal{D}_i$ are disjoint. If each client $i \in [K]$ trains a diffusion model $\boldsymbol{\theta}_i$ for $(\epsilon_i, \delta_i)$-DP given $\epsilon_i > 0$ and $\delta_i \geq 0$, the synthetic dataset $\mathcal{D}_0^{\star}$ generated by DfD compositely satisfies $(\max_i \epsilon_i, \max_i \delta_i)$-DP.*

*Proof.* As the server processes differentially private local predictions, the post-processing property of DP [74] also ensures that subsequent steps (i.e., global score computation, ULA updates) to preserve DP. The parallel composition theorem [75] provides a composite DP guarantee across clients with disjoint datasets with each other, in terms of the maximum privacy budget and failure probability. □

## 5 Experimental Results

### 5.1 Setup

**Datasets.** We use three benchmark datasets: MNIST [76], CIFAR-10 [77], and CelebA [78], after resizing all inputs to have spatial dimension of 32×32. As each dataset has separate train & test folds, we use the train fold to split into client datasets, and set the test fold aside for server-side evaluation. We distribute the train fold of each dataset into $K = 10$ clients with three different non-IID conditions: i) Dirichlet distribution-based non-IID [79] for MNIST, ii) power-law distribution-based non-IID [21] for CIFAR-10, and iii) pathological non-IID [1] for CelebA.

To further simulate a convincing scenario in which a synthetic dataset should be procured (i.e., *data-limited settings*), we randomly sample local dataset to have a size of 300 on average, following the sample size configurations of the curated benchmark for the cross-silo FL setting [80].

**Baselines.** We compare with FL methods for generative models: FedGAN [42], FedDiffuse [63] and PRISM [67]. All clients are taking 10K steps in total for $T = 1000$ rounds: $E = 10$ local updates for all comparison methods, and $E = 10 \times 1,000 = 10,000$ local updates for DfD as it requires no update during communication rounds. The mini-batch size is set to $B = 32$, and the learning rates are tuned for all methods, and set to $c(1 - \bar{\alpha}_t)^p$ for $c > 0, p \geq 1$ for DfD.

**Evaluation Metrics.** We evaluate both fidelity and utility of the generated synthetic dataset. To evaluate the fidelity of synthetic data, we use the widely-used metrics for generative modeling: Fréchet Inception Distance (FID [81]), Precision & Recall (P&R [82]), and Density & Coverage (D&C [83]). To evaluate utility, we use an accuracy evaluated from a classifier trained at the central server using class-labeled synthetic dataset. We defer the specific experimental setup to Appendix D.

### 5.2 Results

**Quality and Utility.** Table 1 summarizes the quality-based results, i.e., FID, Precision (P), Recall (R), Density (D) and Coverage (D). Our method outperforms other FL methods for generative modeling in synthetic data fidelity. We provide generation results of each method in Figure 3. Table 2 summarizes the test accuracies as synthetic data utility. Following [84], we train three server-side classifiers on each generated synthetic dataset: logistic regression (*LogReg*), multi-layered perceptron (*MLP*), and convolution neural network (*CNN*). We evaluate each classifier on a separate test fold

Table 1: Results on synthetic dataset quality.

| | | FID ↓ | P ↑ | R ↑ | D ↑ | C ↑ |
|---|---|---|---|---|---|---|
| MNIST | FedGAN [42] | **34.8486** | 0.4189 | 0.1240 | 0.1144 | 0.1378 |
| | FedDiffuse [63] | 49.5704 | 0.1842 | **0.7610** | 0.1145 | 0.3428 |
| | PRISM [67] | 36.7945 | 0.4223 | 0.1386 | 0.1639 | 0.1481 |
| | DfD | 37.7354 | **0.6224** | 0.3437 | **0.1816** | **0.3937** |
| CIFAR-10 | FedGAN [42] | 145.5668 | **0.6866** | 0.0221 | **0.4800** | 0.1221 |
| | FedDiffuse [63] | 78.3845 | 0.4142 | 0.2119 | 0.3731 | 0.2958 |
| | PRISM [67] | 330.8488 | 0.0875 | 0.0077 | 0.0334 | 0.0368 |
| | DfD | **59.9761** | 0.5153 | **0.2492** | 0.3521 | **0.3590** |
| CelebA | FedGAN [42] | 98.1784 | 0.3469 | 0.4210 | 0.1349 | 0.1929 |
| | FedDiffuse [63] | 33.3323 | 0.2986 | **0.5176** | **0.2318** | **0.2793** |
| | PRISM [67] | 200.1870 | 0.1479 | 0.1809 | 0.0684 | 0.0769 |
| | DfD | **29.1832** | **0.3734** | 0.4143 | 0.2229 | 0.2370 |

Table 2: Results on synthetic dataset utility.

| | | *LogReg* | *MLP* | *CNN* |
|---|---|---|---|---|
| MNIST | FedGAN [42] | 71.7 | 72.4 | 73.6 |
| | FedDiffuse [63] | 78.2 | 77.5 | 78.8 |
| | PRISM [67] | 43.1 | 41.4 | 45.3 |
| | DfD | **78.5** | **78.1** | **78.9** |
| CIFAR-10 | FedGAN [42] | 19.8 | 21.1 | 24.3 |
| | FedDiffuse [63] | **29.2** | 31.3 | 33.0 |
| | PRISM [67] | 11.5 | 12.9 | 13.2 |
| | DfD | 28.3 | **32.4** | **34.1** |
| CelebA | FedGAN [42] | 42.1 | 43.4 | 45.8 |
| | FedDiffuse [63] | 55.2 | **58.1** | 58.2 |
| | PRISM [67] | 12.2 | 11.3 | 13.4 |
| | DfD | **57.3** | 56.5 | **59.3** |

held in the central server. As a proxy of raw local data samples inaccessible in FL settings, synthetic dataset from DfD have been shown to offer better utility compared to existing baselines.

**Efficiency.** The communication costs differ in DfD compared to other methods. Table 3 summarizes the communication target and computation budget required to generate $N$ samples. During FL, DfD is faster in computation as it only conducts inferences on samples (i.e., $N$ forward passes for $N$ samples), whereas other methods require both backward and forward passes $N \times E$ times to update parameters. Additionally, DfD exchanges predictions of which size is $N \times d$, where the dimension is far smaller than the size of the model parameters (i.e., $d \ll p$. Thus, it can significantly reduce communication ($\because N \times d \ll p$) costs by setting reasonable number of samples, $N$.

Table 3: Comparison on communication cost & computation complexity.

| | Communication | Computation |
|---|---|---|
| FedGAN [42]<br>FedDiffuse [63]<br>PRISM [67] | $\boldsymbol{\theta}_i \in \mathbb{R}^p$ | $\mathcal{O}(N \times E \times p)$<br>(forward & backward $E$ times) |
| DfD | $\left\{\boldsymbol{\epsilon}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_t^{(j)}, t)\right\}_{j=1}^{N} \in \mathbb{R}^{N \times d}$ | $\mathcal{O}(N \times p)$<br>(single forward pass) |

**Privacy.** Thanks to Theorem 4.7, DfD readily satisfies DP. Following [85], we let each client train its diffusion model with DP-SGD [73], under shard-partitioned non-IID setting [1] for $K = 10$ clients:

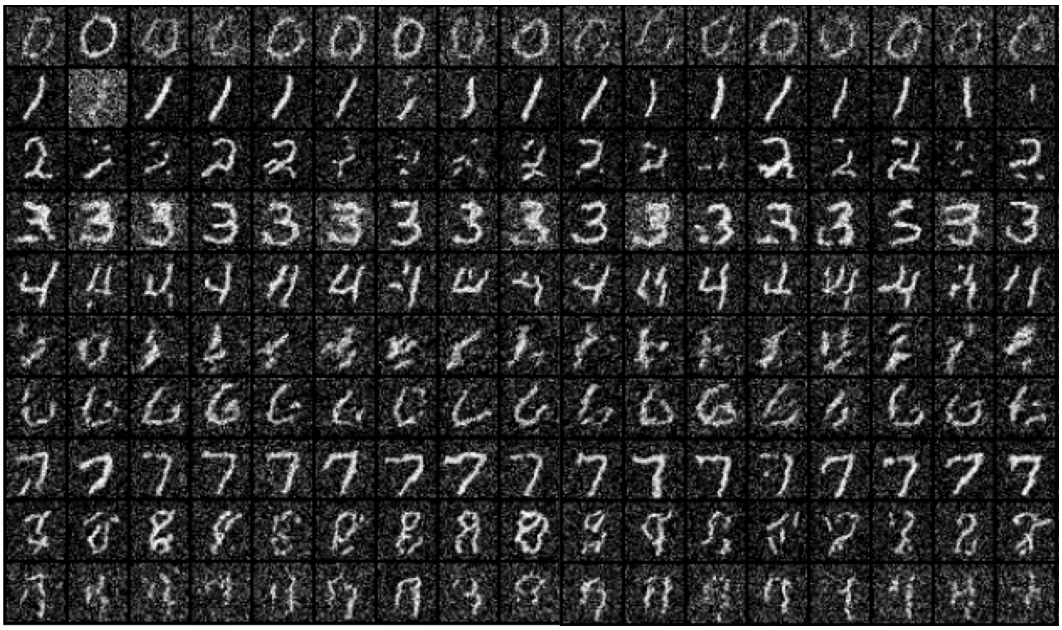

Figure 2: Differentially private synthetic dataset for MNIST from DfD under $(\epsilon = 10, \delta = 10^{-5})$-DP.

each client has samples from only 2 out of 10 classes from MNIST dataset, i.e., digits 0 and 1 for client 0, digits 1 and to for client 1, ... , and digits 9 and 0 for client 9. To achieve $(\max_i \epsilon_i, \max_i \delta_i)$-DP for the resulting synthetic dataset, we set $\epsilon_i = 10$ and $\delta_i = 10^{-5}$ for all $i \in [K]$ clients. We found that applying DP is detrimental to sample quality as expected, and subtle tuning of step size is required to obtain discernible samples. Thereby, improving the quality of ULA sampling from differentially private diffusion models is a promising future direction for DfD in practice.

## 6 Limitation and Discussion

DfD gives clients great flexibility under the assumption of credible participation, such as cross-silo FL settings. In cross-device FL settings, where massive and unreliable clients [4] participate, DfD may fail, so we only consider cross-silo FL settings where a moderate number of credible clients participate. This could be relaxed by allowing partial participation through approximation of a global score at the central server [86].

Currently, the server ends up having synthetic dataset at last, not a generative model. Thus, by training a server-side amortized sampler [87–89] to emulate the collaborative sampling process, we can additionally generate samples even after the collaboration. Moreover, the communication cost can be further reduced by adapting advanced samplers [90–93] or by using model compression techniques, which we leave for future work.

The success of DfD hinges on the faithful, authorized training of local diffusion models by participating clients. However, when local diffusion models are overfit or even memorize samples, this would introduce biased or collapsed sampling, resulting in catastrophic generation results. Therefore, careful training is required (e.g., earlystopping, weight decay) to acquire literally *well-trained* diffusion models. To guarantee trustworthy training, DfD requires a credible consortium of clients. Alternatively, we can use cryptographic tools, such as zero-knowledge proofs, to certify verified pre-training [94].

Lastly, thanks to the advancement in diffusion models, we expect DfD to be extended to other modalities than images [95, 96]. We believe the directions discussed thus far could improve the scalability and practicality of DfD in future works.

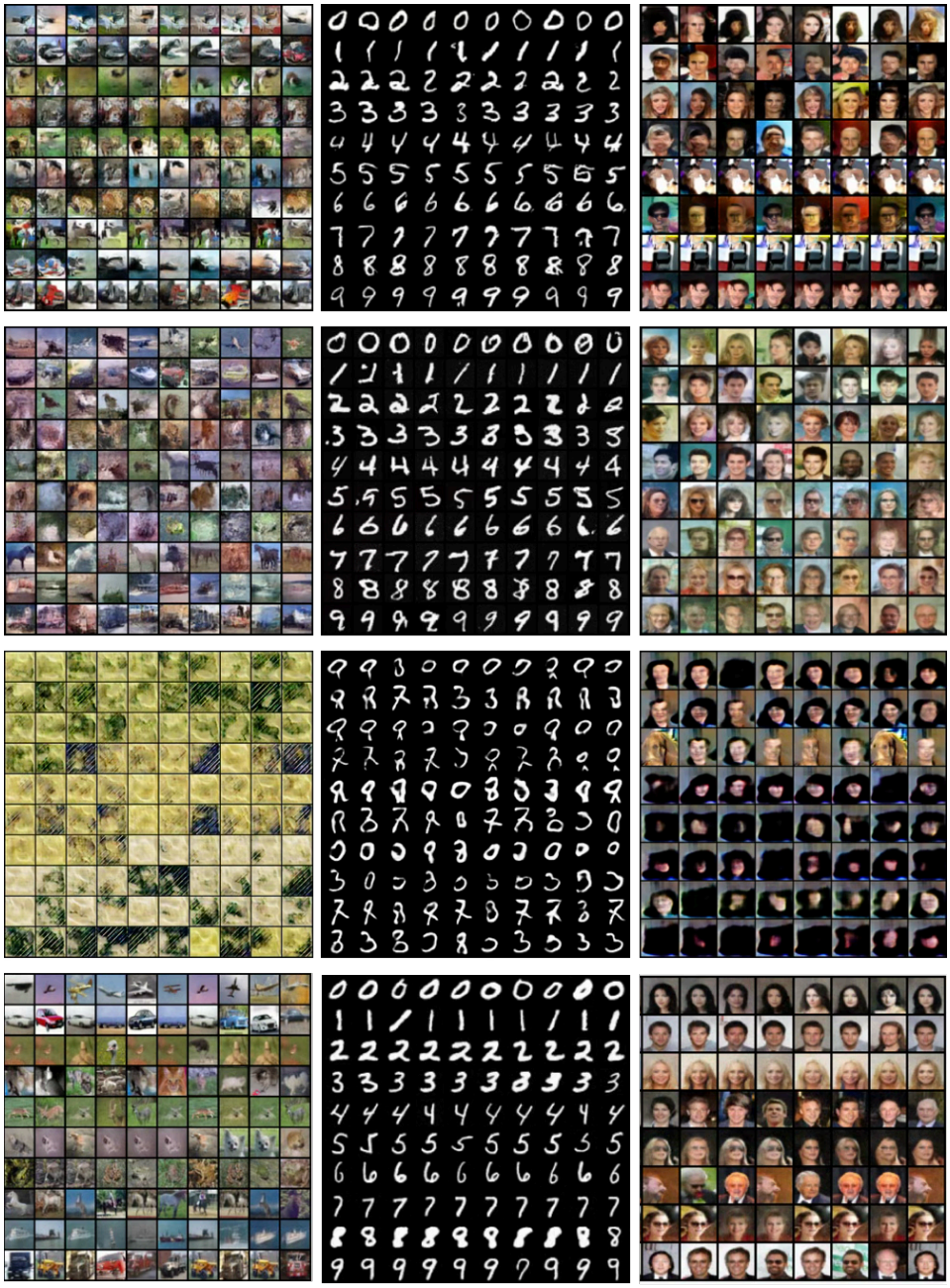

Figure 3: **Visualization of synthetic dataset generated under data-limited non-IID setting.** Each row corresponds to `FedGAN` [42], `FedDiffuse` [63], `PRSIM` [67], and `DfD`. Each column corresponds to CIFAR-10 [77], MNIST [76], and CelebA [78].

## 7 Conclusion

We propose a collaborative synthetic data generation framework, `DfD`, that leverages an energy-based connection for cooperative inference of diffusion models. `DfD` offers improvements in generation quality, communication efficiency, and easy privacy guarantees with theoretically grounded design. Given wide implications of synthetic data in federated settings, we look forward to exploring extensions of `DfD` to diverse modalities and data-intensive domains as a trustworthy framework.

## Broader Impact

The DfD framework enables federated synthetic data generation with privacy guarantees, promoting secure data sharing in privacy-sensitive domains. It produces high-quality synthetic data that preserves statistical properties, improving collaborative research and training of models while complying with e.g., GDPR [97] and HIPAA [98]. However, as synthetic datasets can be possibly misused for malicious purposes, a robust accounting protocol is required for ethical deployment.

## Acknowledgements

This research was conducted when Seok-Ju was at Seoul National University, supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government under Grant No. RS-2025-00516776.

## References

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

[2] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[3] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

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

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[6] Farzan Farnia and Asuman Ozdaglar. Do gans always have nash equilibria? In *International Conference on Machine Learning*, pages 3029–3039. PMLR, 2020.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[8] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

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

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

[11] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[12] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[13] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pages 8489–8510. PMLR, 2023.

[14] Robert A Jacobs. Bias/variance analyses of mixtures-of-experts architectures. *Neural computation*, 9(2):369–383, 1997.

[15] Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. 1996.

[16] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5272–5280, 2020.

[17] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

[18] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

[19] Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. *arXiv preprint arXiv:2301.09604*, 2023.

[20] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machifne learning and systems*, 2:429–450, 2020.

[22] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019.

[23] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[24] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[25] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019.

[26] Seok-Ju Hahn, Gi-Soo Kim, and Junghye Lee. Pursuing overall welfare in federated learning through sequential decision making. In *Proceedings of the 41st International Conference on Machine Learning*, pages 17246–17278, 2024.

[27] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[28] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

[29] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pages 2089–2099. PMLR, 2021.

[30] Yue Wu, Shuaicheng Zhang, Wenchao Yu, Yanchi Liu, Quanquan Gu, Dawei Zhou, Haifeng Chen, and Wei Cheng. Personalized federated learning under mixture of distributions. In *International Conference on Machine Learning*, pages 37860–37879. PMLR, 2023.

[31] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[32] Neel Guha, Ameet Talwalkar, and Virginia Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

[33] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[34] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In *International Conference on Machine Learning*, pages 21111–21132. PMLR, 2022.

[35] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems*, 34:5972–5984, 2021.

[36] Zhiqin Yang, Yonggang Zhang, Yu Zheng, Xinmei Tian, Hao Peng, Tongliang Liu, and Bo Han. Fedfed: Feature distillation against data heterogeneity in federated learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[37] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pages 12878–12889. PMLR, 2021.

[38] Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. Feddm: Iterative distribution matching for communication-efficient federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16323–16332, 2023.

[39] Jack Goetz and Ambuj Tewari. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489*, 2020.

[40] Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020.

[41] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola. Md-gan: Multi-discriminator generative adversarial networks for distributed datasets. In *2019 IEEE international parallel and distributed processing symposium (IPDPS)*, pages 866–877. IEEE, 2019.

[42] Mohammad Rasouli, Tao Sun, and Ram Rajagopal. Fedgan: Federated generative adversarial networks for distributed data. *arXiv preprint arXiv:2006.07228*, 2020.

[43] Wei Li, Jinlin Chen, Zhenyu Wang, Zhidong Shen, Chao Ma, and Xiaohui Cui. Ifl-gan: Improved federated learning generative adversarial network with maximum mean discrepancy model aggregation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10502–10515, 2022.

[44] Zijian Li, Jiawei Shao, Yuyi Mao, Jessie Hui Wang, and Jun Zhang. Federated learning with gan-based data synthesis for non-iid clients. In *International Workshop on Trustworthy Federated Learning*, pages 17–32. Springer, 2022.

[45] Huancheng Chen and Haris Vikalo. Federated learning in non-iid settings aided by differentially private synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5026–5035, 2023.

[46] Clare Elizabeth Heinbaugh, Emilio Luz-Ricca, and Huajie Shao. Data-free one-shot federated learning under very high statistical heterogeneity. In *The Eleventh International Conference on Learning Representations*, 2022.

[47] Huancheng Chen and Haris Vikalo. Federated learning in non-iid settings aided by differentially private synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5027–5036, 2023.

[48] Bangzhou Xin, Yangyang Geng, Teng Hu, Sheng Chen, Wei Yang, Shaowei Wang, and Liusheng Huang. Federated synthetic data generation with differential privacy. *Neurocomputing*, 468:1–10, 2022.

[49] Charlie Hou, Mei-Yu Wang, Yige Zhu, Daniel Lazar, and Giulia Fanti. Private federated learning using preference-optimized synthetic data. *arXiv preprint arXiv:2504.16438*, 2025.

[50] Shanshan Wu, Zheng Xu, Yanxiang Zhang, Yuanbo Zhang, and Daniel Ramage. Prompt public large language models to synthesize data for private on-device applications. *arXiv preprint arXiv:2404.04360*, 2024.

[51] Bo Li, Yasin Esfandiari, Mikkel N Schmidt, Tommy S Alstrøm, and Sebastian U Stich. Synthetic data shuffling accelerates the convergence of federated learning under data heterogeneity. *arXiv preprint arXiv:2306.13263*, 2023.

[52] Jacky Chung-Hao Wu, Hsuan-Wen Yu, Tsung-Hung Tsai, and Henry Horng-Shing Lu. Dynamically synthetic images for federated learning of medical images. *Computer Methods and Programs in Biomedicine*, 242:107845, 2023.

[53] Qi Chang, Zhennan Yan, Mu Zhou, Hui Qu, Xiaoxiao He, Han Zhang, Lohendran Baskaran, Subhi Al'Aref, Hongsheng Li, Shaoting Zhang, et al. Mining multi-center heterogeneous medical data with distributed synthetic learning. *Nature communications*, 14(1):5510, 2023.

[54] Wei Zhu and Jiebo Luo. Federated medical image analysis with virtual sample synthesis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 728–738. Springer, 2022.

[55] Jinbao Wang, Guoyang Xie, Yawen Huang, Jiayi Lyu, Feng Zheng, Yefeng Zheng, and Yaochu Jin. Fedmed-gan: Federated domain translation on unsupervised cross-modality brain image synthesis. *Neurocomputing*, 546:126282, 2023.

[56] Daiqing Li, Amlan Kar, Nishant Ravikumar, Alejandro F Frangi, and Sanja Fidler. Federated simulation for medical imaging. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part I 23*, pages 159–168. Springer, 2020.

[57] Shuai Li, Liang Hu, Chengyu Sun, Juncheng Hu, and Hongtu Li. Federated edge learning for medical image augmentation. *Applied Intelligence*, 55(1):56, 2025.

[58] Claire Little, Mark Elliot, and Richard Allmendinger. Federated learning for generating synthetic data: a scoping review. *International Journal of Population Data Science*, 8(1), 2023.

[59] Bangzhou Xin, Wei Yang, Yangyang Geng, Sheng Chen, Shaowei Wang, and Liusheng Huang. Private fl-gan: Differential privacy synthetic data generation based on federated learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2927–2931. IEEE, 2020.

[60] Sean Augenstein, H Brendan McMahan, Daniel Ramage, Swaroop Ramaswamy, Peter Kairouz, Mingqing Chen, Rajiv Mathews, et al. Generative models for effective ml on private, decentralized datasets. *arXiv preprint arXiv:1911.06679*, 2019.

[61] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *Advances in Neural Information Processing Systems*, 33:12673–12684, 2020.

[62] Monik Raj Behera, Sudhir Upadhyay, Suresh Shetty, Sudha Priyadarshini, Palka Patel, and Ker Farn Lee. Fedsyn: Synthetic data generation using federated learning. *arXiv preprint arXiv:2203.05931*, 2022.

[63] Matthijs de Goede, Bart Cox, and Jérémie Decouchant. Training diffusion models with federated learning. *arXiv preprint arXiv:2406.12575*, 2024.

[64] Jayneel Vora, Nader Bouacida, Aditya Krishnan, and Prasant Mohapatra. Feddm: Enhancing communication efficiency and handling data heterogeneity in federated diffusion models. *arXiv preprint arXiv:2407.14730*, 2024.

[65] Zihao Peng, Xijun Wang, Shengbo Chen, Hong Rao, Cong Shen, and Jinpeng Jiang. Federated learning for diffusion models. *IEEE Transactions on Cognitive Communications and Networking*, 2025.

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

[67] Kyeongkook Seo, Dong-Jun Han, and Jaejun Yoo. Prism: Privacy-preserving improved stochastic masking for federated generative models. *arXiv preprint arXiv:2503.08085*, 2025.

[68] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[69] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[70] Santosh Vempala and Andre Wibisono. Rapid convergence of the unadjusted langevin algorithm: Isoperimetry suffices. *Advances in neural information processing systems*, 32, 2019.

[71] Tim Salimans and Jonathan Ho. Should ebms model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*, 2021.

[72] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[73] 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*, pages 308–318, 2016.

[74] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[75] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30, 2009.

[76] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[77] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

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

[79] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[80] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, et al. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *Advances in Neural Information Processing Systems*, 35:5315–5334, 2022.

[81] 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.

[82] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32, 2019.

[83] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International conference on machine learning*, pages 7176–7185. PMLR, 2020.

[84] Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. *Advances in Neural Information Processing Systems*, 34:12480–12492, 2021.

[85] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *arXiv preprint arXiv:2210.09929*, 2022.

[86] Wei Deng, Qian Zhang, Yi-An Ma, Zhao Song, and Guang Lin. On convergence of federated averaging langevin dynamics. *arXiv preprint arXiv:2112.05120*, 2021.

[87] David Duvenaud, Jacob Kelly, Kevin Swersky, Milad Hashemi, Mohammad Norouzi, and Will Grathwohl. No mcmc for me: Amortized samplers for fast and stable training of energy-based models. In *International Conference on Learning Representations (ICLR)*, 2021.

[88] Yaxuan Zhu, Jianwen Xie, Yingnian Wu, and Ruiqi Gao. Learning energy-based models by cooperative diffusion recovery likelihood. *arXiv preprint arXiv:2309.05153*, 2023.

[89] Jiali Cui and Tian Han. Learning energy-based model via dual-mcmc teaching. *Advances in Neural Information Processing Systems*, 36:28861–28872, 2023.

[90] Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. 1996.

[91] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[92] Radford M Neal. Monte carlo implementation. *Bayesian learning for neural networks*, pages 55–98, 1996.

[93] Tejas Jayashankar, J Jon Ryu, and Gregory Wornell. Score-of-mixture training: Training one-step generative models made simple. *arXiv preprint arXiv:2502.09609*, 2025.

[94] Sanjam Garg, Aarushi Goel, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Guru-Vamsi Policharla, and Mingyuan Wang. Experimenting with zero-knowledge proofs of training. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 1880–1894, 2023.

[95] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR, 2023.

[96] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[97] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A practical guide, 1st ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[98] Accountability Act. Health insurance portability and accountability act of 1996. *Public law*, 104:191, 1996.

[99] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[100] Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. Layer-wised model aggregation for personalized federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10092–10101, 2022.

[101] Seok-Ju Hahn, Minwoo Jeong, and Junghye Lee. Connecting low-loss subspace for personalized federated learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 505–515, 2022.

[102] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

[103] Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

[104] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.

[105] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pages 12878–12889. PMLR, 2021.

[106] Ali Reza Ghavamipour, Fatih Turkmen, Rui Wang, and Kaitai Liang. Federated synthetic data generation with stronger security guarantees. In *Proceedings of the 28th ACM Symposium on Access Control Models and Technologies*, pages 31–42, 2023.

[107] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, Mingyuan Zhou, et al. Patch diffusion: Faster and more data-efficient training of diffusion models. *Advances in neural information processing systems*, 36:72137–72154, 2023.

[108] Xinlong He, Yang Xu, Sicong Zhang, Weida Xu, and Jiale Yan. Enhance membership inference attacks in federated learning. *Computers & Security*, 136:103535, 2024.

[109] 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.

[110] Djalil Chafaï and Joseph Lehec. Logarithmic sobolev inequalities essentials. *Accessed on*, page 4, 2024.

[111] Hong-Bin Chen, Sinho Chewi, and Jonathan Niles-Weed. Dimension-free log-sobolev inequalities for mixture distributions. *Journal of Functional Analysis*, 281(11):109236, 2021.

[112] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[113] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[114] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[115] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE, 2022.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] .

   Justification: We clearly mentioned our contribution (i.e., federated synthetic data generation through cooperative inference, without exchanging model parameters as in traditional federated learning) and scope (i.e., cross-silo FL setting) in abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes] .

   Justification: We secure a separate section to discuss limitation of our work and its potential remedy at the end.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes] .

Justification: We provided separate sections for detailed derivations and proofs in appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes] .

   Justification: We provided a code link and separate sections for experimental details and configurations in appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes] .

Justification: We only used publicly available datasets in our experiments, and we provided self-contained code implementations as a link.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes] .

Justification: We provided separate sections for experimental details and configurations in appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes] .

Justification: We faithfully report all results with reproducible details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes] .

   Justification: We provided separate sections for experimental details and configurations in appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes] .

   Justification: We sincerely understood and followed research ethics, as in NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes] .

    Justification: We described broader impacts statement at the end of the manuscript.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA] .

    Justification: We do not release any data or models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes] .

    Justification: We clearly described and cited related assets (data, models, previous findings).

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes] .

    Justification: We only used publicly available datasets in our experiments, and we provided code implementations as a link.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

    Justification: We do not use crowdsourcing or experiments with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: Our work did not require IRB approval and not involve any related risks.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA] .

    Justification: We did not use LLMs except for polishing some sentences and table formatting.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

# Appendix

## Table of Contents



## A Derivation of Gaussian Diffusion Models

Diffusion models are a class of generative models that aim to learn a data distribution $p_{\text{data}}(\boldsymbol{x}) \equiv q(\boldsymbol{x}_0)$ by learning to transform random Gaussian noise into original data through an iterative denoising process. In other words, the underlying Markov chain from the noise ($\boldsymbol{x}_T$) to the data ($\boldsymbol{x}_0$) defines diffusion models, and they are realized by two main processes: a forward process and a reverse process.

In the forward process, data $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$ is gradually perturbed over $T$ timesteps by adding Gaussian noise as

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \mathbf{I}_d),$$

where $\beta_t \in (0,1)$ controls the noise schedule, until $x_T \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$. Other constants satisfy $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{\tau=1}^{T} \alpha_\tau$. Thus, the forward process models $q(\boldsymbol{x}_1, ..., \boldsymbol{x}_T|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$.

To reiterate, as in Eq. (2), diffusion models have useful property that enables calculation of anytime marginal distribution in a closed form:

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

By training a parameterized deep network, diffusion models can denoise from noise to the data by approximating the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \approx q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$, through the reverse process as defined in Eq. (1). Thus, the reverse process models $p_{\boldsymbol{\theta}}(\boldsymbol{x}_0, ..., \boldsymbol{x}_T) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$.

With these two paired processes, diffusion models maximize the lower bound of log-likelihood defined as:

$$\begin{aligned}
\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0) &\geq \mathbb{E}_{q(\boldsymbol{x}_1,...,\boldsymbol{x}_T|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_0, ..., \boldsymbol{x}_T)}{q(\boldsymbol{x}_1, ..., \boldsymbol{x}_T|\boldsymbol{x}_0)}\right] \\
&= \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0) - \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{x}_1, ..., \boldsymbol{x}_T|\boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_1, ..., \boldsymbol{x}_T|\boldsymbol{x}_0)) \\
&= \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0) - \sum_{t=1}^{T} \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)),
\end{aligned}$$

where the decomposition is due to the Markov property of both forward and reverse processes.

From this, we can maximize the lower bound of log-likelihood by minimizing the sum of KL divergence terms instead:

$$\sum_{t=1}^{T} \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)).$$

Since both $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ and $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ are Gaussian, the KL divergence simplifies to a mean squared error between the true noise $\boldsymbol{\epsilon}$ and the estimated noise $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$. Hence, we have

$$\begin{aligned}
&\sum\nolimits_{t=1}^{T} \mathrm{D}_{\mathrm{KL}}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \\
&= \sum\nolimits_{t=1}^{T} a_t \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)} \left[ \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2 \right] \\
&=: \sum\nolimits_{t=1}^{T} a_t \mathcal{L}(\boldsymbol{\theta}, t),
\end{aligned}$$

where $a_t$ is a weight that is typically treated equal as $a_1 = ... = a_T = 1$ [9] for all time-dependent loss $\mathcal{L}(\boldsymbol{\theta}, t)$, which was defined in Eq. (3).

After the training is completed by optimizing the above composite loss $\mathcal{L}(\boldsymbol{\theta}) = \sum_{t=1}^{T} \mathcal{L}(\boldsymbol{\theta}, t)$, we can draw samples through ancestral sampling: starting from $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$ using $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$, due to the connection $\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) + \sqrt{\tilde{\beta}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$ from the reverse process. Note here that $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ is computed from the estimated noise $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$.

## B Connection of Energy Based Models and Diffusion Models

EBMs and diffusion models share a profound theoretical connection through denoising score matching in Eq. (5). This connection not only provides an alternative interpretation of diffusion models but also enables ULA in Eq. (8).

Again, EBMs model an unnormalized probability density of the form of:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{\exp(-\lambda f_{\boldsymbol{\theta}}(\boldsymbol{x}))}{Z_{\boldsymbol{\theta}}},$$

where $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}$ is the energy function parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$, $\lambda \in \mathbb{R}^+$ is a scale factor, and $Z_{\boldsymbol{\theta}} = \int_{\boldsymbol{x}\in\mathcal{X}} \exp(-\lambda f_{\boldsymbol{\theta}}(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}$ is the normalizing constant, which is typically intractable in practice.

Due to the intractable property of the normalizing constant $Z_{\boldsymbol{\theta}}$, direct computation of the likelihood is challenging. This necessitates alternative training methods using a score, defined as follows. The *score* is the gradient of the log-density as:

$$\nabla_{\boldsymbol{x}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = -\lambda \nabla_{\boldsymbol{x}} f_{\boldsymbol{\theta}}(\boldsymbol{x}).$$

To train EBMs, we use denoising score matching objective [69] in Eq. (5) to minimize the Fisher divergence between the score of a model's distribution and the score of a noise-perturbed data distribution:

$$q(\boldsymbol{x}_\sigma) = \int_{\boldsymbol{x}\in\mathcal{X}} q_\sigma(\boldsymbol{x}_\sigma|\boldsymbol{x}) p_{\text{data}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},$$

where the perturbation is realized as:

$$\boldsymbol{x}_\sigma = \boldsymbol{x} + \sigma\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d),$$

and $\sigma$ is the noise scale. Hence, the DSM objective is:

$$\mathcal{L}(\boldsymbol{\theta}, \sigma) = \mathbb{E}_{\boldsymbol{x}_\sigma\sim q(\boldsymbol{x}_\sigma|\boldsymbol{x}), \boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \left[ \|\nabla_{\boldsymbol{x}_\sigma} \log q(\boldsymbol{x}_\sigma|\boldsymbol{x}) - \nabla_{\boldsymbol{x}_\sigma} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_\sigma)\|_2^2 \right]$$

Rewriting $q(\boldsymbol{x}_\sigma|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}_\sigma; \boldsymbol{x}, \sigma^2\mathbf{I}_d)$, we can explicitly have that

$$\nabla_{\boldsymbol{x}_\sigma} \log q(\boldsymbol{x}_\sigma|\boldsymbol{x}) = -\frac{(\boldsymbol{x}_\sigma - \boldsymbol{x})}{\sigma^2} = -\sigma\boldsymbol{\epsilon}.$$

By substituting the EBM score as $\nabla_{\boldsymbol{x}_\sigma} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_\sigma) = -\lambda \nabla_{\boldsymbol{x}_\sigma} f_{\boldsymbol{\theta}}(\boldsymbol{x}_\sigma)$, the objective becomes equivalent (up to a constant) to Eq. (6) as:

$$\sigma^2 \mathcal{L}(\boldsymbol{\theta}, \sigma) = \mathbb{E}_{\boldsymbol{x}_\sigma\sim q(\boldsymbol{x}_\sigma|\boldsymbol{x}), \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)} \left[ \|\boldsymbol{\epsilon} - \sigma\lambda \nabla_{\boldsymbol{x}_\sigma} f_{\boldsymbol{\theta}}(\boldsymbol{x}_\sigma)\|_2^2 \right].$$

Diffusion models, as defined in Section A, optimize a similar objective. To reiterate, the objective of diffusion models is given as:

$$\mathcal{L}(\boldsymbol{\theta}, t) = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}} \left[ \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2 \right].$$

From this, we can easily draw an analogy with DSM objective, by replacing the $\sigma$ with a time-dependent noise scale $\sigma_t$, with the score interpretation as:

$$\nabla_{\boldsymbol{x}_{\sigma_t}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_{\sigma_t}) = -\lambda \nabla_{\boldsymbol{x}_{\sigma_t}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_{\sigma_t}) \equiv -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{\sigma_t}.$$

This connection shows that the noise prediction $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ in diffusion models directly corresponds to the score of an implicit EBM, if scaled by the noise level $\sigma_t$. Thus, diffusion models can be viewed as learning EBMs implicitly where the score is approximated by the noise prediction network parameterized by $\boldsymbol{\theta}$. This energy-based interpretation allows for alternative sampling methods in diffusion models, such as ULA in Eq. (8). Note that for the compositional generation, this requires the energy-based parameterization tricks of diffusion models, discussed in Section 3.3 and Section 3.4.

# C Proofs

## C.1 Proof of Lemma 4.3

*Proof.* In this proof, we need to show $\mathbb{E}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right] \leq 0$ from diffusion reverse process in Eq. (1). First, following [9], we equivalently define for variance schedule constants $\beta_t, t \in [T]$ that other constants are defined as follows.

$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{\tau=1}^{t} \alpha_\tau, \quad \tilde{\beta}_t = \beta_t \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}.$$

Recall that the reverse process in Eq. (1) can be written as

$$\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) + \tilde{\beta}_t \boldsymbol{z} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right) + \tilde{\beta}_t \boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d).$$

From this, we have from the law of total expectation that

$$\mathbb{E}\left[\|\boldsymbol{x}_{t-1}\|_2^2\right] = \mathbb{E}\left[\|\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)\|_2^2\right] + \tilde{\beta}_t d$$

For $\mathbb{E}\left[\|\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)\|_2^2\right]$, we have

$$\begin{aligned}
&\mathbb{E}\left[\|\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)\|_2^2\right] \\
&= \frac{1}{\alpha_t}\left(\mathbb{E}\left[\|\boldsymbol{x}_t\|_2^2\right] - \frac{2\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbb{E}\left[\langle \boldsymbol{x}_t, \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\rangle\right] + \frac{\beta_t^2}{1-\bar{\alpha}_t}\mathbb{E}\left[\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right]\right) \\
&\underset{\text{Definition 4.1}}{\approx} \frac{1}{\alpha_t}\left(\mathbb{E}\left[\|\boldsymbol{x}_t\|_2^2\right] - \frac{2\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbb{E}\left[\langle \boldsymbol{x}_t, \boldsymbol{\epsilon}\rangle\right] + \frac{\beta_t^2}{1-\bar{\alpha}_t}\mathbb{E}\left[\|\boldsymbol{\epsilon}\|_2^2\right]\right) \\
&= \frac{1}{\alpha_t}\left(\mathbb{E}\left[\|\boldsymbol{x}_t\|_2^2\right] - \frac{2\beta_t}{\sqrt{1-\bar{\alpha}_t}} \cdot \sqrt{1-\bar{\alpha}_t} d + \frac{\beta_t^2}{1-\bar{\alpha}_t} d\right) \\
&= \frac{1}{\alpha_t}\left(\mathbb{E}\left[\|\boldsymbol{x}_t\|_2^2\right] - 2\beta_t d + \frac{\beta_t^2}{1-\bar{\alpha}_t} d\right),
\end{aligned}$$

where we used $\mathbb{E}\left[\|\boldsymbol{\epsilon}\|_2^2\right] = d$ for any $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$ and $\mathbb{E}\left[\langle \boldsymbol{x}_t, \boldsymbol{\epsilon}\rangle\right] = \sqrt{1-\bar{\alpha}_t} d$ from Eq. (2). Thus, we have

$$\mathbb{E}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right] = \left(\frac{1}{\alpha_t} - 1\right)\mathbb{E}\left[\|\boldsymbol{x}_t\|_2^2\right] - \frac{2\beta_t d}{\alpha_t} + \frac{\beta_t^2 d}{\alpha_t(1-\bar{\alpha}_t)} + \tilde{\beta}_t d$$

From Eq. (2), we have

$$\mathbb{E}[\|\boldsymbol{x}_t\|^2] = \bar{\alpha}_t \mathbb{E}\left[\|\boldsymbol{x}_0\|_2^2\right] + (1-\bar{\alpha}_t) d,$$

as $\boldsymbol{x}_0$ and $\boldsymbol{\epsilon}$ are independent and $\mathbb{E}\left[\|\boldsymbol{\epsilon}\|_2^2\right] = d$.

Using this, we derive

$$
\begin{aligned}
&\mathbb{E}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right] \\
&= \left(\frac{1}{\alpha_t} - 1\right)\left(\bar{\alpha}_t \mathbb{E}\left[\|\boldsymbol{x}_0\|_2^2\right] + (1-\bar{\alpha}_t)d\right) - \frac{2\beta_t d}{\alpha_t} + \frac{\beta_t^2 d}{\alpha_t(1-\bar{\alpha}_t)} + \tilde{\beta}_t d \\
&\leq \left(\frac{1}{\alpha_t} - 1 - \frac{2\beta_t}{\alpha_t} + \frac{\beta_t^2}{\alpha_t(1-\bar{\alpha}_t)} + \tilde{\beta}_t\right) d \\
&= \left(\frac{1}{\alpha_t} - 1 - \frac{2\beta_t}{\alpha_t} + \frac{\beta_t^2}{\alpha_t(1-\bar{\alpha}_t)} + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right) d \\
&= \left(\frac{1-\alpha_t}{\alpha_t} - \frac{2\beta_t}{\alpha_t} + \frac{\beta_t^2}{\alpha_t(1-\bar{\alpha}_t)} + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right) d \\
&= \left(\frac{\beta_t}{\alpha_t} - \frac{2\beta_t}{\alpha_t} + \frac{\beta_t^2}{\alpha_t(1-\bar{\alpha}_t)} + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right) d \\
&= \left(-\frac{\beta_t}{\alpha_t} + \frac{\beta_t^2}{\alpha_t(1-\bar{\alpha}_t)} + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right) d \\
&= \frac{\beta_t}{\alpha_t(1-\bar{\alpha}_t)}\left(-(1-\bar{\alpha}_t) + \beta_t + \alpha_t(1-\bar{\alpha}_{t-1})\right),
\end{aligned}
$$

where the first inequality is due to $\|\boldsymbol{x}\|_2 \leq \sqrt{d}\|\boldsymbol{x}\|_\infty, \forall \boldsymbol{x} \in \mathbb{R}^d$, along with typical assumption in diffusion models that $\|\boldsymbol{x}_0\|_\infty = 1$ as inputs are normalized into $[-1,1]^d$ [9].

Rearranging, we have

$$
\begin{aligned}
&\mathbb{E}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right] \\
&\leq \frac{\beta_t}{\alpha_t(1-\bar{\alpha}_t)}\left(-(1-\bar{\alpha}_t) + \beta_t + \alpha_t(1-\bar{\alpha}_{t-1})\right) \\
&= \frac{\beta_t}{\alpha_t(1-\bar{\alpha}_t)}\left(-(1-\beta_t) + \alpha_t + \bar{\alpha}_t - \alpha\bar{\alpha}_{t-1})\right) \\
&= \frac{\beta_t}{\alpha_t(1-\bar{\alpha}_t)}\left(-\alpha_t + \alpha_t + \bar{\alpha}_t - \alpha\bar{\alpha}_{t-1})\right) \\
&= \frac{\beta_t}{\alpha_t(1-\bar{\alpha}_t)}\left(-\alpha_t + \alpha_t + \bar{\alpha}_t - \bar{\alpha}_t\right) \\
&= 0,
\end{aligned}
$$

where the second last and the third last equalities are due to the definition of $\bar{\alpha}_t$ and $\beta_t$ each.

We finally have $\mathbb{E}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right] \leq 0$, thus $\mathbb{E}\left[\|\boldsymbol{x}_{t-1}\|_2^2\right] \leq \mathbb{E}\left[\|\boldsymbol{x}_t\|_2^2\right]$. □

### C.2 Proof of Lemma 4.4

#### C.2.1 Proofs

*Proof.* In this proof, we need to show $\mathbb{E}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right] \leq 0$ from ULA update in Eq. (8).

With the energy-based $\ell_2$ parameterization in Eq. (11), denote from ULA update that

$$
\Delta\boldsymbol{x}_t := \boldsymbol{x}_{t-1} - \boldsymbol{x}_t = -\frac{\lambda\eta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) + \sqrt{2\eta_t}\boldsymbol{z}_t, \quad \boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d).
$$

With this, we have that

$$
\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2 = \|\boldsymbol{x}_t + \Delta\boldsymbol{x}_t\|_2^2 - \|\boldsymbol{x}_t\|_2^2 = 2\langle x_t, \Delta\boldsymbol{x}_t\rangle + \|\Delta\boldsymbol{x}_t\|_2^2.
$$

Now, taking expectations over $\boldsymbol{z}_t$ and $\boldsymbol{x}_t|\boldsymbol{x}_0$, we have that

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\mathbb{E}_{\boldsymbol{z}_t}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right]\right] \\
&= \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\mathbb{E}_{\boldsymbol{z}_t}\left[2\langle\boldsymbol{x}_t, \Delta\boldsymbol{x}_t\rangle + \|\Delta\boldsymbol{x}_t\|_2^2\right]\right] \\
&= 2\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\langle\boldsymbol{x}_t, \mathbb{E}_{\boldsymbol{z}_t}[\Delta\boldsymbol{x}_t]\rangle\right] + \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\mathbb{E}_{\boldsymbol{z}_t}[\|\Delta\boldsymbol{x}_t\|_2^2]\right].
\end{aligned} \tag{A1}
$$

Let us demystify the inner expectation first. Since $\mathbb{E}_{\boldsymbol{z}_t}[\boldsymbol{z}_t] = 0$, we have that

$$\mathbb{E}_{\boldsymbol{z}_t}[\Delta \boldsymbol{x}_t] = -\frac{\lambda \eta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t).$$

Next, for $\mathbb{E}_{\boldsymbol{z}_t}[\|\Delta \boldsymbol{x}_t\|_2^2]$, we have that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{z}_t}\left[\frac{\lambda^2 \eta_t^2}{1-\bar{\alpha}_t} \left\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right\|_2^2 - \frac{2\lambda \eta_t \sqrt{2\eta_t}}{\sqrt{1-\bar{\alpha}_t}} \langle \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \boldsymbol{z}_t \rangle + 2\eta_t \left\|\boldsymbol{z}_t\right\|^2\right] \\
&= \frac{\lambda^2 \eta_t^2}{1-\bar{\alpha}_t} \|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2 - \frac{2\lambda \eta_t \sqrt{2\eta_t}}{\sqrt{1-\bar{\alpha}_t}} \langle \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \mathbb{E}_{\boldsymbol{z}_t}[\boldsymbol{z}_t] \rangle + 2\eta_t \mathbb{E}_{\boldsymbol{z}_t}[\|\boldsymbol{z}_t\|_2^2] \\
&= \frac{\lambda^2 \eta_t^2}{1-\bar{\alpha}_t} \|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2 + 2\eta_t d,
\end{aligned}$$

where $\mathbb{E}[\|\boldsymbol{z}_t\|_2^2] = d$ for $\boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$.

To sum up, for the inner expectation of Eq. (A1), we have that

$$\mathbb{E}_{\boldsymbol{z}_t}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right] = -\frac{2\lambda \eta_t}{\sqrt{1-\bar{\alpha}_t}} \langle \boldsymbol{x}_t, \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \rangle + \frac{\lambda^2 \eta_t^2}{1-\bar{\alpha}_t} \|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2 + 2\eta_t d.$$

Going on for the outer expectation, we have that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\mathbb{E}_{\boldsymbol{z}_t}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right]\right] \\
&= -\frac{2\lambda \eta_t}{\sqrt{1-\bar{\alpha}_t}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\langle \boldsymbol{x}_t, \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \rangle\right] + \frac{\lambda^2 \eta_t^2}{1-\bar{\alpha}_t} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right] + 2\eta_t d.
\end{aligned}$$

Since it is for well-trained diffusion models, we have by using Eq. (2) that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\mathbb{E}_{\boldsymbol{z}_t}\left[\|\boldsymbol{x}_{t-1}\|_2^2 - \|\boldsymbol{x}_t\|_2^2\right]\right] \\
&= -\frac{2\lambda \eta_t}{\sqrt{1-\bar{\alpha}_t}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\left\langle \boldsymbol{x}_t, \frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}} \right\rangle\right] \\
&\quad + \frac{\lambda^2 \eta_t^2}{1-\bar{\alpha}_t} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\left\| \frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}} \right\|_2^2\right] + 2\eta_t d.
\end{aligned} \tag{A2}$$

From this, the first conditional expectation becomes that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\left\langle \boldsymbol{x}_t, \frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}} \right\rangle\right] \\
&= \frac{1}{\sqrt{1-\bar{\alpha}_t}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\left\langle \boldsymbol{x}_t, \boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 \right\rangle\right] = \frac{1}{\sqrt{1-\bar{\alpha}_t}} \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\|\boldsymbol{x}_t\|_2^2 - \sqrt{\bar{\alpha}_t}\langle \boldsymbol{x}_t, \boldsymbol{x}_0 \rangle\right],
\end{aligned}$$

where the former term inside the expectation is that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\|\boldsymbol{x}_t\|_2^2\right] &= \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\sum_{i=1}^d (x_i)^2\right] = \sum_{i=1}^d \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[(\boldsymbol{x}_i)^2\right] \\
&= \sum_{i=1}^d \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\mathrm{Var}[\boldsymbol{x}_{t,i}|\boldsymbol{x}_0] + (\mathbb{E}[\boldsymbol{x}_{t,i}|\boldsymbol{x}_0])^2\right] \\
&= \sum_{i=1}^d \left\{(1-\bar{\alpha}_t) + \left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_{0,i}\right)^2\right\} = (1-\bar{\alpha}_t)d + \bar{\alpha}_t \|\boldsymbol{x}_0\|_2^2,
\end{aligned}$$

and the second term inside the expectation is that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\langle \boldsymbol{x}_t, \boldsymbol{x}_0 \rangle\right] &= \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\sum_{i=1}^d \boldsymbol{x}_{0,i}\boldsymbol{x}_{t,i}\right] = \sum_{i=1}^d \boldsymbol{x}_{0,i}\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\boldsymbol{x}_{t,i}\right] \\
&= \sum_{i=1}^d \boldsymbol{x}_{0,i}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_{0,i}\right) = \sqrt{\bar{\alpha}_t}\sum_{i=1}^d \boldsymbol{x}_{0,i}^2 = \sqrt{\bar{\alpha}_t}\|\boldsymbol{x}_0\|_2^2.
\end{aligned}$$

Taken together, we have for the first conditional expectation that

$$\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\left\langle \boldsymbol{x}_t, \frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}}\right\rangle\right] = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\left((1-\bar{\alpha}_t)d + \bar{\alpha}_t\left\|\boldsymbol{x}_0\right\|_2^2 - \sqrt{\bar{\alpha}_t}\cdot\sqrt{\bar{\alpha}_t}\|\boldsymbol{x}_0\|_2^2\right) = \sqrt{1-\bar{\alpha}_t}d.$$

Next, for the second conditional expectation term in Eq. (A2), we have that

$$\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\left\|\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}}\right\|_2^2\right] = \mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\|\boldsymbol{\epsilon}\|_2^2\right] = d,$$

due to Eq. (2) and it is for well-trained diffusion models.

Putting all together, the original expectation in Eq. (A2) becomes that

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}_t|\boldsymbol{x}_0}\left[\mathbb{E}_{\boldsymbol{z}_t}\left[\left\|\boldsymbol{x}_{t-1}\right\|_2^2 - \left\|\boldsymbol{x}_t\right\|_2^2\right]\right] \\
&= -\frac{2\lambda\eta_t}{\sqrt{1-\bar{\alpha}_t}}\cdot\sqrt{1-\bar{\alpha}_t}d + \frac{\lambda^2\eta_t^2}{1-\bar{\alpha}_t}\cdot d + 2\eta_t d \\
&= \left(-2\lambda + \frac{\lambda^2\eta_t}{1-\bar{\alpha}_t} + 2\right)\eta_t d.
\end{aligned}$$

Since we want to guarantee this term to be non-increasing for the non-expansiveness w.r.t. L2 norm as in Lemma 4.3, we need to have that

$$\left(-2\lambda + \frac{\lambda^2\eta_t}{1-\bar{\alpha}_t} + 2\right)\eta_t d \leq 0$$

Due to $d > 0, \eta_t \geq 0$ and $\lambda > 0$, we have that

$$-2\lambda + \frac{\lambda^2\eta_t}{1-\bar{\alpha}_t} + 2 \leq 0 \Leftrightarrow \eta_t \leq \frac{2(\lambda-1)}{\lambda^2}\left(1-\bar{\alpha}_t\right).$$

To ensure $\eta_t \geq 0$, we should have $\lambda \geq 1$. From $\max_t \sqrt{1-\bar{\alpha}_t} = 1$, we can conservatively set

$$\eta_t \leq \frac{2(\lambda-1)}{\lambda^2}.$$

As $g(\lambda) = \frac{2(\lambda-1)}{\lambda^2}$ has its maximum in $\lambda \geq 1$ when $g(2) = \frac{1}{2}$, we have $\eta_t \in [0, \frac{1}{2}]$ when $\lambda = 2$. □

### C.3 Proof of Theorem 4.5

In this section, we present materials related to the proof of Theorem 4.5. For the convergence analysis, we adapt the assumptions and result of [70]. First, we introduce the essential definitions, then we provide the technical lemmas and present a proof of the main theorem. Note that these proofs demonstrate the exponential convergence of ULA under the minimal isoperimetric condition (i.e. the Log-Sobolev inequality), without the need for strict and often impractical assumptions such as log-concavity or boundedness of higher derivatives [70].

#### C.3.1 Definitions

**Definition C.1** (Kullback-Leibler (KL) divergence)**.** The Kullback-Leibler (KL) divergence of $p$ with respect to $q$ is defined as

$$\mathrm{D}_{\mathrm{KL}}(p \parallel q) = \int_{\boldsymbol{x}\in\mathcal{X}\subseteq\mathbb{R}^d} p(\boldsymbol{x})\log\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\mathrm{d}\boldsymbol{x}.$$

**Definition C.2** (Log-Sobolev Inequality (LSI))**.** A probability distribution $p$ satisfies the log-Sobolev inequality with a constant $\gamma > 0$ if for all smooth function $g : \mathbb{R}^d \to \mathbb{R}$ with $\mathbb{E}_p[g^2] < \infty$, and

$$\mathbb{E}_p[g^2\log g^2] - \mathbb{E}_p[g^2]\log\mathbb{E}_p[g^2] \leq \frac{2}{\gamma}\mathbb{E}_p[\|\nabla g\|^2].$$

### C.3.2 Technical Lemmas

In this section, we introduce the essential lemmas and corollaries required to prove the main theorem, i.e., Theorem 4.5. Note that we omit the proofs of adapted lemmas and refer to the original paper cited, i.e., Lemma C.3, Lemma C.7, and Lemma C.8 (for the adapted intermediate result).

**Lemma C.3** (Strong convexity and LSI; Corollary 5.11 of [110])**.** *Let $\mu$ be a probability measure on $\mathbb{R}^d$ of the form $\mathrm{d}\mu = \exp(-h(\boldsymbol{x}))\mathrm{d}\boldsymbol{x}$. If $h$ satisfies $\nabla^2_{\boldsymbol{x}} h(\boldsymbol{x}) \geq \gamma \mathbf{I}_d$ for some $\gamma > 0$ then $\mu$ satisfies the LSI with a constant $\gamma$.*

**Corollary C.4** (LSI of well-trained diffusion models with non-expansiveness guarantee)**.** *A well-trained diffusion model with L2 norm-driven energy-based reparameterization as in Eq. (11) and Lemma 4.4 satisfies LSI with constant $\frac{2}{1-\bar{\alpha}_t}$.*

*Proof.* For a well-trained diffusion model, we have $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \propto \exp(-\frac{\lambda}{2}\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2)$ from Eq. (11). With the property of well-trained diffusion models stated in Remark 4.2, we have that

$$\begin{aligned}
&\nabla^2_{\boldsymbol{x}_t}\left(\frac{\lambda}{2}\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right) \\
&= \nabla_{\boldsymbol{x}_t}\left(\frac{\lambda}{2}\nabla_{\boldsymbol{x}_t}\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2\right) = \nabla_{\boldsymbol{x}_t}\left(\lambda \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)^\intercal \nabla_{\boldsymbol{x}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right) = \nabla_{\boldsymbol{x}_t}\left(\frac{\lambda}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right) \\
&= \frac{\lambda}{\sqrt{1-\bar{\alpha}_t}}\nabla_{\boldsymbol{x}_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\lambda}{\sqrt{1-\bar{\alpha}_t}} \cdot \frac{1}{\sqrt{1-\bar{\alpha}_t}}\mathbf{I}_d = \frac{\lambda}{1-\bar{\alpha}_t}\mathbf{I}_d.
\end{aligned}$$

Due to Lemma 4.4 and Lemma C.3, the LSI constant $\gamma$ of well-trained diffusion models with L2 norm-driven energy-based reparameterization that guarantees non-expansiveness w.r.t. L2 norm is given as $\gamma = \frac{2}{1-\bar{\alpha}_t}$ since $\lambda = 2$. □

**Corollary C.5** (Lipschitz smoothness of an energy function of well-trained diffusion models with non-expansiveness guarantee)**.** *A well-trained diffusion model with L2 norm-driven energy-based reparameterization is $\frac{2}{1-\bar{\alpha}_t}$-Lipschitz smooth.*

*Proof.* It is directly implied from Corollary C.4. □

**Assumption C.6** (Bounded dissimilarity)**.** The pairwise chi-squared divergence between two different local distributions is uniformly bounded by $\kappa$, $\sup_{i\neq j\in[K]} \chi^2\left(p_i \parallel p_j\right) < \kappa < \infty$.

**Lemma C.7** (LSI constant of a mixture of distributions; Theorem 1 of [111])**.** *Denote a mixture of distributions $p^\star := \sum_{i=1}^K w_i p_i$ for $w_i \geq 0, \sum_{i=1}^K w_i = 1$, where each $p_i$ satisfies the LSI with $\gamma_i$. If Assumption C.6 holds, then $p^\star$ also satisfies LSI with a constant of*

$$\gamma^\star = \frac{\min_{i\in[K]}\gamma_i}{3(1+\kappa)(1+\log(1+\kappa))}$$

**Lemma C.8** (One-step contraction of ULA; Lemma 3 of [70])**.** *Let $\boldsymbol{x}_t \sim \tilde{p}_t$ be the output iterate one-step ULA. In one step, ULA can sample from a distribution $p_t \equiv p_{\boldsymbol{\theta}}(\cdot, t)$ encoded by a single well-trained diffusion model, satisfying*

$$\mathrm{D_{KL}}\left(\tilde{p}_{t+1} \parallel p_{t+1}\right) \leq \exp\left(\left(8\varsigma^2 - \frac{3}{2}\right)\gamma_t\eta_t\right)\mathrm{D_{KL}}\left(\tilde{p}_t \parallel p_t\right) + 6d\varsigma\eta_t, \tag{A3}$$

*with step size $0 < \eta_t \leq \frac{\varsigma\gamma_t}{L_t^{p+1}}$, where $L_t$ is Lipschitz smoothness constant, $\gamma_t$ is LSI constant, $p \geq 1$ and $0 < \varsigma < \frac{\sqrt{3}}{4}$.*

*Proof.* Consider the continuous interpolation $\tilde{p}_\tau$, where $\tau \in [0, \eta_t]$ with

$$\tilde{p}_{\tau=0} = \tilde{p}_t, \qquad \tilde{p}_{\tau=\eta_t} = \tilde{p}_{t+1}. \tag{A4}$$

Denote the LSI constant of a distribution encoded by well-trained diffusion models as $\gamma_t$ and the Lipschitz smoothness constant as $L_t$. For all $\tau \in [0, \eta_t]$, we can directly adapt the intermediate result of Lemma 3 of [70] as

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{D_{KL}}\left(\tilde{p}_\tau \parallel p_\tau\right) \leq -\frac{3\gamma_\tau}{2}\mathrm{D_{KL}}\left(\tilde{p}_\tau \parallel p_\tau\right) + \frac{4\tau^2 L_\tau^4}{\gamma_\tau}\mathrm{D_{KL}}\left(\tilde{p}_0 \parallel p_0\right) + 2d\tau^2 L_\tau^3 + 2d\tau L_\tau^2.$$

Denote

$$A_\tau := \frac{4\tau^2 L_\tau^4}{\gamma_\tau} \mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) + 2d\tau^2 L_\tau^3 + 2d\tau L_\tau^2,$$

and introduce the integrating factor as

$$\mu(\tau) = \exp\left(\frac{3}{2}\int_0^\tau \gamma_s \mathrm{d}s\right).$$

We wish to integrate over $\tau = 0$ to $\tau = \eta_t$, thus we have $\tau \leq \eta_t$. Further assume that for any $\tau \in [0, \eta_t]$ we have

$$\gamma_t \leq \gamma_\tau \leq L_\tau \leq L_t \tag{A5}$$

Then, we can upper bound as

$$A_\tau \leq \frac{4\eta_t^2 L_t^4}{\gamma_t} \mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) + 2d\eta_t^2 L_t^3 + 2d\eta_t L_t^2 := A_t, \tag{A6}$$

as it becomes irrelevant to $\tau$.

Then, we can rewrite the inequality as

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\left(\mu(\tau)\,\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_\tau \parallel p_\tau\right)\right) \leq \mu(\tau) A_t.$$

Integrating this inequality from $\tau = 0$ to $\tau = \eta_t$, we have that

$$\begin{aligned}
&\mu(\eta_t)\,\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_{\eta_t} \parallel p_{\eta_t}\right) - \mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) \\
&\leq A_t \int_0^{\eta_t} \mu(\tau)\mathrm{d}\tau = A_t \int_0^{\eta_t} \exp\left(\frac{3}{2}\int_0^\tau \gamma_s \mathrm{d}s\right)\mathrm{d}\tau
\end{aligned}$$

Rearranging, we have that

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_{\eta_t} \parallel p_{\eta_t}\right) \leq& \exp\left(-\frac{3}{2}\int_0^{\eta_t} \gamma_\tau \mathrm{d}\tau\right) \mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) \\
&+ A_t \exp\left(-\frac{3}{2}\int_0^{\eta_t} \gamma_\tau \mathrm{d}\tau\right) \int_0^{\eta_t} \exp\left(\frac{3}{2}\int_0^\tau \gamma_s \mathrm{d}s\right)\mathrm{d}\tau \\
\leq& \exp\left(-\frac{3}{2}\gamma_t \int_0^{\eta_t} \mathrm{d}\tau\right) \mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) \\
&+ A_t \exp\left(-\frac{3}{2}\gamma_t \int_0^{\eta_t} \mathrm{d}\tau\right) \int_0^{\eta_t} \exp\left(\frac{3}{2}L_t \tau\right)\mathrm{d}\tau \\
=& \exp\left(-\frac{3}{2}\gamma_t \eta_t\right) \mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) \\
&+ A_t \exp\left(-\frac{3}{2}\gamma_t \eta_t\right) \frac{2}{3L_t}\left(\exp\left(\frac{3}{2}L_t \eta_t\right) - 1\right),
\end{aligned}$$

where the second inequality is due to Eq. (A5).

Using the inequality that $e^c \leq 1 + 2c$ for $0 < c = \frac{3}{2}L_t\eta_t \leq 1$ (which holds due to the assumption that $\eta_t \leq \frac{c}{L_t^p} \leq \frac{2}{3L_t}$) along with Eq. (A6) we have that

$$\begin{aligned}
&\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_{\eta_t} \parallel p_{\eta_t}\right) \\
&\leq \exp\left(-\frac{3}{2}\gamma_t\eta_t\right) \mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) + A_t \exp\left(-\frac{3}{2}\gamma_t\eta_t\right) \cdot 2\eta_t \\
&= \exp\left(-\frac{3}{2}\gamma_t\eta_t\right)\left(1 + \frac{8\eta_t^3 L_t^4}{\gamma_t}\right) \mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) \\
&+ \exp\left(-\frac{3}{2}\gamma_t\eta_t\right)\left(4d\eta_t^3 L_t^3 + 4d\eta_t^2 L_t^2\right) \\
&\leq \exp\left(-\frac{3}{2}\gamma_t\eta_t\right)\left(1 + \frac{8\eta_t^3 L_t^4}{\gamma_t}\right) \mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) + \left(4d\eta_t^3 L_t^3 + 4d\eta_t^2 L_t^2\right),
\end{aligned}$$

where the last inequality is due to $\exp\left(-\frac{3}{2}\gamma_t\eta_t\right) \le 1$.

Using the assumption that $\eta_t \le \frac{\varsigma\gamma_t}{L_t^{p+1}} \le \frac{\varsigma}{L_t^p}$, we have

$$1 + \frac{8\eta_t^3 L_t^4}{\gamma_t} \le 1 + \frac{8\varsigma\eta_t^2 L_t^{4-p}}{\gamma_t} \le 1 + 8\varsigma^2\gamma_t\eta_t \le \exp\left(8\varsigma^2\gamma_t\eta_t\right).$$

Thus, the inequality above becomes that

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_{\eta_t} \parallel p_{\eta_t}\right) &\le \exp\left(-\frac{3}{2}\gamma_t\eta_t\right)\exp\left(8\varsigma^2\gamma_t\eta_t\right)\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) + 4d\eta_t^3 L_t^3 + 4d\eta_t^2 L_t^2 \\
&= \exp\left(-\frac{3}{2}\gamma_t\eta_t\right)\exp\left(8\varsigma^2\gamma_t\eta_t\right)\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) + 4d\eta_t^2 L_t^2(\eta_t L_t + 1) \\
&= \exp\left((8\varsigma^2 - \gamma_t)\eta_t\right)\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) + 4d\eta_t^2 L_t^2(\eta_t L_t + 1)
\end{aligned}$$

As $\eta_t \le \frac{\varsigma}{L_t^p} \le \frac{1}{2L_t}$ for $p \ge 1$, we have $\eta_t L_t \le \frac{1}{2}$ and $\eta_t L_t^2 \le \varsigma$

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_{\eta_t} \parallel p_{\eta_t}\right) &\le \exp\left(\left(8\varsigma^2 - \frac{3}{2}\right)\gamma_t\eta_t\right)\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) + 4d\eta_t^2 L_t^2(\eta_t L_t + 1) \\
&\le \exp\left(\left(8\varsigma^2 - \frac{3}{2}\right)\gamma_t\eta_t\right)\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_0 \parallel p_0\right) + 6d\varsigma\eta_t.
\end{aligned}$$

Finally, replacing with Eq. (A4), we finally have that

$$\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_{t+1} \parallel p_{t+1}\right) \le \exp\left(\left(8\varsigma^2 - \frac{3}{2}\right)\gamma_t\eta_t\right)\mathrm{D}_{\mathrm{KL}}\left(\tilde{p}_t \parallel p_t\right) + 6d\varsigma\eta_t. \tag{A7}$$

□

**Lemma C.9** (Convergence of ULA in KL divergence)**.** *Let $p_t \equiv p_{\boldsymbol{\theta}}(\cdot, t)$ be the probability distribution defined by a single well-trained diffusion model and let $B_\varsigma = \frac{3}{2} - 8\varsigma^2 > 0$. Assume that the iterates $\boldsymbol{x}_t \sim \tilde{p}_t$ are generated by the Unadjusted Langevin Algorithm (ULA) in Eq. (8), and that $\mathrm{D}_{\mathrm{KL}}(\tilde{p}_0 \parallel p_0) < \infty$. Then, for all $t \ge 0$, we have*

$$\mathrm{D}_{\mathrm{KL}}(\tilde{p}_t \parallel p_t) \le \exp\left(-B_\varsigma t\gamma_t\eta_0\right)\mathrm{D}_{\mathrm{KL}}(\tilde{p}_0 \parallel p_0) + \frac{9d\varsigma\eta_t}{B_\varsigma\gamma_t\eta_0}. \tag{A8}$$

*Hence, for any $\delta \ge \frac{18d\varsigma}{B_\varsigma\gamma_t}$, it suffices to run ULA for*

$$t \ge \frac{1}{B_\varsigma\gamma_t\eta_0}\log\left(\frac{2\,\mathrm{D}_{\mathrm{KL}}(\tilde{p}_0 \parallel p_0)}{\delta}\right)$$

*steps with step size*

$$\eta_t \le \min\left\{\frac{B_\varsigma\gamma_t\eta_0\delta}{18d\varsigma}, \frac{\varsigma\gamma_t}{L_t^{p+1}}\right\},$$

*for $p \ge 1$ and LSI constant $\gamma_t$, in order to guarantee $\mathrm{D}_{\mathrm{KL}}(\tilde{p}_t \parallel p_t) \le \delta$.*

*Proof.* From Lemma C.8, recursively applying Eq. (A7) gives

$$\mathrm{D}_{\mathrm{KL}}(\tilde{p}_t \parallel p_t) \le \exp\left(-B_\varsigma\sum_{s=0}^{t-1}\gamma_s\eta_s\right)\mathrm{D}_{\mathrm{KL}}(\tilde{p}_0 \parallel p_0) + 6d\varsigma\sum_{r=0}^{t-1}\eta_r\exp\left(-B_\varsigma\sum_{s=r+1}^{t-1}\gamma_s\eta_s\right).$$

Since we have $\eta_t \ge \cdots \ge \eta_0$ and $\gamma_0 \ge \cdots \ge \gamma_t$, we can bound that

$$\sum_{s=0}^{t-1}\gamma_s\eta_s \ge t\gamma_t\eta_0, \qquad \sum_{s=r+1}^{t-1}\gamma_s\eta_s \ge (t-r-1)\gamma_t\eta_0.$$

Because $B_\varsigma > 0$, we get:

$$\mathrm{D_{KL}}(\tilde{p}_t \parallel p_t) \leq \exp(-B_\varsigma t\gamma_t\eta_0)\,\mathrm{D_{KL}}(\tilde{p}_0 \parallel p_0) + 6d\varsigma \sum_{r=1}^{t} \eta_t \exp(-B_\varsigma r\gamma_t\eta_0).$$

The remaining sum is for a geometric series, thus

$$\begin{aligned}\sum_{r=1}^{t} \exp(-B_\varsigma r\gamma_t\eta_0) &= \exp(-B_\varsigma\gamma_t\eta_0) \cdot \frac{1-\exp(-B_\varsigma\gamma_t\eta_0 t)}{1-\exp(-B_\varsigma\gamma_t\eta_0)} \\ &\leq \frac{1}{1-\exp(-B_\varsigma\gamma_t\eta_0)} \leq \frac{3}{2B_\varsigma\gamma_t\eta_0},\end{aligned}$$

where the last inequality uses

$$\frac{2c}{3} \leq 1 - e^{-c}, \quad 0 < c = B_\varsigma\gamma_t\eta_0 \leq \varsigma < \tfrac{\sqrt{3}}{4},$$

from Lemma C.8.

Thus,

$$\mathrm{D_{KL}}(\tilde{p}_t \parallel p_t) \leq \exp(-B_\varsigma t\gamma_t\eta_0)\,\mathrm{D_{KL}}(\tilde{p}_0 \parallel p_0) + \frac{9d\varsigma\eta_t}{B_\varsigma\gamma_t\eta_0}.$$

To ensure $\mathrm{D_{KL}}(\tilde{p}_t \parallel p_t) \leq \delta$, it suffices to assume:

$$\frac{9d\varsigma\eta_t}{B_\varsigma\gamma_t\eta_0} \leq \tfrac{\delta}{2}, \quad \exp(-B_\varsigma t\gamma_t\eta_0)\,\mathrm{D_{KL}}(\tilde{p}_0 \parallel p_0) \leq \tfrac{\delta}{2},$$

which hold when

$$\eta_t \leq \frac{B_\varsigma\gamma_t\eta_0\delta}{18d\varsigma} \quad \text{and} \quad t \geq \frac{1}{B_\varsigma\gamma_t\eta_0} \log\left(\frac{2\,\mathrm{D_{KL}}(\tilde{p}_0 \parallel p_0)}{\delta}\right).$$

□

### C.3.3 Proof of Theorem 4.5

Denote $p_{ti} \equiv p_{\boldsymbol{\theta}_i}(\cdot, t)$ as a distribution encoded by a locally-trained diffusion model of client $i$. For the mixture of distribution $p_t^\star = \sum_{i=1}^K w_i p_{ti}$, it is trivial that the energy function of $p_t^\star$ is $L_t^\star = \frac{2}{1-\bar{\alpha}_t}$-Lipschitz smooth since each local distribution is Lipschitz smooth due to Corollary C.5.

From the result of Lemma C.7, the LSI constant of the mixture is $\gamma^\star = \frac{\min_{i\in[K]}\gamma_i}{3(1+\kappa)(1+\log(1+\kappa))}$. As each local distribution has LSI constant $\gamma_{ti} = \frac{2}{1-\bar{\alpha}_t}$, we can further refine as

$$\gamma_t^\star = \frac{2}{3(1-\bar{\alpha}_t)(1+\kappa)(1+\log(1+\kappa))}.$$

Denote $0 < \tilde{\kappa} = \frac{2}{3(1+\kappa)(1+\log(1+\kappa))} < \frac{2}{3}$, we set the LSI constant as $\gamma_t^\star = \frac{\tilde{\kappa}}{1-\bar{\alpha}_t}$.

From the result of Lemma C.9, we finally have that

$$\begin{aligned}\mathrm{D_{KL}}(\tilde{p}_t^\star \parallel p_t^\star) &\leq \exp\left(-B_\varsigma t\gamma_t^\star\eta_0\right)\mathrm{D_{KL}}(\tilde{p}_0^\star \parallel p_0^\star) + \frac{9d\varsigma\eta_t}{B_\varsigma\gamma_t^\star\eta_0} \\ &= \exp\left(-\frac{B_\varsigma\tilde{\kappa}\eta_0 t}{1-\bar{\alpha}_t}\right)\mathrm{D_{KL}}(\tilde{p}_0^\star \parallel p_0^\star) + \frac{9d\varsigma\eta_t(1-\bar{\alpha}_t)}{B_\varsigma\tilde{\kappa}\eta_0}\end{aligned}$$

with step size

$$\eta_t \leq \min\left\{\frac{B_\varsigma\gamma_t^\star\eta_0\delta}{18d\varsigma}, \frac{\varsigma\gamma_t^\star}{(L_t^\star)^{p+1}}\right\} = \min\left\{\frac{B_\varsigma\tilde{\kappa}\eta_0\delta}{18d\varsigma(1-\bar{\alpha}_t)}, \frac{\varsigma\tilde{\kappa}(1-\bar{\alpha}_t)^p}{2^p}\right\}.$$

In practice, however, we cannot directly quantify $\bar{\kappa}$. Thus, we instead manually adjust a constant $\rho := \varsigma\bar{\kappa} < \frac{\sqrt{3}}{6}$. Further denote $\upsilon := B_\varsigma\tilde{\kappa}\eta_0$.

Finally, we have that

$$\mathrm{D_{KL}}(\tilde{p}_t^\star \parallel p_t^\star) \leq \exp\left(-\frac{\upsilon t}{1-\bar{\alpha}_t}\right) \mathrm{D_{KL}}(\tilde{p}_0^\star \parallel p_0^\star) + \frac{9d\varsigma\eta_t(1-\bar{\alpha}_t)}{\upsilon},$$

with step size

$$\eta_t \leq \min\left\{\frac{\upsilon\delta}{18d\varsigma(1-\bar{\alpha}_t)}, \frac{\rho(1-\bar{\alpha}_t)^p}{2^p}\right\},$$

for any $\delta \geq \frac{18d\varsigma\eta_0(1-\bar{\alpha}_t)}{\upsilon}$ in $t \geq \frac{1-\bar{\alpha}_t}{\upsilon}\log\left(\frac{2\,\mathrm{D_{KL}}(\tilde{p}_0\|p_0)}{\delta}\right)$ steps. Finally, replacing $0 \to T$ and $t \to T - t$ for the compatibility with ULA reaches the theorem statement.

# D Experimental Details

**Specification.** We conduct all experiments in a single server with Intel® Xeon® Gold 6226R CPU (@ 2.90GHz) and a single NVIDIA® Ampere® A100 GPU (w/ 40GB VRAM). For the implementation of diffusion models, we resort to `diffusers` [112] library using `PyTorch` [113].

**Simulation of Statistical Heterogeneity.** For the faithful evaluation of practical FL setting, we simulate non-IID data split to $K = 10$ clients for all benchmark datasets.

For MNIST dataset, we use Dirichlet distribution with concentration parameter $\alpha = 0.1$, following the setting of [79].

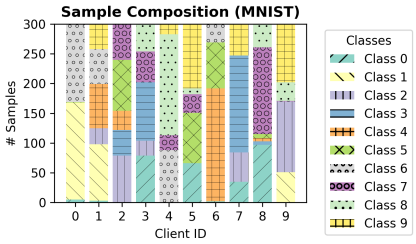


Figure A1: Non-IID local distributions of MNIST dataset

For CIFAR-10 dataset, we follow the setting of [21] using log-normal distribution with location=0 and scale=2.

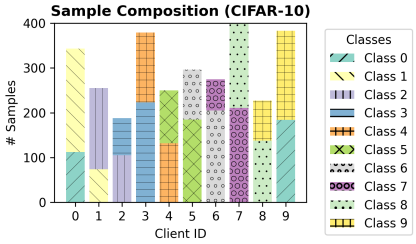


Figure A2: Non-IID local distributions of CIFAR-10 dataset

For CelebA dataset, which has 40 different attributes, we first construct classes by combining gender (male/female), smiling (0/1), and eyeglasses (0/1) attributes, i.e., 8 classes as a result. We randomly distribute samples to clients so that they have only three distinct classes.

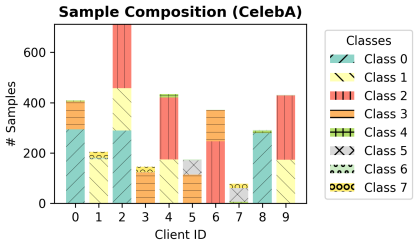


Figure A3: Non-IID local distributions of CelebA dataset

**Model and Training Hyperparameters.** We summarize detailed configurations of models used for experiments in Table A1.

Table A1: Model and Training Configurations.

| | MNIST | CIFAR-10 | CelebA |
|---|---|---|---|
| *Model Configuration* | | | |
| Spatial dimension | | $32 \times 32$ | |
| Attention resolution | | $8 \times 8$ | |
| Base channels | | 128 | |
| Channel multipliers | 1, 1, 1, 1 | 1, 2, 2, 2 | |
| Model size | 44.77MB | 136.38MB | 136.38MB |
| Base architecture | | DDPM [9] | |
| Scheduling scheme | | linear scheduling [9] | |
| *Training Configuration* | | | |
| Optimizer | | Adam [114] | |
| Learning rate | | $2 \times 10^{-4}$ | |