Generating solution paths of Markovian stochastic differential equations using diffusion models

Xuefeng Gao* Jiale Zha† Xun Yu Zhou‡

January 12, 2026

Abstract

This paper introduces a new approach to generating sample paths of unknown Markovian stochastic differential equations (SDEs) using diffusion models, a class of generative AI methods commonly employed in image and video applications. Unlike the traditional Monte Carlo methods for simulating SDEs, which require explicit specifications of the drift and diffusion coefficients, ours takes a model-free, data-driven approach. Given a finite set of sample paths from an SDE, we utilize conditional diffusion models to generate new, synthetic paths of the same SDE. Numerical experiments show that our method consistently outperforms two alternative methods in terms of the Kullback–Leibler (KL) divergence between the distributions of the target SDE paths and the generated ones. Moreover, we present a theoretical error analysis deriving an explicit bound on the said KL divergence. Finally, in simulation and empirical studies, we leverage these synthetically generated sample paths to boost the performance of reinforcement learning algorithms for continuous-time mean–variance portfolio selection, hinting promising applications of our study in financial analysis and decision-making.

Key words. Stochastic differential equations, Markov property, generative conditional diffusion models, error analysis, KL divergence, portfolio selection, reinforcement learning.

1 Introduction

Stochastic differential equations (SDEs) are an important class of equations for continuous-time stochastic models that have been widely employed in numerous fields, including finance, physics, operations research and machine learning. In many applications, generating sample paths of SDEs is

*Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong, China. E-mail: xfgao@se.cuhk.edu.hk

†Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong, China. E-mail: jialezha@link.cuhk.edu.hk

‡Department of Industrial Engineering and Operations Research and The Data Science Institute, Columbia University, New York, NY 10027, USA. Email: xz2574@columbia.edu

crucial (Glasserman, 2004; Kloeden and Platen, 1999; Song et al., 2021). When an underlying SDE is known and given, i.e., its drift and diffusion coefficients are specified, Monte Carlo simulation is the classical approach for sample path generation with various methods proposed and developed in the literature; see e.g. Kloeden and Platen (1999) and the references therein. This paper instead studies the problem of simulating a Markovian SDE (i.e. the coefficients of the equation depend on the current, instead of the historical, state of the solution) when the underlying equation is *unknown*, with access only to a finite set of sample paths from the unknown SDE. Beyond theoretical curiosity, such a problem is motivated by many applications where one discretely observes only collections of time series whose dynamics can be reasonably captured by Markovian, otherwise totally unknown, SDEs.

Traditional approaches to generating paths from an unknown SDE largely rely on *model-based* methods, where the SDE is assumed to belong to a specific class such as geometric Brownian motions. Parameters of the equation are then estimated using statistical inference techniques (Iacus et al., 2008; Kessler et al., 2012). More recent studies use neural networks to represent drift and diffusion coefficients of SDEs, and construct appropriate loss functions to train these networks; see e.g. Kidger et al. (2021); Zhu and Li (2024). Once such a neural SDE is learned, additional sample paths can then be generated from it by Monte Carlo. On the other hand, several recent papers explore *model-free* methods, leveraging generative models to generate samples directly from unknown SDEs, skipping the step of learning the SDEs themselves.[1] For instance, van Rhijn et al. (2023) utilizes GANs (Goodfellow et al., 2014) to simulate unknown SDEs, while Yang et al. (2024) employs normalizing flows to generate samples from SDEs.

In this paper, we introduce a new model-free (up to the Markovian, diffusion SDE structure), data-driven approach to generate sample paths of SDEs using *conditional* diffusion models, a cutting-edge class of generative AI models that have achieved remarkable success in image and video applications whose performances surpass those of GANs and other generative models (Dhariwal and Nichol, 2021; Ho et al., 2020; Rombach et al., 2022). The essential idea of diffusion models is to use a forward process to gradually turn the unknown target distribution into a simple noise distribution, and then reverse this process to generate new samples from the target distribution. Conditional diffusion models, in particular, can generate data samples that are conditioned on specific input variables; see Section 2.2 for a concise review. Employing conditional diffusion models, this paper makes the following main contributions:

- First, we propose to train a sequence of conditional diffusion models to generate new sample paths from an unknown SDE, given a finite set of discrete-time observations (time series) of the SDE. The key observation is that for *Markovian* SDEs, the distribution of the next stochastic increment depends solely on the current state and time. Thus, the state and time naturally become the conditioning input variables, and we train conditional diffusion models to generate a sample of the next increment. In a sequential and autoregressive fashion, we paste these

[1] This is in the same spirit as in image generation, where one generates more cat pictures based on a set of sample cat pictures without attempting to learn the underlying unknown distribution of the samples.

increment samples over time to generate a new sample path from the unknown SDE.

- Second, we demonstrate the effectiveness of our approach by conducting experiments on various types of SDEs and comparing its performance with two benchmark methods including the neural SDE method in Kidger et al. (2021) and a recent alternative generative approach in Liu et al. (2025). We show that our approach consistently and significantly outperforms the two benchmarks across nearly all experiments, in terms of the Kullback–Leibler (KL) divergence between the generated synthetic SDE paths and the given sample training paths.

- Third, we carry out a rigorous theoretical analysis on the errors associated with our approach. Specifically, we establish an explicit bound on the KL divergence between the target and generated SDE path distributions. As a by-product, this KL bound also yields a bound on the total variation distance via Pinsker's inequality. The resulting bound offers a quantitative measure of the accuracy of our method. To our best knowledge, this is the first theoretical error bound on SDE path generation using conditional diffusion models.

- Finally, to illustrate the potential applications of our framework in decision making, we carry out both a simulation study and an empirical one to demonstrate how we can make use of the synthetically generated sample paths to enrich the dataset for training model-free reinforcement learning (RL) algorithms in continuous-time mean–variance portfolio selection (Jia and Zhou, 2023) and to enhance their performance. Both studies show that when synthetic paths are added in the RL training process, the resulting investment policies achieve the highest Sharpe ratios in the out-of-sample test, primarily due to reductions of the terminal wealth variances. Moreover, our empirical study reveals that incorporating synthetic data does *not* enhance the model-based or plug-in policies, where one first estimates the model parameters and then plugs the estimators into the optimal policies for the portfolio selection problem. The implications from these results are that synthetic paths do not improve the accuracy of estimating model parameters because the generated paths do not contain more information, probabilistically or statistically, than the original sample paths, but they enable the RL agent to explore more possible market scenarios (even if fictitious) and thereby to learn more robust investment strategies. So, *AI-generated paths do not help us understand the environment better, but do help us perform better.*

This paper is related to a variety of topics in literature. In terms of SDE path generation, it is closely related to Liu et al. (2025), as both use conditional diffusion models to autoregressively sample SDE increments. The crucial difference lies in the estimation of the score function, a key component of the diffusion models (Ho et al., 2020; Song et al., 2021). While Liu et al. (2025) design a training-free Monte Carlo empirical estimator for the score function and show that their approach outperforms GANs, we opt for training score neural networks which offer significantly greater scalability as demonstrated in our experiments. Moreover, our work differs from Liu et al. (2025) in two other important aspects: (1) we provide a rigorous error analysis for our diffusion-based

generative method; and (2) we explore the practical applications of SDE path generation to portfolio selection problems in terms of improving decision making performance. Such an error analysis and applications to real-world problems were absent in Liu et al. (2025).

For error analysis, our work is related to a growing body of literature on convergence theory and error bounds for *general* diffusion models; see, e.g., Benton et al. (2024); Chen et al. (2023a); Conforti et al. (2025); Chen et al. (2023b); Fu et al. (2024); Gao et al. (2025); Li et al. (2024); Li and Yan (2025); Tang and Zhao (2024) and the references therein. These studies derive various bounds on errors between target distributions and generated ones, informing the accuracy of their methods in modeling and generating *static* data distributions. However, they do not consider the specific problem of SDE path generation and hence the resulting error propagation issue that arises from applying conditional diffusion models in a *dynamic* and auto-regressive manner. Consequently, our error analysis diverges substantially from these prior papers.

When it comes to financial applications, our research is related to two recent working papers Chen et al. (2025); Aghapour et al. (2025). Chen et al. (2025) apply unconditional diffusion models to simulate high-dimensional asset returns and conclude that, in their specific setting, generated data improve the accuracy of parameter estimation (e.g., mean and covariance) and thus help solve a static portfolio optimization problem in a model-based manner.[2] We take a different direction – we consider continuous-time portfolio selection with model-free RL and enrich the training dataset with synthetic paths. The implication of our results is quite different from that of Chen et al. (2025): at least in our setting, synthetic paths help RL policies but not plug-in ones, which has simple intuitions as discussed earlier. When finalizing our paper, we became aware of a recent working paper Aghapour et al. (2025), which applies diffusion models to discrete-time mean–variance portfolio selection. Specifically, the authors directly model and generate stock price time series using conditional diffusion models, and then utilize the resulting synthetic data as a training environment for a policy gradient algorithm (known as TD3). This seems similar to our approach; however, there are differences in several aspects. They concentrate on discrete-time portfolio selection problems without involving SDEs, whereas we focus on the SDE generation task and demonstrate its application through continuous-time portfolio selection. Moreover, their error bounds are established in a variant of the 2-Wasserstein distance, while ours are in terms of KL divergence.

Lastly, our work is related to a broader and more extensive literature on time series generation or forecasting using generative models such as GANs and diffusion models; see, just to name a few, Narasimhan et al. (2024); Ni et al. (2023); Wiese et al. (2020); Yoon et al. (2019) and the references therein. Due to the absence of the Markov property in general time series, conditioning on the entire past history or its embedded representation is often required for effective generation and forecasting. Although these studies report promising empirical results, they often fall short of

[2] On p. 22 of Chen et al. (2025), it is stated that "This result reflects improvements in both the mean and covariance estimation from diffusion-generated data, but most of the improvements come from the improved covariance estimation, which is not surprising given the very design of our diffusion factor model." However, the authors have not offered an intuitive explanation as to why generated data provide more information than the observed ones leading to better estimation.

rigorous theoretical analyses of the associated errors. By contrast, our work provides a distinctive contribution by focusing on the generation of sample paths from Markovian SDEs with theoretical guarantees. Furthermore, we showcase the application of our results to RL for dynamic portfolio optimization, an important problem in finance where it is standard to use Markovian SDEs to model dynamic asset prices.

The remainder of the paper is organized as follows. Section 2 formulates the problem, outlines the diffusion-based methodology for SDE path generation, and presents the related algorithms. Section 3 is devoted to numerical experiments on four different types of SDEs, including a high-dimensional equation, to compare the performances between our method and two benchmark ones. Section 4 focuses on a theoretical error analysis of our SDE generative approach. In Section 5, we present application to continuous-time mean–variance portfolio selection and conduct both simulated and empirical experiments. Finally, Section 6 concludes. Proofs and other supplementary materials are placed in the appendices.

2 Problem Formulation and Methodology

Notation. We adopt the following notation throughout this paper. The identity matrix in $\mathbb{R}^d$ is denoted by I_d, where d is the dimension. We use $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to denote the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, and $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ the corresponding Gaussian density function. We denote by $\|\cdot\|$ the L_2 norm/spectral norm of vectors/matrices, and by $\|\cdot\|_F$ the Frobenius norm of matrices. A function f is called L-Lipschitz, where $L > 0$, if $||f(\mathbf{x}) - f(\mathbf{x}')||_2 \leq L||\mathbf{x} - \mathbf{x}'||_2$ for any $\mathbf{x}, \mathbf{x}'$. We employ the notation $x \lesssim y$ to indicate that $x \leq Cy$ for some positive constant $C > 0$, and define $x \gtrsim y$ analogously. Furthermore, we write $x \asymp y$ if there exist constants $C_1, C_2 > 0$ such that $C_1 y \leq x \leq C_2 y$. For a random variable $\mathbf{X}$, its distribution is denoted by $P_{\mathbf{X}}$. We write $\mathbf{X} \overset{d}{=} \mathbf{Y}$ if $\mathbf{X}$ and $\mathbf{Y}$ have the same distribution, and $P_{\mathbf{X}} \ll P_{\mathbf{Y}}$ if $P_{\mathbf{X}}$ is absolutely continuous with respect to (w.r.t.) $P_{\mathbf{Y}}$. Two distributions $P_{\mathbf{X}}$ and $P_{\mathbf{Y}}$ are said to be mutually absolutely continuous, or equivalent, if $P_{\mathbf{X}} \ll P_{\mathbf{Y}}$ and $P_{\mathbf{Y}} \ll P_{\mathbf{X}}$. The Kullback–Leibler (KL) between probability distributions $P_{\mathbf{X}}$ and $P_{\mathbf{Y}}$ is $\mathrm{KL}(P_{\mathbf{X}}||P_{\mathbf{Y}}) := \int \log(\frac{\mathrm{d}P_{\mathbf{X}}}{\mathrm{d}P_{\mathbf{Y}}})\mathrm{d}P_{\mathbf{X}}$ if $P_{\mathbf{X}} \ll P_{\mathbf{Y}}$, and is defined to be $+\infty$ otherwise.

2.1 Problem Formulation

Consider a d-dimensional *target* SDE defined on a filtered probability space $\big(\Omega, \mathcal{F}, \mathbb{P}, \{\mathcal{F}_t\}_{0 \leq t \leq T}\big)$:

$$\mathrm{d}\mathbf{X}(t) = \mu(t, \mathbf{X}(t))\mathrm{d}t + \sigma(t, \mathbf{X}(t))\mathrm{d}\mathbf{W}(t), \ t \in [0, T], \tag{1}$$

where $\mu : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ is the drift coefficient, $\sigma : [0, T] \times \mathbb{R}^d \to \mathbb{R}^{d \times m}$ is the diffusion coefficient, and $\mathbf{W}$ is a standard m-dimensional Brownian motion. We assume that the equation starts from a fixed, known initial state $\mathbf{X}(0) \equiv \mathbf{x}(0) \in \mathbb{R}^d$ and has a unique strong solution. However, both functions μ and σ are *unknown,* and we have access only to a set of i.i.d. sample paths of the SDE

(1) on $[0, T]$. Our goal, in mathematical terms, is to generate additional sample paths that exhibit the same distributional properties as those of the *unknown* SDE (1).

In practice, one only observes or generates discrete-time states of (1). Without loss of generality, we consider a uniform grid $\mathcal{T} := \{t_n : t_n = n\Delta t \text{ for } n = 0, 1, ..., N_T\}$ with a fixed time step $\Delta t > 0$ and assume that the number of discrete epochs $N_T = T/\Delta t$ is an integer.[3] The dataset available to us, therefore, consists of $H > 1$ sample paths of $\{\mathbf{X}(t) : t \in [0, T]\}$ observed at grid points in $\mathcal{T}$ (i.e. time series), denoted by

$$\left\{\mathbf{x}^{(i)}(t_0), \mathbf{x}^{(i)}(t_1), \cdots, \mathbf{x}^{(i)}(t_{N_T}), \quad i = 1, ..., H\right\}. \tag{2}$$

Our aim is to generate more time series at the same grid points of the sample paths of (1) that are distributionally as close to those of (2) as possible.[4] To do so, we leverage on the Markov property of the SDE (1) and train conditional diffusion models to simulate the stochastic *increments* of (1). Note that Δt is not necessarily small in many applications, in which case the resulting stochastic increments of the SDE are generally non-Gaussian. This rules out the method of directly sampling Gaussian increments, a generally easy endeavor even in high dimensions. On the other hand, in high-dimensional settings where d is large, classical density estimation methods such as kernel density estimation become impractical, begging for alternative generative models like ours.

2.2 Review of Conditional Diffusion Models

For reader's convenience we now give a brief review of the conditional diffusion models, based on the *continuous-time score-based diffusion models* in Song et al. (2021). Consider a d_c-dimensional condition $\mathbf{c} \sim \mathbf{C}$ and an unknown (conditional) target distribution $p_{\text{target}}(\cdot|\mathbf{c}) := q_0(\cdot|\mathbf{c})$ on $\mathbb{R}^{d_{\text{target}}}$. Given a set of independent training data $\{(\mathbf{c}_i, \mathbf{y}_i(0)) : \mathbf{c}_i \sim \mathbf{C}, \mathbf{y}_i(0) \sim q_0(\cdot|\mathbf{c}), i = 1, \cdots N\}$, the conditional diffusion model first noises the training data with a (non-homogeneous) Ornstein–Uhlenbeck (OU) process:

$$\mathrm{d}\mathbf{Y}(\tau) = -f(\tau)\mathbf{Y}(\tau)\mathrm{d}\tau + g(\tau)\mathrm{d}\mathbf{B}(\tau), \ \mathbf{Y}(0) \sim q_0(\cdot|\mathbf{c}), \ \tau \in [0, T_g], \tag{3}$$

where $\mathbf{B}$ is a standard Brownian motion in $\mathbb{R}^{d_{\text{target}}}$ that is independent of $\mathbf{Y}(0)$, and T_g is a fixed horizon length. Throughout this paper, we consider the popular choice of $f(\tau) := \frac{1}{2}\beta(\tau)$, $g(\tau) := \sqrt{\beta(\tau)}$, $\beta(\tau) := a\tau + b$, where a and b are pre-specified positive constants (Ho et al., 2020; Song

[3] Our approach applies to the case when the initial condition $\mathbf{X}(0)$ follows an unknown distribution and the time grid $\mathcal{T}$ is non-uniform. In particular, in Example 2 of the numerical study (Section 3), we consider a non-uniform time grid $\mathcal{T}$. However, for ease of presentation, we will focus mainly on the uniform time grid throughout the paper.

[4] While the original goal was to generate paths following the same distributions of the solution to (1), the only data available to us are the collection of series in (2). Hence to generate new series close to (2) is the best we could do. This is analogous to generating more cat pictures based on a set of existing cat pictures, a typical generative AI (GenAI) task.

et al., 2021). The noising process (3) is called the *forward process* which admits an analytical solution

$$\mathbf{Y}(\tau) = e^{-\int_0^\tau f(v)\mathrm{d}v}\mathbf{Y}(0) + \int_0^\tau e^{-\int_u^\tau f(v)\mathrm{d}v} g(u)\mathrm{d}\mathbf{B}(u).$$

Given a condition $\mathbf{c}$, we let the marginal density of $\mathbf{Y}(\tau)$ be $q_\tau(\cdot|\mathbf{c})$, which is unknown due to the unknown $p_{\text{target}}(\cdot|\mathbf{c})$. For suitable choices of T_g, a and b, the distribution $q_{T_g}(\cdot|\mathbf{c})$ is close to Gaussian by the convergence property of the OU process. When $\mathbf{Y}(0) = \mathbf{y}(0)$ is given, the conditional distribution of $\mathbf{Y}(\tau)$ is clearly Gaussian and known, which we denote by $q_{\tau|0}(\cdot|\mathbf{y}(0), \mathbf{c})$.

Consider the reverse (in time) denoising process $\{\tilde{\mathbf{Y}}(\tau) := \mathbf{Y}(T_g - \tau), \tau \in [0, T_g]\}$. Under mild assumptions (Anderson, 1982; Cattiaux et al., 2023; Haussmann and Pardoux, 1986), $\tilde{\mathbf{Y}}$ satisfies the reverse SDE starting from $\tilde{\mathbf{Y}}(0) \sim q_{T_g}(\cdot|\mathbf{c})$:

$$\mathrm{d}\tilde{\mathbf{Y}}(\tau) = [f(T_g - \tau)\tilde{\mathbf{Y}}(\tau) + g^2(T_g - \tau)\nabla \log q_{T_g - \tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})]\mathrm{d}\tau + g(T_g - \tau)\mathrm{d}\tilde{\mathbf{B}}(\tau), \tag{4}$$

where $\nabla \log q_\tau(\cdot|\mathbf{c})$ is called the (conditional) score function and is unknown. Note that $\tilde{\mathbf{Y}}(T_g) = \mathbf{Y}(0) \sim p_{\text{target}}(\cdot|\mathbf{c})$, which is the target conditional distribution we want to sample from. Hence, the conditional diffusion model generates new samples by simulating the reverse SDE (4). This requires (a) replacing the unknown $q_{T_g}(\cdot|\mathbf{c})$ by its approximation, which is Gaussian and hence easy to sample from; (b) learning the score function $\nabla \log q_\tau(\cdot|\mathbf{c})$; and (3) discretizing (4) for sample generations.

To learn the unknown conditional score, one can consider training a neural network s_θ parameterized by θ to minimize $\int_0^{T_g} \mathbb{E}\|s_\theta(\tau, \mathbf{Y}(\tau), \mathbf{c}) - \nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\|^2 \mathrm{d}\tau$. This optimization problem is intractable because the objective involves the unknown score. In this paper we apply the following denoising score matching procedure (Song et al., 2021; Vincent, 2011):

$$\min_\theta \mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\mathbb{E}_{\tau\sim\text{Unif}[0,T_g]}\mathbb{E}_{\mathbf{Y}(0)}\mathbb{E}_{\mathbf{Y}(\tau)|(\mathbf{Y}(0),\mathbf{c})}\left[\lambda(\tau)\left\|s_\theta(\tau, \mathbf{Y}(\tau), \mathbf{c}) - \nabla \log q_{\tau|0}(\mathbf{Y}(\tau)|\mathbf{Y}(0), \mathbf{c})\right\|^2\right], \tag{5}$$

where $\text{Unif}[0, T_g]$ is the uniform distribution on $[0, T_g]$, $\mathbf{Y}(0) \sim p_{\text{target}}(\cdot|\mathbf{c})$, $\mathbf{Y}(\tau)|(\mathbf{Y}(0), \mathbf{c}) \sim q_{\tau|0}(\cdot|\mathbf{c})$ which is Gaussian, and $\lambda : [0, T_g] \to \mathbb{R}_+$ is some positive weighting function (e.g. $\lambda(\tau) = g^2(\tau)$). The stochastic gradient descent algorithm and its variants can be applied to solve the empirical version of (5) given samples from $(\mathbf{Y}(0), \mathbf{c})$.[5]

With the conditional score function trained and learned, the sampling process is to simulate (4) in K steps, using the Euler–Maruyama discretization scheme with a uniform time step size of $\Delta\tau = T_g/K$. Specifically, let $\tau_k = k\Delta\tau$ and $\tilde{\mathbf{Z}}(0) \sim \mathcal{N}(0, I_d)$. For $k = 0, 1, \cdots, K-1$, set

$$\tilde{\mathbf{Z}}(\tau_{k+1}) = \tilde{\mathbf{Z}}(\tau_k) + [\tilde{f}(\tau_k)\tilde{\mathbf{Z}}(\tau_k) + \tilde{g}^2(\tau_k)s_\theta(T_g - \tau_k, \tilde{\mathbf{Z}}(\tau_k)), \mathbf{c})]\Delta\tau + \tilde{g}(\tau_k)\sqrt{\Delta\tau}\cdot\zeta_k, \tag{6}$$

where $\tilde{f}(\tau) = f(T_g - \tau), \tilde{g}(\tau) = g(T_g - \tau)$, and ζ_k's are i.i.d. standard Gaussian random vectors. The terminal state $\tilde{\mathbf{Z}}(\tau_K)$ is a generated, desired sample.

[5] We do not employ other popular training techniques for conditional diffusion models, such as the classifier-free guidance (Ho and Salimans, 2021) commonly used for image generation tasks. This is because the data distribution in our SDE path generation task is fundamentally different from images, and applying classifier-free guidance does not yield performance gains in our experiments.

2.3 Generation of SDE Paths via Conditional Diffusion Models

We now apply the general conditional diffusion model to generate sample paths of an unknown SDE. Due to the Markov property of the SDE (1), we construct the desired generative model in an autoregressive manner. Specifically, we generate samples of the stochastic increment $\Delta\mathbf{X}(t;\mathbf{x}) := \mathbf{X}^{t,\mathbf{x}}(t+\Delta t) - \mathbf{x}$ defined by the solution to (1) conditional on $\mathbf{X}^{t,\mathbf{x}}(t) = \mathbf{x}$. For each t_n, we set condition $\mathbf{c}_n = (t_n, \mathbf{x}(t_n))$ sampled from $\mathbf{C}_n = (t_n, \mathbf{X}(t_n))$ and train a conditional diffusion model $G(t_n, \mathbf{x}; \theta_n)$ where $\mathbf{x}(t_n) = \mathbf{x}$, such that $G(t_n, \mathbf{x}; \theta_n)$ consists of a score network s_{θ_n} trained via (5) and generates new samples $\tilde{\mathbf{Z}}_n(\tau_K)$ distributed as $\Delta\mathbf{X}(t_n;\mathbf{x})$ by running (6), i.e.,

$$\tilde{\mathbf{Z}}_n(\tau_K) \overset{d}{\approx} \Delta\mathbf{X}(t_n;\mathbf{x}) := \mathbf{X}(t_n+\Delta t) - \mathbf{x} = \int_{t_n}^{t_n+\Delta t} \mu(t, \mathbf{X}(t))\mathrm{d}t + \int_{t_n}^{t_n+\Delta t} \sigma(t, \mathbf{X}(t))\mathrm{d}\mathbf{W}(t). \quad (7)$$

The entire training process is stipulated in Algorithm 1, where N_T score networks $\{s_{\theta_n} : n = 0, \cdots, N_T - 1\}$ are trained in parallel. Next, to generate new sample paths (time series) from the unknown SDE, we start from $\hat{\mathbf{x}}(t_0) = \mathbf{x}(0)$ and recursively produce

$$\hat{\mathbf{X}}(t_{n+1}) = \hat{\mathbf{x}}(t_n) + \tilde{\mathbf{Z}}_n(\tau_K), \quad \text{for } n = 0, ..., N_T - 1. \quad (8)$$

The detailed generation process of SDE paths is summarized in Algorithm 2.

Remark 1. *Instead of using conditional diffusion models to generate SDE paths in a sequential and autoregressive fashion, an alternative approach is to train a single diffusion model to generate samples of the entire trajectory* $(\mathbf{X}(t_1), \cdots, \mathbf{X}(t_{N_T}))$ *directly. In such a case, the dimension of the target distribution is* $d \times N_T$*, which becomes large with a high-dimensional SDE (large* d*) and/or a large number of observation slots (large* N_T*). While diffusion models can indeed deal with high-dimensional distributions, they nonetheless require more complex network architectures to train the score functions and longer training times. The autoregressive approach we choose instead reduces the dimension and complexity of the generation task, which allows us to use a simple neural network such as a multilayer perceptron (MLP) for training the conditional score function leading to a more efficient solution.*

Algorithm 1 Training process of the SDE generative model $\{G(t_n, \cdot; \theta_n) : n = 0, \cdots, N_T - 1\}$

Require: Training data $\{(\mathbf{x}^{(i)}(t_n), \Delta\mathbf{x}^{(i)}(t_n; \mathbf{x}^{(i)}(t_n))) : i = 1, \cdots, H, n = 0, \cdots, N_T - 1\}$, score networks $\{s_{\theta_n}\}_{n=0}^{N_T-1}$, batch size $m \leq H$, learning rate function $\alpha(\cdot)$, noise schedulers $f(\tau)$ and $g(\tau)$, diffusion horizon T_g, number of diffusion steps K, diffusion time step $\Delta\tau = T_g/K$

for episode $j = 1, ..., J$ **do**

for SDE slot $t_n = 0, \Delta t, 2\Delta t, ..., N_T\Delta t$ **do**

1. Randomly sample a batch of data at t_n:

$$\left\{(\mathbf{c}^{(i_l)}, \mathbf{y}^{(i_l)}(0)) = ((t_n, \mathbf{x}^{(i_l)}(t_n)), \Delta\mathbf{x}^{(i_l)}(t_n; \mathbf{x}^{(i_l)}(t_n))) : l = 1, \cdots, m\right\}$$

2. Randomly sample a diffusion step $k \sim \text{Unif}[1, K]$ and set $\tau_k = k\Delta\tau$
3. Generate noised samples $\{\mathbf{y}^{(i_l)}(\tau_k) : l = 1, \cdots, m\}$, where

$$\mathbf{y}^{(i_l)}(\tau_k) = e^{-\int_0^{\tau_k} f(v)\mathrm{d}v}\mathbf{y}^{i_l}(0) + \left(\int_0^{\tau_k} e^{-\int_u^{\tau_k} 2f(v)\mathrm{d}v} g^2(u)\mathrm{d}u\right)^{1/2} \zeta_l,\ \zeta_l \sim \mathcal{N}(0, I_d)$$

4. Compute the denosing score matching loss

$$L(\theta_n) = \frac{1}{m}\sum_{l=1}^{m} g^4(\tau_k)\left\|s_{\theta_n}(\tau_k, \mathbf{y}^{(i_l)}(\tau_k), \mathbf{c}^{(i_l)}) - \nabla \log q_{\tau_k|0}(\mathbf{y}^{(i_l)}(\tau_k)|\mathbf{y}^{(i_l)}(0), \mathbf{c}^{(i_l)})\right\|^2$$

5. Update $\theta_n \leftarrow \theta_n + \alpha(j)\nabla_{\theta_n} L(\theta_n)$

end for

end for

return score network s_{θ_n} of $G(t_n, \cdot; \theta_n)$ for $n = 0, \cdots, N_T - 1$

Algorithm 2 Generation of synthetic SDE paths

Require: network $\{s_{\theta_n}\}_{n=0}^{N_T-1}$, initial position $\hat{\mathbf{x}}(t_0) = \mathbf{x}(0)$, noise schedulers $f(\tau)$ and $g(\tau)$, diffusion horizon T_g, number of diffusion steps K, diffusion time step $\Delta\tau = T_g/K$

for SDE slot $t_n = 0, \Delta t, 2\Delta t, ..., N_T\Delta t$ **do**

1. Set the condition $\mathbf{c}_n = (t_n, \hat{\mathbf{x}}(t_n))$
2. Simulate $\tilde{\mathbf{y}}(\tau_0) \sim \mathcal{N}(0, I_d)$
3. Simulate the reverse process of the conditional diffusion model:

for diffusion step $\tau_k = 0, \Delta\tau, 2\Delta\tau, \cdots, K\Delta\tau$ **do**

$$\tilde{\mathbf{y}}(\tau_{k+1}) = \tilde{\mathbf{y}}(\tau_k) + [f(T_g - \tau_k)\tilde{\mathbf{y}}(\tau_k) + g^2(T_g - \tau_k)s_{\theta_n}(k, \tilde{\mathbf{y}}(\tau_k), \mathbf{c}_n)]\Delta\tau + g(T_g - \tau_k)\sqrt{\Delta\tau}\zeta_k$$

where $\zeta_k \sim \mathcal{N}(0, I_d)$

end for

4. Set and store the next state $\hat{\mathbf{x}}(t_{n+1}) = \hat{\mathbf{x}}(t_n) + \tilde{\mathbf{y}}(\tau_K)$

end for

return generated path $(\hat{\mathbf{x}}(t_0), \hat{\mathbf{x}}(t_1), \cdots, \hat{\mathbf{x}}(t_{N_T}))$

3 Numerical Study For Path Generation

In this section, we numerically evaluate our generative model using training sample paths that are simulated from the underlying SDEs (Glasserman, 2004). We test our method on four classical examples, three of which are one-dimensional SDEs including a time-homogeneous OU process, a time-inhomogeneous geometric Brownian motion (GBM) and a time-homogeneous Cox–Ingersoll–Ross (CIR) process, and the last a 100-dimensional time-homogeneous GBM. We choose these SDEs because, for the purpose of generating training data, the solutions to these SDEs can be simulated *exactly* in the sense that the joint distribution of simulated values coincides with that of the continuous-time SDE on the simulation time grid (Glasserman, 2004, Chapter 3).

We benchmark our approach against two alternative methods. The first one (termed as SDM-MC), proposed in Liu et al. (2025), is also an autoregressive generative model built on diffusion models but with a Monte Carlo score function estimator (so that score training is not required). The second method is based on Neural SDE proposed in Kidger et al. (2021). We compare the three approaches with metrics including the KL-divergences between the distribution of the training path and that of the generated synthetic SDE path.

For this numerical study, we display the network structure and other configurations of our diffusion models $\{G(t_n, \cdot; \theta_n)\}_{n=0}^{N_T=1}$ in Appendix E. For the two benchmarks, we directly follow their hyperparameter settings in Liu et al. (2025) and Kidger et al. (2021), respectively.

Example 1. A (one-dimensional) time-homogeneous OU process is governed by

$$\mathrm{d}X(t) = \theta(\mu - X(t))\mathrm{d}t + \sigma \mathrm{d}W(t), \; X(0) = x(0). \tag{9}$$

Set $x(0) = 1.5, \mu = 1.2, \theta = 1, \sigma = 0.3$, and $T = 1, \Delta t = 0.05$, resulting in $N_T = 20$ observation slots in total. We collect $H = 100$ "real" OU trajectories in the training dataset (2), each simulated from (9) via Monte Carlo. Then in each experiment we generate 100 synthetic OU paths with each of the three methods, and compute the corresponding KL divergences between the joint distributions of the (discrete observations of) 100 real and synthetic paths using the method from Wang et al. (2009). We repeat 5,000 such experiments.

Table 1 presents the 95% confidence intervals of the resulting KL-divergence values for the three methods. The results show that our model produces paths with distributions much closer to those of the target OU process, significantly outperforming both benchmarks.

Method	Ours	SDM-MC	Neural SDE
KL divergence ↓	**0.1811 ± 0.009**	0.2932 ± 0.010	0.9592 ± 0.014

Table 1: The 95% confidence intervals of KL-divergences between 100 real and synthetic OU paths on time grid $\mathcal{T}$.

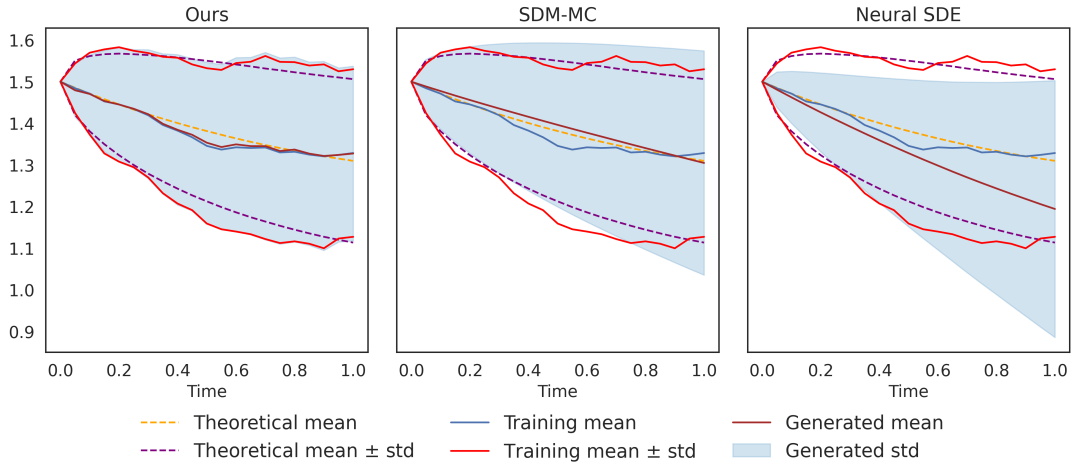

Figure 1: One-dimensional OU process: comparison of mean and standard deviation (std) of solutions obtained by three generative models with 100 synthetic paths. Theoretical and training mean and std are also plotted.

Next, following Liu et al. (2025), we evaluate the performance of the three generative models by comparing the mean and standard deviation (std) of the solutions obtained using 100 synthetic paths. The results are presented in Figure 1, which also includes the theoretical (i.e. oracle) mean and std of the OU process $X(t)$ from (9) for references. As shown, both SDM-MC and our method effectively capture the evolution trend of the training paths, although the latter seems to fit better both the theoretical/training mean and std. By contrast, the Neural SDE method performs poorly.

Finally, a maximum likelihood estimation (MLE) method can be applied to estimate the parameters μ, θ and σ of the target OU process (9) based on a given set of solution paths, as described in Tang and Chen (2009). We apply it to compare the performance (in terms of the estimation accuracy) of the three methods based on 1,000 repeated experiments each on 100 synthetic paths from the corresponding generative models. We also include the parameter estimations based on the given 100 real OU paths for training as reference points, and report the comparison results in Figure 2.

There are two notable observations from Figure 2. First, none of the four can estimate the drift parameters μ and θ as good as the diffusion parameter σ. This nevertheless is consistent with the common knowledge that "volatility" can be estimated reasonably well using standard statistical method whereas "mean" is inherently hard to estimate. Moreover, none of the three generative methods can *improve* the estimation of drift parameters obtained using the 100 OU training trajectories. The reason is because generated paths do not give any more probabilistic/statistical information than those already contained in the original sample/training paths, not to mention that the former inherently contain bias and deviations from the latter. This, in turn, has a significant

implication: *generative methods do not really generate new information.* However, for decision making, generative synthetic data provide more scenarios to train and improve action *strategies*. More on this in Section 5 where we present a concrete application of our method. The other observation is that our method consistently yields more accurate estimates than those from the two benchmark models, with large margins in some cases. For the drift parameters, even though estimation accuracy is not good in general, our method still significantly outperforms the other two and is the closest to the ones obtained by the training paths.

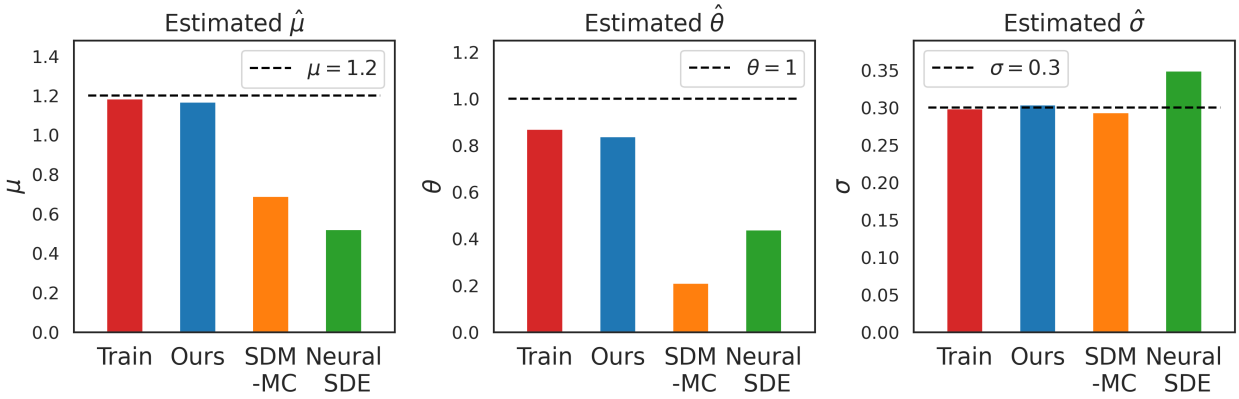

Figure 2: Parameters estimated from 100 training OU paths and 100 synthetic paths obtained from three generative methods.

Example 2. A (one-dimensional) time-inhomogeneous GBM is driven by the following SDE:

$$dX(t) = \mu(t)X(t)dt + \sigma(t)X(t)dW(t), \ X(0) = x(0), \tag{10}$$

where we choose $\mu(t) = 4t, \sigma(t) = \sqrt{4t}$ and $x(0) = 1$. In this example, we set $T = \frac{1}{2}$ and consider a non-uniform time grid $\mathcal{T}$ to demonstrate the flexibility and generalizability of our approach. Specifically, we generate $N_T = 126$ observation time points *at random* on $[0, T]$, and simulate $H = 40$ trajectories of (10) on $\mathcal{T}$ to construct the training dataset.

Method	Ours	SDM-MC	Neural SDE
KL divergence ↓	**1.2265 ± 0.059**	N.A.	6.0490 ± 0.174

Table 2: The 95% confidence intervals of KL-divergences between 40 real and synthetic OU paths on time grid $\mathcal{T}$.

Since the SDM-MC method is designed for time-homogeneous SDEs, we compare our approach with Neural SDE only in this example. As shown in Table 2, the KL divergence between the joint

distribution of the real GBM paths and that of our synthetically generated paths is significantly smaller than the Neural SDE counterpart.[6] Furthermore, Figure 3 illustrates the various means and standard deviations of the solutions generated by the two approaches, revealing that our method accurately captures the evolution trend of the training paths, whereas Neural SDE fails to do so. These results further demonstrate the superiority of our approach over Neural SDE.

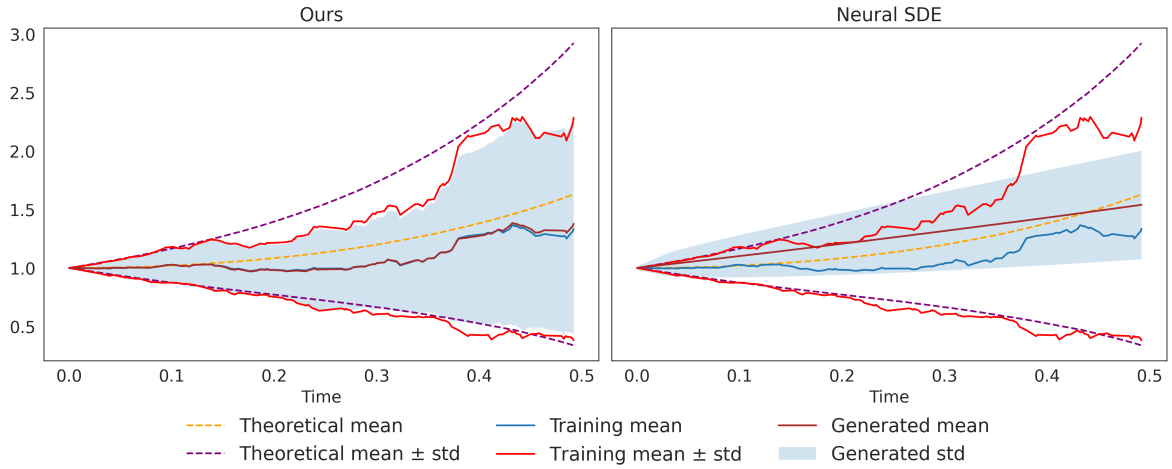

Figure 3: One-dimensional time-inhomogeneous GBM: comparison of mean and standard deviation (std) of solutions obtained by three generative models with 40 synthetic paths. Theoretical and training mean and std are also plotted.

Example 3. A (one-dimensional) time-homogeneous Cox–Ingersoll–Ross (CIR) process is

$$dX(t) = \alpha(b - X(t))dt + \sigma\sqrt{X(t)}dW(t),\ X(0) = x(0).$$

We take $x(0) = 0.5, \alpha = 0.2, b = 0.05, \sigma = 0.1$ and the training data are $H = 50$ real CIR paths with $T = 0.5$ and $\Delta t = 0.01$.

Table 3 presents a comparison of the KL divergences between the real CIR paths and the synthetic ones generated by the three different models. Again, the Neural SDE method exhibits considerably poorer performance, whereas our model beats (if slightly) SDM-MC.

Example 4. A multivariate GBM in $\mathbb{R}^d$ with $d = 100$ is defined as:

$$\mathrm{d}\mathbf{X}_j(t) = \mu_j \mathbf{X}_j(t)\mathrm{d}t + \sigma_j \mathbf{X}_j(t)\mathrm{d}\mathbf{W}_j(t),\ \mathbf{X}_j(0) = x_j(0), \quad j = 1, \cdots, 100,$$

[6]The method in Wang et al. (2009) for computing the KL-divergence between (generally multidimensional) continuous densities uses the distance to the nearest neighbor to represent the likelihood at a given point; hence it works even if the supports of the two datasets concerned are different.

Method	Ours	SDM-MC	Neural SDE
KL divergence ↓	**0.1588 ± 0.012**	0.1641 ± 0.011	1.8196 ± 0.040

Table 3: The 95% confidence intervals of KL-divergences between 50 real and synthetic OU paths on time grid $\mathcal{T}$.

where $\mathbf{W}_j$ is a one-dimensional standard Brownian motion, and $\mathbf{W}_j(t)$ and $\mathbf{W}_l(t)$ have a correlation $\rho_{jl} \in [-1, 1]$ for $j, l = 1, ..., 100$. We randomly generate $\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\rho}$, the corresponding vector or matrices consisting of those model parameters. Set $T = 7$ and $\Delta t = 1$ so that the time step is "large". We use $H = 10,000$ real paths to train our diffusion models.

Method	Ours	SDM-MC	Neural SDE
KL divergence ↓	**38.73 ± 4.51**	1221.54 ± 25.35	1580.06 ± 17.13

Table 4: The 95% confidence intervals of KL-divergences between 10,000 real and synthetic OU paths on time grid $\mathcal{T}$..

Table 4 reports the KL divergence results. This time our method outperforms the other two *massively*. Indeed, both SDM-MC and Neural SDE fail to even maintain the theoretical positivity in over 90% of the generated trajectories. On the other hand, Figure 4 illustrates that our synthetic paths closely track the training trajectories in terms of mean and standard deviation at multiple dimensions.

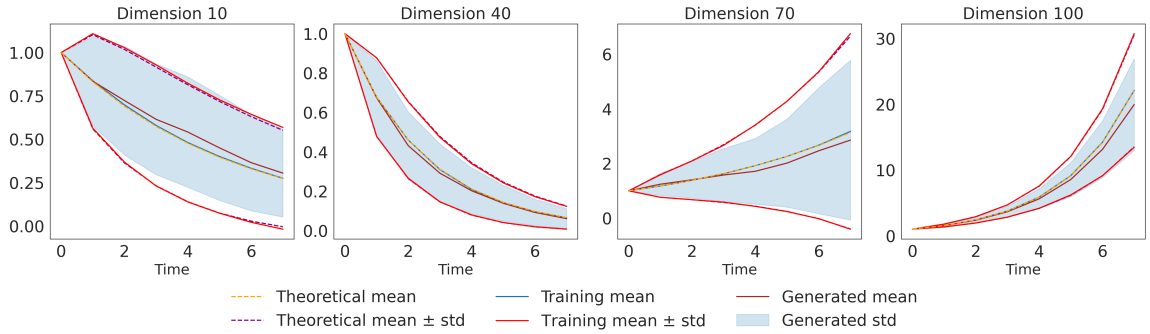

Figure 4: 100-dimensional time-homogeneous GBM: the mean and the standard deviation (std) of solutions (across 4 different dimensions) obtained by our generative method (10,000 synthetic paths) and the ground truth. The mean and std of the 10,000 training paths are also plotted.

Concluding, our approach consistently outperforms the two benchmarks across nearly all experiments in generating synthetic SDE paths. The SDM-MC method requires an extensive amount of

data to construct an accurate Monte-Carlo score estimator for a high-quality generative model. By contrast, our model excels even with limited training trajectories (relative to the dimension) and is applicable to time-inhomogeneous SDEs, offering greater flexibility. Moreover, our model scales effectively to high-dimensional generation tasks, whereas SDM-MC struggles to do so. Meanwhile, the Neural SDE method is often plagued by unstable training dynamics and exhibits high sensitivity to hyperparameter choices, a phenomenon also observed in other studies (Issa et al., 2023; Liao et al., 2024). This inherent instability ultimately contributes to its poor performance in our experiments.

4 Error Analysis

In this section, we provide a theoretical error analysis of our SDE generative model. Specifically, consider the time series corresponding to the target unknown SDE (1) on grid $\mathcal{T}$: $(\mathbf{X}(t_0), \cdots, \mathbf{X}(t_{N_T}))$. Given a fixed $\mathbf{X}(t_0) \equiv \mathbf{x}(0)$, we denote by $P_{1:N_T|0}$ the joint distribution of the target path $(\mathbf{X}(t_1), \cdots, \mathbf{X}(t_{N_T}))$, and by $\hat{P}_{1:N_T|0}$ that of the generated path $(\hat{\mathbf{X}}(t_1), \cdots, \hat{\mathbf{X}}(t_{N_T}))$. Our goal is to derive an explicit bound in the KL-divergence between $P_{1:N_T|0}$ and $\hat{P}_{1:N_T|0}$, so as to quantify the aggregated error of our generative model. In the following, we discuss the well-posedness and decomposition of the KL error in Section 4.1, the error bound for conditional diffusion models in Section 4.2, and the overall error bound for the SDE generative model in Section 4.3.

4.1 Decomposition of the KL Error

The very existence of the KL-divergence $\mathrm{KL}\big(P_{1:N_T|0}||\hat{P}_{1:N_T|0}\big)$ requires $P_{1:N_T|0}$ to be absolutely continuous with w.r.t. $\hat{P}_{1:N_T|0}$. To guarantee such a condition is met and thereby $\mathrm{KL}\big(P_{1:N_T|0}||\hat{P}_{1:N_T|0}\big)$ well defined, we make the following assumption on the target SDE.

Assumption 1. *(a) The coefficients μ and σ of the target SDE (1) satisfy the global Lipschitz and linear growth conditions:*

$$||\mu(t,\mathbf{x}) - \mu(t,\mathbf{x}')||_2 + ||\sigma(t,\mathbf{x}) - \sigma(t,\mathbf{x}')||_2 \le C_L||\mathbf{x}-\mathbf{x}'||_2,$$
$$||\mu(t,\mathbf{x})||_2 + ||\sigma(t,\mathbf{x})||_2 \le C_L(1+||\mathbf{x}||_2),$$

for every $t \in [0,T], \mathbf{x} \in \mathbb{R}^d, \mathbf{x}' \in \mathbb{R}^d$, where C_L is a positive constant independent of $(t,\mathbf{x},\mathbf{x}')$.

(b) For any $0 \le s \le t \le T$, the conditional distribution of $\mathbf{X}(t)$ given $\mathbf{X}(s) = \mathbf{x}(s)$ is absolutely continuous w.r.t. the Lebesgue measure, and it admits a probability density function that is positive almost everywhere on $\mathbb{R}^d$.

Assumption 1(a) is the classical (sufficient) condition for a general SDE to have a unique strong solution (Karatzas and Shreve, 2014). Moreover, under this assumption the solution to (1) admits the following moment bounds (Kloeden and Platen, 1999, Theorem 4.5.4): there exists a positive constant C_M depending only on C_L such that

$$\mathbb{E}\|\mathbf{X}(t)\|^2 \le (1+\mathbb{E}\|\mathbf{X}(0)\|^2)e^{tC_M}$$

for any $t \in [0, T]$, and

$$\mathbb{E}\|\mathbf{X}(t_{n+1}) - \mathbf{X}(t_n)\|^2 \leq M_2(t_n; T) \cdot e^{\Delta t C_M} \Delta t \tag{11}$$

for any $t_n \in \mathcal{T}$, where

$$\begin{aligned} M_2(t_n; T) &:= D_1 \cdot (T - t_n + 1)(1 + \mathbb{E}\|\mathbf{X}(t_n)\|^2) \\ &\leq D_2 \cdot (T - t_n + 1)(1 + \mathbb{E}\|\mathbf{X}(0)\|^2) e^{t_n C_M}, \end{aligned}$$

and D_1 and D_2 are some positive constants depending only on C_L. These moment bounds will be useful for the subsequent error analysis.

Assumption 1(b) is imposed to ensure that the joint distributions $P_{1:N_T|0}$ and $\hat{P}_{1:N_T|0}$ share the same support and the absolute continuity condition holds. Drifted Brownian motions, OU processes, and Langevin SDEs are classical examples satisfying Assumption 1(b). General sufficient conditions for Assumption 1(b) to hold have also been extensively studied in Malliavian calculus and support theorem of SDE; see, e.g. Gyöngy and Pröhle (1990); Michel and Pardoux (1990); Nualart (2006); Stroock and Varadhan (1972). Some SDEs, such as GBMs, do not satisfy Assumption 1(b) because the marginal distribution is not supported on the entire space $\mathbb{R}^d$. For such SDEs, one may conduct some transformations (e.g. the logarithmic one) on their states so that the transformed SDEs have transition densities supported on $\mathbb{R}^d$.

We now introduce a few more notations to facilitate exposition. For any $t_n \in \mathcal{T}$, denote by $P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}$ the conditional distribution of the increment $\Delta\mathbf{X}(t_n; \mathbf{x}(t_n))$ of the target SDE given $\mathbf{X}(t_n) = \mathbf{x}(t_n) \in \mathbb{R}^d$. Let $\Delta\hat{\mathbf{X}}(t_n; \mathbf{x}(t_n))$ be the corresponding generated increment (i.e. $\tilde{\mathbf{Z}}_n(\tau_K)$ in (7)), with $\hat{P}_{\Delta\hat{\mathbf{X}}(t_n;\mathbf{X}(t_n))}$ its distribution. Under Assumption 1, we have the following useful result, the proof of which is deferred to Appendix A.

Proposition 1. *Suppose that Assumption 1 holds. Then*

1. *Both $P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}$ and $\hat{P}_{\Delta\hat{\mathbf{X}}(t_n;\mathbf{X}(t_n))}$ are equivalent to the Lebesgue measure and admit density functions that are strictly positive everywhere on $\mathbb{R}^d$.*

2. *Both $P_{1:N_T|0}$ and $\hat{P}_{1:N_T|0}$ are equivalent to the Lebesgue measure and admit density functions that are strictly positive everywhere on $\mathbb{R}^{d\times N_T}$. Moreover, we have*

$$KL\big(P_{1:N_T|0}||\hat{P}_{1:N_T|0}\big) = \sum_{n=0}^{N_T-1} \mathbb{E}_{\mathbf{X}(t_n)\sim P_{n|0}}\Big[KL\big(P_{\Delta\mathbf{X}(t_n;\mathbf{X}(t_n))}\big|\big|\hat{P}_{\Delta\hat{\mathbf{X}}(t_n;\mathbf{X}(t_n))}\big)\Big], \tag{12}$$

where $P_{n|0}$ is the marginal distribution of the SDE solution $\mathbf{X}(t_n)$ conditional on $\mathbf{X}(t_0) \equiv \mathbf{x}(t_0)$.

The decomposition in (12) arises from the chain rule for KL divergence (Wainwright, 2019, Exercise 3.2) together with the Markovian nature of the target and generated paths. This fundamental

property of chain rule is the primary technical motivation for us to use KL divergence for the error analysis. As a direct consequence of (12), we can simplify the task of bounding the KL-divergence between the target and generated paths. Specifically, it suffices to derive an error bound for *each* conditional diffusion model of producing the increment $\Delta\hat{\mathbf{X}}(t_n; \mathbf{X}(t_n))$. We will delve into this crucial step in the next section.

4.2 Error Bound for Conditional Diffusion Models

In this section, we establish a KL error bound for the general conditional diffusion model discussed in Section 2.2, providing a theoretical guarantee on its performance. The result actually goes beyond the specific application of SDE path generation. While its proof is similar in spirit to the general error results for the unconditional diffusion model, there are certain subtle differences which is the reason why we present it for reader's convenience.

Recall that $p_{\text{target}}(\cdot|\mathbf{c})$ denotes the unknown target conditional distribution. We denote by $\tilde{p}^{\theta}_{\tau_K}(\cdot|\mathbf{c})$ the marginal distribution of the generated sample $\tilde{\mathbf{Z}}(\tau_K)$ in (6) given condition $\mathbf{c}$.

The following result provides the desired KL error bound for the conditional diffusion model. Its proof is lengthy, which is delayed to Appendix B.

Proposition 2. *Suppose the following conditions hold:*

1. *For any given condition $\mathbf{c}$, the target conditional distribution admits a twice continuously differentiable density function $p_{target}(\cdot|\mathbf{c})$, where the score function $\nabla \log p_{target}(\cdot|\mathbf{c})$ is $L(\mathbf{c})$-Lipschitz continuous with*

$$\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}[L(\mathbf{c})] = L_1 < \infty. \tag{13}$$

2. *The target distribution has a finite second moment, i.e.*

$$\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\left[\mathbb{E}_{\mathbf{Y}(0)\sim p_{target}(\cdot|\mathbf{c})}||\mathbf{Y}(0)||^2\right] < M_2 \tag{14}$$

   for some positive constant M_2.

3. *There exists a small constant $\varepsilon_{score} > 0$ such that*

$$\frac{1}{K}\sum_{k=1}^{K}\left[\max\{g^4(\tau_k), 1\}\cdot \mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\left[\mathbb{E}_{\mathbf{Y}(\tau_k)\sim q_{\tau_k}(\cdot|\mathbf{c})}\left\|\nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c}) - s_\theta(\tau_k, \mathbf{Y}(\tau_k), \mathbf{c})\right\|^2\right]\right]^{1/2}$$
$$\leq \varepsilon_{score}. \tag{15}$$

Then, we have

$$\begin{aligned}
&\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\left[KL\big(p_{target}(\cdot|\mathbf{c})||\tilde{p}^{\theta}_{\tau_K}(\cdot|\mathbf{c})\big)\right] \\
&\lesssim \underbrace{e^{-\frac{a}{2}T_g^2-bT_g}\big[M_2+de^{-\frac{a}{2}T_g^2-bT_g}\big]}_{(i)} + \underbrace{\sqrt{M_2+d}\cdot K\varepsilon_{score}+T_gK\varepsilon^2_{score}}_{(ii)} + \underbrace{\Delta\tau(M_2+d)(T_g+1)^5}_{(iii)} \\
&\quad + \underbrace{M_2\Delta\tau(T_g+1)^2+d\Delta\tau\big[L_1+(T_g+1)^4\big]}_{(iv)} + \underbrace{d(\Delta\tau)^2(L_1+T_g+1)}_{(v)}.
\end{aligned} \tag{16}$$

In the following, we provide interpretations of the error bound components in (16), and discuss the conditions imposed in Proposition 2:

1. The term (i) of (16) is the *initialization error*, caused by sampling the reverse SDE (4) from a Gaussian noise rather than the terminal state of the forward SDE (3). This error will be small if $e^{-\frac{a}{2}T_g^2-bT_g}$ is small. For image generation tasks, one typical choice is to set $T_g = 1$, $a = 19.9$ and $b = 0.1$ (Song et al., 2021, Appendix C), resulting in a very small value of $e^{-\frac{a}{2}T_g^2-bT_g}$. This is also the setting we take in this paper, and it is different from the one in the prior theoretical studies (Benton et al., 2024; Chen et al., 2023a) where $a = 0$ and $b = 1$, in which case one needs to choose a large T_g to make the initialization error small.

2. The term (ii) of (16) is the *score matching error*, which measures the quality of the neural network s_θ for approximating the unknown score function $\nabla \log q_\tau$ at each $\tau = \tau_k$. This term will be small, if $\varepsilon^2_{\text{score}}$ in (15) is small. Meanwhile, the condition (15) quantifies the score matching error averaged over K steps of the generation process. This condition is standard and used in most convergence analysis of diffusion models; see e.g. Chen et al. (2023b); Benton et al. (2024); Chen et al. (2023a); Gao et al. (2025).

3. The terms (iii), (iv) and (v) represent the *discretization error* of simulating the reverse SDE (4) by the Euler–Maruyama scheme in (6). In particular, for $\tau \in [\tau_k, \tau_{k+1})$,

   3(a) the term (iii) corresponds to the state discretization error induced by approximating $f(\tau)\mathbf{Y}(\tau)$ with $f(\tau_k)\mathbf{Y}(\tau_k)$,

   3(b) the term (iv) is the score function discretization error caused by using $g(\tau_k)\nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})$ to replace $g(\tau)\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})$, and

   3(c) the term (v) captures the approximation error of simulating the diffusion term $g(T_g - \tau)\mathrm{d}\mathbf{B}(\tau)$ by $g(T_g - \tau_k)\sqrt{\Delta\tau}\zeta_k$.

   Clearly, the discretization error can be made small if $\Delta\tau$ is small, i.e., if the number of sampling steps K in the generation process is large, so that the Euler–Maruyama scheme approximates the reverse SDE (4) well. Because the drift of (4) involves the score function, we impose the first (Lipschitz score) condition in Proposition 2 to control the discretization error.

Our proof of Proposition 2 builds on two recent papers on KL error analysis of unconditional diffusion models (Benton et al., 2024; Chen et al., 2023a), which focus on a simplified case with $f = \frac{1}{2}$ and $g = 1$ in the forward process (3). We consider conditional diffusion models with time-dependent functions f and g, where the error characteristics of the (discretized) sample generation process are different. It is worth noting that Benton et al. (2024) and Chen et al. (2023a) do not require the Lipschitz score condition owing to an early-stopping technique: the reverse process (6) is terminated at time $T_g - \Delta\tau$ instead of T_g. By doing so, they can bound the KL error between the distribution of generated samples and the forward process distribution at time $\Delta\tau$, which corresponds to a small noise perturbation of the original data distribution. We do not pursue the early stopping in this paper for two main reasons. First, we prefer a simpler and cleaner presentation. Second, applying early stopping to our autoregressive SDE generation raises subtle technical issues when we apply conditional diffusion models sequentially over the time grid. This will be discussed in more details in Section 4.3.

We also note that a recent working paper (Fu et al., 2024) presents an error analysis of conditional diffusion models with classifier-free guidance, focusing on the total variation (TV) distance. Proposition 2 in the present paper offers a complementary result, exploring the error bound in KL divergence (which is stronger than TV by Pinsker's inequality) for conditional diffusion models without classifier-free guidance.

4.3 Error Bound for Our SDE Generative Model

In this section, we derive the error bound for our SDE generative model, which is the main theoretical result of the paper. According to Proposition 1, we only need to apply Proposition 2 to bound the KL-error for each conditional diffusion model $G(t_n, \mathbf{x}(t_n); \theta_n)$. The following two assumptions ensure the conditions of Proposition 2 hold.

Assumption 2 (Lipschitz SDE increment)**.** *For any $t_n \in \mathcal{T}$, the increment $\Delta\mathbf{X}(t_n; \mathbf{x}(t_n))$ of the target SDE (1) conditional on $\mathbf{X}(t_n) = \mathbf{x}(t_n)$ admits a twice continuously differentiable density function $P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}$, and $\nabla \log P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}$ is $L_n(\mathbf{x}(t_n))$-Lipschitz continuous with*

$$\mathbb{E}_{\mathbf{X}(t_n)\sim P_{n|0}}[L_n(\mathbf{X}(t_n))] = L_1(t_n) < \infty.$$

Assumption 3 (Score matching error)**.** *There exists $\varepsilon_{score} > 0$ such that for each t_n in $\mathcal{T}$, the corresponding learned score function $s_{\theta_n}(k, \cdot, t_n, \mathbf{x}(t_n))$ of $G(t_n, \mathbf{x}(t_n); \theta_n)$ satisfies*

$$\frac{1}{K_n}\sum_{k=1}^{K_n}\left[\max\{g^4(\tau_k), 1\} \cdot \mathbb{E}_{\mathbf{c}_n\sim\mathbf{C}_n}\left[\mathbb{E}_{\mathbf{Y}(\tau_k)\sim q_{\tau_k}(\cdot|\mathbf{c}_n)}||\nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c}_n) - s_{\theta_n}(\tau_k, \mathbf{Y}(\tau_k), \mathbf{c}_n)||^2\right]\right]^{1/2}$$
$$\leq \varepsilon_{score},$$

where $\mathbf{C}_n = (t_n, \mathbf{X}(t_n))$ with $\mathbf{X}(t_n) \sim P_{n|0}$ as in Proposition 1, the score function $\nabla \log q_{\tau_k}(\cdot|\mathbf{c}_n)$ is defined based on the forward process (3) for which the target distribution is that of $\Delta\mathbf{X}(t_n; \mathbf{x}(t_n))$ in (7), and K_n is the number of the diffusion steps of $G(t_n, \mathbf{x}(t_n); \theta_n)$.

In the context of generating SDE paths, Assumption 2 can be verified to hold for certain special equations such as OU processes. For a more general target SDE, if it is observed at a high frequency, i.e., the time intervals $t_{n+1} - t_n$ are small, then the increment $\Delta\mathbf{X}(t_n; \mathbf{x}(t_n))$ is approximately Gaussian by virtue of the Euler–Maruyama method. In this case, Assumption 2 is also likely to hold. As mentioned earlier, applying the early stopping technique to relax Assumption 2 is particularly challenging for autoregressive SDE path generation. Specifically, if each increment $\Delta\mathbf{X}(t_n; \mathbf{x}(t_n))$ is perturbed by a small amount of noise due to the early stopping, these noises will propagate and accumulate over time along the SDE path, making it difficult to establish a bound on the KL-divergence between the perturbed target SDE paths and generated paths. Hence, we opt out of this approach.

We now state our main theoretical result, which provides an explicit bound on the KL-divergence between the distributions of $P_{1:N_T|0}$ and $\hat{P}_{1:N_T|0}$.

Theorem 1. *Consider a target SDE (1) satisfying Assumption 1 and 2. If Assumption 3 holds for the conditional diffusion model $G(t_n, \cdot; \theta_n)$ at each $t_n \in \mathcal{T}$, then*

$$
\begin{aligned}
&KL(P_{1:N_T|0} || \hat{P}_{1:N_T|0}) \\
&\lesssim \sum_{n=0}^{N_T-1} \Bigg\{ \underbrace{e^{-\frac{a}{2}T_g^2(t_n) - bT_g(t_n)} \left[M_2(t_n; T) e^{\Delta t C_M} \Delta t + d e^{-\frac{a}{2}T_g^2(t_n) - bT_g(t_n)} \right]}_{\textit{initialization error}} \\
&\qquad + \underbrace{\left[(M_2(t_n; T) e^{\Delta t C_M} \Delta t + d \right]^{1/2} \cdot K_n \varepsilon_{score} + T_g(t_n) K_n \varepsilon_{score}^2}_{\textit{score matching error}} \\
&\qquad + \underbrace{\frac{1}{K_n} \big(M_2(t_n; T) e^{\Delta t C_M} \Delta t + d \big) (T_g(t_n) + 1)^6 + \frac{d T_g(t_n) L_1(t_n)}{K_n}}_{\textit{discretization error}} \Bigg\},
\end{aligned}
$$

where Δt is the time step of observing (1), $M_2(t_n; T)$ and C_M are defined in (11), and $T_g(t_n) = K_n \Delta\tau_n$ is the diffusion time horizon of $G(t_n, \mathbf{x}; \theta_n)$.

The proof of Theorem 1 directly follows from Proposition 1, Proposition 2 and the moment bound of the SDE increment in (11). We can also immediately infer from Theorem 1 and Pinsker's inequality to obtain an error bound in the total variation (TV) distance between $P_{1:N_T|0}$ and $\hat{P}_{1:N_T|0}$:

$$
\mathrm{TV}\big(P_{1:N_T|0} || \hat{P}_{1:N_T|0}\big) \leq \sqrt{\frac{1}{2} \mathrm{KL}\big(P_{1:N_T|0} || \hat{P}_{1:N_T|0}\big)}.
$$

These obtained bounds quantify the accuracy of our method.

5 Application to Mean–Variance Portfolio Selection

In this section, we conduct both a simulation study (Section 5.1) and an empirical one (Section 5.2) to demonstrate how our generative approach can be useful in financial portfolio optimization.

5.1 Simulation Study

We first briefly review the continuous-time mean–variance portfolio selection problem where SDE modeling is essential; see e.g. Wang and Zhou (2020); Zhou and Li (2000). An agent invests in a risk-free asset (e.g., a saving account) whose interest rate is r and a risky asset (e.g., a stock), the price of which is governed by an SDE $\{S(t) : t \geq 0\}$, which is a geometric Brownian motion with drift μ and volatility $\sigma > 0$. The agent has a fixed investment horizon T and an initial endowment x_0. She rebalances her portfolio with a strategy $a = \{a(t), t \in [0, T]\}$ in a self-financing fashion, where $a(t)$ is the discounted dollar value invested in the risky asset at time t. Then her discounted wealth process satisfies (Jia and Zhou, 2023, Section 7.1):

$$\mathrm{d}X^a(t) = a_t \frac{\mathrm{d}\big(e^{-rt}S(t)\big)}{e^{-rt}S(t)}, \ X^a(0) = x(0). \tag{17}$$

The problem is to find a strategy a that minimizes the variance of the terminal wealth for a given target mean wealth level z, i.e.,

$$\min_a \mathrm{Var}(X^a(T)), \ \ s.t. \ \mathbb{E}[X^a(T)] = z. \tag{18}$$

Recently, Jia and Zhou (2023) apply a general model-free continuous-time reinforcement learning (RL) approach, termed as $q-$learning, to solve the above problem when the dynamic of the risky asset $S(\cdot)$ is *unknown* (hence following an unknown SDE). We now show how synthetic asset price paths can help improve their $q-$learning algorithm by enriching the policy training.

In our simulation study, we take $X^a(0) = 1$, $r = 2\%$, $T = 0.5$, $S(0) = 1$, and $z = 1.2$. We consider different parameter configurations with drift $\mu \in \{0, \pm 0.1, \pm 0.3, \pm 0.5\}$ and volatility $\sigma \in \{0.2, 0.3, 0.4\}$ for simulating the market environment. For each pair of (μ, σ), we are given only $H = 40$ "real" GBM paths simulated via Monte Carlo with a time step $\Delta t = 1/252$ representing one trading day. These 40 GBM paths consistute the training dataset to train our SDE generative model, whose specific settings are shown in Appendix E. We implement the $q-$learning algorithm in Jia and Zhou (2023) with various choices of H_s, the number of synthetic GBM paths generated by our model at each episode to augment the original 40 training paths. This includes $H_s = 0, 1, 10, 40$ and 360. The details of the RL algorithm we implement are given as Algorithm 3 in Appendix D, which is a direct extension of Algorithm 5 in Jia and Zhou (2023) by including synthetic price paths. After training the RL policies with Algorithm 3, we perform out-of-sample tests on 100,000 GBM paths (simulated from the oracle model via Monte Carlo) to empirically estimate $\mathbb{E}[X^a(T)]$ and $\mathrm{Var}(X^a(T))$ and calculate the corresponding Sharpe ratio, which is $(\mathbb{E}[X^a(T)] - 1)/\sqrt{\mathrm{Var}(X^a(T))}$.

The results are presented in Tables 5 and 6, which compare the performances of the RL algorithm, Algorithm 3, with various numbers of additional synthetic price paths ($H_s = 1, 10, 40$, and 360) together with the case where no synthetic paths are used ($H_s = 0$). We summarize the key findings as follows:

- Incorporating synthetic paths into the RL training dataset substantially enhances the Sharpe ratio of the resulting RL policies across nearly all parameter configurations (21 in total).

setup		$H(40)+H_s(0)$		$H(40)+H_s(1)$		$H(40)+H_s(10)$		$H(40)+H_s(40)$		$H(40)+H_s(360)$	
μ	σ	Mean(Var)	Sharpe	Mean(Var)	Sharpe	Mean(Var)	Sharpe	Mean(Var)	Sharpe	Mean(Var)	Sharpe
0.5	0.4	1.164 (0.034)	0.8929	1.045 (0.002)	0.9255	1.052 (0.003)	0.9348	1.054 (0.003)	**0.9354**	1.054 (0.003)	**0.9354**
0.3	0.4	1.070 (0.020)	0.5011	1.124 (0.055)	**0.5290**	1.128 (0.059)	0.5265	1.123 (0.055)	0.5252	1.116 (0.049)	0.5253
0.1	0.4	1.063 (0.237)	0.1292	1.078 (0.292)	**0.1441**	1.076 (0.275)	**0.1441**	1.071 (0.243)	**0.1441**	1.068 (0.224)	**0.1441**
0	0.4	1.161 (33.540)	0.0278	1.152 (29.993)	0.0278	1.151 (29.945)	0.0278	1.124 (19.902)	0.0278	1.124 (19.912)	0.0278
-0.1	0.4	1.229 (2.273)	0.1521	1.209 (1.705)	0.1600	1.196 (1.409)	0.1647	1.186 (1.263)	**0.1659**	1.180 (1.174)	**0.1659**
-0.3	0.4	1.309 (0.361)	0.5152	1.217 (0.176)	0.5166	1.184 (0.127)	0.5176	1.156 (0.090)	**0.5177**	1.133 (0.066)	**0.5177**
-0.5	0.4	1.271 (0.113)	0.8058	1.195 (0.055)	0.8340	1.171 (0.041)	0.8436	1.165 (0.038)	**0.8459**	1.165 (0.038)	**0.8459**
0.5	0.3	1.191 (0.019)	1.3739	1.199 (0.019)	1.4435	1.201 (0.018)	**1.4939**	1.202 (0.018)	1.4892	1.202 (0.018)	1.4892
0.3	0.3	1.139 (0.044)	0.6594	1.150 (0.050)	0.6673	1.148 (0.049)	0.6690	1.146 (0.047)	0.6696	1.143 (0.045)	**0.6698**
0.1	0.3	1.094 (0.271)	0.1797	1.217 (1.412)	0.1823	1.207 (1.273)	0.1837	1.193 (1.102)	**0.1840**	1.182 (0.983)	**0.1840**
0	0.3	1.075 (11.706)	0.0219	1.055 (3.352)	0.0302	1.048 (2.019)	0.0335	1.043 (1.610)	**0.0338**	1.039 (1.321)	0.0337
-0.1	0.3	1.225 (1.007)	0.2244	1.205 (0.788)	0.2305	1.193 (0.687)	0.2329	1.186 (0.638)	**0.2330**	1.180 (0.601)	**0.2330**
-0.3	0.3	1.255 (0.148)	0.6641	1.216 (0.102)	0.6770	1.188 (0.075)	0.6856	1.182 (0.070)	**0.6876**	1.182 (0.070)	**0.6876**
-0.5	0.3	1.274 (0.063)	1.0864	1.233 (0.044)	1.1060	1.198 (0.031)	1.1194	1.189 (0.028)	**1.1226**	1.188 (0.028)	**1.1226**
0.5	0.2	1.189 (0.003)	3.4002	1.191 (0.002)	3.5812	1.191 (0.002)	3.8123	1.191 (0.002)	**3.9788**	1.190 (0.002)	**3.9788**
0.3	0.2	1.206 (0.030)	1.1755	1.206 (0.027)	1.2427	1.206 (0.026)	1.2611	1.206 (0.026)	**1.2729**	1.208 (0.026)	**1.2729**
0.1	0.2	1.151 (0.277)	0.2884	1.213 (0.540)	0.2898	1.210 (0.526)	0.2900	1.210 (0.528)	**0.2902**	1.201 (0.483)	**0.2902**
0	0.2	1.268 (13.638)	0.0727	1.237 (10.540)	**0.0731**	1.232 (10.062)	**0.0731**	1.232 (10.061)	**0.0731**	1.232 (9.992)	**0.0731**
-0.1	0.2	1.148 (0.121)	0.4261	1.217 (0.245)	0.4393	1.223 (0.257)	0.4396	1.222 (0.255)	**0.4397**	1.226 (0.264)	0.4395
-0.3	0.2	1.245 (0.059)	1.0076	1.175 (0.028)	1.0368	1.159 (0.023)	1.0440	1.154 (0.021)	**1.0462**	1.154 (0.021)	**1.0462**
-0.5	0.2	1.270 (0.027)	1.6224	1.172 (0.010)	1.6881	1.161 (0.009)	1.6962	1.151 (0.007)	**1.7040**	1.151 (0.007)	**1.7040**

Table 5: Out-of-sample performance of the $q-$Learning algorithm (Algorithm 3) in terms of mean, variance and Sharpe ratio. Here, $H = 40$ "real" asset price paths and H_s synthetic price paths are used in each learning episode of Algorithm 3.

Improvement (%)	$H_s : 0 \Rightarrow 1$	$H_s : 0 \Rightarrow 10$	$H_s : 0 \Rightarrow 40$	$H_s : 0 \Rightarrow 360$
Maximum	37.8995	52.9680	**54.3378**	53.8812
Minimum	0	0	0	0
Median	2.9979	3.7002	**3.8316**	**3.8316**
Average	4.9482	6.7627	**7.2161**	7.1945

Table 6: Sharpe ratio improvement for $q-$learning policy trained by Algorithm 3 when different numbers of synthetic price paths are added in each learning episode.

- Even adding one single synthetic path per learning episode ($H_s = 1$) yields a notable improvement, with an average Sharpe ratio increase of approximately 5% across the 21 parameter settings.
- The most significant performance gain is observed when $H_s = 40$, with an average Sharpe ratio improvement of around 7.21% over the case of pure "real" paths. As indicated in Table 5, this improvement is largely attributed to reduced variance of terminal wealth with similar mean return when synthetic paths are included in training.
- Increasing H_s to 360 does not provide additional benefits, while doubling the training time compared to the case of $H_s = 40$. This suggests that, in practice, a moderate number of synthetic paths (e.g. on the same order of the number of training paths) is sufficient to train effective investment policies using Algorithm 3.

5.2 Empirical Study

We now conduct an empirical study to demonstrate how our generative approach can be useful practically in portfolio optimization, despite the fact that real financial time series may not be discrete observations of SDEs and may not even be Markovian.

We still consider the mean–variance portfolio selection problem in (18), with S&P 500 index as the risky asset and a riskless asset with interest rate $r = 2\%$. The investment horizon is set to be half a year in the empirical study. A classical model-based solution to (18) is to use historical data of S&P 500 to estimate a GBM model and then plug the estimated model parameters in the analytical expression of the optimal deterministic policy of (18), which is known from Zhou and Li (2000). The resulting policy (called *plug-in policy*) has been shown to be inferior compared with the model-free *RL policy* proposed by Jia and Zhou (2023).

We now describe the setup of our empirical study. We split daily observations of S&P 500 data from 1990-2009 into 40 half-year trajectories (henceforth called *split paths*), and normalize each split path to start from 1. These paths are used to train our generative model. As in the SDE setting, we simulate the daily increment of S&P 500, and autoregressively generate a synthetic index path for half a year. These synthetic paths can be pooled together with the original 40 split paths to

either estimate the GBM coefficients (for plug-in policies) or train RL policies (similar to that in Section 5.1). We than test the resulting policies on S&P 500 data out-of-sample from 2010 to 2019.

More specifically, we test the plug-in policy when only the 40 split paths are used and when the 40 split and 40 synthetic paths are combined to estimate the parameters of GBM. For the RL policy trained via Algorithm 3, we apply the block bootstrap method (Lahiri, 2013, Chapter 2.6) with a block size of 21 trading days to derive 5,000 resulting portfolio wealth trajectories. Moreover, we consider three different training sets in each learning episode: the first one consists of the 40 split paths only, the second one has 40 bootstrap paths as in Jia and Zhou (2023),[7] and the last one is based on the 40 split and 40 synthetic paths.[8]

Target Level	Policy	SDE Paths	Mean	Variance	Sharpe Ratio ↑
$z = 1.10$	Plug-in	Split	1.1610	0.3501	0.2722
		Split + Synthetic	1.1652	0.3704	0.2715
	RL	Split	1.0857	0.0264	0.5273
		Bootstrap	1.0884	0.0264	0.5438
		Split + Synthetic	1.0860	0.0225	**0.5729**
$z = 1.20$	Plug-in	Split	1.3221	1.4004	0.2722
		Split + Synthetic	1.3371	1.4385	0.2810
	RL	Split	1.1745	0.1120	0.5215
		Bootstrap	1.1768	0.1061	0.5427
		Split + Synthetic	1.1704	0.0892	**0.5707**

Table 7: Performances of different portfolio strategies with and without synthetic paths.

We present the results in Table 7, covering two cases of targeted mean terminal wealth levels, $z = 1.10$ and $z = 1.20$. The two cases exhibit qualitatively similar results, so here we discuss only the former case. First, including synthetic paths fails to improve the plug-in policy in terms of the Sharpe ratio. This is consistent with the earlier observation (see Example 1) that generated paths are not helpful in improving the accuracy of the model parameters. Second, RL strategies significantly outperform the plug-in ones by nearly doubling the Sharpe ratio due to massively reduced variances. Finally, adding synthetic paths to the training dataset provides a further boost in performance to the original RL method with an approximately 8% increase in the Sharpe ratio. The grand conclusion is that AI generated data cannot be employed to better understand the unknown environment, but provide more training "playgrounds" to train better decision policies with more robust performances.

[7] Precisely, we construct 40 bootstrap paths by randomly sampling 40 starting points from the 1990-2009 S&P 500 price trajectory, appending six months of subsequent index data to each starting point and normalizing the starting point to 1.

[8] The finding from Section 5.1 suggests that we take the same number of synthetic paths.

6 Conclusions

In this paper we provide a model-free, data-driven approach for generating sample paths of unknown Markovian SDEs via conditional diffusion models. Numerical experiments demonstrate the superiority of our method over two benchmark approaches. Moreover, we establish a KL error bound for our path generation approach, giving a theoretical guarantee on its accuracy. As a concrete application, we make use of the generated sample paths to enhance the performance of RL algorithms in model-free continuous-time mean–variance portfolio selection, illustrating the potential of synthetic paths in creating more diverse scenarios for training decision-making strategies. A key insight from the study is that AI-generated paths do not help model parameter estimation, but enhance decision capabilities.

This work opens the gate for several avenues of future research. First and foremost, it will be important to remove the Markovian assumption for both theoretical and practical reasons. In this case, the conditioning variables will be all the historical time–state pairs, resulting in a much higher dimensional problem. The theoretical error analysis will need to involve *functional* (or path-dependent) SDEs. Next, it will be interesting to study path generations for SDEs with jumps or those driven by fractional Brownian motions, the latter being an instance of non-Markovian equations. Moreover, relaxing the assumptions on the target SDE and establishing error bounds for alternative metrics such as the Wasserstein distance, will further strengthen the theoretical foundation of our method. Another intriguing direction is to investigate the potential of diffusion models in what-if simulation analysis (e.g. what if the volatility increases by 10%) and counterfactual generations. Finally, exploring the use of synthetic data or paths in other decision-making problems may yield more valuable insights. In sum, the application of diffusion models to strategic data generations represents a largely uncharted territory where most exciting research awaits.

References

Ahmad Aghapour, Erhan Bayraktar, and Fengyi Yuan. Solving dynamic portfolio selection problems via score-based diffusion models. *arXiv preprint arXiv:2507.09916*, 2025.

Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

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

Rabi Bhattacharya and Edward C Waymire. *A basic course in probability theory*. Springer, 2016.

Patrick Cattiaux, Giovanni Conforti, Ivan Gentil, and Christian Léonard. Time reversal of diffusion processes under a finite entropy condition. *Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques*, 59(4):1844–1881, 2023.

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

Minshuo Chen, Renyuan Xu, Yumin Xu, and Ruixun Zhang. Diffusion factor models: Generating high-dimensional returns with factor structure. *arXiv preprint arXiv:2504.06566*, 2025.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023b.

Giovanni Conforti, Alain Durmus, and Marta Gentiloni Silveri. Kl convergence guarantees for score diffusion models under minimal data assumptions. *SIAM Journal on Mathematics of Data Science*, 7(1):86–109, 2025.

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

Hengyu Fu, Zhuoran Yang, Mengdi Wang, and Minshuo Chen. Unveil conditional diffusion models with classifier-free guidance: A sharp statistical theory. *arXiv preprint arXiv:2403.11968*, 2024.

Xuefeng Gao, Hoang M Nguyen, and Lingjiong Zhu. Wasserstein convergence guarantees for a general class of score-based generative models. *Journal of Machine Learning Research*, 26(43): 1–54, 2025.

Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

I Gyöngy and T Pröhle. On the approximation of stochastic differential equation and on stroock-varadhan's support theorem. *Computers & Mathematics with Applications*, 19(1):65–70, 1990.

Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, pages 1188–1205, 1986.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

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

Stefano M Iacus et al. *Simulation and inference for stochastic differential equations: with R examples*, volume 486. Springer, 2008.

Zacharia Issa, Blanka Horvath, Maud Lemercier, and Cristopher Salvi. Non-adversarial training of neural sdes with signature kernel scores. *Advances in Neural Information Processing Systems*, 36: 11102–11126, 2023.

Yanwei Jia and Xun Yu Zhou. q-learning in continuous time. *Journal of Machine Learning Research*, 24(161):1–61, 2023.

Ioannis Karatzas and Steven Shreve. *Brownian motion and stochastic calculus*, volume 113. springer, 2014.

Mathieu Kessler, Alexander Lindner, and Michael Sørensen. Statistical methods for stochastic differential equations. *Monographs on Statistics and Applied Probability*, 124:7–12, 2012.

Patrick Kidger, James Foster, Xuechen Li, and Terry J Lyons. Neural sdes as infinite-dimensional gans. In *International Conference on Machine Learning*, pages 5453–5463. PMLR, 2021.

Peter E Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 1999.

Soumendra Nath Lahiri. *Resampling methods for dependent data*. Springer Science & Business Media, 2013.

Gen Li and Yuling Yan. $O(d/T)$ convergence theory for diffusion probabilistic models under minimal assumptions. *International Conference on Learning Representations*, 2025.

Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards non-asymptotic convergence for diffusion-based generative models. In *The Twelfth International Conference on Learning Representations*, 2024.

Shujian Liao, Hao Ni, Marc Sabate-Vidales, Lukasz Szpruch, Magnus Wiese, and Baoren Xiao. Sig-wasserstein gans for conditional time series generation. *Mathematical Finance*, 34(2):622–670, 2024.

Yanfang Liu, Yuan Chen, Dongbin Xiu, and Guannan Zhang. A training-free conditional diffusion model for learning stochastic dynamical systems. *SIAM Journal on Scientific Computing*, 47(5): C1144–C1171, 2025.

Dominique Michel and Etienne Pardoux. An introduction to Malliavin calculus and some of its applications. *Lecture Notes*, 1990.

Andrea Montanari. Sampling, diffusions, and stochastic localization. *arXiv preprint arXiv:2305.10690*, 2023.

Sai Shankar Narasimhan, Shubhankar Agarwal, Oguzhan Akcin, Sujay Sanghavi, and Sandeep P Chinchali. Time weaver: A conditional time series generation model. In *International Conference on Machine Learning*, pages 37293–37320. PMLR, 2024.

Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. Conditional sig-wasserstein gans for time series generation. *Mathematical Finance*, 2023.

David Nualart. *The Malliavin calculus and related topics*. Springer, 2006.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

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

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

D Stroock and SRS Varadhan. On the support of diffusion processes with applications to the strong maximum principle. In *Proceedings of the Berkeley symposium on mathematical statistics and probability*, volume 1, page 333, 1972.

Cheng Yong Tang and Song Xi Chen. Parameter estimation and bias correction for diffusion processes. *Journal of Econometrics*, 149(1):65–81, 2009.

Wenpin Tang and Hanyang Zhao. Contractive diffusion probabilistic models. *arXiv preprint arXiv:2401.13115*, 2024.

Jorino van Rhijn, Cornelis W Oosterlee, Lech A Grzelak, and Shuaiqiang Liu. Monte carlo simulation of sdes using gans. *Japan Journal of Industrial and Applied Mathematics*, 40(3):1359–1390, 2023.

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

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Haoran Wang and Xun Yu Zhou. Continuous-time mean-variance portfolio selection: A reinforcement learning framework. *Mathematical Finance*, 30(4):1273–1308, 2020.

Qing Wang, Sanjeev R Kulkarni, and Sergio Verdú. Divergence estimation for multidimensional densities via k-nearest-neighbor distances. *IEEE Transactions on Information Theory*, 55(5): 2392–2405, 2009.

Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. Quant GANs: deep generation of financial time series. *Quantitative Finance*, 20(9):1419–1440, 2020.

Minglei Yang, Pengjun Wang, Diego del Castillo-Negrete, Yanzhao Cao, and Guannan Zhang. A pseudoreversible normalizing flow for stochastic dynamical systems with various initial distributions. *SIAM Journal on Scientific Computing*, 46(4):C508–C533, 2024.

Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Xun Yu Zhou and Duan Li. Continuous-time mean-variance portfolio selection: A stochastic LQ framework. *Applied Mathematics & Optimization*, 42(1):19–33, 2000.

Aiqing Zhu and Qianxiao Li. Dyngma: a robust approach for learning stochastic differential equations from data. *Journal of Computational Physics*, page 113200, 2024.

A Proof of Proposition 1

Proof of Proposition 1. Throughout this proof, if a probability distribution is absolutely continuous w.r.t. the Lebesgue measure and admits a probability density function that is strictly positive everywhere on $\mathbb{R}^d$, we simply say this distribution is ACL and strictly positive on $\mathbb{R}^d$.

We first consider the target paths. For any $t_n \in \mathcal{T}$, denote by $P_{n+1|n}(\cdot|\mathbf{x}(t_n))$ the conditional distribution of $\mathbf{X}(t_{n+1})$ given $\mathbf{X}(t_n) = \mathbf{x}(t_n)$, which is ACL and strictly positive on $\mathbb{R}^d$ according to Assumption 1(b). It follows that the conditional distribution $P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}$ of the increment is also ACL and strictly positive on $\mathbb{R}^d$. Therefore, for any $A \subset \mathbb{R}^d$,

$$\int_{\mathbf{x}\in A} 1 \mathrm{d}\mathbf{x} = 0 \quad \text{if and only if} \quad \int_{\mathbf{x}\in A} 1 \cdot \frac{\mathrm{d}P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}(\mathbf{x})}{\mathrm{d}\mathbf{x}} \mathrm{d}\mathbf{x} = 0, \tag{19}$$

where $\frac{\mathrm{d}P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}(\mathbf{x})}{\mathrm{d}\mathbf{x}}$ is the Radon–Nikodym derivative (i.e., the density function) of $P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}(\mathbf{x})$ w.r.t. the Lebesgue measure. This implies that $P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}$ and the Lebesgue measure are equivalent.

Next, consider $P_{1:N_T|0}$. For any $0 \le m < n \le N_T$, let $\mathbf{X}(m:n) := (\mathbf{X}(t_m), \mathbf{X}(t_{m+1}), \cdots, \mathbf{X}(t_n))$, and denote by $P_{m:n|0}$ the joint distribution of $\mathbf{X}(m:n)$ conditional on $\mathbf{X}(t_0) \equiv \mathbf{x}(t_0)$. Let $\mathbf{x}(m:n) = (\mathbf{x}(t_m), \cdots, \mathbf{x}(t_n))$ be a realization of $\mathbf{X}(m:n)$. Due to the Markov property of the Target SDE (1), we have

$$\begin{aligned}
\mathrm{d}P_{m:n|0}(\mathbf{x}(m:n)) &= \mathrm{d}\left[P_{m|0}(\mathbf{x}(t_m)) \prod_{j=m}^{n-1} P\big(\mathbf{X}(t_{j+1}) = \mathbf{x}(t_{j+1}) | \mathbf{X}(m:j) = \mathbf{x}(m:j)\big)\right] \\
&= \mathrm{d}\left[P_{m|0}(\mathbf{x}(t_m)) \prod_{j=m}^{n-1} P\big(\mathbf{X}(t_{j+1}) = \mathbf{x}(t_{j+1}) | \mathbf{X}(t_j) = \mathbf{x}(t_j)\big)\right] \\
&= \mathrm{d}P_{m|0}(\mathbf{x}(t_m)) \prod_{j=m}^{n-1} \mathrm{d}P_{j+1|j}\big(\mathbf{x}(t_{j+1})|\mathbf{x}(t_j)\big),
\end{aligned} \tag{20}$$

where we recall that $P_{m|0}(\mathbf{x}(t_m))$ and $P_{j+1|j}\big(\mathbf{x}(t_{j+1})|\mathbf{x}(t_j)\big)$ are ACL and strictly positive on $\mathbb{R}^d$ for any m and j because of Assumption 1(b). Hence, $P_{m:n|0}$ is ACL and strictly positive on $\mathbb{R}^{d\times(n-m+1)}$ for any $m < n$. In particular, $P_{1:N_T|0}$ is ACL and strictly positive on $\mathbb{R}^{d\times N_T}$. Using a similar argument as in (19), we deduce that $P_{1:N_T|0}$ is equivalent to the Lebesgue measure.

On the other hand, consider the distribution of the generated increment $\hat{P}_{\Delta\hat{\mathbf{X}}(t_n;\mathbf{x}(t_n))}$ for a given t_n and $\mathbf{x}(t_n)$. Note that $\Delta\hat{\mathbf{X}}(t_n;\mathbf{x}(t_n))$ is the terminal state (i.e. $\tilde{\mathbf{Z}}(\tau_K)$) at the last step of the generation process (6), which means it is the sum of a random variable determined by the state of the second last step of (6) (i.e. $\tilde{\mathbf{Z}}(\tau_{K-1})$) and an independent Gaussian variable (with mean 0 and variance $g^2(T_g - \tau_{K-1})\Delta\tau \cdot I_d$). It follows that $\hat{P}_{\Delta\hat{\mathbf{X}}(t_n;\mathbf{x}(t_n))}$ is ACL and strictly positive on $\mathbb{R}^d$ by Corollary 2.2 of Bhattacharya and Waymire (2016), and consequently also equivalent to the Lebesgue measure for each n.

By the autoregressive generation (8), the conditional distribution $\hat{P}_{n+1|n}(\cdot|\hat{\mathbf{x}}(t_n))$ of $\hat{\mathbf{X}}(t_{n+1})$ conditional on $\hat{\mathbf{X}}(t_n) = \hat{\mathbf{x}}(t_n)$ is ACL and strictly positive on $\mathbb{R}^d$ for each n. Using similar arguments leading to (19) and (20), we conclude that the distribution $\hat{P}_{1:N_T|0}$ of the entire generated path $\big(\hat{\mathbf{X}}(t_1, \cdots, \hat{\mathbf{X}}(t_{N_T})\big)$ is equivalent to the Lebesgue measure and strictly positive on $\mathbb{R}^{d\times N_T}$.

Finally, we establish the decomposition result (12) for the KL-divergence between the target and generated paths. Note that given $\mathbf{X}(t_0) \equiv \mathbf{x}(t_0)$, the Markov property holds for both $\mathbf{X}(1:N_T)$ and $\hat{\mathbf{X}}(1:N_T)$. Then, the chain rule of KL-divergence (Wainwright, 2019, Exercise 3.2) implies that[9]

$$\begin{aligned}
\mathrm{KL}\big(P_{1:N_T|0}||\hat{P}_{1:N_T|0}\big) &= \sum_{n=0}^{N_T-1} \mathbb{E}_{\mathbf{x}(0:n)\sim\mathbf{X}(0:n)|\mathbf{X}(t_0)}\Big[\mathrm{KL}\big(P_{\mathbf{X}(t_{n+1})|\mathbf{X}(0:n)=\mathbf{x}(0:n)}\big|\big|\hat{P}_{\hat{\mathbf{X}}(t_{n+1})|\hat{\mathbf{X}}(0:n)=\mathbf{x}(0:n)}\Big]\\
&= \sum_{n=0}^{N_T-1} \mathbb{E}_{\mathbf{x}(0:n)\sim\mathbf{X}(0:n)|\mathbf{X}(t_0)}\Big[\mathrm{KL}\big(P_{\mathbf{X}(t_{n+1})|\mathbf{X}(t_n)=\mathbf{x}(t_n)}\big|\big|\hat{P}_{\hat{\mathbf{X}}(t_{n+1})|\hat{\mathbf{X}}(t_n)=\mathbf{x}(t_n)}\Big]\\
&= \sum_{n=0}^{N_T-1} \mathbb{E}_{\mathbf{x}(t_n)\sim\mathbf{X}(t_n)|\mathbf{X}(t_0)}\Big[\mathrm{KL}\big(P_{\Delta\mathbf{X}(t_n;\mathbf{x}(t_n))}\big|\big|\hat{P}_{\Delta\hat{\mathbf{X}}(t_n;\mathbf{x}(t_n))}\Big]\\
&= \sum_{n=0}^{N_T-1} \mathbb{E}_{\mathbf{X}(t_n)\sim P_{n|0}}\Big[\mathrm{KL}\big(P_{\Delta\mathbf{X}(t_n;\mathbf{X}(t_n))}\big|\big|\hat{P}_{\Delta\hat{\mathbf{X}}(t_n;\mathbf{X}(t_n))}\big)\Big].
\end{aligned}$$

This completes the proof. □

B Proof of Proposition 2

We first define some constants that will be used throughout the proof. The forward equation (3) admits an analytical solution

$$\begin{aligned}
\mathbf{Y}(s) &= e^{-\int_\tau^s f(v)\mathrm{d}v}\mathbf{Y}(\tau) + \int_\tau^s e^{-\int_u^s f(v)\mathrm{d}v} g(u)\mathrm{d}\mathbf{B}(u)\\
&\overset{d}{=} e^{-\frac{a}{4}(s^2-\tau^2)-\frac{b}{2}(s-\tau)}\mathbf{Y}(\tau) + \sqrt{1-e^{-\frac{a}{2}(s^2-\tau^2)-b(s-\tau)}}\cdot\zeta,
\end{aligned} \tag{21}$$

for any $0 < \tau < s < T_g$, where $\zeta \sim \mathcal{N}(0, I_d)$ is independent of $\mathbf{Y}(\tau)$. Let

$$\lambda_{\tau,s} := e^{-\frac{a}{4}(s^2-\tau^2)-\frac{b}{2}(s-\tau)}, \quad \lambda_\tau := \lambda_{0,\tau}, \quad \sigma^2_{\tau,s} := 1 - e^{-\frac{a}{2}(s^2-\tau^2)-b(s-\tau)}, \quad \text{and} \quad \sigma^2_\tau = \sigma^2_{0,\tau}. \tag{22}$$

We then have $\mathbf{Y}(s)|\mathbf{Y}(\tau) \sim \mathcal{N}(\lambda_{\tau,s}\mathbf{Y}(\tau), \sigma^2_{\tau,s} I_d)$, and we denote by $q_{s|\tau}(\cdot|\mathbf{y}, \mathbf{c})$ the corresponding Gaussian density when $\mathbf{Y}(\tau) = \mathbf{y}$ under the conditional $\mathbf{c}$.

The proof of Proposition 2 is long, accomplished through several lemmas. For reader's convenience, Figure 5 provides the relations and logical flows of various lemmas in proving Proposition 2.

[9] A proof of the chain rule can be found in this note.

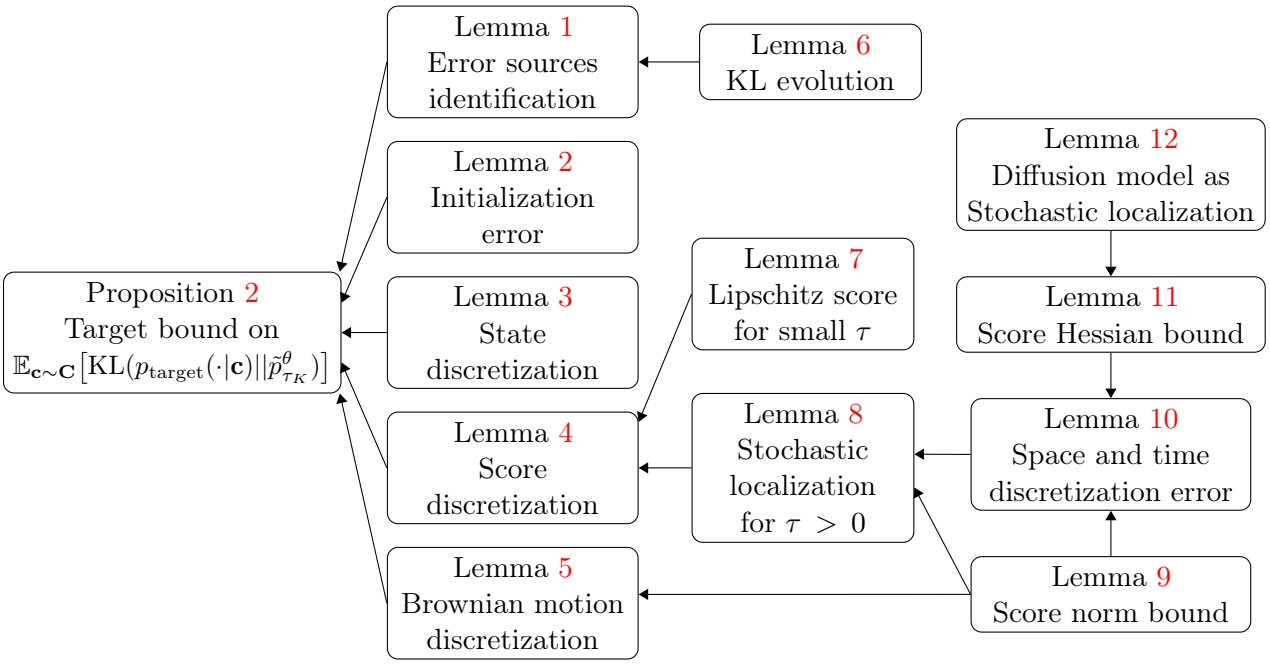

Figure 5: Roadmap of Proving Proposition 2

. Given a fixed condition $\mathbf{c}$, we first state Lemma 1 below, which decomposes $\mathrm{KL}\big(p_{\mathrm{target}}(\cdot|\mathbf{c})||\tilde{p}^{\theta}_{\tau_K}(\cdot|\mathbf{c})\big)$ and identifies various sources of approximation errors.

Lemma 1. *For a given condition $\mathbf{c}$, denote by $\tilde{p}^{\theta}_{\tau_K}(\cdot|\mathbf{c})$ the marginal density of the generated $\tilde{\mathbf{Z}}(\tau_K)$ of the reverse process (6). Under the conditions of Proposition 2, we have*

$$
\begin{aligned}
&KL\big(p_{target}(\cdot|\mathbf{c})||\tilde{p}^{\theta}_{\tau_K}(\cdot|\mathbf{c})\big)\\
&\lesssim KL\big(q_{T_g}(\cdot|\mathbf{c})||\mathcal{N}(0,I_d)\big)+\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k}\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\big[||f(\tau)\mathbf{Y}(\tau)-f(\tau_k)\mathbf{Y}(\tau_k)||^2\big]\mathrm{d}\tau\\
&\quad+\sum_{k=1}^{K}\int_{\tau_{n-1}}^{\tau_k}\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\big[||g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})-g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})||^2\big]\mathrm{d}\tau\\
&\quad+\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k}[g^2(\tau_k)-g^2(\tau)]^2\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\big[\big\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\big\|^2\big]\mathrm{d}\tau\\
&\quad+\sum_{k=1}^{K}\Big[\big(\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2]+d\big)\cdot\mathbb{E}_{\mathbf{Y}(\tau_k)\sim q_{\tau_k}(\cdot|\mathbf{c})}\big\|\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})-s_\theta(\tau_k,\mathbf{Y}(\tau_k),\mathbf{c})\big\|^2\Big]^{1/2}
\end{aligned}
$$

$$+\Delta\tau\sum_{k=1}^{K}\mathbb{E}_{\mathbf{Y}\sim q_{\tau_k}(\cdot|\mathbf{c})}\big[||g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})-g^2(\tau_k)s_\theta(\tau_k,\mathbf{Y}(\tau_k),\mathbf{c})||^2\big]. \tag{23}$$

The proof of Lemma 1 can be found in Appendix B.1. We next discuss how to bound each term on the right-hand-side of (23).

The first term of (23) is the initialization error arising from the use of the Gaussian noise, instead of the terminal state of the forward process, as the initialization of the generation process (6). We bound this initialization error in Lemma 2, leveraging on the exponential convergence property of the forward OU process (3). The proof of Lemma 2 is given in Appendix B.2.

Lemma 2. *For the terminal marginal density $q_{T_g}(\cdot|\mathbf{c})$ induced by the forward process defined in (3), if the moment condition (14) is satisfied, then we have:*

$$\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\Big[KL\big(q_{T_g}(\cdot|\mathbf{c})||\mathcal{N}(0,I_d)\big)\Big]\lesssim e^{-\frac{a}{2}T_g^2-bT_g}\big[M_2+de^{-\frac{a}{2}T_g^2-bT_g}\big].$$

Next, we bound the second term of (23), which accounts for the state discretization error of using $f(\tau_k)\mathbf{Y}(\tau_k)$ to approximate $f(\tau)\mathbf{Y}(\tau)$ for $\tau\in[\tau_{k-1},\tau_k]$. Lemma 3 shows that this error can be controlled under the finite moment condition (14). The proof, deferred to Appendix B.3, is based on applying Itô's formula to $f(\tau)\mathbf{Y}(\tau)$.

Lemma 3. *For the forward process $\mathbf{Y}$ in (3), if the moment condition (14) is satisfied, then we have:*

$$\begin{aligned}&\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\Bigg[\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k}\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\big[||f(\tau)\mathbf{Y}(\tau)-f(\tau_k)\mathbf{Y}(\tau_k)||^2\big]\mathrm{d}\tau\Bigg]\\&\lesssim\Delta\tau(M_2+d)(T_g+1)^5.\end{aligned}$$

The third error term in (23) represents the time and space discretization error for approximating the score function $g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})$ by $g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})$ when $\tau\in[\tau_{k-1},\tau_k]$. We bound this error in Lemma 4. Our proof is inspired by that of Theorem 5 of Chen et al. (2023a), but we improve their dependence on the data dimension d by applying the stochastic localization technique introduced in Benton et al. (2024); see Appendix B.4 for the complete proof.

Lemma 4. *Suppose the first condition of Proposition 2 holds. Then for the forward process $\{\mathbf{Y}(\tau):\tau\in[0,T_g]\}$ in (3) and the corresponding score function $\{\nabla\log q_\tau(\cdot|\mathbf{c}):\tau\in[0,T_g]\}$, we have*

$$\begin{aligned}&\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\Bigg[\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k}\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\big[\big\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})-g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\big\|\big]^2\mathrm{d}\tau\Bigg]\\&\lesssim M_2\Delta\tau(T_g+1)^2+d\Delta\tau\big[L_1+(T_g+1)^4\big].\end{aligned}$$

The fourth term of (23) is the discretization error of simulating the Brownian motion $g(\tau)\mathrm{d}\mathbf{B}(\tau)$ in (4) by $g(\tau_k)\sqrt{\Delta\tau}\zeta_k \overset{d}{=} g(\tau_k)\mathrm{d}\mathbf{B}(\tau)$ from τ_k to τ_{k+1} in (6). We provide a bound on this error in Lemma 5, whose proof is given in Appendix B.5.

Lemma 5. *Under the same condition as in Lemma 4, we have*

$$\begin{aligned}&\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\Bigg[\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k}[g^2(\tau_k)-g^2(\tau)]^2\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\left[\left\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\right\|^2\right]\mathrm{d}\tau\Bigg]\\&\lesssim d(\Delta\tau)^2[L_1+T_g+1].\end{aligned}$$

The last two terms of (23) capture the score matching error out of the neural network approximation of the true score function. Given conditions (14) and (15), we have

$$\begin{aligned}&\sum_{k=1}^{K}\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\Bigg[\left(\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2]+d\right)\cdot\mathbb{E}_{\mathbf{Y}(\tau_k)\sim q_{\tau_k}(\cdot|\mathbf{c})}\left\|\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})-s_\theta(\tau_k,\mathbf{Y}(\tau_k),\mathbf{c})\right\|^2\Bigg]^{1/2}\\&\quad+\Delta\tau\sum_{k=1}^{K}\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\left[\mathbb{E}_{\mathbf{Y}\sim q_{\tau_k}(\cdot|\mathbf{c})}\left[||g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})-g^2(\tau_k)s_\theta(\tau_k,\mathbf{Y}(\tau_k),\mathbf{c})||^2\right]\right]\\&\le\sum_{k=1}^{K}\Bigg[\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\left(\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2]+d\right)\cdot\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\mathbb{E}_{\mathbf{Y}(\tau_k)\sim q_{\tau_k}(\cdot|\mathbf{c})}\left\|\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})-s_\theta(\tau_k,\mathbf{Y}(\tau_k),\mathbf{c})\right\|^2\Bigg]^{1/2}\\&\quad+\Delta\tau\sum_{k=1}^{K}g^4(\tau_k)\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\left[\mathbb{E}_{\mathbf{Y}\sim q_{\tau_k}(\cdot|\mathbf{c})}\left[||\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})-s_\theta(\tau_k,\mathbf{Y}(\tau_k),\mathbf{c})||^2\right]\right]\\&\le\sqrt{M_2+d}\cdot K\varepsilon_{\text{score}}+T_gK\varepsilon_{\text{score}}^2,\end{aligned}\tag{24}$$

where, in the last step, we use the fact that $\sum_{k=1}^{K}c_k\le(\sum_{k=1}^{K}\sqrt{c_k})^2$ for any $c_1,\cdots,c_k\ge0$.

Finally, by combining the results of Lemma 1 through Lemma 5, along with the bound (24) on the score matching error, we obtain the error bound in (16), proving Proposition 2. □

In the following five subsections, we present the proofs of Lemmas 1 through 5.

B.1 Proof of Lemma 1

To prove Lemma 1, we first consider the reverse SDE (4) and the discrete simulation (6) when $\tau\in(\tau_k,\tau_{k+1}]$. Given $\tilde{\mathbf{Y}}(\tau_k)=\tilde{\mathbf{z}}\in\mathbb{R}^d$, rewrite (4) as

$$\mathrm{d}\tilde{\mathbf{Y}}(\tau)=[\tilde{f}(\tau)\tilde{\mathbf{Y}}(\tau)+\tilde{g}^2(\tau)\nabla\log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})]\mathrm{d}\tau+\tilde{g}(\tau)\mathrm{d}\tilde{\mathbf{B}}(\tau),\tag{25}$$

where $\tilde{f}(\tau):=f(T_g-\tau)$ and $\tilde{g}(\tau):=g(T_g-\tau)$. On

$$(\tau_k,\tau_{k+1}]$$

, write (6) starting from $\tilde{\mathbf{Z}}(\tau_k) = \tilde{\mathbf{z}}$ as

$$\mathrm{d}\tilde{\mathbf{Z}}(\tau) = [\tilde{f}(\tau_k)\tilde{\mathbf{Z}}(\tau_k) + \tilde{g}^2(\tau_k)s_\theta(T_g - \tau_k, \tilde{\mathbf{Z}}(\tau_k), \mathbf{c})]\mathrm{d}\tau + \tilde{g}(\tau_k)\mathrm{d}\tilde{\mathbf{B}}(\tau), \tag{26}$$

where, for notational simplicity, we use the same Brownian motion $\tilde{\mathbf{B}}$ as we only care about distributions. To facilitate presentation, we introduce additional notations summarized in Table 8:

Dynamics	State	Marginal Distribution (τ)	Conditional Distribution ($\tau < s$)	Comment
Forward SDE (3)	$\mathbf{Y}(\tau)$	$q_\tau(\cdot\|\mathbf{c})$	$q_{s\|\tau}(\cdot\|\mathbf{c})$	$q_0(\cdot\|\mathbf{c}) = p_{\text{target}}(\cdot\|\mathbf{c})$; $q_\tau(\cdot\|\mathbf{c}) = \tilde{p}_{T_g-\tau}$
Reverse SDE (25)	$\tilde{\mathbf{Y}}(\tau)$	$\tilde{p}_\tau$	$\tilde{p}_{s\|\tau}$	omit $\mathbf{c}$ for simplicity; $\tilde{p}_\tau = q_{T_g-\tau}(\cdot\|\mathbf{c})$
Simulated SDE (26)	$\tilde{\mathbf{Z}}(\tau)$	$\tilde{p}^\theta_\tau$	$\tilde{p}^\theta_{s\|\tau}$	omit $\mathbf{c}$ for simplicity

Table 8: Notations for the proof of Lemma 1.

The key ingredient of the proof is the following Lemma 6, which characterizes the KL-divergence between conditional densities of the two Itô processes, (25) and (26), with different drift and diffusion coefficients. Note that (Chen et al., 2023a, Lemma 6, Lemma 7(2)) provide also KL-divergence characterizations, which however are for the case where two processes share the same diffusion coefficients and not applicable to our setting. For the same reason, we cannot apply the Girsanov theorem to characterize the KL-divergence between two path measures as in Benton et al. (2024). The proof of Lemma 6 is provided in Appendix C.1.

Lemma 6. *Given a condition* $\mathbf{c}$*, consider the conditional densities* $\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})$ *and* $\tilde{p}^\theta_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})$ *induced by* (25) *and* (26)*, respectively. We have the following results.*

(a) If $\tilde{\mathbf{Y}}(\tau_k)$ *follows the marginal density* $\tilde{p}_{\tau_k}$ *of the reverse SDE* (4)*, then*

$$\begin{aligned}
&\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)}\left[\lim_{\tau\to\tau_k+} KL\big(\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))||\tilde{p}^\theta_{\tau|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))\big)\right] \\
&\le 2\left[\big(\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2] + d\big)\cdot \mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)}\big\|\nabla \log q_{T_g-\tau_k}(\tilde{\mathbf{Y}}(\tau_k)|\mathbf{c}) - s_\theta(T_g-\tau_k, \tilde{\mathbf{Y}}(\tau_k), \mathbf{c})\big\|^2\right]^{1/2}.
\end{aligned}$$

(b) For any given initial $\tilde{\mathbf{z}}$ *and* $\tau \in (\tau_k, \tau_{k+1}]$*,*

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\tau} KL\big(\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})||\tilde{p}^\theta_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})\big) \\
&\le \frac{1}{C_{KL}}\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big\|\tilde{f}(\tau)\tilde{\mathbf{Y}}(\tau) - \tilde{f}(\tau_k)\tilde{\mathbf{z}} + \tilde{g}^2(\tau)\nabla\log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) - \tilde{g}^2(\tau_k)s_\theta(T_g-\tau_k, \tilde{\mathbf{z}}, \mathbf{c})\big\|^2
\end{aligned}$$

$$+\frac{[\tilde{g}^2(\tau_k)-\tilde{g}^2(\tau)]^2}{4C_{KL}}\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\big\|\nabla\log\tilde{p}_\tau(\tilde{\mathbf{Y}}(\tau))\big\|^2\big]$$
$$+\frac{1}{2}\Big[C_{KL}-\tilde{g}^2(\tau_k)\Big]\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\bigg\|\nabla\log\frac{\tilde{p}_{\tau|\tau_k}(\tilde{\mathbf{Y}}(\tau)|\tilde{\mathbf{z}})}{\tilde{p}^\theta_{\tau|\tau_k}(\tilde{\mathbf{Y}}(\tau)|\tilde{\mathbf{z}})}\bigg\|^2, \tag{27}$$

for any fixed constant $C_{KL}>0$, where the expectation is taken w.r.t. the conditional density $\tilde{\mathbf{Y}}(\tau)\sim\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})$ of the reverse SDE (25) .

Now, we are ready to prove Lemma 1.

Proof of Lemma 1. We apply the result of Lemma 6(b) and set $C_{KL}=b$. Note that $\tilde{g}^2(\tau)=g^2(T_g-\tau)=a(T_g-\tau)+b\geq b$ for any $\tau\in[0,T_g]$. Then, integrating on both sides of (27) from τ_k to τ_{k+1}, we have

$$\begin{aligned}
&\mathrm{KL}\big(\tilde{p}_{\tau_{k+1}|\tau_k}(\cdot|\tilde{\mathbf{z}})||\tilde{p}^\theta_{\tau_{k+1}|\tau_k}(\cdot|\tilde{\mathbf{z}})\big)-\lim_{\tau\to\tau_k+}\mathrm{KL}\big(\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})||\tilde{p}^\theta_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})\big)\\
&\leq\frac{1}{b}\int_{\tau_k}^{\tau_{k+1}}\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)\sim\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})}\big\|\tilde{f}(\tau)\tilde{\mathbf{Y}}(\tau)-\tilde{f}(\tau_k)\tilde{\mathbf{z}}+\tilde{g}^2(\tau)\nabla\log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})-\tilde{g}^2(\tau_k)s_\theta(K-k,\tilde{\mathbf{z}},\mathbf{c})\big\|^2\mathrm{d}\tau\\
&\quad+\frac{1}{4b}\int_{\tau_k}^{\tau_{k+1}}[\tilde{g}^2(\tau_k)-\tilde{g}^2(\tau)]^2\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)\sim\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})}\big[\big\|\nabla\log\tilde{p}_\tau(\tilde{\mathbf{Y}}(\tau))\big\|^2\big]\mathrm{d}\tau.
\end{aligned}$$

Since the above result holds when $\tilde{\mathbf{Y}}(\tau_k)=\tilde{\mathbf{Z}}(\tau_k)=\tilde{\mathbf{z}}$ for any fixed $\tilde{\mathbf{z}}\in\mathbb{R}^d$, we can integrate on both sides of the above inequality w.r.t. $\tilde{\mathbf{z}}=\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}$. Then, by Lemma 6(a), we have

$$\begin{aligned}
&\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\big[\mathrm{KL}\big(\tilde{p}_{\tau_{k+1}|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))||\tilde{p}^\theta_{\tau_{k+1}|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))\big)\big]\\
&\leq 2\bigg[\big(\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2]+d\big)\cdot\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)}\big\|\nabla\log q_{T_g-\tau_k}(\tilde{\mathbf{Y}}(\tau_k)|\mathbf{c})-s_\theta(T_g-\tau_k,\tilde{\mathbf{Y}}(\tau_k),\mathbf{c})\big\|^2\bigg]^{1/2}\\
&\quad+\frac{1}{b}\int_{\tau_k}^{\tau_{k+1}}\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau_k),\tilde{\mathbf{Y}}(\tau))}\Big[\big\|\tilde{g}^2(\tau)\nabla\log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})-\tilde{g}^2(\tau_k)s_\theta(T_g-\tau_k,\tilde{\mathbf{Y}}(\tau_k),\mathbf{c})\\
&\qquad\qquad+\tilde{f}(\tau)\tilde{\mathbf{Y}}(\tau)-\tilde{f}(\tau_k)\tilde{\mathbf{Y}}(\tau_k)\big\|^2\Big]\mathrm{d}\tau\\
&\quad+\frac{1}{4b}\int_{\tau_k}^{\tau_{k+1}}[\tilde{g}^2(\tau_k)-\tilde{g}^2(\tau)]^2\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\big\|\nabla\log\tilde{p}_\tau(\tilde{\mathbf{Y}}(\tau))\big\|^2\big]\mathrm{d}\tau,
\end{aligned}\tag{28}$$

where the expectation is taken w.r.t. $\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}$ and $\tilde{\mathbf{Y}}(\tau)\sim\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))$.

On the other hand, consider the reverse SDE (25) starting from $\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}$, and the simulated SDE (26) starting from $\tilde{\mathbf{Z}}(\tau_k)\sim\tilde{p}^\theta_{t_k}$. Using the chain rule of KL divergence, we have the following result:

$$\mathrm{KL}\big(\tilde{p}_{\tau_{k+1}}||\tilde{p}^\theta_{\tau_{k+1}}\big)\leq\mathrm{KL}\big(\tilde{p}_{\tau_k}||\tilde{p}^\theta_{\tau_k}\big)+\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\big[\mathrm{KL}\big(\tilde{p}_{\tau_{k+1}|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))||\tilde{p}^\theta_{\tau_{k+1}|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))\big)\big]. \tag{29}$$

With (28) and (29), we have

$$
\begin{aligned}
&\mathrm{KL}(p_{\text{target}}(\cdot|\mathbf{c})||\tilde{p}^{\theta}_{\tau_K}(\cdot|\mathbf{c})) = \mathrm{KL}(q_0(\cdot|\mathbf{c})||\tilde{p}^{\theta}_{\tau_K}(\cdot|\mathbf{c})) = \mathrm{KL}\big(\tilde{p}_{T_g}||\tilde{p}^{\theta}_{\tau_K}\big) = \mathrm{KL}\big(\tilde{p}_{\tau_K}||\tilde{p}^{\theta}_{\tau_K}\big) \\
&\overset{(29)}{\leq} \mathrm{KL}\big(\tilde{p}_0||\tilde{p}^{\theta}_0\big) + \sum_{k=0}^{K-1} \mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\big[\mathrm{KL}\big(\tilde{p}_{\tau_{k+1}|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))||\tilde{p}^{\theta}_{\tau_{k+1}|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))\big)\big] \\
&\overset{(28)}{\leq} \mathrm{KL}\big(q_{T_g}(\cdot|\mathbf{c})||\mathcal{N}(0,I_d)\big) + \frac{1}{4b}\sum_{k=0}^{K-1}\int_{\tau_k}^{\tau_{k+1}} [\tilde{g}^2(\tau_k)-\tilde{g}^2(\tau)]^2 \mathbb{E}_{\tilde{\mathbf{Y}}(\tau)\sim\tilde{p}_\tau}\big[\big\|\nabla\log\tilde{p}_\tau(\tilde{\mathbf{Y}}(\tau))\big\|^2\big]\mathrm{d}\tau \\
&\quad + \frac{1}{b}\sum_{k=0}^{K-1}\int_{\tau_k}^{\tau_{k+1}} \mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k},\tilde{\mathbf{Y}}(\tau)\sim\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))}\Big[\big\|\tilde{g}^2(\tau)\nabla\log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) - \tilde{g}^2(\tau_k)s_\theta(K-k,\tilde{\mathbf{Y}}(\tau_k),\mathbf{c}) \\
&\qquad\qquad + \tilde{f}(\tau)\tilde{\mathbf{Y}}(\tau) - \tilde{f}(\tau_k)\tilde{\mathbf{Y}}(\tau_k)\big\|^2\Big]\mathrm{d}\tau \\
&\quad + 2\sum_{k=0}^{K-1}\Big[\big(\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2]+d\big)\cdot\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)}\big\|\nabla\log q_{T_g-\tau_k}(\tilde{\mathbf{Y}}(\tau_k)|\mathbf{c}) - s_\theta(T_g-\tau_k,\tilde{\mathbf{Y}}(\tau_k),\mathbf{c})\big\|^2\Big]^{1/2} \\
&\leq \mathrm{KL}\big(q_{T_g}(\cdot|\mathbf{c})||\mathcal{N}(0,I_d)\big) + \frac{1}{4b}\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k} [g^2(\tau_k)-g^2(\tau)]^2 \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\big[\big\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\big\|^2\big]\mathrm{d}\tau \\
&\quad + \frac{1}{b}\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k} \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\Big[\big\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_k)s_\theta(k,\mathbf{Y}(\tau_k),\mathbf{c}) \\
&\qquad\qquad + f(\tau)\mathbf{Y}(\tau) - f(\tau_k)\mathbf{Y}(\tau_k)\big\|^2\Big]\mathrm{d}\tau \\
&\quad + 2\sum_{k=1}^{K}\Big[\big(\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2]+d\big)\cdot\mathbb{E}_{\mathbf{Y}(\tau_k)\sim q_{\tau_k}(\cdot|\mathbf{c})}\big\|\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c}) - s_\theta(\tau_k,\mathbf{Y}(\tau_k),\mathbf{c})\big\|^2\Big]^{1/2} \\
&\lesssim \mathrm{KL}\big(q_{T_g}(\cdot|\mathbf{c})||\mathcal{N}(0,I_d)\big) + \sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k} \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\big[\big\|f(\tau_k)\mathbf{Y}(\tau_k) - f(\tau)\mathbf{Y}(\tau)\big\|^2\big]\mathrm{d}\tau \\
&\quad + \sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k} \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\big[\big\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\big\|\big]^2\mathrm{d}\tau \\
&\quad + \sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k} [g^2(\tau_k)-g^2(\tau)]^2 \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\big[\big\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\big\|^2\big]\mathrm{d}\tau \\
&\quad + \sum_{k=1}^{K}\Big[\big(\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2]+d\big)\cdot\mathbb{E}_{\mathbf{Y}(\tau_k)\sim q_{\tau_k}(\cdot|\mathbf{c})}\big\|\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c}) - s_\theta(\tau_k,\mathbf{Y}(\tau_k),\mathbf{c})\big\|^2\Big]^{1/2}
\end{aligned}
$$

$$+\Delta\tau\sum_{k=1}^{K}\mathbb{E}_{\mathbf{Y}(\tau_k)\sim q_{\tau_k}(\cdot|\mathbf{c})}\left\|g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})-g^2(\tau_k)s_\theta(k,\mathbf{Y}(\tau_k),\mathbf{c})\right\|^2,$$

where the last step is obtained by adding and subtracting a term of $g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})$, and using the fact that $\|\mathbf{x}+\mathbf{y}+\mathbf{z}\|^2\le 9\|\mathbf{x}\|^2+9\|\mathbf{y}\|^2+9\|\mathbf{z}\|^2$. The proof is hence completed. □

B.2 Proof of Lemma 2

Proof. We adapt the proof of Proposition 4 of Benton et al. (2024) to our setting where the forward OU process is time-inhomogeneous. Recall that $q_{\tau|0}(\cdot|\mathbf{y}(0),\mathbf{c})$ is the conditional distribution of $\mathbf{Y}(\tau)$ given $\mathbf{Y}(0)=\mathbf{y}(0)$ under condition $\mathbf{c}$, and $q_{\tau|0}(\cdot|\mathbf{y}(0),\mathbf{c})\sim\mathcal{N}(\lambda_\tau\mathbf{Y}(0),\sigma_\tau^2 I_d)$ from (21) and (22). Then, one can directly compute that

$$\mathrm{KL}(q_{\tau|0}(\cdot|\mathbf{y}(0),\mathbf{c})||\mathcal{N}(0,I_d))=\frac{1}{2}\big(d\log\sigma_\tau^{-2}-d+d\sigma_\tau^2+\|\lambda_\tau\mathbf{y}(0)\|^2\big).$$

By the convexity of the KL divergence and Jensen's inequality, we have

$$\begin{aligned}\mathrm{KL}\big(q_{T_g}(\cdot|\mathbf{c})||\mathcal{N}(0,I_d)\big)&=\mathrm{KL}\bigg(\int_{\mathbf{y}(0)\in\mathbb{R}^d}q_{T_g|0}(\cdot|\mathbf{y}(0),\mathbf{c})\mathrm{d}q_0(\mathbf{y}(0)|\mathbf{c})\bigg|\bigg|\mathcal{N}(0,I_d)\bigg)\\&\le\int_{\mathbf{y}(0)\in\mathbb{R}^d}\mathrm{KL}\big(q_{T_g|0}(\cdot|\mathbf{y}(0),\mathbf{c})||\mathcal{N}(0,I_d)\big)\mathrm{d}q_0(\mathbf{y}(0)|\mathbf{c})\\&=\frac{1}{2}\bigg(d\log\sigma_{T_g}^{-2}-d+d\sigma_{T_g}^2+\lambda_{T_g}^2\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}\|\mathbf{Y}(0)\|^2\bigg)\\&\lesssim de^{-aT_g^2-2bT_g}+e^{-\frac{a}{2}T_g^2-bT_g}\big(\mathbb{E}_{\mathbf{Y}(0)\sim p_{\text{target}}(\cdot|\mathbf{c})}\|\mathbf{Y}(0)\|^2\big)\end{aligned}$$

where we use (22) and the approximation that $-\ln(1-x)\lesssim x+\frac{x^2}{2}$ for $x\in(-1,1)$. Applying condition (14), we immediately obtain

$$\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\Big[\mathrm{KL}\big(q_{T_g}(\cdot|\mathbf{c})||\mathcal{N}(0,I_d)\big)\Big]\lesssim e^{-\frac{a}{2}T_g^2-bT_g}\big[M_2+de^{-\frac{a}{2}T_g^2-bT_g}\big].$$

This completes the proof of Lemma 2. □

B.3 Proof of Lemma 3

Proof of Lemma 3. By Itô's formula, with $\mathbf{Y}(\tau)$ following the the forward SDE (3), we have

$$\mathrm{d}(f(\tau)\mathbf{Y}(\tau))=f'(\tau)\mathbf{Y}(\tau)\mathrm{d}\tau+f(\tau)\mathrm{d}\mathbf{Y}(\tau)=\bigg[\frac{a}{2}-\frac{1}{4}(a\tau+b)^2\bigg]\mathbf{Y}(\tau)\mathrm{d}\tau+\frac{1}{2}(a\tau+b)^{\frac{3}{2}}\mathrm{d}\mathbf{B}(\tau).$$

Let $u(\tau) := \frac{a}{2} - \frac{1}{4}(a\tau + b)^2$ and $v(\tau) := \frac{1}{2}(a\tau + b)^{\frac{3}{2}}$. Then

$$\begin{aligned}
&\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\left[||f(\tau)\mathbf{Y}(\tau) - f(\tau_k)\mathbf{Y}(\tau_k)||^2\right] \\
&= \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),(\mathbf{B})}\left[\left|\left|\int_\tau^{\tau_k} u(s)\mathbf{Y}(s)\mathrm{d}s + \int_\tau^{\tau_k} v(s)\mathrm{d}\mathbf{B}_s\right|\right|^2\right] \\
&\lesssim \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),(\mathbf{B})}\left[\left|\left|\int_\tau^{\tau_k} u(s)\mathbf{Y}(s)\mathrm{d}s\right|\right|^2 + \left|\left|\int_\tau^{\tau_k} v(s)\mathrm{d}\mathbf{B}_s\right|\right|^2\right] \\
&\lesssim d\int_\tau^{\tau_k} v^2(s)\mathrm{d}s + \left(\int_\tau^{\tau_k} u^2(s)\mathrm{d}s\right)\left(\int_\tau^{\tau_k} \mathbb{E}_{\mathbf{Y}(s)\sim q_s(\cdot|\mathbf{c})}[||\mathbf{Y}(s)||^2]\mathrm{d}s\right),
\end{aligned}$$

where the last inequality follows from the Cauchy–Schwarz inequality. Note that $\mathbf{Y}(s)|\mathbf{Y}(0) \sim \mathcal{N}(\lambda_s\mathbf{Y}(0), \sigma_s^2 I_d)$ for $\mathbf{Y}(0) \sim p_{\text{target}}(\cdot|\mathbf{c})$. Since $\lambda_s^2, \sigma_s^2 \le 1$ for any $s \in [0, T_g]$, it follows that

$$\mathbb{E}_{\mathbf{Y}(s)\sim q_s(\cdot|\mathbf{c})}[||\mathbf{Y}(s)||^2] \le \mathbb{E}_{\mathbf{Y}(0)\sim p_{\text{target}}(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2] + d.$$

Meanwhile, given that $\int_\tau^{\tau_k} u^2(s)\mathrm{d}s \lesssim \int_\tau^{\tau_k}(s+1)^4\mathrm{d}s \lesssim (\Delta\tau)(\tau+1)^4$ and $\int_\tau^{\tau_k} v^2(s)\mathrm{d}s \lesssim (\Delta\tau)(\tau+1)^3$, we derive

$$\begin{aligned}
&\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\left[\sum_{k=1}^K \int_{\tau_{k-1}}^{\tau_k} \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\left[||f(\tau)\mathbf{Y}(\tau) - f(\tau_k)\mathbf{Y}(\tau_k)||^2\right]\mathrm{d}\tau\right] \\
&\lesssim \sum_{k=1}^K \int_{\tau_{k-1}}^{\tau_k} \left[d(\Delta\tau)(\tau+1)^3 + \Delta\tau(M_2+d)(\tau+1)^4\right]\mathrm{d}\tau \\
&\lesssim \Delta\tau(M_2+d)\int_0^{T_g}(\tau+1)^4\mathrm{d}\tau \\
&\lesssim \Delta\tau(M_2+d)(T_g+1)^5,
\end{aligned}$$

which completes the proof. □

B.4 Proof of Lemma 4

Lemma 4 bounds the (cumulative) time and space discretization error from approximating the score function $g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})$ by $g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})$ for $\tau \in [\tau_{k-1}, \tau_k]$. To establish this bound, we consider the case when τ is small and when τ is large separately.

Define an index threshold:

$$k_1 := \min\{0 \le k \le K : \tau_k \ge \min\{1, T_g/2\}\}. \tag{30}$$

With k_1, we adopt the following approximation for σ_τ^2 throughout the proof in this subsection,

$$\sigma_\tau^2 = 1 - \exp(-a\tau^2/2 - b\tau) \asymp \begin{cases} a\tau^2/2 + b\tau, & \text{for } \tau < \tau_{k_1}, \\ 1, & \text{otherwise.} \end{cases} \tag{31}$$

The following result bounds the discretization error when τ is small, based on the Lipschitz property of $\nabla \log q_0(\cdot|\mathbf{c})$. In particular, it can be shown that the score $\nabla \log q_\tau(\cdot|\mathbf{c})$ is still Lipschitz for small τ, an important property that will be used in the proof. The proof of Lemma 7 is inspired by Chen et al. (2023a) and is deferred to Appendix B.4.1.

Lemma 7. *Assume $\nabla \log p_{target}(\cdot|\mathbf{c})$ is $L(\mathbf{c})$-Lipschitz continuous for a given condition $\mathbf{c}$. If for some $1 \le k \le k_1$, there exists $s \in [\tau_{k-1}, \tau_k]$ satisfying $\sigma_s^2 \le \frac{\lambda_s}{2L(\mathbf{c})}$, then*

$$\int_{\tau_{k-1}}^{s} \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\left[\|g^2(\tau)\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_k)\nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\right] \mathrm{d}\tau$$
$$\lesssim d(\Delta\tau)^2 L(\mathbf{c})(L(\mathbf{c})+1)(\tau_k+1)^5.$$

However, the condition $\sigma_\tau^2 \le \lambda_\tau/(2L(\mathbf{c}))$ may not hold for large τ, and the score function is no longer Lipschitz continuous. To get around, we apply the stochastic localization technique introduced in Benton et al. (2024) to control the error (which improves the technique in Chen et al., 2023a). In particular, we have the following result, whose proof is given in Appendix B.4.2.

Lemma 8. *Given a condition $\mathbf{c}$, for any $1 \le k_0 \le k_1$ and any $s \in (\tau_{k_0-1}, \tau_{k_0}]$, we have*

$$\int_{s}^{\tau_{k_0}} \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_{k_0})\sim q_{\tau_{k_0}|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\left[\|g^2(\tau)\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_{k_0})\nabla \log q_{\tau_{k_0}}(\mathbf{Y}(\tau_{k_0})|\mathbf{c})\|^2\right] \mathrm{d}\tau$$
$$+ \sum_{k=k_0+1}^{K} \int_{\tau_{k-1}}^{\tau_k} \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\left[\|g^2(\tau)\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_k)\nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\right] \mathrm{d}\tau$$
$$\lesssim \Delta\tau(T_g+1)^2 \mathbb{E}_{\mathbf{Y}(0)\sim p_{target}(\cdot|\mathbf{c})}\left[\|\mathbf{Y}(0)\|^2\right] + d\Delta\tau\left[\ln\left(\frac{1}{s}\right) + \frac{1}{s} + (T_g+1)^4\right].$$

Now, we are ready to prove Lemma 4.

Proof of Lemma 4. From (22), we note that $\lambda_\tau \sigma_\tau^{-2}$ is decreasing in τ, with $\lim_{\tau\to 0+} \lambda_\tau\sigma_\tau^{-2} = +\infty$ and $\lim_{\tau\to\infty} \lambda_\tau\sigma_\tau^{-2} = 0$. Hence, for the given condition $\mathbf{c}$, there must exist τ_0 such that $\lambda_{\tau_0}\sigma_{\tau_0}^{-2} = 2L(\mathbf{c})$. This allows us to apply Lemma 7 for $\tau < \min\{\tau_0, \tau_{k_1}\}$. However, when $L(\mathbf{c})$ is small, τ_0 can be large. To control the magnitude of (the sum of) $(\tau_k+1)^5$ in Lemma 7, we introduce a larger Lipschitz constant

$$L_2(\mathbf{c}) := L(\mathbf{c}) + \lambda_{\min\{1,T_g/2\}}\sigma_{\min\{1,T_g/2\}}^{-2}. \tag{32}$$

It is clear that $\nabla \log q_0(\cdot|\mathbf{c})$ is also $L_2(\mathbf{c})$-Lipschitz continuous. We can then define

$$\tau_{lipz}(\mathbf{c}) = \max\left\{0 \le \tau \le T_g : \lambda_\tau\sigma_\tau^{-2} \ge 2L_2(\mathbf{c})\right\}.$$

Note that $\lambda_{\min\{1,T_g/2\}}\sigma_{\min\{1,T_g/2\}}^{-2} \le 2L_2(\mathbf{c})$ and $\lambda_\tau\sigma_\tau^{-2}$ is decreasing in τ. Thus, for any given and fixed condition $\mathbf{c}$, $\tau_{lipz}(\mathbf{c}) < \min\{1, T_g/2\} \le \tau_{k_1}$. Let $k_0(\mathbf{c}) = \min\{1 \le k \le K : \tau_k \ge \tau_{lipz}(\mathbf{c})\} \le k_1$.

We can now apply Lemma 7 with settings of $s = \tau_1, \cdots, \tau_{k_0(\mathbf{c})-1}$ and $s = \tau_{lipz}(\mathbf{c})$. Then, for $\mathbf{Y}(\tau) \sim q_\tau(\cdot|\mathbf{c}), \mathbf{Y}(\tau_k) \sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau), \mathbf{c})$,

$$
\begin{aligned}
&\sum_{k=1}^{k_0(\mathbf{c})-1} \int_{\tau_{k-1}}^{\tau_k} \mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_k))}\big[\|g^2(\tau)\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_k)\nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big]\mathrm{d}\tau \\
&\quad + \int_{\tau_{k_0(\mathbf{c})-1}}^{\tau_{lipz}(\mathbf{c})} \mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_{k_0(\mathbf{c})}))}\big[\|g^2(\tau)\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_{k_0(\mathbf{c})})\nabla \log q_{\tau_{k_0(\mathbf{c})}}(\mathbf{Y}(\tau_{k_0(\mathbf{c})})|\mathbf{c})\|^2\big]\mathrm{d}\tau \\
&\lesssim d(\Delta\tau)^2 L_2(\mathbf{c})(L_2(\mathbf{c})+1)\Big[\sum_{k=1}^{k_0(\mathbf{c})}(\tau_k+1)^5\Big] \\
&\lesssim d(\Delta\tau)^2 L_2(\mathbf{c})(L_2(\mathbf{c})+1)k_0(\mathbf{c}),
\end{aligned}
$$

where we use the fact that $\tau_{k_0(\mathbf{c})} \le \tau_{k_1} \le 1 + \Delta\tau$. Note that when $\tau = \tau_{lipz}(\mathbf{c}) \le 1$, we can apply (31) to obtain

$$
2L_2(\mathbf{c}) \asymp \frac{\lambda_\tau}{\sigma_\tau^2} = \frac{e^{-\frac{a}{4}\tau^2 - \frac{b}{2}\tau}}{1 - e^{-\frac{a}{2}\tau^2 - b\tau}} \asymp \frac{1 - a\tau^2/4 - b\tau/2}{a\tau^2/2 + b\tau} = \frac{1}{a\tau^2/2 + b\tau} - \frac{1}{2} \asymp \frac{1}{\tau} - \frac{1}{2},
$$

yielding

$$
\tau_{lipz}(\mathbf{c}) \asymp \frac{2}{4L_2(\mathbf{c})+1}, \quad k_0(\mathbf{c}) \asymp \frac{2}{(4L_2(\mathbf{c})+1)\Delta\tau}. \tag{33}
$$

Therefore,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\Bigg[\sum_{k=1}^{k_0(\mathbf{c})-1} \int_{\tau_{k-1}}^{\tau_k} \mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_k))}\big[\|g^2(\tau)\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_k)\nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big]\mathrm{d}\tau\Bigg] \\
&\quad + \mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\Bigg[\int_{\tau_{k_0(\mathbf{c})-1}}^{\tau_{lipz}(\mathbf{c})} \mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_{k_0(\mathbf{c})}))}\big[\|g^2(\tau)\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_{k_0(\mathbf{c})})\nabla \log q_{\tau_{k_0(\mathbf{c})}}(\mathbf{Y}(\tau_{k_0(\mathbf{c})})|\mathbf{c})\|^2\big]\mathrm{d}\tau\Bigg] \\
&\lesssim \mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\big[d(\Delta\tau)^2 L_2(\mathbf{c})(L_2(\mathbf{c})+1)k_0(\mathbf{c})\big] \\
&\overset{(33)}{\lesssim} d\Delta\tau\,\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\big[L_2(\mathbf{c})+1\big] \\
&= d\Delta\tau\,\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\big[L(\mathbf{c}) + \lambda_{\min\{1,T_g/2\}}\sigma^{-2}_{\min\{1,T_g/2\}} + 1\big] \\
&\lesssim d\Delta\tau(L_1+1),
\end{aligned} \tag{34}
$$

where L_1 is defined in (13), and the last step follows from the approximation that

$$
\lambda_{\min\{1,T_g/2\}}\sigma^{-2}_{\min\{1,T_g/2\}} \asymp \frac{1}{a(\min\{1,T_g/2\})^2/2 + b\min\{1,T_g/2\}} - \frac{1}{2} \asymp 1. \tag{35}
$$

On the other hand, for those $k \geq k_0(\mathbf{c})$, Lemma 8 implies that

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\bigg[\int_{\tau_{lipz}(\mathbf{c})}^{\tau_{k_0(\mathbf{c})}} \mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_{k_0(\mathbf{c})}))}\big[\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_{k_0(\mathbf{c})})\nabla\log q_{\tau_{k_0(\mathbf{c})}}(\mathbf{Y}(\tau_{k_0(\mathbf{c})})|\mathbf{c})\|^2\big]\mathrm{d}\tau\bigg] \\
&\quad + \mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\bigg[\sum_{k=k_0(\mathbf{c})+1}^{K}\int_{\tau_{k-1}}^{\tau_k} \mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_k))}\big[\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big]\mathrm{d}\tau\bigg] \\
&\lesssim \Delta\tau(T_g+1)^2\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\big[\mathbb{E}_{\mathbf{Y}(0)\sim p_{\text{target}}(\cdot|\mathbf{c})}\big[\|\mathbf{Y}(0)\|^2\big]\big] + d\Delta\tau\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\bigg[\ln\bigg(\frac{1}{\tau_{lipz}(\mathbf{c})}\bigg) + \frac{1}{\tau_{lipz}(\mathbf{c})} + (T_g+1)^4\bigg] \\
&\overset{(33)}{\lesssim} M_2\Delta\tau(T_g+1)^2 + d\Delta\tau\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\big[\ln(2L_2(\mathbf{c})+1/2) + L_2(\mathbf{c}) + (T_g+1)^4\big] \\
&\lesssim M_2\Delta\tau(T_g+1)^2 + d\Delta\tau\big[L_1 + (T_g+1)^4\big], \qquad (36)
\end{aligned}
$$

where in the last step we use the fact that $\ln(x+1/2) \leq x+1/2$ for any $x \geq 0$ and the approximation in (35). The proof is complete by combining (34) and (36). □

B.4.1 Proof of Lemma 7

Proof of Lemma 7. Given a condition $\mathbf{c}$, we can compute, for $\tau < \tau_k$, $\mathbf{Y}(\tau) \sim q_\tau(\cdot|\mathbf{c})$ and $\mathbf{Y}(\tau_k) \sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})$,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\big[\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big] \\
&\leq 2\mathbb{E}_{\mathbf{Y}(\tau)}\big[\|(g^2(\tau)-g^2(\tau_k))\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\|^2\big] \\
&\qquad + 2\mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_k))}\big[g^4(\tau_k)\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - \nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big] \\
&\overset{\text{(i)}}{\leq} 2\big[\big(g^2(\tau)-g^2(\tau_k)\big)^2 + 2\big(\lambda_{\tau,\tau_k}^{-1}-1\big)^2\big]\mathbb{E}_{\mathbf{Y}(\tau)}\big[\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\|^2\big] \\
&\qquad + 4\mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_k))}\big[g^4(\tau_k)\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - \nabla\log q_\tau(\lambda_{\tau,\tau_k}^{-1}\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big] \\
&\overset{\text{(ii)}}{\lesssim} (\tau_k-\tau)^2(\tau_k+1)^2\mathbb{E}_{\mathbf{Y}(\tau)}\big[\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\|^2\big] \\
&\qquad + (\tau_k+1)^2\mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_k))}\big[\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - \nabla\log q_\tau(\lambda_{\tau,\tau_k}^{-1}\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big] \qquad (37)
\end{aligned}
$$

where the inequality (i) is from Lemma 11 of Chen et al. (2023a), and the inequality (ii) employs the approximation that for small $\tau_k - \tau$,

$$
\lambda_{\tau,\tau_k}^{-1} - 1 = e^{\frac{a}{4}(\tau_k^2-\tau^2)+\frac{b}{2}(\tau_k-\tau)} - 1 \lesssim \frac{a}{4}(\tau_k^2-\tau^2) + \frac{b}{2}(\tau_k-\tau) \lesssim (\tau_k-\tau)(\tau_k+1).
$$

Next, from the condition that $\sigma_s^2 \leq \frac{\lambda_s}{2L(\mathbf{c})}$ and the fact that $\lambda_\tau\sigma_\tau^{-2}$ is decreasing in τ, we know that $\sigma_\tau^2 \leq \frac{\lambda_\tau}{2L(\mathbf{c})}$ for all $\tau \leq s$, indicating that Lemma 14 of Chen et al. (2023a) holds and $\nabla\log q_\tau(\cdot|\mathbf{c})$ is $(2L(\mathbf{c})\lambda_\tau^{-1})$-Lipschitz on $\mathbb{R}^d$ for all $\tau \leq s$. Then,

$$
\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}[\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - \nabla\log q_\tau(\lambda_{\tau,\tau_k}^{-1}\mathbf{Y}(\tau_k)|\mathbf{c})\|^2]
$$

$$
\begin{aligned}
&\le 4L^2(\mathbf{c})\lambda_\tau^{-2}\mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_k))}\big[\|\mathbf{Y}(\tau)-\lambda_{\tau,\tau_k}^{-1}\mathbf{Y}(\tau_k)\|^2\big]\\
&= 4dL^2(\mathbf{c})\lambda_\tau^{-2}\lambda_{\tau,\tau_k}^{-2}\sigma_{\tau,\tau_k}^2 = 4dL^2(\mathbf{c})e^{\frac{a}{2}\tau_k^2+b\tau_k}\big[1-e^{-\frac{a}{2}(\tau_k^2-\tau^2)-b(\tau_k-\tau)}\big]\\
&\overset{(31)}{\lesssim} dL^2(\mathbf{c})(1+a\tau_k^2/2+b\tau_k)\cdot(\tau_k-\tau)(a\tau_k+b)\\
&\lesssim dL^2(\mathbf{c})(\tau_k-\tau)(\tau_k+1)^3,
\end{aligned}
$$

where we use the condition that $k \le k_1$ so that we can apply (31) to approximate $(\lambda_\tau\lambda_{\tau,\tau_k})^{-2}$. Next, the Lipschitz continuity of $\nabla \log q_\tau(\cdot|\mathbf{c})$ implies that $\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\|\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\|^2 \le 2dL(\mathbf{c})\lambda_\tau^{-1}$ as in Lemma 21 of Chen et al. (2023a). Hence, we can infer from (37) that

$$
\begin{aligned}
&\int_{\tau_{k-1}}^{s}\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\big[\|g^2(\tau)\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})-g^2(\tau_k)\nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big]\mathrm{d}\tau\\
&\lesssim \int_{\tau_{k-1}}^{s}(\tau_k-\tau)^2(\tau_k+1)^2dL(\mathbf{c})\lambda_\tau^{-1}+dL^2(\mathbf{c})(\tau_k-\tau)(\tau_k+1)^5\mathrm{d}\tau\\
&\lesssim \int_{\tau_{k-1}}^{s}(\tau_k-\tau)^2(\tau_k+1)^4dL(\mathbf{c})+dL^2(\mathbf{c})(\tau_k-\tau)(\tau_k+1)^5\mathrm{d}\tau\\
&\lesssim d(\Delta\tau)^2L(\mathbf{c})(L(\mathbf{c})+1)(\tau_k+1)^5,
\end{aligned}
$$

where the second inequality is obtained by applying (31) to approximate λ_τ^{-1}. The proof is complete. □

B.4.2 Proof of Lemma 8

The following Lemma 9 follows immediately from Lemmas 20 and 21 in Chen et al. (2023a).

Lemma 9. *For any given condition* $\mathbf{c}$ *and* $\tau \in (0, T_g]$*, and the marginal density* $q_\tau(\cdot|\mathbf{c})$ *defined by the forward process* (3)*, we have*

$$
\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\|\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\|^2 \le \frac{d}{\sigma_\tau^2},
$$

where σ_τ^2 *is the conditional variance defined in* (22)*.*

We next apply the stochastic localization technique in Benton et al. (2024) (see Lemmas 3 to 6 therein) to derive a bound on the space and time discretization error of the score function in Lemma 10 below. This result is the key to the proof of Lemma 8, whose proof is placed at the end of this subsection due to its length.

Lemma 10. *For any given condition* $\mathbf{c}$ *and* $\tau \in (0, T_g]$*, let* $q_{0|\tau}(\cdot|\mathbf{y}(\tau), \mathbf{c})$ *be the conditional distribution of* $\mathbf{Y}(0)$ *given* $\mathbf{c}$ *and* $\mathbf{Y}(\tau) = \mathbf{y}(\tau)$*, and* $\mathbf{\Sigma}_\tau(\mathbf{y}(\tau), \mathbf{c}) = Cov_{\mathbf{Y}(0)\sim q_{0|\tau}(\cdot|\mathbf{y}(\tau),\mathbf{c})}\big(\mathbf{Y}(0)\big)$ *be the corresponding conditional covariance matrix. Then we have for any* $\tau \in [\tau_{k-1}, \tau_k]$

$$
\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_k)\sim q_{\tau_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\big[\|\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})-\nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big]
$$

$$\leq \int_{\tau}^{\tau_k} f(s)d\sigma_{\tau_k}^{-2} + g^2(s)d\sigma_s^{-4}\mathrm{d}s$$
$$+ \sigma_{\tau_k}^{-4}\mathbb{E}_{\mathbf{Y}(\tau_k)\sim q_{\tau_k}(\cdot|\mathbf{c})}[trace(\mathbf{\Sigma}_{\tau_k}(\mathbf{Y}(\tau_k),\mathbf{c}))] - \sigma_{\tau}^{-4}\mathbb{E}_{\mathbf{Y}(\tau)\sim q_{\tau}(\cdot|\mathbf{c})}[trace(\mathbf{\Sigma}_{\tau}(\mathbf{Y}(\tau),\mathbf{c}))].$$

Now, we prove Lemma 8.

Proof of Lemma 8. For notational simplicity, we introduce

$$\tau'_{k_0-1} = s,\ \tau'_k = \tau_k,\ \text{for } k = k_0, \cdots, K,$$

which means that $\tau'_k - \tau'_{k-1} \leq \Delta\tau$ for any $k = k_0, \cdots, K$.

Now, given the condition $\mathbf{c}$, note that for $\mathbf{Y}(\tau) \sim q_\tau(\cdot|\mathbf{c})$ and $\mathbf{Y}(\tau'_k) \sim q_{\tau'_k|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})$,

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau'_k))}\left[\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau'_k)\nabla\log q_{\tau'_k}(\mathbf{Y}(\tau'_k)|\mathbf{c})\|^2\right]\\
&\leq 2\mathbb{E}_{\mathbf{Y}(\tau)}\left[\|(g^2(\tau) - g^2(\tau'_k))\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\|^2\right]\\
&\quad + 2\mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau'_k))}\left[g^4(\tau'_k)\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - \nabla\log q_{\tau'_k}(\mathbf{Y}(\tau'_k)|\mathbf{c})\|^2\right]\\
&\lesssim d(\Delta\tau)^2\sigma_\tau^{-2} + \mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau'_k))}\left[g^4(\tau'_k)\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - \nabla\log q_{\tau'_k}(\mathbf{Y}(\tau'_k)|\mathbf{c})\|^2\right],
\end{aligned}$$

where the last inequality is from Lemma 9. Then, since $\tau'_k = \tau_k$ for $k \geq k_0$, by Lemma 10 we have

$$\begin{aligned}
&\int_s^{\tau_{k_0}} \mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c}),\mathbf{Y}(\tau_{k_0})\sim q_{\tau_{k_0}|\tau}(\cdot|\mathbf{Y}(\tau),\mathbf{c})}\left[\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_{k_0})\nabla\log q_{\tau_{k_0}}(\mathbf{Y}(\tau_{k_0})|\mathbf{c})\|^2\right]\mathrm{d}\tau\\
&\quad + \sum_{k=k_0+1}^{K}\int_{\tau_{k-1}}^{\tau_k} \mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_k))}\left[\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau_k)\nabla\log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\right]\mathrm{d}\tau\\
&= \sum_{k=k_0}^{K}\int_{\tau'_{k-1}}^{\tau'_k} \mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau'_k))}\left[\|g^2(\tau)\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - g^2(\tau'_k)\nabla\log q_{\tau'_k}(\mathbf{Y}(\tau'_k)|\mathbf{c})\|^2\right]\mathrm{d}\tau\\
&\lesssim \sum_{k=k_0}^{K}\int_{\tau'_{k-1}}^{\tau'_k}\left[d(\Delta\tau)^2\sigma_\tau^{-2} + \int_\tau^{\tau'_k} g^4(\tau'_k)f(s)d\sigma_{\tau'_k}^{-2} + g^4(\tau'_k)g^2(s)d\sigma_s^{-4}\mathrm{d}s\right]\mathrm{d}\tau \qquad (38)\\
&\quad + \sum_{k=k_0}^{K}\int_{\tau'_{k-1}}^{\tau'_k} g^4(\tau'_k)\left(\sigma_{\tau'_k}^{-4}\mathbb{E}_{\mathbf{Y}(\tau'_k)}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_k})] - \sigma_\tau^{-4}\mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)]\right)\mathrm{d}\tau, \qquad (39)
\end{aligned}$$

where we omit the dependency of $\mathbf{\Sigma}_\tau$ on $\mathbf{Y}(\tau)$ and $\mathbf{c}$ for simplicity since the context is clear.

We first bound (38). Recall that $f(\tau) = \frac{1}{2}(a\tau + b), g(\tau) = \sqrt{a\tau + b}$. In addition, σ_τ^{-2} is decreasing in τ from (22) and we have its approximation based on (31). Then

$$\sum_{k=k_0}^{K}\int_{\tau'_{k-1}}^{\tau'_k}\left[d(\Delta\tau)^2\sigma_\tau^{-2} + \int_\tau^{\tau'_k} g^4(\tau'_k)f(s)d\sigma_{\tau'_k}^{-2} + g^4(\tau'_k)g^2(s)d\sigma_s^{-4}\mathrm{d}s\right]\mathrm{d}\tau$$

$$
\begin{aligned}
&\lesssim \sum_{k=k_0}^{K} \int_{\tau'_{k-1}}^{\tau'_k} \left[d(\Delta\tau)^2 \sigma^{-2}_{\tau'_{k-1}} + \int_{\tau}^{\tau'_k} g^4(\tau'_k) f(\tau'_k) d\sigma^{-2}_{\tau'_{k-1}} + g^4(\tau'_k) g^2(\tau'_k) d\sigma^{-4}_{\tau'_{k-1}} \, \mathrm{d}s \right] \mathrm{d}\tau \\
&\lesssim \sum_{k=k_0}^{k_1} \left[\frac{d(\Delta\tau)^3}{a(\tau'_{k-1})^2/2 + b\tau'_{k-1}} + \frac{d(\Delta\tau)^2 (a\tau'_k + b)^3}{a(\tau'_{k-1})^2/2 + b\tau'_{k-1}} + \frac{d(\Delta\tau)^2 (a\tau'_k + b)^3}{[a(\tau'_{k-1})^2/2 + b\tau'_{k-1}]^2} \right] + \sum_{k_1+1}^{K} d(\Delta\tau)^2 (a\tau'_k + b)^3 \\
&\lesssim d\Delta\tau \int_{\tau'_{k_0-1}}^{\tau'_{k_1}} \left[1 + \tau + \frac{1}{\tau} + \frac{1}{\tau^2} \right] \mathrm{d}\tau + d\Delta\tau \int_{\tau'_{k_1}}^{T_g} (\tau+1)^3 \mathrm{d}\tau \\
&\lesssim d\Delta\tau \left[\ln\left(\frac{1}{\tau'_{k_0-1}} \right) + \frac{1}{\tau'_{k_0-1}} + (T_g+1)^4 \right] \\
&= d\Delta\tau \left[\ln\left(\frac{1}{s} \right) + \frac{1}{s} + (T_g+1)^4 \right]. \qquad (40)
\end{aligned}
$$

We next bound (39). Noting that both σ_τ^2 and $\mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)]$ are increasing in τ (see (61) and the discussion thereafter), we rearrange the summation to have

$$
\begin{aligned}
&\sum_{k=k_0}^{K} \int_{\tau'_{k-1}}^{\tau'_k} g^4(\tau'_k) \big(\sigma^{-4}_{\tau'_k} \mathbb{E}_{\mathbf{Y}(\tau'_k)}[\mathrm{trace}(\mathbf{\Sigma}'_{\tau_k})] - \sigma_\tau^{-4} \mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)] \big) \mathrm{d}\tau \\
&\le \sum_{k=k_0}^{K} g^4(\tau'_k) \Delta\tau \big(\sigma^{-4}_{\tau'_k} \mathbb{E}_{\mathbf{Y}(\tau'_k)}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_k})] - \sigma^{-4}_{\tau'_k} \mathbb{E}_{\mathbf{Y}(\tau'_{k-1})}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_{k-1}})] \big) \\
&\lesssim \sum_{k=k_0-1}^{k_1} \Delta\tau \mathbb{E}_{\mathbf{Y}(\tau'_k)}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_k})] \big(g^4(\tau'_k) \sigma^{-4}_{\tau'_k} - g^4(\tau'_{k+1}) \sigma^{-4}_{\tau'_{k+1}} \big) \\
&\quad + g^4(T_g) \sigma^{-4}_{\tau'_{k_1}} \Delta\tau \mathbb{E}_{\mathbf{Y}(\tau'_{k_1+1})}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_{k_1+1}})] \\
&\quad + g^4(T_g) \sigma^{-4}_{\tau'_{k_1}} \Delta\tau \left[\sum_{k=k_1+2}^{K} \big(\mathbb{E}_{\mathbf{Y}(\tau'_k)}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_k})] - \mathbb{E}_{\mathbf{Y}(\tau'_{k-1})}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_{k-1}})] \big) \right] \\
&\overset{(31)}{\lesssim} \Delta\tau (T_g+1)^2 \mathbb{E}_{\mathbf{Y}(T_g)}[\mathrm{trace}(\mathbf{\Sigma}_{T_g})] \\
&\quad + \sum_{k=k_0-1}^{k_1} \Delta\tau \mathbb{E}_{\mathbf{Y}(\tau'_k)}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_k})] \left[\frac{(a\tau'_k + b)^2}{[a(\tau'_k)^2/2 + b\tau'_k]^2} - \frac{(a\tau'_{k+1} + b)^2}{[a(\tau'_{k+1})^2/2 + b\tau'_{k+1}]^2} \right] \\
&\lesssim \Delta\tau (T_g+1)^2 \mathbb{E}_{\mathbf{Y}(T_g)}[\mathrm{trace}(\mathbf{\Sigma}_{T_g})] + \sum_{k=k_0-1}^{k_1} \Delta\tau \mathbb{E}_{\mathbf{Y}(\tau'_k)}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_k})] \cdot \left[\frac{1}{(\tau'_k)^2} - \frac{1}{(\tau'_{k+1})^2} \right] \\
&\lesssim \Delta\tau (T_g+1)^2 \mathbb{E}_{\mathbf{Y}(T_g)}[\mathrm{trace}(\mathbf{\Sigma}_{T_g})] + \sum_{k=k_0-1}^{k_1} \frac{(\Delta\tau)^2}{(\tau'_k)^3} \cdot \mathbb{E}_{\mathbf{Y}(\tau'_k)}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_k})]. \qquad (41)
\end{aligned}
$$

Recall that $\mathbf{\Sigma}_\tau = \mathbf{\Sigma}_\tau(\mathbf{Y}(\tau), \mathbf{c})$ is the conditional variance of $\mathbf{Y}(0)$ given $\mathbf{Y}(\tau)$ and $\mathbf{c}$. Then

$$\mathbb{E}_{\mathbf{Y}(T_g)}[\mathrm{trace}(\mathbf{\Sigma}_{T_g})] \le \mathbb{E}_{\mathbf{Y}(T_g)}\big[\mathbb{E}_{\mathbf{Y}(0)|\mathbf{Y}(T_g)}\|\mathbf{Y}(0)\|^2\big] = \mathbb{E}_{\mathbf{Y}(0)\sim p_{\mathrm{target}}(\cdot|\mathbf{c})}\big[\|\mathbf{Y}(0)\|^2\big]. \tag{42}$$

On the other hand, $\mathbf{Y}(\tau)|\mathbf{Y}(0) \sim \mathcal{N}\big(\lambda_\tau \mathbf{Y}(0), \sigma_\tau^2 I_d\big)$ from (21). Then, for $\tau < \tau'_{k_1} = \tau_{k_1}$ in (30), using a similar approximation as in (31) we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)] &= \mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathrm{trace}\big(\mathrm{Cov}_{\mathbf{Y}(0)|\mathbf{Y}(\tau)}[\mathbf{Y}(0)]\big)\big] \\
&= \lambda_\tau^{-2}\mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathrm{trace}\big(\mathrm{Cov}_{\mathbf{Y}(0)|\mathbf{Y}(\tau)}[\lambda_\tau\mathbf{Y}(0) - \mathbf{Y}(\tau)]\big)\big] \\
&\le \lambda_\tau^{-2}\mathbb{E}_{(\mathbf{Y}(0),\mathbf{Y}(\tau))}\big[\|\lambda_\tau\mathbf{Y}(0) - \mathbf{Y}(\tau)\|^2\big] \\
&= \lambda_\tau^{-2} d\sigma_\tau^2 = d\big(e^{a\tau^2/2+b\tau} - 1\big) \\
&\asymp d(a\tau^2/2 + b\tau).
\end{aligned} \tag{43}$$

Substituting (42) and (43) into (41), we have

$$\begin{aligned}
&\sum_{k=k_0}^{K}\int_{\tau'_{k-1}}^{\tau'_k} g^4(\tau'_k)\big(\sigma_{\tau'_k}^{-4}\mathbb{E}_{\mathbf{Y}(\tau'_k)}[\mathrm{trace}(\mathbf{\Sigma}_{\tau'_k})] - \sigma_\tau^{-4}\mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)]\big)\mathrm{d}\tau \\
&\lesssim \Delta\tau(T_g+1)^2\mathbb{E}_{\mathbf{Y}(0)}\big[\|\mathbf{Y}(0)\|^2\big] + \sum_{k=k_0-1}^{k_1}\frac{(\Delta\tau)^2 d[a(\tau'_k)^2/2 + b\tau'_k]}{(\tau'_k)^3} \\
&\lesssim \Delta\tau(T_g+1)^2\mathbb{E}_{\mathbf{Y}(0)}\big[\|\mathbf{Y}(0)\|^2\big] + d\Delta\tau\int_{\tau'_{k_0-1}}^{\tau'_{k_1}}\left[\frac{1}{\tau} + \frac{1}{\tau^2}\right]\mathrm{d}\tau \\
&\lesssim \Delta\tau(T_g+1)^2\mathbb{E}_{\mathbf{Y}(0)}\big[\|\mathbf{Y}(0)\|^2\big] + d\Delta\tau\left[\ln\left(\frac{1}{\tau'_{k_0-1}}\right) + \frac{1}{\tau'_{k_0-1}}\right] \\
&= \Delta\tau(T_g+1)^2\mathbb{E}_{\mathbf{Y}(0)\sim p_{\mathrm{target}}(\cdot|\mathbf{c})}\big[\|\mathbf{Y}(0)\|^2\big] + d\Delta\tau\left[\ln\left(\frac{1}{s}\right) + \frac{1}{s}\right].
\end{aligned} \tag{44}$$

The proof is complete by combining (40) and (44). □

It remains to prove Lemma 10. To this end, we need a bound on the expectation of $\|\nabla^2 \log q_\tau(\mathbf{Y}_\tau|\mathbf{c})\|_F^2$, which is provided in Lemma 11 below. Its proof, deferred to Appendix C.2, is based on the stochastic localization technique.

Recall that $q_{0|\tau}(\cdot|\mathbf{y}(\tau), \mathbf{c})$ is the conditional distribution of $\mathbf{Y}(0)$ given $\mathbf{c}$ and $\mathbf{Y}(\tau) = \mathbf{y}(\tau)$, and $\mathbf{\Sigma}_\tau(\mathbf{y}(\tau), \mathbf{c}) = \mathrm{Cov}_{\mathbf{Y}(0)\sim q_{0|\tau}(\cdot|\mathbf{y}(\tau),\mathbf{c})}\big(\mathbf{Y}(0)|\mathbf{y}(\tau)\big)$ is the corresponding conditional covariance matrix.

Lemma 11. *For any given condition* $\mathbf{c}$ *and any* $\tau \in (0, T_g]$*, we have*

$$\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\big[\|\nabla^2 \log q_\tau(\mathbf{Y}_\tau|\mathbf{c})\|_F^2\big] \le d\sigma_\tau^{-4} + \frac{1}{g^2(\tau)}\cdot\frac{\mathrm{d}}{\mathrm{d}\tau}\Big(\sigma_\tau^{-4}\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}[\mathit{trace}(\mathbf{\Sigma}_\tau(\mathbf{Y}(\tau), \mathbf{c}))]\Big).$$

Proof of Lemma 10. Following Benton et al. (2024), we first consider the stochastic dynamic of the score function $\{\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) : \tau \in [0, T_g]\}$ under a given condition $\mathbf{c}$. Recall that the reverse SDE (4) of diffusion models is

$$\mathrm{d}\tilde{\mathbf{Y}}(\tau) = [\tilde{f}(\tau)\tilde{\mathbf{Y}}(\tau) + \tilde{g}^2(\tau)\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})]\mathrm{d}\tau + \tilde{g}(\tau)\mathrm{d}\tilde{\mathbf{B}}(\tau),$$

where $\tilde{f}(\tau) := f(T_g - \tau)$ and $\tilde{g}(\tau) := g(T_g - \tau)$. According to Lemma 7(1) of Chen et al. (2023a) and Proposition 1 of Conforti et al. (2025), $q_\tau(\cdot|\mathbf{c})$ is twice continuously differentiable for $\tau \in [0, T_g]$. Hence we can apply Itô's formula to $\nabla \log q_{T_g-\tau}(\cdot|\mathbf{c})$ to obtain

$$\begin{aligned}
\mathrm{d}\big[\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big] = \Big\{ & \nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) \cdot \big[\tilde{f}(\tau)\tilde{\mathbf{Y}}(\tau) + \tilde{g}^2(\tau)\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big] \\
& + \frac{1}{2}\tilde{g}^2(\tau) \cdot \Delta\big(\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big)\Big\}\mathrm{d}\tau + \frac{\mathrm{d}\big[\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big]}{\mathrm{d}\tau}\mathrm{d}\tau \\
& + \tilde{g}(\tau)\nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\mathrm{d}\tilde{\mathbf{B}}(\tau). \qquad (45)
\end{aligned}$$

To further simplify (45), consider the Fokker–Planck equation for the forward process (3):

$$\mathrm{d}q_\tau(\mathbf{y}|\mathbf{c}) = \Big(-\nabla \cdot \big[(-f(\tau)q_\tau(\mathbf{y}|\mathbf{c})\mathbf{y}\big] + \frac{1}{2}\Delta\big[g^2(\tau)q_\tau(\mathbf{y}|\mathbf{c})\big]\Big)\mathrm{d}\tau,$$

leading to

$$\begin{aligned}
\mathrm{d}(\log q_\tau(\mathbf{y}|\mathbf{c})) &= \frac{1}{q_\tau(\mathbf{y}|\mathbf{c})}\Big(-\nabla \cdot \big[(-f(\tau)q_\tau(\mathbf{y}|\mathbf{c})\mathbf{y}\big] + \frac{1}{2}\Delta\big[g^2(\tau)q_\tau(\mathbf{y}|\mathbf{c})\big]\Big)\mathrm{d}\tau \\
&= \Big(f(\tau)d + f(\tau)\mathbf{y}^T\nabla \log q_\tau(\mathbf{y}|\mathbf{c}) + \frac{1}{2}g^2(\tau)\Delta \log q_\tau(\mathbf{y}|\mathbf{c}) + \frac{1}{2}g^2(\tau)\|\nabla \log q_\tau(\mathbf{y}|\mathbf{c})\|^2\Big)\mathrm{d}\tau.
\end{aligned}$$

It follows that

$$\begin{aligned}
&\frac{\mathrm{d}\big[\nabla \log q_{T_g-\tau}(\mathbf{y}|\mathbf{c})\big]}{\mathrm{d}\tau} = \nabla\frac{\mathrm{d}(\log q_{T_g-\tau}(\mathbf{y}|\mathbf{c}))}{\mathrm{d}\tau} \\
&= -\Big(\tilde{f}(\tau)\nabla \log q_{T_g-\tau}(\mathbf{y}|\mathbf{c}) + \tilde{f}(\tau)\nabla^2 \log q_{T_g-\tau}(\mathbf{y}|\mathbf{c}) \cdot \mathbf{y} \qquad (46) \\
&\quad + \frac{1}{2}\tilde{g}^2(\tau)\nabla\big(\Delta \log q_{T_g-\tau}(\mathbf{y}|\mathbf{c})\big) + \tilde{g}^2(\tau)\nabla^2 \log q_{T_g-\tau}(\mathbf{y}|\mathbf{c}) \cdot \nabla \log q_{T_g-\tau}(\mathbf{y}|\mathbf{c})\Big).
\end{aligned}$$

Substituting (46) into (45) and simplifying, we obtain

$$\mathrm{d}\big[\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big] = -\tilde{f}(\tau)\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\mathrm{d}\tau + \tilde{g}(\tau)\nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\mathrm{d}\tilde{\mathbf{B}}(\tau). \qquad (47)$$

Now, we establish the desired bound. First, for fixed $s \in (0, \tau)$, we have

$$\mathrm{d}\big[(\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c}))^T\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big]$$

$$= -\tilde{f}(\tau)(\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c}))^T \nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\mathrm{d}\tau$$
$$+ \tilde{g}(\tau)(\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c}))^T \nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\mathrm{d}\tilde{\mathbf{B}}(\tau).$$

By Lemma 11, $\int_s^\tau \mathbb{E}_{\tilde{\mathbf{Y}}(u)}\big[\|\nabla^2 \log q_{T_g-u}(\tilde{\mathbf{Y}}(u))\|_F^2\big]\mathrm{d}u < \infty$ where $\tilde{\mathbf{Y}}(u) \sim q_{T_g-u}(\cdot|\mathbf{c})$, implying that the final term in the above equation is a square-integrable martingale. Integrating on both sides and taking expectations w.r.t. the joint distribution of $(\tilde{\mathbf{Y}}(\tau), \tilde{\mathbf{Y}}(s))$ induced by (4), we get

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[(\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c}))^T \nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big] \\ &= -\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[\tilde{f}(\tau)(\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c}))^T \nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big].\end{aligned} \tag{48}$$

Next, if setting $\tilde{\Lambda}(\tau) = \exp\big(\int_0^\tau \tilde{f}(s)\mathrm{d}s\big)$ and applying Itô's formula to (47), we obtain

$$\mathrm{d}\big[\tilde{\Lambda}(\tau)\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big] = \tilde{\Lambda}(\tau)\tilde{g}(\tau)\nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\mathrm{d}\tilde{\mathbf{B}}(\tau).$$

We know from Lemma 11 that $\int_s^\tau \tilde{\Lambda}^2(u)\mathbb{E}_{\tilde{\mathbf{Y}}(u)}\big[\|\nabla^2 \log q_{T_g-u}(\tilde{\mathbf{Y}}(u))\|_F^2\big]\mathrm{d}u < \infty$ for any fixed $s > 0$. By Itô's isometry and chain rule, we obtain

$$\begin{aligned}&\tilde{\Lambda}^2(\tau)\tilde{g}^2(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|_F^2\big] \\ &= \frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[\|\tilde{\Lambda}(\tau)\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) - \tilde{\Lambda}(s)\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\|^2\big] \\ &= \tilde{\Lambda}^2(\tau)\cdot\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|^2 + 2\tilde{f}(\tau)\tilde{\Lambda}^2(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|^2 \\ &\quad - 2\tilde{\Lambda}(\tau)\tilde{\Lambda}(s)\cdot\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[(\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}))^T \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\big] \\ &\quad - 2\tilde{f}(\tau)\tilde{\Lambda}(\tau)\tilde{\Lambda}(s)\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[(\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}))^T \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\big] \\ &\overset{(48)}{=} \tilde{\Lambda}^2(\tau)\cdot\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|^2 + 2\tilde{f}(\tau)\tilde{\Lambda}^2(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|^2.\end{aligned}$$

It follows that

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|^2 + 2\tilde{f}(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|^2 \\ &= \tilde{g}^2(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|_F^2\big].\end{aligned} \tag{49}$$

For any fixed $s > 0$ and $\tau > s$, (48) and (49) lead to

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) - \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\|^2\big] \\ &= \frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|^2\big] - 2\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[(\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}))^T \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\big]\end{aligned}$$

$$
\begin{aligned}
&= \tilde{g}^2(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|_F^2\big] - 2\tilde{f}(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|^2 \\
&\quad + 2\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[\tilde{f}(\tau)(\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c}))^T \nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\big] \\
&= \tilde{g}^2(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|_F^2\big] \\
&\quad - 2\tilde{f}(\tau)\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) - \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\|^2\big] \\
&\quad + 2\tilde{f}(\tau)\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[(\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c}) - \nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}))^T \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\big].
\end{aligned}
$$

However, by Young's inequality,

$$
\begin{aligned}
&\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[(\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) - \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c}))^T \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\big] \\
&\le \frac{1}{2}\Big[\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) - \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\|^2\big] + \mathbb{E}_{\tilde{\mathbf{Y}}(s)}\big[\|\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\|^2\big]\Big].
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(s))}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) - \nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\|^2\big] \\
&\le \tilde{f}(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(s)}\big[\|\nabla \log q_{T_g-s}(\tilde{\mathbf{Y}}(s)|\mathbf{c})\|^2\big] + \tilde{g}^2(\tau)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)}\big[\|\nabla^2 \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c})\|_F^2\big].
\end{aligned}
$$

Let $s = \tau_k$ and $\tau \in [\tau_k, \tau_{k+1}]$. Integrate both sides of the above equation from τ_k to τ and use Lemma 9 and Lemma 11 to obtain

$$
\begin{aligned}
&\mathbb{E}_{(\tilde{\mathbf{Y}}(\tau),\tilde{\mathbf{Y}}(\tau_k))}\big[\|\nabla \log q_{T_g-\tau}(\tilde{\mathbf{Y}}(\tau)|\mathbf{c}) - \nabla \log q_{T_g-\tau_k}(\tilde{\mathbf{Y}}(\tau_k)|\mathbf{c})\|^2\big] \\
&\le \int_{\tau_k}^{\tau} \tilde{f}(u)\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)}\big[\|\nabla \log q_{T_g-\tau_k}(\tilde{\mathbf{Y}}(\tau_k)|\mathbf{c})\|^2\big] + \tilde{g}^2(u)\mathbb{E}_{\tilde{\mathbf{Y}}(u)}\big[\|\nabla^2 \log q_{T_g-u}(\tilde{\mathbf{Y}}(u)|\mathbf{c})\|_F^2\big]\mathrm{d}u \\
&\le \int_{\tau_k}^{\tau} \tilde{f}(u)d\sigma_{T_g-\tau_k}^{-2} + \tilde{g}^2(u)d\sigma_{T_g-u}^{-4} - \frac{\mathrm{d}}{\mathrm{d}r}\big(\sigma_{T-r}^{-4}\mathbb{E}_{\tilde{\mathbf{Y}}(r)\sim q_{T_g-r}(\cdot|\mathbf{c})}[\mathrm{trace}(\mathbf{\Sigma}_{T_g-r})]\big)\big|_{r=u}\mathrm{d}u \\
&\le \int_{\tau_k}^{\tau} \tilde{f}(u)d\sigma_{T_g-\tau_k}^{-2} + \tilde{g}^2(u)d\sigma_{T_g-u}^{-4}\mathrm{d}u \\
&\quad + \sigma_{T-\tau_k}^{-4}\mathbb{E}_{\mathbf{Y}(T_g-\tau_k)}[\mathrm{trace}(\mathbf{\Sigma}_{T_g-\tau_k})] - \sigma_{T-\tau}^{-4}\mathbb{E}_{\mathbf{Y}(T_g-\tau)}[\mathrm{trace}(\mathbf{\Sigma}_{T_g-\tau})],
\end{aligned}
$$

where we omit to write out the dependency of $\mathbf{\Sigma}_\tau$ on $\mathbf{Y}(\tau)$ and $\mathbf{c}$ for simplicity.

Finally, for the forward process $\mathbf{Y}$ in (3), since $\tilde{\mathbf{Y}}(\tau) = \mathbf{Y}(T_g - \tau)$ for any $\tau \in [0, T_g]$, we have for $\tau \in [\tau_{k-1}, \tau_k]$,

$$
\begin{aligned}
&\mathbb{E}_{(\mathbf{Y}(\tau),\mathbf{Y}(\tau_k))}\big[\|\nabla \log q_\tau(\mathbf{Y}(\tau)|\mathbf{c}) - \nabla \log q_{\tau_k}(\mathbf{Y}(\tau_k)|\mathbf{c})\|^2\big] \\
&\le \int_{\tau}^{\tau_k} f(s)d\sigma_{\tau_k}^{-2} + g^2(s)d\sigma_s^{-4}\mathrm{d}s + \big(\sigma_{\tau_k}^{-4}\mathbb{E}_{\mathbf{Y}(\tau_k)}[\mathrm{trace}(\mathbf{\Sigma}_{\tau_k})] - \sigma_\tau^{-4}\mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)]\big).
\end{aligned}
$$

This completes our proof. □

B.5 Proof of Lemma 5

Proof of Lemma 5. For a given condition $\mathbf{c}$, recall the definition of $\tau_{lipz}(\mathbf{c}), k_0(\mathbf{c})$ and k_1 in the proof of Lemma 4 in Appendix B.4. By Lemma 14 and Lemma 21 of Chen et al. (2023a) and our Lemma 9, we can derive

$$
\begin{aligned}
&\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k}[g^2(\tau_k)-g^2(\tau)]^2\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\left[\left\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\right\|^2\right]\mathrm{d}\tau\\
&\le\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k}(\Delta\tau)^2\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\left[\left\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\right\|^2\right]\mathrm{d}\tau\\
&\le\sum_{k=1}^{k_0(\mathbf{c})-1}\left[\int_{\tau_{k-1}}^{\tau_k}(\Delta\tau)^2\cdot dL(\mathbf{c})\lambda_\tau^{-1}\mathrm{d}\tau\right]+\int_{\tau_{k_0(\mathbf{c})-1}}^{\tau_{lipz}(\mathbf{c})}(\Delta\tau)^2\cdot dL(\mathbf{c})\lambda_\tau^{-1}\mathrm{d}\tau\\
&\quad+\int_{\tau_{lipz}(\mathbf{c})}^{\tau_{k_0(\mathbf{c})}}(\Delta\tau)^2\cdot d\sigma_\tau^{-2}\mathrm{d}\tau+\sum_{k_0(\mathbf{c})+1}^{K}\left[\int_{\tau_{k-1}}^{\tau_k}(\Delta\tau)^2\cdot d\sigma_\tau^{-2}\mathrm{d}\tau\right]\\
&\overset{(31)}{\lesssim}\int_0^{\tau_{lipz}(\mathbf{c})}d(\Delta\tau)^2L(\mathbf{c})(1+a\tau^2/4+b\tau/2)\mathrm{d}\tau+\int_{\tau_{lipz}(\mathbf{c})}^{\tau_{k_1}}\frac{d(\Delta\tau)^2}{a\tau^2/2+b\tau}\mathrm{d}\tau+\int_{\tau_{k_1}}^{T_g}d(\Delta\tau)^2\mathrm{d}\tau\\
&\lesssim d(\Delta\tau)^2L(\mathbf{c})\cdot\tau_{lipz}(\mathbf{c})+d(\Delta\tau)^2\int_{\tau_{lipz}(\mathbf{c})}^{\tau_{k_1}}\left(\frac{1}{\tau}+\frac{1}{\tau^2}\right)\mathrm{d}\tau+d(\Delta\tau)^2T_g\\
&\overset{(33)}{\lesssim} d(\Delta\tau)^2\Big(L_2(\mathbf{c})+\ln\left(2L_2(\mathbf{c})+1/2\right)+T_g+1\Big)\\
&\overset{(32)}{\lesssim} d(\Delta\tau)^2\Big(L(\mathbf{c})+T_g+1\Big).
\end{aligned}
$$

Taking expectation on both sides w.r.t. $\mathbf{c}\sim\mathbf{C}$, we obtain

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\left[\sum_{k=1}^{K}\int_{\tau_{k-1}}^{\tau_k}[g^2(\tau_k)-g^2(\tau)]^2\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\left[\left\|\nabla\log q_\tau(\mathbf{Y}(\tau)|\mathbf{c})\right\|^2\right]\mathrm{d}\tau\right]\\
&\lesssim d(\Delta\tau)^2\mathbb{E}_{\mathbf{c}\sim\mathbf{C}}\Big[L(\mathbf{c})+T_g+1\Big]\\
&\lesssim d(\Delta\tau)^2[L_1+T_g+1].
\end{aligned}
$$

The proof is therefore complete. □

C Additional Proofs

In this section, we prove Lemmas 6 and 11 that are used in Appendix B for the proof of Proposition 2.

C.1 Proof of Lemma 6

Proof of Lemma 6. Denote the following drift functions of (25) and (26) for simplicity:

$$\tilde{F}(\mathbf{y},\tau) := \tilde{f}(\tau)\mathbf{y} + \tilde{g}^2(\tau)\nabla \log q_{T_g-\tau}(\mathbf{y}|\mathbf{c}), \quad \tilde{F}_\theta(\tilde{\mathbf{z}}) := \tilde{f}(\tau_k)\tilde{\mathbf{z}} + \tilde{g}^2(\tau_k)s_\theta(T_g-\tau_k,\tilde{\mathbf{z}},\mathbf{c}).$$

We first prove Lemma 6(a). Consider an auxiliary process $\{\tilde{\mathbf{Y}}^a(\tau) : \tau \in (\tau_k,\tau_{k+1}]\}$, which also starts from $\tilde{\mathbf{Y}}^a(\tau_k) = \tilde{\mathbf{z}}$, with the dynamic

$$\mathrm{d}\tilde{\mathbf{Y}}^a(\tau) = [\tilde{f}(\tau_k)\tilde{\mathbf{z}} + \tilde{g}^2(\tau_k)\nabla \log q_{T_g-\tau_k}(\tilde{\mathbf{z}}|\mathbf{c})]\mathrm{d}\tau + \tilde{g}(\tau)\mathrm{d}\tilde{\mathbf{B}}(\tau) := \tilde{F}_a(\tilde{\mathbf{z}})\mathrm{d}\tau + \tilde{g}(\tau)\mathrm{d}\tilde{\mathbf{B}}(\tau). \tag{50}$$

Denote by $\tilde{p}^a_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})$ the conditional density of $\tilde{\mathbf{Y}}^a(\tau)$ given $\tilde{\mathbf{Y}}^a(\tau_k) = \tilde{\mathbf{z}}$. Then, $\tilde{p}^a_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})$ is a Gaussian density with mean $(\tau-\tau_k)\tilde{F}_a(\tilde{\mathbf{z}})$ and covariance $\int_{\tau_k}^{\tau}\tilde{g}^2(s)\mathrm{d}s \cdot I_d$.

Note that, for any $\tau \in (\tau_k,\tau_{k+1}]$ and $\tilde{\mathbf{z}} \in \mathbb{R}^d$,

$$\begin{aligned}\mathrm{KL}\big(\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})||\tilde{p}^\theta_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})\big) &= \int_{\mathbb{R}^d}\tilde{p}_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})\log\frac{\tilde{p}_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})}{\tilde{p}^\theta_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})}\mathrm{d}\mathbf{y}\\ &= \mathrm{KL}\big(\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})||\tilde{p}^a_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})\big) + \int_{\mathbb{R}^d}\tilde{p}_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})\log\frac{\tilde{p}^a_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})}{\tilde{p}^\theta_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})}\mathrm{d}\mathbf{y}. \end{aligned}\tag{51}$$

Using a similar argument as in the proof of Lemma 7(2) in Chen et al. (2023a), we can obtain that, for almost every $\tilde{\mathbf{z}} \in \mathbb{R}^d$,

$$\lim_{\tau\to\tau_k+}\mathrm{KL}(\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})||\tilde{p}^a_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})) = 0.$$

It follows that

$$\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\left[\lim_{\tau\to\tau_k+}\mathrm{KL}(\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k))||\tilde{p}^a_{\tau|\tau_k}(\cdot|\tilde{\mathbf{Y}}(\tau_k)))\right] = 0. \tag{52}$$

On the other hand, both $\tilde{p}^a_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})$ and $\tilde{p}^\theta_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})$ are Gaussian densities, and $\int_{\tau_k}^{\tau}\tilde{g}^2(s)\mathrm{d}s = [\tilde{g}^2(\tau_k) - a(\tau-\tau_k)/2](\tau-\tau_k) \le \tilde{g}^2(\tau_k)(\tau-\tau_k)$. Then, for almost every $\tilde{\mathbf{z}} \in \mathbb{R}^d$,

$$\lim_{\tau\to\tau_k+}\int_{\mathbb{R}^d}\tilde{p}_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})\log\frac{\tilde{p}^a_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})}{\tilde{p}^\theta_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})}\mathrm{d}\mathbf{y}$$

$$
\begin{aligned}
&= \lim_{\tau\to\tau_k+}\int_{\mathbb{R}^d}\tilde{p}_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})\bigg[-\frac{\|\mathbf{y}-(\tau-\tau_k)\tilde{F}_a(\tilde{\mathbf{z}})\|^2}{\int_{\tau_k}^{\tau}\tilde{g}^2(\tau)\mathrm{d}\tau}+\frac{\|\mathbf{y}-(\tau-\tau_k)\tilde{F}_\theta(\tilde{\mathbf{z}})\|^2}{\tilde{g}^2(\tau_k)(\tau-\tau_k)}\bigg]\mathrm{d}\mathbf{y}\\
&\le \lim_{\tau\to\tau_k+}\int_{\mathbb{R}^d}2\tilde{p}_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{z}})\bigg[\frac{\tilde{F}_\theta(\tilde{\mathbf{z}})}{\tilde{g}^2(\tau_k)}-\frac{\tilde{F}_a(\tilde{\mathbf{z}})}{\tilde{g}^2(\tau_k)-a(\tau-\tau_k)/2}\bigg]^T\mathbf{y}\mathrm{d}\mathbf{y}\\
&\le 2\lim_{\tau\to\tau_k+}\bigg\|\frac{\tilde{F}_\theta(\tilde{\mathbf{z}})}{\tilde{g}^2(\tau_k)}-\frac{\tilde{F}_a(\tilde{\mathbf{z}})}{\tilde{g}^2(\tau_k)-a(\tau-\tau_k)/2}\bigg\|\big(\mathbb{E}_{\tilde{\mathbf{Y}}(\tau)\sim\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})}\big[\|\tilde{\mathbf{Y}}(\tau)\|^2\big]\big)^{1/2}\\
&= \frac{2}{\tilde{g}^4(\tau_k)}\big\|\tilde{F}_\theta(\tilde{\mathbf{z}})-\tilde{F}_a(\tilde{\mathbf{z}})\big\|\cdot\|\tilde{\mathbf{z}}\|.
\end{aligned}
$$

The Cauchy-Schwarz inequality gives

$$
\begin{aligned}
&\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\bigg[\lim_{\tau\to\tau_k+}\int_{\mathbb{R}^d}\tilde{p}_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{Y}}(\tau_k))\log\frac{\tilde{p}^a_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{Y}}(\tau_k))}{\tilde{p}^\theta_{\tau|\tau_k}(\mathbf{y}|\tilde{\mathbf{Y}}(\tau_k))}\mathrm{d}\mathbf{y}\bigg]\\
&\le \frac{2}{\tilde{g}^4(\tau_k)}\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\big[\big\|\tilde{F}_\theta(\tilde{\mathbf{Y}}(\tau_k))-\tilde{F}_a(\tilde{\mathbf{Y}}(\tau_k))\big\|\cdot\|\tilde{\mathbf{Y}}(\tau_k)\|\big]\\
&\le \frac{2}{\tilde{g}^4(\tau_k)}\bigg[\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\big[\big\|\tilde{F}_\theta(\tilde{\mathbf{Y}}(\tau_k))-\tilde{F}_a(\tilde{\mathbf{Y}}(\tau_k))\big\|^2\big]\cdot\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\big[\|\tilde{\mathbf{Y}}(\tau_k)\|^2\big]\bigg]^{1/2}\\
&\le 2\bigg[\frac{1}{\tilde{g}^4(\tau_k)}\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\big\|\tilde{F}_\theta(\tilde{\mathbf{Y}}(\tau_k))-\tilde{F}_a(\tilde{\mathbf{Y}}(\tau_k))\big\|^2\cdot\big(\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2]+d\big)\bigg]^{1/2},
\end{aligned}
\tag{53}
$$

where the last inequality is from the solution (21) of the forward SDE (3) and

$$
\mathbb{E}_{\tilde{\mathbf{Y}}(\tau_k)\sim\tilde{p}_{\tau_k}}\big[\|\tilde{\mathbf{Y}}(\tau_k)\|^2\big]=\mathbb{E}_{\mathbf{Y}(T_g-\tau_k)\sim q_{T_g-\tau_k}}\big[\|\mathbf{Y}(T_g-\tau_k)\|^2\big]\le\mathbb{E}_{\mathbf{Y}(0)\sim q_0(\cdot|\mathbf{c})}[\|\mathbf{Y}(0)\|^2]+d.
$$

Combining (51), (52) and (53) yields Lemma 6(a).

Next, we consider bounding $\frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{KL}(\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})||\tilde{p}^\theta_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}})$ for Lemma 6(b). For any given $\tilde{\mathbf{z}}\in\mathbb{R}^d$, we use the following notations for simplicity,

$$
\bar{p}_\tau:=\tilde{p}_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}}),\quad \bar{p}^\theta_\tau:=\tilde{p}^\theta_{\tau|\tau_k}(\cdot|\tilde{\mathbf{z}}).
$$

Using the same argument as in the proof of Lemma 6 in Chen et al. (2023a), we have

$$
\frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{KL}(\bar{p}_\tau||\bar{p}^\theta_\tau)=\int_{\mathbf{y}\in\mathbb{R}^d}\log\frac{\bar{p}_\tau}{\bar{p}^\theta_\tau}\cdot\frac{\partial\bar{p}_\tau}{\partial\tau}(\mathbf{y})\mathrm{d}\mathbf{y}-\int_{\mathbf{y}\in\mathbb{R}^d}\frac{\bar{p}_\tau}{\bar{p}^\theta_\tau}\cdot\frac{\partial\bar{p}^\theta_\tau}{\partial\tau}(\mathbf{y})\mathrm{d}\mathbf{y}.
\tag{54}
$$

Applying the Fokker-Planck equation to (25) and (26), we get the evolutions of $\bar{p}_\tau$ and $\bar{p}^\theta_\tau$ as

$$
\frac{\partial\bar{p}_\tau}{\partial\tau}(\mathbf{y})=\nabla\cdot\bigg[-\tilde{F}(\mathbf{y},\tau)\bar{p}_\tau(\mathbf{y})+\frac{1}{2}\tilde{g}^2(\tau)\nabla\bar{p}_\tau(\mathbf{y})\bigg],
$$

$$\frac{\partial \bar{p}_\tau^\theta}{\partial \tau}(\mathbf{y}) = \nabla \cdot \bigg[- \tilde{F}_\theta(\tilde{\mathbf{z}}) \cdot \bar{p}_\tau^\theta(\mathbf{y}) + \frac{1}{2}\tilde{g}^2(\tau_k)\nabla \bar{p}_\tau^\theta(\mathbf{y}) \bigg].$$

It again follows from the Fokker-Planck equation that the first term of (54) can be expressed as

$$\int_{\mathbf{y}\in\mathbb{R}^d} \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta} \cdot \frac{\partial \bar{p}_\tau}{\partial \tau}(\mathbf{y})\mathrm{d}\mathbf{y} = \int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\tilde{F}(\mathbf{y},\tau)]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} - \frac{\tilde{g}^2(\tau)}{2}\int_{\mathbf{y}\in\mathbb{R}^d} [\nabla \bar{p}_\tau(\mathbf{y})]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y},$$

and the second term of (54) as

$$\int_{\mathbf{y}\in\mathbb{R}^d} \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta} \cdot \frac{\partial \bar{p}_\tau^\theta}{\partial \tau}(\mathbf{y})\mathrm{d}\mathbf{y} = \int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau^\theta(\mathbf{y})\tilde{F}_\theta(\tilde{\mathbf{z}})^T \nabla \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} - \frac{\tilde{g}^2(\tau_k)}{2}\int_{\mathbf{y}\in\mathbb{R}^d} [\nabla \bar{p}_\tau^\theta(\mathbf{y})]^T \nabla \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y}.$$

We compute

$$\begin{aligned}
&\int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\tilde{F}(\mathbf{y},\tau)]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} - \int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau^\theta(\mathbf{y})\tilde{F}_\theta(\tilde{\mathbf{z}})^T \nabla \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} \\
&= \int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\tilde{F}(\mathbf{y},\tau)]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} - \int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})\tilde{F}_\theta(\tilde{\mathbf{z}})^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} \\
&= \int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\tilde{F}(\mathbf{y},\tau) - \tilde{F}_\theta(\tilde{\mathbf{z}})]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} \\
&\le \int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y}) \bigg[\frac{1}{C_{KL}} \big\| \tilde{F}(\mathbf{y},\tau) - \tilde{F}_\theta(\tilde{\mathbf{z}}) \big\|^2 + \frac{C_{\mathrm{KL}}}{4} \bigg\| \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y}) \bigg\|^2 \bigg] \mathrm{d}\mathbf{y} \\
&= \frac{1}{C_{KL}} \mathbb{E}_{\bar{\mathbf{Y}}(\tau)\sim\bar{p}_\tau} \big[\big\| \tilde{F}(\bar{\mathbf{Y}}(\tau),\tau) - \tilde{F}_\theta(\tilde{\mathbf{z}}) \big\|^2 \big] + \frac{C_{\mathrm{KL}}}{4} \mathbb{E}_{\bar{\mathbf{Y}}(\tau)\sim\bar{p}_\tau} \bigg[\bigg\| \nabla \log \frac{\bar{p}_\tau(\bar{\mathbf{Y}}(\tau))}{\bar{p}_\tau^\theta(\bar{\mathbf{Y}}(\tau))} \bigg\|^2 \bigg],
\end{aligned} \tag{55}$$

where the inequality is from the fact that $\mathbf{x}^T\mathbf{y} \le \frac{1}{C_{\mathrm{KL}}}\|\mathbf{x}\|^2 + \frac{C_{\mathrm{KL}}}{4}\|\mathbf{y}\|^2$ for any $\mathbf{x},\mathbf{y}\in\mathbb{R}^d$ and any fixed constant $C_{\mathrm{KL}} > 0$.

In addition, we have

$$\begin{aligned}
&\frac{\tilde{g}^2(\tau_k)}{2}\int_{\mathbf{y}\in\mathbb{R}^d} [\nabla \bar{p}_\tau^\theta(\mathbf{y})]^T \nabla \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} - \frac{\tilde{g}^2(\tau)}{2}\int_{\mathbf{y}\in\mathbb{R}^d} [\nabla \bar{p}_\tau(\mathbf{y})]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} \\
&= \frac{\tilde{g}^2(\tau_k)}{2}\int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau^\theta(\mathbf{y})[\nabla \log \bar{p}_\tau^\theta(\mathbf{y})]^T \nabla \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} - \frac{\tilde{g}^2(\tau)}{2}\int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\nabla \log \bar{p}_\tau(\mathbf{y})]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} \\
&= \frac{\tilde{g}^2(\tau_k)}{2}\int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\nabla \log \bar{p}_\tau^\theta(\mathbf{y})]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} - \frac{\tilde{g}^2(\tau)}{2}\int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\nabla \log \bar{p}_\tau(\mathbf{y})]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} \\
&= -\frac{1}{2}\int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\tilde{g}^2(\tau)\nabla \log \bar{p}_\tau(\mathbf{y}) - \tilde{g}^2(\tau_k)\nabla \log \bar{p}_\tau^\theta(\mathbf{y})]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} \\
&= \frac{1}{2}\int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\tilde{g}^2(\tau_k) - \tilde{g}^2(\tau)][\nabla \log \bar{p}_\tau(\mathbf{y})]^T \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\mathrm{d}\mathbf{y} - \frac{1}{2}\int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})\tilde{g}^2(\tau_k) \bigg\| \nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y}) \bigg\|^2 \mathrm{d}\mathbf{y}
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{4C_{\mathrm{KL}}} \int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y})[\tilde{g}^2(\tau_k) - \tilde{g}^2(\tau)]^2 \left\|\nabla \log \bar{p}_\tau(\mathbf{y})\right\|^2 \mathrm{d}\mathbf{y} \\
&\quad + \left[\frac{C_{\mathrm{KL}}}{4} - \frac{1}{2}\tilde{g}^2(\tau_k)\right] \int_{\mathbf{y}\in\mathbb{R}^d} \bar{p}_\tau(\mathbf{y}) \left\|\nabla \log \frac{\bar{p}_\tau}{\bar{p}_\tau^\theta}(\mathbf{y})\right\|^2 \mathrm{d}\mathbf{y} \\
&= \frac{[\tilde{g}^2(\tau_k) - \tilde{g}^2(\tau)]^2}{4C_{\mathrm{KL}}} \mathbb{E}_{\bar{\mathbf{Y}}(\tau)\sim\bar{p}_\tau}\left[\left\|\nabla \log \bar{p}_\tau(\bar{\mathbf{Y}}(\tau))\right\|^2\right] \\
&\quad + \frac{1}{2}\left[\frac{C_{\mathrm{KL}}}{2} - \tilde{g}^2(\tau_k)\right] \mathbb{E}_{\bar{\mathbf{Y}}(\tau)\sim\bar{p}_\tau} \left\|\nabla \log \frac{\bar{p}_\tau(\bar{\mathbf{Y}}(\tau))}{\bar{p}_\tau^\theta(\bar{\mathbf{Y}}(\tau))}\right\|^2,
\end{aligned} \tag{56}
$$

where the inequality is due to $\mathbf{x}^T\mathbf{y} \leq \frac{1}{2C_{\mathrm{KL}}}\|\mathbf{x}\|^2 + \frac{C_{\mathrm{KL}}}{2}\|\mathbf{y}\|^2$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and any fixed constant $C_{\mathrm{KL}} > 0$.

Finally, by combining (55) and (56), we can infer from (54) that (27) holds. This completes the proof of Lemma 6. □

C.2 Proof of Lemma 11 via Stochastic Localization

In this section, we briefly introduce the technique of stochastic localization, based on which we prove Lemma 11. The main idea of the proof follows Benton et al. (2024), but we consider diffusion models with time-*inhomogeneous* forward processes. Throughout this section, we consider a given condition $\mathbf{c}$ and the target distribution $p_{\mathrm{target}}(\cdot|\mathbf{c})$.

We first discuss stochastic localization; for details, see e.g. Benton et al. (2024); Montanari (2023). Give a random variable $\xi \sim p_{\mathrm{target}}(\cdot|\mathbf{c})$, the stochastic localization considers a process $\big(\mathbf{U}(s)\big)_{s\geq 0}$ defined by

$$
\mathbf{U}(s) = s\xi + \bar{\mathbf{B}}(s), \text{ for } s \geq 0, \tag{57}
$$

where $\bar{\mathbf{B}}$ is a d-dimensional standard Brownian motion independent of ξ. Denote the posterior covariance of ξ given $\mathbf{U}(s)$ and $\mathbf{c}$ by $\mathbf{A}_s(\mathbf{U}(s), \mathbf{c}) = \mathrm{Cov}_{\xi|(\mathbf{U}(s),\mathbf{c})}(\xi)$. Then, we have the following characterization (Benton et al., 2024, Proposition 2):

$$
\frac{\mathrm{d}}{\mathrm{d}s}\mathbb{E}_{\mathbf{U}(s)}\big[\mathbf{A}_s\big] = -\mathbb{E}_{\mathbf{U}(s)}\big[\mathbf{A}_s^2\big], \quad \text{for } s \geq 0, \tag{58}
$$

where the expectation is taken w.r.t. the marginal distribution of $\mathbf{U}(s)$ given $\mathbf{c}$, and we omit the dependence of $\mathbf{A}_s$ on $\mathbf{U}(s)$ and $\mathbf{c}$ for notational simplicity.

We then have the following result, which shows that the forward OU process $\mathbf{Y}$ can be viewed as a stochastic localization under a time change. The proof of this lemma is deferred to the end of this section.

Lemma 12. *Under condition* $\mathbf{c}$*, if we define the process* $\mathbf{Y}$ *and* $\mathbf{U}$ *according to* (3) *and* (57)*, respectively, and consider a time change*

$$
\tau(s) = \inf\left\{u : \exp(au^2/2 + bu) - 1 > s^{-1}\right\}, \tag{59}
$$

then $\{\mathbf{U}(s) : s \geq 0\}$ *and* $\{s\lambda_{\tau(s)}^{-1}\mathbf{Y}(\tau(s)) : s \geq 0\}$ *have the same law, where* λ_τ *is defined in* (22).

Denote by $q_{0|\tau}(\cdot|\mathbf{y}(\tau), \mathbf{c})$ the posterior distribution of $\mathbf{Y}(0)$ given $\mathbf{c}$ and $\mathbf{Y}(\tau) = \mathbf{y}(\tau)$. Let the posterior covariance matrix be $\mathbf{\Sigma}_\tau(\mathbf{y}(\tau), \mathbf{c}) := \mathrm{Cov}_{\mathbf{Y}(0)\sim q_{0|\tau}(\cdot|\mathbf{y}(\tau),\mathbf{c})}\big[\mathbf{Y}(0)\big]$, for which we may omit its dependence on $\mathbf{y}(\tau)$ and $\mathbf{c}$ when the context is clear. Then, for any $s > 0$, when $\tau = \tau(s)$, it follows from Lemma 12 that

$$\begin{aligned}
\mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathbf{\Sigma}_\tau^2\big] &= \int_{\mathbf{y}\in\mathbb{R}^d} [\mathrm{Cov}(\mathbf{Y}(0)|\mathbf{Y}(\tau) = \mathbf{y}, \mathbf{c})]^2 \mathrm{d}P_{\mathbf{Y}(\tau)}(\mathbf{y}) \\
&= \int_{\mathbf{y}\in\mathbb{R}^d} [\mathrm{Cov}(\mathbf{Y}(0)|s\lambda_\tau^{-1}\mathbf{Y}(\tau) = s\lambda_\tau^{-1}\mathbf{y}, \mathbf{c})]^2 \mathrm{d}P_{s\lambda_\tau^{-1}\mathbf{Y}(\tau)}(s\lambda_\tau^{-1}\mathbf{y}) \\
&= \int_{\mathbf{y}\in\mathbb{R}^d} [\mathrm{Cov}(\xi|\mathbf{U}(s) = s\lambda_\tau^{-1}\mathbf{y}, \mathbf{c})]^2 \mathrm{d}P_{\mathbf{U}(s)}(s\lambda_\tau^{-1}\mathbf{y}) \\
&= \int_{\mathbf{u}\in\mathbb{R}^d} [\mathrm{Cov}(\xi|\mathbf{U}(s) = \mathbf{u}, \mathbf{c})]^2 \mathrm{d}P_{\mathbf{U}(s)}(\mathbf{u}) \\
&= \mathbb{E}_{\mathbf{U}(s)}\big[\mathbf{A}_s^2\big],
\end{aligned}$$

and similarly $\mathbb{E}_{\mathbf{U}(s)}\big[\mathbf{A}_s\big] = \mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathbf{\Sigma}_\tau\big]$, where the expectations involving $\mathbf{\Sigma}_\tau = \mathbf{\Sigma}_\tau(\mathbf{Y}(\tau), \mathbf{c})$ are taken w.r.t. $\mathbf{Y}(\tau) \sim q_\tau(\cdot|\mathbf{c})$ of the forward process (3), and $P_{\mathbf{M}}$ is the distribution function of a general random variable $\mathbf{M}$. Equation (58) yields

$$\mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathbf{\Sigma}_\tau^2\big] = \mathbb{E}_{\mathbf{U}(s)}\big[\mathbf{A}_s^2\big] = -\frac{\mathrm{d}}{\mathrm{d}s}\mathbb{E}_{\mathbf{U}(s)}\big[\mathbf{A}_s\big] = -\frac{\mathrm{d}}{\mathrm{d}s}\mathbb{E}_{\mathbf{Y}(\tau(s))}\big[\mathbf{\Sigma}_{\tau(s)}\big] = -\frac{\mathrm{d}\tau(s)}{\mathrm{d}s}\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathbf{\Sigma}_\tau\big]. \quad (60)$$

Note that $\sigma_\tau^2 = 1 - \exp(-a\tau^2/2 - b\tau)$, and $\frac{\mathrm{d}(\sigma_\tau^2)}{\mathrm{d}\tau} = 2\sigma_\tau\frac{\mathrm{d}\sigma_\tau}{\mathrm{d}\tau} := 2\sigma_\tau\dot{\sigma}_\tau = (a\tau + b)e^{-a\tau^2/2-b\tau}$. Hence, for $\tau = \tau(s)$,

$$s^{-1} = \exp(a\tau^2/2 + b\tau) - 1 = \frac{1}{1-\sigma_\tau^2} - 1 = \frac{\sigma_\tau^2}{1-\sigma_\tau^2},$$

which indicates that $\frac{\mathrm{d}s}{\mathrm{d}\tau} = \big[-2\dot{\sigma}_\tau\sigma_\tau \cdot \sigma_\tau^2 - 2\dot{\sigma}_\tau\sigma_\tau(1-\sigma_\tau^2)\big]/\sigma_\tau^4 = -2\dot{\sigma}_\tau/\sigma_\tau^3$. Then, by taking the trace of both sides of (60), we obtain

$$\mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathrm{trace}(\mathbf{\Sigma}_\tau^2)\big] = -\frac{\mathrm{d}\tau(s)}{\mathrm{d}s}\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathrm{trace}(\mathbf{\Sigma}_\tau)\big] = \frac{\sigma_\tau^3}{2\dot{\sigma}_\tau}\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathrm{trace}(\mathbf{\Sigma}_\tau)\big], \quad (61)$$

which also implies that $\mathbb{E}_{\mathbf{Y}(\tau)}\big[\mathrm{trace}(\mathbf{\Sigma}_\tau)\big]$ is increasing in τ since $\sigma_\tau\dot{\sigma}_\tau \geq 0$ for all $\tau \geq 0$.

Now we are ready to prove Lemma 11.

Proof of Lemma 11. Recall that $\mathbf{Y}(\tau)|\mathbf{Y}(0) \sim \mathcal{N}(\lambda_\tau\mathbf{Y}(0), \sigma_\tau^2 I_d)$. Using a similar argument as in the proof of Lemma 5 of Benton et al. (2024), we obtain

$$\nabla^2 \log q_\tau(\mathbf{y}_\tau|\mathbf{c}) = -\sigma_\tau^{-2}I_d + \sigma_\tau^{-4}\mathrm{Cov}_{q_{0|\tau}(\cdot|\mathbf{y}(\tau),\mathbf{c})}\big(\mathbf{y}(\tau) - \lambda_\tau\mathbf{Y}(0)\big) = -\sigma_\tau^{-2}I_d + \sigma_\tau^{-4}\lambda_\tau^2\mathbf{\Sigma}_\tau.$$

Noting that $\|A\|_F^2 = \mathrm{trace}(A^T A)$ for a matrix A and that $1 > \lambda_\tau^2 = \exp(-a\tau^2/2 - b\tau) = 2\sigma_\tau\dot{\sigma}_\tau/(a\tau + b)$, we can directly compute

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}\big[\|\nabla^2 \log q_\tau(\mathbf{Y}_\tau|\mathbf{c})\|_F^2\big] \\
&= d\sigma_\tau^{-4} - 2\sigma_\tau^{-6}\lambda_\tau^2 \mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)] + \sigma_\tau^{-8}\lambda_\tau^4 \mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau^2)] \\
&\le d\sigma_\tau^{-4} - 2\sigma_\tau^{-6}\lambda_\tau^2 \mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)] + \sigma_\tau^{-8}\lambda_\tau^2 \mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau^2)] \\
&\overset{(61)}{=} d\sigma_\tau^{-4} - \frac{4\sigma_\tau^{-5}\dot{\sigma}_\tau}{a\tau + b}\mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)] + \frac{2\sigma_\tau^{-7}\dot{\sigma}_\tau}{a\tau + b}\cdot\frac{\sigma_\tau^3}{2\dot{\sigma}_\tau}\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)] \\
&= d\sigma_\tau^{-4} - \frac{4\sigma_\tau^{-5}\dot{\sigma}_\tau}{a\tau + b}\mathbb{E}_{\mathbf{Y}(\tau)}[\mathrm{trace}(\mathbf{\Sigma}_\tau)] + \frac{\sigma_\tau^{-4}}{a\tau + b}\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbb{E}_{\mathbf{Y}(\tau))}[\mathrm{trace}(\mathbf{\Sigma}_\tau)] \\
&= d\sigma_\tau^{-4} + \frac{1}{a\tau + b}\cdot\frac{\mathrm{d}}{\mathrm{d}\tau}\left(\sigma_\tau^{-4}\mathbb{E}_{\mathbf{Y}(\tau)\sim q_\tau(\cdot|\mathbf{c})}[\mathrm{trace}(\mathbf{\Sigma}_\tau)]\right).
\end{aligned}
$$

The proof is complete by virtue of $g^2(\tau) = a\tau + b$. □

Proof of Lemma 12. Letting $\Lambda(\tau) = \exp\big(\int_0^\tau f(s)\mathrm{d}s\big)$ and applying Itô's formula to the forward SDE (3) under the condition $\mathbf{c}$, we have

$$
\mathrm{d}[\Lambda(\tau)\mathbf{Y}(\tau)] = f(\tau)\Lambda(\tau)\mathbf{Y}(\tau)\mathrm{d}\tau + \Lambda(\tau)\big[-f(\tau)\mathbf{Y}(\tau)\mathrm{d}\tau + g(\tau)\mathrm{d}\mathbf{B}(\tau)\big] = \Lambda(\tau)g(\tau)\mathrm{d}\mathbf{B}(\tau).
$$

By the Dambis-Dubins–Schwarz theorem (Karatzas and Shreve, 2014, Theorem 3.4.6), there is a process $(\hat{\mathbf{B}}(s))_{s\ge 0}$ such that under the time change

$$
\kappa(s) := \inf\left\{\tau : \int_0^\tau \Lambda^2(u)g^2(u)\mathrm{d}u > s\right\} = \inf\left\{\tau : \exp\big(a\tau^2/2 + b\tau\big) - 1 > s\right\},
$$

we have

$$
\hat{\mathbf{B}}(e^{a\tau^2/2 + b\tau} - 1) = \int_0^\tau \Lambda(u)g(u)\mathrm{d}\mathbf{B}(u),
$$

where $(\hat{\mathbf{B}}(s))_{s\ge 0}$ is a standard Brownian motion w.r.t. the filtration $\big(\mathcal{F}_{\kappa(s)}\big)_{s\ge 0}$, while $(\mathcal{F}_\tau)_{\tau\ge 0}$ is the natural filtration generated by $(\mathbf{B}(\tau))_{\tau\ge 0}$. Then, for any $s > 0$, we have

$$
\Lambda(\kappa(s))\mathbf{Y}(\kappa(s)) = \mathbf{Y}(0) + \int_0^{\kappa(s)} \Lambda(u)g(u)\mathrm{d}\mathbf{B}(u) = \mathbf{Y}(0) + \hat{\mathbf{B}}(s).
$$

Now, if we set $\mathbf{U}(0) = 0$ and

$$
\mathbf{U}(s) := s\Lambda(\kappa(s^{-1}))\mathbf{Y}(\kappa(s^{-1})) = s\mathbf{Y}(0) + s\hat{\mathbf{B}}(s^{-1}), \text{ for } s > 0,
$$

then we find that $\mathbf{U}$ satisfies the definition of the stochastic localization in (57), because $\mathbf{Y}(0) \sim p_{\text{target}}(\cdot|\mathbf{c})$ and the law of $\big(s\hat{\mathbf{B}}(s^{-1})\big)_{s\ge 0}$ is the same as the law of $\big(\hat{\mathbf{B}}(s)\big)_{s\ge 0}$. The proof is complete by noting that $\kappa(s^{-1}) = \tau(s)$ in (59). □

D The $q-$Learning Algorithm for Continuous-Time Mean-Variance Portfolio Selection

In this section, we present the RL algorithm, Algorithm 3, inplemeted for the continuous-time mean-variance portfolio selection problem discussed in Section 5. Of the actor-critic type, it is a direct extension of Algorithm 5 in Jia and Zhou (2023) by taking into account of synthetic price paths. We refer to Jia and Zhou (2023) for further details about the theoretical foundation and the design principle of this so-called $q-$learning algorithm.

E Experiment Settings

In this section we outline the settings of our experiments. We first introduce the neural network structure we use for the score network $s_\theta(\tau_k, \mathbf{x}, \mathbf{c})$, where the Multilayer Perceptron (MLP) is employed as the base model. To utilize the temporal information τ_k better, we apply the idea of Feature-wise Linear Modulation (FiLM) introduced in Perez et al. (2018). We use the conditional linear layer $L(\mathbf{z}, \tau_k)$, which is an element-wise scaling of a linear transformation, to replace the standard linear layer in MLP. Specifically, let $\mathbf{z} \in \mathbb{R}^{d_{\text{in}}}$ be the input vector and $k \in \{1, \dots, T_g\}$ be a discrete diffusion step. The output of $L(\mathbf{z}, \tau_k) \in \mathbb{R}^{d_{\text{out}}}$ is given by:

$$L(\mathbf{z}, \tau_k) = \boldsymbol{\gamma}(k) \odot (\mathbf{W}\mathbf{z} + \mathbf{b}),$$

where $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and $\mathbf{b} \in \mathbb{R}^{d_{\text{out}}}$ are respectively the learnable weight and bias of the linear projection, $\boldsymbol{\gamma} : \{1, \dots, T_g\} \to \mathbb{R}^{d_{\text{out}}}$ is a learnable time-step embedding vector, and $\odot$ denotes the Hadamard (element-wise) product. The neural network $s_\theta(\tau_k, \mathbf{x}, \mathbf{c})$ is constructed by the composition of m conditional linear layers $L_1, \cdots L_m$, and m activation functions $A_1, \cdots, A_m$:

$$s_\theta(\tau_k, \mathbf{x}, \mathbf{c}) = A_m \circ L_m(\cdot, \tau_k) \circ A_{m-1} \circ L_{m-1}(\cdot, \tau_k) \circ \cdots \circ A_1 \circ L_1((\mathbf{x}, \mathbf{c}), \tau_k).$$

Next, Tables 9 and 10 present the hyperparameters used to define the network structure for the 1-dimensional and 100-dimensional examples, respectively. In addition, Table 11 and Table 12 display the hyperparameter settings of the diffusion models for the 1-dimensional and 100-dimensional examples, respectively.

Algorithm 3 Offline q-Learning Mean–Variance Algorithm with Synthetic Wealth Simulator

Inputs: initial state $x(0)$, horizon T, time step Δt, number of steps $N_T = T/\Delta t$, number of episodes M, number of real asset price paths H, number of synthetic paths (generated at each episode) H_s, parameterized value function $J_\varphi(\cdot,\cdot;w)$ and policy $\pi_\psi(\cdot,\cdot;w)$, initial Lagrange multiplier w, learning rate functions $\alpha_\varphi(\cdot), \alpha_\psi(\cdot), \alpha_w(\cdot)$

Required program: a real wealth simulator $x' = Wealth_{\Delta t}(t,x,a)$ that takes current time–state pair (t,x) and action a as inputs and generates state x' at time $t+\Delta t$ according to (17), a synthetic wealth simulator $x' = Wealth_{\Delta t}(t,x,a;G(\theta))$ based on synthetic paths from a generative model $G(\theta)$, q-function $q_\psi(t,x,a;w) = \log \pi_\psi(t,x,a;w)$.

Learning procedure:

for episode $m = 1$ **to** M **do**

Generate H_s synthetic asset price paths by $G(\theta)$.

Initialize $n = 0$ and store $x^{(j)}(t_n) \leftarrow x(0)$ for $j = 1, ..., H + H_s$.

while $n < N_T$ **do**

for path $j = 1$ **to** $H + H_s$ **do**

Generate action $a^{(j)}(t_n) \sim \pi_\psi(\cdot|t_n, x^{(j)}(t_n); w)$.

Compute and store the test functions

$$\xi^{(j)}(t_n) = \frac{\partial J_\varphi}{\partial \varphi}(t_n, x^{(j)}(t_n); w), \quad \zeta^{(j)}(t_n) = \frac{\partial q_\psi}{\partial \psi}(t_n, x^{(j)}(t_n), a^{(j)}(t_n); w).$$

if $j \le H$ **then**

In the real wealth simulator: $x^{(j)}(t_{n+1}) = Wealth_{\Delta t}(t_n, x^{(j)}(t_n), a^{(j)}(t_n))$.

else

In the synthetic wealth simulator: $x^{(j)}(t_{n+1}) = Wealth_{\Delta t}(t_n, x^{(j)}(t_n), a^{(j)}(t_n); G(\theta))$.

end if

end for

Update $n \leftarrow n + 1$.

end while

Store the terminal wealth $x^{(j)}(T; m) \leftarrow x^{(j)}(t_{N_T})$ for $j = 1, ..., H + H_s$.

For path $j = 1, \cdots H + H_s$, compute

$$\Delta\varphi^{(j)} = \sum_{n=0}^{N_T-1} \xi^{(j)}(t_n)\big[J_\varphi(t_{n+1}, x^{(j)}(t_{n+1}); w) - J_\varphi(t_n, x^{(j)}(t_n); w) - q_\psi(t_n, x^{(j)}(t_n), a^{(j)}(t_n); w)\Delta t\big],$$

$$\Delta\psi^{(j)} = \sum_{n=0}^{N_T-1} \zeta^{(j)}(t_n)\big[J_\varphi(t_{n+1}, x^{(j)}(t_{n+1}); w) - J_\varphi(t_n, x^{(j)}(t_n); w) - q_\psi(t_n, x^{(j)}(t_n), a^{(j)}(t_n); w)\Delta t\big].$$

Update φ and ψ: $\varphi \leftarrow \varphi + \frac{\alpha_\varphi(m)}{H+H_s}\sum_{j=1}^{H+H_s} \Delta\varphi^{(j)}$, $\psi \leftarrow \psi + \cdot\frac{\alpha_\psi(m)}{H+H_s}\sum_{j=1}^{H+H_s} \Delta\psi^{(j)}$

Update w (Lagrange multiplier) every L episodes:

if $m \equiv 0 \bmod L$ **then**

$w \leftarrow w - \alpha_w \frac{1}{L}\sum_{i=m-L+1}^{m} \frac{1}{H+H_s}\sum_{j=1}^{H+H_s} x^{(j)}(T; i)$

end if

end for

Layer ID	d_{in}	d_{out}	Activation function
1 (Input)	$1+1$	32	Softplus
2 (Hidden)	32	32	Softplus
3 (Hidden)	32	16	Softplus
4 (Hidden)	16	16	Softplus
5 (Output)	16	1	-

Table 9: Neural network structure for the 1-dimensional examples in Sections 3 and 5

Layer ID	d_{in}	d_{out}	Activation function
1 (Input)	$100+100$	256	Softplus
2 (Hidden)	256	512	Softplus
3 (Hidden)	512	512	Softplus
4 (Hidden)	512	128	Softplus
5 (Output)	128	100	-

Table 10: Neural network structure for the 100-dimensional example in Section 3

Hyperparameters	Values
Drift coef. $f(\tau)$	$\frac{1}{2}(19.9\tau+0.1)$
Diffusion coef. $g(\tau)$	$\sqrt{19.9\tau+0.1}$
Diffusion horizon T_g	1
Time step $\Delta\tau$	10^{-2}
Number of Episodes	6000
Batch Size m	H
Learning Rate $\alpha(j)$	10^{-3} if $j<4000$, 10^{-5} o.w.

Table 11: Hyperparameters of diffusion models for the 1-dimensional examples, where H is the number of training paths.

Hyperparameters	Values
Drift coef. $f(\tau)$	$\frac{1}{2}(19.9\tau+0.1)$
Diffusion coef. $g(\tau)$	$\sqrt{19.9\tau+0.1}$
Diffusion horizon T_g	1
Time step $\Delta\tau$	10^{-3}
Number of Episodes	120,000
Batch Size m	H
Learning Rate $\alpha(j)$	10^{-3} if $j<100,000$, 10^{-5} o.w.

Table 12: Hyperparameters of diffusion models for the 100-dimensional example, where H is the number of training paths.