Conditional Diffusion Guidance under Hard Constraint: A Stochastic Analysis Approach

Zhengyi Guo [*] Wenpin Tang [*] Renyuan Xu [†]

Janurary 21, 2026

Abstract

We study conditional generation in diffusion models under hard constraints, where generated samples must satisfy prescribed events with probability one. Such constraints arise naturally in safety-critical applications and in rare-event simulation, where soft or reward-based guidance methods offer no guarantee of constraint satisfaction. Building on a probabilistic interpretation of diffusion models, we develop a principled conditional diffusion guidance framework based on Doob's h-transform, martingale representation and quadratic variational process. Specifically, the resulting guided dynamics augment a pretrained diffusion with an explicit drift correction involving the logarithmic gradient of a conditioning function $\mathbb{P}(Y_T \in S | Y_t = y)$, without modifying the pretrained score network. Leveraging martingale and quadratic-variation identities, we propose two novel off-policy learning algorithms based on a martingale loss and a martingale–covariation loss to estimate h and its gradient using only trajectories from the pretrained model. We provide non-asymptotic guarantees for the resulting conditional sampler in both total variation and Wasserstein distances, explicitly characterizing the impact of score approximation and guidance estimation errors. Numerical experiments demonstrate the effectiveness of the proposed methods in enforcing hard constraints and generating rare-event samples.

1 Introduction

Diffusion models have emerged as a powerful class of generative models capable of producing high-quality samples across a wide range of domains, including image synthesis, molecular design, and time series generation. Notably, they have played a central role in the success in text-to-image creators such as DALL·E 2 [69] and Stable Diffusion [72]; in text-to-video generators such as Sora [66], Make-A-Video [75] and Veo [32]; and more recently, in diffusion large language models such as Mercury [52] and LLaDA [64]. In many downstream applications, however, the objective goes beyond unconditional sampling from a data distribution. Instead, we are often interested in generating samples that satisfy prescribed structural, functional, or feasibility constraints. Such requirements naturally give rise to the problem of guided or fine-tuned diffusion sampling, in which a generative model is adapted to target a conditional distribution associated with a desired event or property.

This demand for conditional generation arises from two closely related but distinct considerations. First, in many applications, generated data must respect hard constraints dictated by physical laws, operational rules, or feasibility requirements. Examples include safety-critical systems, regulated decision-making pipelines, and constrained design problems, where violations are not allowed and must be ruled out at the level of the generated distribution. Second, in domains such as finance, healthcare, and large-scale service systems, there is a growing need for

[*] Department of Industrial Engineering and Operations Research, Columbia University. **Email:** zg2525@columbia.edu and wt2319@columbia.edu.

[†] Department of Management Science and Engineering, Stanford University. **Email:** renyuanxu@stanford.edu

stress testing and scenario analysis. In these settings, we seek to simulate rare but high-impact events (such as extreme market movements, adverse clinical outcomes, or system overloads) that occur with very low probability under the empirical data distribution. Addressing such problems requires generative models that can reliably produce samples from rare-event regimes rather than typical scenarios.

Modern diffusion models are typically trained in a pretraining phase to approximate the unconditional data distribution. Concretely, a pretrained score-based diffusion model specifies a stochastic sampling dynamics of the form:

$$dY_t = \Big(\bar{f}(t, Y_t) + s_\theta(t, Y_t)\Big)dt + \bar{g}(t)dB_t,$$

where $\{B_t\}_{t\geq 0}$ is standard Brownian motion, $\bar{f}(t, y)$ and $\bar{g}(t)$ are prescribed model parameters, and $s_\theta(t, y)$ is a learned score function parametrized by θ. The function s_θ is trained offline, using samples from the data distribution, to approximate the score (the gradient of the log-density) of the forward diffusion process (See Section 2 for details). For a well-trained diffusion model, the terminal distribution of Y_T is expected to be close to the target data distribution $p_{\text{data}}(\cdot)$.

The pretraining of diffusion models is computationally intensive and generally requires access to large and representative datasets. As a result, such models are typically trained once to approximate the unconditional data distribution and subsequently reused across multiple downstream tasks using guidance or fine-tuning. Given a pretrained diffusion model, existing guidance mechanisms can be broadly categorized into several paradigms, including supervised fine-tuning with regularization (often referred to as soft guidance) [51, 83, 86], conditioning-based approaches [20, 38], reinforcement learning–based methods [25, 26, 31, 55, 73, 98, 99], and more recent preference-optimization frameworks [90, 95]. While soft-constrained or reward-based guidance methods are often computationally convenient, they incorporate target criteria through penalty or reward terms in the optimization objective and, as a consequence, generally do not guarantee that the resulting generated distribution is supported on the constraint set or that the constraint is satisfied with probability one. As a result, such methods may generate samples that violate prescribed constraints, which is a critical limitation in applications requiring strict constraint adherence or reliable rare-event simulation.

In contrast, hard conditioning aims to generate samples by replacing the original generative law with the conditional law given a prescribed constraint set, under which the terminal output satisfies the constraint with probability one. This setting is substantially more challenging, particularly when the conditioning event is rare under the data distribution. In such cases, the conditional law concentrates on low-probability or geometrically thin regions of the data domain, rendering naive rejection sampling or importance reweighting ineffective. Enforcing hard constraints therefore requires modifying the entire generative trajectory, rather than only its terminal distribution. Designing a principled, theoretically grounded, and lightweight post-training mechanism that achieves this goal—without modifying the underlying pretrained score function—remains an *open problem* in the diffusion model literature.

Our work and contribution. This paper develops a principled framework for Conditional Diffusion Guidance that directly addresses these challenges. Our approach is rooted in classical stochastic analysis and exploits the probabilistic structure underlying diffusion models. Specifically, we adopt a conditioning-based perspective closely related to Doob's h-transform, which characterizes the law of a diffusion process conditioned on a terminal constraint via a change of measure. In this formulation, the central object is the function:

$$h(t, y) = \mathbb{P}(Y_T \in S | Y_t = y)$$

where S denotes the constraint set. The function h represents the conditional probability under the pretrained diffusion dynamics, and induces a drift correction that enforces the constraint.

Namely, the resulting guided dynamics take the form:

$$dY_t^S = \Big(\bar{f}(t, Y_t^S) + s_\theta(t, Y_t^S) + \overline{g}(t)^2 \nabla \log h(t, Y_t^S)\Big) dt + \overline{g}(t) d\overline{B}_t.$$

where $\{\overline{B}_t\}_{t\geq 0}$ is a copy of Brownian motion. We provide theoretical justification for this construction by proving that the terminal output satisfies

$$Y_T^S \sim (Z \mid Z \in S) \text{ with } Z \sim p_{\text{data}}(\cdot).$$

Importantly, the proposed guidance mechanism does not modify the pretrained score network s_θ; instead, it augments the sampling dynamics with an additive guidance term $\nabla \log h$, making the method lightweight in both training and implementation.

Although Doob's transform has been explored in the literature for guidance, existing approaches typically rely on stochastic control or reinforcement-learning formulations to enforce soft constraints. In contrast, our viewpoint is fundamentally probability-theoretic: we exploit martingale properties as well as quadratic-variation identities to derive learning objectives for the h-function and its gradient directly from the dynamics of the pretrained model. This leads to a powerful learning framework that is effective in forcing hard constraints.

Specifically, leveraging the *(local) martingale property* of $h(t, Y_t)$, we propose the Conditional Diffusion Guidance via Martingale Loss (CDG-ML) algorithm, which learns $h(\cdot, \cdot)$ by minimizing the L_2 loss

$$\min_{\ell(\cdot,\cdot)} \mathbb{E}_{[0,T]} \left[\int_0^T \Big(\ell(t, Y_t) - 1(Y_T \in S)\Big)^2 dt \right].$$

This is because $h(\cdot, \cdot)$ is the unique minimizer of the above objective function (under suitable conditions) and $\mathbb{E}_{[0,T]}[\cdot]$ is the expectation with respect to the law of the process $\{Y_t\}_{0\leq t\leq T}$.

However, learning a good approximation to h alone does not guarantee a good approximation of $\nabla \log h = \frac{\nabla h}{h}$, a difficulty that is well documented in practice. To address this challenge, we introduce a novel approach that learns ∇h directly via quadratic variation. Observing the quadratic variational process follows $d[h, Y]_t = \overline{g}(t)^2 \nabla h(t, Y_t) dt$, we propose the Conditional Diffusion Guidance via Martingale–Covariation Loss (CDG-MCL):

$$\min_{q(\cdot,\cdot)} \mathbb{E}_{[0,T]} \left[\int_0^T \left(\frac{1}{\overline{g}(t)^2} \frac{d[h, Y]_t}{dt} - q(t, Y_t) \right)^2 \right],$$

which estimates ∇h via quadratic-variation information.

Both CDG-ML and CDG-MCL operate solely on samples from the pretrained diffusion, require no access to the underlying data distribution, and are grounded in tools from stochastic analysis. To the best of our knowledge, this combination of martingale and covariation-based learning objectives for diffusion guidance is novel in the literature. In practice, we parameterize $l(\cdot, \cdot)$ and $q(\cdot, \cdot)$ in both loss functions using neural networks.

A central contribution of this work is a rigorous theoretical analysis of the proposed framework. We quantify the discrepancy between the target conditional data distribution and the distribution induced by the learned guided dynamics using two complementary metrics: total variation distance and Wasserstein distance. In particular, we establish nonasymptotic total variation bounds that decompose the error into contributions from (i) the pretrained model approximation and (ii) the learned guidance error (Lemmas 4.2 and 4.3), leading to an explicit end-to-end guarantee for conditional sampling (Theorem 4.4). Total variation provides strong distributional guarantees under minimal structural assumptions. In contrast, under additional regularity and stability conditions, we derive Wasserstein-2 error bounds for the guided dynamics (Theorem 4.7), which offer a geometrically meaningful notion of error closely tied to contraction and stability properties of stochastic differential equations. Together, these results elucidate the trade-offs between statistical strength and analytical tractability in conditional diffusion guidance.

To complement the theoretical analysis, we present numerical experiments that investigate the effectiveness of the proposed Conditional Diffusion Guidance framework in enforcing hard constraints and generating rare-event scenarios. The experiments illustrate how guidance learned from pretrained diffusion trajectories reshapes the sampling dynamics to concentrate on low-probability regions of the data distribution that are inaccessible to naive sampling. We compare the CDG-ML and CDG-MCL algorithms in terms of constraint satisfaction, sampling stability, and distributional accuracy, thereby highlighting the practical implications of learning the guidance function versus its gradient. These results provide empirical support for the theoretical guarantees and demonstrate the applicability of the proposed framework to stress testing and rare-event simulation tasks.

Closely related literature. In this paper, we focus on supervised guidance via endogenous conditioning. The underlying idea is closely related to Doob's h-transform, with the central task being the estimation of the conditioning function h. Related works based on the h-transform framework, such as [19, 23, 67, 42], formulate the problem from a stochastic control perspective and treat $\frac{\nabla h}{h}$ as the optimal control to be learned. In contrast, our approach is fundamentally probability-theoretic, relying on martingale properties and quadratic-variation identities of the pretrained diffusion. A key limitation of control-based conditional diffusion methods is that they are *inherently on-policy*: learning the guidance requires simulating trajectories under the evolving post-training dynamics. As a consequence, the data distribution used for learning depends on the current control approximation, introducing distribution shift and feedback effects that complicate stability analysis and the separation of modeling error from guidance error. Our framework avoids this issue by learning the conditioning function entirely off-policy using trajectories from the fixed pretrained diffusion, thereby decoupling learning from sampling. In addition, different from our rigorous theoretical foundation, none of these control-based papers provides approximation guarantees for their proposed conditional sampling methods.

Our work is closely related to the literature on classifier guidance [4, 20, 74, 87], in which an auxiliary classifier/label is used to steer diffusion sampling towards desired attributes or outcomes. This line of work includes both algorithmic developments [5, 63] and recent efforts to understand the statistical properties of classifier-guided diffusion [96]. Classifier guidance also provides an important conceptual link to reinforcement learning from human feedback (RLHF), where human preference or reward models play a role analogous to classifiers by shaping the sampling dynamics of generative models [38, 93]. For broader overviews of fine-tuning diffusion models using RLHF, we refer to [13, 15, 36, 60, 85, 92, 95, 97].

Our framework is also closely connected to pluralistic alignment [79, 57][1], in which diffusion guidance is driven by multidimensional reward or preference structures (see (3.1) and the discussion thereafter). From a learning perspective, this connection places our work in close relation to multi-task learning, where multiple objectives or constraints are incorporated within a unified modeling framework.

Finally, the targeted applications are closely related to the literature on rare-event simulation [3, 11], where the objective is to efficiently sample from low-probability regimes, often beyond the reach of classical importance sampling techniques [8, 9]. There is also growing interest in the use of diffusion models for problems in operations research and stochastic simulation; see, for example, [58].

Organization of the paper. The remainder of the paper is organized as follows. We start with background on diffusion models in Section 2. In Section 3, we build the foundations

[1] Pluralistic alignment refers to multi-objective alignment aimed at integrating complex and potentially conflicting real-world values; see `https://pluralistic-alignment.github.io/` for recent developments.

for conditional diffusion guidance, leading to novel methodologies. In Section 4, we provide theoretical results of the proposed methodologies. Extensions of the proposed methods are discussed in Section 5. Numerical experiments are reported in Section 6. We conclude with Section 7.

Notations. Below we collect a few notations that will be used throughout.

- $\mathbb{R}$ is the set of real numbers.
- For $x, y \in \mathbb{R}^d$, $x \cdot y$ denotes the scalar product of x and y, and $|x| := \sqrt{x \cdot x}$ is the Euclidean norm of x.
- For $A = (a_{ij})_{1 \le i,j \le d}$ a matrix, $|A|_F := \sqrt{\sum_{i,j=1}^d a_{ij}^2}$ denotes its Frobenius norm.
- For f a function on X, $|f|_\infty := \sup_X |f(x)|$ denotes its sup-norm.
- For $f : [0, \infty) \times \mathbb{R}^d \ni (t, x) \to \mathbb{R}$, $\partial_t f$ denotes its time derivative, ∇f is the gradient of f and $\partial_k f := \frac{\partial f}{\partial x_k}$ its k^{th} coordinate, and $\Delta f := \sum_{k=1}^d \frac{\partial^2 f}{\partial x_k^2}$ is the Laplacian of f.
- For Z a random variable, $\mathbb{E}Z$ denotes the expectation of Z.
- Let $p(\cdot)$ and $q(\cdot)$ be two probability distributions defined on the same measurable space. The total variation distance between p and q is defined by $d_{\mathrm{TV}}(p, q) := \sup_A \big|p(A) - q(A)\big|$. The Kullback–Leibler (KL) divergence from q to p is given by $d_{\mathrm{KL}}(p, q) := \int \log\left(\frac{dp}{dq}\right) dp$, whenever p is absolutely continuous with respect to q. Finally, the Wasserstein–2 distance between p and q is defined as $W_2(p, q) := \left(\inf_\gamma \mathbb{E}_{(X,Y)\sim\gamma}\left[|X - Y|^2\right]\right)^{1/2}$, where the infimum is taken over all couplings γ of p and q.

We use C for a generic constant whose values may change from line to line.

2 Background on diffusion models

This section provides preliminaries of score-based diffusion models. We follow the presentation of [81]. Diffusion models rely on a forward-backward procedure: the forward process transforms the target data to noise, and the backward process recovers the data from noise.

Let $p_{\mathrm{data}}(\cdot) \in \mathcal{P}(\in \mathbb{R}^d)$ be the target data distribution. Fixing $T > 0$, the forward process $\{X_t\}_{0 \le t \le T}$ is governed by the stochastic differential equation (SDE):

$$dX_t = f(t, X_t)dt + g(t)dW_t, X_0 \sim p_{\mathrm{data}}(\cdot), \tag{2.1}$$

where $f : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^d$, $g : \mathbb{R}_+ \to \mathbb{R}_+$, and $\{W_t\}_{t \ge 0}$ is Brownian motion in $\mathbb{R}^d$. We require some usual measurable conditions on $f(\cdot, \cdot)$ and $g(\cdot)$ so that the SDE (2.1) is well-defined, and that X_t has a smooth probability density $p(t, x) := \mathbb{P}(X_t \in dx)/dx$ (see [80]). By the time reversal formula [1, 35], let

$$d\overline{X}_t = \overline{f}^\circ(t, \overline{X}_t)dt + g(T - t)dB_t, \overline{X}_0 \sim p(T, \cdot), \tag{2.2}$$

whose drift is:

$$\overline{f}^\circ(t, y) = -f(T - t, y) + \overline{g}(t)^2 \nabla \log p(T - t, y). \tag{2.3}$$

Here $\{B_t\}_{t \ge 0}$ is an independent Brownian motion in $\mathbb{R}^d$. The processes $\{\overline{X}_t\}_{0 \le t \le T}$ and $\{X_{T-t}\}_{0 \le t \le T}$ have the same marginal distribution [1, 35]. Hence the output $\overline{X}_T$ follows the desirable distribution $p_{\mathrm{data}}(\cdot)$.

Remark 2.1 (Examples)**.** *In practice, notable examples include the* Variance Exploding *(VE) model [49] and* Variance Preserving *(VP) model [78], which serves as a demonstrative case throughout the paper:*

- *(VE): $f(t,x)=0$, $g(t)=\sqrt{2t+\epsilon}$ for some (small) $\epsilon>0$.*
- *(VP): $f(t,x)=-\frac{1}{2}\left(a+\frac{(b-a)t}{T}\right)x$, $g(t)=\sqrt{a+\frac{(b-a)t}{T}}$ for some $b>a>0$.*

The main challenge in sampling the process $\overline{X}$ is that the data distribution $p_{\text{data}}(\cdot)$ is unknown, and consequently the associated score function $\nabla\log p(\cdot,\cdot)$ is not available. Moreover, the initialization $\overline{X}_0\sim p(T,\cdot)$ required for each sample generation also depends on $p_{\text{data}}(\cdot)$. The idea of score-based diffusion models is to learn the score function via a parametrized family of functions $\{s_\theta(t,x)\}_\theta$ (e.g., neural networks), with a limited number of samples from $p_{\text{data}}(\cdot)$ (see [37, 76, 78]). The resulting backward process $\{Y_t\}_{0\le t\le T}$ for sampling is:

$$dY_t=\overline{f}(t,Y_t)dt+\overline{g}(t)dB_t, Y_0\sim p_{\text{noise}}(\cdot), \tag{2.4}$$

where $\overline{g}(t):=g(T-t)$ and

$$\overline{f}(t,y):=-f(T-t,y)+g(T-t)^2 s_{\theta_*}(T-t,y), \tag{2.5}$$

serves as an approximation to $\overline{f}^{\circ}$, which is defined in (2.3). Here,

- $p_{\text{noise}}(\cdot)$ is a proxy to $p(T,\cdot)$ for generating the target distribution from noise, which should *not* depend on $p_{\text{data}}(\cdot)$. The form of $p_{\text{noise}}(\cdot)$ is related to the design of the diffusion model, i.e., the pair $(f(\cdot,\cdot),g(\cdot))$. In practice, we usually take $p_{\text{noise}}(\cdot)$ as $\mathcal{N}(0,T^2I)$ for the VE model and $p_{\text{noise}}(\cdot)$ as $\mathcal{N}(0,I)$ for the VP model.
- $\{s_\theta(t,x)\}_\theta$ are function approximations to the score $\nabla\log p(t,x)$, which are trained by solving some stochastic optimization schemes. This technique is known as *score matching*, and the learned $s_{\theta_*}(t,x)$ is the *score matching function.* There are several existing score matching methods, among which the most widely used one is the *denoising score matching* (DSM) [43, 88]:

$$\min_\theta \int_0^T \lambda(t)\,\mathbb{E}_{X_0\sim p_{data}(\cdot)}\left[\mathbb{E}_{X_t|X_0}\left|s_\theta(t,X_t)-\nabla\log p(t,X_t|X_0)\right|^2\right]dt,$$

  where $\lambda:\mathbb{R}_+\to\mathbb{R}_+$ is a weight function.

Most successful diffusion models rely on extensive score-matching training on large, generic datasets and are therefore trained once in advance and reused across downstream tasks; such models are commonly referred to as *pretrained models.* [2]. Throughout, we let the backward process (2.4) represent such a pretrained model. Denote by $P_{[0,T]}(\cdot)$ the law of the process $\{Y_t\}_{0\le t\le T}$ on path space under which $\mathbb{E}_{[0,T]}[\cdot]$ is the associated expectation, and by $P_t(\cdot)$ its marginal distribution at time t with $\mathbb{E}_t[\cdot]$ the associated expectation. We write

$$p_{\text{pre}}(\cdot):=P_T(\cdot),$$

to denote the output distribution of the pretrained model. A well-trained model is expected to generate reliable samples in the sense that $p_{\text{pre}}(\cdot)$ provides a good approximation of the data distribution $p_{\text{data}}(\cdot)$. Under suitable conditions on the model components $f(\cdot,\cdot),g(\cdot),T,p_{\text{noise}}(\cdot),s_\theta(\cdot,\cdot)$ and on the target distribution $p_{\text{data}}(\cdot)$, this approximation can be made quantitative by bounding appropriate discrepancies between $p_{\text{pre}}(\cdot)$ and $p_{\text{data}}(\cdot)$. The following proposition bounds the total variation and the Wasserstein distance between $p_{\text{pre}}(\cdot)$ and $p_{\text{data}}(\cdot)$ for the aforementioned VE and VP models.

[2] It was shown in [49] that a wide class of diffusion models can be obtained from the VE model via reparameterization. Consequently, in practice it suffices to pretrain a high-quality VE model, that is, to learn its score function with sufficient accuracy.

Proposition 2.2. *Assume that* $\mathbb{E}_{p_{\mathrm{data}}(\cdot)}|X|^2 < \infty$*, and that the score matching satisfies:*

$$\sup_{0 \le t \le T} \mathbb{E}_{p(t,\cdot)}|s_{\theta_*}(t,X) - \nabla \log p(t,X)|^2 \le \varepsilon_{\mathit{VP}}^2 (\textit{resp. } \varepsilon_{\mathit{VE}}^2), \tag{2.6}$$

for the VE (resp. VP) model. Then the following results hold.

1. *(Total variation) [16, 81] There are* $C_{VE}, C_{VP} > 0$ *(independent of* T*) such that*

$$d_{TV}\left(p_{\mathrm{pre}}(\cdot), p_{\mathrm{data}}(\cdot)\right) \le \begin{cases} C_{\mathit{VE}}\left(T^{-1}\sqrt{\mathbb{E}_{p_{\mathrm{data}}(\cdot)}|X|^2} + \varepsilon_{\mathit{VE}}\sqrt{T}\right) & \textit{for VE,} \\ C_{\mathit{VP}}\left(e^{-C_{\mathit{VP}}T}\sqrt{\mathbb{E}_{p_{\mathrm{data}}(\cdot)}|X|^2} + \varepsilon_{\mathit{VP}}\sqrt{T}\right) & \textit{for VP.} \end{cases} \tag{2.7}$$

2. *(Wasserstein) [30, 81] Assume further that* $p_{\mathrm{data}}(\cdot)$ *is* κ*-strongly log-concave* [3] *for* κ *sufficiently large. There are* $C_{VE}, C_{VP} > 0$ *(independent of* T*) such that*

$$W_2\left(p_{\mathrm{pre}}(\cdot), p_{\mathrm{data}}(\cdot)\right) \le \begin{cases} C_{\mathit{VE}}\left(T^{-1}\sqrt{\mathbb{E}_{p_{\mathrm{data}}(\cdot)}|X|^2} + \varepsilon_{\mathit{VE}}T^2\right) & \textit{for VE,} \\ C_{\mathit{VP}}\left(e^{-C_{\mathit{VP}}T}\sqrt{\mathbb{E}_{p_{\mathrm{data}}(\cdot)}|X|^2} + \varepsilon_{\mathit{VP}}\right) & \textit{for VP.} \end{cases} \tag{2.8}$$

The condition (2.6) formalizes a black-box score-matching error, an assumption that is standard in the recent literature on the convergence analysis of diffusion models [16, 30, 54, 56, 82]. In this line of work, the statistical and algorithmic aspects of score estimation are treated as a separate problem, and the learned score function is taken as an approximate oracle whose error is summarized by (2.6). Complementary to this approach, there exists a body of work [14, 34, 53, 91] that derives quantitative rates for score matching itself, typically under structural assumptions such as low-dimensional or parametric representations of the score function class $s_\theta(t,x)_\theta$. However, state-of-the-art score-based diffusion models in practice rely on highly expressive and deep neural network architectures [49, 78], for which such structural assumptions are difficult to justify. For this reason, and in line with the convergence-focused literature, we adopt the black-box score-matching assumption (2.6) throughout this paper.

3 Diffusion guidance by conditioning

In this section, we introduce a *Conditional Diffusion Guidance* framework and establish the mathematical foundations. Our focus is on enforcing *hard constraints* on the generated samples, as opposed to encouraging desirable outcomes through soft reward signals.

Let $S \subset \mathbb{R}^d$ be a Borel set, which we refer to as the *guidance set*, and denote by $p^S_{\mathrm{data}}(\cdot)$ the target distribution conditioned on S. That is, for a random variable $Z \sim p_{\mathrm{data}}(\cdot)$,

$$(Z \mid Z \in S) \sim p^S_{\mathrm{data}}(\cdot).$$

The set S encodes the event or structural constraint of interest and may represent a wide range of conditions on the data, including discrete labels, functional constraints, rare events, or application-specific admissible regions. For example, in time-series generation, S may characterize regimes exhibiting certain statistical or dynamical properties, such as high or low volatility periods, prescribed terminal values, or specific temporal patterns. Throughout, we assume that S is sufficiently regular (e.g., Borel and non-negligible) so that the conditional distribution $p^S_{\mathrm{data}}(\cdot)$ is well-defined.

From a modeling perspective, conditioning on S imposes a *hard constraint*, in the sense that generated samples are required to satisfy the event $\{Z \in S\}$ almost surely. This should

[3] A smooth function $\ell : \mathbb{R}^d \to \mathbb{R}$ is κ-strongly log-concave if $\langle \nabla \log \ell(x) - \nabla \log \ell(y), (x-y)\rangle \le -\kappa|x-y|^2$ for all x, y.

be contrasted with the more commonly studied *soft conditioning* or reward-based guidance approaches. While soft constraints are often easier to optimize, they generally do not guarantee constraint satisfaction and may yield samples that violate physical laws, feasibility requirements, or operational rules. Hard conditioning is substantially more challenging, particularly when the event $\{Z \in S\}$ is rare under the data distribution, causing the conditional law $p^S_{\mathrm{data}}(\cdot)$ to concentrate on low-probability or geometrically thin regions of the state space. In such settings, naive rejection sampling or reward-based reweighting becomes ineffective, and enforcing the constraint typically requires modifying the entire generative trajectory rather than only its terminal distribution. These considerations motivate a conditioning-based framework that directly targets the conditional law associated with the event $\{Z \in S\}$.

Our framework can be easily extended to settings in which the guidance set S is specified implicitly through a measurable functional $F : \mathbb{R}^d \to \mathcal{Y}$, so that the conditioning event takes the form

$$F(Y_T) \in S. \tag{3.1}$$

This formulation covers constraints such as conditions on cumulative or averaged quantities, path-dependent risk measures, tail events of aggregate observables, and system-level performance metrics arising in operations research and related application domains. It also accommodates multi-objective guidance by taking $F = (F_1, \cdots, F_K) : \mathbb{R}^d \to \mathbb{R}^K$ and imposing hard constraints $F_k(Y_T) \in S_k$ for each objective. Pluralistic alignment can be modeled by allowing multiple acceptable guidance sets $\{S^{(j)}\}_{j=1}^K \subset \mathbb{R}^K$, corresponding to distinct alignment criteria, and targeting a mixture of the resulting conditional distributions.

The problem of interest is to exploit a given pretrained models to generate samples that approximate the conditional distribution $p^S_{\mathrm{data}}(\cdot)$. Recall that $p_{\mathrm{pre}}(\cdot)$ denotes the output distribution of the pretrained model (2.4), and let $p^S_{\mathrm{pre}}(\cdot)$ be its conditioning on S. As discussed in Section 2, a well-trained pretrained model yields $p_{\mathrm{pre}}(\cdot) \approx p_{\mathrm{data}}(\cdot)$, which in turn implies $p^S_{\mathrm{pre}}(\cdot) \approx p^S_{\mathrm{data}}(\cdot)$ under mild assumptions on the guidance set S (see Section 4). Therefore, our goal is to use the pretrained model to sample from $p^S_{\mathrm{pre}}(\cdot)$, or more precisely, to generate sample paths

$$\{Y^S_t\}_{0 \le t \le T} \sim P_{[0,T]}(Y \mid Y_T \in S). \tag{3.2}$$

3.1 Conditional diffusion sampling

We start by discussing how the process $\{Y^S_t\}_{0 \le t \le T}$ is generated. To make the framework meaningful, assume that $P_{[0,T]}(Y_T \in S) > 0$. For $t \ge 0$, let

$$h(t, y) := P_{[0,T]}(Y_T \in S \,|\, Y_t = y). \tag{3.3}$$

By the Bayes rule, we have for $s > 0$,

$$\begin{aligned} P_{[0,T]}(Y_{t+s} = y' \,|\, Y_t = y,\, Y_T \in S) &= \frac{P_{[0,T]}(Y_{t+s} = y', Y_T \in S \,|\, Y_t = y)}{P_{[0,T]}(Y_T \in S \,|\, Y_t = x)} \\ &= \frac{h(t+s, y')}{h(t, y)} P_{[0,T]}(Y_{t+s} = y' \,|\, Y_t = y), \end{aligned} \tag{3.4}$$

which is known as *Doob's h-transform* [4]. Note that $h(t, y) := P_{[0,T]}(Y_T \in S \,|\, Y_t = y)$ is harmonic with respect to the generator of $\{Y_t\}_{0 \le t \le T}$ which satisfies the following second-order PDE:

$$\partial_t h + \overline{f}(t, y) \cdot \nabla h + \frac{1}{2} \overline{g}(t)^2 \Delta h = 0, \tag{3.5}$$

[4] Doob's h-transform is a general concept of conditioning a Markov process, provided that h is harmonic with respect to the Markov generator. In our setting, the h function (3.3) arises as a special case by conditioning on the terminal data. The "bridge" calculation in (3.4) can be understood in terms of transition densities, see [27, Section 2] for a justification.

with the terminal condition $h(T, \cdot) = 1(\cdot \in S)$. It is well known that the solution to the linear parabolic equation (3.5) is smooth in $[0, T) \times \mathbb{R}^d$ if $\overline{g}(\cdot)$ is bounded away from zero, and $\overline{f}(\cdot, \cdot)$, $\overline{g}(\cdot)$ are Hölder continuous (see [29, Chapter 3, Theorem 5] or [44, Theorem 6] for the interior a priori estimate).

The h-transform (3.4) reveals a change of measure between the processes $\{Y_t\}_{0 \le t \le T}$ and $\{Y_t^S\}_{0 \le t \le T}$. As a result, $\{Y_t^S\}_{0 \le t \le T}$ is also a diffusion process, whose dynamics is given in the following proposition.

Proposition 3.1. *The distribution of* $\{Y_t^S\}_{0 \le t < T}$ *is governed by:*

$$dY_t^S = \left(\overline{f}(t, Y_t^S) + \overline{g}(t)^2 \nabla \log h(t, Y_t^S)\right) dt + \overline{g}(t) d\overline{B}_t, Y_0^S \sim p_{\text{noise}}(\cdot), \tag{3.6}$$

where $\overline{B}_t$ *is an* $\mathcal{F}$*-adapted Brownian motion;* $\overline{f}(\cdot, \cdot)$ *and* $\overline{g}(\cdot)$ *are given by the pretrained model* (2.4)*. Define* $Y_T^S = \lim_{t \to T} Y_t^S$*, then we have*

$$Y_T^S \sim p_{\text{pre}}^S(\cdot). \tag{3.7}$$

The proof for (3.6) is a direct application of (3.4) and Girsanov's theorem (see [71, IV.39] and [48, Theorem 2]). The key is to prove (3.7) using a limiting argument.

Proof. By standard PDE estimates, we have for each $\varepsilon > 0$, $h \in C^{1,2}([0, T - \varepsilon] \times \mathbb{R}^d)$ and $h(t, y) > 0$ on $[0, T - \varepsilon] \times \mathbb{R}^d$. In addition,

$$\mathbb{E}_P \left[\exp\left(\frac{1}{2} \int_0^{T-\varepsilon} \|g(t) \nabla \log h(t, Y_t)\|^2 dt \right) \right] < \infty. \tag{3.8}$$

Let $\Omega := C([0, T]; \mathbb{R}^d)$, $Y_t(\omega) := \omega(t)$ be the coordinate process and $\mathcal{F}_t = \sigma(Y_s : s \le t)$ the canonical filtration.

First, let us defined the normalized terminal indicator $Z := \frac{1(Y_T \in S)}{P_{[0,T]}(Y_T \in S)}$. For $t \in [0, T]$, set $M_t = \mathbb{E}[Z | \mathcal{F}_t]$. Then

$$\mathbb{E}_P[Z | \mathcal{F}_t] = \frac{1}{P_{[0,T]}(Y_T \in S)} P_{[0,T]}(Y_T \in S | \mathcal{F}_t) = \frac{1}{P_{[0,T]}(Y_T \in S)} h(t, Y_t),$$

using the Markov property. Morevoer, since (M_t) is uniformly integrable martingale (bounded by $\frac{1}{P_{[0,T]}(Y_T \in S)}$), we have

$$M_t \to M_T = Z \in L^1 \text{ and } a.s., \text{ as } t \to T. \tag{3.9}$$

For each $\varepsilon > 0$, define a probability measure on $\mathcal{F}_{T-\varepsilon}$ by

$$Q^\varepsilon(A) := \mathbb{E}_P[1_A M_{T-\varepsilon}], \qquad A \in \mathcal{F}_{T-\varepsilon}.$$

If $0 < \varepsilon' < \varepsilon$, then Q^ε and $Q^{\varepsilon'}$ agree on $\mathcal{F}_{T-\varepsilon}$:

$$Q^{\varepsilon'}(A) = \mathbb{E}_P[1_A M_{T-\varepsilon'}] = \mathbb{E}_P[1_A \mathbb{E}_P[M_{T-\varepsilon'} | \mathcal{F}_{T-\varepsilon}]] = Q^\varepsilon(A) \tag{3.10}$$

since $\{M_t\}_{t \ge 0}$ is a martingale. By Carathéodory's extension theorem on the increasing sigma-fields, there is a unique probability measure Q on $\mathcal{F}_{T-} := \sigma(\cup_{t<T} \mathcal{F}_t)$ such that $Q|_{\mathcal{F}_{T-\varepsilon}} = Q^\varepsilon$ for all $\varepsilon > 0$.

Let $A \in \mathcal{F}_{T-}$, by taking a sequence $A_n \in \mathcal{F}_{t_n}$ with $t_n \to T$ such that $1_{A_n} \to 1_A$ in L^1, we can show that

$$Q(A_n) = \mathbb{E}_P[1_{A_n} M_{t_n}] = \frac{P_{[0,T]}(A_n \cap \{Y_T \in S\})}{P_{[0,T]}(Y_T \in S)}$$

Let $n \to \infty$ and use dominated convergence on both sides (everything bounded) to obtain

$$Q(A) = P_{[0,T]}(A|Y_T \in S) \tag{3.11}$$

Define the conditional process Y^S as the coordinate process under Q: on canonical path space Ω with coordinate map $Y_t(\omega) = \omega(t)$, set

$$Y_t^S := Y_t, \text{ as a random variable on } (\Omega, \mathcal{F}_{T-}, Q), \qquad 0 \le t < T \tag{3.12}$$

Fix $\varepsilon > 0$ and for $t \in [0, T-\varepsilon]$, applying Itô's, (3.5) and (3.9) give

$$dM_t = M_t g(t) \nabla \log h(t, Y_t) \cdot dB_t$$

Thanks to (3.8), Girsanov can be applied on $[0, T-\varepsilon]$. In addition, $B_t^Q := B_t - \int_0^t g(s) \nabla \log h(s, Y_s) ds$ is a Q-Brownian motion for $t \le T - \varepsilon$. Substituting into (2.4), for $t \le T - \varepsilon$,

$$dY_t = (\overline{f}(t, Y_t) + g(t)^2 \nabla \log h(t, Y_t))dt + g(t) dB_t^Q \tag{3.13}$$

Using the identification (3.12), this is exactly the claimed dyamics for Y^S on $[0, T-\varepsilon]$ in (3.6), and since $\varepsilon > 0$ is arbitrary, it holds for all $t < T$.

Finally, define the terminal value by $Y_T^S := \lim_{t \to T} Y_t^S$ which exists Q-a.s by path continuity. Moreover, for any Borel $A \in \mathbb{R}^d$, the event $\{Y_T^S \in A\}$ belongs to $\mathcal{F}_{T-}$ because $Y_T^S := \lim_{n \to \infty} Y_{T-1/n}^S$

Hence, by applying (3.11) with A replaced by $\{Y_T^S \in A\}$ yields

$$Q(Y_T^S \in A) = P_{[0,T]}(Y_T \in A | Y_T \in S), \qquad A \in \mathcal{B}(\mathbb{R}^d).$$

This completes the proof.

□

As noted in [86, Appendix B], the dynamics (3.6) can be interpreted as classifier-guided diffusion sampling [20] in continuous time. Let $P_{[0,T]}^S(\cdot)$ denote the law of the guided process $\{Y_t^S\}_{0 \le t \le T}$, and $P_t^S(\cdot)$ its marginal distribution at time t. By (3.7), simulating the dynamics (3.6) produces a terminal sample

$$Y_T^S \sim P_T^S(\cdot) := p_{\text{pre}}^S(\cdot).$$

Given a pretrained model, the only unknown component in (3.6) is the guidance term $\nabla \log h(\cdot, \cdot)$, which enters the dynamics directly.

3.2 Learning h function

As explained in Section 3.1, the key challenge in sampling the guided process $\{Y_t^S\}_{0 \le t \le T}$ lies in learning the function h, or more precisely, its logarithmic gradient $\nabla \log h$. Since $\nabla \log h = (\nabla h)/h$, a good approximator h_θ to h does not guarantee that $\nabla \log h_\theta$ is close to $\nabla \log h(\cdot, \cdot)$. Hence, a natural strategy is to learn the numerator ∇h and the denominator h separately. Accordingly, Section 3.2.1 focuses on learning h, while Section 3.2.2 addresses the estimation of ∇h. The main analytical tools we employ are drawn from stochastic analysis, in particular martingale theory and the quadratic variation of stochastic processes.

3.2.1 Learning h via martingale loss

By applying Itô's formula to $h(t, Y_t)$, we get:

$$dh(t, Y_t) = \overline{g}(t)\nabla h(t, Y_t) \cdot dB_t, \tag{3.14}$$

which leads to the following classical result.

Proposition 3.2. *Let $\{Y_t\}_{0\le t\le T}$ be the pretrained model defined by* (2.4)*, and $h(\cdot,\cdot)$ be defined by* (3.3)*. Then the process $\{h(t, Y_t)\}_{0\le t\le T}$ is a local martingale.*

A natural idea is to exploit the (local) martingale property [5] of $\{h(t, Y_t)\}_{0\le t\le T}$ to learn the function h. Assume that $\{h(t, Y_t)\}_{0\le t\le T}$ is a (true) martingale. By the L^2 projection of conditional expectations, the h function (3.3) *uniquely* solves the following optimization problem:

$$\min_{\ell(\cdot,\cdot)\in\mathcal{K}} \mathbb{E}_{[0,T]}\left[\int_0^T \Big(\ell(t, Y_t) - 1(Y_T \in S)\Big)^2 dt\right].$$

where

$$\mathcal{K} := \left\{\ell : [0, T] \times \mathbb{R}^d | \ell \text{ is Borel measurable and } \mathbb{E}\Big[\int_0^T \ell(t, Y_t)^2\Big] < \infty\right\}.$$

Now we restrict our search within a class of parametrized functions $\{h_\phi(t, y)\}_\phi$ to approximate $h(t, y)$. This leads to the *martingale loss* objective to learn the h function:

$$\min_\phi \mathbb{E}_{[0,T]}\left[\int_0^T \Big(h_\phi(t, Y_t) - 1(Y_T \in S)\Big)^2 dt\right] \tag{3.15}$$

Remark 3.3 (Efficient off-policy learning)**.** *Note that our learning objective is purely off-policy, in the sense that it relies only on trajectories generated by the pre-trained diffusion model $\{Y_t\}_{0\le t\le T}$, rather than on on-policy dynamics induced by the currently learned control or guidance mechanism. The latter evolves during training and may be unstable over time, particularly when the imposed constraints correspond to rare events. This feature fundamentally distinguishes our approach from control-based methods that rely on Doob's h-transform for guidance, which inherently operate on on-policy controlled dynamics [19, 23, 67, 42]. Related off-policy ideas have recently been proposed and analyzed in [59] for a different purpose, namely to address observability challenges in reinforcement learning under model uncertainty.*

Denote by $h_{\phi_*}(t, y)$ the learned h function by solving the stochastic optimization problem (3.15). The algorithm for *conditional diffusion guidance via martingale loss* (CDG-ML) is summarized as follows.

The logic of Algorithm A is that if h_{ϕ_*} is a good approximation to h, then so is $\nabla \log h_{\phi_*}$ to $\nabla \log h$. This is not always true mathematically, but can still serve as a simple computational proxy to $\nabla \log h$.

3.2.2 Learning ∇h via quadratic variation

Recall that $\nabla \log h = \frac{\nabla h}{h}$. We have seen that the denominator h can be learned by means of the martingale loss (3.15). Now we explain how to learn its gradient ∇h by exploiting the quadratic variation of $\{h(t, Y_t)\}_{0\le t\le T}$. Recall from (3.14) that

$$dh(t, Y_t) = \overline{g}(t)\sum_{k=1}^d \partial_k h(t, Y_t) dB_t^k,$$

[5] Local martingales are a continuous-time phenomenon [62, Theorem II.42], i.e., there are no (strict) local martingales in discrete time. It is well known that a uniformly integrable local martingale is a (true) martingale, see [70, Chapter II]. In our setting, a sufficient condition for $\{h(t, Y_t)\}_{0\le t\le T}$ to be a (true) martingale is that $\overline{g}(\cdot)$, $|\nabla h(\cdot,\cdot)|$ are bounded.

```
Algorithm A Conditional diffusion guidance via martingale loss (CDG-ML)
  Input: pretrained model $\{Y_t\}_{0 \le t \le T}$ (2.4), guidance set $S$, parametrized family $\{h_\phi(t, y)\}_\phi$
  Step 1. Solve the stochastic optimization problem (3.15) that outputs $\phi_*$.
  Step 2. Sample
      $$dY_t^S = \left(\overline{f}(t, Y_t^S) + \overline{g}(t)^2 \nabla \log h_{\phi_*}(t, Y_t^S)\right) dt + \overline{g}(t) dB_t = \left(\overline{f}(t, Y_t^S) + \overline{g}(t)^2 \frac{\nabla h_{\phi_*}(t, Y_t^S)}{h_{\phi_*}(t, Y_t^S)}\right) dt + \overline{g}(t) dB_t, Y_0^S \sim p_{\text{noise}}(\cdot).$$
  Output: $Y_T^S$.
```

where $\{B_t^k\}_{0 \le t \le T}$ is the k-th coordinate of $\{B_t\}_{0 \le t \le T}$. Also denote by $\{Y_t^k\}_{0 \le t \le T}$ the k-th coordinate of the pretrained model $\{Y_t\}_{0 \le t \le T}$. For each $k = 1, \dots, d$, the covariation of $h(t, Y_t)$ and Y_t^k is:

$$d[h, Y^k]_t = \overline{g}(t)^2 \partial_k h(t, Y_t) dt, \tag{3.16}$$

see [70, Chapter IV] [6]. Put it compactly,

$$d[h, Y]_t = \overline{g}(t)^2 \nabla h(t, Y_t) dt. \tag{3.17}$$

By substituting h on the right side of (3.17) with h_{ϕ_*}, and approximating $\nabla h(t, y)$ by a class of parametrized functions $\{q_\psi(t, y)\}_\psi$, we derive the *covariation loss* objective:

$$\min_\psi \mathbb{E}_{[0,T]} \left[\int_0^T \left(\frac{1}{\overline{g}(t)^2} \frac{d[h_{\phi_*}, Y]_t}{dt} - q_\psi(t, Y_t) \right)^2 dt \right]. \tag{3.18}$$

Denote by $q_{\psi_*}(t, y)$ the learned gradient of h by solving the stochastic optimization problem (3.18). The algorithm for conditional diffusion guidance via martingale-covariation loss (CDG-MCL) is summarized as follows.

```
Algorithm B Conditional diffusion guidance via martingale-covariation loss (CDG-MCL)
  Input: pretrained model $\{Y_t\}_{0 \le t \le T}$ (2.4), guidance set $S$, parametrized families $\{h_\phi(t, y)\}_\phi$, $\{q_\psi(t, y)\}_\psi$
  Step 1. Solve the stochastic optimization problem (3.15) that outputs $\phi_*$.
  Step 2. Solve the stochastic optimization problem (3.18) that outputs $\psi_*$.
  Step 3. Sample
      $$dY_t^S = \left(\overline{f}(t, Y_t^S) + \overline{g}(t)^2 \frac{q_{\psi_*}(t, Y_t^S)}{h_{\phi_*}(t, Y_t^S)}\right) dt + \overline{g}(t) dB_t, Y_0^S \sim p_{\text{noise}}(\cdot).$$
  Output: $Y_T^S$.
```

In Algorithm B, we write $\nabla \log h = \frac{\nabla h}{h}$, and estimate the numerator ∇h and the denominator h separately. We first learn the h function via the martingale property of $\{h(t, Y_t)\}_{0 \le t \le T}$, and

[6] The equation (3.16) can be understood as:

$$\frac{(h(t+\delta, Y_{t+\delta}) - h(t, Y_t))(Y_{t+\delta}^k - Y_t^k)}{\overline{g}(t)^2 \delta} \approx \partial_k h(t, Y_t), \text{for sufficiently small } \delta > 0. \tag{$\star$}$$

There have been a body of works [6, 17, 24, 39, 40, 46] on the statistical estimation of quadratic variation. These papers consider how to estimate the quadratic variation of a process from a single trajectory in the context of (financial) time series. Here we have a different scenario, where the pretrained model $\{Y_t\}_{0 \le t \le T}$ under $P_{[0,T]}(\cdot)$ can be sampled repeatedly. The relation ($\star$) naturally provides an approximation to $\partial_k h$ by regressing the left side term over (t, Y_t).

then learn its gradient by using the (quadratic) covariation of $\{h(t, Y_t)\}_{0 \le t \le T}$ and $\{Y_t\}_{0 \le t \le T}$. Denote by $\widetilde{p}^S_{\text{pre}}(\cdot)$ the distribution of the output Y^S_T in Algorithm A or B.

4 Theoretical results

In this section, we provide theoretical results of the Conditional Diffusion Guidance framework introduced in Section 2. The total variation and Wasserstein distance between the conditional target distribution $p^S_{\text{data}}(\cdot)$, and the diffusion guidance $\widetilde{p}^S_{\text{pre}}(\cdot)$ output by Algorithm A or B are studied in Section 4.1 and 4.2 respectively. In Section 4.3, we explore the convergence of the stochastic optimization algorithms to learn the h function.

4.1 Total variation bounds

This part studies the total variation distance between $p^S_{\text{data}}(\cdot)$ and $\widetilde{p}^S_{\text{pre}}(\cdot)$.

Recall from Section 2 that $p(t, x)$ is the probability density of the forward process (2.1), and $s_{\theta_*}(t, x)$ is the score matching function of the pretrained model. Also recall from Section 3.1 that $P^S_t(\cdot)$ is the (marginal) distribution of Y^S_t defined by (3.6). Below we present a few assumptions.

Assumption 4.1.

(i) $d_{TV}(p(T, \cdot), p_{\text{noise}}(\cdot)) < \infty$.

(ii) The score matching satisfies: $\sup_{0 \le t \le T} \mathbb{E}_{p(t,\cdot)} |s_{\theta_*}(t, X) - \nabla \log p(t, X)|^2 \le \varepsilon^2$.

(iii) There is $\rho > 0$ *such that* $p_{\text{data}}(S) \ge \rho$.

(iv) There is $\eta > 0$ *such that* $|\nabla \log h - \nabla \log h_{\phi_*}|_\infty \le \eta$ *for Algorithm A, or* $\left|\nabla \log h - \frac{q_{\psi_*}}{h_{\phi_*}}\right|_\infty < \infty$ *for Algorithm B.*

The assumptions (i)-(ii) ensure that $p_{\text{data}}(\cdot)$ and $p_{\text{pre}}(\cdot)$ are close. The assumption (iii) indicates that the guidance set S is non-negligible, so the conditional distributions on S are well-defined. The assumption (iv) provides blackbox errors for learning $\nabla \log h$ [7], which will be further developed in Section 4.3.

First, we bound the total variation distance between $p^S_{\text{data}}(\cdot)$ and $p^S_{\text{pre}}(\cdot)$.

Lemma 4.2. *Let Assumption 4.1 (iii) hold. We have:*

$$d_{TV}(p^S_{\text{pre}}(\cdot), p^S_{\text{data}}(\cdot)) \le \frac{3}{2\rho} d_{TV}(p_{\text{pre}}(\cdot), p_{\text{data}}(\cdot)). \tag{4.1}$$

[7] The assumption (iv) means that the function $\nabla \log h$ can be learned pointwise. Note that in Algorithm A and B, the h function is learned by solving stochastic optimization problems using the pretrained samples under $P_{[0,T]}(\cdot)$. So a more "reasonable" hypothesis is that $\nabla \log h$ can be learned under the pretrained distribution:

$$\sup_{0 \le t \le T} \mathbb{E}_t |\nabla \log h(t, Y) - \nabla \log h_{\phi_*}(t, Y)| \le \eta^2 \quad \text{or} \quad \sup_{0 \le t \le T} \mathbb{E}_t \left|\nabla \log h(t, Y) - \frac{q_{\psi_*}(t, Y)}{h_{\phi_*}(t, Y)}\right|^2 \le \eta^2, \tag{$\star\star$}$$

which will be studied in Section 4.3. As will be clear in the proof of Lemma 4.3, we need (technically):

$$\sup_{0 \le t \le T} \mathbb{E}^S_t |\nabla \log h(t, Y) - \nabla \log h_{\phi_*}(t, Y)|^2 \le \eta^2 \text{ or } \sup_{0 \le t \le T} \mathbb{E}^S_t \left|\nabla \log h(t, Y) - \frac{q_{\psi_*}(t, Y)}{h_{\phi_*}(t, Y)}\right|^2 \le \eta^2.$$

to establish the total variation bound. That is, $\nabla \log h$ can be learned under the conditional guided distribution $P^S_{[0,T]}(\cdot)$. Our conjecture is that using sufficiently rich function approximations, the function $\nabla \log h$ can be learned pointwise, so it does not matter whether the evaluation is under $P_{[0,T]}(\cdot)$ or $P^S_{[0,T]}(\cdot)$.

Proof. Without loss of generality, assume that $p_{\text{pre}}(\cdot)$, $p_{\text{data}}(\cdot)$ have densities. We have:

$$\begin{aligned} d_{TV}(p^S_{\text{pre}}(\cdot),\, p^S_{\text{data}}(\cdot)) &= \frac{1}{2}\int |p^S_{\text{pre}}(x) - p^S_{\text{data}}(x)| dx \\ &= \frac{1}{2}\int_S \left| \frac{p_{\text{pre}}(x)}{p_{\text{pre}}(S)} - \frac{p_{\text{data}}(x)}{p_{\text{data}}(S)} \right| dx \\ &\le \frac{1}{2p_{\text{data}}(S)} \left(|p_{\text{pre}}(S) - p_{\text{data}}(S)| + \int_S |p_{\text{pre}}(x) - p_{\text{data}}(x)| dx \right) \\ &\le \frac{3}{2p_{\text{data}}(S)} d_{TV}(p_{\text{pre}}(\cdot),\, p_{\text{data}}(\cdot)), \end{aligned} \tag{4.2}$$

where the third inequality follows the triangle inequality $|p_{\text{pre}}(x)p_{\text{data}}(S) - p_{\text{data}}(x)p_{\text{pre}}(S)| \le p_{\text{pre}}(x)|p_{\text{pre}}(S) - p_{\text{data}}(S)| + p_{\text{pre}}(S)|p_{\text{pre}}(x) - p_{\text{data}}(x)|$. Combining (4.2) with the fact that $p_{\text{data}}(S) \ge \rho$ yields the bound (4.1). □

The next lemma bounds the total variation distance between the conditional pretrained distribution $p^S_{\text{pre}}(\cdot)$, and $\widetilde{p}^S_{\text{pre}}(\cdot)$ output by Algorithm A or B.

Lemma 4.3. *Let Assumption 4.1 (iv) hold. We have:*

$$d_{TV}(p^S_{\text{pre}}(\cdot),\, \widetilde{p}^S_{\text{pre}}(\cdot)) \le \eta\sqrt{\frac{T}{2}}. \tag{4.3}$$

Proof. Recall that $\mu_{\phi_*}(t,y)$ denotes the function approximation for $\nabla \log h(t,y)$ in Algorithm A or B (i.e., $\nabla \log h_{\phi_*}(t,y)$ in Algorithm A and $\frac{q_{\psi_*}(t,y)}{h_{\phi_*}(t,y)}$ in Algorithm B). Note that

$$\begin{aligned} d_{KL}(p^S_{\text{pre}}(\cdot),\, \widetilde{p}^S_{\text{pre}}(\cdot)) &\le d_{KL}(Y^S_T,\, \widetilde{Y}^S_T) \\ &= \mathbb{E}\int_0^T \left|\nabla \log h(t, Y^S_t) - \mu_{\phi_*}(t, Y^S_t)\right|^2 dt \le \eta^2 T, \end{aligned} \tag{4.4}$$

where the first inequality follows the data processing inequality, and the second identity is a consequence of Girsanov's theorem. Further by Pinsker's inequality, we get the bound (4.3). □

Combining the above two lemmas yields the following result on the total variation distance between $\widetilde{p}^S_{\text{data}}(\cdot)$ and $\widetilde{p}^S_{\text{pre}}(\cdot)$.

Theorem 4.4. *Let Assumption 4.1 hold. We have:*

$$d_{TV}(p^S_{\text{data}}(\cdot),\, \widetilde{p}^S_{\text{pre}}(\cdot)) \le \frac{3}{2\rho} d_{TV}(p(T,\cdot),\, p_{\text{noise}}(\cdot)) + \left(\frac{3\varepsilon}{2\rho} + \eta\right)\sqrt{\frac{T}{2}}. \tag{4.5}$$

In particular, assuming that $\mathbb{E}_{p_{\text{data}}(\cdot)}|X|^2 < \infty$, there are $C_{VE}, C_{VP} > 0$ (independent of T) such that

$$d_{TV}(p^S_{\text{data}}(\cdot),\, \widetilde{p}^S_{\text{pre}}(\cdot)) \le \begin{cases} \frac{C_{VE}}{\rho} T^{-1}\sqrt{\mathbb{E}_{p_{\text{data}}(\cdot)}|X|^2} + \left(\frac{3\varepsilon_{VE}}{2\rho} + \eta\right)\sqrt{\frac{T}{2}} & \textit{for VE}, \\ \frac{C_{VP}}{\rho} e^{-C_{VP}T}\sqrt{\mathbb{E}_{p_{\text{data}}(\cdot)}|X|^2} + \left(\frac{3\varepsilon_{VP}}{2\rho} + \eta\right)\sqrt{\frac{T}{2}} & \textit{for VP}. \end{cases} \tag{4.6}$$

Proof. It follows from [81, Theorem 5.2] that

$$d_{TV}(p_{\text{pre}}(\cdot),\, p_{\text{data}}(\cdot)) \le d_{TV}(p(T,\cdot),\, p_{\text{noise}}(\cdot)) + \varepsilon\sqrt{\frac{T}{2}}.$$

Combining this with Lemmas 4.2 and 4.3 yields the bound (4.5). The rest of the theorem follows from (4.5) and Proposition 2.2. □

4.2 Wasserstein bounds

Here we consider the Wasserstein-2 distance between $p^S_{\text{data}}(\cdot)$ and $\widetilde{p}^S_{\text{pre}}(\cdot)$, which is more involved than the total variation bounds. Note that we cannot bound the Wasserstein distance between $p^S_{\text{data}}(\cdot)$ and $p^S_{\text{pre}}(\cdot)$ in terms of that between $p_{\text{pre}}(\cdot)$ and $p_{\text{data}}(\cdot)$ as in Lemma 4.2. Our analysis relies on coupling and Malliavin calculus.

Let us introduce few more notations. Denote by $P^{\circ}_{[0,T]}(\cdot)$ the distribution of the (true) time reversal (2.2) of $\{X_t\}_{0 \le t \le T}$ and let $\mathring{h}(t,y) := \mathbb{P}^{\circ}_{[0,T]}(Y_T \in S \,|\, Y_t = y)$ be the associated h function. Let $P^{\circ,S}_{[0,T]}(\cdot)$ be the conditional distribution of $P^{\circ}_{[0,T]}(\cdot)$ on $\{Y_T \in S\}$, and $P^{\circ,S}_t(\cdot)$ be its marginal distribution at $Y_t \in S$. We need the following assumptions.

Assumption 4.5. *We make the following assumptions and take $t \in [0,T)$ below.*

(i) $W_2(p(T,\cdot), p_{\text{noise}}(\cdot)) < \infty$.

(ii) There is $\alpha > 0$ such that $(x-y)\cdot(f(t,x)-f(t,y)) \ge \alpha|x-y|^2$ for all t, x, y.

(iii) There is $\kappa_1 > 0$ such that $p(t,\cdot)$ is κ_1-strongly log-concave for all t.

(iv) There is $\varepsilon > 0$ sufficiently small: $|s_{\theta_*} - \nabla \log p|_\infty \le \varepsilon$.

(v) There is $\kappa_2 > 0$ such that $h(t,\cdot)$ is κ_2-strongly log-concave for all t.

(vi) There is $G : \mathbb{R}^d \to \mathbb{R}_+$ such that $|\nabla \log h(t,y)| \le \frac{G(y)}{T-t}$ for all t, y.

(vii) There is $\eta > 0$ such that $|\nabla \log h - \nabla \log h_{\phi_*}|_\infty \le \eta$ for Algorithm A, or $\left|\nabla \log h - \frac{q_{\psi_*}}{h_{\phi_*}}\right|_\infty < \infty$ for Algorithm B.

(viii) There is $F > 0$ such that $\mathbb{E}^{\circ}_{[0,T]}(\int_t^T |e^{\int_t^u \nabla \overline{f}(r,Y_r)dr}|_F^2 du \,|\, Y_t = y) \le F^2$ for all t, y.

(ix) There is $\gamma > 0$ such that $\mathbb{E}^{\circ}_{[0,T]}\left[\int_t^T |e^{\int_t^u \nabla \overline{f}(r,Y_r)dr} - e^{\int_t^u \nabla \overline{f}^{\circ}(r,Y_r)dr}|_F^2 du \,\middle|\, Y_t = y\right] \le \gamma^2$ for all t, y.

(x) There is $K > 0$ such that $\mathbb{E}^{\circ,S}_t\left[\mathring{h}(t,Y)^{-\frac{3}{2}}\right]$, $\mathbb{E}^{\circ,S}_t\left[G(Y)^2\mathring{h}(t,Y)^{-\frac{3}{2}}\right] \le K$ for all t, where $G(\cdot)$ is defined in (vi).

Before stating our result, we make several comments on Assumption 4.5.

The conditions (*i*)–(*iv*) can be used to bound $W_2(p_{\text{data}}(\cdot), p_{\text{pre}}(\cdot))$, which together with (*vii*) yields an estimate of $W_2(p^S_{\text{data}}(\cdot), p^S_{\text{pre}}(\cdot))$ involving a perturbation bound on $\nabla \log h$. The conditions (*v*)–(*x*) are required for the perturbation analysis of $\nabla \log h$ via Malliavin calculus.

Note that the condition (*iii*) holds for the VE and VP models, if $p_{\text{data}}(\cdot)$ is strongly log-concave. The condition (*iv*) is stronger than Assumption 4.1 (*ii*) for the same reason as explained in the footnote [7]. (In fact, it suffices to assume an L^2 bound under the guided distribution in Algorithm A or B.) Finally, the condition (*vi*) is reasonable, since it holds for heat(-like) kernels. Finally, conditions (viii) and (ix) are satisfied when $\overline{f}$ and $\overline{f}^{\circ}$ are differentiable in x and uniformly Lipschitz.

Remark 4.6 (Comparison to Assumption 4.1 for the TV bound.). *Although the Wasserstein distance is a weaker notion of discrepancy than total variation, it has shown to align with human judgment on image similarity [10]. Technically, the derivation of Wasserstein bounds typically relies on pathwise stability estimates for the underlying SDE, which require stronger regularity and growth assumptions on the drift. This explains why Assumption 4.5 is more restrictive than Assumption 4.1. In contrast, the total variation bound is obtained via change-of-measure and martingale arguments, which do not rely on Lipschitz continuity or contractivity of the dynamics. Thus, the two results are complementary: total variation bounds provide stronger distributional* *guarantees under weaker structural assumptions, whereas Wasserstein bounds yield weaker—but geometrically interpretable—guarantees under stronger regularity conditions.*

The following theorem provides a bound on $W_2(p^S_{\mathrm{data}}(\cdot),\, \widetilde{p}^S_{\mathrm{pre}}(\cdot))$.

Theorem 4.7. *Let Assumption 4.5 hold, and set* $\Lambda := \alpha + (\kappa_1+\kappa_2)g^2_{\max}$. *We have:*

$$W_2(p^S_{\mathrm{data}}(\cdot),\, \widetilde{p}^S_{\mathrm{pre}}(\cdot)) \le e^{-\Lambda T} W_2(p(T,\cdot),\, p_{\mathrm{noise}}(\cdot)) + C(\varepsilon+\eta+\gamma), \tag{4.7}$$

for some $C>0$ *(independent of* T*). In particular, assuming that* $\mathbb{E}_{p_{\mathrm{data}}(\cdot)}|X|^2 < \infty$, $p_{\mathrm{data}}(\cdot)$ *is* κ*-strongly log-concave for* κ *sufficiently large and Assumption 4.5 (iv)–(x) holds, there are* $C_{\mathtt{VE}}, C_{\mathtt{VP}} > 0$ *(independent of* T*) such that*

$$W_2(p^S_{\mathrm{data}}(\cdot),\, \widetilde{p}^S_{\mathrm{pre}}(\cdot)) \le \begin{cases} C_{\mathtt{VE}}\left(e^{-C_{\mathtt{VE}}T}\sqrt{\mathbb{E}_{p_{\mathrm{data}}(\cdot)}|X|^2} + \varepsilon+\eta+\gamma\right) & \textit{for VE,} \\ C_{\mathtt{VP}}\left(e^{-C_{\mathtt{VP}}T}\sqrt{\mathbb{E}_{p_{\mathrm{data}}(\cdot)}|X|^2} + \varepsilon+\eta+\gamma\right) & \textit{for VP.} \end{cases} \tag{4.8}$$

Proof. The proof is split into three steps.

Step 1. We start by establishing a coupling bound on $W_2(p^S_{\mathrm{data}}(\cdot),\, \widetilde{p}^S_{\mathrm{pre}}(\cdot))$. Recall that $\mu_{\phi_*}(t,y)$ denotes the function approximation for $\nabla \log h(t,y)$ in Algorithm A or B (i.e., $\nabla \log h_{\phi_*}(t,y)$ in Algorithm A and $\frac{q_{\psi_*}(t,y)}{h_{\phi_*}(t,y)}$ in Algorithm B).

Consider the coupled equations:

$$\begin{cases} dU_t = \left(\overline{f}^\circ(t,U_t) + \overline{g}(t)^2 \nabla \log \mathring{h}(t,U_t)\right)dt + \overline{g}(t)dB_t, \\ dV_t = \left(\overline{f}(t,V_t) + \overline{g}(t)^2 \mu_{\phi^*}(t,V_t)\right)dt + \overline{g}(t)dB_t, \end{cases}$$

where (U_0,V_0) are coupled to achieve $W_2(p(T,\cdot),\, p_{\mathrm{noise}}(\cdot))$. Note that $W_2^2(p^S_{\mathrm{data}}(\cdot),\, \widetilde{p}^S_{\mathrm{pre}}(\cdot)) \le \mathbb{E}|U_T - V_T|^2$, so our goal is to bound $\mathbb{E}|U_T-V_T|^2$. By Itô's formula, we get:

$$\begin{aligned} d|U_t - V_t|^2 = 2(U_t - V_t)\cdot\Big(& -f(T-t,U_t) + \overline{g}(t)^2\nabla\log p(T-t,U_t) + \overline{g}(t)^2\nabla\log\mathring{h}(t,U_t) \\ & + f(T-t,V_t) - \overline{g}(t)^2 s_{\theta_*}(T-t,V_t) - \overline{g}(t)^2\mu_{\phi_*}(t,V_t)\Big)dt, \end{aligned}$$

which implies that

$$\begin{aligned} \frac{1}{2}\frac{d\mathbb{E}|U_t-V_t|^2}{dt} = & -\underbrace{\mathbb{E}[(U_t-V_t)\cdot(f(T-t,U_t)-f(T-t,V_t))]}_{(a)} \\ & + \overline{g}(t)^2\underbrace{\mathbb{E}[(U_t-V_t)\cdot(\nabla\log p(T-t,U_t) - s_{\theta_*}(T-t,V_t))]}_{(b)} \\ & + \overline{g}(t)^2\underbrace{\mathbb{E}[(U_t-V_t)\cdot(\nabla\log\mathring{h}(t,U_t) - \mu_{\phi_*}(t,V_t))]}_{(c)}. \end{aligned} \tag{4.9}$$

By Assumption 4.5 (*ii*), the term $(a) \ge \alpha\mathbb{E}|U_t-V_t|^2$. For the term (b), we get:

$$\begin{aligned} (b) = & \ \mathbb{E}[(U_t-V_t)\cdot(\nabla\log p(T-t,U_t) - \nabla\log p(T-t,V_t))] \\ & + \mathbb{E}[(U_t-V_t)\cdot(\nabla\log p(T-t,V_t) - s_{\theta_*}(T-t,V_t))] \\ \le & -\kappa_1\mathbb{E}|U_t-V_t|^2 + \varepsilon\sqrt{\mathbb{E}|U_t-V_t|^2}, \end{aligned} \tag{4.10}$$

which follows from Assumption 4.5 (*iii*) and (*iv*). Similarly, we have:

$$\begin{aligned} (c) = & \ \mathbb{E}[(U_t-V_t)\cdot(\nabla\log\mathring{h}(t,U_t) - \nabla\log h(t,U_t))] \\ & + \mathbb{E}[(U_t-V_t)\cdot(\nabla\log h(t,U_t) - \nabla\log h(t,V_t))] \\ & + \mathbb{E}[(U_t-V_t)\cdot(\nabla\log h(t,V_t) - \mu_{\phi_*}(t,V_t))] \\ \le & -\kappa_2\mathbb{E}|U_t-V_t|^2 + \left(\eta + \sqrt{\mathbb{E}|\nabla\log\mathring{h}(t,U_t) - \nabla\log h(t,U_t))|^2}\right)\sqrt{\mathbb{E}|U_t-V_t|^2} \end{aligned} \tag{4.11}$$

which follows from Assumption 4.5 (v) and (vii). Combining (4.9), (4.10) and (4.11) yields:

$$\begin{aligned}\frac{d\mathbb{E}|U_t - V_t|^2}{dt} &\le -2\Lambda\, \mathbb{E}|U_t - V_t|^2 \\ &\quad + 2g_{\max}^2 \left(\sqrt{\mathbb{E}|\nabla \log \mathring{h}(t, U_t) - \nabla \log h(t, U_t))|^2} + \varepsilon + \eta \right) \sqrt{\mathbb{E}|U_t - V_t|^2}.\end{aligned} \tag{4.12}$$

Step 2. Throughout this step, we fix $t \in [0, T)$. Now we apply Malliavin calculus to bound $|\nabla \log h(t, y) - \nabla \log \mathring{h}(t, y)|$. First we consider $|h(t, y) - \mathring{h}(t, y)|$. It follows from [28, Proposition 3.1] that for $\varepsilon > 0$ sufficiently small,

$$\begin{aligned}&|h(t, y) - \mathring{h}(t, y)| \\ &\le C \left| \mathbb{E}^{\circ}_{[0,T]} \left[1(Y_T \in S) \int_t^T \overline{g}(u)(\nabla \log p(T-u, Y_u) - s_{\theta^*}(T-u, Y_u)) dB_u \,\middle|\, Y_t = y \right] \right| \\ &\le C g_{\max} \sqrt{\mathring{h}(t, y)} \sqrt{\mathbb{E}^{\circ}_{[0,T]} \left[\int_t^T \left| \nabla \log p(T-u, Y_u) - s_{\theta^*}(T-u, Y_u)) \right|^2 du \,\middle|\, Y_t = y \right]} \\ &\le C g_{\max} \sqrt{T - t} \sqrt{\mathring{h}(t, y)}\, \varepsilon,\end{aligned} \tag{4.13}$$

where the second inequality is by the Cauchy–Schwarz inequality, and the last inequality is due to Assumption 4.5 (iv).

Next we bound $|\nabla h(t, y) - \nabla \mathring{h}(t, y)|$. Introduce the *first variation process* $\{Z_u\}_{t \le u \le T}$ which solves:

$$dZ_u = \nabla \overline{f}(u, Y_u) Z_u du, Z_t = I.$$

So $Z_u = \exp\left(\int_t^u \nabla \overline{f}(r, Y_r) dr\right)$ (here $\nabla \overline{f}$ is a matrix.) By [28, Proposition 3.2],

$$\nabla h(t, y) = \mathbb{E}_{[0,T]} \left(\frac{1(Y_T \in S)}{T - t} \int_t^T \frac{Z_u}{\overline{g}(u)} dB_u \right) = \mathbb{E}_{[0,T]} \left(\frac{1(Y_T \in S)}{T - t} \int_t^T \frac{e^{\int_t^u \nabla \overline{f}(r, Y_r) dr}}{\overline{g}(u)} dB_u \right).$$

A similar argument as before shows that

$$\begin{aligned}&|\nabla h(t, y) - \nabla \mathring{h}(t, y)| \le (d) + (e), \text{ where} \\ &(d) = C \Bigg| \mathbb{E}^{\circ}_{[0,T]} \Bigg(\frac{1(Y_T \in S)}{T - t} \int_t^T \frac{e^{\int_t^u \nabla \overline{f}(r, Y_r) dr}}{\overline{g}(u)} dB_u \\ &\qquad\qquad \int_t^T \overline{g}(u)(\nabla \log p(T-u, Y_u) - s_\theta(T-u, Y_u)) dB_u \,\Bigg|\, Y_t = y \Bigg) \Bigg|, \\ &(e) = \left| \mathbb{E}^{\circ}_{[0,T]} \left(\frac{1(Y_T \in S)}{T - t} \int_t^T \frac{e^{\int_t^u \nabla \overline{f}(r, Y_r) dr} - e^{\int_t^u \nabla \overline{f}^{\circ}(r, Y_r) dr}}{\overline{g}(u)} dB_u \,\middle|\, Y_t = y \right) \right|.\end{aligned} \tag{4.14}$$

For the term (d), we have:

$$\begin{aligned}(d) &\le \frac{C}{(T - t) g_{\min}} \mathring{h}(t, y)^{\frac{1}{4}} \left\{ \mathbb{E}^{\circ}_{[0,T]} \left(\int_t^T |e^{\int_t^u \nabla \overline{f}(r, Y_r) dr}|_F^2 du \,\middle|\, Y_t = y \right) \right\}^{\frac{1}{2}} \\ &\qquad \left\{ \mathbb{E}^{\circ}_{[0,T]} \left(\int_t^T \overline{g}(u)(\nabla \log p(T-u, Y_u) - s_\theta(T-u, Y_u)) dB_u \,\middle|\, Y_t = y \right)^4 \right\}^{\frac{1}{4}} \\ &\le \frac{C}{(T - t) g_{\min}} \varepsilon \mathring{h}(t, y)^{\frac{1}{4}} \left\{ \mathbb{E}^{\circ}_{[0,T]} \left(\int_t^T |e^{\int_t^u \nabla \overline{f}(r, Y_r) dr}|_F^2 du \,\middle|\, Y_t = y \right) \right\}^{\frac{1}{2}} \left(g_{\max} \varepsilon \sqrt{T - t} \right) \\ &\le \frac{C g_{\max} F}{g_{\min}} \varepsilon \mathring{h}(t, y)^{\frac{1}{4}},\end{aligned} \tag{4.15}$$

where the first inequality is due to Hölder's inequality, the second inequality follows from the moment inequality (see [70, Chapter IV, §4]) and Assumption 4.5 (iv), and the final inequality is by Assumption 4.5 $(viii)$. For the term (e), we have:

$$
\begin{aligned}
(e) &\le \frac{1}{(T-t)g_{\min}}\sqrt{\mathring{h}(t,y)}\sqrt{\mathbb{E}^{\circ}_{[0,T]}\left(\int_t^T |e^{\int_t^u \nabla\overline{f}(r,Y_r)dr} - e^{\int_t^u \nabla\overline{f}^{\circ}(r,Y_r)dr}|_F^2 du \,\middle|\, Y_t = y\right)} \\
&\le \frac{1}{g_{\min}}\frac{\gamma}{\sqrt{T-t}}\sqrt{\mathring{h}(t,y)},
\end{aligned} \tag{4.16}
$$

where the first inequality is by the Cauchy–Schwarz inequality, and the second inequality follows from Assumption 4.5 (ix). Applyinng (4.15) and (4.16) into (4.14) and combing with the fact that $\mathring{h}(t,y) \le 1$ yields:

$$
|\nabla h(t,y) - \nabla \mathring{h}(t,y)| \le C\left(F\varepsilon + \frac{\gamma}{\sqrt{T-t}}\right)\mathring{h}(t,y)^{\frac{1}{4}}, \text{for some } C > 0. \tag{4.17}
$$

Combining (4.13), (4.17) and Assumption 4.5 (vi) leads to:

$$
\begin{aligned}
|\nabla \log h(t,y) - \nabla \log \mathring{h}(t,y)| &\le \left|\frac{\mathring{h}(t,y) - h(t,y)}{\mathring{h}(t,y)}\nabla \log h(t,y)\right| + \left|\frac{\nabla h(t,y) - \nabla\mathring{h}(t,y)}{\mathring{h}(t,y)}\right| \\
&\le C\left\{\left(F + \frac{G(y)}{\sqrt{T-t}}\right)\varepsilon + \frac{\gamma}{\sqrt{T-t}}\right\}\mathring{h}(t,y)^{-\frac{3}{4}}.
\end{aligned} \tag{4.18}
$$

Step 3. Observe that $\{U_t\}_{0\le t\le T}$ is distributed by $P^{\circ,S}_{[0,T]}(\cdot)$. By (4.12), (4.18) and Assumption 4.5 (x), we have:

$$
\frac{d\mathbb{E}|U_t - V_t|^2}{dt} \le -\Lambda\,\mathbb{E}|U_t - V_t|^2 + C\left(\varepsilon + \eta + \frac{\varepsilon+\gamma}{\sqrt{T-t}}\right)\sqrt{\mathbb{E}|U_t - V_t|^2}. \tag{4.19}
$$

By Grönwall's inequality (see [22, Theorem 21]), we get:

$$
\begin{aligned}
\mathbb{E}|U_T - V_T|^2 &\le \left(e^{-\Lambda T}W_2(p(T,\cdot),\, p_{\text{noise}}(\cdot)) + C\int_0^T\left(\varepsilon + \eta + \frac{\varepsilon+\gamma}{\sqrt{T-t}}\right)e^{-\Lambda(T-t)}dt\right)^2 \\
&\le \left(e^{-\frac{1}{2}\Lambda T}W_2(p(T,\cdot),\, p_{\text{noise}}(\cdot)) + C(\varepsilon+\eta+\gamma)\right)^2,
\end{aligned}
$$

which leads to the bound (4.7).

The rest of the theorem follows from the fact that $W_2^2(p(T,\cdot),\, p_{\text{noise}}(\cdot)) \le \mathbb{E}_{p_{\text{data}}(\cdot)}|X|^2$ for VE, and $W_2(p(T,\cdot),\, p_{\text{noise}}(\cdot)) \le e^{-CT}\mathbb{E}_{p_{\text{data}}(\cdot)}|X|^2$ with $C > 0$ for VP. □

4.3 Learning the h function

In this section, we study the convergence of the stochastic optimization problems outlined in Section 3.2 (Alogrithm A and B) to learn the function $\nabla \log h$. Our approach is generic, and does not require any explicit structure of function approximations.

4.3.1 Learning h

We first consider the convergence of the stochastic optimization problem (3.15). The stochastic approximation to the martingale loss is given by

$$
\phi_{n+1} = \phi_n + \delta_n \mathcal{V}(\phi_n, \tau^{(n)}, Y^{(n)}), \tag{4.20}
$$

where $\delta_n > 0$ is the step size, $\tau^{(n)} \sim \mathrm{Unif}\,[0,T]$, $Y^{(n)} = \{Y_t^{(n)}\}_{0 \le t \le T}$ is a copy of the pretrained model $P_{[0,T]}(\cdot)$, and

$$\mathcal{V}(\phi, \tau, Y) := -2\partial_\phi h_\phi(\tau, Y_\tau)(h_\phi(\tau, Y_\tau) - 1(Y_T \in S)). \tag{4.21}$$

Our goal is to provide a quantitative bound on $|h_{\phi_n}(t,y) - h(t,y)|$ (in some weak sense). Our idea follows from [84, Section 4], which relies on [7] for stochastic approximations. Set

$$V(\phi) := \mathbb{E}_{\tau \sim \mathrm{Unif}\,[0,T]}\Big[\mathbb{E}_{[0,T]}[\mathcal{V}(\phi, \tau, Y)]\Big]. \tag{4.22}$$

We need the following assumptions.

Assumption 4.8.

(i) The ODE $\phi'(u) = V(\phi(u))$ has a unique stable equilibrium ϕ_* [8]*.*

(ii) There is $C > 0$ such that $\mathbb{E}_{[0,T]}[\mathcal{V}(\phi_{n+1}, Y) \,|\, \phi_n] \le C(1 + \phi_n^2)$.

(iii) There is $\ell > 0$ such that $(\phi - \phi_*)V(\phi) \le -\ell|\phi - \phi_*|^2$.

(iv) There is a function $\omega : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\omega(r)/r^\nu$ is bounded for some $\nu \le 2$, and $|h_\phi - h_{\phi'}|_\infty \le \omega(|\phi - \phi'|)$ for all ϕ, ϕ'.

The assumptions (i)–(iii) guarantees that the stochastic approximation (4.20) converges, and the assumption (iv) quantifies the sensitivity of the function approximation $\{h_\phi(t,x)\}_\phi$ with respect to the parameter.

Theorem 4.9. *Let Assumption 4.8 hold, and $\delta_n = \frac{A}{n^\zeta + B}$ for some $\zeta \le 1$, $A > \frac{\zeta}{2\ell}$ and $B > 0$. We have:*

$$\mathbb{E}_{[0,T]}|h_{\phi_n} - h|_\infty \le |h - h_{\phi_*}|_\infty + Cn^{-\frac{\zeta\nu}{2}}, \tag{4.23}$$

where h is defined in (3.3)*.*

Note that the upper bound in (4.23) consists of two terms. The first term, $\|h - h_{\phi_*}\|_\infty$, represents the approximation error induced by the functional class $\{h_\phi\}_\phi$. If the class is sufficiently expressive to contain h, this term vanishes. The second term converges to zero as $n \to \infty$. In particular, choosing $\zeta = 1$ and $\nu = 2$ yields a linear convergence rate of order n^{-1} for the second term.

Proof. It follows from [7, Theorem 22] that under Assumption 4.8 (i)–(iii) and $\delta_n = \frac{A}{n^\zeta + B}$,

$$\mathbb{E}_{[0,T]}|\phi_n - \phi_*|^2 \le Cn^{-\zeta}. \tag{4.24}$$

As a result,

$$\begin{aligned}\mathbb{E}_{[0,T]}|h_{\phi_n} - h|_\infty &\le |h - h_{\phi_*}|_\infty + \mathbb{E}_{[0,T]}|h_{\phi_n} - h_{\phi_*}|_\infty \\ &\le |h - h_{\phi_*}|_\infty + \mathbb{E}_{[0,T]}[\omega(|\phi_n - \phi_*|)] \\ &\le |h - h_{\phi_*}|_\infty + C(\mathbb{E}|\phi_n - \phi_*|^2)^{\frac{\nu}{2}},\end{aligned} \tag{4.25}$$

where the first inequality is from the triangle inequality, the second inequality is due to Assumption 4.8 (iv), and the last inequality is by the Cauchy-Schwarz inequality. Combining (4.24) and (4.25) yields the bound (4.23). □

The first term $|h - h_{\phi_*}|_\infty$ on the right side of (4.23) quantifies how well the family $\{h_\phi(t,y)\}_\phi$ approximates the h function, and the second term $n^{-\frac{\zeta\nu}{2}}$ gives the convergence rate of the stochastic approximation (4.20). In particular, if the family $\{h_\phi(t,y)\}_\phi$ is rich enough to contain the h function (i.e., $|h - h_{\phi_*}|_\infty = 0$), and $\{h_\phi(t,y)\}_\phi$ is Lipschitz in ϕ (i.e, $\nu = 1$), then h_{ϕ_n} converges to h at a rate $n^{-\frac{1}{2}}$ by taking the step size $1/n$.

[8] ϕ_* is the unique stable equilibrium means that $V(\phi) = 0$ has a unique root ϕ_*, and $V'(\phi_*) < 0$.

4.3.2 Learning ∇h

Now we establish similar results for the stochastic optimization problem (3.18). Fixing $n > 0$, we use h_{ϕ_n} to approximate the covariation, so the stochastic approximation to the covariation loss is:

$$\psi_{m+1} = \psi_m + \delta'_m \mathcal{U}_n(\psi_m, \tau^{(m)}, Y^{(m)}), \tag{4.26}$$

where

$$\mathcal{U}_n(\psi, \tau, Y) := -2\partial_\psi q_\phi(\tau, Y_\tau)\left(q_\psi(\tau, Y_\tau) - \frac{1}{\overline{g}(\tau)^2}\frac{d[h_{\phi_n}, Y]_t}{dt}|_{t=\tau}\right). \tag{4.27}$$

Also set

$$U_n(\psi) := \mathbb{E}_{\tau\sim\text{Unif}\,[0,T]}\{\mathbb{E}_{[0,T]}[\mathcal{U}_n(\psi, \tau, Y)]\}. \tag{4.28}$$

We need the following assumptions.

Assumption 4.10.

(i) The ODE $\psi'(u) = U_n(\psi(u))$ has a unique stable equilibrium ψ_*.

(ii) There is $C > 0$ such that $\mathbb{E}_{[0,T]}[\mathcal{U}_n(\psi_{m+1}, Y) \mid \psi_m] \le C(1 + \psi_m^2)$.

(iii) There is $\ell > 0$ such that $(\psi - \psi_*)U_n(\psi) \le -\ell|\psi - \psi_*|^2$.

(iv) There is a function $\omega : \mathbb{R}_+ \to \mathbb{R}_+$ such that $\omega(r)/r^{\nu'}$ is bounded for some $\nu' \le 2$, and $|q_\psi - q_{\psi'}|_\infty \le \omega(|\psi - \psi'|)$ for all ψ, ψ'.

Theorem 4.11. *Let Assumption 4.10 hold, and $\delta'_m = \frac{A}{m^{\zeta'}+B}$ for some $\zeta' \le 1$, $A > \frac{\zeta'}{2\ell}$ and $B > 0$. We have:*

$$\mathbb{E}_{[0,T]}|q_{\psi_m} - \nabla h|_\infty \le \mathbb{E}_{[0,T]}\left|\frac{1}{\overline{g}(t)^2}\frac{d[h_{\phi_n}, Y]_t}{dt} - \nabla h(t, Y_t)\right| + \mathbb{E}_{[0,T]}|\nabla h_{\phi_n} - q_{\psi_{*n}}|_\infty + Cm^{-\frac{\zeta'\nu'}{2}}. \tag{4.29}$$

The proof of the theorem is in the same vein as that of Theorem 4.9. The first term $\mathbb{E}_{[0,T]}\left(\left|\frac{1}{\overline{g}(t)^2}\frac{d[h_{\phi_n},Y]_t}{dt} - \nabla h(t, Y_t)\right|\right)$ on the right side of (4.29) quantifies how close the covariation $\frac{d[h_{\phi_n},Y]_t}{dt}$ is to $\frac{d[h,Y]_t}{dt}$. However, it is generally hard to provide an explicit bound on this term [9], and we simply denote it by $\theta(n)$. Also note that the estimation of $\frac{d[h_{\phi_n},Y]_t}{dt}$ also incurs a sample error [10], which we do not pursue here. The second term $|\nabla h_{\phi_n} - q_{\psi_{*n}}|_\infty$ measures how well the family $\{q_\psi(t, y)\}_\psi$ approximates ∇h_{ϕ_n}, and the third term $m^{-\frac{\zeta'\nu'}{2}}$ is the convergence rate of the stochastic approximation (4.26).

Combining Theorem 4.9 and 4.11, we have (at least heuristically) that the learning error η of $\nabla \log h$ is of order:

$$\theta(n) + n^{-\frac{\zeta\nu}{2}} + m^{-\frac{\zeta'\nu'}{2}} + \text{discrepancy of approximations } \{h_\phi(t, y)\}_\phi, \{q_\psi(t, y)\}_\psi. \tag{4.30}$$

Again if the families $\{h_\phi(t, y)\}_\phi$, $\{q_\psi(t, y)\}_\psi$ are rich enough and Lipschitz in the parameter (i.e., $\nu = \nu' = 1$), then η is of order $\theta(n) + n^{-\frac{1}{2}} + m^{-\frac{1}{2}}$ by taking the step sizes $\delta_n = 1/n$, $\delta'_m = 1/m$.

[9] It follows from [47, Chapter VI, §6c] (and also [45]) that if a diffusion process $\{Z^n_t\}_{0\le t\le T}$ converges in distribution to $\{Z_t\}_{0\le t\le T}$, and under very technical conditions, the quadratic variation of Z^n converges in distribution to that of Z. So $\frac{d[h_{\phi_n},Y]_t}{dt}$ and $\frac{d[h,Y]_t}{dt}$ are expected to be close, because $h_{\phi_n} \approx h$ by Theorem 4.9. However, the proofs in [45, 47] rely on soft measure-theoretic arguments, and it seems to be a challenging task to provide an explicit convergence rate of quadratic variation.

[10] As mentioned in the footnote [6], the quadratic variation can be estimated by sampling the pretrained model repeatedly. So the central limit theorem implies that the sample error of estimating the covariation $\frac{d[h_{\phi_n},Y]_t}{dt}$ is of order $1/M$, with M the sample size.

5 Extensions

This section provides two extensions of the algorithms proposed in Section 3. In Section 5.1, we consider a more efficient probability-flow ODE sampler. In Section 5.2, we discuss how conditioning can be reinforced in the context of classifier guidance.

5.1 ODE sampling

Diffusion models admit an equivalent probability-flow ODE whose deterministic trajectories preserve the target marginals, and ODE-based samplers are often more efficient than stochastic SDE samplers due to reduced variance accumulation, improved numerical stability, and the ability to leverage higher-order adaptive solvers. In particular, our Diffusion Conditional Guidance framework naturally extends to ODE-based sampling with minimal modification.

The ODE sampler for the pretrained model (2.4) follows:

$$\frac{dY_t}{dt} = -f(t, Y_t) + \frac{1}{2} g(T-t)^2 s_{\theta_*}(T-t, Y_t), Y_0 \sim p_{\text{noise}}(\cdot). \tag{5.1}$$

The following proposition is key to the ODE sampling of $p^S_{\text{data}}(\cdot)$.

Proposition 5.1. *Let $\{X_t\}_{0 \le t \le T}$ be defined by the SDE*

$$dX_t = f(t, X_t)dt + g(t)dW_t, X_0 \sim p^S_{\text{data}}(\cdot),$$

and let $\{X'_t\}_{0 \le t \le T}$ be defined by the ODE

$$\frac{dX'_t}{dt} = f(t, X'_t) - \frac{1}{2}\left(\nabla \log p(t, X'_t) + \nabla \log \mathring{h}(t, X'_t)\right), X_0 \sim p^S_{\text{data}}(\cdot). \tag{5.2}$$

Then X_t and X'_t have the same distribution for each t.

Proof. Let $p(t, x; S) := \frac{\mathbb{P}(X_t \in dx, X_0 \in S)/dx}{p_{\text{data}}(S)}$, and note that $\nabla \log p(t, x; S) = \nabla \log p(t, x) + \nabla \log \mathring{h}(t, x)$. Thus, $\frac{dX'_t}{dt} = f(t, X'_t) - \frac{1}{2} \nabla \log p(t, X'_t; S)$. The conclusion follows from [81, Theorem 6.1]. □

It is expected that $\nabla \log \mathring{h}(t, y) \approx \nabla \log h(t, y)$. That is, the h functions are close under $P^\circ_{[0,T]}(\cdot)$ and $P_{[0,T]}(\cdot)$. Denote by $\mu_{\phi_*}(t, y)$ the function approximation for $\nabla \log h(t, y)$ in Algorithm A or B (i.e., $\nabla \log h_{\phi_*}(t, y)$ in Algorithm A and $\frac{q_{\psi_*}(t,y)}{h_{\phi_*}(t,y)}$ in Algorithm B). The ODE sampler of $p^S_{\text{data}}(\cdot)$ follows

$$\frac{dY^S_t}{dt} = -f(t, Y_t) + \frac{1}{2} g(T-t)^2 \left(s_{\theta_*}(T-t, Y^S_t) + \mu_{\phi_*}(t, Y^S_t)\right), Y_0 \sim p_{\text{noise}}(\cdot). \tag{5.3}$$

Algorithm A and B can be easily adapted to labeling and ODE sampling, which are summarized as follows.

Note that Algorithm B' also requires the SDE sampler (2.4), which is used to estimate the covariation $\frac{d[h_{\phi_*}, Y]_t}{dt}$ in Step 2. This is because the ODE and the SDE sampler only agree in distribution marginally, but not at the level of the process that is needed to approximate the quadratic variation.

We also point out various numerical schemes to discretize the above continuous-time samplers, see [81, Section 5.3] and [94] for the references. Since we rely on the pretrained model for sampling, we will simply follow its built-in schemes (so we do not pursue this direction here).

```
Algorithm A' Conditional diffusion guidance via martingale loss (CDG-ML)
  Input: pretrained model $\{Y_t\}_{0\le t\le T}$ (5.1), label $r(y)$, guidance set $S_r$, parametrized family $\{h_\phi(t,y)\}_\phi$
  Step 1. Solve the stochastic optimization problem (3.15) that outputs $\phi_*$.
  Step 2. Sample
      $$\frac{dY^S_t}{dt} = -f(t,Y_t) + \frac{1}{2}\overline{g}(t)^2 s_{\theta_*}(T-t,Y^S_t) + \frac{1}{2}\overline{g}(t)^2\nabla\log h_{\phi_*}(t,Y^S_t), Y^S_0 \sim p_{\text{noise}}(\cdot).$$
  Output: $Y^S_T$.
```

```
Algorithm B' Conditional diffusion guidance via martingale-covariation loss (CDG-MCL)
  Input: pretrained model $\{Y_t\}_{0\le t\le T}$ (2.4) and (5.1), label $r(y)$, guidance set $S_r$, parametrized families $\{h_\phi(t,y)\}_\phi$, $\{q_\psi(t,y)\}_\psi$
  Step 1. Solve the stochastic optimization problem (3.15) that outputs $\phi_*$.
  Step 2. Use the ODE (5.1) to sample $Y_t$, and then the SDE (2.4) to estimate $\frac{d[h_{\phi_*},Y]_t}{dt}$.
      Solve the stochastic optimization problem (3.18) that outputs $\psi_*$.
  Step 3. Sample
      $$\frac{dY^S_t}{dt} = -f(t,Y_t) + \frac{1}{2}\overline{g}(t)^2 s_{\theta_*}(T-t,Y^S_t) + \frac{1}{2}\overline{g}(t)^2\frac{q_{\psi_*}(t,Y^S_t)}{h_{\phi_*}(t,Y^S_t)}, Y^S_0 \sim p_{\text{noise}}(\cdot).$$
  Output: $Y^S_T$.
```

5.2 Reinforce conditioning

Central to the conditional sampling (3.6) is the Bayes rule $P_{[0,T]}(Y|Y_T \in S) \propto P_{[0,T]}(Y_T \in S|Y)P_{[0,T]}(Y)$, which guides the sample Y to a mode specified by the classifier $P_{[0,T]}(Y_T \in S|Y)$. Classifier guidance [20] further strengthens conditioning with a guidance scale $\eta \ge 0$:

$$P_{[0,T]}(Y|Y_T \in S) \propto P_{[0,T]}(Y_T \in S|Y)^{1+\eta}P_{[0,T]}(Y),$$

which echos with [65] that empirical data sometimes infer physical laws differing from those assumed in traditional modeling. See [41] for detailed discussions.

The resulting SDE sampler is:

$$\begin{aligned} dY^{S,\eta}_t &= \left(\overline{f}(t,Y^{S,\eta}_t) + (1+\eta)\overline{g}^2(t)\nabla\log h(t,Y^{S,\eta}_t)\right)dt + \overline{g}(t)dB_t \\ &= (\underbrace{\overline{f}(t,Y^{S,\eta}_t) + \overline{g}^2(t)\nabla\log h(t,Y^{S,\eta}_t)}_{\text{conditional generation}} + \underbrace{\eta\overline{g}^2(t)\nabla\log h(t,Y^{S,\eta}_t)}_{\text{strengthened conditioning}})dt + \overline{g}(t)dB_t. \end{aligned} \tag{5.4}$$

So the sampling step in CDG-ML (Algorithm A and A') reads as:

$$dY^{S,\eta}_t = \left(\overline{f}(t,Y^{S,\eta}_t) + (1+\eta)\overline{g}(t)^2\frac{\nabla h_{\phi_*}(t,Y^{S,\eta}_t)}{h_{\phi_*}(t,Y^{S,\eta}_t)}\right)dt + \overline{g}(t)dB_t, \quad Y^{S,\eta}_0 \sim p_{\text{noise}}(\cdot), \tag{5.5}$$

or

$$\frac{dY^{S,\eta}_t}{dt} = -f(t,Y^{S,\eta}_t) + \frac{1}{2}\overline{g}(t)^2 s_{\theta_*}(T-t,Y^{S,\eta}_t) + \frac{1+\eta}{2}\overline{g}(t)^2\nabla\log h_{\phi_*}(t,Y^{S,\eta}_t), \quad Y^{S,\eta}_0 \sim p_{\text{noise}}(\cdot), \tag{5.6}$$

and that in CDG-MCL (Algorithm B and B') reads as:

$$dY^{S,\eta}_t = \left(\overline{f}(t,Y^{S,\eta}_t) + (1+\eta)\overline{g}(t)^2\frac{q_{\psi_*}(t,Y^{S,\eta}_t)}{h_{\phi_*}(t,Y^{S,\eta}_t)}\right)dt + \overline{g}(t)dB_t, \quad Y^{S,\eta}_0 \sim p_{\text{noise}}(\cdot), \tag{5.7}$$

or

$$\frac{dY_t^{S,\eta}}{dt} = -f(t, Y_t^{S,\eta}) + \frac{1}{2}\bar{g}(t)^2 s_{\theta_*}(T-t, Y_t^{S,\eta}) + \frac{1+\eta}{2}\bar{g}(t)^2 \frac{q_{\psi_*}(t, Y_t^{S,\eta})}{h_{\phi_*}(t, Y_t^{S,\eta})}, \quad Y_0^{S,\eta} \sim p_{\text{noise}}(\cdot). \tag{5.8}$$

A crucial problem in classifier guidance is the choice of the scale η. It was shown in [93] that as w increases, the generation is steered toward the high-probability modes induced by the classifier, while the sample diversity (in entropy) decreases. On the other hand, overly large η will lead to mode collapse.

6 Numerical experiments

In this section, we present numerical experiments to illustrate and validate the proposed Conditional Diffusion Guidance framework. We test our framework on a synthetic example to visualize the power of enforcing a sharp constraint. We then apply our framework to perform stress testing in finance and supply chain systems. Across all three testcases, ur experiments are designed to assess (i) the effectiveness of the martingale-based objectives for learning the guidance function h and its gradient, (ii) the quality of the resulting conditional samples compared with the target conditional distribution, and (iii) the empirical behavior of the method under finite-sample and model-misspecification effects. In particular, we focus on demonstrating how the proposed algorithms translate the theoretical guarantees developed in Section 4 into practical performance, and how different guidance strategies impact sample fidelity and stability. All experiments are conducted using pretrained diffusion models, emphasizing that the guidance procedure operates as a lightweight post-training mechanism without modifying the underlying score network.

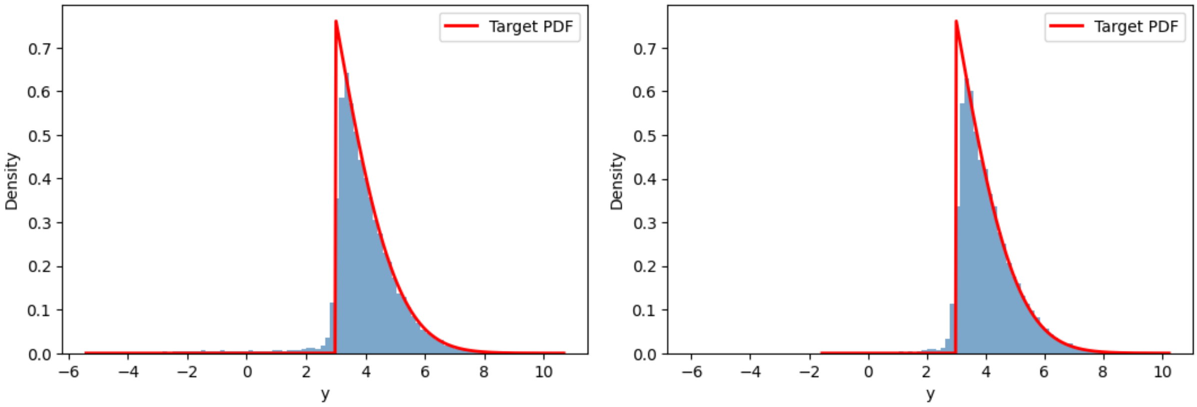
6.1 Synthetic examples

To demonstrate the effectiveness of our framework, we consider one-dimensional and two-dimensional toy examples.

First, suppose we aim to generate data from the Gaussian distribution $\mathcal{N}(1,4)$, and define the endogenous guidance set as $S = (3,\infty)$. When the target distribution is Gaussian, its marginal distributions along the diffusion process remain Gaussian (e.g., under a variance-exploding SDE of the form $d\mathbf{x} = \sigma^t d\mathbf{w}, t \in [0,1]$). As a result, the corresponding score functions are available in closed form, and we do not need to train a neural network to estimate them via *denoising score matching* (DSM) [89].

We evaluate Algorithm A, *Conditional Diffusion Guidance via Martingale Loss* (CDG-ML), and Algorithm B, *Conditional Diffusion Guidance via Martingale–Covariation Loss* (CDG-MCL). The corresponding results are reported in Figures 1a and 1b, respectively. To further quantify the agreement between the generated samples and the target distribution, we conduct a **Kolmogorov–Smirnov (K–S) test**. CDG-ML yields a K–S statistic of 0.0694 with a p-value of 7.1×10^{-126}, while CDG-MCL yields a K–S statistic of 0.0437 with a p-value of 4.1×10^{-50}. In both cases, the relatively small K–S statistics indicate that the empirical distributions closely match the target conditional distribution $\mathcal{N}(1,4) \mid (3,\infty)$, with CDG-MCL exhibiting a closer fit.

Next, we consider a two-dimensional synthetic example with target distribution $\mathcal{N}(0,4I_2)$ and endogenous guidance set $S = (1,\infty) \times (1,\infty)$. As in the one-dimensional case, the marginal distributions along the diffusion process admit closed-form expressions. The histogram of samples from the true conditional distribution is shown in Figure 2. To quantitatively assess performance, we compute the **2-Wasserstein distance**. Algorithm A yields a distance of 0.3451, while Algorithm B achieves a substantially smaller distance of 0.0765. The corresponding generated histograms are presented in Figures 3a and 3b, respectively. Both methods produce reasonable approximations of the target conditional distribution $\mathcal{N}(0,4I_2) \mid \big((1,\infty) \times (1,\infty)\big)$, with Algorithm B providing a closer match.



(a) CDG-ML on $\mathcal{N}(1,4)|(3,\infty)$

(b) CDG-MCL on $\mathcal{N}(1,4)|(3,\infty)$

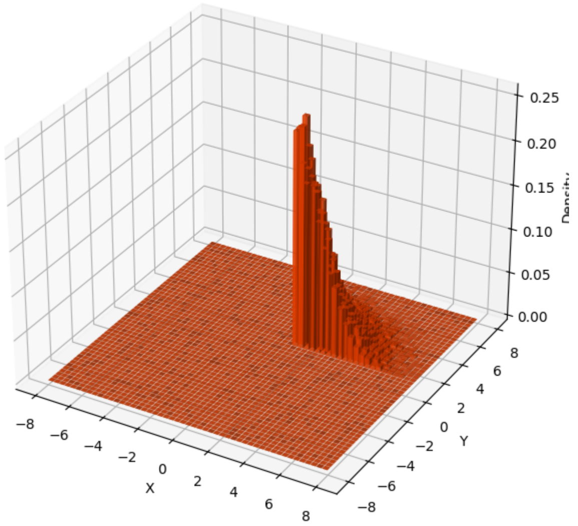
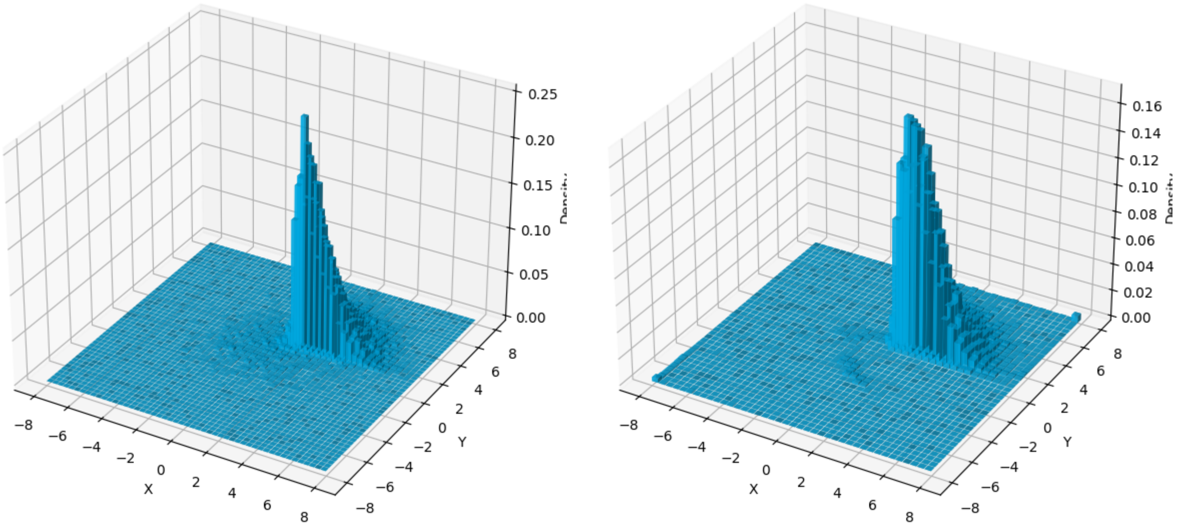


Figure 2: Histogram of Truncated Normal $\mathcal{N}(0,4I_2)|(1,\infty)\times(1,\infty)$



(a) Generated histogram via CDG-ML

(b) Generated histogram via CDG-MCL

6.2 Stress Testing

In this section, we illustrate how our Conditional Diffusion Guidance framework can produce controlled perturbations of high-dimensional return distributions, allowing one to probe the sensitivity of portfolio-level risk and tail behavior under targeted distributional shifts while preserving realistic dependence structures. Specifically, we apply our **CDG-ML** and **CDG-MCL** algorithms to real-world financial data and evaluate its economic relevance in constructing [33]: (1) equal weight portfolios, (2) Markowitz minimum variance portfolios[61] and (3) risk parity portfolios[68].

We use daily log-return data for U.S. stocks: AAPL, AMZN, TSLA, JPM from Jun 30, 2010, to Oct 7, 2025. We applied a mild 0.5% winsorization to the data, removed weekday (Monday–Friday) seasonal effects, and standardized the series by dividing by its long-term standard deviation. We use standardized daily log-return data to construct a joint dataset of d stocks over a rolling window of N consecutive days. Each window is shifted by one day to create overlapping samples. In our setting, $N = 64$ (approximately three months) and $d = 4$.

We choose our *guidance set* S to be some stress scenario, e.g. selecting certain stocks whose cumulative log returns over the last k days fall below a specified threshold (In our setting, let TSLA last 10 days cumulative log-return smaller than -10%). We then apply three portfolio management strategies: (1) equal weight, (2) Markowitz minimum variance, and (3) risk parity on the first $N-k$ days of the $N \times d$ data generated by **CDG-ML** and **CDG-MCL**, respectively.

The resulting portfolios are evaluated by comparing their cumulative returns over the final m days ($m \leq k$) with those observed under real market conditions, where the same subset of stocks exhibits cumulative log returns below the threshold. In our empirical example, we set $k = 10$ and $m = 5$. The detailed pipeline is shown in Appendix E.

In our inference process, we employ the DDIM sampler, a deterministic sampler obtained by discretizing the probability-flow ODE associated with the diffusion model [73]. Moreover, thanks to the idea of the generalized Bayesian method by introducing a power likelihood [41], we add one more hyperparameter η to adjust the magnitude of guidance. So the reverse time sampling process becomes:

$$dY_t^S = \left(-f(T-t, Y_t^S) + \frac{1}{2}\, g^2(T-t)\left(s_\theta(T-t, Y_t^S) + \eta \nabla \log h(t, Y_t^S)\right)\right) dt, \tag{6.1}$$

with $Y_0^S \sim p_{\text{noise}}(\cdot)$ and we have:

$$f(t,x) = -\frac{1}{2}\beta(t)x, \quad g(t) = \sqrt{\beta(t)}.$$

The noise scales of chosen to be linear:

$$\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})\, t, \quad t \in [0,1].$$

Moreover, using the idea that the step size should be scheduled according to the noise level [77], we build the VP-SDE time grid by spacing time points according to the noise level. This produces a decreasing time grid that allocates smaller steps when approaching the final stage. Figures 4 and 5 show how **CDG-ML** and **CDG-MCL** perform in recovering last-week portfolio returns under the scenario where the cumulative log-return of TSLA over the last 10 days is smaller than -10%. We can see that **CDG-ML** provides better results by introducing less bias. The reason is that the training of q_{ψ_*} in Algorithm B highly relies on the accuracy of h_{ϕ_*}, which introduces more uncertainty. Moreover, as we have the freedom to choose the hyperparameter η to adjust the magnitude of our guidance, we find that **CDG-ML** admits larger η (10^2 in practice), whereas **CDG-MCL** only admits much smaller η (2 to 4 in practice). The detailed results are shown in Table 1, and the bold entries indicate the closest statistics to the real ones under the three portfolios, respectively.

Algorithm E Portfolio Construction and Evaluation under Guidance Conditions

```
Input: Standardized daily log-return data $\{r_{t,i}\}$ for $d$ stocks over $T$ days; window size $N$ (e.g., 64); guidance window $k$ (e.g., 10); evaluation window $m$ (e.g., 5); threshold $\tau$ for guidance selection.
Output: Cumulative return comparison between generated and real portfolios.

Step 1. Data Preparation:
for $t = 1$ to $T - N + 1$ do
    Construct a rolling window dataset $\mathbf{R}_t = [r_{t,i}, r_{t+1,i}, \ldots, r_{t+N-1,i}]_{i=1}^{d}$.
end for

Step 2. Guidance Set Selection:
Select a subset of stocks $\mathcal{G}$ such that their cumulative log-returns over the last $k$ days satisfy $\sum_{j=N-k+1}^{N} r_{t+j,i} < \tau, \ \forall i \in \mathcal{G}$.

Step 3. Portfolio Optimization:
for each model $\in \{\text{CDG-ML}, \text{CDG-MCL}\}$ do
    Generate synthetic data $\hat{\mathbf{R}}^{(\text{model})} \in \mathbb{R}^{N \times d}$.
    Apply the following portfolio strategies on the first $N - k$ days:
      1. Equal-weight portfolio: $w_i = 1/d$
      2. Markowitz mean-variance portfolio
      3. Risk-parity portfolio
end for

Step 4. Evaluation:
For each portfolio, compute the cumulative return $R_{\text{cum}}$ over the final $m$ days ($m \leq k$):
Compare $R_{\text{cum}}$ under synthetic data with that under real market conditions where $\sum_{j=N-k+1}^{N} r_{t+j,i} < \tau$ for $i \in \mathcal{G}$.
```

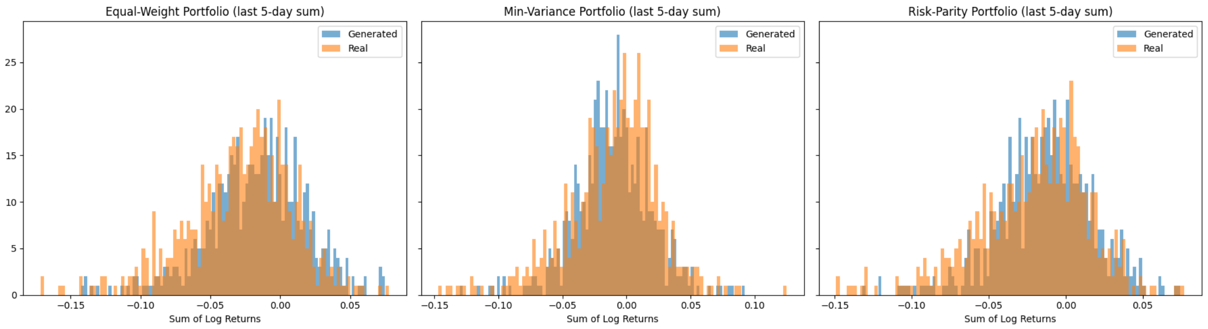


Figure 4: Histograms of portfolio constructions via CDG-ML

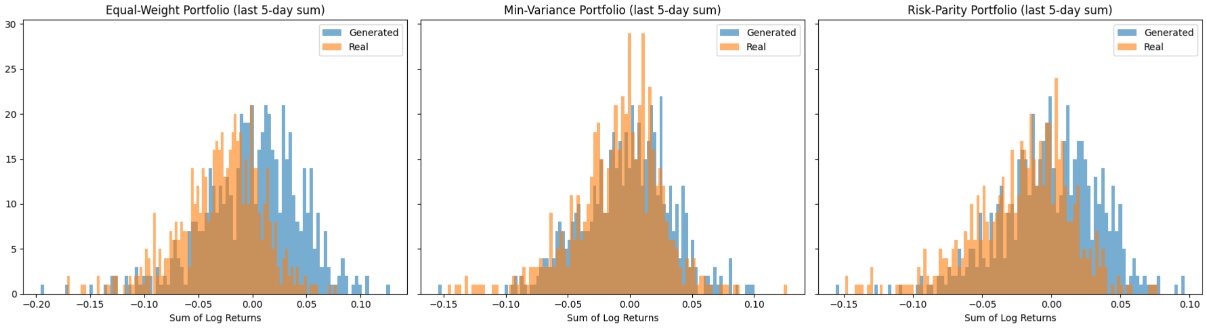


Figure 5: Histograms of portfolio constructions via CDG-MCL

Table 1: Different portfolios under CDG-ML and CDG-MCL.

Portfolio	η	Mean	Std	5% Quantile	10% Quantile
		CDG-ML			
Equal Weight	150	-0.01671	0.03528	-0.07765	-0.05969
Min Variance	150	**-0.0107**	0.02935	-0.05792	-0.04515
Risk Parity	150	-0.01346	0.03045	-0.06351	-0.04830
Equal Weight	200	-0.02283	0.03366	-0.07734	-0.06229
Min Variance	200	-0.01408	0.02814	-0.05851	-0.04636
Risk Parity	200	-0.01832	0.02904	-0.06501	-0.05206
Equal Weight	300	**-0.02684**	**0.03805**	-0.09318	**-0.07436**
Min Variance	300	-0.01645	0.03274	**-0.06807**	**-0.05313**
Risk Parity	300	**-0.02144**	**0.03316**	-0.07917	**-0.05969**
Equal Weight	400	-0.03362	0.03651	**-0.09959**	-0.07433
Min Variance	400	-0.01781	0.03219	-0.06747	-0.05156
Risk Parity	400	-0.02606	0.03171	**-0.08027**	-0.05933
		CDG-MCL			
Equal Weight	2.00	0.00269	0.03852	-0.06214	-0.04575
Min Variance	2.00	0.00318	0.03157	-0.04954	-0.03688
Risk Parity	2.00	0.00310	0.03302	-0.05156	-0.03843
Equal Weight	3.50	0.00038	0.04985	-0.08207	-0.06466
Min Variance	3.50	0.00376	**0.03548**	-0.05806	-0.04131
Risk Parity	3.50	0.00232	0.03936	-0.06270	-0.04871
Equal Weight	3.75	-0.00027	0.05263	-0.08791	-0.06763
Min Variance	3.75	0.00254	0.03717	-0.06120	-0.04240
Risk Parity	3.75	0.00142	0.04193	-0.06425	-0.05295
Equal Weight	4.00	-0.00351	0.05507	-0.10200	-0.07298
Min Variance	4.00	0.00223	0.03862	-0.06777	-0.04824
Risk Parity	4.00	-0.00054	0.04347	-0.07861	-0.05559
		Real Data			
Equal Weight	–	-0.02882	0.03820	-0.09688	-0.07879
Min Variance	–	-0.01029	0.03601	-0.07422	-0.05638
Risk Parity	–	-0.02125	0.03606	-0.08584	-0.06821

6.3 Supply Chain Simulation

In this section, we demonstrate how conditional diffusion guidance can be used as a principled, data-driven tool for generating stress scenarios in complex supply-chain and queueing systems, enabling downstream simulation-based evaluation and capacity planning.

To start, we first demonstrate the performance of our framework in simulating conditional

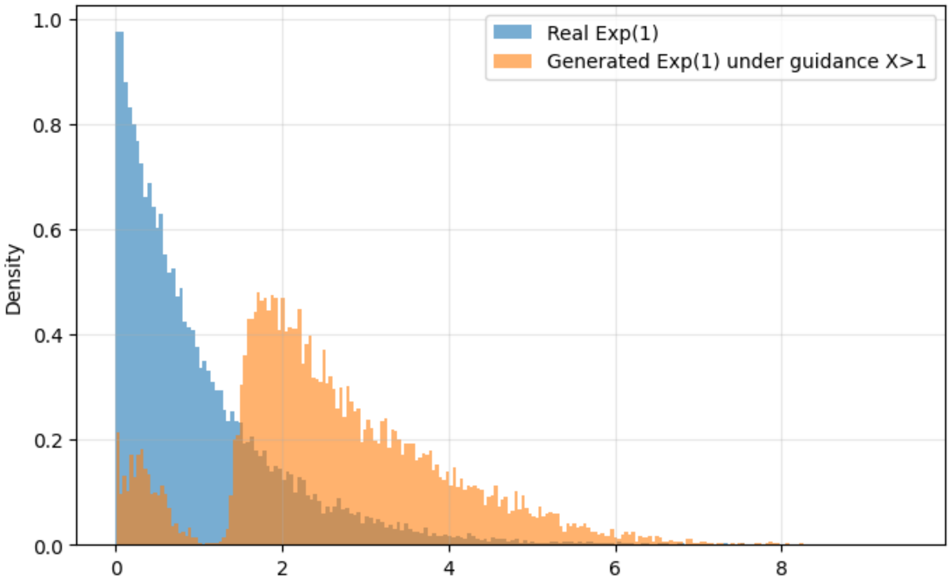


Figure 6: Density Comparison of Exp(1) and Exp(1)$|(1, \infty)$

arrivals of events. Note that one advantage of using our conditional diffusion guidance is that we can tune the degree of softness by varying η, and generating samples that are empirically biased to guidance set but still maintain diversity to some extent. As we shown in Figure 6, although we generate exponential(1) data conditioning on $(1, +\infty)$, it still admits small proportion (around 9 %) of data in $(0, 1)$, which is more realistic compared to hard conditioning.

Next, we test our framework on generating conditional arrival times and service times in queueing networks. We consider the hospital environment provided by QGym [12], which consists of $M = 8$ patient types (Cardiology, Surgery, Orthopedics, Respiratory disease, Gastroenterology and endoscopy, Renal disease, General Medicine, Neurology) and $N = 11$ wards, each ward having a fixed number of beds corresponding to the available servers. Admission is subject to routing constraints: each ward can serve only 1–2 specific patient types, and patients who cannot be admitted immediately wait in the appropriate queue. When a bed becomes available, the ward follows a classic policy called $c\mu$ assignment rule [18]: the action (i.e., job-server assignment) is decided by

$$\max_{a \in \mathcal{A}} \sum_{1 \le i \le M, 1 \le j \le N} c_i \mu_{ij} a_{ij}$$

where $\mathcal{A} \subset \mathbb{R}^{M \times N}$ captures the routing constraints mentioned above. In this case, server j prioritizes the queue with a larger $c_i\mu_{ij}$-index, where c_i denotes the holding cost per job per unit time for queue i $(1 \le i \le M)$, and μ_{ij} is the service rate when server j $(1 \le j \le N)$ processes a job from queue i.

Based on the literature [2, 100, 50, 21], there is consistent evidence across temperate climate zones (e.g., North America, Europe, and North China) that healthcare burdens exhibit strong seasonal concentration, peaking in winter and, for certain specialties such as surgery and surgical site infections, also during summer or high-temperature periods. In the context of a conditional diffusion model, this periodic structure can be naturally exploited by *designing guidance* that allocates higher arrival intensities and slower service rates—reflecting congestion—during these critical seasonal windows. As a representative stress scenario, we consider the flu season, in which elevated arrival rates combined with reduced service capacity imply that more patients enter the system while recovery takes longer, leading intuitively to a pronounced increase in total queue length.

Using the default hospital setting in the **Qgym** environment [12], the $c\mu$**-rule** yields a

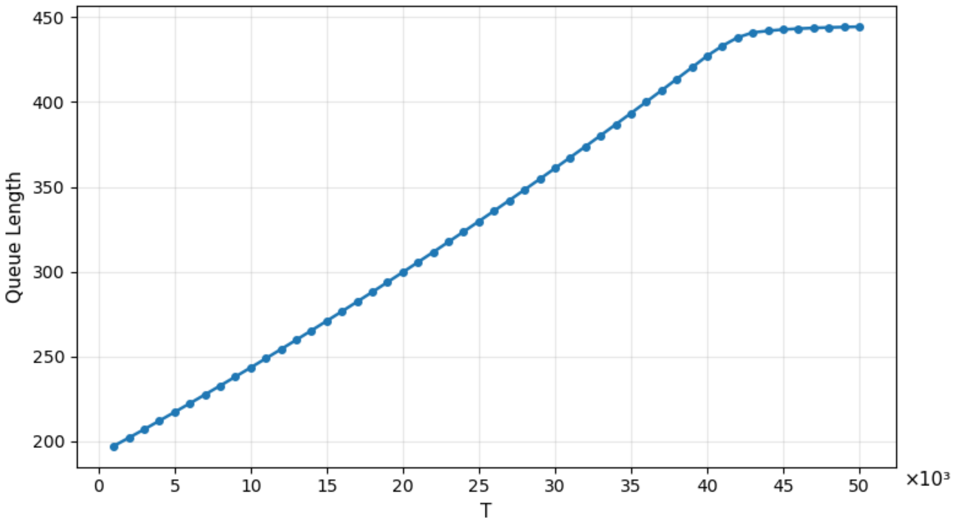


Figure 7: Limiting number of original queue length

stable total queue length (aggregated over 8 patient types) that converges to approximately 400 (Figure 7). Let $Z_1 \sim \text{Exp}(\lambda_1)$ and $Z_2 \sim \text{Exp}(\lambda_2)$ denote the inter-arrival and service times, respectively. To model seasonal effects, we condition the arrivals and services as $Z_1 \mid Z_1 < \frac{3}{2}\frac{1}{\lambda_1}$ and $Z_2 \mid Z_2 \frac{1}{2}\frac{1}{\lambda_2}$, which induces an unstable queueing regime.

Here, we implement the diffusion process using a drift-free VE–SDE with exponentially increasing diffusion $g(t) = \sigma^t$, and discretize time on a non-uniform grid constructed so that the marginal perturbation standard deviation increases linearly across steps, yielding an analytically tractable schedule that improves numerical stability during sampling. In addition, we set $\lambda_1 = \lambda_2 = 1$ and choose $\eta \approx 12$.

Under this setting, both the hard truncated exponential model (which admits an explicit solution) and our soft-guidance framework in (6.1) lead to diverging queue lengths. Nevertheless, Figure 8 shows that the diffusion model with soft guidance leads to smaller divergence rates, as it captures a broader range of arrival and service times.

We observe that queue lengths in wards with restricted routing—where certain patient types can be served only by specific wards—are highly sensitive to variations in both arrival and service rates. By inspecting the simulation logs, we find that the queues for patient types 6 and 7 exhibit explosive growth; accordingly, we double the number of servers assigned to these wards. The resulting performance is shown in Figure 9, where the system converges after approximately 30,000 days of simulation. As in the unstable regime, conditional diffusion generation consistently produces shorter queues than the blue benchmark, reflecting the smoothing effect induced by soft guidance. These results highlight the sensitivity of the queueing system to small distributional shifts and underscore the need for additional server capacity in critical wards during peak seasons. They also emphasize the importance of realistic simulators for stress testing, especially in unstable or near-unstable regimes.

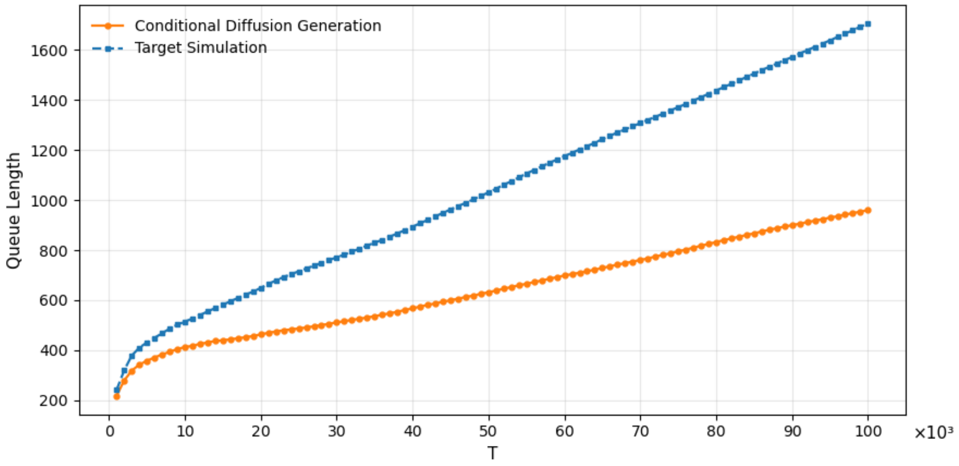


Figure 8: Comparison of queue length under the guidance of shorter patient inter-arrival times and longer service times (blue: hard truncated exponential distribution; orange: soft guidance).

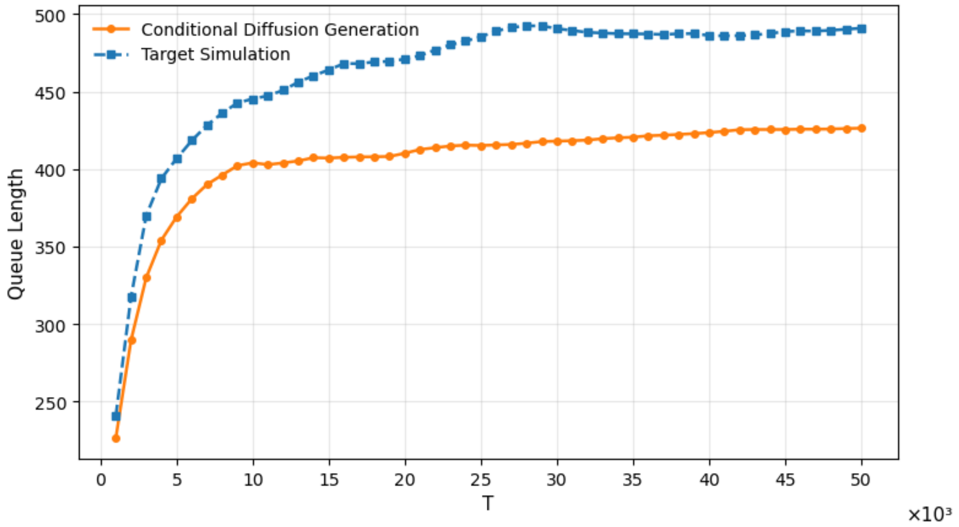


Figure 9: Comparison of queue length with adding more servers.

7 Conclusion

Acknowledgment: We thank David Yao for helpful discussions. Tang is partially supported by NSF grant DMS-2206038, the Innovation Hub grant, and the Tang family Assistant Professorship. Xu is partially supported by the NSF CAREER Award DMS-2524465 and a gift fund from Point72.

References

[1] B. D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Process. Appl.*, 12(3):313–326, 1982.

[2] C. A. Anthony et al. The seasonal variability in surgical site infections and the association with warmer weather: A population-based investigation. *Infection Control & Hospital Epidemiology*, 38(7):809–816, 2017.

[3] S. Asmussen and P. W. Glynn. *Stochastic simulation: algorithms and analysis*, volume 57 of *Stochastic Modelling and Applied Probability*. Springer, New York, 2007.

[4] A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein. Universal guidance for diffusion models. In *CVPR*, pages 843–852, 2023.

[5] A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein. Universal guidance for diffusion models. In *CVPR*, pages 843–852, 2023.

[6] O. E. Barndorff-Nielsen and N. Shephard. Econometric analysis of realized covariation: High frequency based covariance, regression, and correlation in financial economics. *Econometrica*, 72(3):885–925, 2004.

[7] A. Benveniste, M. Métivier, and P. Priouret. *Adaptive algorithms and stochastic approximations*, volume 22 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, 1990.

[8] J. Blanchet and H. Lam. Rare event simulation techniques. In *WSC*, pages 146–160, 2011.

[9] J. Blanchet and H. Lam. State-dependent importance sampling for rare-event simulation: An overview and recent advances. *Surveys in Operations Research and Management Science*, 17(1):38–59, 2012.

[10] A. Borji. Pros and cons of GAN evaluation measures. *Comput. Vis. Image Underst.*, 179:41–65, 2019.

[11] J. A. Bucklew. *Introduction to rare event simulation*. Springer Series in Statistics. Springer-Verlag, New York, 2004.

[12] H. Chen, A. Li, E. Che, J. Dong, T. Peng, and H. Namkoong. Qgym: Scalable simulation and benchmarking of queuing network controllers. In *Advances in Neural Information Processing Systems*, volume 37, pages 92401–92419, 2024.

[13] J. Chen, J.-S. Byun, M. Elsner, and A. Perrault. Dlpo: Diffusion model loss-guided reinforcement learning for fine-tuning text-to-speech diffusion models. 2024. arXiv:2405.14632.

[14] M. Chen, K. Huang, T. Zhao, and M. Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *ICML*, volume 40, pages 4672–4712, 2023.

[15] M. Chen, S. Mei, J. Fan, and M. Wang. An overview of diffusion models: Applications, guided generation, statistical rates and optimization. 2024. arXiv:2404.07771.

[16] S. Chen, S. Chewi, J. Li, Y. Li, A. Salim, and A. R. Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *ICLR*, 2023.

[17] K. Christensen, M. S. Nielsen, and M. Podolskij. High-dimensional estimation of quadratic variation based on penalized realized variance. *Stat. Inference Stoch. Process.*, 26(2):331–359, 2023.

[18] D. R. Cox and W. L. Smith. *Queues.* Methuen, London, 5th edition, 1961.

[19] A. Denker, F. Vargas, S. Padhy, K. Didi, S. V. Mathis, R. Barbano, V. Dutordoir, E. Mathieu, U. J. Komorowska, and P. Lio. Deft: Efficient fine-tuning of diffusion models by learning the generalised h-transform. In *Neurips*, volume 37, 2024.

[20] P. Dhariwal and A. Nichol. Diffusion models beat GANs on image synthesis. In *Neurips*, volume 34, pages 8780–8794, 2021.

[21] D. H. Do. Winter peaks in in-hospital cardiac arrest: Patient susceptibility or hospital overcrowding? *JACC: Clinical Electrophysiology*, 9(8 Pt 3):1768–1770, 2023.

[22] S. S. Dragomir. *Some Gronwall type inequalities and applications.* Nova Science Publishers, Inc., Hauppauge, NY, 2003.

[23] Y. Du, M. Plainer, R. Brekelmans, C. Duan, F. Noé, C. P. Gomes, A. Aspuru-Guzik, and K. Neklyudov. Doob's lagrangian: A sample-efficient variational approach to transition path sampling. In *Neurips*, volume 37, 2024.

[24] J. Fan. A selective overview of nonparametric methods in financial econometrics. *Stat. Sci.*, pages 317–337, 2005.

[25] Y. Fan and K. Lee. Optimizing DDPM sampling with shortcut fine-tuning. 2023. arXiv:2301.13362.

[26] Y. Fan, O. Watkins, Y. Du, H. Liu, M. Ryu, C. Boutilier, P. Abbeel, M. Ghavamzadeh, K. Lee, and K. Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Neurips*, volume 36, 2023.

[27] P. Fitzsimmons, J. Pitman, and M. Yor. Markovian bridges: construction, Palm interpretation, and splicing. In *Seminar on Stochastic Processes, 1992 (Seattle, WA, 1992)*, volume 33 of *Progr. Probab.*, pages 101–134. Birkhäuser Boston, Boston, MA, 1993.

[28] E. Fournié, J.-M. Lasry, J. Lebuchoux, P.-L. Lions, and N. Touzi. Applications of Malliavin calculus to Monte Carlo methods in finance. *Finance Stoch.*, 3(4):391–412, 1999.

[29] A. Friedman. *Partial differential equations of parabolic type.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1964.

[30] X. Gao, H. M. Nguyen, and L. Zhu. Wasserstein convergence guarantees for a general class of score-based generative models. 2023. arXiv:2311.11003.

[31] X. Gao, J. Zha, and X. Y. Zhou. Reward-directed score-based diffusion models via q-learning. 2024. arXiv:2409.04832.

[32] Google. State-of-the-art video and image generation with veo 2 and imagen 3. `https://blog.google/technology/google-labs/video-image-generation-update-december-2024/`, 2024. Accessed: 2025-09-17.

[33] Z. Guo, J. Li, W. Tang, and D. D. Yao. Diffusion generative models meet compressed sensing, with applications to imaging and finance. 2025. arXiv:2509.03898.

[34] Y. Han, M. Razaviyayn, and R. Xu. Neural network-based score estimation in diffusion models: Optimization and generalization. In *ICLR*, 2024.

[35] U. G. Haussmann and E. Pardoux. Time reversal of diffusions. *Ann. Probab.*, 14(4):1188–1205, 1986.

[36] A. Hiranaka, S.-F. Chen, C.-H. Lai, D. Kim, N. Murata, T. Shibuya, W.-H. Liao, S.-H. Sun, and Y. Mitsufuji. Hero: Human-feedback efficient reinforcement learning for online diffusion model finetuning. 2024. arXiv:2410.05116.

[37] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Neurips*, volume 33, pages 6840–6851, 2020.

[38] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

[39] M. Hoffmann. Adaptive estimation in diffusion processes. *Stochastic Process. Appl.*, 79(1):135–163, 1999.

[40] M. Hoffmann. L_p estimation of the diffusion coefficient. *Bernoulli*, 5(3):447–481, 1999.

[41] C. C. Holmes and S. G. Walker. Assigning a value to a power likelihood in a general bayesian model. *Biometrika*, 104(2):497–503, 2017.

[42] S. Howard, N. Nüsken, and J. Pidstrigach. Control consistency losses for diffusion bridges. *arXiv preprint arXiv:2512.05070*, 2025.

[43] A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, 2005.

[44] A. M. Ilin, A. S. Kalasnikov, and O. A. Oleinik. Second-order linear equations of parabolic type. *Uspehi Mat. Nauk*, 17(3):3–146, 1962.

[45] J. Jacod. Convergence en loi de semimartingales et variation quadratique. In *Seminar on Probability, XV (Univ. Strasbourg, Strasbourg, 1979/1980) (French)*, volume 850 of *Lecture Notes in Math.*, pages 547–560. Springer, Berlin, 1981.

[46] J. Jacod. Asymptotic properties of realized power variations and related functionals of semimartingales. *Stochastic Process. Appl.*, 118(4):517–559, 2008.

[47] J. Jacod and A. N. Shiryaev. *Limit theorems for stochastic processes*, volume 288 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, second edition, 2003.

[48] B. Jamison. The Markov processes of Schrödinger. *Z. Wahrscheinlichkeitstheor. Verw. Geb.*, 32(4):323–331, 1975.

[49] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Neurips*, volume 35, pages 26565–26577, 2022.

[50] J. Kaufman, A. M. Vicedo-Cabrera, V. Tam, L. Song, E. Coffel, and G. E. Tasian. The impact of heat on kidney stone presentations in south carolina under two climate change scenarios. *Scientific Reports*, 12:369, 2022.

[51] S. Khalafi, D. Ding, and A. Ribeiro. Constrained diffusion models via dual training. In *Neurips*, volume 37, 2024.

[52] S. Khanna, S. Kharbanda, S. Li, H. Varma, E. Wang, S. Birnbaum, Z. Luo, Y. Miraoui, A. Palrecha, and S. Ermon. Mercury: Ultra-fast language models based on diffusion. 2025. arXiv:2506.17298.

[53] F. Koehler, A. Heckett, and A. Risteski. Statistical efficiency of score matching: The view from isoperimetry. In *ICLR*, 2023.

[54] H. Lee, J. Lu, and Y. Tan. Convergence for score-based generative modeling with polynomial complexity. In *Neurips*, volume 35, pages 22870–22882, 2022.

[55] K. Lee, H. Liu, M. Ryu, O. Watkins, Y. Du, C. Boutilier, P. Abbeel, M. Ghavamzadeh, and S. S. Gu. Aligning text-to-image models using human feedback. 2023. arXiv:2302.12192.

[56] G. Li, Y. Wei, Y. Chen, and Y. Chi. Towards faster non-asymptotic convergence for diffusion-based generative models. In *ICLR*, 2024.

[57] B. Liu, S. Shao, B. Li, L. Bai, Z. Xu, H. Xiong, J. Kwok, S. Helal, and Z. Xie. Alignment of diffusion models: Fundamentals, challenges, and future. 2024. arXiv:2409.07253.

[58] H. Liu, T. Zhu, N. Jia, J. He, and Z. Zheng. Learning to simulate from heavy-tailed distribution via diffusion model. 2024. SSRN 4975931.

[59] J. Ma, G. Wang, J. Zhang, and Z. X. Yu. Reinforcement learning for continuous time stochastic optimization with model uncertainty. 2026. Working paper.

[60] Z. Ma, Y. Zhang, G. Jia, L. Zhao, Y. Ma, M. Ma, G. Liu, K. Zhang, J. Li, and B. Zhou. Efficient diffusion models: A comprehensive survey from principles to practices. 2024. arXiv:2410.11795.

[61] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[62] P.-A. Meyer. *Martingales and stochastic integrals. I. Lecture Notes in Mathematics. Vol. 284.* Springer-Verlag, Berlin-New York, 1972.

[63] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. Mcgrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, volume 39, pages 16784–16804, 2022.

[64] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li. Large language diffusion models. 2025. arXiv:2502.09992.

[65] P. Norvig. The unreasonable effectiveness of data. 2012. Video available at `https://www.youtube.com/watch?v=yvDCzhbjYWs&t=6s`, 37:42–39:06.

[66] OpenAI. Sora: Creating video from text. `https://openai.com/sora`, 2024. Accessed: 2025-09-17.

[67] J. Pidstrigach, E. Baker, C. Domingo-Enrich, G. Deligiannidis, and N. Nüsken. Conditioning diffusions using malliavin calculus. 2025. arXiv:2504.03461.

[68] E. Qian. Risk parity portfolios: Efficient portfolios through true diversification. Working paper, PanAgora Asset Management, 2005. PanAgora Asset Management Working Paper.

[69] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. 2022. arXiv:2204.06125.

[70] D. Revuz and M. Yor. *Continuous martingales and Brownian motion*, volume 293 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, third edition, 1999.

[71] L. C. G. Rogers and D. Williams. *Diffusions, Markov processes, and martingales. Vol. 2*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 2000. Itô calculus, Reprint of the second (1994) edition.

[72] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[73] J. Sheng, H. Zhao, H. Chen, D. D. Yao, and W. Tang. Understanding sampler stochasticity in training diffusion models for rlhf. 2025. arXiv:2510.10767.

[74] R. Shenoy, Z. Pan, K. Balakrishnan, Q. Cheng, Y. Jeon, H. Yang, and J. Kim. Gradient-free classifier guidance for diffusion model sampling. 2024. arXiv:2411.15393.

[75] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, and O. Gafni. Make-a-video: Text-to-video generation without text-video data. In *ICLR*, 2023.

[76] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Neurips*, volume 32, 2019.

[77] Y. Song and S. Ermon. Improved techniques for training score-based generative models. In *Neurips*, volume 33, pages 12438–12448, 2020.

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

[79] T. Sorensen, J. Moore, J. Fisher, M. L. Gordon, N. Mireshghallah, C. M. Rytting, A. Ye, L. Jiang, X. Lu, N. Dziri, T. Althoff, and Y. Choi. Position: A roadmap to pluralistic alignment. In *ICML*, volume 41, 2024.

[80] D. W. Stroock and S. R. S. Varadhan. *Multidimensional diffusion processes*, volume 233 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, 1979.

[81] W. Tang and H. Zhao. Score-based diffusion models via stochastic differential equations. *Statistic Surveys*, 19:28–64, 2025.

[82] W. Tang and H. Zhao. Contractive diffusion probabilistic models. 2026. To appear in SIAM J. Imaging Sci.

[83] W. Tang and F. Zhou. Fine-tuning of diffusion models via stochastic control: entropy regularization and beyond. 2026. To appear in ACC.

[84] W. Tang and X. Y. Zhou. Regret of exploratory policy improvement and q-learning. 2024. arXiv:2411.01302.

[85] M. Uehara, Y. Zhao, T. Biancalani, and S. Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. 2024. arXiv:2407.13734.

[86] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, T. Biancalani, and S. Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. 2024. arXiv:2402.15194.

[87] P. Vaeth, D. Kumar, B. Paassen, and M. Gregorová. Diffusion classifier guidance for non-robust classifiers. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 206–221, 2025.

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

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

[90] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik. Diffusion model alignment using direct preference optimization. In *CVPR*, pages 8228–8238, 2024.

[91] Y. Wang, Y. He, and M. Tao. Evaluating the design space of diffusion-based generative models. In *Neurips*, volume 37, pages 19307–19352, 2024.

[92] G. I. Winata, H. Zhao, A. Das, W. Tang, D. D. Yao, S.-X. Zhang, and S. Sahu. Preference tuning with human feedback on language, speech, and vision tasks: A survey. 2024. arXiv:2409.11564.

[93] Y. Wu, M. Chen, Z. Li, M. Wang, and Y. Wei. Theoretical insights for diffusion guidance: A case study for Gaussian mixture models. In *ICML*, volume 41, 2024.

[94] Y. Wu, Y. Chen, and Y. Wei. Stochastic Runge-Kutta methods: Provable acceleration of diffusion models. 2024. arXiv:2410.04760.

[95] K. Yang, J. Tao, J. Lyu, C. Ge, J. Chen, W. Shen, X. Zhu, and X. Li. Using human feedback to fine-tune diffusion models without any reward model. In *CVPR*, pages 8941–8951, 2024.

[96] H. Yuan, K. Huang, C. Ni, M. Chen, and M. Wang. Reward-directed conditional diffusion: Provable distribution estimation and reward improvement. In *Neurips*, volume 36, 2023.

[97] Y. Zhang, E. Tzeng, Y. Du, and D. Kislyuk. Large-scale reinforcement learning for diffusion models. In *ECCV*, pages 1–17, 2024.

[98] H. Zhao, H. Chen, J. Zhang, D. D. Yao, and W. Tang. Scores as Actions: a framework of fine-tuning diffusion models by continuous-time reinforcement learning. 2024. arXiv:2409.08400.

[99] H. Zhao, H. Chen, J. Zhang, D. D. Yao, and W. Tang. Scores as Actions: fine tuning diffusion generative models by continuous-time reinforcement learning. In *ICML*, 2025.

[100] Y. Zhou, J. Smith, L. Tan, and Q. Wang. Seasonal variation of stroke incidence in a temperate climate zone: a multi-centre cohort study. *Scientific Reports*, 14(1):75765, 2024.