ACCELERATED SEQUENTIAL FLOW MATCHING: A BAYESIAN FILTERING PERSPECTIVE

Yinan Huang, Hans Hao-Hsun Hsu, Junran Wang, Bo Dai, Pan Li
Georgia Institute of Technology
{yhuang903,hhsu72,jwang3668}@gatech.edu, bodai@cc.gatech.edu
panli@gatech.edu

ABSTRACT

Sequential prediction from streaming observations is a fundamental problem in stochastic dynamical systems, where inherent uncertainty often leads to multiple plausible futures. While diffusion and flow-matching models are capable of modeling complex, multi-modal trajectories, their deployment in real-time streaming environments typically relies on repeated sampling from a non-informative initial distribution, incurring substantial inference latency and potential system backlogs. In this work, we introduce *Sequential Flow Matching*, a principled framework grounded in Bayesian filtering. By treating streaming inference as learning a probability flow that transports the predictive distribution from one time step to the next, our approach naturally aligns with the recursive structure of Bayesian belief updates. We provide theoretical justification that initializing generation from the previous posterior offers a principled warm start that can accelerate sampling compared to naïve re-sampling. Across a wide range of forecasting, decision-making and state estimation tasks, our method achieves performance competitive with full-step diffusion while requiring only one or very few sampling steps, therefore with faster sampling. It suggests that framing sequential inference via Bayesian filtering provides a new and principled perspective towards efficient real-time deployment of flow-based models. Our code is available at https://github.com/Graph-COM/Sequential_Flow_Matching.

1 INTRODUCTION

Making predictions of evolving real-world systems from streaming observations is a fundamental problem, spanning diverse applications from weather forecasting (Lorenz, 2017), future video frame anticipation for decision making in autonomous systems (Yang et al., 2024), state estimation and control of complex physical systems such as fluid flows or magnetic plasma (Kalnay, 2003; Bewley, 2001; Degrave et al., 2022). A critical characteristic of these streaming environments is their inherent stochasticity: a single history of observations often admits multiple plausible futures. Deterministic predictions (such as simple regression) are often insufficient as they tend to average out distinct possibilities, leading to physically invalid states or blurred predictions. Moreover, many applications require long-horizon prediction, where the system anticipates future trajectories spanning multiple steps ahead for planning and monitoring. Ultimately, reliable decision-making requires moving beyond point estimation to maintain a probabilistic belief over the system's future trajectories.

Probability flow-based models, such as diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) and flow matching (Lipman et al., 2023; Liu et al., 2023), have emerged as compelling data-driven frameworks for modeling high-dimensional, multi-modal distributions. Their capacity to capture complex uncertainty and generate high-fidelity samples makes them particularly effective for modeling system dynamics and synthesizing coherent long-horizon trajectories. Consequently, there is growing interest in applying these models to sequential tasks, ranging from forecasting and state estimation to planning and control (Chen et al., 2024; Gao et al., 2023; Janner et al., 2022; Wei et al., 2024).

However, real-time deployment of these models for streaming environments remains a challenge. Continuously integrating *streaming observations* demands rapid processing to avoid delayed esti-

Table 1: Examples of sequential probabilistic inference $p(x_t|z_{\leq t})$.

Task	Predictive Variable	Observation	Inference Goal
Forecasting	$x_t = s_{t+1:t+H}$	$z_t = s_t$	$p(s_{t+1:t+H}\|s_{\leq t})$
Planning (imitation learning)	$x_t = (s_{t+1:t+H}, a_{t:t+H-1})$	$z_t = (s_t, a_{t-1})$	$p(s_{t+1:t+H}, a_{t:t+H-1}\|s_{\leq t}, a_{\leq t-1})$
State Estimation	$x_t = s_t$	$z_t \sim p(z_t\|s_t)$	$p(s_t\|z_{\leq t})$

mations, which typically involves a trade-off between accuracy and latency. Existing approaches either operate in an open-loop manner, generating a long-horizon future trajectory once from initial observations, thereby ignoring streaming data and accumulating errors over time; or they re-predict future trajectories by incorporating new observations, which requires tens to hundreds of network evaluations per update and incurs prohibitive inference latency (Janner et al., 2022; Chen et al., 2024; Zhou et al., 2025). Some recent work attempts to mitigate this cost by asynchronous denoising and reuse of historical predictions (Janner et al., 2022; Wei et al., 2025; Høeg et al., 2025). However, these methods fail to fully exploit the information encoded in prior predictions and to yield accurate long-horizon estimation (see more discussion in Section 4).

In this article, we formulate sequential prediction from streaming observations as a sequential probabilistic inference problem, and study how to accelerate flow-based models from a principled perspective grounded in **Bayesian filtering**. We argue that streaming tasks exhibit strong *temporal correlation*: the distribution of the system state at time $t-1$ is coherent with that of time t, and therefore constitutes a natural prior for subsequent inference. Rather than discarding past predictions, we propose **Sequential Flow Matching**, which explicitly learns the probability flow that transports the predictive distribution from one time step to the next. This recursive belief update naturally aligns with the Bayesian filtering framework and provides a principled warm start for generation. Our theoretical result further supports this intuition, showing that initializing generation from the previous belief reduces sampling error compared to restarting from a non-informative Gaussian distribution.

To learn the belief-to-belief probability flow, we propose a flow matching objective function that can be used to effectively finetune a pretrained flow-based model. We evaluate the proposed approach on a diverse set of streaming tasks, including forecasting, planning and control, and state estimation. Our results show that belief-update-based inference achieves a compelling trade-off between fidelity and efficiency: competitive performance to full-step diffusion with only one or a few sampling steps.

2 Preliminary

Flow matching and diffusion models. The idea of flow matching is to define an ordinary differential equation (ODE) $\frac{d}{d\tau}x(\tau) = v(x(\tau), \tau)$ that transports between the source distribution $x(1) \sim p_1$ and the target distribution $x(0) \sim p_0$ (Lipman et al., 2023; Liu et al., 2023). To avoid confusion with physical time t, throughout the paper we use the symbol τ to denote the virtual time variable (or noise level) in flow matching. By defining an interpolation path $x(\tau) = \alpha(\tau)x(0) + \sigma(\tau)x(1)$ and conditional velocity $\dot{x}(\tau) = \dot{\alpha}(\tau)x(0) + \dot{\sigma}(\tau)x(1)$, the marginal velocity $v(x(\tau), \tau)$ can be expressed by $v(x(\tau), \tau) = \mathbb{E}(\dot{x}(\tau)|x(\tau))$. The marginal velocity is learned by a neural network $v_\theta(x(\tau), \tau)$ through optimization:

$$\min_\theta \mathbb{E}_{(x(0),x(1))\sim\pi,\tau\sim\text{Uniform}(0,1)} \|v_\theta(x(\tau), \tau) - \dot{x}(\tau)\|^2, \tag{1}$$

where π is a coupling of p_0, p_1, i.e., π represents a joint distribution of $x(0), x(1)$ whose marginals are p_0, p_1. To sample from target distribution p_0, we solve the ODE $dx(\tau)/d\tau = v_\theta(x(\tau), \tau)$ starting from source distribution $x(1) \sim p_1$ that is easy to sample from. The probability flow ODE counterpart of Diffusion models can be viewed as a special case of flow matching with $p_1 = \mathcal{N}(0, I)$ (Song et al., 2021b).

3 Bayesian Sequential Flow Matching

3.1 Problem Setup and Bayesian Filtering

Sequential probabilistic inference. We study a sequential probabilistic inference problem, i.e., performing probabilistic inference online over multiple rounds as new data arrives, which can cover a

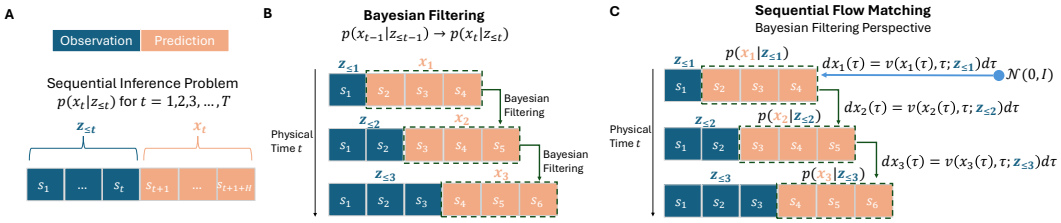

Figure 1: Illustration of the sequential inference problems using streaming forecasting as examples. **A**: The sequential inference problem $p(x_t|z_{\leq t})$ that predicts future states given historical observations over time $t = 1, 2, ..., T$. **B**: Bayesian filtering framework refines previous predictions based on latest observations to obtain new estimations. **C**: Following the idea of Bayesian filtering, sequential flow matching leverages a probability flow to recursively transport from previous belief $p(x_{t-1}|z_{\leq t-1})$ to current belief $p(x_t|z_{\leq t})$.

wide range of streaming tasks. Let x_t be a random variable to be predicted at time t, and z_t be the arrived observation at time t. Given streaming observations $z_{\leq t} := (z_1, z_2, ..., z_t)$, the goal is to infer the posterior distribution $p(x_t|z_{\leq t})$, over physical time $t = 1, 2, ..., T$.

Many streaming tasks can be viewed as special instances of this formulation. Let s_t, a_t denote the real system state and action at time t. For example, in streaming forecasting, the task is to infer H-step future states $x_t = s_{t+1:t+H}$ with historical states $z_{\leq t}$ where $z_i = s_i$. In planning via imitation learning (Janner et al., 2022), the goal is to infer future expert actions and states $x_t = (s_{t+1:t+H}, a_{t:t+H-1})$ conditioned on historical states and actions $z_{\leq t}$ where $z_i = (s_i, a_{i-1})$. Note that the variable x_t in this formulation typically represents a *long-horizon* future trajectory. Table 1 summarizes these representative examples.

Bayesian filtering. Sequential probabilistic inference involves repeatedly performing inference about future system states as new observations arrive. Bayesian filtering provides a principled framework to recursively update estimations $p(x_t|z_{\leq t})$ over time based on previous estimation $p(x_{t-1}|z_{\leq t-1})$. By Bayesian rule, the posterior belief of current state $p(x_t|z_{\leq t})$ satisfies the key recursive formula:

$$\boldsymbol{p(x_t|z_{\leq t})} \propto p(z_t|x_t, z_{\leq t-1})p(x_t|z_{\leq t-1}), \tag{2}$$

$$p(x_t|z_{\leq t-1}) = \int p(x_t|x_{t-1}, z_{\leq t-1})\boldsymbol{p(x_{t-1}|z_{\leq t-1})}dx_{t-1}.$$

(Note that the process may not be Markovian.) (3)

Equations 2,3 highlight a key structural property of sequential inference: the current belief $p(x_t|z_{\leq t})$ can be obtained by propagating the previous belief $p(x_{t-1}|z_{\leq t-1})$ via the dynamics $p(x_t|x_{t-1}, z_{\leq t-1})$ and a correction using new observation z_t. Although this exact formulation is often intractable due to unknown system dynamics, Bayesian filtering can be interpreted abstractly as an operator:

$$p(x_t|z_{\leq t}) = \text{Bayesian_filtering}(p(x_{t-1}|z_{t-1}); z_{\leq t}), \tag{4}$$

which maps the previous belief to the current posterior.

3.2 Sequential Inference with Flow-based Models: Existing Practices

At each fixed time t, the goal is to sample from the target distribution $p(x_t|z_{\leq t})$. This distribution can be modeled using a conditional flow-based generative model, which incorporates the latest observations $z_{\leq t}$ and transports from a fixed base distribution (e.g., $\mathcal{N}(0, I)$) to the target $p(x_t|z_{\leq t})$. Concretely, we consider the following ODE:

$$\frac{d}{d\tau}x_t(\tau) = v(x_t(\tau), \tau; z_{\leq t}), \quad x_t(0) \sim p(x_t|z_{\leq t}), \quad x_t(1) \sim \mathcal{N}(0, I), \tag{5}$$

where $\tau \in [0, 1]$ denotes the flow matching time and t denotes the physical time step. Ideally, learning a parameterized velocity field $v_\theta(x_t|z_{\leq t}) \approx v(x_t|z_{\leq t})$ allows to sample from $p(x_t|z_{\leq t})$ by solving the ODE.

To deploy such a model in sequential inference, a common practice is to repeatedly apply the conditional prediction model over time t. This receding-horizon practice is widely adopted in

time-series forecasting (Box et al., 2015) and decision-making systems such as model predictive control (Garcia et al., 1989). For flow-based models, this naïve practice requires restarting a fresh sampling process and solving ODE Eq. 5 from scratch to incorporate any new observation (Janner et al., 2022).

Consequently, this pipeline suffers from high inference latency for real-time systems: solving ODE Eq. 5 from the non-informative Gaussian demands tens or hundreds of numerical solver iterations and often fails to match the observation arrival frequency, producing potential delayed predictions. This problem becomes more severe when it comes to long-horizon prediction, as each round of estimation is more expensive to compute. Fundamentally, the restarting-sampling paradigm discards historical estimations and, by initializing each update from an independent Gaussian prior, treats temporally correlated physical processes as independent across time.

3.3 Sequential Flow Matching

From the perspective of Bayesian filtering (Eqs. 2-3), inference over future states should proceed recursively: the belief at time t is obtained by refining the previous belief using new observations. This structure suggests that historical estimations should serve as prior information rather than being discarded. For example, consider a forecasting task where $x_t = s_{t+1:t+H}$. The prediction $x_{t-1} = s_{t:t+H-1}$ at time $t-1$ already provides a coarse estimate of future states and naturally approximates $x_t = s_{t:t+H}$. Ignoring such information results in "redundant" computation and unnecessary sampling overhead.

Motivated by the recursive structure, we propose to directly parameterize and learn the filtering operator in Eq. 4 via a probability flow. We introduce *Sequential Flow Matching* model, which learns a probability flow of predictive distribution from time $t-1$ to time t, conditioned on newly arrived observations $z_{<t} \to z_{\leq t}$. The key step is to replace the non-informative Gaussian distribution with the previous estimation $p(x_{t-1}|z_{\leq t-1})$ as the source distribution in flow ODE:

$$\frac{d}{d\tau}x_t(\tau) = v(x_t(\tau), \tau; z_{\leq t}), \quad x_t(0) \sim p(x_t|z_{\leq t}), \quad \boldsymbol{x_t(1) \sim p(x_{t-1}|z_{\leq t-1})}. \tag{6}$$

The process of solving this ODE can be viewed as performing Bayesian filtering in Eq. 4. Figure 1 C illustrates the inference pipeline of sequential flow matching.

Intuitively, physical process x_t naturally exhibits temporal correlation, and using successive x_{t-1}, x_t as the source and target distribution allows flow matching to exploit this natural temporal coupling, yielding a more regular velocity field and reduced sampling error. This aligns with findings that improved source-target coupling leads to straighter transport paths (Liu et al., 2023). While our method shares conceptual roots with warm-start heuristics in diffusion models (Janner et al., 2022; Scholz & Turner, 2025), it provides a principled Bayesian filtering formulation of warm-starting tailored to streaming environments. Moreover, the use of the previous Bayesian posterior is information-theoretically optimal: The posterior is the distribution that maximally preserves historical information for the recursive update. The following result formally justifies this advantage by showing that temporal correlation induced by the underlying physical process provides a natural and informative coupling to reduce sampling error compared to a non-informative, independent Gaussian source.

Proposition 3.1 (One-step Sampling Error)**.** *Denote the short hand for distribution $p(x_t|z_{\leq t})$ by p. Consider flow matching with straight interpolation $x_t(\tau) = (1-\tau)x_t(0) + \tau x_t(1)$ and two different couplings of $(x_t(0), x_t(1))$:*
***(A) independent Gaussian coupling**. Suppose $x_t(0) \sim p(x_t|z_{\leq t})$ and $x_t(1) \sim \mathcal{N}(0, I)$ are independently sampled. Let $p_{Gaussian}$ be the distribution of $\hat{x}_t(0) = x_t(1) - v(x_t(1), 1; z_{\leq t})$ using one-step sampling by ODE Eq. 5. Then, $\mathcal{W}_2^2(p_{Gaussian}, p) = \mathrm{Var}(x_t|z_{\leq t})$, where $\mathcal{W}_2(p, q)$ is the 2-Wasserstein distance.*
***(B) temporally-correlated coupling**. Suppose $(x_{t-1}, x_t) \sim p(x_{t-1}|z_{\leq t-1})p(x_t|x_{t-1}, z_{\leq t})$, and $x_t(0) = x_t, x_t(1) = x_{t-1}$. Let p_{Bayes} be the distribution of one-step sampling by ODE Eq. 6. Then*

$$\mathcal{W}_2^2(p_{Bayes}, p) \leq \mathbb{E}_{x_{t-1}|z_{\leq t-1}} \mathrm{Var}(x_t|z_{\leq t}, x_{t-1}), \tag{7}$$

which implies $\mathcal{W}_2(p_{Gaussian}, p) - \mathcal{W}_2(p_{Bayes}, p) = \mathrm{Var}_{x_{t-1}|z_{\leq t-1}} \mathbb{E}(x_t|z_{\leq t}, x_{t-1}) \geq 0$ by law of total variance.

The proof is deferred to Appendix A. While Proposition 3.1 focuses on the one-step case, the same advantage extends to multi-step sampling under mild regularity conditions (e.g., Lipschitz continuity

of the velocity field), where the total error can be decomposed into a sum of single-step errors via the triangle inequality, similar to the analysis of Proposition 3.5 in Hu et al. (2024).

The key observation is that the one-step sampling error under temporal coupling reduces to a conditional variance $\mathbb{E}_{x_{t-1}|z_{\leq t-1}}\mathrm{Var}(x_t|z_{\leq t}, x_{t-1})$, rather than the unconditional variance from the independent Gaussian coupling. The gap of sampling error is strictly greater than zero unless $\mathbb{E}(x_t|z_{\leq t}, x_{t-1}) = \mathbb{E}(x_t|z_{\leq t})$, i.e., unless x_{t-1} carries no information about x_t. In particular, if the physical process is a deterministic process $x_t = f(x_{t-1})$ given x_{t-1}, then the conditional variance vanishes and one-step sampling exactly recovers the target distribution of x_t.

A natural question is if we can construct an artificial coupling, e.g., by pairing samples from $\mathcal{N}(0, I)$ and $p(x_t|z_{\leq t})$ using mini-batch optimal transport (Tong et al., 2023), to accelerate the sampling process. However, such a method requires multi-sampling from $p(x_t|z_{\leq t})$ via controlled rollouts with identical history, which is infeasible in real-world dynamic physical systems. In contrast, temporal coupling arises naturally from the sequential structure of the data, where the temporal pairs (x_{t-1}, x_t) are inherently jointly observed given the history $z_{\leq t}$.

3.4 Practical Implementation

In order to learn sequential flow matching (Eq. 6), the key ingredient is to draw samples of x_{t-1}, x_t and construct a training dataset. In this section, we discuss practical considerations for dataset construction to train a parameterized velocity v_θ for sequential flow matching.

A natural approach is to draw ground truth pairs from an offline dataset. In practice, however, naïvely constructing such pairs from ground-truth trajectories leads to a mismatch between training and test time: the source distribution is the ground truth distribution $p(x_{t-1}|z_{\leq t-1})$, but at test time only model predictive distribution $p_\theta(x_{t-1}|z_{\leq t-1})$ is available. This discrepancy becomes crucial in long-horizon settings, where predictions of later physical time are inherently more inaccurate.

Distribution alignment via pretrained model rollouts. To reduce train-test mismatch, we leverage a pretrained flow-based model which better approximates x_{t-1} encountered at test time. Concretely, we first pretrain a flow matching v_{θ_0} to learn $p_{\theta_0}(x_t|z_{\leq t}) \approx p(x_t|z_{\leq t})$ in a standard sampling-from-Gaussian manner (Eq. 5). By letting the neural architecture of v_{θ_0} be the same as v_θ, we can expect $p_{\theta_0}(x_{t-1}|z_{\leq t-1})$ better capture the test-time source distribution $p_\theta(x_{t-1}|z_{\leq t-1})$. In practice, we construct pairs (x_{t-1}, x_t) where x_{t-1} comes from pretrained model and x_t comes from ground truth. This construction aligns the source distribution with the model predictive distribution encountered at test time, while avoiding compounding errors that would arise from fully synthetic trajectories. The constructed dataset is used to finetune θ initialized from θ_0. We found this hybrid construction effectively reduce train-test mismatch and adopt it throughout our experiments.

Re-noising mechanism. While using generated data from pretrained model reduces train-test mismatch, the small deviations in model prediction at test-time can still accumulate and lead to compounding errors over multiple rounds of belief update. To mitigate the error accumulation, we adopt a re-noising mechanism that injects noise into the previous estimation of x_{t-1} at both training and test time, before applying the flow. Concretely, in flow ODE Eq. 6, we replace source distribution $p(x_{t-1}|z_{\leq t-1})$ with a noisy version $\tilde{x}_{t-1} := \alpha(\tau_{\text{renoise}})x_{t-1} + \sigma(\tau_{\text{renoise}}) \cdot \mathcal{N}(0, I)$, where $\alpha(\tau_{\text{renoise}}), \sigma(\tau_{\text{renoise}})$ are the coefficients of the probability flow path at noise level (flow time) τ_{renoise}. We treat τ_{renoise} as a hyperparameter that can be tuned in practice.

The re-noise level τ_{renoise} controls the trade-offs between preserving historical prediction information for sampling acceleration and accommodating uncertainty during belief refinement. If $\tau_{\text{renoise}} = 1$, we have $\tilde{x}_{t-1} \sim \mathcal{N}(0, I)$, which reduces to the restarting-sampling paradigm that requires a large number of sampling steps. If $\tau_{\text{renoise}} = 0$, it relies on a precise modeling of $x_{t-1} \sim p(x_{t-1}|z_{\leq t-1})$, which might induce error accumulation for long-horizon prediction. In Appendix C.4 we study the impact of τ_{renoise}, and we observe that a suitable re-noise level varies with the level of environmental stochasticity.

We present the finetuning algorithm (Algorithm 1) and the inference algorithm (Algorithm 2).

Algorithm 1 Sequential Flow Matching Finetuning

Require: Ground-truth x_t, context $z_{\leq t}$, renoise level τ_{renoise}, and fixed pretrained model v_{θ_0} (from Eq. 5)

1: Generate $x_{t-1} \sim p_{\theta_0}(x_{t-1}|z_{\leq t-1})$ using v_{θ_0}
2: Renoise $\tilde{x}_{t-1} = \alpha(\tau_{\text{renoise}}) \cdot x_{t-1} + \sigma(\tau_{\text{renoise}}) \cdot \mathcal{N}(0, I)$
3: Sample interpolation time $\tau \sim \text{Uniform}(0, 1)$
4: Compute $x_t(\tau) = \alpha(\tau)x_t + \sigma(\tau)\tilde{x}_{t-1}$
5: **return** loss $L(\theta) = \|v_\theta(x_t(\tau), \tau; z_{\leq t}) - \dot{x}_t(\tau)\|^2$

Algorithm 2 Sequential Flow Matching Inference

Require: Pretrained model v_{θ_0} (from Eq. 5), sequential flow matching model v_θ finetuned by Algorithm 1, initial observation z_1

1: initialize $x_1 \sim p_{\theta_0}(x_1|z_1)$
2: **for** $t = 2$ to T **do**
3: Receive observation z_t
4: $\tilde{x}_{t-1} = \alpha(\tau_{\text{renoise}}) \cdot x_{t-1} + \sigma(\tau_{\text{renoise}}) \cdot \mathcal{N}(0, I)$
5: Solve $dx_t(\tau)/d\tau = v_\theta(x_t(\tau), \tau; z_{\leq t}), x_t(1) = \tilde{x}_{t-1}$ and obtain $x_t = x_t(0)$
6: **end for**
7: **return** $x_1, x_2, \ldots, x_T$

4 Related Works

Flow-based models for sequence generation. Diffusion and flow matching models have been widely used for sequence generation in an offline setting. Existing approaches mainly differ in how denoising is scheduled across the prediction horizon. Broadly, prior work can be categorized into three classes: (1) full-sequence denoising: all variables are denoised synchronously to clean sequence (Li et al., 2022; Ho et al., 2022); (2) autoregressive denoising: denoise variables one at a time in temporal order (Hoogeboom et al., 2021; Rasul et al., 2021); (3) asynchronous denoising: allows assigning and denoising variables of different noise levels along the time axis (Ruhe et al., 2024; Wu et al., 2023; Chen et al., 2024). However, these methods are primarily designed for static or offline sequence generation and do not account for streaming data. When applied in streaming settings, they rely on repeated sampling from scratch.

Flow-based models for streaming tasks. Recent work has begun to explicitly study the efficient deployment of flow-based models in streaming settings, primarily in control and decision-making applications. Diffuser (Janner et al., 2022) uses a warm-start heuristic that perturbs a previously generated plan with noise and denoises it using the original diffusion model. While this reduces re-planning cost, denoising perturbed previous predictions does not guarantee that the resulting samples follow the posterior distribution at the next time step. In contrast, our method directly learns the flow that transports the posterior distribution across time steps. Wei et al. (2025); Høeg et al. (2025) independently develop asynchronous denoising to amortize computation in streaming settings. They maintain a partially denoised future action sequence with increasing noise levels along the time axis, and progressively denoise the sequence as new observations arrive. Though with improved efficiency, denoising over the prediction horizon is intentionally left incomplete, leaving long-horizon predictions noisy, and later predictions cannot fully leverage information from these partially denoised variables. Overall, existing methods fall short of the optimal long-horizon estimation offered by principled Bayesian filtering.

Efficient sampling of flow-based models. Prior work accelerates sampling along three main directions. One line of work applies advanced numerical solvers to the flow ODE to reduce the sampling steps (Lu et al., 2022; Zhang & Chen, 2023). Distillation-based methods learn a student model by compressing multi-step sampling trajectories from a teacher model into fewer steps (Salimans & Ho, 2022; Yin et al., 2024; Geng et al., 2023). More recently, flow map matching methods modify the training objective to directly learn the solution operator of the flow ODE and enable few-step generation (Song et al., 2023; Boffi et al., 2025; Geng et al., 2025a). However, the training objective is known to be unstable to optimize (Lu & Song, 2025; Geng et al., 2025b). In our experiment of streaming settings, we observe that training of consistency models (Geng et al., 2025c) and Mean-

Flow (Geng et al., 2025a) is difficult to converge and they generalize poorly. Extending such one-step generative models to streaming tasks is a nontrivial direction, and could potentially be combined with our framework in future work.

Bayesian inference of diffusion models. Bayesian inference with diffusion priors has been widely studied, with early work motivated by inverse problems such as image inpainting (Cardoso et al., 2024; Chung et al., 2023; Dou & Song, 2024). The problem is to sample from the posterior $p(x|y)$ given measurements $y = f(x)$ and a diffusion model $p(x)$. See Daras et al. (2024) for a comprehensive review. A natural question is whether Bayesian filtering update (Eqs. 2-3) can be implemented directly for flow-based models based on these approaches. On one hand, these approaches mainly study static inference, while our work focuses on sequential inference in streaming settings under latency constraints. Extending these approaches to streaming settings remains relatively underexplored. Crucially, most existing approaches rely on likelihood-based guidance, requiring explicit, often linear, observation models and system dynamics that are often unavailable in the real world.

5 Experiment

We extensively evaluate *Sequential Flow Matching* across diverse streaming tasks to assess its effectiveness in streaming settings. Specifically, we study: (1) *Physical system forecasting* (Moro & Chamon, 2025; Rasp et al., 2024): we demonstrate the model's ability to predict long-horizon trajectories over time; (2) *Decision-making* (Fu et al., 2020; Holl et al., 2020): we test model's performance on long-horizon planning and control problems; (3) *State estimation* (He et al., 2025): we evaluate the model's performance in state estimation with different levels of environmental uncertainty. The proposed sequential flow models are obtained by fine-tuning a pretrained model on a small dataset. Details of dataset statistics, model implementation are deferred to Appendix B.

Pretrained Model. We mainly adopt Diffusion Forcing (Chen et al., 2024) and Rectified flow (Liu et al., 2023) as the pretrained model. For Rectified flow, we apply the asynchronous denoising idea of Diffusion Forcing to the flow time scheduling (Appendix B.3). Note that the sequential flow matching is compatible with general flow-based models, though we evaluate it using these two representative models for demonstration.

Baselines. We consider several classes of efficient flow-based baselines: (1) *One-step diffusion*: We evaluate consistency models (Song et al., 2023) and MeanFlow (Geng et al., 2025a). For consistency models, we adopt ECT (Geng et al., 2025c), an efficient fine-tuning method for consistency distillation. (2) *Warm-start diffusion*: Following Diffuser (Janner et al., 2022), we add a controlled amount of noise to the previous prediction x_{t-1} and denoise it under the updated condition z_t using the same pretrained diffusion model. For direct comparison, we set the noise level to τ_{renoise}, matching our re-noise level. This warm-start heuristic can be viewed as an ablation of Sequential Flow Matching, where fine-tuning is removed and the pretrained model is used directly. (3) **Asynchronous denoising diffusion**: We adopt partial denoising and reuse strategies from CL-Diffusion (Wei et al., 2025).

Metrics of inference latency. In the main tables, we report the Number of Function Evaluations (NFE) for measuring sampling efficiency. We uniformly use first-order forward Euler as the ODE solver (equivalent to DDIM (Song et al., 2021a) for diffusion models). Additionally, we study the trade-offs between wall-clock latency and prediction fidelity in Section 5.4. Experiments are conducted on a single NVIDIA RTX 6000 Ada Generation (48 GB memory).

5.1 Physical System Forecasting

Datasets. We consider streaming forecasting on two physical systems: (a) fluid dynamic system governed by the 1D Burgers' equation, adapted from Wei et al. (2025). (b) real-world weather forecasting adapted from WeatherBench2 (Rasp et al., 2024). The model takes historical states $z_{\leq t} := s_{\leq t}$ as conditions and predicts H-step future states $x_t := s_{t+1:t+H}$ over t. The prediction horizon is $H = 10$ for fluid system and $H = 12$ for weather forecasting.

RMSE v.s. Energy Score. We adopt root mean squared error (RMSE) and energy score Gneiting & Raftery (2007) as evaluation metrics, both of which are lower the better. RMSE is a point-wise metric that quantifies prediction error, and the energy score is a distributional metric widely used in probabilistic time-series forecasting to evaluate predictive distributions, defined by $\mathbb{E}_{\hat{x}_t \sim p_\theta} \|\hat{x}_t - x_t\|_1 - \frac{1}{2}\mathbb{E}_{\hat{x}_t, \hat{x}'_t \sim p_\theta} \|\hat{x}_t - \hat{x}'_t\|_1$. Basically, the energy score measure the absolute error

Table 2: Results of physical system forecasting. Prediction horizon $H = 10$ for Burgers'equation and $H = 12$ for weather forecasting.

Method	Burgers' Equation			Weather Forecasting		
	NFE	RMSE ↓	Energy Score ↓	NFE	Temperature RMSE ↓	Temperature Energy Score ↓
Autoregressive	prediction horizon	$0.253_{\pm 0.022}$	$0.133_{\pm 0.009}$	prediction horizon	$3.764_{\pm 0.042}$	$2.583_{\pm 0.026}$
Diffusion Forcing	10	$0.258_{\pm 0.005}$	$\mathbf{0.100}_{\pm 0.001}$	20	$4.249_{\pm 0.162}$	$2.173_{\pm 0.084}$
Rectified Flow	5	$0.262_{\pm 0.001}$	$0.103_{\pm 0.001}$	10	$3.990_{\pm 0.146}$	$\mathbf{2.030}_{\pm 0.080}$
Diffusion Forcing	1	$0.247_{\pm 0.006}$	$0.108_{\pm 0.002}$	1	$4.974_{\pm 0.279}$	$2.849_{\pm 0.197}$
Rectified Flow	1	$0.250_{\pm 0.006}$	$0.114_{\pm 0.002}$	1	$4.509_{\pm 0.228}$	$2.608_{\pm 0.157}$
MeanFlow	1	$0.418_{\pm 0.070}$	$0.234_{\pm 0.047}$	1	$8.632_{\pm 0.030}$	$5.700_{\pm 0.017}$
Consistency Model	1	$0.252_{\pm 0.004}$	$0.107_{\pm 0.003}$	1	$6.269_{\pm 0.067}$	$2.610_{\pm 0.033}$
CL-Diffusion	1	$0.559_{\pm 0.011}$	$0.264_{\pm 0.015}$	1	$6.248_{\pm 0.086}$	$3.368_{\pm 0.124}$
Warm-start Heuristic						
Warm-start Diffusion Forcing	1	$0.487_{\pm 0.033}$	$0.231_{\pm 0.013}$	1	$7.072_{\pm 0.014}$	$4.064_{\pm 0.065}$
Warm-start Rectified Flow	1	$0.475_{\pm 0.004}$	$0.218_{\pm 0.009}$	1	$6.236_{\pm 0.108}$	$3.209_{\pm 0.003}$
Ours						
Sequential Rectified Flow	1	$\mathbf{0.239}_{\pm 0.005}$	$\mathbf{0.101}_{\pm 0.002}$	1	$\mathbf{3.660}_{\pm 0.061}$	$2.353_{\pm 0.032}$

but with a penalty of over-deterministic prediction. It is known that the expected energy score is minimized if and only if the model prediction exactly recovers the ground-truth distribution (Gneiting & Raftery, 2007). Together, RMSE and the energy score offer complementary perspectives on model performance.

Table 2 reports the RMSE and energy scores of different methods. We observe that a deterministic auto-regressive model (using the same backbone as others) attains RMSE comparable to or better than diffusion and flow matching models. However, it fails to capture the underlying predictive uncertainty, as evidenced by substantially worse energy scores. Asynchronous denoising approaches such as CL-Diffusion intentionally maintain partially denoised future trajectories. For evaluation, we fully denoise these trajectories into clean samples using one-step DDIM. Despite this, their performance remains inferior. Warm-start heuristic exhibits similar behavior, as the generated samples are not guaranteed to align with the target predictive distribution. In contrast, sequential flow models attain both strong point-wise accuracy and well-calibrated predictive distributions using one sampling step.

Long-horizon behavior. In appendix C we study the RMSE of different models as a function of the forecast lead time $h = 1, 2, ..., H$. We find the sequential flow model with one sampling step performs competitively to the full-step pretrained model across the entire forecast window.

5.2 Planning and Control

Datasets. We evaluate planning and control on: (a) maze planning from the D4RL benchmark (Fu et al., 2020), where an agent navigates in a maze with obstacles to reach a goal; (b) smoke control (Holl et al., 2020; Wei et al., 2025), which steers smoke toward a target exit in a 2D incompressible fluid governed by the Navier–Stokes equations. For smoke control, we consider fixed maps (identical obstacle layouts at training and test time) and random maps (perturbed obstacle positions at test time). The model conditions on the current state and previous action $z_{\leq t} := (s_t, a_{t-1})$ and predicts H-step future states and actions $x_t := (s_{t+1:t+H}, a_{t:t+H-1})$. The planning horizon is $H = 600$ for Maze-Medium, $H = 800$ for Maze-Large, and $H = 10$ for smoke control. The model re-plans every 50 steps for maze planning and at every step for smoke control.

Except for flow-based models, we also present the results of other classic baselines: MPPI (Williams et al., 2015), CQL (Kumar et al., 2020), IQL (Kostrikov et al., 2022) in maze planning, and BC (Pomerleau, 1988) and BPPO (Zhuang et al., 2023) in smoke control. The performance numbers are directly from Chen et al. (2024) and Wei et al. (2025) under the same benchmark settings.

Diffusion guidance. Maze planning task requires diffusion guidance based on rewards, which relies on modifying local score/velocity function during sampling. The consistency model and MeanFlow baselines lack a well-defined score/velocity function, and therefore we omit these baselines on maze planning. On the other hand, the sequential flow models are finetuned on the imitation learning dataset of trajectories produced by guided pretrained models (see Appendix B). Therefore, the finetuned sequential flow model learns the transportation between guided trajectories, thus removing the need for explicit guidance in our implementation.

Results. Tables 3 and 4 report results on maze planning and smoke control respectively. In maze planning, we observe that sequential flow models require more than one sampling step to achieve strong performance. We hypothesize that this arises from the substantial changes in plans across successive time steps (i.e., $p(x_{t-1}|z_{t-1}), p(x_t|z_{\leq t})$ differs significantly), which makes single-step

Table 3: Results of maze planning. The reported numbers are the accumulated reward over the entire episode. Planning horizon is $H = 600$ for Maze Medium and $H = 800$ for Maze Large.

Method	NFE	Maze Medium ↑	Maze Large ↑
MPPI	N/A	10.2	5.1
CQL	N/A	5.0	12.5
IQL	N/A	34.9	58.6
Diffuser	256	$121.5_{\pm 2.7}$	$123.0_{\pm 6.4}$
Diffusion Forcing	50	$149.4_{\pm 7.5}$	$159.0_{\pm 2.7}$
Diffusion Forcing	3	$110.4_{\pm 11.6}$	$67.3_{\pm 27.1}$
Diffusion Forcing	1	$22.1_{\pm 12.3}$	$2.2_{\pm 1.9}$
CL-Diffusion	1	$74.2_{\pm 2.4}$	$96.1_{\pm 3.6}$
CL-Diffusion	3	$78.3_{\pm 0.7}$	$103.9_{\pm 4.1}$
Warm-start Heuristic			
Warm-start Diffusion Forcing	1	$85.1_{\pm 12.6}$	$61.9_{\pm 22.1}$
Warm-start Diffusion Forcing	3	$81.8_{\pm 8.2}$	$62.3_{\pm 20.7}$
Ours			
Sequential Diffusion Forcing	1	$57.5_{\pm 29.3}$	$114.6_{\pm 49.0}$
Sequential Diffusion Forcing	3	$\mathbf{168.6}_{\pm 11.9}$	$\mathbf{233.9}_{\pm 8.8}$

Table 4: Results of smoke control. The reported numbers are the percentage of smoke that is not leaking via the desired bucket at the end of the entire episode (60 steps). Planning horizon is $H = 10$.

Method	NFE	Fixed Map ↓	Random Map ↓
BC	N/A	0.672	0.705
BPPO	N/A	0.634	0.652
DiffPhyCon	600	0.545	0.375
CL-Diffusion	60	0.337	0.346
Diffusion Forcing	10	$0.166_{\pm 0.036}$	$0.291_{\pm 0.042}$
Rectified Flow	10	$0.173_{\pm 0.078}$	$0.190_{\pm 0.020}$
Diffusion Forcing	1	$0.362_{\pm 0.068}$	$0.280_{\pm 0.019}$
Rectified Flow	1	$0.432_{\pm 0.009}$	$0.260_{\pm 0.020}$
MeanFlow	1	$0.369_{\pm 0.117}$	$0.462_{\pm 0.166}$
Consistency Model	1	$0.295_{\pm 0.013}$	$0.293_{\pm 0.002}$
Warm-start Heuristic			
Warm-start Diffusion Forcing	1	$0.547_{\pm 0.026}$	$0.268_{\pm 0.016}$
warm-start Rectified Flow	1	$0.291_{\pm 0.039}$	$0.286_{\pm 0.017}$
Ours			
Sequential Diffusion Forcing	1	$\mathbf{0.097}_{\pm 0.012}$	$\mathbf{0.231}_{\pm 0.006}$

sampling challenging. Nevertheless, our method still consistently achieves highly competitive performance compared to full-step diffusion on both tasks with a small number of sampling steps.

5.3 State Estimation

We additionally evaluate Sequential Flow Matching on state estimation of the Lorenz system, a simplified mathematical model used to capture chaotic behavior and to understand atmospheric convection (Lorenz, 2017). We find Sequential Flow Matching achieves competitive performance across different system stochasticity levels with one single sampling step. Detailed results are provided in Appendix C.

5.4 Performance and Latency Tradeoffs

We study the trade-off between model performance and real-time inference latency, defined as the wall-clock time required to produce a prediction for a single instance upon receiving a new observation. Figure 2 illustrates the latency–performance trade-off on maze planning by varying the number of sampling steps. Pretrained diffusion models typically require at least ten steps (latency > 100 ms) to achieve strong performance, whereas the fine-tuned sequential diffusion model performs comparably or better within two to three steps (latency ~ 30 ms). Similar trends are observed on other datasets (Appendix C).

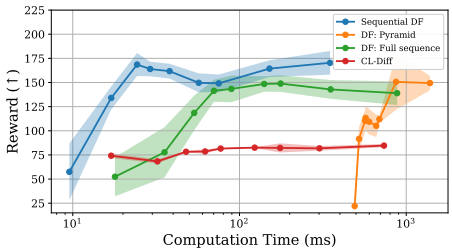

Figure 2: Inference latency v.s. performance for maze planning under varying sampling steps (1, 2, 3, 4, 5, 8, 10, 20 and 50 steps). DF: Pyramid and DF: Full-sequence refer to two denoising schedules of diffusion forcing. Sequential DF uses full sequence denoising.

5.5 Ablation Study

The warm-start heuristic can be viewed as an ablation of Sequential Flow Matching, where fine-tuning stage is removed and the pretrained model is used directly. We also conduct an ablation study on the renoising mechanism and the choice of ground-truth versus model-rollout finetuning (Appendix C.4).

6 Conclusion and Limitations

This work studies the efficient deployment of flow-based models in streaming settings, where models are required to perform long-horizon prediction and continuously assimilate streaming observations under latency constraints. We propose Sequential Flow Matching, which transports the predictive distribution across time steps and can be viewed as learning the Bayesian belief update. Experiments show that it achieves performance comparable to full-step diffusion using only a few sampling steps. This suggests a principled perspective for real-time deployment of flow-based models through Bayesian filtering. A limitation of our work is that the current implementation relies on a pretraining-generation-finetuning pipeline. Exploring simulation-free training from scratch is a promising future direction.

ACKNOWLEDGMENTS

We would like to thank Prof. Tailin Wu, as well as Ruiqi Feng, Long Wei and Haodong Feng for their insightful discussions and feedback on earlier versions of this work.

This work is primarily supported by NSF awards PHY-2117997, IIS-2239565, and IIS-2428777, as well as Meta Research Award and Nvidia Academic Award. Prof. Bo Dai would like to acknowledge support from NSF ECCS-2401391, NSF IIS-2403240, and ONR N000142512173.

REFERENCES

Thomas R Bewley. Flow control: new challenges for a new renaissance. *Progress in Aerospace sciences*, 37(1):21–58, 2001.

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Gabriel Cardoso, Yazid Janati el idrissi, Sylvain Le Corff, and Eric Moulines. Monte carlo guided denoising diffusion models for bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nHESwXvxWK.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Giannis Daras, Hyungjin Chung, Chieh-Hsin Lai, Yuki Mitsufuji, Jong Chul Ye, Peyman Milanfar, Alexandros G Dimakis, and Mauricio Delbracio. A survey on diffusion models for inverse problems. *CoRR*, 2024.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Pierre Del Moral. Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 325(6):653–658, 1997.

Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *The Twelfth International Conference on Learning Representations*, 2024.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Zhihan Gao, Xingjian Shi, Boran Han, Hao Wang, Xiaoyong Jin, Danielle Maddix, Yi Zhu, Mu Li, and Yuyang Bernie Wang. Prediff: Precipitation nowcasting with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:78621–78656, 2023.

Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

Zhengyang Geng, Ashwini Pokle, and J Zico Kolter. One-step diffusion distillation via deep equilibrium models. *Advances in Neural Information Processing Systems*, 36:41914–41931, 2023.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025a.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025b.

Zhengyang Geng, Ashwini Pokle, Weijian Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. In *The Thirteenth International Conference on Learning Representations*, 2025c. URL https://openreview.net/forum?id=xQVxo9dSID.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Yangguang He, Wenhao Li, Minzhe Li, Juan Zhang, Xiangfeng Wang, and Bo Jin. Trackdiffuser: Nearly model-free bayesian filtering with diffusion model. *arXiv preprint arXiv:2502.05629*, 2025.

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

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

Sigmund H Høeg, Yilun Du, and Olav Egeland. Fast policy synthesis with variable noise diffusion models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4821–4828. IEEE, 2025.

Philipp Holl, Nils Thuerey, and Vladlen Koltun. Learning to control pdes with differentiable physics. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HyeSin4FPB.

Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.

Xixi Hu, Qiang Liu, Xingchao Liu, and Bo Liu. Adaflow: Imitation learning with variance-adaptive flow-based policies. *Advances in Neural Information Processing Systems*, 37:138836–138858, 2024.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pp. 9902–9915. PMLR, 2022.

Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pp. 182–193. Spie, 1997.

Eugenia Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press, 2003.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

H Kushner. Approximations to optimal nonlinear filters. *IEEE Transactions on Automatic Control*, 12(5):546–556, 1967.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343, 2022.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Edward N Lorenz. Deterministic nonperiodic flow 1. In *Universality in Chaos, 2nd edition*, pp. 367–378. Routledge, 2017.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LyJi5ugyJx.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Viggo Moro and Luiz F. O. Chamon. Solving differential equations with constrained learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5KqveQdXiZ.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*, pp. 8857–8868. PMLR, 2021.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42818–42835. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ruhe24a.html.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Jonas Scholz and Richard E Turner. Warm starts accelerate conditional diffusion. *arXiv preprint arXiv:2507.09212*, 2025.

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 32211–32252, 2023.

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian FATRAS, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023. URL https://openreview.net/forum?id=HgDwiZrpVq.

Long Wei, Peiyan Hu, Ruiqi Feng, Haodong Feng, Yixuan Du, Tao Zhang, Rui Wang, Yue Wang, Zhi-Ming Ma, and Tailin Wu. Diffphycon: A generative approach to control complex physical systems. *Advances in Neural Information Processing Systems*, 37:4090–4147, 2024.

Long Wei, Haodong Feng, Yuchen Yang, Ruiqi Feng, Peiyan Hu, Xiang Zheng, Tao Zhang, Dixia Fan, and Tailin Wu. CL-diffphycon: Closed-loop diffusion control of complex physical systems. In *The Thirteenth International Conference on Learning Representations*, 2025.

Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling. *arXiv preprint arXiv:1509.01149*, 2015.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, Jian Jiao, Juntao Li, Jian Guo, Nan Duan, Weizhu Chen, et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *Advances in Neural Information Processing Systems*, 36:39957–39974, 2023.

Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, et al. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14662–14672, 2024.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Loek7hfb46P.

Guangyao Zhou, Sivaramakrishnan Swaminathan, Rajkumar Vasudeva Raju, J Swaroop Guntupalli, Wolfgang Lehrach, Joseph Ortiz, Antoine Dedieu, Miguel Lazaro-Gredilla, and Kevin Patrick Murphy. Diffusion model predictive control. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=pvtgffHtJm.

Zifeng Zhuang, Kun LEI, Jinxin Liu, Donglin Wang, and Yilang Guo. Behavior proximal policy optimization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3c13LptpIph.

A Deferred Proof and Derivation

Lemma A.1 (One-step Sampling Distribution). *Let π be a coupling of p_1, p_0 and $x(1) \sim p_1, x(0) \sim p_0$. Define a straight interpolation path $x(\tau) = (1-\tau)x(0) + \tau x(1)$. Consider the flow ODE*

$$\frac{d}{d\tau}x(\tau) = v(x(\tau), \tau), \quad x(0) \sim p_0, x(1) \sim p_1, \tag{8}$$

where $v(x(\tau), \tau) = \mathbb{E}[x(1) - x(0)|x(\tau)]$. Then the marginal distribution $\hat{p}_0$ of one-step sampling $\hat{x}(0) := x(1) - v(x(1), 1)$ is

$$\hat{p}_0 = \int \delta_{\mathbb{E}[x(0)|x(1)]} dp_1(x(1)), \tag{9}$$

where δ_a is a Dirac delta measure at a, and

$$\mathcal{W}_2^2(p_0, \hat{p}_0) \leq \mathbb{E}_{x(1)\sim p_1}\mathbb{E}_{x(0)|x(1)}[x(0) - \mathbb{E}[x(0)|x(1)]] = \mathbb{E}_{x(1)\sim p_1} \operatorname{Var}(x(0)|x(1)). \tag{10}$$

In particular, if $x(0), x(1)$ are independent, then the upper bound is tight and becomes

$$\mathcal{W}_2^2(p_0, \hat{p}_0) = \operatorname{Var}(x(0)). \tag{11}$$

Proof. The velocity field at $t = 1$ is

$$v(x(1), 1) = \mathbb{E}[x(1) - x(0)|x(1)] = x(1) - \mathbb{E}[x(0)|x(1)]. \tag{12}$$

Therefore, the one-step sampling yields $\hat{x}(0) = x(1) - v(x(1), 1) = \mathbb{E}[x(0)|x(1)]$ given initial condition $x(1)$. The marginal distribution of $\hat{x}(0)$ is therefore

$$\hat{p}_0 = \int \delta_{\mathbb{E}[x(0)|x(1)]} dp_1(x(1)). \tag{13}$$

The 2-Wasserstein distance $\mathcal{W}_2^2(p_0, \hat{p}_0)$, by definition, is

$$\mathcal{W}_2^2(p_0, \hat{p}_0) = \inf_{\gamma \in \Pi(p_0, \hat{p}_0)} \mathbb{E}_{(a,b)\sim\gamma}\|a - b\|^2, \tag{14}$$

where $\gamma \in \Gamma(p_0, \hat{p}_0)$ is a coupling between $p_0, \hat{p}_0$. Let us take $a := x(0) \sim p_0$ and $b := \hat{x}(0) \sim \hat{p}_0$, which is a valid coupling, then

$$\mathcal{W}_2^2(p_0, \hat{p}_0) \leq \mathbb{E}_{x(0),\hat{x}(0)}\|x(0) - \hat{x}(0)\|^2 = \mathbb{E}_{x(0),x(1)}\|x(0) - \mathbb{E}[x(0)|x(1)]\| = \mathbb{E}_{x(1)\sim p_1} \operatorname{Var}(x(0)|x(1)). \tag{15}$$

Now consider a special case that $x(0), x(1)$ are independent, i.e., $\pi(x(0), x(1)) = p_0(x(0))p_1(x(1))$. The one-step sampling $\hat{x}(0)$ becomes $\mathbb{E}x(0)$, and $\hat{p}_0 = \delta_{\mathbb{E}x(0)}$, a Dirac delta measure. Since there is only one coupling between a Dirac delta measure and any other distribution, we have

$$\mathcal{W}_2^2(p_0, \hat{p}_0) = \mathcal{W}_2^2(p_0, \delta_{\mathbb{E}x(0)}) = \mathbb{E}_{x(0)}\|x(0) - \mathbb{E}x(0)\|^2 = \operatorname{Var}(x(0)), \tag{16}$$

which concludes the proof. □

A.1 Proof of Proposition 3.1

Proposition 3.1 (restated). *Denote the short hand for distribution $p(x_t|z_{\leq t})$ by p. Consider flow matching with straight interpolation $x_t(\tau) = (1-\tau)x_t(0) + \tau x_t(1)$ and two different couplings of $(x_t(0), x_t(1))$:*
***(A) independent Gaussian coupling**. Suppose $x_t(0) \sim p(x_t|z_{\leq t})$ and $x_t(1) \sim \mathcal{N}(0, I)$ are independently sampled. Let $p_{Gaussian}$ be the distribution of $\hat{x}_t(0) = x_t(1) - v(x_t(1), 1; z_{\leq t})$ using one-step sampling by ODE Eq. 5. Then, $\mathcal{W}_2^2(p_{Gaussian}, p) = \operatorname{Var}(x_t|z_{\leq t})$, where $\mathcal{W}_2(p, q)$ is the 2-Wasserstein distance.*
***(B) temporally-correlated coupling**. Suppose $(x_{t-1}, x_t) \sim p(x_{t-1}|z_{\leq t-1})p(x_t|x_{t-1}, z_{\leq t})$, and $x_t(0) = x_t, x_t(1) = x_{t-1}$. Let p_{Bayes} be the distribution of one-step sampling by ODE Eq. 6. Then*

$$\mathcal{W}_2^2(p_{Bayes}, p) \leq \mathbb{E}_{x_{t-1}|z_{\leq t-1}} \operatorname{Var}(x_t|z_{\leq t}, x_{t-1}), \tag{17}$$

which implies $\mathcal{W}_2(p_{Gaussian}, p) - \mathcal{W}_2(p_{Bayes}, p) = \operatorname{Var}_{x_{t-1}|z_{\leq t-1}} \mathbb{E}(x_t|z_{\leq t}, x_{t-1}) \geq 0$ by law of total variance.

Proof. We directly apply Lemma A.1, where p_0 is replaced by $p(x_t|z_{\leq t})$ and p_1 is replaced by $p(x_{t-1}|z_{\leq t-1})$. □

B EXPERIMENTAL DETAILS

B.1 DATASET

Table 5: Dataset statistics. The trajectory length refers to the total length of prediction for a specific task, and the prediction horizon is the length of rolling prediction window per physical time. The prediction horizon = "shrinking" means the task requires a shrinking prediction window whose length equals to the trajectory length at the beginning but decreases over time. #pretraining and #finetuning refer to the number of pretraining and finetuning trajectories.

Task	Dataset	Trajectory length	Prediction horizon	Feature dim.	#pretraining	#finetuning
Forecasting	Burger's Equation	16	10	64	90,000	10,000
	WeatherBench2	28	12	32×32×16	73,112	5,000
State Estimation	Lorentz Attractor	100	1	3	75,000	5,000
Planning&Control	Maze Medium	600	600 (Shrinking)	2	1,999,400	16,384
	Maze Large	800	800 (Shrinking)	2	3,999,200	32,768
	Smoke Control	65	10	64×64× 6	36,000	2,000

We present the dataset statistics in Table 5. The trajectory length refers to the total prediction duration (or episode) and prediction horizon is the length of prediction window at each time step. #pretrain and #finetuning refer to the number of trajectories we use for model pretraining and finetuning. We provide details data construction below.

Burgers' equation. We adopt the dataset from Wei et al. (2025). The 1D Burgers' equation follows:

$$\begin{cases} \dfrac{\partial s}{\partial t} = -s \cdot \dfrac{\partial s}{\partial x} + \nu \dfrac{\partial^2 s}{\partial x^2} + a(x,t), & \text{in } [0,T] \times \Omega, \\ s(x,t) = 0, & \text{in } [0,T] \times \partial\Omega, \\ s(x,0) = s_0(x), & \text{in } \{\tau = 0\} \times \Omega. \end{cases} \tag{18}$$

Here state states $s(x,t)$ is a field over space x and time t, and input (action) $a(x,t)$ is also a function of space and time. This system dynamic is a deterministic process. To impose uncertainty, we introduce: (1) partial observation: only half of the space is observable, i.e., $s(x,t)$ for $x \in \Omega_{\text{right}}$ is removed from model input; (2) agnostic input: the input $a(x,t)$ (which is randomly generated in training and test dataset) is also excluded from model input.

Weather forecasting. We adopt the WeatherBench2 (Rasp et al., 2024). The dataset consists of global weather data from year 1959 to 2023, measured with 6 hours as interval. The data is a 2D/3D temporal data, including 2D features (surface variables) such as `sea_level_pressure`, `2m_temperature`, and 3D features such `geopotential` as a function as height (measured by Atmospheric pressure). Due to computational resources, we only choose partial features (2D:`2m_temperature`, `10m_u_component_of_wind`, `10m_v_component_of_wind`, `mean_sea_level_pressure`, 3D: `geopotential`,`temperature`,`u_component_of_wind`,`v_component_of_wind`, at height $500, 850, 1000$ Pa.) as our system states. We also constraint the latitude from 37.25 to 45.0 and longitude from 115.0 to 122.75. We use the data from 1959 to 2011 for pretraining, 2011 to 2015 for finetning and 2021 to 2023 for testing. We report `2m_temperature` in our main table 2.

Maze planning. We adopt maze-medium and maze-large from D4RL benchmark (Fu et al., 2020). The offline dataset consists of random-walk trajectories and at test time the goal is to reach a target position.

Smoke Control. We adopt the dataset from Wei et al. (2025). The system state is a 2D incompressible fluid following the Navier-Stokes equations:

$$\begin{cases} \dfrac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} - \nu \nabla^2 \mathbf{v} + \nabla p = \mathbf{f}, \\ \nabla \cdot \mathbf{v} = 0, \\ \mathbf{v}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}). \end{cases} \tag{19}$$

Here the system state s consists of velocity field $\mathbf{v}$ and pressure field p, and external force field $\mathbf{f}$ is the action. The task is to generate $\mathbf{f}$ to guide an initial smoko in the field to avoid obstacles and reach

a target exit area. There are two settings in (Wei et al., 2025): large domain control and boundary control. We adopt large domain control setting, where force signals are applied to all peripheral regions outside the obstacles, consisting of 1,792 cells.

State Estimation. We adopt the Lorenz attractor simulator from (He et al., 2025) to generate data. The system state s_t is a 3D vector following an nonlinear state-space model:

$$s_t = \mathbf{F}(s_{t-1})s_{t-1} + w_t, \tag{20}$$

$$\mathbf{F}(s_{t-1}) = \exp\left(\begin{bmatrix} -10 & 10 & 0 \\ 28 & -1 & -s_{t-1,1} \\ 0 & s_{t-1,1} & -\frac{8}{3} \end{bmatrix} \Delta\right), \tag{21}$$

where w_t is a noise term (we use Gaussian noise $w_t \sim \mathcal{N}(0, q^2 I)$) and Δ is the time interval. The observation is $z_t = g(x_t) + \eta_t$, where $g(x_t)$ is a rotation operation and $\eta_t \sim \mathcal{N}(0, r^2 I)$. The task is to estimation current state system s_t given historical observation $z_{\leq t}$ (s_t cannot be directly observed).

B.2 Implementation Details of Algorithms 1,2

Forecasting. In forecasting task we have $x_t = s_{t+1:t+H}$ and $z_t = x_t$, where s_t is the real physical states we want to predict. To apply Algorithm 1, we take a small finetuning dataset $\{s^{(i)}_{1:T}, z^{(i)}_{1:T}\}_{i=1,2,\ldots,n}$ and leverage a pretrained Rectified Flow model v_{θ_0} to generate some trajectories $\hat{s}^{(i)}_{t+1:t+H} \sim p_{\theta_0}(s_{t+1:t+H}|z^{(i)}_{\leq t})$. We could directly use $\hat{s}^{(i)}_{t:t+H-1}$ as the source distribution and $s^{(i)}_{t+1:t+H}$ as the target distribution in sequential flow ODE Eq. 6. In practice, to better align the prediction at the same physical time, we instead drop the first $\hat{s}^{(i)}_t$ (as it is already observed at time t) and pad a $\hat{s}^{(i)}_{t+H-1}$ (can be seen as a moving average) and construct $(\hat{s}^{(i)}_{t+1:t+H-1}, \hat{s}^{(i)}_{t+H-1})$ as the source distribution instead.

Planning and Control. In planning and control task we have $x_t = (s_{t+1:t+H}, a_{t:t+H-1})$ and $z_t = (s_t, a_{t-1})$. To apply Algorithm 1, we require an imitation learning dataset consisting of expert state-action trajectories. We treat the pretrained flow model p_{θ_0} as an expert policy and let it interact with the environment to collect a few state-action trajectories. Ultimately, we have model predicted trajectory $\hat{s}^{(i)}_{t+1:t+H}, \hat{a}^{(i)}_{t:t+H-1} \sim p_{\theta_0}(s_{t+1:t+H}, a_{t:t+H-1}|s^{(i)}_{1:t}, \hat{a}^{(i)}_{1:t-1})$ at any time t and a resulting actual full trajectory $(s^{(i)}_{1:T}, a^{(i)}_{1:T})$, where $a^{(i)}_t$ is the actual executed action (i.e., the first predicted action at each time). We treat $(\hat{s}^{(i)}_{t:t+H-1}, \hat{a}^{(i)}_{t-1:t+H-2})$ as the source distribution and $(s^{(i)}_{t+1:t+H}, a^{(i)}_{t:t+H-1})$ as the target distribution in sequential flow matching. Again, since at time t we observe $z^{(i)}_t = (s^{(i)}_t, a^{(i)}_{t-1})$, in practice we remove already-observed state-action $\hat{s}^{(i)}_t, \hat{a}^{(i)}_{t-1}$ and instead adopt $((\hat{s}^{(i)}_{t+1:t+H-1}, \hat{s}^{(i)}_{t+H-1}), (\hat{a}^{(i)}_{t:t+H-2}, \hat{a}^{(i)}_{t+H-2})$ as the source distribution.

State Estimation. In state estimation task we have $x_t = s_t$. Similarly we call a pretrained model to generate $\hat{s}^{(i)}_t \sim p_{\theta_0}(s_t|z^{(i)}_{\leq t})$. We use $\hat{s}^{(i)}_t$ as the source distribution and the actual physical state $s^{(i)}_t$ as the target distribution.

B.3 Implementation Details of Baselines

Pretraining. Diffusion forcing (Chen et al., 2024) proposes an asynchronous-noising training of diffusion models, which shows promising results for offline sequence generation. The key idea is to add independently random noise levels to different tokens in a sequence, and the score function takes this sequence of noise levels as its input for denoising. Adopting this idea, we train Rectified flow in a similar approach: we independently take random interpolation time for different tokens, and let the velocity function be aware of the interpolation time for each token.

Warm-start Diffusion. The warm-start diffusion is a heuristic that denoise a previously noisy x_{t-1}, but conditioning on new observation z_t, to obtain updated prediction of x_t. In our implementation, warm-start diffusion share the exact pipeline as our sequential flow models, the only difference is that it use the pretrained model v_{θ_0} for denoising, while sequential flow matching uses a finetuned flow v_θ for generation.

CL-Diffusion (Wei et al., 2025). The original CL-Diffusion requires to train two diffusion models, a synchronous diffusion and an asynchronous diffusion that allows different denoising scheduling. In our implementation, since Diffusion Forcing (Chen et al., 2024) pretrained model allows to denoise in an arbitrary schedule, we use a single pretrained diffusion forcing to replace the two models in CL-Diffusion.

C Additional Experimental Results

C.1 State Estimation

Datasets. We consider state estimation of Lorenz attractor, a three-dimensional chaotic dynamical system following the nonlinear state-space model: $s_t = f(s_t)s_t + \mathcal{N}(0, q^2 I)$ and $z_t = g(s_t) + \mathcal{N}(0, r^2 I)$, where $s_t \in \mathbb{R}^3, f(s_t) \in \mathbb{R}^{3\times 3}, g(s_t) \in \mathbb{R}^3$. It is a simplified mathematical model used to capture chaotic behavior and to understand atmospheric convection (Lorenz, 2017). We follow He et al. (2025) to set g be a rotation matrix operation, $q = r/10$ and varies r to test the model performance under different environmental stochasticity. The model takes historical observations $z_{\leq t}$ as conditions and predicts current state $x_t := s_t$. The prediction horizon is $H = 1$.

Except for learning-based approaches, we also compare against model-based approaches, including Extended Kalman Filtering (Kushner, 1967), Unscented Kalman Filtering (Julier & Uhlmann, 1997), and Particle Filtering (Del Moral, 1997). These methods have explicit access to the underlying system dynamics $f(x_t)$ and measurement function $g(x_t)$, and therefore serve as oracle-style baselines.

Results. Table 6 reports performance measured by $10\log_{10}(\text{MSE})$ under varying levels of environmental uncertainty (larger $1/r^2[\textbf{dB}]$ indicates lower stochasticity). As expected, the performance of all methods degrades as environmental stochasticity increases. Notably, warm-start approaches perform reasonably well when environmental stochasticity is low, but their performance becomes worse as uncertainty increases. This behavior is expected, since warm-start methods could approximate the correct predictive distribution when successive states x_{t-1}, x_t follow similar distributions, which is more likely to hold in near-deterministic system dynamics. In contrast, *Sequential Flow Matching* explicitly learns the probability flow across successive time steps. It consistently achieves competitive performance across all uncertainty levels with one sampling step among the learning-based methods.

Table 6: Results of state estimation. The reported numbers are the $10\log_{10}(\text{MSE})$ (lower the better) with MSE averaged over the entire episode (100 steps). $1/r^2[\textbf{dB}] = 10\log_{10}(1/r^2)$ represents different uncertainty levels of measurements (smaller the $1/r^2[\textbf{dB}]$ larger the stochasticity).

Method	**NFE**	Environment Stochasticity $1/r^2$**[dB]**			
		-10	0	10	20
Model-based					
Extended KF	N/A	2.69	-6.19	-16.49	-25.18
Unscented KF	N/A	9.05	2.58	-3.52	-16.24
Particle Filtering	N/A	3.76	-4.76	-14.68	-22.93
Diffusion Forcing	10	$11.74_{\pm 0.01}$	$0.68_{\pm 0.02}$	$-9.57_{\pm 0.07}$	$-17.08_{\pm 1.56}$
Rectified Flow	5	$11.93_{\pm 0.02}$	$0.57_{\pm 0.02}$	$-9.30_{\pm 0.30}$	$-18.65_{\pm 0.33}$
Diffusion Forcing	1	$10.59_{\pm 0.07}$	$1.15_{\pm 0.17}$	$-8.74_{\pm 0.30}$	$-15.01_{\pm 2.07}$
Rectified Flow	1	$15.39_{\pm 0.10}$	$4.90_{\pm 1.04}$	$-5.94_{\pm 0.26}$	$-13.25_{\pm 1.41}$
MeanFlow	1	$12.87_{\pm 0.62}$	$6.20_{\pm 0.44}$	$2.82_{\pm 2.00}$	$0.85_{\pm 1.70}$
Consistency Model	1	$10.85_{\pm 0.00}$	$4.15_{\pm 0.16}$	$2.78_{\pm 0.15}$	$2.45_{\pm 0.04}$
Warm-start Heuristic					
Warm-start Diffusion Forcing	1	$13.52_{\pm 0.03}$	$1.39_{\pm 0.17}$	$-9.42_{\pm 0.04}$	$-17.11_{\pm 1.51}$
Warm-start Rectified Flow	1	$16.86_{\pm 0.11}$	$5.07_{\pm 0.40}$	$-7.66_{\pm 0.42}$	$-18.45_{\pm 0.48}$
Ours					
Sequential Rectified Flow	1	$10.12_{\pm 0.01}$	$0.26_{\pm 0.20}$	$-9.79_{\pm 0.06}$	$-19.58_{\pm 0.09}$

C.2 Long-horizon Behavior

Figure 3 compares the RMSE as a function of the forecast lead time $h = 1, 2, ..., H$. While the auto-regressive model performs better at short horizons, its error increases more rapidly with horizon than flow-based models. In contrast, the sequential flow model with one sampling step performs competitively to the full-step pretrained model across the entire forecast window.

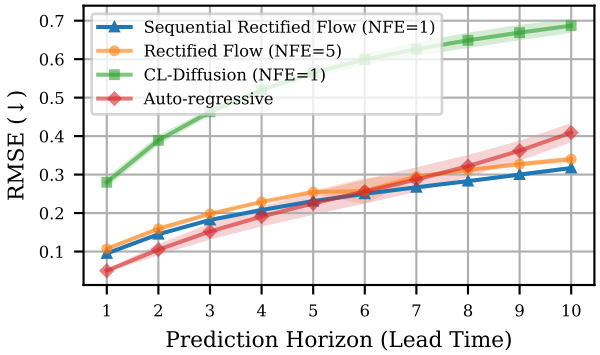

Figure 3: RMSE of Burgers' equation as a function of forecast lead time.

C.3 Performance-Latency Trade-offs

We further show the performance-latency trade-off on smoke control in Figure 4. We find sequential flow models only require one step to achieve saturate performance.

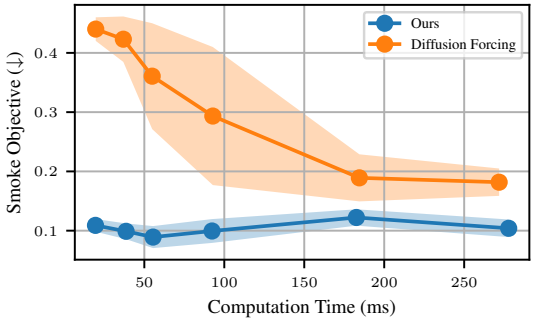

Figure 4: Inference latency for smoke control under varying sampling timesteps (1, 2, 3, 5, 10, and 15 steps). Wall-clock time is computed per instance and averaged across both physical time steps and the test set.

C.4 Ablation Study

We conduct ablation study on re-noise mechanisms and training with model-generated trajectories.

Re-noise level. Figure 5 reports performance under different re-noise levels τ_{renoise}. We observe that both using a fully clean previous estimate ($\tau_{\text{renoise}} = 0$) and completely discarding the previous estimate ($\tau_{\text{renoise}} = 1$) lead to degraded performance. In contrast, there exists an intermediate range of τ_{renoise} that yields consistently strong and robust results. Moreover, this optimal range shifts toward larger values as system uncertainty increases (e.g., the optimal renoise level is $0.4 \sim 0.6$ for high system uncertainty $1/r^2$[dB]=-10, and is $0.2 \sim 0.4$ for low system uncertainty $1/r^2$[dB]=10). This behavior is expected: higher system uncertainty induces greater uncertainty in the model predictions, which in turn requires a higher re-noise level to adequately accommodate this stochasticity.

Training with model generated trajectories. Table 7 compares the performance of Burgers' equation forecasting and state estimation ($1/r^2$[dB]=-10) of using model-generated trajectories ($(\hat{x}_{t-1}, x_t)$) against pure ground-truth trajectories (x_{t-1}, x_t) in sequential flow matching finetuning. We see a significant performance degrade when exclusively using ground-truth trajectories on forecasting task, while comparable performance on state tracking. We hypothesis for long-horizon task like Burgers' equation forecasting, the model's prediction will be largely deviated from ground truth, while for short-horizon task like state estimation ($H = 1$), the model prediction is close to

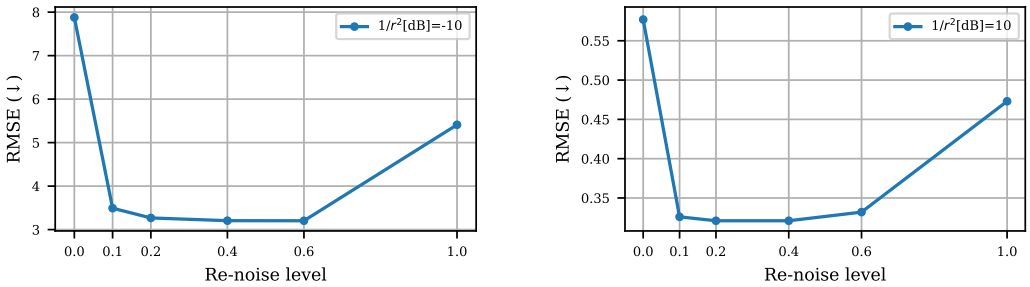

Figure 5: Performance as a function of re-noise level τ_{renoise} on state estimation. Left and right figures are at different system uncertainty levels.

ground truth. The different levels of train-test mismatch will decide if model-generated trajectories are necessary for finetuning sequential flow models.

Table 7: Ablation study of the choice of finetuning on model-generated trajectories or ground-truth trajectories.

Method	NFE	**Burgers' Equation** RMSE ↓	Energy Score ↓	**State Estimation** $10\log_{10}$ MSE ↓
Sequential Flow Matching (finetuned from model-generated trajectories)	1	0.239	0.101	10.12
Sequential Flow Matching (finetuned from ground-truth trajectories)	1	0.251	0.108	10.11

D VISUALIZATION

We provide a visualization of maze planning (Figure 7) and smoke control (Figure 6) using pretrained diffusion model and finetuned sequential diffusion model.

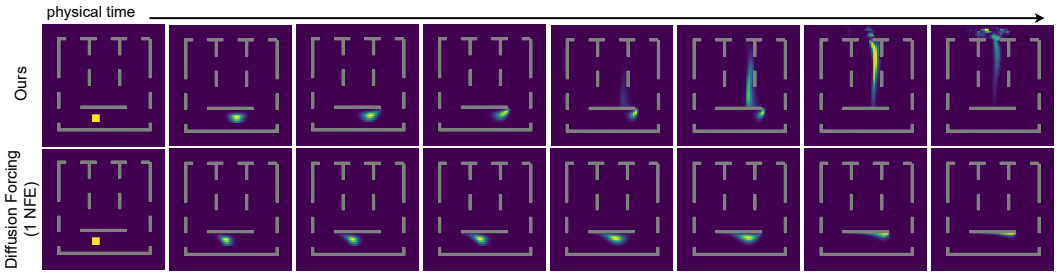

Figure 6: **Smoke Control.** With the same NFE, Sequential Diffusion Forcing effectively controls the smoke to reach the target exit, whereas Diffusion Forcing fails to circumvent the bottom obstacle.

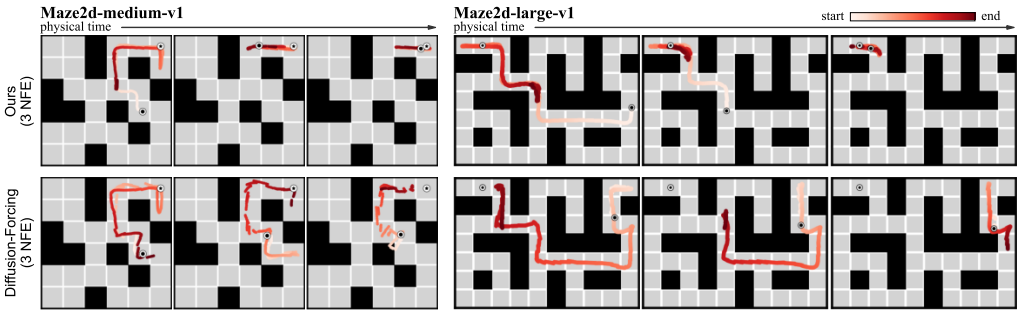

Figure 7: **Maze Planning.** With the same NFE, Sequential Diffusion Forcing can utilize the previous plans to effectively reach the target, while Diffusion Forcing fails drastically with small NFE.