Discrete State Diffusion Models: A Sample Complexity Perspective

Aadithya Srikanth* **Mudit Gaur*** **Vaneet Aggarwal**
Purdue University, West Lafayette, IN
{srikan18, mgaur, vaneet}@purdue.edu
*Equal contribution

Abstract

Diffusion models have demonstrated remarkable performance in generating high-dimensional samples across domains such as vision, language, and the sciences. Although continuous-state diffusion models have been extensively studied both empirically and theoretically, discrete-state diffusion models, essential for applications involving text, sequences, and combinatorial structures, remain significantly less understood from a theoretical standpoint. In particular, all existing analyses of discrete-state models assume score estimation error bounds without studying sample complexity results. In this work, we present a principled theoretical framework for discrete-state diffusion, providing the first sample complexity bound of $\widetilde{\mathcal{O}}(\epsilon^{-2})$. Our structured decomposition of the score estimation error into statistical, approximation, optimization, and clipping components offers critical insights into how discrete-state models can be trained efficiently. This analysis addresses a fundamental gap in the literature and establishes the theoretical tractability and practical relevance of discrete-state diffusion models.

1 INTRODUCTION

Diffusion models (Song et al., 2021) have gained significant attention lately due to their empirical success in various generative modeling tasks. They have been used in computer vision and audio generation (Ulhaq and Akhtar, 2022; Bansal et al., 2023), text generation (Li et al., 2022), sequential data modeling (Tashiro et al., 2021), policy learning (Chen et al., 2024). They have also shown promise in life sciences (Jing et al., 2022; Malusare and Aggarwal, 2025; Malusare et al., 2025), and biomedical image reconstruction (Chung and Ye, 2022).

They model an unknown and unstructured distribution through forward and reverse stochastic processes. In the forward process, samples from the dataset are gradually corrupted to obtain a stationary distribution. In the reverse process, a well-defined noisy distribution is used as the initialization to iteratively produce samples that resemble the learnt distribution using the *score* function. Most of the earlier works have been on continuous-state diffusion models, where the data is defined on the Euclidean space $\mathcal{R}^d$, unlike the discrete data, which is defined on $\mathcal{S}^d$, where $\mathcal{S}$ is the discrete set of values each component can take and d is the number of components or dimension of the data.

Generative AI applications involving discrete data have grown rapidly in recent years. These include text generation (Zhou et al., 2024) and summarization (Do et al., 2025). Other examples include combinatorial optimization (Li et al., 2024), molecule generation and drug discovery (Malusare and Aggarwal, 2025), protein and DNA sequence design Alamdari et al. (2023); Avdeyev et al. (2023), and graph generation Qin et al. (2023), there has been an increased interest in diffusion with discrete data. For a broad review of related literature, see Ren et al. (2025b).

Chen and Ying (2024) initiated the theoretical study of score-based discrete-state diffusion models. They proposed a sampling algorithm for the hypercube $\{0,1\}^d$, using uniformization under the Continuous Time Markov Chain (CTMC) framework to establish convergence guarantees and bounds on iteration complexity. Zhang et al. (2025) further extended this theory to a general state space, $[S]^d$, and proposed a discrete-time sampling algorithm for high-dimensional data. They provided KL convergence bounds centered on truncation, discretization, and score esti-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

mation error decomposition. Ren et al. (2025a) proposed a Lévy-type stochastic integral framework for the error analysis of discrete-state diffusion models. They implemented both τ leaping and uniformization algorithms and presented error bounds in KL divergence, the first for the τ leaping scheme. However, it is important to note that these error analyses were performed under the assumption of access to an ϵ_{score}-accurate score estimator. Prior works by Chen and Ying (2024), Zhang et al. (2025), and Ren et al. (2025a) laid the foundation for the theoretical study and analyzed iteration complexity, while the sample complexity is an unexplored topic. In this work, we bridge this gap by providing the first sample complexity bounds for discrete-state diffusion models, offering rigorous insight into the number of samples required to estimate the score function to within a desired accuracy. We perform a detailed analysis of the score estimation error bound ϵ_{score}, which was simply assumed in earlier works.

Specifically, we seek an answer to the following question.

How many samples are required for a sufficiently expressive neural network to estimate the discrete-state score function well enough to generate high-quality samples such that the KL divergence between the data distribution and the generated marginal at the final step is at most ϵ?

We leverage the strong convexity of the negative entropy function over the closed set on which the true and approximate score functions are defined to upper bound the Bregman divergence of the score estimation error by the squared Euclidean norm. We further decompose this into *approximation, statistical, optimization, and clipping errors*. Approximation error arises from the limited capacity of the chosen function class to represent the target function. Statistical error results from having only a finite dataset, which leads to estimation inaccuracies. Optimization error reflects the inability to reach the global minimum during training. Clipping error is due to violating the constraints on the network output. We thus provide a practical context that involves limitations in neural network estimation, a limited number of dataset samples, and a finite number of stochastic gradient descent (SGD) steps.

To the best of our knowledge, this work offers the first theoretical investigation into the sample complexity of discrete-state diffusion models. We introduce a recursive analytical framework that traces the optimization error incurred at each step of SGD, allowing us to quantify the effect of performing only a limited number of optimization steps when learning the discrete-state score. Our approach bounds the discrepancy between the empirical and expected loss directly over a finite hypothesis class, thereby bypassing complexity measures that scale poorly with network size. An essential component of our analysis is the application of the Polyak–Łojasiewicz (PL) condition (Assumption 1), which ensures a quadratic lower bound on the error and is crucial for regulating the overall error in score estimation.

We summarize our main contributions as follows:

- **First sample complexity bounds for discrete-state diffusion models** To the best of our knowledge, this is the first work to present rigorous sample complexity analyses under reasonable and practical assumptions for discrete-state diffusion models. We derive an order optimal bound of $\widetilde{\mathcal{O}}(\epsilon^{-2})$, and thus providing deeper insight into the sample efficiency of discrete-state diffusion. In particular we prove that $\widetilde{\mathcal{O}}(\epsilon^{-2})$ many samples are enough for discrete-state diffusion models to generate samples from a distribution that is ϵ-close to the target distribution in KL divergence.

- **Principled error decomposition** We propose a structured decomposition of the score estimation error into approximation, statistical, optimization, and clipping components, enabling the characterization of how each factor contributes to the overall sample complexity.

2 RELATED WORK

2.1 Discrete-state Diffusion Models

Discrete-state diffusion models were initially studied as discrete-time Markov chains, which train and sample the model at discrete time points. Austin et al. (2021) first proposed Discrete Denoising Diffusion Probabilistic Models (D3PMs) as the discrete-state analog of the "mean predicting" DDPMs (Ho et al., 2020). These models were largely applied to fields other than language due to empirical challenges (Lou et al., 2024). Although these models offer various practical sampling algorithms, they lack flexibility and theoretical guarantees (Zhang et al., 2025). Campbell et al. (2022) first introduced the continuous-time (CTMC) framework, similar to the NCSNv3 by Song et al. (2021). The time-reversal of a forward CTMC is fully determined by its rate matrix and the forward marginal probability ratios, which together define the discrete-state score function.

2.2 Convergence and Theoretical Analysis of Discrete-state Diffusion Models

While convergence analysis for continuous diffusion models is well-established, theoretical understanding in the discrete setting remains comparatively underdeveloped. Campbell et al. (2022) provided one of the earliest error bounds for discrete-state diffusion by analyzing a τ-leaping sampling algorithm under total variation distance, assuming bounded forward probability ratios and L^∞ control on the rate matrix approximation. However, the bound exhibits at least quadratic growth in the dimension d. In contrast, Chen and Ying (2024) introduced a score-based sampling method for discrete-state diffusion models by simulating the reverse process via CTMC uniformization on the binary hypercube, where forward transitions involve independent bit flips. Their method yields convergence guarantees and algorithmic complexity bounds that scale nearly linearly with d, comparable to the best-known results in the continuous setting (Benton et al., 2024a). Zhang et al. (2025) and Ren et al. (2025a) perform KL divergence error analysis by decomposing into truncation, discretization, and approximation error, and provide iteration complexities of sampling algorithms. However, these analyses assume the score estimation error as simply bounded by a constant (denoted by ϵ_{score}), effectively bypassing the score-learning problem.

Several recent works have analyzed the sample complexity of score-based continuous diffusion models. For example, Gupta et al. (2024) presents a bound of $\widetilde{\mathcal{O}}(\epsilon^{-5})$ using a quantile-based reformulation and assuming access to the ERM. In addition, Gaur et al. (2025b) derives a sample complexity bound of $\widetilde{\mathcal{O}}(\epsilon^{-4})$ by relaxing the assumption of access to ERM. Extending these results to discrete-state diffusion models, we establish for the first time a sample complexity bound of $\widetilde{\mathcal{O}}(\epsilon^{-2})$.

Notation Lowercase letters are used to denote scalars, and boldface lowercase letters to represent vectors. The i-th entry of a vector $\mathbf{x}$ is denoted by x^i. We use $x^{\setminus i}$ to refer to all dimensions of x except the i-th, and $x^{\setminus i} \odot \hat{x}^i$ to denote a vector whose i-th dimension takes the value $\hat{x}^i$, while the other dimensions remain as $\mathbf{x}^{\setminus i}$. For a positive integer n, we denote $[n]$ as the set $\{1, 2, \ldots, n\}$, $1_n \in \mathbb{R}^n$ as the vector of ones, and $I_n \in \mathbb{R}^{n\times n}$ as the identity matrix. The notation $y \lesssim z$ means that there exists a universal constant $C' > 0$ such that $y \leq C'z$. We use $x \asymp y$ to indicate x is asymptotically on the order of y. The notation e_i refers to a one-hot vector with a 1 in the i-th position. $\delta_{x,y}$ denotes the Kronecker delta, which equals 1 if $x = y$ and 0 otherwise.

3 PRELIMINARIES AND PROBLEM FORMULATION

We begin this section by formally introducing the fundamentals of discrete-state diffusion through the lens of Continuous-time Markov Chain (CTMC).

3.1 Discrete-state diffusion processes

The probability distributions are considered over a finite support $\mathcal{X} = [S]^d$, where each sequence $x \in \mathcal{X}$ has d components drawn from a discrete set of size S. The forward marginal distribution q_t can be represented as a probability mass vector $q_t \in \Delta^N$, where Δ^N is the probability simplex in $\mathbb{R}^N$, and $N = S^d$ is the cardinality of the discrete-state space. The forward dynamics of this distribution are governed by a continuous-time Markov process described by the Kolmogorov forward equation (Campbell et al., 2022).

$$\frac{dq_t}{dt} = Q_t^\top q_t,\ q_0 \approx q_{\text{data}} \tag{1}$$

Here, $Q_t \in \mathbb{R}^{N\times N}$ is a time-dependent generator (or rate) matrix with non-negative off-diagonal entries and rows summing to zero. This ensures conservation of total probability mass as the distribution evolves over time. We assume a finite uniform departure rate as required by the uniformization algorithm, which bounds the state-dependent exit rates by a single global rate (Zhang et al., 2025; Chen and Ying, 2024).

$$\lambda = \sup_{x\in\mathcal{X}} \sum_{y\neq x} Q_t(x, y) < \infty \tag{2}$$

To approximate this continuous evolution in practice, a small-step Euler discretization is applied. The resulting transition probability for a step of size Δt is given by (Zhang et al., 2025; Lou et al., 2024).

$$q(x_{t+\Delta t} = y \mid x_t = x) = \delta\{x, y\} + Q_t(x, y)\Delta t + o(\Delta t) \tag{3}$$

Equation (3) captures the probability of transitioning from state x to y, $Q_t(x, y)$ governing the instantaneous transition rate and $q(x_{t+\Delta t} = y \mid x_t = x)$ is the infinitesimal transition probability of being in state y at time $t + \Delta t$ given state x at time t. The process also admits a well-defined reverse-time evolution (Kelly, 2011), with q_{T-t} the time-reversed marginal and is characterized by another diffusion matrix, $\overline{Q}_t$, as follows:

$$\frac{dq_{T-t}}{dt} = \overline{Q}_{T-t}^T q_{T-t},\ \overline{Q}_t(x, y) = \tfrac{q_t(y)}{q_t(x)} Q_t(y, x) \tag{4}$$

The ratio $\left(\frac{q_t(y)}{q_t(x)}\right)_{y\neq x} \in \mathbb{R}^{(N-1)}$ in Equation (4) is known as the *concrete score* $s_t(x)$ (Meng et al., 2022)

and is analogous to $\nabla_x \log(p_t)$ in continuous-state diffusion models (Song and Ermon, 2019). A neural network-based score estimator $\hat{s}_{\theta,t}(\cdot)$ of $s_t(\cdot)$ is learned by minimizing the score entropy loss (Lou et al., 2024; Zhang et al., 2025)for $t \in [0, T]$ as given by:

$$\mathcal{L}_{\mathrm{SE}}(\hat{s}_\theta) = \int_0^T \mathbb{E}_{x_t} \sum_{y \neq x_t} Q_t(y, x_t)\, D_I(s_t(x_t)_y \,\|\, \hat{s}_{\theta,t}(x_t)_y)\, dt \tag{5}$$

where the expectation is taken over $x_t \sim q_t$ and is characterized by the Bregman divergence $D_I(\cdot)$ unlike the L^2 loss widely adopted in the continuous case. The Bregman divergence $D_I(\cdot)$, with respect to the negative entropy function $I(\cdot)$, also known as the generalized I-divergence, is defined as:

$$D_I(\mathbf{x} \| \mathbf{y}) = \sum_{i=1}^{d} \left[-x^i + y^i + x^i \log \frac{x^i}{y^i} \right] \tag{6}$$

In Equation (6), $I(\cdot)$ refers to the negative entropy function, where $I(x) = \Sigma_{i=1}^d x^i \log x^i$ for a d dimensional data. It is important to note that the Bregman divergence does not satisfy the triangle inequality. We defer more details on the Bregman divergence and convexity of $I(\cdot)$ to Lemma 4 in Appendix B.

3.2 Forward process

The forward evolution $X = (X_t)_{t \geq 0}$ is formulated as a CTMC over the discrete state space $[S]^d$. It is initialized by drawing X_0 from the data distribution p_{data}. The marginal law at time t, denoted by $q_t = \mathrm{Law}(X_t)$, evolves according to the Kolmogorov forward equation (Equation (1)).

Since directly working with a rate matrix Q_t of size $S^d \times S^d$ is infeasible in high dimension, the model assumes a factorized structure (Zhang et al., 2025; Chen and Ying, 2024). In particular, the conditional distribution of the process can be written as $q_{t|s}(x_t \mid x_s) = \prod_{i=1}^d q^i_{t|s}(x^i_t \mid x^i_s)$, so that each coordinate evolves independently as its own CTMC, with forward rate Q^{tok}_t. Under this construction, the global rate matrix Q_t is sparse: transitions between states are only possible when their Hamming distance is one as adopted by Campbell et al. (2022). A common choice for the per-coordinate dynamics is the time-homogeneous uniform flip rate, given by

$$Q^{\mathrm{tok}} = \frac{1}{S} 1_S 1_S^\top - I_S \tag{7}$$

Combining these coordinate-wise generators produces the global time-homogeneous rate matrix Q, whose entries are given by (Zhang et al., 2025)

$$Q(x, y) = \begin{cases} \frac{1}{S}, & \text{if } \mathrm{Ham}(x, y) = 1, \\ \left(\frac{1}{S} - 1\right) d, & \text{if } x = y, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The chain evolves by allowing each coordinate to flip independently, but at any given instant only one coordinate flip occurs. Thus transitions always correspond to Hamming distance one. We defer the details on the factorisation and effective score space to Appendix F. As $t \to \infty$, the marginal q_t converges to the uniform distribution π^d over the entire state space $[S]^d$ (Chen and Ying, 2024; Zhang et al., 2025).

3.3 Reverse process

Formally, the true reverse process $(U_t)_{t \in [0,T]}$ begins at $U_0 \sim q_T$ and obeys $\mathrm{Law}(U_t) = q_{T-t}$. Its generator is denoted $Q^{\leftarrow}_t$ and is defined so that the reverse marginal q_{T-t} satisfies the reverse Kolmogorov equation as given in Equation (4). In analogy with the forward dynamics, the reverse generator $Q^{\leftarrow}_t(x, \tilde{x})$ only has nonzero entries when the states x and $\tilde{x}$ differ in exactly one coordinate. More precisely, it takes the form (Zhang et al., 2025)

$$Q^{\leftarrow}_t(x, \tilde{x}) \;=\; \sum_{i=1}^{d} Q^{\mathrm{tok}}(\tilde{x}^i, x^i)\, \delta\{x^{\setminus i}, \tilde{x}^{\setminus i}\} \frac{q_{T-t}(\tilde{x})}{q_{T-t}(x)} \tag{9}$$

Here, Q^{tok} is the time-homogenous per-coordinate rate matrix of the forward chain, the indicator $\delta\{x^{\setminus i}, \tilde{x}^{\setminus i}\}$ enforces that x and $\tilde{x}$ differ only in coordinate i. This construction shows that the reverse process also evolves by flipping one coordinate at a time, but the rates are adjusted by the relative likelihood of the target state compared to the current state under the reverse marginal q_{T-t}. This ratio defines the *discrete-state score function*, which is central to training the reverse-time model. From the structure of the rate matrix as defined above, Equation (5) can be compactly written as follows:

$$\mathcal{L}_{\mathrm{SE}}(\hat{s}_\theta) = \frac{1}{S} \int_0^T \mathbb{E}_{\mathbf{x}_t \sim q_t} D_I(s_t(\mathbf{x}_t) \,\|\, \hat{s}_{\theta,t}(\mathbf{x}_t))\, dt \tag{10}$$

In particular, Lou et al. (2024) further showed that the score entropy loss is equivalent to minimizing the path measure KL divergence between the true and approximate reverse processes.

3.4 Problem Formulation

Let the discrete-state score function be approximated by a parameterized family of functions $\mathcal{F}_\Theta = \{\hat{s}_\theta : \theta \in$

$\Theta\}$, where each $\hat{s}_\theta : [S]^d \times [0,T] \to \mathbb{R}^{d(S-1)}$ is typically represented by a neural network with depth D and width W. Where D indicates the number of layers in the neural network and W indicates the maximum number of neurons in any layer of the neural network. Given i.i.d. samples $\{x_i\}_{i=1}^n \sim p_{\text{data}}$ from the data distribution, the score network is trained by minimizing the following time-indexed loss:

$$\mathcal{L}_{\text{SE}}(\hat{s}_\theta) = \frac{1}{S}\int_0^T \mathbb{E}_{\mathbf{x}_t\sim q_t} D_I(s_t(\mathbf{x}_t) \,\|\, \hat{s}_{\theta,t}(\mathbf{x}_t))\, dt \quad (11)$$

where $q_t = \text{Law}(X_t)$ is the forward CTMC marginal at time t, s_t is the true discrete-state score function defined by

$$s_t(x)_{i,\hat{x}^i} = \frac{q_t(x^{\setminus i} \odot \hat{x}^i)}{q_t(x)},\ x \in [S]^d,\ i \in [d],\ \hat{x}^i \neq x^i \quad (12)$$

and $D_I(\cdot\|\cdot)$ is the Bregman divergence induced by negative entropy.

Reverse-time marginal process. Let $V = (V_t)_{t\in[0,T]}$ denote the learned reverse process constructed from score estimators $\hat{s}_\theta(\cdot)$, initialized from the noise distribution π^d. Its law at time t is denoted by

$$p_t = \text{Law}(V_t).$$

Objective. Our goal is to quantify how well the learned reverse-time process approximates the true data distribution p_{data} in terms of the KL divergence. Specifically, we aim to show the number of samples needed, so that with high probability,

$$D_{\text{KL}}(p_{\text{data}} \,\|\, p_T) \ \le\ \widetilde{\mathcal{O}}(\epsilon). \quad (13)$$

4 ASSUMPTIONS

To formally establish our sample complexity results, we begin by outlining the core assumptions that underlie our analysis. We then define both the population and empirical loss functions used in training score-based discrete-state diffusion models. This is followed by a decomposition of the total error into statistical, approximation, optimization, and clipping errors, which are individually analyzed in the upcoming sections. Let us define the population loss at time k for $k \in [0, K-1]$ as

$$\mathcal{L}_k(\theta) = \int_{kh}^{(k+1)h} \mathbb{E}_{\mathbf{x}_t\sim q_t}\left[D_I\left(s_k(\mathbf{x}_t) \,\|\, \hat{s}_{\theta,k}(\mathbf{x}_t)\right)\right] dt \quad (14)$$

The corresponding empirical loss is:

$$\widehat{\mathcal{L}}_k(\theta) = \frac{1}{n_k}\sum_{i=1}^{n_k} D_I\left(s_t(\mathbf{x}_{k,t,i}) \,\|\, \hat{s}_{\theta,k}(\mathbf{x}_{k,t,i})\right), \quad t \sim \text{Unif}[kh, (k+1)h] \quad (15)$$

where n_k is the sample size at each time step. $x_{k,t,i}$ denotes the forward-diffused sample obtained at a random time t within the k-th discretization interval, where i indexes the training example from a dataset of size n_k. Here, k denotes the discretization step corresponding to the interval $[kh, (k+1)h]$, and $t \sim \text{Unif}[kh, (k+1)h]$ is a uniformly sampled continuous time.

Assumption 1 (Polyak - Łojasiewicz (PL) condition.)**.** *The loss $\mathcal{L}_k(\theta)$ for all $k \in [0, K-1]$ satisfies the Polyak–Łojasiewicz condition, i.e., there exists a constant $\mu_k > 0$ such that*

$$\frac{1}{2}\|\nabla\mathcal{L}_k(\theta)\|^2 \ge \mu_k\left(\mathcal{L}_k(\theta) - \mathcal{L}_k(\theta^*)\right), \quad \forall\ \theta \in \Theta \quad (16)$$

where $\theta^* = \arg\min_{\theta\in\Theta} \mathcal{L}_k(\theta)$ denotes the global minimizer of the population loss.

The Polyak-Łojasiewicz (PL) condition is significantly weaker than strong convexity and is known to hold in many non-convex settings, including overparameterized neural networks (Liu et al., 2022). Prior works in continuous-state diffusion models such as Gupta et al. (2024) and Block et al. (2020) implicitly assume access to an exact empirical risk minimizer (ERM) for score function estimation, as reflected in their sample complexity analyses (see Assumption A2 in Gupta et al. (2024) and the definition of $\hat{f}$ in Theorem 13 of Block et al. (2020)). This assumption, however, introduces a major limitation for practical implementations, where the exact ERM is not attainable. In contrast, the PL condition allows us to derive sample complexity bounds under realistic optimization dynamics, without requiring exact ERM solutions.

Assumption 2 (Smoothness and bounded gradient variance of the score loss.)**.** *For all $k \in [0, K-1]$, the population loss $\mathcal{L}_k(\theta)$ is κ-smooth with respect to the parameters θ, i.e., for all $\theta, \theta' \in \Theta$*

$$\|\nabla\mathcal{L}_k(\theta) - \nabla\mathcal{L}_k(\theta')\| \le \kappa\|\theta - \theta'\| \quad (17)$$

We assume that the estimators of the gradients $\nabla\mathcal{L}_k(\theta)$ have bounded variance.

$$\mathbb{E}\|\nabla\widehat{\mathcal{L}}_k(\theta) - \nabla\mathcal{L}_k(\theta)\|^2 \le \sigma^2 \quad (18)$$

The bounded gradient variance assumption is a standard assumption used in SGD convergence results such as Koloskova et al. (2022) and Ajalloeian and Stich (2020). This assumption holds for a broad class of neural networks under mild conditions, such as Lipschitz-continuous activations (e.g., GELU), bounded inputs, and standard initialization schemes (Allen-Zhu et al., 2019; Du et al., 2019). The smoothness condition ensures that the gradients of the loss function do not

change abruptly, which is crucial for stable updates during optimization. This stability facilitates controlled optimization dynamics when using first-order methods like SGD or Adam, ensuring that the learning process proceeds in a well-behaved manner.

Assumption 3 (Bounded initial score)**.** *The data distribution p_{data} has full support on $\mathcal{X}$, and there exists a constant $B > 0$, such that for all $\mathbf{x} \in \mathcal{X}$, $i \in [d]$, and $x^i \neq \hat{x}^i \in [S]$,*

$$s_0(\mathbf{x})_{i,\hat{x}^i} = \frac{q_0\big(\mathbf{x}^{\setminus i} \odot \hat{x}^i\big)}{q_0(\mathbf{x})} = \frac{p_{data}\big(\mathbf{x}^{\setminus i} \odot \hat{x}^i\big)}{p_{data}(\mathbf{x})} \in \left(\frac{1}{B}, B\right) \tag{19}$$

We assume the data distribution p_{data} has full support and satisfies a uniform score bound independent of the dimension d (Zhang et al., 2025; Chen and Ying, 2024). Consequently, our results in Theorem 1 do not require early stopping, and we set $\delta = 0$. Even though Zhang et al. (2025) do not explicitly assume a lower bound $1/B$, they require it for their Proposition 3 to hold. This assumption ensures well-posedness of the score function at initialization and enables dimension-free control of the score magnitude. Additionally, $\kappa_i = \frac{(p^i_{\text{data}})_{\max}}{(p^i_{\text{data}})_{\min}}$ is the ratio of the largest to smallest entry in the marginal distribution of the i-th dimension, measuring how unbalanced that marginal is and $\kappa^2 = \sum_{i=1}^d \kappa_i^2$ aggregates this imbalance across all dimensions. Lou et al. (2024) and Zhang et al. (2025) show that minimizing the discrete score entropy loss Equations (5) and (10) is equivalent to minimizing the KL divergence between the path measures of the true reverse and sampling diffusion processes, i.e., the distributions over their entire trajectories.

5 THEORETICAL RESULTS

We now present our main theoretical results, establishing finite-sample complexity bounds and KL convergence analyses for the discrete-state diffusion model under the stated assumptions.

Theorem 1 (Sample complexity and KL guarantee)**.** *Suppose Assumptions 1, 2, and 3 hold. With a probability at least $1 - \lambda$ the KL divergence between p_{data} and p_T is bounded by*

$$D_{KL}(p_{data} \,\|\, p_T) \lesssim de^{-T} \log S + C\kappa^2 S^2 h^2 T + \frac{M\lambda T h}{S} + \frac{C^3}{S} \sum_{k=0}^{K-1} h\, \mathcal{O}\left((W)^L \big((S-1)d + \tfrac{L}{W}\big) \sqrt{\frac{\log\big(\frac{2K}{\gamma}\big)}{n_k}} \right) \tag{20}$$

where $M = C(S-1)d$, provided that the score function estimator $\hat{s}_{\theta,t}(\cdot)$ is parameterized by a neural network with sufficient expressivity such that its width $W \geq (S-1)d$, and the number of training samples per step n_k satisfies

$$n_k \;=\; \tilde{\Omega}\left(\frac{C^6}{S^2\epsilon^2}\, W^{2L} \left((S-1)d + \frac{L}{W} \right)^2 \right)$$

In this case, with probability at least $1 - \gamma$, the distribution p_T obtained from simulating the reverse CTMC up to time T matches the target distribution p_{data} up to error $\tilde{O}(\epsilon)$.

Corollary 1. *Suppose Assumptions 1, 2, and 3 hold. For any $\epsilon > 0$, if we set*

$$T \asymp \log\left(\frac{d \log S}{\epsilon}\right), h \asymp \min\left\{ \left(\tfrac{\epsilon}{C\kappa^2 S^2 T}\right)^{1/2}, \tfrac{\epsilon S}{M\lambda T} \right\}$$

then the discrete diffusion model requires

$$\max\left\{ \sqrt{\tfrac{C\kappa^2 S^2}{\epsilon}} \left[\log(\tfrac{d\log S}{\epsilon})\right]^{3/2}, \tfrac{M\lambda}{\epsilon S} \left[\log(\tfrac{d\log S}{\epsilon})\right]^2 \right\}$$

steps to reach a distribution p_T such that

$$D_{\text{KL}}(p_{data} \,\|\, p_T) \;\lesssim\; \tilde{O}(\epsilon)$$

provided the following order optimal sample complexity bound is satisfied

$$n_k \;=\; \tilde{\Omega}\left(\frac{C^6}{S^2\epsilon^2}\, W^{2L} \left((S-1)d + \frac{L}{W} \right)^2 \right)$$

The first term in the KL divergence bound in (20) captures the truncation error due to the insufficient mixing of the forward process. The second term reflects the discretization error due to approximating the continuous-time reverse process with a fixed number of discrete steps of size h. The third term is a discretization error arising from replacing the continuous-time score-error integral by its discrete-time approximation. The fourth term quantifies the approximation, statistical, and optimization errors arising from finite network capacity, finite training data, and imperfect minimization of the empirical loss.

For some distributions with large KL separation, it implies from the standard mean–estimation lower bound that $n = \Omega(\varepsilon^{-2})$ samples are needed to reach ε-KL accuracy. We defer the details on the hardness of learning to Appendix E. This result provides a rigorous sample complexity guarantee for learning score-based discrete-state diffusion models under realistic training assumptions.

6 PROOF OF THEOREM 1

Let the ideal reverse process be $U = (U_t)_{t\in[0,T]}$, initialized at $U_0 \sim q_T$ with path measure $\mathbb{Q}$. The sampling process is $V = (V_t)_{t\in[0,T]}$, initialized at $V_0 \sim \pi^d$

with path measure $\mathbb{P}^{\pi^d}$, and the auxiliary process is $\widetilde{V} = (\widetilde{V}_t)_{t\in[0,T]}$, which shares the dynamics of V but starts from q_T, with path measure $\mathbb{P}^{q_T}$. At time t, we denote $q_t = \mathrm{Law}(U_{T-t})$ (true reverse-time marginal) and $p_t = \mathrm{Law}(V_t)$ (sampling marginal). By the data processing inequality, the KL divergence between the final-time marginals of the true reverse and sampling processes is always less than the KL divergence over the entire path measure, and can be written as:

$$D_{\mathrm{KL}}(q_0 \,\|\, p_T) \;\le\; D_{\mathrm{KL}}\Big(\mathbb{Q} \,\|\, \mathbb{P}^{\pi^d}\Big) \tag{21}$$

and by the chain rule of KL divergence,

$$D_{\mathrm{KL}}\Big(\mathbb{Q} \,\|\, \mathbb{P}^{\pi^d}\Big) = D_{\mathrm{KL}}\big(q_T \,\|\, \pi^d\big) + D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T}) \tag{22}$$

$D_{\mathrm{KL}}(q_T \,\|\, \pi^d)$ captures the mismatch in initial distributions (truncation error), and the second term $D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T})$ accounts for the discrepancy between the true and approximate reverse trajectories, including both discretization and approximation errors. By combining (21) and (22), we obtain:

$$\begin{aligned} D_{\mathrm{KL}}(q_0 \,\|\, p_T) &\le D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{\pi^d}) \\ &= D_{\mathrm{KL}}(q_T \,\|\, \pi^d) + D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T}) \end{aligned} \tag{23}$$

Therefore, to bound $D_{\mathrm{KL}}(q_0 \,\|\, p_T)$, it is sufficient to bound the two KL divergence terms on the right-hand side. Zhang et al. (2025) have shown in their Lemma 1 through a Girsanov-based method that $D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T})$ is equal to the Bregman divergence between the true reverse process and the discretized reverse sampling process (see Lemma 5). Specifically,

$$\begin{aligned} &D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T}) \\ &= \frac{1}{S}\sum_{k=0}^{K-1}\int_{kh}^{(k+1)h} \mathbb{E}_{x_t}\Big(D_I\big(s_t(\mathbf{x}_t) \,\|\, \hat{s}_{\theta,(k+1)h}(\mathbf{x}_t)\big)\Big)\,dt \end{aligned} \tag{24}$$

where the expectation is taken over $\mathbf{x}_t \sim q_t$ and $D_I\left(s_t(\mathbf{x}_t) \,\|\, \hat{s}_{\theta,(k+1)h}(\mathbf{x}_t)\right)$ is the Bregman divergence characterizing the distance between the true score at continuous time $s_t(\cdot)$ and the approximate score at discrete time points $\hat{s}_{\theta,(k+1)h}(\cdot)$, for $t \in [kh, (k+1)h]$. Here, h denotes the discretization step size, and k is the number of discrete time steps. Zhang et al. (2025) further decomposes Equation (24) into a score-movement term (discretization error) and a score-error term (approximation error), as given in (25), through the strong convexity of $I(\cdot)$. This decomposition facilitates analysis under their assumption of a bounded score estimation error $\epsilon_{\mathrm{score}}$. In contrast, we avoid this assumption and instead perform a rigorous analysis of the score-error term in Theorem 1 to determine the number of samples required to ensure that the overall KL divergence $D_{\mathrm{KL}}(q_0 \,\|\, p_T) \le \tilde{O}(\epsilon)$, as stated in Equation (23). Using the Bregman divergence property established in Lemma 4 (i), following decompostion of $D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T})$ is obtained:

$$\begin{aligned} &D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T}) \lesssim \\ &\frac{C^2}{S}\sum_{k=0}^{K-1}\int_{kh}^{(k+1)h} \mathbb{E}\, D_I\big(s_{(k+1)h}(\mathbf{x}_t) \,\|\, \hat{s}_{\theta,(k+1)h}(\mathbf{x}_t)\big)\,dt \\ &+ \frac{C}{S}\sum_{k=0}^{K-1}\int_{kh}^{(k+1)h} \mathbb{E}\,\|s_t(\mathbf{x}_t) - s_{(k+1)h}(\mathbf{x}_t)\|_2^2\,dt \end{aligned} \tag{25}$$

where the expectations are taken over $\mathbf{x}_t \sim q_t$. Earlier works such as Zhang et al. (2025) directly bound the first term on the right-hand side as $C^2\epsilon_{score}$. Specifically, their paper assumes access to an ϵ_{score}-accurate score estimator. The notation $y \lesssim z$ means that there exists a universal constant $C' > 0$ such that $y \le C'z$. However, further leveraging the strong convexity of $I(\cdot)$, along with Lemma 4 (ii), we further upper bound the path measure KL divergence in (25) as follows:

$$\begin{aligned} &D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T}) \lesssim \\ &\frac{C}{S}\sum_{k=0}^{K-1}\int_{kh}^{(k+1)h} \mathbb{E}_{x_t}\big\|s_t(\mathbf{x}_t) - s_{(k+1)h}(\mathbf{x}_t)\big\|_2^2\,dt + \\ &\frac{C^3}{S}\sum_{k=0}^{K-1}\int_{kh}^{(k+1)h} \mathbb{E}_{\mathbf{x}_t}\big\|s_{(k+1)h}(\mathbf{x}_t) - \hat{s}_{\theta,(k+1)h}(\mathbf{x}_t)\big\|_2^2\,dt \end{aligned} \tag{26}$$

where the expectations are taken over $\mathbf{x}_t \sim q_t$. Note that in order to satisfy the conditions required for Lemma 4, the score functions $s_t(x_t)$, $s_{(k+1)h}(x_t)$, and $\hat{s}_{\theta,(k+1)h}(x_t)$ must be bounded. While the true score functions are bounded through Lemma 6, the estimated score function is bounded component-wise to the range $[1/C, C]$ through hard-clipping of the final layer (Zhang et al., 2025). From the data processing inequality (23), we obtain the following bound on $D_{\mathrm{KL}}(p_{data} \,\|\, p_T)$ and is explained in detail in Appendix D:

$$\begin{aligned} D_{KL}(p_{data} \,\|\, p_T) \;&=\; D_{KL}(q_0 \,\|\, p_T) \\ &\lesssim\; de^{-T}\log S + C\kappa^2 S^2 h^2 T + \frac{M\lambda T h}{S} + \frac{C^3}{S}\sum_{k=0}^{K-1} hA_k \end{aligned} \tag{27}$$

The last term in (27) characterizes the score approximation error and $M = C(S-1)d$. To bound this error in terms of the number of samples n_k required for training, we now consider the following loss per discrete time step. Here, for notational simplicity, we refer to the discrete time step k obtained by freezing the

continuous time t at the upper limit of each interval. Specifically, we write, $k := (k+1)h$. To facilitate our analysis, we introduce the unbounded ($\overline{s}_{\theta,k}(\cdot)$) and the clipped ($\hat{s}_{\theta,k}(\mathbf{x}_k)$) score functions. During the reverse sampling process, we use the clipped score function $\hat{s}_{\theta,k}(\mathbf{x}_k)$ by projecting $\overline{s}_{\theta,k}(\cdot)$ into the interval $[1/C, C]$ component-wise. Using these definitions, we obtain the following loss per discrete time step k:

$$\begin{aligned} A_k &= \mathbb{E}_{\mathbf{x}_k\sim q_k}\left\|s_k(\mathbf{x}_k) - \hat{s}_{\theta,k}(\mathbf{x}_k)\right\|_2^2 \\ &\leq \underbrace{2\,\mathbb{E}_{\mathbf{x}_k\sim q_k}\left\|s_k(\mathbf{x}_k) - \overline{s}_{\theta,k}(\mathbf{x}_k)\right\|_2^2}_{B_k} \quad + \\ &\quad \underbrace{2\,\mathbb{E}_{\mathbf{x}_k\sim q_k}\left\|\overline{s}_{\theta,k}(\mathbf{x}_k) - \hat{s}_{\theta,k}(\mathbf{x}_k)\right\|_2^2}_{C_k} \end{aligned} \tag{28}$$

where the inequality in (28) follows from $(a+b)^2 \leq 2(a^2+b^2)$ by writing

$$\begin{aligned} A_k = \mathbb{E}_{\mathbf{x}_k\sim q_k}\Big\|&\big(s_k(\mathbf{x}_k) - \overline{s}_{\theta,k}(\mathbf{x}_k)\big) \\ &+ \big(\overline{s}_{\theta,k}(\mathbf{x}_k) - \hat{s}_{\theta,k}(\mathbf{x}_k)\big)\Big\|_2^2. \end{aligned}$$

The term B_k represents the difference between the true score function and the unbounded score function $\overline{s}_{\theta,k}(\cdot)$ obtained during training. The term C_k accounts for the error incurred due to violating the constraints of the network. We show in Lemma 13 that the clipping error $C_k \leq B_k$. This implies that

$$A_k \leq 4B_k \tag{29}$$

Now in order to upper bound B_k, we decompose it into three different components: approximation error, statistical error, and optimization error. In order to facilitate the analysis of the three terms listed above, we define the following neural network parameters.

$$\theta_k^a = \arg\min_{\theta\in\Theta} \int_{kh}^{(k+1)h} \mathbb{E}_{x_t}\left[D_I\left(s_k(\mathbf{x}_t) \,\|\, \overline{s}_{\theta,k}(\mathbf{x}_t)\right)\right] dt \tag{30}$$

$$\begin{aligned} \theta_k^b &= \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} D_I(s_t(\mathbf{x}_{k,t,i}) \,\|\, \overline{s}_{\theta,k}(\mathbf{x}_{k,t,i})) \\ &t \sim \mathrm{Unif}[kh, (k+1)h] \end{aligned} \tag{31}$$

Moreover, the term

$$\begin{aligned} B_k \;\leq\; &\underbrace{4\,\mathbb{E}_{\mathbf{x}_k\sim q_k}\left\|s_{\theta,k}^a(\mathbf{x}_k) - s_k(\mathbf{x}_k)\right\|_2^2}_{\mathcal{E}_k^{\text{approx}}} \\ &+ \underbrace{4\,\mathbb{E}_{\mathbf{x}_k\sim q_k}\left\|s_{\theta,k}^a(\mathbf{x}_k) - s_{\theta,k}^b(\mathbf{x}_k)\right\|_2^2}_{\mathcal{E}_k^{\text{stat}}} \\ &+ \underbrace{4\,\mathbb{E}_{\mathbf{x}_k\sim q_k}\left\|\overline{s}_{\theta,k}(\mathbf{x}_k) - s_{\theta,k}^b(\mathbf{x}_k)\right\|_2^2}_{\mathcal{E}_k^{\text{opt}}} \end{aligned} \tag{32}$$

where, $s_{\theta,k}^a(\cdot)$ and $s_{\theta,k}^b(\cdot)$ are the score functions associated with the parameters θ_k^a and θ_k^b. Here, the loss function given in Equation (30) is the expected value of the loss function minimized at training time to obtain the unbounded estimate of the score function $\overline{s}_\theta(\cdot)$. The loss function in Equation (31) is the empirical loss function that actually is minimized to obtain $\overline{s}_\theta(\cdot)$ at training time.

The term *Approximation error* $\mathcal{E}_k^{\text{approx}}$ captures the error due to the limited expressiveness of the function class $\{\hat{s}_\theta\}_{\theta\in\Theta}$. The *statistical error* $\mathcal{E}_k^{\text{stat}}$ is the error from using a finite sample size. The *optimization error* $\mathcal{E}_k^{\text{opt}}$ is due to not reaching the global minimum during training.

Drawing upon recent advances in statistical learning theory and neural network approximation bounds, we bound each of the above errors through Lemmas 1, 2, and 3 to bound the score-estimation error in terms of the number of samples and neural network parameters. The detailed proofs are provided in Appendix C and D. To prove Theorem 1, we bound each of the errors in (32) and subsequently use (26) to ensure $D_{\mathrm{KL}}(p_{data} \,\|\, p_T) \leq \tilde{O}(\epsilon)$. We introduce the following lemmas to bound the approximation, statistical, and optimization errors. The formal proofs corresponding to each of the lemmas are provided in Appendix C for completeness.

Lemma 1 (Approximation Error)**.** *Let W and L denote the width and depth of the neural network architecture, respectively, and we have that $W \geq (S-1)d$. Then, for all $k \in [0, K-1]$, we have*

$$\mathcal{E}_k^{\text{approx}} = 0. \tag{33}$$

The detailed proof of this lemma is provided in Appendix C.1. We demonstrate in Lemma C.1 that if the width of the neural network W is greater than the number of possible states $W \geq (S-1)d$, then it is possible to construct a neural network that is equal to any specified function defined on the $(S-1)d$ discrete points, where width W is defined as the maximum number of neurons in a layer of the neural network. The discrete nature of the score function allows us to obtain this results. In the case of diffusion models the score functions were defined on a continuous input space. In such a case, this error would have been exponential in the data dimension as is obtained in Jiao et al. (2023).

Lemma 2 (Statistical Error)**.** *Let n_k denote the number of samples used to estimate the score function at each diffusion step t_k. Then, then under Assumptions 1 and 2, for all $k \in [0, K-1]$, with probability at least*

$1-\gamma$, *we have*

$$\mathcal{E}_k^{\text{stat}} \leq \mathcal{O}\left(W^L\left((S-1)d+\frac{L}{W}\right)\sqrt{\frac{\log\left(\frac{2}{\gamma}\right)}{n_k}}\right) \quad (34)$$

The proof of this lemma follows from utilizing the definitions of s_t^a and s_t^b. Our novel analysis here allows us to avoid the exponential dependence on the neural networks parameters in the sample complexity. This is achieved without resorting to the quantile formulation and assuming access to the empirical risk minimizer of the score estimation loss which leads to a sample complexity of $\widetilde{\mathcal{O}}(\epsilon^{-2})$. The detailed proof of this lemma is provided in Appendix C.2.

Next, we have the following lemma to bound the optimization error.

Lemma 3 (Optimization Error)**.** *Let n_k be the number of samples used to estimate the score function at diffusion step k. If the learning rate $0 \leq \eta \leq \frac{1}{\kappa}$, then under Assumptions 1 and 2, for all $k \in [0, K-1]$, the optimization error due to imperfect minimization of the training loss satisfies with probability at least $1-\gamma$*

$$\mathcal{E}_k^{\text{opt}} \leq \mathcal{O}\left(W^L\left((S-1)d+\frac{L}{W}\right)\sqrt{\frac{\log\left(\frac{2}{\gamma}\right)}{n_k}}\right) \quad (35)$$

We leverage assumptions 1 and 2 alongside our unique analysis to derive an upper bound on $\mathcal{E}_{\text{opt}}$. The key here is our recursive analysis of the error at each stochastic gradient descent (SGD) step, which captures the error introduced by the finite number of SGD steps in estimating the score function, while prior works assumed no such error, treating the empirical loss minimizer as if it were known exactly. Here, the dependence on the number of SGD steps is implicit through n_k, since each SGD update at step k uses a fresh sample and the total number of optimization iterations equals the effective sample size n_k.

Now, using the Lemmas 1, 2, and 3, with probability at least $1-\gamma$, we have

$$\begin{aligned} A_k &= \mathbb{E}_{\mathbf{x}_k\sim q_k}\left\|s_k(\mathbf{x}_k)-\hat{s}_{\theta,k}(\mathbf{x}_k)\right\|_2^2 \\ &\lesssim \underbrace{\mathcal{O}\left(W^L\left((S-1)d+\frac{L}{W}\right)\sqrt{\frac{\log(2/\gamma)}{n_k}}\right)}_{\text{Statistical Error}} \quad (36) \\ &+ \underbrace{\mathcal{O}\left(W^L\left((S-1)d+\frac{L}{W}\right)\sqrt{\frac{\log(2/\gamma)}{n_k}}\right)}_{\text{Optimization Error}} \\ &\lesssim \mathcal{O}\left(W^L\left((S-1)d+\frac{L}{W}\right)\sqrt{\frac{\log(2/\gamma)}{n_k}}\right) \quad (37) \end{aligned}$$

Plugging the results of Lemma 1,2,3 into (32) gives us an upper bound on B_k. Plugging the upper bound on B_k (37) into (29) yields the bound on A_k. Finally, plugging the bound on A_k into (27) gives us the result required for Theorem 1.

7 CONCLUSION

We investigate the sample complexity of training discrete-state diffusion models using sufficiently expressive neural networks. We derive a sample complexity bound of $\widetilde{\mathcal{O}}(\epsilon^{-2})$, which, to our knowledge, is the first such result under this setting. Notably, our analysis avoids exponential dependence on the data dimension. For comparison, existing works assume a bound over the efficiency of the trained network, while we have decomposed this error, using SGD for training through a dataset of finite size.

Recent work by Sahoo et al. (2026) shows that rectified flow matching (RFM) also achieves an order optimal $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity, improving upon the $\widetilde{\mathcal{O}}(\epsilon^{-4})$ rate known for continuous flow matching (Gaur et al., 2025a). Developing guarantees for discrete flow matching counterparts remains an important direction for future work, with the potential to improve practical performance. Additionally, a more refined treatment of approximation error arising from finite-capacity neural networks remains an open and promising avenue for further study.

8 ACKNOWLEDGEMENT

This work is supported in part by the National Science Foundation under grant CCF-2149588.

REFERENCES

Ajalloeian, A. and Stich, S. U. (2020). On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*.

Alamdari, S., Thakkar, N., van den Berg, R., Lu, A. X., Fusi, N., Amini, A. P., and Yang, K. K. (2023). Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*.

Allen-Zhu, Z., Li, Y., and Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. (2021). Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993.

Avdeyev, P., Shi, C., Tan, Y., Dudnyk, K., and Zhou, J. (2023). Dirichlet diffusion score model for biological sequence generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 1276–1301. PMLR.

Bansal, A., Chu, H.-M., Schwarzschild, A., Sengupta, S., Goldblum, M., Geiping, J., and Goldstein, T. (2023). Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852.

Benton, J., Bortoli, V. D., Doucet, A., and Deligiannidis, G. (2024a). Nearly d-linear convergence bounds for diffusion models via stochastic localization. In *The Twelfth International Conference on Learning Representations*.

Benton, J., Deligiannidis, G., and Doucet, A. (2024b). Error bounds for flow matching methods. *Transactions on Machine Learning Research*.

Bing, X., He, X., and Wang, C. (2025). Kernel ridge regression with predicted feature inputs and applications to factor-based nonparametric regression. *arXiv preprint arXiv:2505.20022*.

Block, A., Mroueh, Y., and Rakhlin, A. (2020). Generative modeling with denoising auto-encoders and langevin sampling. *arXiv preprint arXiv:2002.00107*.

Campbell, A., Benton, J., De Bortoli, V., Rainforth, T., Deligiannidis, G., and Doucet, A. (2022). A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279.

Chen, H. and Ying, L. (2024). Convergence analysis of discrete diffusion model: Exact implementation through uniformization. *arXiv preprint arXiv:2402.08095*.

Chen, J., Ganguly, B., Xu, Y., Mei, Y., Lan, T., and Aggarwal, V. (2024). Deep generative models for offline policy learning: Tutorial, survey, and perspectives on future directions. *Transactions on Machine Learning Research*.

Chung, H. and Ye, J. C. (2022). Score-based diffusion models for accelerated mri. *Medical image analysis*, 80:102479.

Do, D. A., Tuan, L. A., Buntine, W., et al. (2025). Discrete diffusion language model for efficient text summarization. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6278–6290.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. (2019). Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR.

Gaur, M., Aggarwal, V., and Agarwal, M. (2023a). On the global convergence of fitted q-iteration with two-layer neural network parametrization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 11013–11049. PMLR.

Gaur, M., Bedi, A., Wang, D., and Aggarwal, V. (2024). Closing the gap: Achieving global convergence (Last iterate) of actor-critic under Markovian sampling with neural network parametrization. In *Proceedings of the 41st International Conference on Machine Learning*, pages 15153–15179.

Gaur, M., Bedi, A. S., Wang, D., and Aggarwal, V. (2023b). On the global convergence of natural actor-critic with two-layer neural network parametrization. *arXiv preprint arXiv:2306.10486*.

Gaur, M., Trivedi, P., Aeron, S., Bedi, A. S., Atia, G. K., and Aggarwal, V. (2025a). Generative modeling with continuous flows: Sample complexity of flow matching. *arXiv preprint arXiv:2512.01286*.

Gaur, M., Trivedi, P., Kunapuli, S., Bedi, A. S., and Aggarwal, V. (2025b). Improved sample complexity for diffusion model training without empirical risk minimizer access. *arXiv preprint arXiv:2505.18344*.

Gower, R., Sebbouh, O., and Loizou, N. (2021). Sgd for structured nonconvex functions: Learning rates, minibatching and interpolation. In *International Conference on Artificial Intelligence and Statistics*, pages 1315–1323. PMLR.

Guan, Y., Balasubramanian, K., and Ma, S. (2025). Mirror flow matching with heavy-tailed priors for generative modeling on convex domains. *arXiv preprint arXiv:2510.08929*.

Gupta, S., Parulekar, A., Price, E., and Xun, Z. (2024). Improved sample complexity bounds for diffusion model training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. PMLR.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *NeurIPS*.

Jiao, Y., Shen, G., Lin, Y., and Huang, J. (2023). Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics*, 51(2):691–716.

Jing, B., Corso, G., Chang, J., Barzilay, R., and Jaakkola, T. (2022). Torsional diffusion for molecular conformer generation. *Advances in neural information processing systems*, 35:24240–24253.

Kelly, F. P. (2011). *Reversibility and Stochastic Networks*. Cambridge University Press.

Koloskova, A., Stich, S. U., and Jaggi, M. (2022). Sharper convergence guarantees for asynchronous sgd for distributed and federated learning. *Advances in Neural Information Processing Systems*, 35:17202–17215.

Li, X., Thickstun, J., Gulrajani, I., Liang, P. S., and Hashimoto, T. B. (2022). Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343.

Li, Y., Guo, J., Wang, R., Zha, H., and Yan, J. (2024). Fast t2t: Optimization consistency speeds up diffusion-based training-to-testing solving for combinatorial optimization. *Advances in Neural Information Processing Systems*, 37:30179–30206.

Liu, C., Zhu, L., and Belkin, M. (2022). Loss landscapes and optimization in over-parameterized nonlinear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116.

Lou, A., Meng, C., and Ermon, S. (2024). Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32819–32848. PMLR.

Mai, T. T. (2025). Pac-bayesian risk bounds for fully connected deep neural network with gaussian priors. *arXiv preprint arXiv:2505.04341*.

Malusare, A. and Aggarwal, V. (2025). Improving molecule generation and drug discovery with a knowledge-enhanced generative model. *IEEE Transactions on Computational Biology and Bioinformatics*, 22(1):375–381.

Malusare, A., Punyamoorty, V., and Aggarwal, V. (2025). Augmenting generative models with biomedical knowledge graphs improves targeted drug discovery. *IEEE Transactions on Artificial Intelligence*.

Meng, C., Choi, K., Song, J., and Ermon, S. (2022). Concrete score matching: Generalized score matching for discrete data. *Advances in Neural Information Processing Systems*, 35:34532–34545.

Qin, Y., Vignac, C., and Frossard, P. (2023). Sparse training of discrete diffusion models for graph generation. *arXiv preprint arXiv:2311.02142*.

Ren, Y., Chen, H., Rotskoff, G. M., and Ying, L. (2025a). How discrete and continuous diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral framework. In *The Thirteenth International Conference on Learning Representations*.

Ren, Y., Chen, H., Zhu, Y., Guo, W., Chen, Y., Rotskoff, G. M., Tao, M., and Ying, L. (2025b). Fast solvers for discrete diffusion models: Theory and applications of high-order algorithms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Sahoo, H. K., Gaur, M., and Aggarwal, V. (2026). Order-optimal sample complexity of rectified flows. *arXiv preprint arXiv:2601.20250*.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *ICLR*.

Tashiro, Y., Song, J., Song, Y., and Ermon, S. (2021). Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, 34:24804–24816.

Ulhaq, A. and Akhtar, N. (2022). Efficient diffusion models for vision: A survey. *arXiv preprint arXiv:2210.09292*.

Wainwright, M. J. (2019). *List of chapters*, page vii–viii. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Zhang, Z., Chen, Z., and Gu, Q. (2025). Convergence of score-based discrete diffusion models: A discrete-time analysis. In *The Thirteenth International Conference on Learning Representations*.

Zhou, K., Li, Y., Zhao, X., and Wen, J.-R. (2024). Diffusion-nat: Self-prompting discrete diffusion for

non-autoregressive text generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1438–1451.

Zhou, Y. and Liang, Y. (2017). Characterization of gradient dominance and regularity conditions for neural networks. *arXiv preprint arXiv:1710.06910*.

CHECKLIST

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Not Applicable

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes
   (b) Complete proofs of all theoretical results. Yes
   (c) Clear explanations of any assumptions. Yes

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Not Applicable
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Not Applicable
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Not Applicable
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Not Applicable

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Not Applicable
   (b) The license information of the assets, if applicable. Not Applicable
   (c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable
   (d) Information about consent from data providers/curators. Not Applicable
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

Supplementary material

A TRAINING AND SAMPLING ALGORITHM

In this section, we provide a practical training and sampling algorithm for discrete-state diffusion models. The forward process (Step 6 in Algorithm 1) is based on Proposition 1 of Zhang et al. (2025) and the training happens through minimizing the empirical score entropy loss Equation (15). Algorithm 2 is from Zhang et al. (2025).

Algorithm 1 A practical training procedure for Discrete-State Score-based Diffusion Models via Score Entropy Loss

Require: Dataset $\mathcal{D}$ of samples $x_0 \in [S]^d$ (alphabet size S); diffusion time T; step-size h; early-stop $\delta \geq 0$; number of steps K with $T = Kh + \delta$; network $s_\theta(x, t) \in \mathbb{R}^{d\times(S-1)}$; learning rate η; batch size B; epochs E.

1: **Per-token forward kernel :** $P_{0,t} = \frac{1}{S}(1 - e^{-t})\,\mathbf{1}\mathbf{1}^\top + e^{-t} I$.
2: **for** epoch $= 1$ to E **do**
3: **for** $k = 0$ **to** $K - 1$ **do**
4: Sample minibatch $\{x_0^i\}_{i=1}^B \sim \mathcal{D}$.
5: Sample $t \sim \mathrm{Unif}\big([kh + \delta,\, (k + 1)h + \delta]\big)$.
6: **Forward corruption (factorized CTMC):** for each sample i and coord j, $x_t^{i,j} \sim \mathrm{Categorical}\big(P_{0,t}[\,x_0^{i,j}, :]\big)$.
7: **Build ratio targets:** for each (i, j) and each $a \in [S]$, $a \neq x_t^{i,j}$, $r_t^i(j, a) = \dfrac{P_{0,t}[x_0^{i,j}, a]}{P_{0,t}[x_0^{i,j}, x_t^{i,j}]}$.
8: **Query score network frozen at the upper-limit of each interval:** $t' = (k + 1)h + \delta$; compute $\hat{s}^i = s_\theta(x_t^i,\, t')$.
9: **Score Entropy Loss:**
$$L(\theta) = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{S}\sum_{j=1}^{d}\sum_{\substack{a=1\\ a\neq x_t^{i,j}}}^{S}\Big(-\, r_t^i(j, a)\log \hat{s}^i(j, a) + \hat{s}^i(j, a)\Big).$$
10: **Update:** $\theta \leftarrow \theta - \eta\,\nabla_\theta L(\theta)$.
11: **end for**
12: **end for**

Algorithm 2 Generative Reverse Process Simulation through Uniformization

Require: Learned discrete score functions $\{\hat{s}_{T-kh}\}_{k=0}^{K-1}$, total time T, step $h > 0$, and $\delta = T - Kh \geq 0$.

Draw $z_0 \sim \pi^d$.
for $k = 0$ **to** $K-1$ **do**
  Set $\lambda_k \leftarrow \max_{x\in[S]^d}\{\, \hat{s}_{T-kh}(x)_x \,\}$.
  Draw $N \sim \text{Poisson}(\lambda_k h)$.
  Set $y_0 \leftarrow z_k$.
  for $j = 0$ **to** $N-1$ **do**

$$y_{j+1} = \begin{cases} y_j^{\setminus i} \odot \hat{y}^{\,i}, & \text{w.p. } \dfrac{\hat{s}_{T-kh}(y_j)_{i,\hat{y}^{\,i}}}{\lambda_k},\ 1 \leq i \leq d,\ \hat{y}^{\,i} \in [S],\ \hat{y}^{\,i} \neq y_j^i, \\ y_j, & \text{w.p. } 1 - \displaystyle\sum_{i=1}^{d} \sum_{\hat{y}^{\,i} \neq y_j^i} \dfrac{\hat{s}_{T-kh}(y_j)_{i,\hat{y}^{\,i}}}{\lambda_k}. \end{cases}$$

  end for
  $z_{k+1} \leftarrow y_N$.
end for
Output: a sample z_K from $p_{T-\delta}$.

Notation: $\hat{s}_t(x)_x := \sum_{i=1}^d \sum_{\hat{x}^{\,i} \neq x^i} \hat{s}_t(x)_{i,\hat{x}^{\,i}}$, and $x^{\setminus i} \odot \hat{x}^{\,i}$ replaces coordinate i of x with $\hat{x}^{\,i}$.

Discussion on C Since we have access to uniform bounds for the true score functions from Lemma 6, which depend on B, we can apply score clipping during training to ensure that the learned score functions are reliable. Specifically, we enforce the following conditions:

$$\max_{\mathbf{x}\in\mathcal{X},\, k\in\{0,\ldots,K-1\}} \left\|\hat{s}_{(k+1)h}(\mathbf{x})\right\|_\infty \leq \frac{3}{2}B. \tag{38}$$

As a result, we can choose that

$$C = \frac{3}{2}B$$

B SUPPLMENTARY LEMMAS

Definition (Bregman divergence). Let ϕ be a strictly convex function defined on a convex set $\mathcal{S} \subset \mathbb{R}^d$ ($d \in \mathbb{N}_+$) and ϕ is differentiable. The Bregman divergence $D_\phi(x\|y) : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ is defined as

$$D_\phi(x\|y) = \phi(x) - \phi(y) - \nabla\phi(y)^\top (x - y).$$

In particular, the generalized I-divergence

$$D_I(x\|y) = \sum_{i=1}^{d} \left[-x^i + y^i + x^i \log \frac{x^i}{y^i} \right]$$

is generated by the negative entropy function $I(x) = \sum_{i=1}^d x^i \log x^i$.

The Bregman divergence does not satisfy the triangle inequality. However, when the domain of the negative entropy is restricted to a closed box contained in $\mathbb{R}_+^d$, we have the following Lemma from Proposition 3 of Zhang et al. (2025), which provides an analogous form of the triangle inequality.

Lemma 4. *Let the negative entropy function $I(x) = \sum_{i=1}^d x^i \log x^i$ be defined on $[a,b]^d \subset \mathbb{R}_+^d$ for constants $0 < a < b$. Then for all $x, y, z \in [a,b]^d$, we have:*

(i)

$$D_I(x\|y) \;\leq\; \frac{1}{a}\,\|x - z\|_2^2 \;+\; \frac{2b}{a}\,D_I(z\|y).$$

(ii)

$$D_I(x\|y) \;\le\; \frac{1}{a}\,\|x-z\|_2^2 \;+\; \frac{b}{a^2}\,\|z-y\|_2^2$$

Proof. Let $I(x) = \sum_{i=1}^{d} x^i \log x^i$, and note that its Hessian is given by

$$\nabla^2 I(x) = \operatorname{diag}\left(\frac{1}{x^1}, \ldots, \frac{1}{x^d}\right). \tag{39}$$

Since $x \in [a,b]^d$, we have

$$\frac{1}{b} I_d \preceq \nabla^2 I(x) \preceq \frac{1}{a} I_d. \tag{40}$$

This implies that I is $\frac{1}{b}$-strongly convex and its gradient ∇I is $\frac{1}{a}$-Lipschitz smooth.

By the second-order Taylor expansion of the Bregman divergence, there exists $\theta \in [0,1]$ such that

$$D_I(x\|y) = \frac{1}{2}(x-y)^\top \nabla^2 I(y+\theta(x-y))(x-y). \tag{41}$$

Using $\nabla^2 I(x) \preceq \frac{1}{a} I_d$, we obtain

$$D_I(x\|y) \le \frac{1}{2a}\|x-y\|_2^2. \tag{42}$$

Now apply the triangle inequality to the right-hand side of (42):

$$\|x-y\|_2^2 \le 2\|x-z\|_2^2 + 2\|z-y\|_2^2. \tag{43}$$

Substituting into the previous inequality (42) gives

$$D_I(x\|y) \le \frac{1}{a}\|x-z\|_2^2 + \frac{1}{a}\|z-y\|_2^2. \tag{44}$$

Since I is $\frac{1}{b}$-strongly convex, we have

$$D_I(z\|y) \ge \frac{1}{2b}\|z-y\|_2^2 \quad \Rightarrow \quad \|z-y\|_2^2 \le 2b \cdot D_I(z\|y). \tag{45}$$

Substituting this into (44) yields

$$D_I(x\|y) \le \frac{1}{a}\|x-z\|_2^2 + \frac{2b}{a} \cdot D_I(z\|y). \tag{46}$$

This completes the proof of the first part ((i)) of the Lemma.

Because $\nabla^2 I(x) = \operatorname{diag}(1/x^1, \ldots, 1/x^d) \preceq \frac{1}{a} I_d$ on $[a,b]^d$, the function I is $\frac{1}{a}$-smooth. Hence, for z, y,

$$D_I(z\|y) \;\le\; \frac{1}{2a}\,\|z-y\|_2^2. \tag{47}$$

Substituting this in ((i)), we obtain the following:

$$\begin{aligned} D_I(x\|y) \;&\le\; \frac{1}{a}\,\|x-z\|_2^2 \;+\; \frac{2b}{a}\,D_I(z\|y) \\ &\le\; \frac{1}{a}\,\|x-z\|_2^2 \;+\; \frac{2b}{a}\cdot\frac{1}{2a}\,\|z-y\|_2^2 \\ &\le\; \frac{1}{a}\,\|x-z\|_2^2 \;+\; \frac{b}{a^2}\,\|z-y\|_2^2. \end{aligned} \tag{48}$$

It is generally a common practice to use the bounds $[1/C, C]$ instead of $[a,b]$ from Proposition 3 in Zhang et al. (2025). By substituting $a = 1/C$ and $b = C$ in (48), we obtain the following bound:

$$D_I(x\,\|\,y) \;\le\; C\,\|x-z\|_2^2 \;+\; C^3\,\|z-y\|_2^2. \tag{49}$$

This proves the second part ((ii)) of the Lemma. □

This completes the proof of Lemma 4

Lemma 5. *The KL divergence between the true and approximate path measures of the reverse process, both starting from q_T, is (Zhang et al., 2025)*

$$D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T}) = \sum_{k=0}^{K-1} \int_{kh+\delta}^{(k+1)h+\delta} \mathbb{E}_{\mathbf{x}_t \sim q_t} \sum_{i=1}^{d} \sum_{\hat{x}_t^i \neq x_t^i} Q_t^{\mathrm{tok}}(\hat{x}_t^i, x_t^i)\, D_I\Big(s_t(\mathbf{x}_t)_{i,\hat{x}_t^i} \,\|\, \hat{s}_{(k+1)h+\delta}(\mathbf{x}_t)_{i,\hat{x}_t^i} \Big)\, dt. \tag{50}$$

Lemma 6 (Score bound - follows from Lemma 4 of Zhang et al. (2025))**.** *Suppose Assumption 3 holds. Then for all $t \in [0,T]$, $\mathbf{x} \in \mathcal{X}$, $i \in [d]$ and $\hat{x}^i \neq x^i \in [S]$, we have*

$$\frac{1}{B} \;\leq\; s_t(\mathbf{x})_{i,\hat{x}^i} \;\leq\; B.$$

Proof. The Proof of Lemma 6 follows from the Proof of Lemma 4 of Zhang et al. (2025) Define the kernel function

$$g_{\mathbf{w}}(t) = \frac{1}{S^d} \prod_{i=1}^{d} \Big[1 + e^{-t}\big(-1 + S \cdot \mathbf{1}\{w^i \equiv 0 \pmod S\}\big) \Big], \qquad \mathbf{w} \in \mathbb{Z}^d,\ t \geq 0.$$

From Proposition 1 of Zhang et al. (2025), the transition probability of the forward process can be written as

$$\begin{aligned} q_{t|s}(\mathbf{x}_t \mid \mathbf{x}_s) &= \prod_{i=1}^{d} q_{t|s}^i(x_t^i \mid x_s^i) = \prod_{i=1}^{d} P_{s,t}^0(x_s^i, x_t^i) \\ &= \frac{1}{S^d} \prod_{i=1}^{d} \Big[1 + e^{-(t-s)}\big(-1 + S \cdot \delta\{x_t^i, x_s^i\}\big) \Big], \qquad \forall\, t > s \geq 0. \end{aligned}$$

Hence $q_{t|0}(\mathbf{y} \mid \mathbf{x}) = g_{\mathbf{y}-\mathbf{x}}(t)$ for $t > 0$. Thus, for all $t \in (0,T]$, $\mathbf{x} \in \mathcal{X}$, $i \in [d]$, and $x^i \neq \hat{x}^i \in [S]$,

$$\begin{aligned} s_t(\mathbf{x})_{i,\hat{x}^i} &= \frac{q_t(\mathbf{x}^{\setminus i} \odot \hat{x}^i)}{q_t(\mathbf{x})} = \frac{q_t\big(\mathbf{x} + (\hat{x}^i - x^i)\mathbf{e}_i\big)}{q_t(\mathbf{x})} \\ &= \frac{\sum_{\mathbf{y} \in \mathcal{X}} q_0(\mathbf{y})\, q_{t|0}\big(\mathbf{x} + (\hat{x}^i - x^i)\mathbf{e}_i \mid \mathbf{y}\big)}{q_t(\mathbf{x})} = \frac{\sum_{\mathbf{y} \in \mathcal{X}} q_0(\mathbf{y})\, g_{\mathbf{x}+(\hat{x}^i - x^i)\mathbf{e}_i - \mathbf{y}}(t)}{q_t(\mathbf{x})} \\ &= \frac{\sum_{\mathbf{y} \in \mathcal{X}} q_0\big(\mathbf{y} + (\hat{x}^i - x^i)\mathbf{e}_i\big)\, g_{\mathbf{x}-\mathbf{y}}(t)}{q_t(\mathbf{x})} = \sum_{\mathbf{y} \in \mathcal{X}} \frac{q_0(\mathbf{y})\, q_{t|0}(\mathbf{x} \mid \mathbf{y})}{q_t(\mathbf{x})}\, \frac{q_0\big(\mathbf{y} + (\hat{x}^i - x^i)\mathbf{e}_i\big)}{q_0(\mathbf{y})} \\ &= \frac{1}{B} \;\leq\; \mathbb{E}_{\mathbf{y} \sim q_{0|t}(\cdot|\mathbf{x})} \left[\frac{q_0\big(\mathbf{y} + (\hat{x}^i - x^i)\mathbf{e}_i\big)}{q_0(\mathbf{y})} \right] \;\leq\; B, \end{aligned}$$

where the last inequality uses Assumption 3. The sum $\mathbf{y} + (\hat{x}^i - x^i)\mathbf{e}_i$ is interpreted in modulo-S sense. □

□

Lemma 7 (Score movement bound - Lemma 5 of Zhang et al. (2025))**.** *Suppose Assumption 3 holds. Let $\delta = 0$. Then for all $k \in \{0, 1, \ldots, K-1\}$, $t \in [kh, (k+1)h]$, $\mathbf{x} \in \mathcal{X}$, $i \in [d]$, and $\hat{x}^i \neq x^i \in [S]$, we have*

$$\big| s_t(\mathbf{x})_{i,\hat{x}^i} - s_{(k+1)h}(\mathbf{x})_{i,\hat{x}^i} \big| \;\lesssim\; \left[\frac{1}{1 - e^{-(k+1)h}} + S \right] \kappa_i h.$$

Lemma 8 (Theorem 26.5 of Shalev-Shwartz and Ben-David (2014))**.** *Assume that for all z and $h \in \mathcal{H}$ we have that $|\ell(h,z)| \leq c$. Then,*

1. *With probability of at least $1 - \delta$, for all $h \in \mathcal{H}$,*

$$L_D(h) - L_S(h) \leq 2 \mathop{\mathbb{E}}_{S' \sim D^m} R(\ell \circ \mathcal{H} \circ S') + c\sqrt{\frac{2\ln(2/\delta)}{m}}. \tag{51}$$

In particular, this holds for $h = ERM_{\mathcal{H}}(S)$.

2. *With probability of at least* $1-\delta$*, for all* $h \in \mathcal{H}$*,*

$$L_D(h) - L_S(h) \leq 2R(\ell \circ \mathcal{H} \circ S) + 4c\sqrt{\frac{2\ln(4/\delta)}{m}}. \tag{52}$$

In particular, this holds for $h = ERM_{\mathcal{H}}(S)$*.*

3. *For any* h^**, with probability of at least* $1-\delta$*,*

$$L_D(ERM_{\mathcal{H}}(S)) - L_D(h^*) \leq 2R(\ell \circ \mathcal{H} \circ S) + 5c\sqrt{\frac{2\ln(8/\delta)}{m}}. \tag{53}$$

Lemma 9 (Extension of Massart's Lemma)**.** *Let* Θ'' *be a finite function class. Then, for any* $\theta \in \Theta''$*, we have*

$$\mathbb{E}_\sigma\left[\max_{\theta\in\Theta''}\sum_{i=1}^{n} f(\theta)\sigma_i\right] \leq ||f(\theta)||_2 \leq (BW)^L\left(d + \frac{L}{W}\right) \tag{54}$$

where σ_i *are i.i.d random variables such that* $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}$*. We get the second inequality by denoting* L *as the number of layers in the neural network,* W *and* B *a constant such all parameters of the neural network upper bounded by* B*.* d *is the dimension of the function* $f(\theta)$*.*

Proof. Let $h_0 = x$, and for $\ell = 0, \ldots, L-1$ define the layer recursion

$$h_{\ell+1} = \sigma(W_\ell h_\ell + b_\ell),$$

where $W_\ell \in \mathbb{R}^{n_{\ell+1}\times n_\ell}$, $b_\ell \in \mathbb{R}^{n_{\ell+1}}$, and $n_\ell \leq W$ for hidden layers. We work with the ℓ_∞ operator norm:

$$\|W_\ell\|_\infty = \max_i \sum_j |(W_\ell)_{ij}| \;\leq\; B\, n_\ell \;\leq\; BW = \alpha.$$

Since σ is 1-Lipschitz with $\sigma(0) = 0$, we have $\|\sigma(u)\|_\infty \leq \|u\|_\infty$ and thus

$$\|h_{\ell+1}\|_\infty \;\leq\; \|W_\ell\|_\infty\, \|h_\ell\|_\infty + \|b_\ell\|_\infty \;\leq\; \alpha\, \|h_\ell\|_\infty + B.$$

With $\|h_0\|_\infty \leq d$, iterating this affine recursion yields the standard geometric-series bound

$$\|h_L\|_\infty \;\leq\; \alpha^L d \;+\; B\sum_{i=0}^{L-1}\alpha^{\,i} \;=\; \alpha^L d \;+\; B\,\frac{\alpha^L - 1}{\alpha - 1} \quad (\alpha \neq 1),$$

and for $\alpha = 1$, $\|h_L\|_\infty \leq d + BL$. The scalar output $f(x)$ is either a coordinate of h_L or obtained by applying the same 1-Lipschitz activation to a linear form of h_L; in either case, $|f(x)| \leq \|h_L\|_\infty$, giving the stated bound.

For the $\alpha \geq 1$ simplification, use $\sum_{i=0}^{L-1}\alpha^i \leq L\alpha^{L-1}$ to obtain

$$|f(x)| \;\leq\; \alpha^L d + BL\alpha^{L-1} \;=\; (BW)^L\left(d + \frac{L}{W}\right).$$

For $\alpha < 1$, since $\alpha^i \leq 1$, $\sum_{i=0}^{L-1}\alpha^i \leq L$ and hence $|f(x)| \leq d + BL$. Note that this is a tighter bound that the one for $\alpha \geq 1$. However, we retain the bound for $\alpha \geq 1$ to make our result hold for both cases. □

Lemma 10 (Theorem 4.6 of Gower et al. (2021))**.** *Let* f *be* L*-smooth. Assume* $f \in PL(\mu)$ *and* $g \in ER(\rho)$*. Let* $\gamma_k = \gamma \leq \frac{1}{1+2\rho/\mu}\cdot\frac{1}{L}$*, for all* k*, then SGD update given by* $x^{k+1} = x^k - \gamma^k g(x^k)$ *converges as follows*

$$\mathbb{E}[f(x^k) - f^*] \leq (1-\gamma\mu)^k[f(x^0) - f^*] + \frac{L\gamma\sigma^2}{\mu}. \tag{55}$$

Hence, given $\varepsilon > 0$ *and using the step size* $\gamma = \frac{1}{L}\min\left\{\frac{\mu\varepsilon}{2\sigma^2}, \frac{1}{1+2\rho/\mu}\right\}$ *we have that*

$$k \geq \frac{L}{\mu}\max\left\{\frac{2\sigma^2}{\mu\varepsilon}, 1 + \frac{2\rho}{\mu}\right\}\log\left(\frac{2(f(x^0) - f^*)}{\varepsilon}\right) \quad\Longrightarrow\quad \mathbb{E}[f(x^k) - f^*] \leq \varepsilon. \tag{56}$$

Lemma 11 (Discrete–continuous score-error: absolute $O(h)$ gap)**.** *Let $(X_t)_{t\in[\delta,T]}$ be a continuous-time Markov chain on a finite state space Ω with generator $Q = (q(x,y))_{x,y\in\Omega}$, and write q_t for the law of X_t. Fix a stepsize $h > 0$ and the grid $t_k := \delta + kh$ for $k = 0,\dots,K-1$ with intervals $I_k = [t_k, t_{k+1}]$ and $Kh = T - \delta$. For each k, define*

$$f_k(x) \;:=\; D\big(s_{t_k}(x) \,\big\|\, \hat{s}_{t_k}(x)\big)\,,$$

and assume a finite uniform departure rate

$$\lambda \;:=\; \sup_{x\in\Omega} \sum_{y\neq x} q(x,y) \;<\; \infty. \tag{57}$$

Let $S \geq 1$ denote the (harmless) scaling constant. Define the continuous- and discrete-time score-error terms

$$\mathrm{CT} := \frac{1}{S}\sum_{k=0}^{K-1} \int_{I_k} \mathbb{E}_{q_t}\big[f_k(X)\big]\,dt, \qquad \mathrm{DT} := \frac{1}{S}\sum_{k=0}^{K-1} h\,\mathbb{E}_{q_{t_k}}\big[f_k(X)\big]$$

Then the absolute difference satisfies

$$\big|\mathrm{CT} - \mathrm{DT}\big| \;\leq\; \frac{(S-1)dC\,\lambda\,T\,h}{S} \tag{58}$$

Proof. Fix k and set $g_k(t) := \mathbb{E}_{q_t}[f_k(X)]$ for $t \in I_k$. By the Kolmogorov forward equation ($\dot{q}_t = Q^\top q_t$),

$$\frac{d}{dt}\,g_k(t) = \sum_x (Q^\top q_t)(x) f_k(x) = \sum_x q_t(x)\,(Qf_k)(x) = \mathbb{E}_{q_t}\big[(Qf_k)(X)\big],$$

where $(Qf_k)(x) = \sum_{y\neq x} q(x,y)\big(f_k(y) - f_k(x)\big)$. Using (6) and the triangle inequality,

$$\big|(Qf_k)(x)\big| \;\leq\; \sum_{y\neq x} q(x,y)\,\big|f_k(y) - f_k(x)\big| \;\leq\; \sum_{y\neq x} q(x,y)\cdot 2(S-1)dC \;\leq\; 2(S-1)dC\lambda.$$

Hence $\sup_{t\in I_k}\big|g_k'(t)\big| \leq 2(S-1)dC\lambda$. From, Lemma 12 we have

$$\left|\int_{t_k}^{t_{k+1}} g_k(t)\,dt - h\,g_k(t_k)\right| \;\leq\; \frac{h^2}{2}\sup_{u\in I_k}\big|g_k'(u)\big| \;\leq\; 2(S-1)dC\lambda h^2$$

Summing over $k = 0,\dots,K-1$ gives

$$\left|\sum_{k=0}^{K-1}\int_{I_k}\mathbb{E}_{q_t}[f_k]\,dt - \sum_{k=0}^{K-1} h\,\mathbb{E}_{q_{t_k}}[f_k]\right| \;\leq\; \sum_{k=0}^{K-1}(S-1)dC\lambda h^2 \;=\; (S-1)dC\lambda K h^2 \;\leq\; (S-1)dC\lambda T h,$$

since $Kh = T - \delta \leq T$. Dividing both sides by S yields (58). □

If $\|s_t\|_\infty, \|\hat{s}_t\|_\infty \leq C$ on $[\delta, T]$ and D is a Bregman divergence whose generator is smooth on the corresponding bounded set, then $M \leq c\,C^2$ for a constant c. For single–coordinate refresh dynamics in d dimensions with uniform symbol choice, $\lambda \leq d\,(S-1)/S$. Consequently, (58) gives $\big|\mathrm{CT} - \mathrm{DT}\big| \leq (c\,C_1^2)\,d\,T\,h/S$.

Lemma 12 (First-order quadrature remainder (left rectangle rule))**.** *Let $g : [a,b] \to \mathbb{R}$ be absolutely continuous with $g' \in L^\infty([a,b])$. Then*

$$\int_a^b g(t)\,dt \;-\; (b-a)\,g(a) \;=\; \int_a^b (b-u)\,g'(u)\,du, \tag{59}$$

and consequently

$$\left|\int_a^b g(t)\,dt \;-\; (b-a)\,g(a)\right| \;\leq\; \frac{(b-a)^2}{2}\,\|g'\|_{\infty,[a,b]}. \tag{60}$$

Moreover, the constant $1/2$ is sharp (equality holds for affine $g(t) = \alpha + \beta t$).

Proof. By absolute continuity and the fundamental theorem of calculus, $g(t) = g(a) + \int_a^t g'(u)\,du$ for all $t \in [a, b]$. Integrating both sides in t gives

$$\int_a^b g(t)\,dt = (b-a)\,g(a) + \int_a^b \left(\int_a^t g'(u)\,du \right) dt.$$

By Tonelli/Fubini (justified since $g' \in L^\infty$),

$$\int_a^b \left(\int_a^t g'(u)\,du \right) dt = \int_a^b \left(\int_u^b dt \right) g'(u)\,du = \int_a^b (b-u)\,g'(u)\,du,$$

which proves Equation (59). Taking absolute values and using $0 \le b-u \le b-a$,

$$\left| \int_a^b (b-u)\,g'(u)\,du \right| \le \|g'\|_{\infty,[a,b]} \int_a^b (b-u)\,du = \|g'\|_{\infty,[a,b]} \frac{(b-a)^2}{2},$$

which is (60). Sharpness follows by direct calculation for $g(t) = \alpha + \beta t$, where the error equals $\beta(b-a)^2/2$. □

[Right rectangle, midpoint, and nonuniform steps] (1) *Right rectangle.* An identical argument with $g(b)$ in place of $g(a)$ yields

$$\int_a^b g(t)\,dt - (b-a)\,g(b) = -\int_a^b (u-a)\,g'(u)\,du, \qquad \left| \int_a^b g(t)\,dt - (b-a)\,g(b) \right| \le \frac{(b-a)^2}{2}\,\|g'\|_\infty.$$

(2) *Midpoint rule (needs a second derivative).* If g' is absolutely continuous and $g'' \in L^\infty([a,b])$, then

$$\left| \int_a^b g(t)\,dt - (b-a)\,g\big(\tfrac{a+b}{2}\big) \right| \le \frac{(b-a)^3}{24}\,\|g''\|_\infty.$$

We do not use this in the theorem since we only assume a bound on g'.

(3) *Nonuniform mesh.* For intervals $I_k = [t_k, t_{k+1}]$ with $h_k = t_{k+1} - t_k$ and g_k absolutely continuous on I_k,

$$\left| \int_{I_k} g_k(t)\,dt - h_k\,g_k(t_k) \right| \le \frac{h_k^2}{2}\,\|g_k'\|_{\infty,I_k}, \qquad \sum_k \left| \int_{I_k} g_k - h_k g_k(t_k) \right| \le \frac{1}{2} \sum_k h_k^2 \|g_k'\|_{\infty,I_k}.$$

If $h_{\max} := \max_k h_k$, this implies $\sum_k \left| \int_{I_k} g_k - h_k g_k(t_k) \right| \le \frac{h_{\max}}{2} \sum_k h_k \|g_k'\|_{\infty,I_k}$.

How this plugs into the theorem. In the proof of Theorem 11, set $a = t_k$, $b = t_{k+1}$ and $g_k(t) := \mathbb{E}_{q_t}[f_k(X)]$. From the CTMC calculus we had $g_k'(t) = \mathbb{E}_{q_t}[(Qf_k)(X)]$ and $\|g_k'\|_{\infty,I_k} \le 2(S-1)dC\lambda$. Applying Lemma 12 gives the per-interval bound

$$\left| \int_{t_k}^{t_{k+1}} g_k(t)\,dt - h\,g_k(t_k) \right| \le \frac{h^2}{2} \cdot 2(S-1)dC\lambda = M\lambda h^2.$$

Summing over k and using $Kh = T - \delta \le T$ yields $\big|\mathrm{CT} - \mathrm{DT}\big| \le (2(S-1)dC\lambda/S)\,Kh^2 \le (2(S-1)dC\lambda/S)\,Th$, which is exactly the absolute $O(h)$ gap.

C INTERMEDIATE LEMMAS

In this section, we present the proofs of intermediate lemmas used to bound the approximation error, statistical error, and optimization error in our analysis.

C.1 Bounding the Approximation error

Recall Lemma 1 (Approximation Error.) *Let W and D denote the width and depth of the neural network architecture, respectively, and let Sd represent the input data dimension. Then, for all $k \in [0, K-1]$ we have*

$$\mathcal{E}_k^{\text{approx}} = 0 \tag{61}$$

Proof. We construct a network explicitly such that it matches any given function defined on a fixed set of points. The proof has two parts: a base construction of depth 2 (one hidden layer) and an inductive padding by identity layers that preserves the values on X while increasing depth without increasing width.

We give a constructive proof. The idea is (i) build a depth-2 interpolant using only S of the W hidden units and zero out the remaining $W - S$, then (ii) increase depth by inserting identity blocks that act as the identity on the finite set of hidden codes, still using width W.

Step 0 (Distinct scalar coordinates). Since $x_1, \dots, x_S$ are distinct, choose $a \in \mathbb{R}^d$ so that $t_i := a^\top x_i$ are pairwise distinct. Relabel so $t_1 < t_2 < \cdots < t_S$.

Step 1 (Depth-2 interpolant of width $W \geq S$). Pick biases $b_1, \dots, b_S$ such that

$$b_1 < t_1, \qquad b_k \in \left(\tfrac{t_{k-1}+t_k}{2},\, t_k\right) \text{ for } k = 2, \dots, S.$$

For $\alpha > 0$ define the $S \times S$ matrix $A^{(\alpha)}$ with entries

$$A_{ik}^{(\alpha)} := \phi\big(\alpha(t_i - b_k)\big), \qquad 1 \leq i, k \leq S.$$

As in the triangular-limit argument, the rescaled matrix

$$\widetilde{A}^{(\alpha)} := \frac{A^{(\alpha)} - \phi_- \mathbf{1}\mathbf{1}^\top}{\phi_+ - \phi_-}$$

converges entrywise to the unit lower-triangular matrix L as $\alpha \to \infty$, hence is invertible for some finite $\alpha^\star > 0$. Let $y = (f(x_1), \dots, f(x_S))^\top$ and choose any $c_0 \in \mathbb{R}$. Solve for $c = (c_1, \dots, c_S)^\top$ in

$$A^{(\alpha^\star)} c = y - c_0 \mathbf{1}.$$

Now define a one-hidden-layer network with width W by using the first S hidden units exactly as above and *setting the remaining $W - S$ hidden units to arbitrary parameters but with output weights equal to zero.* Concretely,

$$N_2(x) := c_0 + \sum_{k=1}^{S} c_k\, \phi\big(\alpha^\star(a^\top x - b_k)\big) + \sum_{k=S+1}^{W} 0 \cdot \phi(\,\cdot\,).$$

Then $N_2(x_i) = f(x_i)$ for all i. Thus we have a depth-2 interpolant of width W.

Step 2 (Identity-on-a-finite-set block with width $W \geq S$). We show that for any finite set $U = \{u_1, \dots, u_S\} \subset \mathbb{R}^m$ there exists a one-hidden-layer map

$$G(u) = C\, \phi\big(\alpha(Bu + d)\big) + e, \qquad \text{(hidden width } W \geq S\text{)},$$

such that $G(u_i) = u_i$ for all i.

Construction. Choose S rows of B and corresponding entries of d so that the $S \times S$ matrix

$$M_{ik}^{(\alpha)} := \phi\big(\alpha(b_k^\top u_i + d_k)\big), \qquad 1 \leq i, k \leq S,$$

has the same lower-triangular limit as in Step 1; hence $M^{(\alpha^\star)}$ is invertible for some finite $\alpha^\star$. Let $U := [u_1 \cdots u_S] \in \mathbb{R}^{m \times S}$ and define $C \in \mathbb{R}^{m \times W}$ by

$$C = \big[\, U\, (M^{(\alpha^\star)})^{-1} \quad 0_{m \times (W-S)} \,\big],$$

i.e., the last $W - S$ columns are zeros. Set the last $W - S$ rows of B and entries of d arbitrarily, and take $e = 0$. Then for each i,

$$G(u_i) = C\,\phi\big(\alpha^\star(Bu_i + d)\big) = U\,(M^{(\alpha^\star)})^{-1}\,M_{\cdot i}^{(\alpha^\star)} = u_i.$$

Hence G acts as the identity on U while using width $W \geq S$ (extra units are harmlessly zeroed out).

Step 3 (Depth lifting to any $L \geq 2$ with width W). Let $h_i \in \mathbb{R}^W$ denote the hidden code of x_i produced by the hidden layer of N_2 (the last $W - S$ coordinates may be arbitrary but are fixed). Apply Step 2 to $U = \{h_1, \ldots, h_S\} \subset \mathbb{R}^W$ to obtain a width-W block G with $G(h_i) = h_i$ for all i. Insert G between the hidden layer and the output of N_2; this does not change the outputs on X. Repeating this insertion yields a depth-L network of width W that still satisfies $N(x_i) = f(x_i)$ for all i.

Combining the steps, for any $L \geq 2$ and any $W \geq S$ there exists a depth-L, width-W network with activation ϕ that interpolates f exactly on X.

1) The construction plainly subsumes the case $W = S$ by taking the last $W - S$ (i.e., zero) columns to be vacuous. 2) For vector-valued $f : X \to \mathbb{R}^m$, share the same hidden layers and use an m-dimensional linear readout, or solve the linear systems coordinate-wise; the width requirement remains $W \geq S$. □

C.2 Bounding the Statistical Error

Next, we bound the statistical error $\mathcal{E}_{\text{stat}}$. This error is due to estimating $s^*_{\theta,k}(x_k)$ from a finite number of samples rather than the full data distribution.

Recall Lemma 2 (Statistical Error.) *Let n_k denote the number of samples used to estimate the score function at each diffusion step. Then, for all $k \in [0, K-1]$, with probability at least $1 - \gamma$, we have*

$$\mathcal{E}_k^{\text{stat}} \leq \mathcal{O}\left((S-1)d\sqrt{\frac{\log\left(\frac{2}{\gamma}\right)}{n_k}}\right) \tag{62}$$

Proof. Let us define the population loss at time k for $k \in [0, K-1]$ as follows:

$$\mathcal{L}_k(\theta) = \int_{kh+\delta}^{(k+1)h+\delta} \mathbb{E}_{\mathbf{x}_k \sim q_t}\left[D_I\left(s_k(\mathbf{x}_t) \,\|\, \overline{s}_{\theta,k}(\mathbf{x}_t)\right)\right] dt \tag{63}$$

The corresponding empirical loss is:

$$\widehat{\mathcal{L}}_k(\theta) = \frac{1}{n_k}\sum_{i=1}^{n_k} D_I\left(s_k(\mathbf{x}_{k,t,i}) \,\|\, \overline{s}_{\theta,k}(\mathbf{x}_{k,t,i})\right), \quad t \sim \text{Unif}[kh, (k+1)h]$$

Let θ_k^a and θ_k^b be the minimizers of $\mathcal{L}_k(\theta)$ and $\widehat{\mathcal{L}}_k(\theta)$, respectively, corresponding to score functions $s^a_{\theta,k}$ and $s^b_{\theta,k}$. By the definitions of minimizers, we can write

$$\mathcal{L}_k(\theta_k^b) - \mathcal{L}_k(\theta_k^a) \leq \mathcal{L}_k(\theta_k^b) - \mathcal{L}_k(\theta_k^a) + \widehat{\mathcal{L}}_k(\theta_k^a) - \widehat{\mathcal{L}}_k(\theta_k^b) \tag{64}$$

$$\leq \underbrace{\left|\mathcal{L}_k(\theta_k^b) - \widehat{\mathcal{L}}_k(\theta_k^b)\right|}_{\text{(I)}} + \underbrace{\left|\mathcal{L}_k(\theta_k^a) - \widehat{\mathcal{L}}_k(\theta_k^a)\right|}_{\text{(II)}}. \tag{65}$$

Note that right hand side of (64) is less then the left hand side since we have added the quantity $\widehat{\mathcal{L}}_k(\theta_k^a) - \widehat{\mathcal{L}}_k(\theta_k^b)$ which is strictly positive since θ_k^b is the minimizer of the function $\widehat{\mathcal{L}}_k(\theta)$ by definition. We then take the absolute value on both sides of (65) to get

$$|\mathcal{L}_k(\theta_k^b) - \mathcal{L}_k(\theta_k^a)| \leq \underbrace{\left|\mathcal{L}_k(\theta_k^b) - \widehat{\mathcal{L}}_t(\theta_k^b)\right|}_{\text{(I)}} + \underbrace{\left|\mathcal{L}_k(\theta_k^a) - \widehat{\mathcal{L}}_k(\theta_k^a)\right|}_{\text{(II)}}. \tag{66}$$

Note that this preserves the direction of the inequality since the left-hand side of (65) is strictly positive since θ_k^a is the minimizer of $\mathcal{L}_k(\theta)$ by definition.

We now bound terms (I) and (II) using standard generalization results. From lemma 8, if the loss function is uniformly bounded over the parameter space $\Theta'' = \{\theta_k^a, \theta_k^b\}$, then with probability at least $1-\gamma$, we have

$$\mathbb{E}\left[\left|\mathcal{L}_k(\theta) - \widehat{\mathcal{L}}_k(\theta)\right|\right] \leq \widehat{R}(\theta) + \mathcal{O}\left(\sqrt{\frac{\log\frac{1}{\gamma}}{n_k}}\right), \quad \forall\, \theta \in \Theta'' \tag{67}$$

where $\widehat{R}(\theta)$ denotes the empirical Rademacher complexity of the function class restricted to Θ''. Since Θ'' is a finite class (just two functions) and the inputs are finite, the loss function $\mathcal{L}_k(\theta)$ is bounded, thus, We have from 8

$$\mathbb{E}\left[\left|\mathcal{L}_k(\theta) - \widehat{\mathcal{L}}_k(\theta)\right|\right] \leq \widehat{R}(\theta) + \mathcal{O}\left(\sqrt{\frac{\log\frac{1}{\gamma}}{n_k}}\right), \qquad \forall\, \theta \in \Theta''. \tag{68}$$

From Lemma 9 we have

$$\mathbb{E}_\sigma \max_{\theta\in\Theta''} \sum_{i=1}^{n_k} f(\theta)\sigma_i \leq ||f(\theta)||_2 \leq (BW)^L\left((S-1)d + \frac{L}{W}\right) \tag{69}$$

Now, the definition of $\hat{R}(\theta)$ is $\hat{R}(\theta) = \frac{1}{m}\mathbb{E}_\sigma \max_{\theta\in\Theta''}\sum_{i=1}^n f(\theta)\sigma_i$. Thus we get:

$$\mathbb{E}\left[\left|\mathcal{L}_k(\theta) - \widehat{\mathcal{L}}_k(\theta)\right|\right] \leq \mathcal{O}\left(\frac{(W)^L\left((S-1)d + \frac{L}{W}\right)}{n_k}\right) + \mathcal{O}\left(\sqrt{\frac{\log\frac{1}{\gamma}}{n_k}}\right), \qquad \forall\, \theta \in \Theta''. \tag{70}$$

Here $||\Theta||$ is a value such that $||\Theta|| = \sup_{\theta\in\Theta''}||\theta||_\infty$. Which implies that Θ is the norm of the maximum parameter value that can be attained in the finite function class $\Theta^{''}$, which yields:

$$\mathbb{E}\left[\left|\mathcal{L}_k(\theta) - \widehat{\mathcal{L}}_k(\theta)\right|\right] \leq \mathcal{O}\left((W)^L\left((S-1)d + \frac{L}{W}\right)\sqrt{\frac{\log\frac{1}{\gamma}}{n_k}}\right), \qquad \forall\, \theta \in \Theta'' \tag{71}$$

Finally, using the Polyak-Łojasiewicz (PL) condition for $\mathcal{L}_k(\theta)$, from Assumption 1, we have from the quadratic growth condition of PL functions the following,

$$\|\theta_k^a - \theta_k^b\|^2 \leq \mu\left|\mathcal{L}_k(\theta_k^a) - \mathcal{L}_k(\theta_k^b)\right|, \tag{72}$$

Define the constant L' as $L' = \sup_{L_k'}\|s_{\theta,k}^a(x_k) - s_{\theta,k}^b(x_k)\|^2 \leq L_k'.\|\theta_k^a - \theta_k^b\|^2$. Then we have

$$\begin{aligned}
\|s_{\theta,k}^a(x_k) - s_{\theta,k}^b(x_k)\|^2 &\leq L'\|\theta_k^a - \theta_k^b\|^2 && (73)\\
&\leq L'\mu\left|\mathcal{L}_k(\theta_k^a) - \mathcal{L}_k(\theta_k^b)\right| && (74)\\
&\leq \mathcal{O}\left((W)^L\left((S-1)d + \frac{L}{W}\right)\sqrt{\frac{\log\frac{1}{\gamma}}{n_k}}\right) && (75)
\end{aligned}$$

Taking expectation with respect to $x \sim q_k$ on both sides completes the proof. □

C.3 Bounding the Optimization Error

The optimization error ($\mathcal{E}_{\text{opt}}$) accounts for the fact that gradient-based optimization does not necessarily find the optimal parameters due to limited steps, local minima, or suboptimal learning rates. This can be bounded as follows.

Recall Lemma 3 (Optimization Error) Let n_k be the number of samples used to estimate the score function at diffusion step k. If the learning rate $0 \le \eta \le \frac{1}{\kappa}$, then under Assumptions 1 and 2, for all $k \in [0, K-1]$, the optimization error due to imperfect minimization of the training loss satisfies with probability at least $1-\gamma$

$$\mathcal{E}_k^{\text{opt}} \le \mathcal{O}\left((S-1)d\sqrt{\frac{\log\left(\frac{2}{\gamma}\right)}{n_k}} \right). \tag{76}$$

Proof. Let $\mathcal{E}_k^{\text{opt}}$ denote the optimization error incurred when performing stochastic gradient descent (SGD), with the empirical loss defined by:

$$\widehat{\mathcal{L}}_k^i(\theta) = D_I\left(s_k(\mathbf{x}_{k,t,i}) \,\|\, \overline{s}_{\theta,k}(\mathbf{x}_{k,i})\right) \quad t \sim \text{Unif}[kh, (k+1)h] \tag{77}$$

The corresponding population loss is:

$$\mathcal{L}_k(\theta) = \int_{kh+\delta}^{(k+1)h+\delta} \mathbb{E}_{\mathbf{x}_k \sim q_t}\left[D_I\left(s_k(\mathbf{x}_t) \,\|\, \overline{s}_{\theta,k}(\mathbf{x}_t)\right)\right] dt \tag{78}$$

Thus, $\mathcal{E}_k^{\text{opt}}$ captures the error incurred during the stochastic optimization at each fixed time step k. We now derive upper bounds on this error.

From the smoothness of $\mathcal{L}_k(\theta)$ through Assumption 2, we have

$$\mathcal{L}_k(\theta_{i+1}) \le \mathcal{L}_k(\theta_i) + \langle \nabla \mathcal{L}_k(\theta_i), \theta_{i+1} - \theta_i \rangle + \frac{\kappa}{2}\|\theta_{i+1} - \theta_i\|^2. \tag{79}$$

Taking conditional expectation given θ_i, and using the unbiased-ness of the stochastic gradient $\nabla \widehat{\mathcal{L}}_k(\theta_i)$, we get:

$$\begin{aligned} \mathbb{E}[\mathcal{L}_k(\theta_{i+1}) \mid \theta_i] \le \mathcal{L}_k(\theta_i) &- \alpha_t \left\|\nabla \mathcal{L}_k(\theta_i)\right\|^2 \\ &+ \frac{\kappa \alpha_t^2}{2} \mathbb{E}\left[\left\|\nabla \widehat{\mathcal{L}}_k(\theta_i)\right\|^2 \mid \theta_i\right] \end{aligned} \tag{80}$$

Now using the variance bound on the stochastic gradients using Assumption 2, we have

$$\mathbb{E}[\|\nabla \widehat{\mathcal{L}}_k(\theta_i)\|^2 \mid \theta_i] \le \|\nabla \mathcal{L}_k(\theta_i)\|^2 + \sigma^2, \tag{81}$$

Using this in the previous equation, we have that

$$\begin{aligned} \mathbb{E}[\mathcal{L}_k(\theta_{t+1}) \mid \theta_i] &\le \mathcal{L}_k(\theta_i) - \eta \left\|\nabla \mathcal{L}(\theta_i)\right\|^2 + \frac{\kappa \eta^2}{2}\left(\left\|\nabla \mathcal{L}(\theta_i)\right\|^2 + \sigma^2\right) \\ &= \mathcal{L}(\theta_i) - \left(\eta - \frac{\kappa \eta^2}{2}\right)\left\|\nabla \mathcal{L}(\theta_i)\right\|^2 + \frac{\kappa \eta^2 \sigma^2}{2} \end{aligned} \tag{82}$$

Now applying the PL inequality (Assumption 1), $\|\nabla L(\theta_i)\|^2 \ge 2\mu\left(L(\theta_i) - L^*\right)$, we substitute in the above inequality to get

$$\mathbb{E}[\mathcal{L}(\theta_{i+1}) \mid \theta_i] - \mathcal{L}^* \le \left(1 - 2\mu\left(\eta - \frac{\kappa \eta^2}{2}\right)\right)\left(\mathcal{L}(\theta_i) - \mathcal{L}^*\right) + \frac{\kappa \eta^2 \sigma^2}{2}. \tag{83}$$

Define the contraction factor

$$\rho := 1 - 2\mu\left(\eta - \frac{\kappa \eta^2}{2}\right) \tag{84}$$

Taking total expectation and defining $\delta_k = \mathbb{E}[L(\theta_i) - L^*]$, we get the recursion:

$$\delta_{k+1} \leq \rho \cdot \delta_k + \frac{\kappa\eta^2\sigma^2}{2}. \tag{85}$$

When $\eta \leq \frac{1}{\kappa}$, we have

$$\eta - \frac{\kappa\eta^2}{2} \geq \frac{\eta}{2} \Rightarrow \rho \leq 1 - \mu\eta. \tag{86}$$

Unrolling the recursion we have

$$\delta_k \leq (1-\mu\eta)^k \delta_0 + \frac{\kappa\eta^2\sigma^2}{2} \sum_{j=0}^{t-1} (1-\mu\eta)^j. \tag{87}$$

Using the geometric series bound:

$$\sum_{j=0}^{k-1} (1-\mu\eta)^j \leq \frac{1}{\mu\eta}, \tag{88}$$

we conclude that

$$\delta_k \leq (1-\mu\eta)^k \delta_0 + \frac{\kappa\eta\sigma^2}{2\mu}. \tag{89}$$

Hence, we have the convergence result

$$\mathbb{E}[\mathcal{L}(\theta_i) - \mathcal{L}^*] \leq (1-\mu\eta)^k \delta_0 + \frac{\kappa\eta\sigma^2}{2\mu}. \tag{90}$$

From theorem 4.6 of Gower et al. (2021) states that this result implies a sample complexity of $\mathcal{O}\left(\frac{1}{n}\right)$ where n is the number of steps of the SGD algorithm performed. Note that the result from Gower et al. (2021) requires bounded gradient and smoothness assumption which are satisfied by Assumption 2.

Note that $\overline{s}_k$ and $\overline{\theta}_k$ denote our estimate of the score function and associated parameter obtained from the SGD. Also note that $\mathcal{L}^*$ is the loss function corresponding to whose minimizer is the neural network s_k^a and the neural parameter θ_k^a is our estimated score parameter. Thus, applying the quadratic growth inequality.

$$\left\|\overline{s}_k(\mathbf{x}_k) - s_k^a(\mathbf{x}_k)\right\|^2 \leq L \left\|\overline{\theta}_k - \theta_k^a\right\|^2 \leq \left|\mathcal{L}_k(\theta_k) - \mathcal{L}_k^*\right| \tag{91}$$

$$\leq \mathcal{O}\left(\frac{1}{n}\right) \tag{92}$$

From Lemma 2, we have with probability $1-\gamma$ that

$$\left\|s_k^a(\mathbf{x}_k) - s_k^b(\mathbf{x}_k)\right\|^2 \leq L \left\|\theta_k^a - \theta_k^b\right\|^2 \tag{93}$$

$$\leq L\mu \left|\mathcal{L}_k(\theta_k^a) - \mathcal{L}_k(\theta_k^b)\right| \tag{94}$$

$$\leq \mathcal{O}\left((W)^L \left((S-1)d + \frac{L}{W}\right) \sqrt{\frac{\log\frac{2}{\gamma}}{n_k}}\right) \tag{95}$$

Thus, we have with probability at least $1-\gamma$,

$$\begin{aligned}\left\|\overline{s}_{\theta,k}(\mathbf{x}_k) - s_{\theta,k}^b(\mathbf{x}_k)\right\|^2 &\leq 2\left\|\overline{s}_{\theta,k}(\mathbf{x}_k) - s_{\theta,k}^a(\mathbf{x}_k)\right\|^2 + 2\left\|s_{\theta,k}^a(\mathbf{x}_k) - s_{\theta,k}^b(\mathbf{x}_k)\right\|^2 \\ &\leq \mathcal{O}\left(\log\left(\frac{1}{n}\right)\right) + \mathcal{O}\left((W)^L \left((S-1)d + \frac{L}{W}\right) \sqrt{\frac{\log\frac{2}{\gamma}}{n_k}}\right) \\ &\leq \mathcal{O}\left((W)^L \left((S-1)d + \frac{L}{W}\right) \sqrt{\frac{\log\frac{2}{\gamma}}{n_k}}\right)\end{aligned} \tag{96}$$

Taking expectation with respect to $\mathbf{x}_k \sim q_k$ on both sides completes the proof. □

C.4 Bounding the Clipping Error

The clipping error ($\mathcal{E}_{\text{clip}}$) accounts for the deviation of the unclipped score network $\overline{s}_{\theta,k}(.)$ from the interval $[a,b]^{(S-1)d}$. This can be bounded as follows.

Lemma 13 (Clipping Error)**.** *Let $\mathcal{X} = [S]^d$ and for each $k \in [0, K-1]$ let the unclipped score be $\overline{s}_{\theta,k}(\cdot) : \mathcal{X} \to \mathbb{R}^{d(S-1)}$. Let the true score $s_k(\cdot) : \mathcal{X} \to [1/C, C]^{d(S-1)}$ and the clipped score $\hat{s}_{\theta,k}(\cdot)$ be component-wise bounded in $[1/C, C]$. The clipped score $\hat{s}_{\theta,k}(\cdot)$ can be component-wise restricted to $[1/C, C]$ as follows:*

$$\hat{s}_{\theta,k}(x_k)_{i,c} \;=\; \min\big\{C,\ \max\{1/C,\ \bar{s}_{\theta,k}(x_k)_{i,c}\}\big\}, i \in [d],\ c \in [S]\backslash\{x_k^i\}. \tag{97}$$

and hence, for any distribution q_k on $\mathcal{X}$,

$$\mathbb{E}_{x_k\sim q_k}\big[\|\hat{s}_{\theta,k}(x_k) - \bar{s}_{\theta,k}(x_k)\|_2^2\big] \;\le\; \mathbb{E}_{x_k\sim q_k}\big[\|\overline{s}_{\theta,k}(x_k) - s_k(x_k)\|_2^2\big]. \tag{98}$$

Proof. Partition the $d(S-1)$ component indices (i,c) of the score vector according to the *unclipped* value:

$$\begin{aligned}
&\kappa_1(x_k) = \{(i,c) : \bar{s}_{\theta,k}(x_k)_{i,c} < a\}, \qquad \kappa_2(x_k) = \{(i,c) : a \le \bar{s}_{\theta,k}(x_k)_{i,c} \le b\},\\
&\kappa_3(x_k) = \{(i,c) : \bar{s}_{\theta,k}(x_k)_{i,c} > b\}.
\end{aligned} \tag{99}$$

By construction of the projection onto $[a,b]$, on $\kappa_1(x_k)$ we have $\hat{s}_{\theta,k}(x_k)_{i,c} = a$; on $\kappa_2(x_k)$, $\hat{s}_{\theta,k}(x_k)_{i,c} = \bar{s}_{\theta,k}(x_k)_{i,c}$; and on $\kappa_3(x_k)$, $\hat{s}_{\theta,k}(x_k)_{i,c} = b$. Therefore, the squared error decomposes as

$$\begin{aligned}
\|\hat{s}_{\theta,k}(x_k) - \bar{s}_{\theta,k}(x_k)\|_2^2 = &\sum_{(i,c)\in\kappa_1(x_k)} \big(a - \bar{s}_{\theta,k}(x_k)_{i,c}\big)^2 + \sum_{(i,c)\in\kappa_2(x_k)} \big(\hat{s}_{\theta,k}(x_k)_{i,c} - \bar{s}_{\theta,k}(x_k)_{i,c}\big)^2\\
&+ \sum_{(i,c)\in\kappa_3(x_k)} \big(b - \bar{s}_{\theta,k}(x_k)_{i,c}\big)^2.
\end{aligned} \tag{100}$$

In Equation (100), $\sum_{(i,c)\in\kappa_2(x_k)} \big(\hat{s}_{\theta,k}(x_k)_{i,c} - \bar{s}_{\theta,k}(x_k)_{i,c}\big)^2 = 0$ as $\hat{s}_{\theta,k}(x_k)_{i,c} = \bar{s}_{\theta,k}(x_k)_{i,c}$ on $\kappa_2(x_k)$. Since $s_k(x_k)_{i,c} \in [a,b]$ component-wise, the projection onto $[a,b]$ is non-expansive in each coordinate. Concretely, for $(i,c) \in \kappa_1(x_k)$,

$$\big(a - \bar{s}_{\theta,k}(x_k)_{i,c}\big)^2 \;\le\; \big(\bar{s}_{\theta,k}(x_k)_{i,c} - s_k(x_k)_{i,c}\big)^2, \tag{101}$$

for $(i,c) \in \kappa_3(x_k)$,

$$\big(b - \bar{s}_{\theta,k}(x_k)_{i,c}\big)^2 \;\le\; \big(\bar{s}_{\theta,k}(x_k)_{i,c} - s_k(x_k)_{i,c}\big)^2, \tag{102}$$

and for $(i,c) \in \kappa_2(x_k)$ we have equality

$$\big(\hat{s}_{\theta,k}(x_k)_{i,c} - \bar{s}_{\theta,k}(x_k)_{i,c}\big)^2 = 0. \tag{103}$$

Summing (101)-(103) over the three index sets and using Equation (100) followed by taking expectation over $x \sim q_k$ proves (98). □

D PROOF OF THE MAIN THEOREM

D.1 Path measure KL decomposition

The following decomposition follows from the Proof of Theorem 2 from Zhang et al. (2025).

$$
\begin{aligned}
D_{\mathrm{KL}}(\mathbb{Q} \,\|\, \mathbb{P}^{q_T}) &= \frac{1}{S} \sum_{k=0}^{K-1} \int_{kh}^{(k+1)h} \mathbb{E}_{\mathbf{x}_t \sim q_t} D_I\big(s_t(\mathbf{x}_t) \,\|\, \hat{s}_{(k+1)h}(\mathbf{x}_t)\big)\, dt \\
&\overset{(i)}{\lesssim} \frac{C}{S} \sum_{k=0}^{K-1} \int_{kh}^{(k+1)h} \mathbb{E}_{\mathbf{x}_t \sim q_t} \big\| s_t(\mathbf{x}_t) - s_{(k+1)h}(\mathbf{x}_t) \big\|_2^2 \, dt \\
&\quad + \frac{C^2}{S} \sum_{k=0}^{K-1} \int_{kh}^{(k+1)h} \mathbb{E}_{\mathbf{x}_t \sim q_t} D_I\big(s_{(k+1)h}(\mathbf{x}_t) \,\|\, \hat{s}_{(k+1)h}(\mathbf{x}_t)\big)\, dt \\
&\overset{(ii)}{\lesssim} \frac{C}{S} \sum_{k=0}^{K-1} \int_{kh}^{(k+1)h} \mathbb{E}_{\mathbf{x}_t \sim q_t} \sum_{i=1}^{d} \sum_{\hat{x}^i \neq x^i} \big| s_t(\mathbf{x}_t)_{i,\hat{x}^i} - s_{(k+1)h}(\mathbf{x}_t)_{i,\hat{x}^i} \big|^2 dt \\
&\quad + \frac{C^3}{S} \sum_{k=0}^{K-1} \int_{kh+\delta}^{(k+1)h+\delta} \mathbb{E}_{\mathbf{x}_t \sim q_t} \big\| s_{(k+1)h+\delta}(\mathbf{x}_t) - \hat{s}_{\theta,(k+1)h+\delta}(\mathbf{x}_t) \big\|_2^2 \, dt \\
&\overset{(iii)}{\lesssim} \frac{C}{S} \sum_{k=0}^{K-1} \int_{kh}^{(k+1)h} \mathbb{E}_{\mathbf{x}_t \sim q_t} \sum_{i=1}^{d} \sum_{\hat{x}^i \neq x^i} \left[\frac{1}{1 - e^{-(k+1)h}} + S \right]^2 \kappa_i^2 h^2 \, dt \\
&\quad + \frac{C^3}{S} \sum_{k=0}^{K-1} h \, \mathbb{E}_{\mathbf{x}_{(k+1)h+\delta} \sim q_{(k+1)h+\delta}} \big\| s_{(k+1)h+\delta}(\mathbf{x}_{(k+1)h+\delta}) - \hat{s}_{\theta,(k+1)h+\delta}(\mathbf{x}_{(k+1)h+\delta}) \big\|_2^2 \\
&\quad + \frac{C(S-1)d\lambda T h}{S} \\
&\overset{(iv)}{\lesssim} C \sum_{k=0}^{K-1} \sum_{i=1}^{d} \left[\frac{1}{(1 - e^{-(k+1)h})^2} + S^2 \right] \kappa_i^2 h^3 \\
&\quad + \frac{C^3}{S} \sum_{k=0}^{K-1} h \, A_k + \frac{C(S-1)d\lambda T h}{S} \\
&\overset{(v)}{\lesssim} C\kappa^2 h^3 \int_h^T \frac{1}{(1 - e^{-x})^2} dx + C\kappa^2 S^2 h^2 T \\
&\quad + \frac{C^3}{S} \sum_{k=0}^{K-1} h \, A_k + \frac{C(S-1)d\lambda T h}{S} \\
&\overset{(vi)}{\lesssim} C\kappa^2 h^3 \big[T - \log h + \tfrac{1}{h} \big] + C\kappa^2 S^2 h^2 T \\
&\quad + \frac{C^3}{S} \sum_{k=0}^{K-1} h \, A_k + \frac{C(S-1)d\lambda T h}{S} \\
&\overset{(vii)}{\lesssim} C\kappa^2 h^2 T (h + S^2) \\
&\quad + \frac{C^3}{S} \sum_{k=0}^{K-1} h \, A_k + \frac{C(S-1)d\lambda T h}{S} \\
&\overset{(viii)}{\lesssim} C\kappa^2 S^2 h^2 T \\
&\quad + \frac{C^3}{S} \sum_{k=0}^{K-1} h \, A_k + \frac{C(S-1)d\lambda T h}{S}.
\end{aligned} \tag{104}
$$

The first equality follows from Lemma 5. Lemma 4 (i) and 6 lead to the first inequality (i). We particularly avoid the assumption $\frac{1}{S}\sum_{k=0}^{K-1}\int_{kh}^{(k+1)h}\mathbb{E}_{\mathbf{x}_t\sim q_t} D_I\big(s_{(k+1)h}(\mathbf{x}_t)\,\|\,\hat{s}_{\theta,(k+1)h}(\mathbf{x}_t)\big)\,dt \le \varepsilon_{\text{score}}$ from Zhang et al. (2025) and further upper bound it by squared L2 norm, leading to the second inequality (ii). The third inequality (iii) arises from Lemma 7 and Lemma 11. The fourth inequality (iv) stems from the fact that $(a+b)^2 \le 2(a^2+b^2)$. The sixth inequality (vi) is a consequence of $e^x - 1 \ge x$, and the final inequality $(viii)$ is due to the assumption that $h \le S^2$.

D.2 KL divergence bound

From Zhang et al. (2025) we use the result that $D_{\text{KL}}(q_T\,\|\,\pi^d) = de^{-T}\log S$. Hence, by writing $M = C(S-1)d$ the final bound on $D_{KL}(p_{data}\,\|\,p_T)$ from (23) and (104):

$$D_{KL}(p_{data}\,\|\,p_T) \;\lesssim\; de^{-T}\log S + C\kappa^2 S^2 h^2 T + \frac{C^3}{S}\sum_{k=0}^{K-1} hA_k + \frac{M\lambda T h}{S}. \tag{105}$$

We shall now focus on bounding the third term in (105) which expresses the score approximation error by writing $A_k = \mathbb{E}_{\mathbf{x}_{(k+1)h}\sim q_{(k+1)h}}\left\|s_{(k+1)h}(\mathbf{x}_{(k+1)h}) - \hat{s}_{\theta,(k+1)h}(\mathbf{x}_{(k+1)h})\right\|_2^2$. For notational simplicity we re-write $A_k = \mathbb{E}_{\mathbf{x}_k\sim q_k}\|s_k(\mathbf{x}_k) - \hat{s}_{\theta,k}(\mathbf{x}_k)\|_2^2$ as mentioned above (28). For $D_{\text{KL}}(q_0\|p_T) \le \tilde{O}(\epsilon)$ to hold, we have to ensure that each of the terms from (105) scales as $\tilde{O}(\epsilon)$. Therefore, we bound the score-approximation error term $\frac{C^3}{S}\sum_{k=0}^{K-1} h\,\mathbb{E}_{\mathbf{x}_{(k+1)h}\sim q_{(k+1)h}}\left\|s_{(k+1)h}(\mathbf{x}_{(k+1)h}) - \hat{s}_{\theta,(k+1)h}(\mathbf{x}_{(k+1)h})\right\|_2^2$ based on the number of samples n_k required to achieve an ϵ_{score}-accurate score estimator.

$$D_{KL}(q\,\|\,p_T) \;\lesssim\; de^{-T}\log S \;+\; C\kappa^2 S^2 h^2 T \;+\; \frac{C^3}{S}\sum_{k=0}^{K-1} hA_k \;+\; \frac{M\lambda T h}{S} \tag{106}$$

$$de^{-T}\log S \;\le\; \epsilon \quad\Longrightarrow\quad T \;\gtrsim\; \log\!\Big(\tfrac{d\log S}{\epsilon}\Big) \tag{107}$$

$$C\kappa^2 S^2 h^2 T \;\le\; \epsilon \;\Longrightarrow\; h \;\lesssim\; \Big(\tfrac{\epsilon}{C\kappa^2 S^2 T}\Big)^{1/2}, \qquad \frac{M\lambda T h}{S} \;\le\; \epsilon \;\Longrightarrow\; h \;\lesssim\; \tfrac{\epsilon S}{M\lambda T} \tag{108}$$

$$h \;\lesssim\; \min\!\Big\{\Big(\tfrac{\epsilon}{C\kappa^2 S^2 T}\Big)^{1/2},\; \tfrac{\epsilon S}{M\lambda T}\Big\} \tag{109}$$

$$K \;=\; \frac{T}{h} \;\gtrsim\; \max\!\Big\{T\Big(\tfrac{C\kappa^2 S^2 T}{\epsilon}\Big)^{1/2},\; \tfrac{M\lambda}{\epsilon}\,\tfrac{T^2}{S}\Big\} \tag{110}$$

$$T \asymp \log\!\Big(\tfrac{d\log S}{\epsilon}\Big) \;\Longrightarrow\; K \;\gtrsim\; \max\!\Big\{\sqrt{\tfrac{C\kappa^2 S^2}{\epsilon}}\,\big[\log(\tfrac{d\log S}{\epsilon})\big]^{3/2},\; \tfrac{M\lambda}{\epsilon S}\,\big[\log(\tfrac{d\log S}{\epsilon})\big]^2\Big\} \tag{111}$$

$$D_{\text{KL}}(p_{\text{data}}\,\|\,p_T) \;\le\; \tilde{O}\Big(\epsilon \;+\; \frac{C^3}{S}\sum_{k=0}^{K-1} h\,A_k\Big) \tag{112}$$

Using the following bounds from earlier proofs: Bound on $\mathcal{E}_k^{\text{stat}}$ from Lemma 2 and $\mathcal{E}_k^{\text{opt}}$ from Lemma 3 as follows, $\mathcal{E}_k^{\text{stat}} \le \mathcal{O}\left((W)^L\left((S-1)d + \frac{L}{W}\right)\sqrt{\frac{\log\frac{1}{\gamma}}{n_k}}\right)$ and $\mathcal{E}_k^{\text{opt}} \le \mathcal{O}\left((W)^L\left((S-1)d + \frac{L}{W}\right)\cdot\sqrt{\frac{\log\frac{2}{\gamma}}{n_k}}\right)$ we obtain the following bound on A_k which holds with a probability of at least $1-\gamma$:

$$A_k \leq \underbrace{\mathcal{O}\left((W)^L\left((S-1)d+\frac{L}{W}\right)\sqrt{\frac{\log\frac{1}{\gamma}}{n_k}}\right)}_{\text{Statistical Error}} + \underbrace{\mathcal{O}\left((W)^L\left((S-1)d+\frac{L}{W}\right)\cdot\sqrt{\frac{\log\frac{2}{\gamma}}{n_k}}\right)}_{\text{Optimization Error}} \tag{113}$$

$$\leq \mathcal{O}\left((W)^L\left((S-1)d+\frac{L}{W}\right)\cdot\sqrt{\frac{\log\frac{2}{\gamma}}{n_k}}\right) \tag{114}$$

We obtain (114) by appropriately combining the statistical and optimization error terms. We have to ensure that A_k satisfies the bound in (114) by setting the parameters n_k (sample size at discrete time-step k). Simultaneously, A_k must also satisfy the following inequality to ensure that the score approximation error is less than ϵ_{score}, which is the desired level of accuracy that we want to achieve, and characterize it with the number of samples and the neural network parameters required. We thus shift the analysis from an assumption of access to ϵ_{score}-accurate score estimator to a learning-theoretic view by characterizing the number of samples n_k and network parameters WD required to achieve it.

$$\frac{1}{S}\sum_{k=0}^{K-1} h\, A_k \leq \epsilon_{\text{score}}. \tag{115}$$

where ϵ_{score} is the desired accuracy of the score estimator.

$$A_k \leq \frac{S}{Kh}\,\epsilon_{\text{score}}. \tag{116}$$

which we obtain as the worst-case bound on the per-discrete time-step squared L2 error. Here, we specifically note that the bound on A_k does not depend on k since the number of samples required to train at the kth time-step is available across all time-steps i.e., $n_k = n$.

We isolate the two sources of error: **statistical and optimization**. We will solve for the sample size $n_k = n$.

So the bound on A_k becomes:

$$A_k \;\leq\; \frac{S}{Kh}\,\epsilon_{\text{score}} \;=\; O\left(\frac{S}{Kh\,C^3}\right). \tag{117}$$

To ensure the above inequality holds, we enforce:

$$W^{\,L}\left((S-1)d+\frac{L}{W}\right)\sqrt{\frac{\log(2/\gamma)}{n_k}} \;\leq\; \frac{S}{T\,C^3}\,\epsilon. \tag{118}$$

$$n_k \;\gtrsim\; \frac{C^6\,T^2}{S^2\epsilon^2}\,W^{\,2L}\left((S-1)d+\frac{L}{W}\right)^2\,\log\left(\frac{2}{\gamma}\right) \tag{119}$$

In (119), recall that $Kh = T$, and T can be set as $T \simeq \log\left(\frac{d\log S}{\epsilon}\right)$ from (107).

$$n_k \;\gtrsim\; \frac{C^6}{S^2\epsilon^2}\,W^{2L}\left((S-1)d+\frac{L}{W}\right)^2\left[\log\left(\frac{d\log S}{\epsilon}\right)\right]^2\,\log\left(\frac{2}{\gamma}\right). \tag{120}$$

Hiding the logarithmic factors, we obtain the following bound:

$$n_k \;=\; \tilde{\Omega}\left(\frac{C^6}{S^2\epsilon^2}\,W^{\,2L}\left((S-1)d+\frac{L}{W}\right)^2\right) \tag{121}$$

This completes the proof of the main theorem.

E HARDNESS OF LEARNING IN DISCRETE-STATE DIFFUSION MODELS

Lemma 14 (Hardness of Learning in Discrete-State Diffusion Models)**.** *The generation of diffusion models with KL gap $< \epsilon$ needs at least $\Theta(1/\epsilon^2)$ samples for some distribution.*

To show this result, we will provide a construction which provides a counterexample showing that one cannot learn discrete distributions within ε-KL error using fewer than $\Theta(1/\varepsilon^2)$ samples. Otherwise, such a learner would violate the classical mean-estimation lower bound.

Setup. Let $x_0, x_1 \in \mathbb{R}$ with $x_0 \neq x_1$, and $0 < \varepsilon < \frac{1}{25}$. Define two-point laws

$$P = (1-\varepsilon)\,\delta_{x_0} + \varepsilon\,\delta_{x_1}, \qquad Q = (1-25\,\varepsilon)\,\delta_{x_0} + 25\,\varepsilon\,\delta_{x_1}. \tag{122}$$

Then

$$\mathbb{E}_P[X] = x_0 + \varepsilon\,(x_1 - x_0), \qquad \mathbb{E}_Q[X] = x_0 + 25\,\varepsilon\,(x_1 - x_0), \tag{123}$$

$$\mathrm{Var}_P(X) = \varepsilon(1-\varepsilon)\,(x_1 - x_0)^2, \qquad \mathrm{Var}_Q(X) = (25\,\varepsilon)(1-25\,\varepsilon)\,(x_1 - x_0)^2, \tag{124}$$

so both means and variances are $\Theta(\varepsilon)$ when $|x_1 - x_0| = \Theta(1)$.

For the two-point laws,

$$D_{\mathrm{KL}}(P\|Q) = (1-\varepsilon)\log\frac{1-\varepsilon}{1-25\varepsilon} + \varepsilon\log\frac{\varepsilon}{25\varepsilon}.$$

Using $\log(1-t) = -t - \frac{t^2}{2} + O(t^3)$ and $\log\frac{\varepsilon}{25\varepsilon} = \log\frac{1}{25}$, we get

$$D_{\mathrm{KL}}(P\|Q) = \big(24 - \log 25\big)\varepsilon + O(\varepsilon^2) = c_1\,\varepsilon + O(\varepsilon^2),$$

with $c_1 \approx 20.78$. As $\varepsilon \to 0$, the leading term dominates, and hence $D_{\mathrm{KL}}(P\|Q) = \Theta(\varepsilon)$.

Hellinger-based KL separation. For two-point laws supported on $\{x_0, x_1\}$ with masses $(1-p, p)$ and $(1-q, q)$, the squared Hellinger distance is

$$H^2\big((1-p,p),(1-q,q)\big) = 1 - \Big(\sqrt{(1-p)(1-q)} + \sqrt{pq}\Big). \tag{125}$$

Substituting $p = \varepsilon$ and $q = 25\,\varepsilon$ gives

$$H^2(P,Q) = 1 - \sqrt{(1-\varepsilon)(1-25\,\varepsilon)} - \sqrt{25}\,\varepsilon. \tag{126}$$

Using $\sqrt{1-t} \le 1 - \frac{t}{2}$ for $t \in [0,1]$ with $t = 26\varepsilon - 25\varepsilon^2$, we have

$$\sqrt{(1-\varepsilon)(1-25\,\varepsilon)} \le 1 - 13\,\varepsilon + 12.5\,\varepsilon^2, \tag{127}$$

and hence

$$H^2(P,Q) \ge (13-5)\varepsilon - 12.5\,\varepsilon^2 = 8\,\varepsilon - 12.5\,\varepsilon^2 > 7.5\,\varepsilon \quad \text{for all } 0 < \varepsilon < \tfrac{1}{25}. \tag{128}$$

Implication for learners. Suppose a learner outputs a distribution R such that for the unknown $U \in \{P, Q\}$,

$$D_{\mathrm{KL}}(U\|R) \le \varepsilon. \tag{129}$$

Since $H^2(P,Q) \le D_{\mathrm{KL}}(P\|Q)$, we have $H(U,R) \le \sqrt{\varepsilon}$. Let V be the other element of $\{P,Q\}$. By the triangle inequality for the Hellinger distance and (128),

$$H(R,V) \ge H(P,Q) - H(U,R) > \sqrt{7.5\,\varepsilon} - \sqrt{\varepsilon} = (\sqrt{7.5} - 1)\sqrt{\varepsilon}, \tag{130}$$

and therefore

$$D_{\mathrm{KL}}(R\|V) \ge H^2(R,V) \ge (\sqrt{7.5} - 1)^2\,\varepsilon > 3\,\varepsilon > \varepsilon. \tag{131}$$

Thus, a learner within ε-KL of one distribution must be at least a constant factor more than ε away from the other.

Mean-testing consequence. Setting $x_0 = 0$ and $x_1 = 1$, the mean gap is

$$\Delta = |\mathbb{E}_P[X] - \mathbb{E}_Q[X]| = 24\,\varepsilon, \tag{132}$$

and the variances are bounded by constants. Assume that from n samples of the true distribution, a learner can construct a simulator R such that $D_{\mathrm{KL}}(U\|R) \le \varepsilon$ for the true $U \in \{P, Q\}$. Suppose we are given n samples from one of P or Q (unknown to us), and we use them to produce such a simulator R. Since R is within ε-KL of one distribution but more than ε away from the other as proved above, we can generate arbitrarily many samples from R and thereby determine whether the original samples came from P or from Q. Thus, the ability to learn a simulator R within ε-KL error implies the ability to distinguish P from Q.

Although P and Q are separated by $D_{\mathrm{KL}}(P\|Q) > 20\varepsilon$, their means differ only by $\Theta(\varepsilon)$, i.e., their separation is linear in ε. If a simulator R achieves $D_{\mathrm{KL}}(U\|R) \le \varepsilon$ for some $U \in \{P, Q\}$, it cannot simultaneously satisfy this bound for both distributions. Therefore, distinguishing whether the samples originated from P or Q means that we can distinguish two distributions whose means differ by $\Theta(\varepsilon)$. By standard mean-testing lower bounds (Wainwright, 2019), distinguishing two distributions whose means differ by $\Theta(\varepsilon)$ requires at least $\Omega(1/\varepsilon^2)$ samples. Hence, any learner that produces such an R with $D_{\mathrm{KL}}(U\|R) \le \varepsilon$ must necessarily use $\Omega(1/\varepsilon^2)$ samples.

F DIMENSION OF THE DISCRETE-STATE SCORE

In principle, if we view the discrete-state diffusion process as a continuous-time Markov chain (CTMC) on the full state space $\mathcal{X} = [S]^d$ with $N = S^d$ states, then the forward rate matrix Q_t would be of size $N \times N$, and the corresponding discrete-state score $s_t(\mathbf{x}) = \big(q_t(\mathbf{y})/q_t(\mathbf{x})\big)_{\mathbf{y}\neq\mathbf{x}}$ would indeed live in $\mathbb{R}^{N-1}$. However, such a formulation is computationally intractable for high-dimensional data.

Following prior works (Chen and Ying, 2024) and (Zhang et al., 2025), we impose a *coordinate-wise (token-wise) factorization* of the forward process. Specifically, each of the d coordinates evolves independently according to a shared $S \times S$ rate matrix. Under this assumption, every jump of the forward CTMC modifies exactly *one coordinate at a time*, and transitions are only allowed between states whose Hamming distance is one. As a result, the full rate matrix Q_t is extremely sparse: from any state $\mathbf{x} \in [S]^d$, there are only $d(S-1)$ nonzero off-diagonal transitions.

The reverse-time CTMC inherits exactly the same sparsity structure. Its rate matrix $Q_t^{\leftarrow}$ involves only transitions of the form

$$\mathbf{x} \;\longrightarrow\; \mathbf{x}_{\setminus i} \odot \hat{x}_i, \qquad i \in [d],\ \hat{x}_i \neq x_i,$$

and for each such transition the reverse rate is given by

$$Q_t^{\leftarrow}\big(\mathbf{x}, \mathbf{x}_{\setminus i} \odot \hat{x}_i\big) \;=\; Q_{T-t}^{\mathrm{tok}}(\hat{x}_i, x_i)\,\frac{q_t(\mathbf{x}_{\setminus i} \odot \hat{x}_i)}{q_t(\mathbf{x})}.$$

Thus, the only probability ratios that appear in the reverse dynamics are those corresponding to single-coordinate replacements.

This observation leads us to express the discrete-state score in the form required by the reverse-time CTMC. We therefore define

$$s_t(\mathbf{x})_{i,\hat{x}_i} \;:=\; \frac{q_t(\mathbf{x}_{\setminus i} \odot \hat{x}_i)}{q_t(\mathbf{x})}, \qquad i \in [d],\ \hat{x}_i \neq x_i,$$

which parametrizes all nonzero transition rates of $Q_t^{\leftarrow}$. For each coordinate i, the collection $\{s_t(\mathbf{x})_{i,\hat{x}_i} : \hat{x}_i \neq x_i\}$ has $S-1$ degrees of freedom, since the probabilities over the S possible symbols at that coordinate must sum to one after normalization. Aggregating across all d coordinates yields a score representation in $\mathbb{R}^{d(S-1)}$.

Importantly, this reduction from $\mathbb{R}^{S^d-1}$ to $\mathbb{R}^{d(S-1)}$ is not an approximation or modeling choice. It is a consequence of the sparsity of the forward and reverse generators under the factorized CTMC assumption, and it provides a complete and tractable characterization of the reverse diffusion dynamics.

G PL CONDITION AND SAMPLE-COMPLEXITY SCALING

This section clarifies the role of the Polyak–Łojasiewicz (PL) condition in our analysis and explains how the stated sample-complexity bounds should be interpreted.

G.1 Role and Interpretation of the PL Condition

This section clarifies the role of the Polyak–Łojasiewicz (PL) condition in the analysis and explains how it should be interpreted within the theoretical framework of our work.

The PL condition is imposed as a *structural regularity assumption on the population loss landscape* $\mathcal{L}_k(\theta)$ at each discrete time index k. Its purpose is to enable control of the optimization error by relating the gradient norm to function-value suboptimality. Importantly, this assumption is not tied to strong convexity, nor is it intended to characterize the degree of overparameterization of the neural network used to parameterize the score.

While overparameterized neural networks form one prominent class of models where PL-type or gradient-dominance conditions are known to hold, such conditions have been established in substantially broader settings. In particular, PL or closely related gradient-dominance properties are known to hold even in regimes that are not heavily overparameterized, *i.e.*, when the number of parameters doesn't exceed the number of effective constraints.

Prior work by Zhou and Liang (2017) rigorously characterizes gradient-dominance and regularity conditions for several nonconvex architectures, including deep linear networks, deep linear residual networks, and one-hidden-layer nonlinear networks, in neighborhoods of global minimizers, without requiring a heavily overparameterized regime. These results show that even relatively modestly sized networks can exhibit a PL-type landscape, at least locally around solutions. This type of geometric behavior is precisely what is required in our analysis: a condition ensuring that small gradient norms imply small suboptimality, which in turn yields linear-type convergence rates for stochastic gradient methods.

In our framework, the PL condition is imposed on each time-indexed *population loss* $\mathcal{L}_k(\theta)$, rather than on the empirical objective. It is used solely as an analytical device to convert bounds on gradient norms into bounds on function-value suboptimality, thereby enabling control of both statistical and optimization errors. The sample-complexity requirement appearing in the main results should therefore be interpreted as a *sufficient condition* ensuring that these errors are controlled at the desired level. This requirement does not preclude the population loss from satisfying a PL condition in regimes where the number of samples is comparable to, or larger than, the number of trainable parameters.

G.2 Sample-Complexity Scaling and Width Assumptions

This section clarifies the interpretation of the sample-complexity bounds derived in the paper and explains the role of the width condition appearing in Lemma 1.

The main analysis yields a sample-complexity bound of the form

$$n \gtrsim \Omega\big(W^{2L}\big)\,, \tag{133}$$

where W denotes the width of the neural network parameterizing the score function and L denotes its depth. This dependence arises from standard Rademacher complexity arguments for deep neural networks, which are known to be conservative, particularly with respect to depth. As a result, the bound should be interpreted as a worst-case sufficient condition ensuring convergence, rather than as a sharp or practically tuned estimate of the number of training samples required. Exponential or highly pessimistic depth dependence is common in current deep-learning theory such as in Benton et al. (2024b) and often substantially overestimates actual data requirements.

When this bound is combined with the sufficient-width condition $W \geq (S-1)d$ used in Lemma 1, the resulting scaling becomes $\Omega((Sd)^{2L})$, which can be numerically large for large vocabularies S or deep networks L. This behavior reflects the proof technique and worst-case generalization guarantees rather than an inherent limitation of discrete-state diffusion models. Importantly, the substitution $W = (S-1)d$ is not required by the main generalization theorem itself; it is introduced solely as a convenient sufficient condition in Lemma 1 to ensure zero approximation error via an explicit construction.

More generally, when $W < (S-1)d$ such as with parameter sharing or low-rank structure, the analysis extends naturally by introducing an explicit approximation error term. Let

$$\varepsilon_{\text{approx}}(W, S, d) \tag{134}$$

denote the infimum over all width-W networks in the considered class of the population approximation error to the target score function. Under this formulation, Lemma 1 and the main excess-risk bound can be restated with an additional additive term $\varepsilon_{\text{approx}}(W, S, d)$. In the special case $W \geq (S-1)d$, the explicit construction implies $\varepsilon_{\text{approx}}(W, S, d) = 0$, recovering the original statement of Lemma 1.

This perspective is standard in learning theory. Approximation error induced by restricting to a given function class is often treated as a fixed constant once the class is chosen, allowing the analysis to focus on estimation and optimization errors that depend on the sample size. For example, in PAC-Bayesian analyses, approximation or misspecification errors are absorbed into a constant term once the hypothesis class is fixed (Mai, 2025). Similarly, in NTK or RKHS based analyses of neural networks, it is typically assumed that the target function lies in, or is well approximated by, the chosen function class, with the resulting misspecification error represented as a constant additive term (Bing et al., 2025). Existing theoretical analyses of diffusion models adopt analogous assumptions, treating approximation error as fixed (Gupta et al., 2024; Guan et al., 2025; Gaur et al., 2025a).

H RELATIONSHIP TO PRIOR WORK

This work is inspired by existing theoretical analyses of discrete-state diffusion models (Zhang et al., 2025; Chen and Ying, 2024), particularly those that decompose the overall error into interpretable components and study convergence under idealized assumptions. Similar decomposition is used in works in reinforcement learning such as (Gaur et al., 2023a), (Gaur et al., 2023b),(Gaur et al., 2024) as well as flow-matching (Gaur et al., 2025a) and diffusion models (Gaur et al., 2025b). We adopt a similar high-level decomposition as a conceptual starting point, but develop technical tools and results tailored to *discrete-state diffusion models trained with finite-width neural networks under nonconvex objectives*.

At a high level, the novelty of this work lies in making the abstract components of prior discrete-state diffusion analyses *explicit, data-dependent, and architecture-aware* in the discrete-state setting.

Approximation error (Lemma 1). Prior analyses typically assume realizability of the score function or work in a fixed functional class, thereby avoiding explicit treatment of approximation error. In contrast, Lemma 1 provides a discrete-state-specific approximation result that characterizes when a finite-width neural network can approximate the discrete-state score function and how this depends on the vocabulary size S and dimension d. This lemma explicitly connects neural network expressivity to the statistical behavior of discrete-state diffusion models, a connection that does not appear in existing analyses.

Statistical error (Lemma 2). While previous works derive convergence bounds at an abstract functional level, our analysis yields an explicit, training data dependent sample-complexity bound under a nonconvex PL-type landscape. This result makes the residual error term in prior analyses concrete and interpretable in terms of network width, depth, vocabulary size, and diffusion time. As a result, the statistical component of the error decomposition becomes an explicit quantity rather than an implicit asymptotic remainder.

Optimization error (Lemma 3). We further analyze how optimization error propagates through the diffusion time horizon. This discrete-time stability analysis quantifies how local errors accumulate across steps and affect the final generated distribution. Such propagation effects are absent from prior analyses as they assume access to an empirical risk minimizer (ERM) and are essential for understanding convergence in the discrete-state setting with finite optimization steps.

Together, these results extend existing theory to a substantially more realistic setting, yielding guarantees that are not accessible through existing analyses.