FINITE-SAMPLE TRAINING, MEMORIZATION, AND GENERALIZATION IN TRANSFORMER DDPMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models can either generate novel samples or reproduce data from their training samples, exhibiting different generalization and memorization behaviors. However, the theoretical mechanisms underlying these behaviors remain poorly understood. Specifically, it is unclear how finite-sample training shapes the learned denoiser and when it leads to memorization. To the best of our knowledge, this paper provides the first finite-sample convergence analysis of Transformer-based diffusion models trained by gradient descent and a theoretical characterization of the resulting sample-specific memorization performance. We consider learning data from a Gaussian mixture distribution under the Denoising Diffusion Probabilistic Model (DDPM) objective with a one-layer single-head Transformer model. We show that gradient descent learns the empirical mode means and within-mode variance, driving the training loss close to the empirical oracle denoising risk, while the generalization gap diminishes with the sample size. We further derive an explicit Gaussian-CDF characterization of a reconstruction-based sample-specific memorization rate, which yields a sample complexity threshold separating sample-specific memorization from generalization. A more fundamental investigation uncovers an *empirical-prototype attraction* mechanism underlying memorization in diffusion models, i.e., the trained denoiser interpolates between the scaled noisy input and a learned empirical mode prototype, which transitions from an individual training sample to the population mean as the number of samples increases. Numerical experiments validate these theoretical findings.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021) have become one of the dominant paradigms for generative modeling, exhibiting remarkable capabilities across a wide range of domains, including image and video synthesis (Saharia et al., 2022; Rombach et al., 2022; Ho et al., 2022; Liu et al., 2024), language and text generation (Kong et al., 2021; Zhang et al., 2023; Arriola et al., 2025), and scientific data creation (Watson et al., 2023; Li et al., 2024; Price et al., 2025). A classical diffusion model consists of a forward process and a reverse process. The forward process gradually transforms data into noise by adding Gaussian white noise, while the reverse process generates new samples from noisy inputs. This denoising objective is typically achieved by training a neural network to reconstruct clean data from noisy inputs. However, in some cases, a trained diffusion model reproduces samples from the training dataset during generation, a phenomenon known as *memorization* of the training data (Somepalli et al., 2023; Wen et al., 2024; Kadkhodaie et al., 2024; Gu et al., 2025). In many other cases, the model generates novel and diverse samples that are not present in the training set, a behavior referred to as *generalization* (Kadkhodaie et al., 2024; Zhang et al., 2024; Niedoba et al., 2025).

Understanding when diffusion models memorize training data versus generate novel samples is essential for revealing their underlying mechanism and addressing privacy and copyright concerns (Somepalli et al., 2023; Carlini et al., 2023; Wang et al., 2026b). Some empirical studies develop methods to detect and understand memorization in diffusion models (Wen et al., 2024; Ross et al., 2025; Zhang et al., 2026). However, theoretical analyses of the mechanisms underlying memorization and generalization remain limited due to the difficulty of analyzing the nonconvex optimization dynamics of neural network training. Existing theoretical work has partially investigated this problem. One line of research (Buchanan et al., 2025; Wang et al., 2026a; Zhang et al., 2026; Li et al.,

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

2025b; Ye et al., 2026) mainly focuses on characterizing the optimal denoiser or its sample complexity, but does not explain how neural networks trained by gradient descent evolve toward convergence under finite samples. Another line of work (Han et al., 2024; 2025; Bonnaire et al., 2025) analyzes training dynamics of diffusion models but relies on simplified network architectures. These analyses cannot be extended to diffusion models based on general nonlinear neural networks such as Transformers (Peebles & Xie, 2023) and do not provide a theoretical quantification of model memorization. Please see Section 1.2 for a more detailed comparison between our work and existing papers, and Table 1 for a brief summary. This motivates the following fundamental question:

When and how can a Transformer-based diffusion model memorize and generalize under finite-sample gradient-based training?

Theoretical Works	Model Architecture	Convergence Analysis	Optimization Target	Memorization Mechanism
Bonnaire et al. (2025)	RFNN	✗	N/A	Spectral bias
Buchanan et al. (2025)	GMM denoiser	✗	Global min.	N/A
Wang et al. (2026a)	U-Net	✗	Global min.	N/A
Zhang et al. (2026)	ReLU	✗	Local min.	Data storing
Ye et al. (2026)	ReLU	✗	Emp. score	Stat. separation
Ours	Transformer	✓	EODR	EPA

Table 1: Comparison with prior works on memorization and generalization of diffusion models.

1.1 Main Contributions

To address this question, we develop a finite-sample theory for Transformer-based diffusion models that jointly characterizes optimization, generalization, and sample-specific memorization. We consider learning Gaussian mixture data under the Denoising Diffusion Probabilistic Model (DDPM) objective with a one-layer, single-head Transformer whose architecture is motivated by the posterior mean of the mixture distribution. We define *sample-specific memorization* as the ability of the model to reconstruct an individual training sample from its diffused noisy version. Our analysis characterizes how gradient descent learns the empirical mode mean and variance features and the resulting impact on the memorization and generalization behavior. Our main contributions are:

1. **Finite-sample analysis of optimizing the DDPM loss with Transformers towards convergence.** We theoretically investigate the training dynamics of a one-layer single-head Transformer with softmax attention and quantify the required number of samples and iterations to optimize the DDPM objective. Our results show that, under proper initialization of the learnable prototype parameters, the finite-sample DDPM loss converges to the *empirical oracle denoising risk (EODR)* up to a small error, while the generalization gap vanishes as the sample complexity increases.
2. **Sample complexity analysis of sample-specific memorization.** Memorization is a sample-level probabilistic event that requires a distributional characterization of the finite-sample denoising error. We explicitly characterize the memorization rate as a Gaussian-CDF of the reconstruction tolerance in the high-dimensional regime. Calibrating the tolerance by the oracle denoising risk further yields a critical sample complexity that is proportional to the number of modes and the square root of dimension, separating sample-specific memorization from generalization.
3. **A theoretical characterization of the empirical-prototype attraction mechanism underlying memorization.** A major difficulty in understanding memorization is to characterize what finite-data denoisers learn and how such information is reflected by the generation. We are the first to propose the *empirical-prototype attraction (EPA) mechanism*, i.e., the trained Transformer denoiser approximately interpolates between the scaled noisy input and a learned empirical prototype, which converges to the empirical mean of the corresponding mode. When a mode contains a single training sample, this prototype coincides with that sample, reducing the mechanism to direct training sample attraction (Somepalli et al., 2023; Zhang et al., 2026). As the number of samples per mode increases, the empirical prototype approaches the population mean, continuously transitioning from sample-specific memorization toward generalization.

1.2 Related Works

Theoretical studies of diffusion models. Some recent works analyze memorization and generalization or the phase transition between them in diffusion models through loss landscape analysis

or training dynamics analysis of simplified models (Buchanan et al., 2025; Han et al., 2024; 2025; Bonnaire et al., 2025; Sclocchi et al., 2025; Li et al., 2025b; Wang et al., 2026a; George et al., 2026; Zhang et al., 2026; Ye et al., 2026). However, they do not provide a convergence analysis for Transformer-based diffusion models trained with finite samples. Only Li et al. (2026) investigates the training dynamics of Transformer-based diffusion models to near optimal with a characterization of the denoising mechanism. However, their analysis does not consider the finite-sample setting and therefore cannot be extended to study memorization. Other works study the convergence guarantees of DDPM samplers rather than model training (Li et al., 2023b; Azangulov et al., 2024; Li & Yan, 2024; Liang et al., 2025; Huang et al., 2026), which is beyond the scope of our work.

Optimization and generalization analysis of Transformer models. There are a series of works that study the optimization and generalization of transformers for supervised learning tasks. Jelassi et al. (2022); Li et al. (2023a); Jiang et al. (2024); Li et al. (2025a) study the convergence of Transformer by gradient-based training with a generalization guarantee for simple tasks, such as binary classification or linear regression tasks. Tian et al. (2023); Oymak et al. (2023); Tarzanagh et al. (2023); Nichani et al. (2024); Huang et al. (2025) investigate the loss landscape or training dynamics of Transformers by assuming specific input structure or a proper prediction head. None of these works involves finite-sample convergence analysis of Transformer-based diffusion models.

2 Problem Formulation

Data distribution. Following (Wang et al., 2026a; Li et al., 2026), consider each data point $\boldsymbol{x}_0 \in \mathcal{R}^d$ is sampled i.i.d. from a *mixture of Gaussian (MoG)* distribution $\mathcal{D}(\boldsymbol{\pi}, M, \{\boldsymbol{\mu}_m\}_{m=1}^M, \rho)$ with M modes, such that

$$\boldsymbol{x}_0 | Y = m \sim \mathcal{N}(\boldsymbol{\mu}_m, \rho^2 \boldsymbol{I}), \quad Y \sim \text{Categorical}_M(\boldsymbol{\pi}), \tag{1}$$

where $\{\boldsymbol{\mu}_i \in \mathbb{R}^d\}_{i=1}^M$ is a set of M orthogonal *patterns*, i.e., $\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j = d\delta_{ij}$; $\boldsymbol{\pi} \in \Delta^{M-1}$ is a vector in the probability simplex, where we choose $\pi_m = 1/M$ for any $m \in [M]$ for simplicity in this work; and $\rho^2 = \Theta(1)$ is the within-mode variance.

Learning model. We denote $\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}, t) \in \mathbb{R}^d$ as the output of the learning model, where $\boldsymbol{\Theta}$ is the set of model parameters, $\boldsymbol{x}$ is the input to the model, and t is the diffusion time step. The model architecture is motivated by a parametric model of the data distribution, where the mode means and shared variance are unknown and are learnable parameters. The diffusion forward process together with Tweedie's formula (Efron, 2011) gives a closed-form posterior-mean estimator whose computation can be implemented by a one-layer single-head Transformer. The detailed derivation is in Appendix D.1. Mathematically, given $\boldsymbol{\Theta} = \{\{\lambda_t \in \mathbb{R}\}_{t=1}^T, \{\boldsymbol{z}_m \in \mathbb{R}^d\}_{m=1}^M\}$, $\boldsymbol{f}$ is formulated as

$$\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}, t) = \frac{1 - \lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}} \boldsymbol{x} + (1-\bar{\alpha}_t)\lambda_t \sum_{m=1}^M \text{softmax}\left(-\frac{\lambda_t}{2}\|\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2\right) \boldsymbol{z}_m, \tag{2}$$

where $\text{softmax}(-\lambda_t\|\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2/2) = e^{-\lambda_t\|\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2/2} / \sum_{j=1}^M e^{-\lambda_t\|\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2/2}$.

Remark 1. *Eq (2) contains a residual connection and a Gaussian-kernel attention module, where $\boldsymbol{x}$ serves as the query, $\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m$ as the keys, and $\boldsymbol{z}_m$ as the values. If one removes the shared $-\lambda_t\|\boldsymbol{x}\|^2/2$ by expanding the softmax, the attention logits become $\lambda_t(\sqrt{\bar{\alpha}_t}\boldsymbol{x}^\top \boldsymbol{z}_m - \bar{\alpha}_t\|\boldsymbol{z}_m\|^2/2)$, i.e., a scaled dot-product attention with a key-dependent bias. Each $\boldsymbol{z}_m$ parameterizes a representative vector for mode m. We therefore refer to $\boldsymbol{z}_m$ as a learnable mode prototype. The parameter λ_t represents the inverse scale of the mode variance at time step t. Since the variance is the same across all modes in the data model (1), we consider a shared λ_t across all modes at each time step.*

Training objective and algorithm. We use the training objective of the Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020). For a given time step $t \in [T]$ and an input data point $\boldsymbol{x}_0$, we sample $\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t$, where $\boldsymbol{\epsilon}_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \boldsymbol{I}_d)$ is the additive white Gaussian noise, and $\{\bar{\alpha}_t\}_{t=1}^T$ is the pre-determined noise scheduling coefficients. Given N training samples $\mathcal{R} = \{\boldsymbol{x}_0^n\}_{n=1}^N$ with each $\boldsymbol{x}_0^i \sim \mathcal{D}(\boldsymbol{\pi}, M, \{\boldsymbol{\mu}_i\}_{i=1}^M, \rho)$ and time step $t \sim \text{Unif}([T])$. We then minimize the following DDPM loss as introduced in Ho et al. (2020); Bonnaire et al. (2025):

$$L_{N,T}(\boldsymbol{\Theta}) := \frac{1}{2NT} \sum_{t=1}^T \sum_{n=1}^N \|\boldsymbol{f}(\boldsymbol{\Theta}; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0^n + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n, t) - \boldsymbol{x}_0^n\|^2. \tag{3}$$

Our analysis considers gradient descent algorithm on the learning objective (3) with positive learning rates η_z for $\boldsymbol{z}_m$, and η_λ for λ_t. We randomly initialize each ${\lambda_t}^{(0)}$, $t \in [T]$, from $\mathcal{N}(0, 1/d)$. The

initialization of $\boldsymbol{z}_m$ needs to satisfy Assumption 1. Starting from the initialization $\boldsymbol{z}_m^{(0)}$ for $m \in [M]$ and λ_t for $t \in [T]$, for each training iteration s, the parameters are updated as follows:

$$\boldsymbol{z}_m^{(s+1)} = \boldsymbol{z}_m^{(s)} - \eta_z \nabla_{\boldsymbol{z}_m} L_{N,T}(\boldsymbol{\Theta}), \quad \lambda_t{}^{(s+1)} = \lambda_t{}^{(s)} - \eta_\lambda \nabla_{\lambda_t} L_{N,T}(\boldsymbol{\Theta}), \forall t \in [T]. \tag{4}$$

Generalization and Memorization. Given $\boldsymbol{x} \sim \mathcal{D}(\boldsymbol{\pi}, M, \{\boldsymbol{\mu}_i\}_{i=1}^M, \rho)$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$, we first define the following population risk $\bar{L}$ to evaluate the performance of learning the data distribution.

$$\bar{L}(\boldsymbol{\Theta}) := \frac{1}{2T} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon}}[\|\boldsymbol{f}(\boldsymbol{\Theta}; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t) - \boldsymbol{x}_0\|^2]. \tag{5}$$

The minimum of (5) over all possible denoisers $\boldsymbol{\Theta}$ is achieved by the posterior mean $\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]$, yielding the **Bayes denoising risk** $R_{\text{Bayes}} := \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}_t}[\|\boldsymbol{x}_0 - \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t]\|_F^2/(2dT)]$. We therefore regard a trained model whose population risk approaches R_{Bayes} as achieving **generalization**.

We characterize memorization as the ability of the model to reconstruct a noisy training sample back to the same clean sample. Mathematically, we define it as follows.

Definition 1 ((β,γ)-sample-specific-memorization)**.** *For any $\boldsymbol{x}_0^i \in \mathcal{R}$ define the function g as*

$$g(\boldsymbol{\Theta}; \beta, \boldsymbol{x}_0^i, t) := \Pr_{\boldsymbol{\epsilon}}(\|f(\boldsymbol{\Theta}; \boldsymbol{x}_t^i, t) - \boldsymbol{x}_0^i\|^2/(2d) \le \beta). \tag{6}$$

*We say that $\boldsymbol{x}_0^i$ is (β,γ)-**sample-specifically memorized** at time step t if it holds true that: $g(\boldsymbol{\Theta}; \beta, \boldsymbol{x}_0^i, t) \ge 1-\gamma$. To measure the ratio of training samples that are memorized, we define the **memorization rate** at time step t as*

$$\mathcal{M}_N(\boldsymbol{\Theta}; \beta, t) := \Pr_{i \in \text{Unif}([N]), \mathcal{R}}(g(\boldsymbol{\Theta}; \beta, \boldsymbol{x}_0^i, t) \ge 1-\gamma), \tag{7}$$

where we interpret $\boldsymbol{\Theta} = \boldsymbol{\Theta}(\mathcal{R})$ for the model $\boldsymbol{\Theta}$ trained by $\mathcal{R}$.

Remark 2. *In Definition 1, we quantify memorization by whether a model can reconstruct individual training samples from their noisy perturbations. A large $\mathcal{M}_N(\boldsymbol{\Theta}; \beta, t)$ indicates strong sample-specific memorization. As shown in Figure 2 (A) left, our memorization rate and the generation-based memorization ratio of (Gu et al., 2025), which measures whether generated samples are close to training data, exhibit a similar decreasing trend as the sample size increases.*

3 Main Theoretical Results

Section 3.1 presents the finite-sample convergence and generalization analysis of DDPM training with Transformers. Section 3.2 characterizes the memorization rate of the trained model and the sample complexity threshold that separates memorization and generalization. Section 3.3 reveals the memorization mechanism demonstrated by the learned denoiser.

We first introduce quantities regarding the convergence of the finite-sample DDPM training. Denote $N_m = |\{\boldsymbol{x}_0^i \in \mathcal{R} : Y_i = m\}|$ as the number of training samples from mode $m \in [M]$. Given the *empirical within-mode variance* $\hat{\rho}_N^2 = \frac{1}{dN}\sum_{m=1}^{M}\sum_{n\in[N],Y_n=m} \|\boldsymbol{x}_0^n - \bar{\boldsymbol{\mu}}_m\|^2$, where $\bar{\boldsymbol{\mu}}_m = \frac{1}{N_m}\sum_{n\in[N],Y_n=m} \boldsymbol{x}_0^n$ is the empirical mean of each mode $m \in [M]$, we define two risks based on estimators of the clean sample with the data mean known following (Li et al., 2026).

Definition 2. *For a noisy data $\boldsymbol{x}_t$ obtained from $\boldsymbol{x}_t$ with the latent variable Y defined by (1), the **oracle denoising risk (ODR)** is computed as:*

$$R_{\mathit{Oracle}} := \frac{1}{T}\sum_{t=1}^{T} R_{\mathit{Oracle}}^t, \text{ where } R_{\mathit{Oracle}}^t := \frac{1}{2}\mathbb{E}[\|\boldsymbol{x}_0 - \mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t, Y]\|^2] = \frac{d}{2}\frac{\rho^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha}_t}. \tag{8}$$

*By substituting ρ in (8) with $\hat{\rho}_N$, we can define the **empirical oracle denoising risk (EODR)***

$$\hat{R}_{\mathit{Oracle},N} := \frac{1}{T}\sum_{t=1}^{T} \hat{R}_{\mathit{Oracle},N}^t, \text{ where } \hat{R}_{\mathit{Oracle},N}^t := \frac{d}{2}\frac{\hat{\rho}_N^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t\hat{\rho}_N^2 + 1 - \bar{\alpha}_t}. \tag{9}$$

R_{Oracle}^t and $\hat{R}_{\text{Oracle},N}^t$ in (8) and (9) are time-wise oracle and empirical oracle denoising risks at time step t, respectively. $\hat{R}_{\text{Oracle},N}^t$ can be interpreted as the oracle risk induced by the empirical Gaussian statistics with $\bar{\boldsymbol{\mu}}_m$ and $\hat{\rho}_t^{N2}$ as the mean and variance of each mode at t. Since the prior knowledge of Y provides more information than only conditioning on $\boldsymbol{x}_t$, we have $R_{\text{Oracle}} \le R_{\text{Bayes}}$. We then show that under certain conditions, DDPM loss with the model (2) can be trained to attain $\hat{R}_{\text{Oracle},N} + O(\epsilon)$ for any arbitrarily small $\epsilon > 0$.

3.1 FINITE-SAMPLE DDPM TRAINING AND GENERALIZATION

Before presenting our main theory below, we first characterize the requirements of the model initialization in the following condition.

Assumption 1. *The initialization of learnable prototypes* $\{\boldsymbol{z}_1^{(0)}, \cdots, \boldsymbol{z}_M^{(0)}\}$ *satisfy the condition that there exists a constant* $r_0 \in [0, \sqrt{2}/2)$ *s.t.,*

$$\|\boldsymbol{z}_m^{(0)} - \bar{\boldsymbol{\mu}}_m\| \leq r_0\sqrt{d}, \quad \forall m \in [M]. \tag{10}$$

Remark 3. *Assumption 1 ensures that each prototype starts in the basin of its empirical mode, so that correct-mode attention dominates during optimization. Similar local-initialization conditions are common in analyses of nonconvex neural networks, often paired with tensor or spectral initializers (Zhong et al., 2017; Zhang et al., 2019; Fu et al., 2020). Under our separated mixture model, this condition can also be attained through clustering and centroiding. Indeed, in our synthetic experiments, K-means initialization produces a training curve almost identical to the Assumption 1 initialization with* $r_0 = 0.2$*, indicating that the assumed basin is reachable by a practical initializer.*

The next theorem states the convergence analysis of the finite-sample DDPM training.

Theorem 1. *Under the Assumption 1, for any* $\epsilon > 0$*. If (i) the dimension* $d \gtrsim \max\{N/M^2, M\epsilon^{-2}\} \cdot \log N/\delta$*, (ii) the number of samples per-mode* $N_m \geq 1$*, (iii) the step size* $\eta_\lambda < O(d^{-1}T\bar{\alpha}_T)$*,* $\eta_z \leq O(M)$*, and (iv) the number of iterations* $S \gtrsim \log(d/\epsilon)$*, then with probability of* $1-\delta$*, we have*

$$L_{N,T}(\boldsymbol{\Theta}^{(S)}) \leq \hat{R}_{Oracle,N} + O(d\epsilon), \text{ and} \tag{11}$$

$$\bar{L}(\boldsymbol{\Theta}^{(S)}) - L_{N,T}(\boldsymbol{\Theta}^{(S)}) = \Theta(\rho^2(dM + \log(MT/\delta))/N). \tag{12}$$

Theorem 1 shows that, if the numbers of training samples and iterations are large enough, and each learnable prototype starts sufficiently close to its corresponding empirical mean, then the one-layer single-head Transformer trained by gradient descent achieves a training loss within $O(\epsilon)$ of the EODR $\hat{R}_{\mathrm{Oracle},N}$. Meanwhile, its population risk exhibits an explicit finite-sample generalization gap of order $O(dM\rho^2/N)$. This result illustrates that gradient descent can efficiently learn the empirical mean and variance of the finite training dataset, which determines the generalization gap. We then elaborate on these conditions.

Data dimension. The first lower bound $\epsilon^{-2}\log N/\delta$ controls the concentration error of λ_t within $O(\epsilon)$ of its convergent point, such that the trained model approaches $\hat{R}_{\mathrm{Oracle},N}$ up to the prescribed accuracy. The second lower bound $N/M^2 \log N/\delta$ ensures that the finite-sample shift of the empirical within-mode variance, which is of order $O(M/N)$, dominates its random fluctuation. This condition also leads to $\hat{R}_{\mathrm{Oracle},N} < R_{\mathrm{Oracle}}$, thereby giving the finite-sample picture $L_{N,T}(\boldsymbol{\Theta}^{(S)}) \approx \hat{R}_{\mathrm{Oracle},N} < R_{\mathrm{Oracle}} < \bar{L}(\boldsymbol{\Theta}^{(S)})$.

Number of training samples. Condition (ii) shows that when the number of training samples is no smaller than the number of modes, i.e., under the minimal sample complexity requirement, the empirical risk of the learned model can already approach the EODR. This is because Assumption 1 makes learning the prototypes relatively easy, thereby simplifying the finite-sample optimization problem. However, the generalization gap in (12) decreases as N increases, which means that strong generalization performance requires a large number of training samples.

Number of iterations. The required number of training iterations in condition (iv) scales only logarithmically with d/ϵ. This indicates that, once the model is properly initialized, learning the empirical denoiser with finite samples does not require a number of iterations that grows polynomially with the ambient dimension.

The effect of initialization has been discussed in Remark 3. The different step-size scales for λ_t and $\boldsymbol{z}_m$ in Condition (iii) reflect the different geometries of the parameters. The gradient of the scalar parameters $\lambda_t, t \in [T]$ scale with the ambient dimension, requiring η_λ to decrease as d^{-1}, whereas training $\boldsymbol{z}_m, m \in [M]$ permits a dimension-independent step size.

Corollary 1. *Suppose all the conditions in Theorem 1 hold, and*

$$N \gtrsim \epsilon^{-1}(M\rho^2 + \log(MT/\delta)), \tag{13}$$

then with probability of $1-\delta$*, we have* $|\hat{\rho}_N^2 - \rho^2| \leq \epsilon$*,* $R_{Bayes} - R_{Oracle} \leq O(d\epsilon)$*, and*

$$L_{N,T}(\boldsymbol{\Theta}^{(S)}) = R_{Bayes} + O(d\epsilon), \quad \bar{L}(\boldsymbol{\Theta}^{(S)}) = R_{Bayes} + O(d\epsilon). \tag{14}$$

Corollary 1 characterizes the required number of samples such that both the empirical and population risk approach within $O(d\epsilon)$ of the Bayes risk. This means that, with sufficient training samples, the trained model achieves near-Bayes-optimal denoising performance, which corresponds to generalization. Meanwhile, the empirical within-mode variance $\hat{\rho}_N^{t\,2}$ concentrates around the population variance ρ^2, and the per-dimensional gap between the ODR and the Bayes risk is only $O(\epsilon)$. The required sample size increases with the number of modes and the within-mode variance, reflecting the larger error in estimating the distribution parameters for learning the denoiser.

3.2 Sample-Specific Memorization and Sample Complexity

Next, we investigate the memorization performance of the learned Transformer model. We first denote $R_N^t = \frac{(1-\bar{\alpha}_t)\rho^2(1-M/N)}{2(\bar{\alpha}_t\rho^2(1-M/N)+1-\bar{\alpha}_t)}$, $B_N^t = \sqrt{2}(\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t\rho^2(1-M/N)+1-\bar{\alpha}_t})^2\rho^2(1-M/N)/2$, and $S_N^t = \frac{\rho^3(1-M/N)(1-\bar{\alpha}_t)^{3/2}\sqrt{\bar{\alpha}_t}\sqrt{\bar{\alpha}_t\rho^2(1-M/N)+2(1-\bar{\alpha}_t)(1+O(\min\{\epsilon/M,M^2/N\}))}}{\sqrt{2}(\bar{\alpha}_t\rho^2(1-M/N)+1-\bar{\alpha}_t)^2} + \text{poly}(d^{-1})$. R_N^t is close to the time-wise EODR at time step t since $\rho_t^{N\,2} \approx (1-M/N)\rho^2$ in the high-dimensional regime. Based on the above quantities and the formulation in Definition 1, we can characterize the memorization rate after the model training.

Theorem 2. *Given $\mathbf{\Theta}^{(S)}$ trained with $\mathcal{R}$ by Theorem 1 and $N > M$, for any $t \in [T]$ w.h.p.,*

$$\mathcal{M}_N(\mathbf{\Theta}^{(S)};\beta,t) = \Psi\Big((\sqrt{d}(\beta - R_N^t) + \Psi^{-1}(\gamma + O(d^{-1/2}))S_N^t)/B_N^t\Big) + O(d^{-1/2}). \tag{15}$$

When $N = M$, $\mathcal{M}_N(\mathbf{\Theta}^{(S)};\beta,t) = 1$ for any $\beta \geq$ poly(d^{-1}).

Theorem 2 reveals that, in the high-dimensional regime, the memorization rate of the Transformer trained on N finite samples under the conditions of Theorem 1 can be approximated by the cumulative distribution function (CDF) of the standard Gaussian distribution as a function of the per-dimensional reconstruction tolerance β.

Critical reconstruction threshold. Theorem 2 identifies the *critical reconstruction threshold* $\beta_{N,t}^{\text{crit}} := R_N^t - d^{-\frac{1}{2}}S_N^t\Psi^{-1}(\gamma + O(d^{-\frac{1}{2}}))$, around which the memorization rate begins to change substantially. For fixed γ, the memorization rate changes from close to 0 to close to 1 as β crosses this threshold, over a window of $O(d^{-1/2})$. In the high-dimensional regime, $\beta_{N,t}^{\text{crit}}$ reflects concentration of the reconstruction errors around the time-wise EODR R_N^t.

Sample complexity under the memorization metric. The above result raises a natural question: which choice of β provides a meaningful criterion for sample-specific memorization under finite training data? We choose to use the time-wise ODR R_{Oracle}^t defined in (8) as a fixed threshold. It represents the denoising error attainable from the underlying population distribution at time step t. Hence, reconstruction of training samples below this threshold indicates sample-specific fitting on finite samples beyond population-level denoising. In contrast, a larger threshold may also count models with substantial denoising error, which weakens its interpretation as memorization. Denote $S_\infty^t = \lim_{N\to\infty} S_N^t$ and $B_\infty^t = \lim_{N\to\infty} B_N^t$. We then have the following corollary.

Corollary 2. *Given $\mathbf{\Theta}^{(S)}$ trained with $\mathcal{R}$ by Theorem 1, we have w.h.p., for any $t \in [T]$,*

$$\mathcal{M}_N(\mathbf{\Theta}^{(S)};R_{\text{Oracle}}^t,t)\begin{cases}\to 1, & \text{if } N/(M\rho^2\sqrt{d}) \to 0,\\ \to \Psi(\frac{\Psi^{-1}(\gamma+O(d^{-1/2}))S_\infty^t}{B_\infty^t}) + O(d^{-1/2}), & \text{if } N/(M\rho^2\sqrt{d}) \to \infty.\end{cases} \tag{16}$$

This result identifies $N = \Theta(M\sqrt{d})$ as the sample complexity scale that separates sample-specific memorization from generalization. When $N \ll M\sqrt{d}$, the finite-sample discrepancy $R_{\text{Oracle}}^t - R_N^t = \Theta(M\rho^2/N)$ dominates the $O(d^{-1/2})$-width window of the memorization rate, such that almost all training samples can be reconstructed within the ODR. When $N \gg M\sqrt{d}$, this finite-sample discrepancy becomes smaller than the transition width, and the memorization rate approaches is a small constant. This threshold is consistent with the $O(M\rho^2/\epsilon)$ risk sample complexity in Corollary 1 if $\epsilon = \Theta(d^{-1/2})$, i.e., the gap between the empirical and population risk is the same order of the window width.

3.3 Memorization Mechanism: Empirical-Prototype Attraction

The results in Section 3.2 characterize the sample-specific memorization exhibited by the trained model, but do not yet explain the mechanism through which the trained denoiser exhibits such behavior. A key question is what finite-sample statistics are encoded in the learned parameters and

how they determine the denoising prediction. In this section, we first quantify the convergence of model parameters to the statistics of the training dataset and then show that the trained Transformer performs *empirical-prototype attraction (EPA) mechanism*, where its output approximately interpolates between the scaled noisy input and a learned empirical prototype, given by the empirical mean of the corresponding mode.

Learning finite-sample statistics. For each timestep t, denote $\lambda_t^N := \frac{1}{\bar{\alpha}_t(\hat{\rho}_N)^2+1-\bar{\alpha}_t}$ as the denoising coefficient determined by the empirical within-mode variance. We then have the following result for the convergence of model parameters to finite-sample statistics.

Proposition 1. *Suppose the conditions in Theorem 1 hold, then w.h.p., the trained model $\boldsymbol{\Theta}^{(S)}$ after S iterations satisfies that for any $t \in [T]$, $m \in [M]$,*

$$|\lambda_t{}^{(S)} - \lambda_t^N|,\ \|\boldsymbol{z}_m^{(S)} - \bar{\boldsymbol{\mu}}_m\| \le O(\epsilon). \tag{17}$$

Proposition 1 shows that gradient descent encodes two finite-sample statistics into the trained Transformer. The prototype parameter $\boldsymbol{z}_m$ learns the empirical mean $\bar{\boldsymbol{\mu}}_m$ of each mode, whereas λ_t learns a timestep-dependent denoising coefficient determined by the empirical within-mode variance. Hence, the parameters of the trained denoiser preserve explicit information about the particular finite training dataset, rather than its underlying population distribution.

Empirical-prototype attraction mechanism. We introduce the memorization mechanism of the trained Transformer in the following proposition.

Proposition 2. *Suppose the conditions in Theorem 1 hold. For $\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_m, \rho^2\boldsymbol{I})$ and its noisy observation $\boldsymbol{x}_t$, the trained denoiser $\boldsymbol{\Theta}^{(S)}$ satisfies*

$$\|\boldsymbol{f}(\boldsymbol{\Theta}^{(S)}; \boldsymbol{x}_t, t) - ((1 - v_t^N)\boldsymbol{x}_t/\sqrt{\bar{\alpha}_t} + v_t^N\,\bar{\boldsymbol{\mu}}_m)\|^2 \le O(d\epsilon)\,, \tag{18}$$

where $\bar{\boldsymbol{\mu}}_m$ is the empirical mean of mode m, which we refer to as its empirical prototype*, and $v_t^N = (1-\bar{\alpha}_t)/(\bar{\alpha}_t\hat{\rho}_N^{t\,2} + 1 - \bar{\alpha}_t) \in (0, 1]$. In particular:*

- *If $N_m = 1$, then $\bar{\boldsymbol{\mu}}_m = \boldsymbol{x}_{i_m}^0$, where $\boldsymbol{x}_{i_m}^0$ is the unique training sample from mode m.*
- *If $N_m = 1\ \forall m \in [M]$, then $v_t^N = 1$, s.t., $\|\boldsymbol{f}(\boldsymbol{\Theta}^{(S)}; \boldsymbol{x}_t, t) - \boldsymbol{x}_{i_m}^0\|^2 \le O(d\epsilon)$.*

Proposition 2 provides a geometric interpretation of the memorization mechanism learned from finite data. At each timestep, the denoiser interpolates between the scaled noisy observation $\boldsymbol{x}_t/\sqrt{\bar{\alpha}_t}$ and the empirical prototype $\bar{\boldsymbol{\mu}}_m$ of its corresponding mode. When $N_m = 1$, the empirical prototype coincides with the unique training sample in mode m. EPA therefore reduces to *training-sample attraction*. In the extreme case $N = M$ where $N_m = 1$ for any $m \in [M]$, the empirical within-mode variance $\hat{\rho}_N = 0$ and $v_t^N = 1$, making the generated sample approximately coincide with the corresponding training example with the same mode. This indicates a direct mechanism for sample-specific memorization, i.e., gradient-based training stores an individual sample as the prototype toward which its noisy perturbations are denoised, which is consistent with the memorization studied in (Bonnaire et al., 2025; Zhang et al., 2026).

When $N_m > 1$, the attracting point becomes the empirical mode mean $\bar{\boldsymbol{\mu}}_m$, rather than any individual training example. Moreover, under the Gaussian mixture model, we have $\|\bar{\boldsymbol{\mu}}_m - \boldsymbol{\mu}_m\| \le O(\rho\sqrt{d/N_m})$ with a high probability by concentration inequality. Hence, as the number of samples per mode increases, the empirical prototype progressively approaches the population mean $\boldsymbol{\mu}_m$. The diffusion model output therefore continuously transitions from training-sample attraction in the data-scarce regime, through EPA at a finite number of samples, to population-mean denoising in the data-rich regime.

3.4 Proof Idea, Technical Novelty, and Limitations

Proof idea of Theorems 1 and 2. To prove Theorem 1, we first use Assumption 1 to derive conditions under which, at initialization, the attention weights concentrate with high probability on the prototype corresponding to the same mode as the input. This allows us to approximate the softmax attention in the subsequent analysis. We then compute the gradients of the empirical risk with respect to λ_t and z_m, identify the dominant signal components in these gradients, and upper bound the noise terms using concentration inequalities. Because the gradient noise behaves differently depending on whether every mode contains more than one sample or exactly one sample, we analyze

these two cases separately in Lemmas 2 and 3. Based on the signal components of the gradients, we prove that λ_t and z_m converge approximately linearly to their convergent points, which enables us to characterize the loss and generalization gap of the Transformer trained with finite data.

In the proof of Theorem 2, we first use the Berry–Esseen theorem to approximate the distribution of the loss for an individual noisy input $\boldsymbol{x}_t^i$ by the standard Gaussian-CDF, thereby characterizing $g(\boldsymbol{\Theta}; \beta, \boldsymbol{x}_t^i, t)$ at time step t. We then apply the Berry–Esseen theorem again with $\boldsymbol{x}_t^i$ as the random variable to obtain the final characterization of the memorization rate.

Technical Novelty. Our analysis provides a convergence analysis of gradient-based training for nonlinear Transformers in the finite-sample regime, extending existing convergence analyses based on population risk (Li et al., 2026) or approximately linearized models (Wang et al., 2024; Han et al., 2024). For memorization and generalization analysis, unlike prior work (Gu et al., 2025; Bonnaire et al., 2025; Zhang et al., 2026) that either measures memorization empirically or only characterizes the generalization gap of the optimal denoiser, we mathematically derive the memorization rate of the converged model and characterize the corresponding memorization mechanism (Proposition 2).

Limitations and potential extensions. Although we consider a more complex model and optimization problem, our analysis still relies on several simplifying assumptions, including a one-layer, single-head Transformer and a Gaussian mixture model whose number of learnable prototypes matches the number of underlying modes. Future work could extend the theory to multi-head Transformers for data with multiple tokens and hierarchical structures. Another direction is to develop convergence analyses for settings where the numbers of prototypes and modes differ.

4 Numerical Experiments

In this section, we conduct synthetic experiments in Section 4.1 and real-data experiments in Section 4.2 to verify our findings, respectively.

4.1 Synthetic Experiments

Setup. We generate synthetic data by (1). $d = 128$, $M = 10$, $\rho = 0.5$. $T = 50$ is the total number of time steps. The learning model is a one-layer single-head Transformer as formulated in (2). We adopt a linear schedule, i.e., $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, where $\alpha_t = \alpha_1 - (\alpha_1 - \alpha_T) \cdot (t-1)/(T-1)$. We use $\alpha_1 = 0.98$, $\alpha_T = 0.95$. For the computation of the memorization rate, we choose $\gamma = 0.05$. We consider two initialization schemes: an Assumption-1-based initialization that perturbs each ground-truth empirical mode mean with $r_0 = 0.2$, and a data-driven initialization that sets the prototypes to the K-means centroids of the training samples.

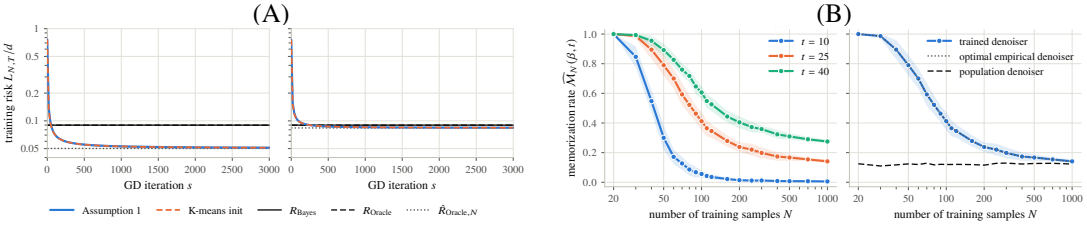


Figure 1: (A) The convergence of the Transformer model (2) trained by gradient descent with finite sample sizes. Blue line: Trained from initialization by Assumption 1. Red dashed line: Trained from initialization by K-means. **Left**: $N = 20$. **Right**: 120. (B) The empirical memorization rate with varying sample sizes. **Left**: $t = 10, 25, 40$. **Right**: The results of the trained model, the optimal empirical denoiser (parameterized by λ_t^N and $\bar{\boldsymbol{\mu}}_m$), and the population denoiser ($N = 1000$) when $t = 25$.

Convergence of finite-sample training. Figure 1 (A) illustrates the convergence of the Transformer model (2) trained by gradient descent with finite sample sizes $N = 20, 120$. We specifically plot the loss curves under both K-means initialization and the initialization specified in Assumption 1. The two curves nearly overlap, which means that Assumption 1 is practical, i.e., the initialization of $\boldsymbol{z}_m, m \in [M]$ can be achieved using standard methods such as K-means without affecting our subsequent theoretical conclusions. As the number of training iterations increases, the DDPM loss gradually converges to $\hat{R}_{\text{Oracle},N}$, which justifies (11) in Theorem 1. Moreover, the Bayes and the oracle denoising risks are very close to each other, as Corollary 1 indicates. Both are larger than $\hat{R}_{\text{Oracle},N}$, with a relatively large gap when $N = 20$ and a much smaller gap when $N = 120$.

Memorization rate. In Figure 1 (B), we showcase the memorization rate of the trained Transformer against the number of training samples N. The left figure shows that, as N increases, the empirical memorization rate continuously decreases from 1 across different time steps. For the same N, the converged memorization rate increases with t, consistent with the theoretical result in Corollary 2 that S_∞^t/B_∞^t increases with t under the linear schedule. The right figure further illustrates that the converged memorization rate approaches that of the population denoiser, which learns the true distribution parameters. This is consistent with the findings in Corollary 2. In addition, the memorization rate curve of the trained model overlaps with that of the model parameterized by the convergent point in (17), validating the characterization of the convergence point in Proposition 1.

4.2 Real-Data Experiments

Setup. We further evaluate whether the results predicted by our theory arise in a standard diffusion model on CIFAR-10 (Krizhevsky et al., 2010). To formulate the mixture of Gaussians with empirical modes, we extract DINOv2 (Oquab et al., 2024) ViT-S/14 features and perform global K-means clustering in the representation space. We choose $K = 10$. For each cluster $m \in [K]$, we split the data into training and held-out sets, where the population mean $\boldsymbol{\mu}_m$ is computed using all data, while the empirical mean $\hat{\boldsymbol{\mu}}_m$ is computed using only the training data. The learning model is a 12-layer DiT (Peebles & Xie, 2023). We use $T = 1K$ diffusion steps with a linear noise schedule.

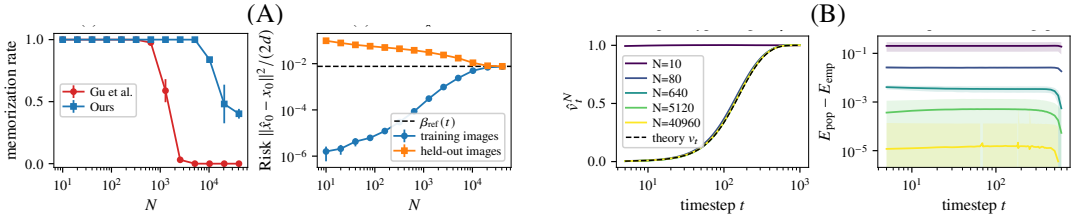


Figure 2: (A) $t = 150$. **Left**: The memorization rates defined in (Gu et al., 2025) and our Definition 1. The two curves exhibit a similar decreasing trend as N increases. **Right**: β_{ref} is the Bayes risk approximated by the held-out reconstruction risk of a large-data reference model trained on all $48K$ non-held-out images. Blue line: Empirical risk on training images. Orange line: Population risk approximated by reconstructed error on held-out images. (B) Results of interpolation fitting to the trained denoiser outputs. **Left**: Coefficient $\hat{v}_t^N$. **Right**: $E_{\text{emp}} := \sum_{m=1}^K \|\hat{\boldsymbol{z}}_m - \bar{\boldsymbol{\mu}}_m\|^2/(dK)$ and $E_{\text{pop}} := \sum_{m=1}^K \|\hat{\boldsymbol{z}}_m - \boldsymbol{\mu}_m\|^2/(dK)$ represent the fitting errors of the learned prototypes versus the empirical means and versus the population means, respectively. $E_{\text{pop}} - E_{\text{emp}}$ shows if the $\hat{\boldsymbol{z}}_m$ is closer to $\bar{\boldsymbol{\mu}}_m$ or $\boldsymbol{\mu}_m$.

Generalization gap. Figure 2 (A) right shows that as the number of training samples increases, the empirical risk increases, while the population risk decreases. Both gradually converge to β_{ref} with a decreasing gap. This is consistent with the conclusion in Corollary 1 for the generalization gap.

The mechanism of memorization. We then examine whether a DiT model exhibits the interpolation structure predicted by Proposition 2. Given noisy input $\boldsymbol{x}_t$ at time step t, we fit the prediction with $\hat{\boldsymbol{x}}_0 \approx (1 - \hat{v}_t^N)\boldsymbol{x}_t/\sqrt{\bar{\alpha}_t} + \hat{v}_t^N \hat{\boldsymbol{z}}_m$. Figure 2 (B) left shows that the interpolation coefficient $\hat{v}_t^N$ closely follows the theoretical form predicted by our analysis. When $N_m = 1, \forall m \in [M]$, we observe $\hat{v}_t^N = 1$ across time steps, as indicated in Proposition 2. When N_m increases, the curve $\hat{v}_t^N$ approaches that of v_t. Figure 2 (B) right reveals that $\hat{\boldsymbol{z}}_m$ is closer to $\bar{\boldsymbol{\mu}}_m$ than $\boldsymbol{\mu}_m$ when t is not too large, and that $\hat{\boldsymbol{z}}_m$ approaches $\boldsymbol{\mu}_m$ as the sample size increases.

5 Conclusion

This paper provides a finite-sample theoretical analysis of a one-layer single-head Transformer in diffusion model training with gradient descent. This work characterizes an explicit sample-specific memorization rate of the trained model with a sample complexity threshold that separates the memorization and generalization regimes. Our analysis further theoretically proves an empirical-prototype attraction mechanism for memorization. Further directions include extending the analysis to more realistic data and model settings.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models or conceptual frameworks, to design or provide feedback on research methodology or experiments, and to implement methods. Specifically, we used large language models (LLMs) as a discussion partner when refining the problem formulation and the conceptual framing of our analysis, to obtain feedback on the design of the synthetic and real-data experiments in Sections 4.1 and 4.2, and to assist in implementing the code for these experiments. We have not used generative AI tools to generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, or interpret results. Translation, cleaning and reformatting of datasets, and qualitative and thematic data analysis are not applicable to this work.

We have reviewed all AI-assisted work. Conceptual suggestions from LLMs were adopted only after the authors independently checked their correctness and confirmed their novelty through a manual literature survey. LLM feedback on experimental design was treated as suggestions, and all final design decisions were made by the authors. LLM-assisted code was reviewed, tested, and verified for correctness by the authors, and all reported results were produced with this verified code. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work is primarily theoretical and does not involve human subjects or the collection of sensitive or private data. The empirical experiments use publicly available benchmark datasets. We are not aware of any specific ethical concerns from the methods or experiments presented in this work.

REPRODUCIBILITY STATEMENT

We present the complete assumptions and theoretical framework in the main text and provide proofs for all theoretical results in the appendix. The experimental section and the appendix specify the data generation process, model configuration, and training setup employed in the experiments.

REFERENCES

Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tyEyYT267x.

Iskander Azangulov, George Deligiannidis, and Judith Rousseau. Convergence of diffusion models under the manifold hypothesis in high-dimensions. *arXiv preprint arXiv:2409.18804*, 2024.

Tony Bonnaire, Raphaël Urfin, Giulio Biroli, and Marc Mézard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. *arXiv preprint arXiv:2505.17638*, 2025.

Sam Buchanan, Druv Pai, Yi Ma, and Valentin De Bortoli. On the edge of memorization in diffusion models. *Advances in Neural Information Processing Systems*, 38:96113–96157, 2025.

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

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. ISSN 01621459. URL http://www.jstor.org/stable/23239562.

Haoyu Fu, Yuejie Chi, and Yingbin Liang. Guaranteed recovery of one-hidden-layer neural networks via cross entropy. *IEEE transactions on signal processing*, 68:3225–3235, 2020.

Anand Jerry George, Rodrigo Veiga, and Nicolas Macris. Denoising score matching with random features: Insights on diffusion models from precise learning curves. *Journal of Statistical Mechanics: Theory and Experiment*, 2026(8):084009, 2026.

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

Andi Han, Wei Huang, Yuan Cao, and Difan Zou. On the feature learning in diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yinbin Han, Meisam Razaviyayn, and Renyuan Xu. Neural network-based score estimation in diffusion models: Optimization and generalization. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Yu Huang, Zixin Wen, Yuejie Chi, and Yingbin Liang. A theoretical analysis of self-supervised learning for vision transformers. In *International Conference on Learning Representations*, volume 2025, pp. 1326–1388, 2025.

Zhihan Huang, Yuting Wei, and Yuxin Chen. Denoising diffusion probabilistic models are optimally adaptive to unknown low dimensionality. *Mathematics of Operations Research*, 2026.

Samy Jelassi, Michael Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 35:37822–37836, 2022.

Jiarui Jiang, Wei Huang, Miao Zhang, Taiji Suzuki, and Liqiang Nie. Unveil benign overfitting for transformer in vision: Training dynamics, convergence, and generalization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=FGJb0peY4R.

Zahra Kadkhodaie, Florentin Guth, Eero Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *International Conference on Learning Representations*, volume 2024, pp. 46543–46567, 2024.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 5(4):1, 2010.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of statistics*, pp. 1302–1338, 2000.

Bingrui Li, Wei Huang, Andi Han, Zhanpeng Zhou, Taiji Suzuki, Jun Zhu, and Jianfei Chen. On the optimization and generalization of two-layer transformers with sign gradient descent. In *International Conference on Learning Representations*, volume 2025, pp. 64406–64484, 2025a.

Gen Li and Yuling Yan. Adapting to unknown low-dimensional structures in score-based diffusion models. *Advances in Neural Information Processing Systems*, 37:126297–126331, 2024.

Hongkang Li, Meng Wang, Sijia Liu, and Pin-Yu Chen. A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=jClGv3Qjhb.

Hongkang Li, Hancheng Min, and Rene Vidal. Transformers learn the optimal DDPM denoiser for multi-token GMMs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=6VyoaDj3zB.

Puheng Li, Zhong Li, Huishuai Zhang, and Jiang Bian. On the generalization properties of diffusion models. *Advances in Neural Information Processing Systems*, 36:2097–2127, 2023b.

Tianyi Li, Luca Biferale, Fabio Bonaccorso, Martino Andrea Scarpolini, and Michele Buzzicotti. Synthetic lagrangian turbulence by generative diffusion models. *Nature Machine Intelligence*, 6 (4):393–403, 2024.

Xiao Li, Zekai Zhang, Xiang Li, Siyi Chen, Zhihui Zhu, Peng Wang, and Qing Qu. Understanding representation dynamics of diffusion models via low-dimensional modeling. *Advances in Neural Information Processing Systems*, 38:107365–107404, 2025b.

Jiadong Liang, Zhihan Huang, and Yuxin Chen. Low-dimensional adaptation of diffusion models: Convergence in total variation. *arXiv preprint arXiv:2501.12982*, 2025.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

Eshaan Nichani, Alex Damian, and Jason D Lee. How transformers learn causal structure with gradient descent. *arXiv preprint arXiv:2402.14735*, 2024.

Matthew Niedoba, Berend Zwartsenberg, Kevin Patrick Murphy, and Frank Wood. Towards a mechanistic explanation of diffusion model generalization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Hrp6jRIKdX.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Samet Oymak, Ankit Singh Rawat, Mahdi Soltanolkotabi, and Christos Thrampoulidis. On the role of attention in prompt-tuning. *arXiv preprint arXiv:2306.03435*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, 2025.

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*, pp. 10684–10695, 2022.

Brendan Ross, Hamidreza Kamkari, Tongzi Wu, Rasa Hosseinzadeh, Zhaoyan Liu, George Stein, Jesse Cresswell, and Gabriel Loaiza-Ganem. A geometric framework for understanding memorization in generative models. In *International Conference on Learning Representations*, volume 2025, pp. 34773–34805, 2025.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. *Proceedings of the National Academy of Sciences*, 122(1): e2408799121, 2025.

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

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6048–6058. IEEE, 2023.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Davoud Ataee Tarzanagh, Yingcong Li, Xuechen Zhang, and Samet Oymak. Max-margin token selection in attention mechanism. *CoRR*, 2023.

Yuandong Tian, Yiping Wang, Zhenyu Zhang, Beidi Chen, and Simon Shaolei Du. Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention. In *Conference on Parsimony and Learning (Recent Spotlight Track)*, 2023.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. Breaking the curse of dimensionality: Diffusion models efficiently learn low-dimensional distributions. *Journal of Machine Learning Research*, 27(179):1–50, 2026a.

Wenhao Wang, Yifan Sun, Zongxin Yang, Zhengdong Hu, Zhentao Tan, and Yi Yang. Replication in visual diffusion models: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026b.

Yuqing Wang, Ye He, and Molei Tao. Evaluating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 37:19307–19352, 2024.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Zeqi Ye, Qijie Zhu, Molei Tao, and Minshuo Chen. Provable separations between memorization and generalization in diffusion models. In *International Conference on Learning Representations*, volume 2026, pp. 143375–143425, 2026.

Chenshuang Zhang, Chaoning Zhang, Sheng Zheng, Mengchun Zhang, Maryam Qamar, Sung-Ho Bae, and In So Kweon. A survey on audio diffusion models: Text to speech synthesis and enhancement in generative ai. *arXiv preprint arXiv:2303.13336*, 2023.

Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *Forty-first international conference on machine learning*, 2024.

Xiao Zhang, Yaodong Yu, Lingxiao Wang, and Quanquan Gu. Learning one-hidden-layer relu networks via gradient descent. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1524–1534. PMLR, 2019.

Zekai Zhang, Xiao Li, Xiang Li, Lianghe Shi, Meng Wu, Molei Tao, and Qing Qu. Generalization of diffusion models arises with a balanced representation space. In *International Conference on Learning Representations*, volume 2026, pp. 101444–101475, 2026.

Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pp. 4140–4149. PMLR, 2017.

We organize the appendix as follows. In Section A, we provide some extra experiments. In Section B, we introduce important notations and lemmas used in the paper. In Section C, we show the proof of the main theorems of the paper. In Section D, we prove the supporting key lemmas for the theoretical results.

A Additional Real-Data Experiments

A.1 Experiment details

We use the CIFAR-10 training split with $50,000$ images and extract DINOv2 ViT-S/14 features for global K-means clustering with $K = 10$. For each cluster, we construct training subsets with n samples per cluster, such that the total number of training samples is $N = 10n$. We reserve 200 images per cluster as a held-out set. The largest sweep model uses $N = 40,960$ training images, while the reference model is trained on all $48,000$ non-held-out images. We train an unconditional DiT-S/4 with 12 blocks, width 384, and 6 attention heads under the DDPM noise-prediction objective with $T = 1000$ diffusion steps. We use AdamW with learning rate 10^{-4}, batch size 256, $200K$ training steps, and EMA decay 0.9999.

To compute the memorization rate in Definition 1, for each training image $\boldsymbol{x}_0^i$, we independently draw P Gaussian noises and compute

$$\hat{g}_i(\boldsymbol{\Theta}; \beta, t) := \frac{1}{P} \sum_{p=1}^{P} \mathbb{1} \left\{ \frac{\left\| f(\boldsymbol{\Theta}; \boldsymbol{x}_t^i, t) - \boldsymbol{x}_0^i \right\|^2}{2d} \leq \beta \right\}. \tag{19}$$

We can then compute the empirical sample-specific memorization rate

$$\hat{\mathcal{M}}_N(\boldsymbol{\Theta}; \beta, t) := \frac{1}{N} \sum_{i=1}^{N} \mathbb{1} \left\{ \hat{g}_i(\boldsymbol{\Theta}; \beta, t) \geq 1 - \gamma \right\}, \tag{20}$$

i.e., the fraction of training images that can be reconstructed within the threshold β for at least a $1 - \gamma$ fraction of their noisy perturbations. We choose $P = 100$, $\gamma = 0.05$, $t = 150$, and $\beta_{\mathrm{ref}} = 0.0076$ by default.For comparison, we also adopt the generation-based memorization ratio R_{Mem} of (Gu et al., 2025). For each generated image $\hat{\boldsymbol{x}}$, let $d_1(\hat{\boldsymbol{x}})$ and $d_2(\hat{\boldsymbol{x}})$ denote its pixel-space ℓ_2 distances to the nearest and second-nearest training samples, respectively. A generated image is considered memorized if $d_1(\hat{\boldsymbol{x}})/d_2(\hat{\boldsymbol{x}}) < 1/3$, and R_{Mem} is the fraction of generated images satisfying this criterion. We estimate R_{Mem} using $10,000$ generated samples for each model. Note that $\hat{\mathcal{M}}_N(\boldsymbol{\Theta}; \beta, t)$ and R_{Mem} capture related but distinct characteristics of memorization.

A.2 Experiment Results

We compare the learned prototype $\hat{z}_m$ with the empirical cluster mean $\hat{\mu}_m$ and the full-data cluster mean μ_m. Figure 3 reports the corresponding per-pixel RMSEs across diffusion time steps. For small and moderate N, the learned prototype is substantially closer to the empirical mean than to the population-mean proxy over a broad range of time steps. As N increases, the two distances gradually approach each other because the empirical cluster mean itself converges toward the population mean. This provides additional evidence that the finite-sample denoiser first learns an empirical prototype and gradually transitions toward population-level structure as the sample size increases.

Figure 4 justifies the sensitivity of $\hat{\mathcal{M}}_N$ to its parameters. (a) varies the confidence parameter $\gamma \in \{0.05, 0.1, 0.2\}$ at $t_\star = 150$. The resulting curves nearly overlap, showing that the memorization is not sensitive to the precise choice of the $1 - \gamma$ reconstruction-success requirement. (b) evaluates three time steps $t \in \{75, 150, 250\}$. All time steps exhibit the same qualitative transition from near-complete sample-specific reconstruction at small N toward a substantially lower memorization rate at large N. (c) varies the number P of independent noise perturbations used to estimate $g(\boldsymbol{\Theta}; \beta, \boldsymbol{x}_0^i, t)$. The memorization estimates are nearly unchanged for $P = 10$ to 100, indicating that the default choice $P = 100$ is sufficient for stable estimation. (d) reports the held-out reconstruction risk used to calibrate β_{ref}. As the training size increases, the held-out risk of the sweep models steadily approaches that of the large-data reference model. This shows that the threshold is determined by an independently trained data-rich reference model.

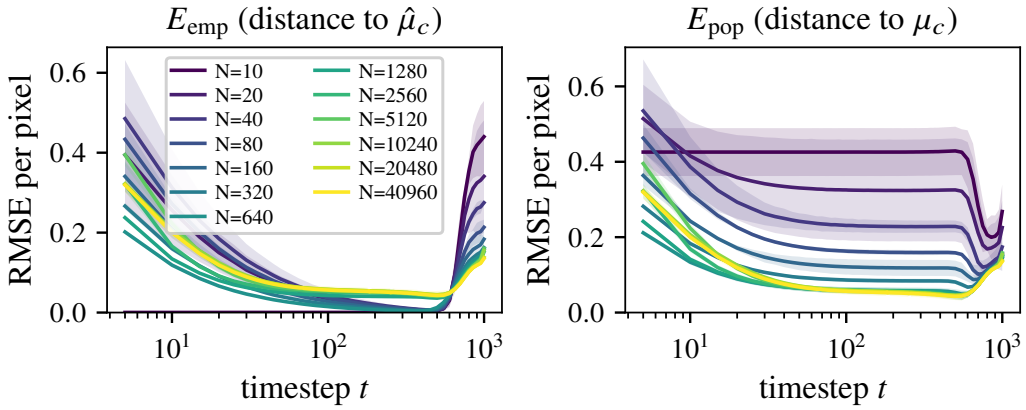


Figure 3: E_{emp} and E_{pop} versus t with different numbers of training samples.

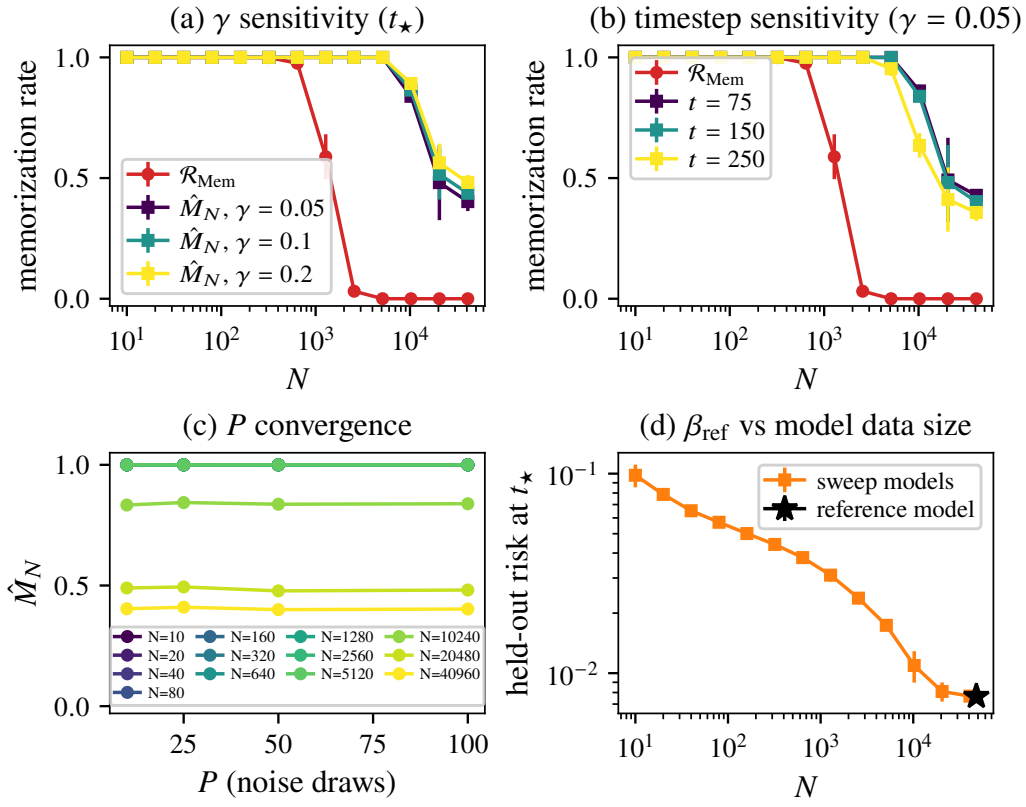


Figure 4: Sensitivity of $\hat{\mathcal{M}}_N$ to its parameters, including γ, P, and β_{ref}.

Figure 5 shows the full dependence of $\hat{\mathcal{M}}_N(\boldsymbol{\Theta}; \beta, t_\star)$ on the reconstruction threshold β. For training images, the empirical memorization rate exhibits sharp Gaussian-CDF-like transitions from nearly zero to nearly one, qualitatively consistent with the threshold behavior predicted by Theorem 2. Small-data models reconstruct their training examples with extremely small error, whereas large-data models require a threshold comparable to the held-out reconstruction risk. As N increases, the curves transition towards larger β. At the fixed threshold β_{ref}, this movement produces the decreasing memorization curve as in Figure 2 (A) left. The held-out control experiment exhibits the opposite result. For small N, held-out images require a much larger threshold before they are reconstructed successfully, reflecting a large generalization gap. As N grows, the curves for held-out images shift toward smaller β and progressively approach the curves for training images. Therefore, when N is large, sample-specific memorization vanishes, which is consistent with the disappearance of the reconstruction gap between training images and held-out images as shown in Figure 2 (A) right.

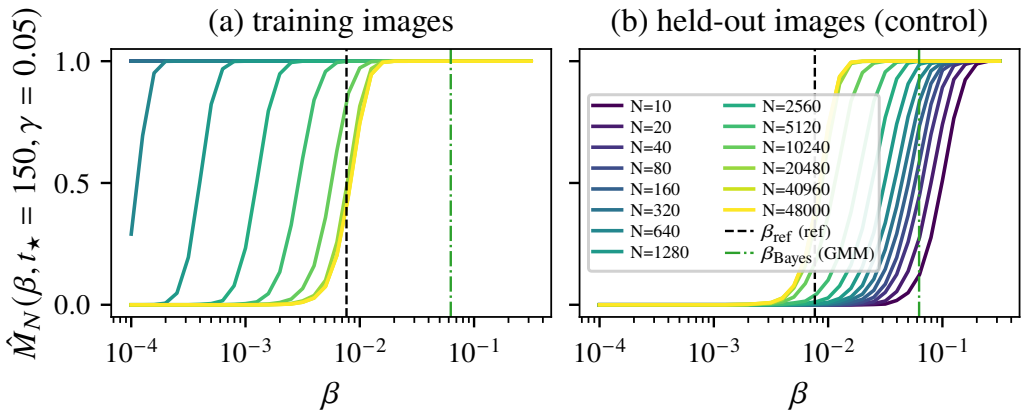


Figure 5: Memorization rates on training images and held-out images.

B PRELIMINARIES

We first present Table 2 for a summary of notations used in the proof.

Definition 3. *(Vershynin, 2010) We say X is a sub-Gaussian random variable with sub-Gaussian norm $K > 0$, if $(\mathbb{E}|X|^p)^{\frac{1}{p}} \leq K\sqrt{p}$ for all $p \geq 1$. In addition, the sub-Gaussian norm of X, denoted $\|X\|_{\psi_2}$, is defined as $\|X\|_{\psi_2} = \sup_{p \geq 1} p^{-\frac{1}{2}} (\mathbb{E}|X|^p)^{\frac{1}{p}}$.*

Lemma 1 (Vershynin (2010) Proposition 5.1, Hoeffding's inequality)**.** *Let $X_1, X_2, \cdots, X_N$ be independent centered sub-gaussian random variables, and let $K = \max_i \|\boldsymbol{X}_i\|_{\psi_2}$. Then for every $\boldsymbol{a} = (a_1, \cdots, a_N) \in \mathbb{R}^N$ and every $t \geq 0$, we have*

$$\Pr\left(\left|\sum_{i=1}^{N} a_i X_i\right| \geq t\right) \leq e \cdot \exp\left(-\frac{Mt^2}{K^2\|\boldsymbol{a}\|^2}\right), \tag{21}$$

where $c > 0$ is an absolute constant.

Lemma 2. *For any $\epsilon > 0$, as long as Assumption 1 holds, (i) the dimension $d \gtrsim \max\{\epsilon^{-2}M, N/M^2\} \cdot \log N/\delta$, (ii) the number of training samples $N \geq M$ and $|\{\boldsymbol{x}_0^i \in \mathcal{R} : Y_i = m\}| > 1$ for any $m \in [M]$, (iii) the number of iterations $S \gtrsim \log(d/\epsilon)$, and (iv) and the step size $\eta_\lambda < O(d^{-1}T\bar{\alpha}_T)$, $\eta_z \leq O(M)$ then with probability of $1 - \delta$, we have*

$$|\lambda_t{}^{(S)} - \hat{\lambda}_t^N|,\ |\lambda_t^N - \hat{\lambda}_t^N|,\ \|\boldsymbol{z}_m^{(S)} - \bar{\boldsymbol{\mu}}_m\| \leq O(\epsilon), \tag{22}$$

where

$$\hat{\lambda}_t^N = \frac{1 + \sqrt{\frac{\log \delta^{-1}}{d}}}{\bar{\alpha}_t(\hat{\rho}_N^2 - \sqrt{\frac{\log \delta^{-1}}{d}} - \frac{M\sqrt{\log(N/\delta)}}{\sqrt{dN}}) + 1 - \bar{\alpha}_t},\ \lambda_t^N = \frac{1}{\bar{\alpha}_t\hat{\rho}_N^2 + 1 - \bar{\alpha}_t}. \tag{23}$$

Table 2: Summary of Notations

Notations	Annotation
d, T	d is the dimension of each data. T is the total number of diffusion steps.
M, N	M is the total number of modes of the Gaussian mixture model. N is the number of training samples.
$\boldsymbol{\pi}$, $\{\boldsymbol{\mu}_i\}_{i=1}^M$, ρ	$\boldsymbol{\pi}$ is the fraction vector of all the Gaussian modes of the GMM. We choose $\pi_m = 1/M$, $m \in [M]$ in this work. $\{\boldsymbol{\mu}_i\}_{i=1}^M$ is the set of all the mean vectors. ρ^2 is the within-mode variance of the GMM.
$\boldsymbol{x}_0$, $\boldsymbol{x}_t$, Y	$\boldsymbol{x}_0$ denotes the clean data, while $\boldsymbol{x}_t$ is the noisy input of $\boldsymbol{x}_0$ at time step t by the diffusion forward process. Y is the mode of $\boldsymbol{x}_0$.
$\{\bar{\alpha}_t\}_{t=1}^T$, $\boldsymbol{\epsilon}$	$\{\bar{\alpha}\}_{t=1}^T$ is the noise schedule coefficients. $\boldsymbol{\epsilon}$ is the additive Gaussian noise.
$\boldsymbol{\Theta}$, $\{\boldsymbol{z}_m\}_{m=1}^M$, $\{\lambda_t\}_{t=1}^T$	$\boldsymbol{\Theta}$ is the set of parameters in the learning model, which includes $\{\boldsymbol{z}_m\}_{m=1}^M$ and $\{\lambda_t\}_{t=1}^T$. $\{\boldsymbol{z}_m\}_{m=1}^M$ are the prototype parameters for learning the mean of each mode, which correspond to the keys and values in the Transformer. $\{\lambda_t\}_{t=1}^T$ are the coefficient parameters for learning the variance of the GMM.
$\mathcal{M}_N(\boldsymbol{\Theta};\beta,t)$, $g(\boldsymbol{\Theta};\beta,\boldsymbol{x}_0,t)$	$\mathcal{M}_N(\boldsymbol{\Theta};\beta,t)$ is the memorization rate at time step t given the training set of N samples, error threshold β, and the input $\boldsymbol{x}_0$. $g(\boldsymbol{\Theta};\beta,\boldsymbol{x}_0,t)$ is the probability metric to define the memorization rate.
$\bar{\boldsymbol{\mu}}_m$, $\hat{\rho}_N^2$	$\bar{\boldsymbol{\mu}}_m$ is the empirical mean of the mode $m \in [M]$. $\hat{\rho}_N^2$ is the empirical within-mode variance.
$L_{N,T}(\boldsymbol{\Theta})$, $L(\boldsymbol{\Theta})$	$L_{N,T}(\boldsymbol{\Theta})$ is the empirical DDPM loss with finite samples. $L(\boldsymbol{\Theta})$ is the population DDPM loss.
R_{Bayes}, R_{oracle}, $\hat{R}_{\text{Oracle},N}$	R_{Bayes} is the optimal risk of minimizing the empirical DDPM loss. R_{oracle} is the oracle denoising risk, where the denoising model is chosen as the MMSE estimator with the mode of the input known. $\hat{R}_{\text{Oracle},N}$ is the empirical oracle denoising risk, which is the oracle denoising risk based on an empirical Gaussian statistics with $\bar{\mu}_m$ and $\hat{\rho}_t^{N\,2}$ as the mean and variance of each mode.
$\mathcal{O}()$, $\Omega()$, $\Theta()$	We follow the convention that $f(x) = O(g(x))$ (or $\Omega(g(x))$, $\Theta(g(x))$) means that $f(x)$ increases at most, at least, or in the order of $g(x)$, respectively. Specifically, if $f(x) = O(g(x))$, then there exists $C > 0$ and $a > 0$, such that $f(x) \leq C \cdot g(x)$ when $x > a$. If $f(x) = \Omega(g(x))$, then there exists $c > 0$ and $a > 0$, such that $f(x) \geq c \cdot g(x)$ when $x > a$. If $f(x) = \Theta(g(x))$, then there exists $C > c > 0$ and $a > 0$, such that $c \cdot g(x) \leq f(x) \leq C \cdot g(x)$ when $x > a$.
$\gtrsim$, $\lesssim$	$f(x) \gtrsim g(x)$ (or $f(x) \lesssim g(x)$) means that $f(x) \geq \Omega(g(x))$ (or $f(x) \lesssim \mathcal{O}(g(x))$).
poly()	If $f(x) = \text{poly}(x)$, then there exists $k > 0$ and a set of constants $\{c_i\}_{i=0}^k$, such that $f(x) = \sum_{i=0}^k c_i x^i$, which means $f(x)$ is a polynomial function of x with a finite maximal power.

Therefore,

$$L_{N,T}(\boldsymbol{\Theta}^{(\boldsymbol{S})}) \leq \hat{R}_{\mathit{Oracle},N} + O(d\epsilon), \tag{24}$$

$$\bar{L}(\boldsymbol{\Theta}^{(\boldsymbol{S})}) - L_{N,T}(\boldsymbol{\Theta}^{(\boldsymbol{S})}) = \Theta(\frac{dM\rho^2 + \rho^2 \log(MT/\delta)}{N}). \tag{25}$$

Lemma 3. *For any $\epsilon > 0$, as long as Assumption 1 holds, (i) the dimension $d \gtrsim \max\{\epsilon^{-2}M, N/M^2\} \cdot \log N/\delta$, (ii) the number of training samples $N \geq M$, and there exists*

$m \in [M]$ such that $|\{\boldsymbol{x}_0^i \in \mathcal{R} : Y_i = m\}| = 1$, (iii) the number of iterations $S \gtrsim \log(d/\epsilon)$, (iv) and the step size $\eta_\lambda < O(d^{-1}T\bar{\alpha}_T)$, $\eta_z \leq O(M)$, then with probability of $1-\delta$, we have

$$|\lambda_t{}^{(S)} - \hat{\lambda}_t^N|,\ |\lambda_N^t - \hat{\lambda}_t^N|,\ \|\boldsymbol{z}_m^{(S)} - \bar{\boldsymbol{\mu}}_m\| \leq O(\epsilon), \tag{26}$$

where

$$\hat{\lambda}_t^N = \frac{1+\sqrt{\frac{\log\delta^{-1}}{d}}}{\bar{\alpha}_t(\hat{\rho}_N^2 - \nu\sqrt{\frac{\log\delta^{-1}}{d}}) + 1 - \bar{\alpha}_t},\ \lambda_N^t = \frac{1}{\bar{\alpha}_t\hat{\rho}_N^2 + 1 - \bar{\alpha}_t}, \tag{27}$$

for some $\nu \in [0,1]$. Therefore,

$$L_{N,T}(\boldsymbol{\Theta}^{(\boldsymbol{S})}) \leq \hat{R}_{Oracle,N} + O(d\epsilon), \tag{28}$$

$$\bar{L}(\boldsymbol{\Theta}^{(\boldsymbol{S})}) - L_{N,T}(\boldsymbol{\Theta}^{(\boldsymbol{S})}) = \Theta(\frac{dM\rho^2 + \rho^2\log(MT/\delta)}{N}). \tag{29}$$

C Proof of Main Theorems

C.1 Proof of Theorem 1

Proof. Given Lemmas 2 and 3, if we combine their conditions (ii), we can derive that all the conclusions hold for $N \geq M$. Then, we can obtain Theorem 1. □

C.2 Proof of Corollary 1

Proof. By (103), we have

$$|\hat{\rho}_N^2 - \rho^2| \leq \epsilon. \tag{30}$$

From (12), we can derive that to make the generalization gap smaller than $O(\epsilon)$, we need

$$N \gtrsim \epsilon^{-1}(M\rho^2 + \log(MT/\delta)). \tag{31}$$

This condition makes (104) small than $O(\epsilon)$, such that both empirical risk and population risk converge to be close to R_{Oracle}. □

C.3 Proof of Theorem 2

We next study the performance of memorization and generalization after the DDPM training. Denote

$$Y_i = \|\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^i, t) - \boldsymbol{x}_0^i\|^2/2. \tag{32}$$

Recall that

$$\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^i, t) - \boldsymbol{x}_0^i = \bar{\boldsymbol{e}}_i^t - \lambda_t{}^{(S)}(1-\bar{\alpha}_t)(\boldsymbol{x}_0^i - \boldsymbol{z}_m) + \frac{1-\lambda_t{}^{(S)}(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t \tag{33}$$

Given $\boldsymbol{x}_t^i$, denote $m_Y(\boldsymbol{x}_t^i) = \mathbb{E}[Y_i|\boldsymbol{x}_t^i]$. Then, we have

$$\begin{aligned} &m_Y(\boldsymbol{x}_t^i) - (\lambda_t{}^{(S)2}(1-\bar{\alpha}_t)^2\|\boldsymbol{x}_t^i - \boldsymbol{z}_m\|^2 + \lambda_t{}^{(S)2}\hat{\rho}_N^4 d\bar{\alpha}_t(1-\bar{\alpha}_t))/2 \\ \leq &O(\text{poly}(d^{-1})). \end{aligned} \tag{34}$$

Denote $s_Y^2(\boldsymbol{x}_t^i) = \text{Var}(Y_i|\boldsymbol{x}_t^i)$. Then, we have

$$\begin{aligned} &s_Y^2(\boldsymbol{x}_t^i) - (4\lambda_t{}^{(S)4}\hat{\rho}_N^4\bar{\alpha}_t(1-\bar{\alpha}_t)^3\|\boldsymbol{x}_t^i - \boldsymbol{z}_m\|^2 + 2d\lambda_t{}^{(S)4}\hat{\rho}_N^8\bar{\alpha}_t^2(1-\bar{\alpha}_t)^2)/4 \\ \leq &O(\text{poly}(d^{-1})). \end{aligned} \tag{35}$$

Note that

$$\begin{aligned} &\mathbb{E}[|(\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^i, t)_j - \boldsymbol{x}_{i,j}^0)^2|^3] \\ \lesssim &\lambda_t{}^{(S)6}\hat{\rho}_N^{12}\bar{\alpha}_t^3(1-\bar{\alpha}_t)^3 + \lambda_t{}^{(S)6}(1-\bar{\alpha}_t)^6|\boldsymbol{x}_{i,j}^t - \boldsymbol{z}_{m,j}|^6 + \text{poly}(d^{-1}) \end{aligned} \tag{36}$$

Therefore, by Berry–Esseen Theorem, we have

$$\begin{aligned}
&\sup_{q\in\mathbb{R}}\Big|\Pr_{\boldsymbol{\epsilon}}\Big(\frac{Y-m_Y(\boldsymbol{x}_t^i)}{s_Y(\boldsymbol{x}_t^i)}\le q\Big)-\Psi(q)\Big|\\
\lesssim&\frac{d{\lambda_t}^{(S)6}\hat\rho_N^{12}\bar\alpha_t^3(1-\bar\alpha_t)^3+{\lambda_t}^{(S)6}(1-\bar\alpha_t)^6\sum_{j=1}^d|\boldsymbol{x}_{i,j}^t-\boldsymbol{z}_{m,j}|^6+\text{poly}(d^{-1})}{(4{\lambda_t}^{(S)4}\hat\rho_N^4\bar\alpha_t(1-\bar\alpha_t)^3\|\boldsymbol{x}_t^i-\boldsymbol{z}_m\|^2+2d{\lambda_t}^{(S)4}\hat\rho_N^8\bar\alpha_t^2(1-\bar\alpha_t)^2+\text{poly}(d^{-1}))^{\frac{3}{2}}}\\
\lesssim&O(\frac{1}{\sqrt{d}}),
\end{aligned}\tag{37}$$

where the last step comes from that with high probability,

$$|\boldsymbol{x}_{i,j}^t-\boldsymbol{z}_{m,j}|\lesssim\hat\rho_N\sqrt{\log dNT},\tag{38}$$

and by Lemma 1 of Laurent & Massart (2000)

$$\Big|\|\boldsymbol{x}_t^i-\boldsymbol{z}_m\|^2-d\hat\rho_N^2\Big|\le 2\hat\rho_N^2\sqrt{d\log NT}+2\hat\rho_N^2\log NT.\tag{39}$$

Hence,

$$\begin{aligned}
&g(\boldsymbol{\Theta}^{(S)};\beta,\boldsymbol{x}_0^i,t)\\
=&\Psi\Big(\frac{d\beta-m_Y(\boldsymbol{x}_t^i)}{s_Y(\boldsymbol{x}_t^i)}\Big)+O(d^{-1/2})\\
=&\Psi\Big(\frac{d(\beta-\frac{(1-\bar\alpha_t)\hat\rho_N^2}{2(\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t)}-\frac{1}{2}(\frac{1-\bar\alpha_t}{\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t})^2(\|\boldsymbol{x}_t^i-\boldsymbol{z}_m^{(S)}\|^2/d-\hat\rho_N^2))+\text{poly}(d^{-1})}{\sqrt{d}\cdot(\frac{\hat\rho_N^2(1-\bar\alpha_t)\sqrt{\bar\alpha_t}\sqrt{\bar\alpha_t\hat\rho_N^4+2(1-\bar\alpha_t)\|\boldsymbol{x}_t^i-\boldsymbol{z}_m^{(S)}\|^2/d}}{\sqrt{2}(\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t)^2}+\text{poly}(d^{-1})}\Big)\\
&+O(d^{-1/2}).
\end{aligned}\tag{40}$$

If we want $g(\boldsymbol{\Theta}^{(S)};\beta,\boldsymbol{x}_0^i,t)\ge 1-\gamma$, we need

$$\begin{aligned}
&\frac{d(\beta-\frac{(1-\bar\alpha_t)\hat\rho_N^2}{2(\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t)}-\frac{1}{2}(\frac{1-\bar\alpha_t}{\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t})^2(\|\boldsymbol{x}_t^i-\boldsymbol{z}_m^{(S)}\|^2/d-\hat\rho_N^2))+\text{poly}(d^{-1})}{\sqrt{d}\cdot(\frac{\hat\rho_N^2(1-\bar\alpha_t)\sqrt{\bar\alpha_t}\sqrt{\bar\alpha_t\hat\rho_N^4+2(1-\bar\alpha_t)\|\boldsymbol{x}_t^i-\boldsymbol{z}_m^{(S)}\|^2/d}}{\sqrt{2}(\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t)^2}+\text{poly}(d^{-1})}\\
\ge&-\Psi^{-1}(\gamma+O(d^{-1/2}))
\end{aligned}\tag{41}$$

With high probability,

$$\|\boldsymbol{x}_t^i-\boldsymbol{z}_m^{(S)}\|^2/d\le\hat\rho_N^2(1+O(\sqrt{\frac{\log NT}{d}})).\tag{42}$$

Then, we need

$$\begin{aligned}
&\frac{1}{2}(\frac{1-\bar\alpha_t}{\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t})^2(\|\boldsymbol{x}_t^i-\boldsymbol{z}_m^{(S)}\|^2/d-\hat\rho_N^2)\\
\le&\beta-(-\Psi^{-1}(\gamma+O(d^{-1/2}))S_{i,N}^t\cdot\frac{1}{\sqrt{d}}-\frac{1}{2}\frac{(1-\bar\alpha_t)\hat\rho_N^2}{\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t}.
\end{aligned}\tag{43}$$

When $N>M$, a sufficient condition is

$$\frac{\|\boldsymbol{x}_t^i-\bar{\boldsymbol{\mu}}_m\|^2-d\hat\rho_N^2}{\hat\rho_N^2\sqrt{2d}}\le\frac{\sqrt{d}(\beta-\frac{1}{2}\frac{(1-\bar\alpha_t)\hat\rho_N^2}{\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t})-(-\Psi^{-1}(\gamma+O(d^{-1/2})))S_N^t}{\sqrt{2}\hat\rho_N^2\cdot(\frac{1-\bar\alpha_t}{\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t})^2/2}.\tag{44}$$

Therefore, by Berry–Esseen Theorem, we can obtain

$$\begin{aligned}
&\Pr(g(\boldsymbol{\Theta}^{(S)};\beta,\boldsymbol{x}_0^i,t)\ge 1-\gamma)\\
=&\Psi\Big(\frac{\sqrt{d}(\beta-\frac{1}{2}\frac{(1-\bar\alpha_t)\hat\rho_N^2}{\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t})-(-\Psi^{-1}(\gamma+O(d^{-1/2})))S_N^t}{(\frac{1-\bar\alpha_t}{\bar\alpha_t\hat\rho_N^2+1-\bar\alpha_t})^2\cdot\sqrt{2}\hat\rho_N^2/2}\Big)+O(d^{-1/2}).
\end{aligned}\tag{45}$$

Since $\hat{\rho}_N^2 = \rho^2(1 - O(M/N))$, we have

$$\mathcal{M}_N(\mathbf{\Theta}^{(S)}; \beta, t) = \Psi\Big(\frac{\sqrt{d}(\beta - R_N^t) + \Psi^{-1}(\gamma + O(d^{-1/2}))S_N^t}{B_N^t}\Big) + O(d^{-1/2}). \tag{46}$$

If $M = N$, we have

$$\|\boldsymbol{f}(\mathbf{\Theta}^{(S)}; \boldsymbol{x}_t^i, t) - \boldsymbol{x}_0^i\|^2 \leq \text{poly}(d^{-1}). \tag{47}$$

Then,

$$g(\mathbf{\Theta}^{(S)}; \beta, \boldsymbol{x}_0^i, t) = \Psi(\text{poly}(d)\beta - \text{poly}(d^{-1})) + O(d^{-1/2}). \tag{48}$$

Hence, as long as $\beta \gtrsim \text{poly}(d^{-1})$, we have

$$\Pr(g(\mathbf{\Theta}^{(S)}; \beta, \boldsymbol{x}_0^i, t) \geq 1 - \gamma) = 1. \tag{49}$$

C.4 Proof of Corollary 2

Proof. Since $R_{\text{Oracle}}^t > \text{poly}(d^{-1})$, we have $\mathcal{M}_N(R_{\text{Oracle}}^t, t) = 1$. Note that

$$R_{\text{Oracle}}^t - R_N^t = \Theta(\frac{M\rho^2}{N}). \tag{50}$$

Then, since $M\rho^2\sqrt{d}/N \to \infty$, we have

$$\frac{\sqrt{d}(R_{\text{Oracle}}^t - R_N^t) + \Psi^{-1}(\gamma + O(d^{-1/2}))S_N^t}{B_N^t} \to \infty. \tag{51}$$

Therefore,

$$\mathcal{M}_N(R_{\text{Oracle}}^t, t) \to 1. \tag{52}$$

When $N/(M\rho^2\sqrt{d}) \to\to$, we have $N \to \infty$, and

$$\frac{\sqrt{d}(R_{\text{Oracle}}^t - R_N^t) + \Psi^{-1}(\gamma + O(d^{-1/2}))S_N^t}{B_N^t} \to \frac{\Psi^{-1}(\gamma + O(d^{-1/2}))S_\infty^t}{B_\infty^t}. \tag{53}$$

Therefore, we have

$$\mathcal{M}_N(\mathbf{\Theta}^{(S)}; R_{\text{Oracle}}^t, t) \to \Psi(\frac{\Psi^{-1}(\gamma + O(d^{-1/2}))S_\infty^t}{B_\infty^t}) + O(d^{-1/2}). \tag{54}$$

□

C.5 Proof of Proposition 1

Proof. Given (22) and (26), by the triangle inequality, we can derive Proposition 1. □

C.6 Proof of Proposition 2

Proof. Let $v^t = (1 - \bar{\alpha})\lambda_t$. We know that for $\boldsymbol{x}$ with $Y = m$,

$$\Big\|\sum_{j=1}^{M} \text{softmax}(-\frac{\lambda_t^{(S)}}{2}\|\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j^{(S)}\|^2)\boldsymbol{z}_j^{(S)} - \boldsymbol{z}_m^{(S)}\Big\|^2 \leq O(d\epsilon). \tag{55}$$

Then, we have

$$\begin{aligned}
&\|\boldsymbol{f}(\mathbf{\Theta}^{(S)}; \boldsymbol{x}_t, t) - ((1 - v^t)\boldsymbol{x}_t/\sqrt{\bar{\alpha}_t} + v^t\bar{\boldsymbol{\mu}}_m)\|^2 \\
=&\Big\|v^t(\sum_{j=1}^{M} \text{softmax}(-\frac{\lambda_t^{(S)}}{2}\|\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j^{(S)}\|^2)\boldsymbol{z}_j^{(S)} - \bar{\boldsymbol{\mu}}_m)\Big\|^2 \\
\leq&\Big\|v^t(\sum_{j=1}^{M} \text{softmax}(-\frac{\lambda_t^{(S)}}{2}\|\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j^{(S)}\|^2)\boldsymbol{z}_j^{(S)} - \boldsymbol{z}_m^{(S)})\Big\|^2 + \Big\|v^t(\boldsymbol{z}_m^{(S)} - \bar{\boldsymbol{\mu}}_m)\Big\|^2 \\
\leq&O(d\epsilon).
\end{aligned} \tag{56}$$

□

D PROOF OF KEY LEMMAS

D.1 THE MOTIVATION OF (2)

Consider $\boldsymbol{x}_0|Y=m \sim \mathcal{N}(\boldsymbol{z}_m, \tau^2\boldsymbol{I}_d)$ with parameters $\boldsymbol{z}_m$ and τ to learn. In the DDPM setting, for a given time step $t \in [T]$ and any input data point $\boldsymbol{x}_0$, we sample

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, \tag{57}$$

where $\boldsymbol{\epsilon}_t \in \mathbb{R}^d$ is the additive white Gaussian noise with $\boldsymbol{\epsilon}_t \overset{i.i.d.}{\sim} \mathcal{N}(0, \boldsymbol{I}_d)$, and $\{\bar{\alpha}_t\}_{t=1}^T$ is the pre-determined noise scheduling coefficients. Hence, $\boldsymbol{x}_t|Y=m \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{z}_c, (\bar{\alpha}_t\tau^2+1-\bar{\alpha}_t)\boldsymbol{I}_d)$. Denote $\lambda_t = (\bar{\alpha}_t\tau^2+1-\bar{\alpha}_t)^{-1}$. Then,

$$p_t(\boldsymbol{x}|Y=m) = (\frac{\lambda_t}{2\pi})^{\frac{d}{2}}\exp(-\frac{\lambda_t}{2}\|\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2). \tag{58}$$

Hence, $p_t(\boldsymbol{x}) = \sum_{m=1}^C \Pr(Y=m)p_t(\boldsymbol{x}|Y=m) = \frac{1}{M}\sum_{m=1}^M p_t(\boldsymbol{x}|Y=m)$. By the Tweedie formula, we can obtain

$$\begin{aligned}
&\mathbb{E}[\boldsymbol{x}_0|\boldsymbol{x}_t\boldsymbol{x}] \\
=&\frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}+(1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}}p_t(\boldsymbol{x})) \\
=&\frac{1}{\sqrt{\bar{\alpha}_t}}\Big(\boldsymbol{x}+(1-\bar{\alpha}_t)\frac{\sum_{m=1}^M \exp(-\frac{\lambda_t}{2}\|\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)(-\lambda_t(\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m))}{\sum_{j=1}^M \exp(-\frac{\lambda_t}{2}\|\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2)}\Big) \\
=&\frac{1-\lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\boldsymbol{x} + (1-\bar{\alpha}_t)\lambda_t\sum_{m=1}^M \frac{\exp(-\frac{\lambda_t}{2}\|\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m}{\sum_{j=1}^M \exp(-\frac{\lambda_t}{2}\|\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2)}.
\end{aligned} \tag{59}$$

Therefore, given the input $\boldsymbol{x}$, the learning model can be mathematically written as

$$\boldsymbol{f}(\boldsymbol{\Theta};\boldsymbol{x},t) = \frac{1-\lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\boldsymbol{x} + (1-\bar{\alpha}_t)\lambda_t\sum_{m=1}^M \frac{\exp(-\frac{\lambda_t}{2}\|\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m}{\sum_{j=1}^M \exp(-\frac{\lambda_t}{2}\|\boldsymbol{x}-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2)}, \tag{60}$$

where $\boldsymbol{\Theta} = \{\{\lambda_t \in \mathbb{R}\}_{t=1}^T, \{\boldsymbol{z}_m \in \mathbb{R}^d\}_{m=1}^M\}$. One can obtain τ once learning $\{\lambda_t\}_{t=1}^T$. (2) can be viewed as a one-layer single-head Transformer with a residual term and $\{\boldsymbol{z}_m \in \mathbb{R}^d\}_{m=1}^M$ as learnable tokens.

D.2 PROOF OF LEMMA 2

Proof. Denote $b_j(\boldsymbol{x}_t^n, t) = \text{softmax}(-\frac{\lambda_t}{2}\|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2)$. By Assumption, we have that for $\boldsymbol{x}_t^n \sim \mathcal{N}(\boldsymbol{\mu}_m, (\bar{\alpha}_t\rho^2+(1-\bar{\alpha}))\boldsymbol{I})$ and $\boldsymbol{z}_m^{(0)}$ that is close to $\boldsymbol{\mu}_m$ and $\boldsymbol{z}_j^{(0)}$ that is close to another $\boldsymbol{\mu}_{m'}$, $m \neq m' \in [M]$, we can compute

$$\begin{aligned}
G_{m,t} :=& \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j^{(0)}\|^2 - \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m^{(0)}\|^2 \\
=& \bar{\alpha}_t(\|\boldsymbol{\mu}_m - \boldsymbol{z}_j^{(0)}\|^2 - \|\boldsymbol{\mu}_m - \boldsymbol{z}_m^{(0)}\|^2) + 2\sqrt{\bar{\alpha}_t}(\boldsymbol{z}_m^{(0)} - \boldsymbol{z}_j^{(0)})^\top(\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}_m).
\end{aligned} \tag{61}$$

Note that by the triangle inequality,

$$\|\boldsymbol{\mu}_m - \boldsymbol{z}_j^{(0)}\| \geq \|\boldsymbol{\mu}_m - \boldsymbol{\mu}_{m'}^{(0)}\| - \|\boldsymbol{\mu}_{m'}^{(0)} - \boldsymbol{z}_m^{(0)}\| \geq \sqrt{2d} - r_0\sqrt{d}. \tag{62}$$

Then,

$$\|\boldsymbol{\mu}_m - \boldsymbol{z}_j^{(0)}\|^2 - \|\boldsymbol{\mu}_m - \boldsymbol{z}_m^{(0)}\|^2 \geq (\sqrt{2d} - r_0\sqrt{d})^2 - r_0^2 d = 2d - 2\sqrt{2}r_0 d. \tag{63}$$

Meanwhile,

$$\|\boldsymbol{z}_m^{(0)} - \boldsymbol{z}_j^{(0)}\| \leq \sqrt{2d} + 2r_0\sqrt{d}. \tag{64}$$

With a probability of at least $1-\delta$,

$$(\boldsymbol{z}_m^{(0)} - \boldsymbol{z}_j^{(0)})^\top(\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}_m) \leq \sqrt{\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha}_t}(\sqrt{2d} + 2r_0\sqrt{d})\sqrt{2\log(2M/\delta)}, \tag{65}$$

for any $j \neq m$. Therefore,

$$G_{m,t} \geq \bar{\alpha}_t(2d - 2\sqrt{2d}r_0\sqrt{d}) - 2\sqrt{\bar{\alpha}_t(\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha})}(\sqrt{2d} + 2r_0\sqrt{d})\sqrt{2\log(2M/\delta)}. \quad (66)$$

To ensure that $b_m(\boldsymbol{x}_t^n, t) \geq 1 - \epsilon_0$ with a probability of $1 - \delta$, we need

$$G_{c,t} \geq \log \frac{M}{\epsilon_0}. \quad (67)$$

A sufficient condition can be

$$\bar{\alpha}_t(2d - 2\sqrt{2d}r_0\sqrt{d}) - 2\sqrt{\bar{\alpha}_t(\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha})}(\sqrt{2d} + 2r_0\sqrt{d})\sqrt{2\log(2M/\delta)} \geq \log \frac{M}{\epsilon_0}, \quad (68)$$

which holds if

$$r_0 \in (0, \frac{\sqrt{2}}{2}), \quad (69)$$

and if we choose $\epsilon_0 = \min\{\Theta(\epsilon), \text{poly}(d^{-1})\}$,

$$d \gtrsim \log \frac{M}{\delta} + \log \frac{M}{\epsilon}. \quad (70)$$

We first take the gradient with respect to λ_t.

$$\begin{aligned}
&\frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \lambda_t} \\
=&\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}(\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n)^\top(-\frac{\boldsymbol{x}_t^n}{\sqrt{\bar{\alpha}_t}} + \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m \\
&+ \lambda_t \sum_{m=1}^{M} \frac{\partial(b_m(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m)}{\partial \lambda_t}) \\
=&\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}(\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n)^\top(-\frac{\boldsymbol{x}_t^n}{\sqrt{\bar{\alpha}_t}} + \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m \\
&+ \frac{1}{2}\lambda_t \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t)\|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 - \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m).
\end{aligned} \quad (71)$$

Recall that for $\boldsymbol{x}_t^n$ with $\boldsymbol{\mu}_m$ as the mean, $b_m(\boldsymbol{x}_t^n, t) \geq 1 - \epsilon_0$. Then,

$$\frac{1}{2}\lambda_t \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t)\|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 - \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2) \lesssim \epsilon_0 d. \quad (72)$$

Note that

$$\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - (\frac{1 - \lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\boldsymbol{x}_t^n + (1-\bar{\alpha}_t)\lambda_t\boldsymbol{z}_m) = \sum_{j=1}^{C}(b_j(\boldsymbol{x}_t^n, t) - \mathbb{1}[j = m])\boldsymbol{z}_j := \bar{\boldsymbol{e}}_n^t, \quad (73)$$

where $\sum_{j=1}^{C} |b_j(\boldsymbol{x}_t^n, t) - \mathbb{1}[j = m]| \lesssim \epsilon_0$. Then,

$$\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n = \bar{\boldsymbol{e}}_n^t - \lambda_t(1-\bar{\alpha}_t)(\boldsymbol{x}_0^n - \boldsymbol{z}_m) + \frac{1 - \lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n. \quad (74)$$

Then,

$$
\begin{aligned}
&\frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \lambda_t} \\
=&\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}(\bar{\boldsymbol{e}}_n^t - \lambda_t(1-\bar{\alpha}_t)(\boldsymbol{x}_0^n - \bar{\boldsymbol{\mu}}_m) - \lambda_t(1-\bar{\alpha}_t)(\bar{\boldsymbol{\mu}}_m - \boldsymbol{z}_m) \\
&+ \frac{1-\lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n)^\top(-\frac{\boldsymbol{x}_t^n}{\sqrt{\bar{\alpha}_t}} + \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m \\
&+ \frac{1}{2}\lambda_t\sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t)\|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 - \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m) \\
=&\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}(\lambda_t(1-\bar{\alpha}_t)\|\boldsymbol{x}_0^n - \bar{\boldsymbol{\mu}}_m\|^2 - \frac{1-\lambda_t(1-\bar{\alpha}_t)}{\bar{\alpha}_t}(1-\bar{\alpha}_t)d) \\
&+ \frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\bar{\boldsymbol{e}}_n^{t\top}(-\frac{\boldsymbol{x}_t^n}{\sqrt{\bar{\alpha}_t}} + \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m + \frac{1}{2}\lambda_t\sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t) \\
&\cdot \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 - \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m) + \frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\lambda_t(1-\bar{\alpha}_t)(\boldsymbol{x}_0^n - \bar{\boldsymbol{\mu}}_m)^\top \\
&(\bar{\boldsymbol{\mu}}_m - \frac{\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n}{\sqrt{\bar{\alpha}_t}} - \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m - \frac{1}{2}\lambda_t\sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t) \\
&\cdot \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 - \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m) - \frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\lambda_t(1-\bar{\alpha}_t)(\bar{\boldsymbol{\mu}}_m - \boldsymbol{z}_m)^\top \\
&(-\frac{\boldsymbol{x}_t^n}{\sqrt{\bar{\alpha}_t}} + \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m + \frac{1}{2}\lambda_t\sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t)\|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 \\
&- \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m) \\
&+ \frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\frac{1-\lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^{n\top}(-\boldsymbol{x}_0^n + \sum_{m=1}^{M} b_m \\
&(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m + \frac{1}{2}\lambda_t\sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t)\|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 - \|\boldsymbol{x}_t^n \\
&- \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m) \\
=:&\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}(\lambda_t(1-\bar{\alpha}_t)\|\boldsymbol{x}_0^n - \bar{\boldsymbol{\mu}}_m\|^2 - \frac{1-\lambda_t(1-\bar{\alpha}_t)}{\bar{\alpha}_t}(1-\bar{\alpha}_t)d) \\
&+ G_1 + G_2 + G_3 + G_4.
\end{aligned}
\tag{75}
$$

We then bound G_1 to G_4. With a probability of at least $1-\delta$,

$$
\begin{aligned}
G_1 :=&\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\bar{\boldsymbol{e}}_n^{t\top}(-\frac{\boldsymbol{x}_t^n}{\sqrt{\bar{\alpha}_t}} + \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m + \frac{1}{2}\lambda_t\sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t) \\
&\cdot \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 - \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m),
\end{aligned}
\tag{76}
$$

with

$$
\begin{aligned}
|G_1| \lesssim&\frac{(1-\bar{\alpha}_t)}{T}\epsilon_0(d + (\frac{\sqrt{2}}{2} - r_0)^2)(1 + \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}} + \epsilon_0 d^2) \\
\lesssim&\frac{(1-\bar{\alpha}_t)}{T}\text{poly}(d^{-1}).
\end{aligned}
\tag{77}
$$

$$
\begin{aligned}
G_2 :=& \frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\lambda_t(1-\bar{\alpha}_t)(\boldsymbol{x}_0^n-\bar{\boldsymbol{\mu}}_m)^\top(\bar{\boldsymbol{\mu}}_m-\frac{\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n}{\sqrt{\bar{\alpha}_t}}-\sum_{m=1}^{M}b_m(\boldsymbol{x}_t^n,t)\boldsymbol{z}_m \\
&-\frac{1}{2}\lambda_t\sum_{m=1}^{M}b_m(\boldsymbol{x}_t^n,t)(\sum_{j=1}^{M}b_j(\boldsymbol{x}_t^n,t)\|\boldsymbol{x}_t^n-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2-\|\boldsymbol{x}_t^n-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m,
\end{aligned}
\tag{78}
$$

with

$$
\begin{aligned}
|G_2| \lesssim& \frac{\lambda_t(1-\bar{\alpha}_t)^2}{T}(\sqrt{2\log(2N/\delta)}(C\rho\sqrt{1+\frac{d}{N}}+\rho\epsilon_0(d+(\frac{\sqrt{2}}{2}-r_0)^2)) \\
&+\sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}\rho\sqrt{d\log\delta^{-1}}) \\
\lesssim& \frac{\lambda_t(1-\bar{\alpha}_t)^2}{T}\cdot\sqrt{d\log\delta^{-1}}.
\end{aligned}
\tag{79}
$$

$$
\begin{aligned}
G_3 :=& -\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\lambda_t(1-\bar{\alpha}_t)(\bar{\boldsymbol{\mu}}_m-\boldsymbol{z}_m)^\top(-\frac{\boldsymbol{x}_t^n}{\sqrt{\bar{\alpha}_t}}+\sum_{m=1}^{M}b_m(\boldsymbol{x}_t^n,t)\boldsymbol{z}_m \\
&+\frac{1}{2}\lambda_t\sum_{m=1}^{M}b_m(\boldsymbol{x}_t^n,t)(\sum_{j=1}^{M}b_j(\boldsymbol{x}_t^n,t)\|\boldsymbol{x}_t^n-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2-\|\boldsymbol{x}_t^n-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m) \\
=& -\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\lambda_t(1-\bar{\alpha}_t)(\bar{\boldsymbol{\mu}}_m-\boldsymbol{z}_m)^\top(\boldsymbol{z}_m-\bar{\boldsymbol{\mu}}_m-\frac{\boldsymbol{x}_t^n-\bar{\boldsymbol{\mu}}_m}{\sqrt{\bar{\alpha}_t}}+\sum_{j=1}^{M}|b_j(\boldsymbol{x}_t^n,t) \\
&-\mathbb{1}[j=m]|\boldsymbol{z}_m+\frac{1}{2}\lambda_t\sum_{m=1}^{M}b_m(\boldsymbol{x}_t^n,t)(\sum_{j=1}^{M}b_j(\boldsymbol{x}_t^n,t)\|\boldsymbol{x}_t^n-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 \\
&-\|\boldsymbol{x}_t^n-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m) \\
\geq& \frac{\lambda_t(1-\bar{\alpha}_t)^2}{T}\cdot(\|\bar{\boldsymbol{\mu}}_m-\boldsymbol{z}_m\|^2-C\rho\sqrt{\frac{d}{N}}(\sqrt{2\log(2N/\delta)}(\rho+\sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}})+\epsilon_0 d^2)) \\
\gtrsim& -\frac{\lambda_t(1-\bar{\alpha}_t)^2}{T}\frac{M\sqrt{d\log(N/\delta)}}{\sqrt{N}}.
\end{aligned}
\tag{80}
$$

$$
\begin{aligned}
G_4 :=& \frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\frac{1-\lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^{n\top}(-\boldsymbol{x}_0^n+\sum_{m=1}^{M}b_m(\boldsymbol{x}_t^n,t)\boldsymbol{z}_m \\
&+\frac{1}{2}\lambda_t\sum_{m=1}^{M}b_m(\boldsymbol{x}_t^n,t)(\sum_{j=1}^{M}b_j(\boldsymbol{x}_t^n,t)\|\boldsymbol{x}_t^n-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2-\|\boldsymbol{x}_t^n-\sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m),
\end{aligned}
\tag{81}
$$

with

$$
\begin{aligned}
|G_4| \lesssim& \frac{(1-\bar{\alpha}_t)^{\frac{3}{2}}(1-\lambda_t(1-\bar{\alpha}_t))}{T\sqrt{\bar{\alpha}_t}}(\sqrt{d}(1+\epsilon_0 d)+\rho\sqrt{d\log\delta^{-1}}) \\
\lesssim& \frac{(1-\bar{\alpha}_t)^2}{\bar{\alpha}_t^2}\sqrt{\frac{d\bar{\alpha}_t}{1-\bar{\alpha}_t}}\frac{(1+\rho\sqrt{\log\delta^{-1}})}{T}.
\end{aligned}
\tag{82}
$$

Denote

$$
\hat{\rho}_N^2=\frac{1}{dN}\sum_{m=1}^{M}\sum_{n\in[N]:Y_n=m}\|\boldsymbol{x}_0^n-\bar{\boldsymbol{\mu}}_m\|^2,
\tag{83}
$$

and

$$
\hat{\lambda}_t^N=\frac{1+\sqrt{\frac{\log\delta^{-1}}{d}}}{\bar{\alpha}_t(\hat{\rho}_N^2-\sqrt{\frac{\log\delta^{-1}}{d}}-\frac{M\sqrt{\log(N/\delta)}}{\sqrt{dN}})+1-\bar{\alpha}_t}.
\tag{84}
$$

Then,

$$\begin{aligned}
&\frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \lambda_t} \\
\geq &\frac{d(1-\bar{\alpha}_t)^2}{\bar{\alpha}_t T}(\lambda_t(\bar{\alpha}_t(\hat{\rho}_N^2 - \sqrt{\frac{\log \delta^{-1}}{d}} - \frac{M\sqrt{\log(N/\delta)}}{\sqrt{dN}}) + 1 - \bar{\alpha}_t) - 1 - \sqrt{\frac{\log \delta^{-1}}{d}}) \\
\geq &\frac{(1-\bar{\alpha}_t)^2(\bar{\alpha}_t(\hat{\rho}_N^2 - \sqrt{\frac{\log \delta^{-1}}{d}} - \frac{M\sqrt{\log(N/\delta)}}{\sqrt{dN}}) + 1 - \bar{\alpha}_t)}{\bar{\alpha}_t T}(\lambda_t - \hat{\lambda}_t^N),
\end{aligned} \tag{85}$$

$$\begin{aligned}
&{\lambda_t}^{(s)} - \hat{\lambda}_t^N \\
=&{\lambda_t}^{(s-1)} - \hat{\lambda}_t^N - \eta_\lambda \frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \lambda_t} \\
\leq &({\lambda_t}^{(s-1)} - \hat{\lambda}_t)(1 - \eta_\lambda d \frac{(1-\bar{\alpha}_t)^2(\bar{\alpha}_t(\hat{\rho}_N^2 - \sqrt{\frac{\log \delta^{-1}}{d}} - \frac{M\sqrt{\log(N/\delta)}}{\sqrt{dN}}) + 1 - \bar{\alpha}_t)}{\bar{\alpha}_t T}).
\end{aligned} \tag{86}$$

Then, if $\eta_\lambda \lesssim d^{-1}T\bar{\alpha}_T$, when $S \gtrsim \log \epsilon^{-1}$, we have

$$|{\lambda_t}^{(S)} - \hat{\lambda}_t^N| \leq \epsilon. \tag{87}$$

When $d \gtrsim \epsilon^{-2}M \cdot \log(N/\delta)$, we have

$$|\lambda_t^N - \hat{\lambda}_t^N| \leq \epsilon. \tag{88}$$

Therefore,

$$|{\lambda_t}^{(S)} - \lambda_t^N| \leq \epsilon. \tag{89}$$

We then take the gradient of $L_{N,T}(\boldsymbol{\Theta})$ with respect to $\boldsymbol{z}_m$. Note that

$$\begin{aligned}
&\frac{\partial b_j(\boldsymbol{x}, t)}{\partial \boldsymbol{z}_m} \\
=&\frac{\exp(-\frac{\lambda_t}{2}\|\boldsymbol{x} - \sqrt{\bar{\alpha}}\boldsymbol{z}_j\|^2)\mathbb{1}_{j=m}\sum_{k=1}^M \exp(-\frac{\lambda_t}{2}\|\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_k\|^2) \cdot \lambda_t\sqrt{\bar{\alpha}_t}(\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_k)}{(\sum_{k=1}^M \exp(-c_t\|\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2))^2} \\
&- \frac{\exp(-\frac{\lambda_t}{2}\|\boldsymbol{x} - \sqrt{\bar{\alpha}}\boldsymbol{z}_j\|^2) \cdot \exp(-\frac{\lambda_t}{2}\|\boldsymbol{x} - \sqrt{\bar{\alpha}}\boldsymbol{z}_m\|^2) \cdot \lambda_t\sqrt{\bar{\alpha}_t}(\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m)}{(\sum_{k=1}^M \exp(-\frac{\lambda_t}{2}\|\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_k\|^2))^2} \\
=&b_j(\boldsymbol{x}, t) \cdot \lambda_t\sqrt{\bar{\alpha}_t}(\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m)(\mathbb{1}_{j=m} - b_m(\boldsymbol{x}, t)).
\end{aligned} \tag{90}$$

Then, we have

$$\begin{aligned}
&\frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \boldsymbol{z}_m} \\
=&\frac{1}{NT}\sum_{n=1}^N\sum_{t=1}^T(\lambda_t(1-\bar{\alpha}_t)b_m(\boldsymbol{x}_t^n, t) + \lambda_t(1-\bar{\alpha}_t)\sum_{j=1}^M \frac{\partial b_j(\boldsymbol{x}, t)}{\partial \boldsymbol{z}_m}\boldsymbol{z}_j)^\top(\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n) \\
=&\frac{1}{NT}\sum_{n=1}^N\sum_{t=1}^T(\lambda_t(1-\bar{\alpha}_t)b_m(\boldsymbol{x}_t^n, t) + \lambda_t(1-\bar{\alpha}_t)\sum_{j=1}^M b_j(\boldsymbol{x}, t) \cdot \lambda_t\sqrt{\bar{\alpha}_t}(\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m) \\
&\cdot (\mathbb{1}_{j=m} - b_m(\boldsymbol{x}, t))\boldsymbol{z}_j)^\top(\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n) \\
=&\frac{1}{NT}\sum_{n=1}^N\sum_{t=1}^T\lambda_t(1-\bar{\alpha}_t)b_m(\boldsymbol{x}_t^n, t)((\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n) + (\boldsymbol{z}_m - \sum_{j=1}^M b_j(\boldsymbol{x}, t)\boldsymbol{z}_j)^\top \\
&\cdot (\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n)\lambda_t\sqrt{\bar{\alpha}_t}(\boldsymbol{x} - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m))
\end{aligned} \tag{91}$$

Note that given (70) and $r_0 \in (0, \sqrt{2}/2)$, we have $b_m(\boldsymbol{x}_t^n, t) \geq 1 - \epsilon_0$ with a probability of $1 - \delta$. Recall that

$$\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n = \bar{\boldsymbol{e}}_n^t - \lambda_t(1-\bar{\alpha}_t)(\boldsymbol{x}_0^n - \boldsymbol{z}_m) + \frac{1 - \lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n. \tag{92}$$

Denote that

$$\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t)\boldsymbol{z}_j = (1-\epsilon_0)\boldsymbol{z}_m + \sum_{j\neq m}\gamma_j \boldsymbol{z}_j, \tag{93}$$

we have, with probability of $1-\delta$,

$$\begin{aligned}
&\Big|(\boldsymbol{z}_m - \sum_{j=1}^{M} b_j(\boldsymbol{x}, t)\boldsymbol{z}_j)^\top (\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n)\Big| \\
=&\Big|(\epsilon_0 \boldsymbol{z}_m + \sum_{j\neq m}\gamma_j \boldsymbol{z}_j)^\top (\bar{\boldsymbol{e}}_n^t - \lambda_t(1-\bar{\alpha}_t)(\boldsymbol{x}_0^n - \bar{\boldsymbol{\mu}}_m) - \lambda_t(1-\bar{\alpha}_t)(\bar{\boldsymbol{\mu}}_m - \boldsymbol{z}_m) \\
&+ \frac{1-\lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n)\Big| \\
\lesssim&\epsilon_0(\sqrt{d} + (\frac{\sqrt{2}}{2} - r_0))(\sqrt{\log\frac{N}{\delta}}(\rho + \frac{1}{\sqrt{\bar{\alpha}_t}}) + (\frac{\sqrt{2}}{2} - r_0) + \epsilon_0\sqrt{d}).
\end{aligned} \tag{94}$$

Based on the computation above, we can obtain

$$\begin{aligned}
&\frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \boldsymbol{z}_m} \\
=&\frac{|\{i\in[N]: Y_i = m\}|}{NT}\sum_{t=1}^{T}(-{\lambda_t}^2(1-\bar{\alpha}_t)^2(\bar{\boldsymbol{\mu}}_m - \boldsymbol{z}_m)) + \frac{1}{NT}\sum_{t=1}^{T}\sum_{n}^{N}\lambda_t(1-\bar{\alpha}_t) \\
&(\bar{\boldsymbol{e}}_n^t \pm \Theta(\epsilon_0)(\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n) + (\boldsymbol{z}_m - \sum_{j=1}^{M} b_j(\boldsymbol{x}, t)\boldsymbol{z}_j)^\top (\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n) \\
&\cdot \lambda_t\sqrt{\bar{\alpha}_t}(\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m))
\end{aligned} \tag{95}$$

Denote $\hat{\boldsymbol{R}}_m = \frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \boldsymbol{z}_m} - \frac{1}{CT}\sum_{t=1}^{T}{\lambda_t}^2(1-\bar{\alpha}_t)^2(\boldsymbol{z}_m - \bar{\boldsymbol{\mu}}_m)$. Then, with a high probability,

$$\begin{aligned}
&\Big\|\frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \boldsymbol{z}_m} - \frac{1}{CT}\sum_{t=1}^{T}{\lambda_t}^2(1-\bar{\alpha}_t)^2(\boldsymbol{z}_m - \bar{\boldsymbol{\mu}}_m)\Big\| \\
\lesssim&\epsilon_0(\sqrt{d} + (\frac{\sqrt{2}}{2} - r_0)) + \epsilon_0(\sqrt{d} + r_0)(\sqrt{\log\frac{N}{\delta}}(\rho + \frac{1}{\sqrt{\bar{\alpha}_t}}) + (\frac{\sqrt{2}}{2} - r_0) + \epsilon_0\sqrt{d})\frac{1}{\rho}\cdot\sqrt{d} \\
\leq&\min\{\epsilon, \text{poly}(d^{-1})\}.
\end{aligned} \tag{96}$$

We can derive

$$\begin{aligned}
&\|\boldsymbol{z}_m^{(s+1)} - \bar{\boldsymbol{\mu}}_m\| \\
=&\|\boldsymbol{z}_m^{(s)} - \bar{\boldsymbol{\mu}}_m - \eta_z\frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \boldsymbol{z}_m}\| \\
=&\|\boldsymbol{z}_m^{(s)} - \bar{\boldsymbol{\mu}}_m - \eta_z\frac{1}{CT}\sum_{t=1}^{T}{\lambda_t}^2(1-\bar{\alpha}_t)^2(\boldsymbol{z}_m^{(s)} - \bar{\boldsymbol{\mu}}_m) + \eta\hat{\boldsymbol{R}}_m\| \\
\leq&\|\boldsymbol{z}_m^{(s)} - \bar{\boldsymbol{\mu}}_m\|\cdot(1 - \eta_z\frac{1}{CT}\sum_{t=1}^{T}{\lambda_t}^2(1-\bar{\alpha}_t)^2) + \eta\epsilon \\
\leq&\|\boldsymbol{z}_m^{(0)} - \bar{\boldsymbol{\mu}}_m\|\cdot(1 - \eta_z\frac{1}{CT}\sum_{t=1}^{T}{\lambda_t}^2(1-\bar{\alpha}_t)^2)^{s+1} + \frac{\eta\epsilon}{\frac{1}{CT}\sum_{t=1}^{T}{\lambda_t}^2(1-\bar{\alpha}_t)^2}.
\end{aligned} \tag{97}$$

Then, when $\eta_z \lesssim C$, and $S \gtrsim \log d\epsilon^{-1}$, we have

$$\|\boldsymbol{z}_m^{(S)} - \bar{\boldsymbol{\mu}}_m\| \leq O(\epsilon). \tag{98}$$

We then compute the loss with the trained model. When $N \geq M$ and $|\{\boldsymbol{x}_0^i \in \mathcal{R} : Y_i = m\}| > 1$ for any $m \in [M]$, we have that if $d \gtrsim \epsilon^{-2} \log N$,

$$
\begin{aligned}
&\|\boldsymbol{f}(\boldsymbol{\Theta}^{(S)}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n\|^2 \\
=&\|\bar{\boldsymbol{e}}_n^t - \lambda_t(1-\bar{\alpha}_t)(\boldsymbol{x}_0^n - \bar{\boldsymbol{\mu}}_m) - \lambda_t(1-\bar{\alpha}_t)(\bar{\boldsymbol{\mu}}_m - \boldsymbol{z}_m) + \frac{1-\lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n\|^2 \\
\leq& d(\epsilon_0^2 + \hat{\rho}^2(1+\epsilon)(\frac{1-\bar{\alpha}_t + \sqrt{\frac{\log\delta^{-1}}{d}}}{\bar{\alpha}_t(\hat{\rho}_N^2 - \sqrt{\frac{\log\delta^{-1}}{d}} - \frac{M\sqrt{\log(N/\delta)}}{\sqrt{dN}} - \Theta(\epsilon)) + 1 - \bar{\alpha}_t} + \epsilon)^2 \\
&+ (\frac{\bar{\alpha}_t(\hat{\rho}_N^2 - \sqrt{\frac{\log\delta^{-1}}{d}} - \frac{M\sqrt{\log(N/\delta)}}{\sqrt{dN}} - \Theta(\epsilon))}{\bar{\alpha}_t(\hat{\rho}_N^2 - \sqrt{\frac{\log\delta^{-1}}{d}} - \frac{M\sqrt{\log(N/\delta)}}{\sqrt{dN}} - \Theta(\epsilon)) + 1 - \bar{\alpha}_t} + \epsilon)^2 \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}) + o(d) \\
\leq& d \cdot \frac{(1-\bar{\alpha}_t)\hat{\rho}_N^2}{\bar{\alpha}_t\hat{\rho}_N^2 + 1 - \bar{\alpha}_t} + d \cdot O(\epsilon).
\end{aligned} \tag{99}
$$

Therefore, with a high probability,

$$
L_{N,T}(\boldsymbol{\Theta}^{(S)}) = \frac{d}{2T}\sum_{t=1}^{T} \frac{\hat{\rho}_N^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t\hat{\rho}_N^2 + 1 - \bar{\alpha}_t} + \Theta(d\epsilon). \tag{100}
$$

We can also obtain that

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}^0,\boldsymbol{\epsilon}}[\|\boldsymbol{f}(\boldsymbol{\Theta}^{(S)}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n\|^2/d] \\
=&\frac{\rho^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha}_t} + (\rho^2 + \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t})(\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t\hat{\rho}^2 + 1 - \bar{\alpha}_t} - \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha}_t})^2 \\
&+ (\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t\hat{\rho}^2 + 1 - \bar{\alpha}_t})^2 \cdot \frac{M\rho^2}{N} + O(\epsilon),
\end{aligned} \tag{101}
$$

and

$$
\begin{aligned}
&\bar{L}(\boldsymbol{\Theta}^{(S)}) \\
=&\frac{1}{2T}\sum_{t=1}^{T} \frac{d\rho^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha}_t} + O(1) \cdot \frac{1}{2T}\sum_{t=1}^{T} d\Big((\rho^2 + \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t})(\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t\hat{\rho}^2 + 1 - \bar{\alpha}_t} \\
&- \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha}_t})^2 + (\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t\hat{\rho}^2 + 1 - \bar{\alpha}_t})^2 \cdot \frac{\rho^2}{N}\Big) + O(d\epsilon).
\end{aligned} \tag{102}
$$

Note that $(Nd/\rho)\hat{\rho}_N^2 \sim \chi^2_{d(N-M)}$ with a probability of $1-\delta$,

$$
|\hat{\rho}_N^2 - (1 - \frac{M}{N})\rho^2| \leq \rho^2 \cdot \Theta(\frac{\log MT/\delta}{dN}) + \rho^2 \cdot \Theta(\sqrt{\frac{\log MT\delta^{-1}}{dN}}) < \Theta(\frac{M\rho^2}{N}), \tag{103}
$$

where the last step holds if $d \gtrsim (N/M^2)\log MT\delta^{-1}$. Then,

$$
\begin{aligned}
&\frac{1}{2T}\sum_{t=1}^{T} \frac{\rho^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha}_t} - \frac{1}{2T}\sum_{t=1}^{T} \frac{\hat{\rho}_N^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t\hat{\rho}_N^2 + 1 - \bar{\alpha}_t} \\
=&\frac{1}{2T}\sum_{t=1}^{T} \frac{(\rho^2 - \hat{\rho}_N^2)(1-\bar{\alpha}_t)^2}{(\bar{\alpha}_t\hat{\rho}_N^2 + 1 - \bar{\alpha}_t)(\bar{\alpha}_t\rho^2 + 1 - \bar{\alpha}_t)} \\
=&\Theta(\frac{M\rho^2 + d^{-1}\log MT\delta^{-1}}{N}),
\end{aligned} \tag{104}
$$

and

$$
\bar{L}(\boldsymbol{\Theta}^{(S)}) - L_{N,T}(\boldsymbol{\Theta}^{(S)}) = \Theta(\frac{dM\rho^2 + \log MT\delta^{-1}}{N}). \tag{105}
$$

□

D.3 Proof of Lemma 3

Proof. When for the mode m, $|\{i \in [N] : Y_i = m\}| = 1$, denote $m^* \in \{i \in [N] : Y_i = m\}$. Then, we have $\boldsymbol{z}_m^{(0)} = \boldsymbol{x}_{m^*} = \bar{\boldsymbol{\mu}}_m$ and $\|\boldsymbol{z}_m^{(0)} - \bar{\boldsymbol{\mu}}_m\| = 0$. We can obtain that $b_m(\boldsymbol{x}_{m^*}^t, t) \geq 1 - \epsilon_0$. By (74), we can derive

$$\boldsymbol{f}(\boldsymbol{\Theta}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n = \bar{\boldsymbol{e}}_n^t + \frac{1 - \lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n. \tag{106}$$

Therefore, (75) is modified into

$$\begin{aligned}
&\frac{\partial L_{N,T}(\boldsymbol{\Theta})}{\partial \lambda_t} \\
=&\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}(\bar{\boldsymbol{e}}_n^t - \lambda_t(1-\bar{\alpha}_t)(\boldsymbol{x}_0^n - \boldsymbol{z}_m) \\
&+ \frac{1-\lambda_t(1-\bar{\alpha}_t)}{\sqrt{\bar{\alpha}_t}}\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n)^\top(-\frac{\boldsymbol{x}_t^n}{\sqrt{\bar{\alpha}_t}} + \sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)\boldsymbol{z}_m \\
&+ \frac{1}{2}\lambda_t\sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t)\|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 - \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m) \\
=:&\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}(\lambda_t(1-\bar{\alpha}_t)\|\boldsymbol{x}_0^n - \boldsymbol{z}_m\|^2 - \frac{1-\lambda_t(1-\bar{\alpha}_t)}{\bar{\alpha}_t}(1-\bar{\alpha}_t)d) + G_1 + G_2 + G_3
\end{aligned} \tag{107}$$

where G_1 is the same as (76), G_3 is the same as (82), and G_2 becomes

$$\begin{aligned}
G_2 :=&\frac{(1-\bar{\alpha}_t)}{NT}\sum_{n=1}^{N}\lambda_t(1-\bar{\alpha}_t)(\boldsymbol{x}_0^n - \boldsymbol{z}_m)^\top(\boldsymbol{z}_m - \frac{\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t^n}{\sqrt{\bar{\alpha}_t}} - \sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t)\boldsymbol{z}_j \\
&- \frac{1}{2}\lambda_t\sum_{m=1}^{M} b_m(\boldsymbol{x}_t^n, t)(\sum_{j=1}^{M} b_j(\boldsymbol{x}_t^n, t)\|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_j\|^2 - \|\boldsymbol{x}_t^n - \sqrt{\bar{\alpha}_t}\boldsymbol{z}_m\|^2)\boldsymbol{z}_m,
\end{aligned} \tag{108}$$

with

$$\begin{aligned}
|G_2| \lesssim&\frac{\lambda_t(1-\bar{\alpha}_t)^2}{dT}(\epsilon_0\sqrt{2d\log(2N/\delta)}\rho\sqrt{1-\frac{1}{N}} + \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}\rho\sqrt{1-\frac{1}{N}}\sqrt{d\log\delta^{-1}})\cdot\nu \\
\lesssim&\frac{\lambda_t(1-\bar{\alpha}_t)^2}{T}\cdot\sqrt{\frac{\log\delta^{-1}}{d}}\cdot\nu.
\end{aligned} \tag{109}$$

Here ν is defined as the fraction of modes with more than one data point. $0 \leq \nu \leq 1$. We can then similarly define

$$\hat{\lambda}_t^N = \frac{1 + \sqrt{\frac{\log\delta^{-1}}{d}}}{\bar{\alpha}_t(\hat{\rho}_N^2 - \nu\sqrt{\frac{\log\delta^{-1}}{d}}) + 1 - \bar{\alpha}_t}. \tag{110}$$

One can still derive that if $\eta_\lambda \lesssim d^{-1}T\bar{\alpha}_T$, when $S \gtrsim \log\epsilon^{-1}$, we have

$$|\lambda_t{}^{(S)} - \hat{\lambda}_t^N| \leq \epsilon. \tag{111}$$

When for every model $m \in [M]$, $|\{i \in [N] : Y_i = m\}| = 1$, i.e., $\hat{\rho}_N = \nu = 0$, we have $\hat{\rho}_N = 0$. Then, the equation of $\hat{\lambda}_t^N$ becomes

$$\hat{\lambda}_t^N = \frac{1 + \sqrt{\frac{\log\delta^{-1}}{d}}}{1 - \bar{\alpha}_t}. \tag{112}$$

Since $\boldsymbol{z}_m^{(0)} = \bar{\boldsymbol{\mu}}_m$, following (96), we have

$$\left\|\frac{\partial L_{N,T}(\boldsymbol{\Theta}^{(0)})}{\partial \boldsymbol{z}_m}\right\| \leq \min\{\Theta(\epsilon), \text{poly}(d^{-1})\}. \tag{113}$$

Then, by induction, we have that for any iteration $s > 1$,

$$\left\| \frac{\partial L_{N,T}(\boldsymbol{\Theta}^{(s)})}{\partial \boldsymbol{z}_m} \right\| \leq \min\{\Theta(\epsilon), \text{poly}(d^{-1})\}. \tag{114}$$

When $d \gtrsim \epsilon^{-1}$, we have

$$\|\boldsymbol{z}_m^{(S)} - \bar{\boldsymbol{\mu}}_m\| \leq O(\epsilon). \tag{115}$$

Then, similar to (99), we have

$$\|\boldsymbol{f}(\boldsymbol{\Theta}^{(S)}; \boldsymbol{x}_t^n, t) - \boldsymbol{x}_0^n)\|^2 \leq d \cdot \frac{(1-\bar{\alpha}_t)\hat{\rho}_N^2}{\bar{\alpha}_t \hat{\rho}_N^2 + 1 - \bar{\alpha}_t} + d \cdot O(\epsilon). \tag{116}$$

Therefore, with high probability,

$$L_{N,T}(\boldsymbol{\Theta}^{(S)}) = \frac{d}{2T} \sum_{t=1}^{T} \frac{\hat{\rho}_N^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t \hat{\rho}_N^2 + 1 - \bar{\alpha}_t} + \Theta(\epsilon). \tag{117}$$

$$\begin{aligned} &\bar{L}(\boldsymbol{\Theta}^{(S)}) \\ =&\frac{d}{2T} \sum_{t=1}^{T} \frac{\rho^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t \rho^2 + 1 - \bar{\alpha}_t} + \frac{d}{2T} \sum_{t=1}^{T} \Big((\rho^2 + \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t})(\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t \hat{\rho}^2 + 1 - \bar{\alpha}_t} - \frac{1-\bar{\alpha}_t}{\bar{\alpha}_t \rho^2 + 1 - \bar{\alpha}_t})^2 \\ &+ (\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t \hat{\rho}^2 + 1 - \bar{\alpha}_t})^2 \cdot \frac{\rho^2}{N} \Big) + O(d\epsilon). \end{aligned} \tag{118}$$

When $N = M$, we have

$$\frac{1}{2T} \sum_{t=1}^{T} \frac{\hat{\rho}_N^2(1-\bar{\alpha}_t)}{\bar{\alpha}_t \hat{\rho}_N^2 + 1 - \bar{\alpha}_t} = 0. \tag{119}$$

We then have

$$\bar{L}(\boldsymbol{\Theta}^{(S)}) - L_{N,T}(\boldsymbol{\Theta}^{(S)}) = \Theta(d \frac{M\rho^2 + \log MT\delta^{-1}}{N}). \tag{120}$$

□