A Mean-Field Framework for Inference-Time Distributional Control of Diffusion Models

Samuel Howard
Department of Statistics
University of Oxford

Nikolas Nüsken
Department of Mathematics
King's College London

Abstract

Diffusion models are increasingly used as controllable samplers, whose generations can be steered at inference time according to a chosen reward function. While such rewards are typically defined on individual samples, for many applications it is desirable to steer according to distribution-level rewards, for example to calibrate with population-level information or to encourage diversity. In both cases, simply incorporating the reward gradient into the dynamics, while often effective, comes with few theoretical guarantees on the sampled distribution. For pointwise rewards, recent work has therefore sought to develop a principled framework for targeting a prescribed tilted distribution using particle reweighting. However, an analogous theoretically-grounded approach for distributional rewards is currently lacking. In this work, we formulate inference-time distributional control as targeting a tilted measure under a mean-field framework, and derive a weighted interacting particle scheme to target it in a principled manner. Our framework recovers pointwise-reward steering as a special case, while providing a theoretical foundation for existing batch-level steering methods. Empirically, we verify that the procedure correctly targets the prescribed distribution in tractable low-dimensional settings, and investigate its behaviour in higher-dimensional protein conformation tasks.

1 Introduction

Diffusion and Flow Models Diffusion and flow-based generative models (Song et al., 2021; Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023) generate novel samples from a data distribution by learning time-dependent dynamics that transport an initial Gaussian distribution p_0 to the target p_1. Such models have achieved state-of-the-art performance across a wide range of domains, including images (Rombach et al., 2022), molecules (Watson et al., 2023), and videos (Ho et al., 2022).

Inference-Time Steering In many applications, the goal is not only to sample from the base distribution p_1, but to instead obtain samples that exhibit certain desired properties. To this end, one can 'steer' the generative process by including additional drift terms in the inference dynamics. Such steering mechanisms can be used to direct a pretrained generative model towards higher-quality samples, or samples that fulfil certain task-specific properties, without requiring further training. Most existing inference-time steering methods aim to improve samples with respect to a *pointwise* reward function $r : \mathbb{R}^d \to \mathbb{R}$ defined on individual generated samples. Early steering approaches generally incorporated the reward gradient $\nabla r(X_t)$ into the inference dynamics, to 'push' the generations towards areas of higher reward (Dhariwal & Nichol, 2021; Chung et al., 2023; Bansal et al., 2024). While effective at improving sample quality, these heuristic modifications to the dynamics come with a poor theoretical understanding of the sampled distribution.

Corresponding author: howard@stats.ox.ac.uk

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

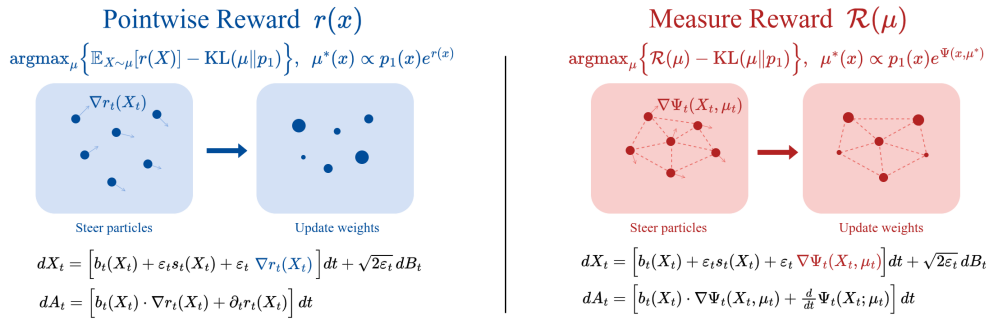


Figure 1: *(Left)* **Standard Inference-Time Steering:** For pointwise rewards, the reward-gradients ∇r_t can be incorporated into the inference dynamics, to steer the particles independently towards areas of higher reward. Feynman-Kac corrections are applied to the log-weights A_t to ensure the weighted particles trace the correct marginals $\mu_t^*(\cdot) \propto p_t(\cdot)e^{r_t(\cdot)}$. *(Right)* **Mean-Field Inference-Time Steering:** For a *measure-defined* reward function $\mathcal{R}(\mu)$, we incorporate the gradient of the first-variation, $\nabla\Psi_t(\cdot, \mu_t)$ as the steering term, leading to interaction between the particles. Through a mean-field framework, we derive the corrective reweighting procedure to ensure the reweighted interacting particles trace out the target tilted measures $\mu_t^*(\cdot) \propto p_t(\cdot)e^{\Psi_t(\cdot,\mu^*)}$.

This limitation has motivated a large number of recent works aiming to place inference-time steering on a firmer theoretical footing. To do so, the dominant paradigm aims to sample from the KL-regularised objective,

$$\arg\max_{\mu} \left\{ \mathbb{E}_{X\sim\mu}[r(X)] - \text{KL}(\mu\|p_1) \right\} \tag{1}$$

which corresponds to the *exponentially-tilted* distribution $\mu_1^*(x_1) \propto p_1(x_1)e^{r(x_1)}$. Since gradient-steering alone does not sample from this target tilted measure, recent works have sought to correct for the discrepancy using Feynman-Kac particle reweighting schemes, enabling correct tracking of the desired distributional path via SMC resampling procedures (Wu et al., 2023; Thornton et al., 2025; Singhal et al., 2025; Skreta et al., 2025; He et al., 2026; Sabour et al., 2025).

Distributional-Reward Steering Following these recent advances, steering according to pointwise-defined reward functions is now well-understood and based upon a principled theoretical foundation. However, many important control objectives are inherently *distributional* rather than pointwise. In many applications, rather than aiming for individual samples to score highly, one may wish to shape the overall generated distribution according to global properties, such as calibration to experimental observations, balancing across modes, or encouraging diversity. To formalise such distributional-steering, we will instead consider optimising according to a *measure-defined* reward function $\mathcal{R} : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$,

$$\arg\max_{\mu} \left\{ \mathcal{R}(\mu) - \text{KL}(\mu\|p_1) \right\}. \tag{2}$$

A number of existing works have incorporated steering terms that operate at the distributional level. For example, Corso et al. (2024); Kirchhof et al. (2025); Lam et al. (2026) steer towards generating diverse samples, Maddipatla et al. (2025) steer according to experimental observations, Saravanan et al. (2026) steer to increase expected information gain, and Sani et al. (2026) guide with MMD gradients. However, as in the pointwise-reward case, such approaches are largely heuristic in nature and it is not clear precisely what distribution these procedures target. Unlike in the pointwise-reward setting, a theoretically-grounded framework and approach for *distributional* inference-time steering of diffusion models is currently lacking.

Contributions In this work, we address this gap by reconciling these two currently distinct literatures through a *mean-field framework*, allowing us to place distributional steering on a similarly principled footing to the pointwise case. In particular, we make the following contributions.

- We formalise distribution-level steering (2) as aiming to sample from a tilted distribution for a measure-defined reward function $\mathcal{R}$, in which the tilt now depends on the measure itself. This gives a distributional analogue of the reward-tilted targets used in the pointwise case.

- Using this mean-field framework, we derive weighted McKean-Vlasov dynamics whose marginals correctly track the desired tilted path. From this, we develop a novel weighted interacting-particle sampling procedure that enables principled distribution-level steering.
- We show that our framework recovers standard pointwise reward steering as a special case, and provides a principled counterpart to existing gradient-steering methods for distributional control. Empirically, we validate that our procedure correctly targets the tilted distribution in verifiable low-dimensional settings, and examine its behaviour on higher-dimensional protein conformation tasks relative to standard gradient-steering approaches.

2 Background: Inference-Time Steering for Pointwise Rewards

In this work, we will largely follow the presentation of Sabour et al. (2025). To begin, let us consider a flow-based model given by the dynamics

$$\mathrm{d}X_t = \big[b_t(X_t) + \varepsilon_t s_t(X_t)\big]\mathrm{d}t + \sqrt{2\varepsilon_t}\,\mathrm{d}B_t. \tag{3}$$

Here, ε_t is an arbitrary noise schedule, and s_t is the score function which can be obtained from a transformation of the vector field b_t (see Albergo et al. (2023), for example). These dynamics trace out a path of marginals p_t which interpolates from p_0 to the data distribution p_1.

Given a reward r defined on the generated samples x_1, inference-time steering methods aim to modify the generation procedure to instead sample from the *reward-tilted* distribution, which is the maximiser of the KL-regularised objective (1),

$$\mu^*(x) \propto p_1(x)e^{r(x)}. \tag{4}$$

We remark that inference-time steering differs from the fine-tuning paradigm (Clark et al., 2024; Xu et al., 2023; Domingo-Enrich et al., 2025), as there is no additional training. Here, the dynamics are modified for the currently-chosen reward function r, and different choices of r can be used without having to perform extra training.

Gradient-Steered Dynamics The aim of inference-time steering methods (Singhal et al., 2025; Skreta et al., 2025; He et al., 2026; Sabour et al., 2025) is to 'push' the particles towards areas of higher reward during the generation, while performing SMC-style reweighting of the particles to ensure they trace out a prescribed path of marginal distributions from p_0 to μ^*. To do so, one chooses a time-dependent reward function r_t which satisfies $r_0 = 0$ and $r_1 = r$; we aim to trace out the *tilted* path of marginals $\mu_t^*(x) \propto p_t(x)e^{r_t(x)}$, which will interpolate from p_0 to the target tilted distribution $\mu^* \propto p_1(x)e^{r(x)}$. A common approach is then to modify the dynamics by incorporating a reward-gradient steering term,

$$\mathrm{d}X_t = \big[b_t(X_t) + \varepsilon_t s_t(X_t) + \varepsilon_t \nabla r_t(X_t)\big]\mathrm{d}t + \sqrt{2\varepsilon_t}\mathrm{d}B_t. \tag{5}$$

As the original reward function r is defined on data, a natural choice is

$$r_t(x_t) = t\, r(\hat{x}_1(t, x_t)), \tag{6}$$

where $\hat{x}_1(t, x_t)$ is an estimate of the terminal point x_1 given the current point (t, x_t) in the trajectory. In this work, we will consider using a reward construction that utilises the denoiser estimate, though we note that this framework is suitable for any such choice of r_t (see Sabour et al. (2025), for example).

Corrective Feynman-Kac Reweighting While the additional steering term pushes the particles towards areas of higher reward, the dynamics (5) do not in general follow the desired tilted marginal path μ_t^*. To correct for the discrepancy, it is necessary to adjust the measure using particle weights, which can then be used in an SMC-style resampling scheme to target μ^* in a theoretically-principled manner. The log-weights A_t^i evolve according to the dynamics

$$\mathrm{d}A_t = \Big[b_t(X_t) \cdot \nabla r_t(X_t) + \partial_t r_t(X_t)\Big]\,\mathrm{d}t. \tag{7}$$

Note that one can incorporate additional 'pushing' terms rather than only weighting by ε_t in (5), though this would introduce additional divergence terms in the weight dynamics (see Sabour et al. (2025), for example); for simplicity, we therefore consider the steering outlined above in our presentation.

3 A Mean-Field Framework for Inference-Time Distributional Control

3.1 Distributional-Reward Measure Tilting

Consider now that we want to tilt towards a desirable *distributional* property of the generated measure, quantified by a measure-dependent reward function $\mathcal{R} : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ defined on the terminal distribution μ_1. We can formalise this as aiming to sample from

$$\mu^* = \arg\max_{\mu} \left\{ \mathcal{R}(\mu) - \mathrm{KL}(\mu \| p_1) \right\}, \tag{8}$$

assuming for the moment that the maximiser is unique. To understand the properties of μ^*, let us introduce the functional derivative $\Psi(x, \mu) := \frac{\delta \mathcal{R}}{\delta \mu}(x, \mu)$, which quantifies the change of the reward $\mathcal{R}$ when the measure μ is perturbed (for a rigorous definition and examples, see Appendix A.2). Using this object, we can characterise μ^* as an exponential tilt, mirroring (4) for pointwise rewards.

Proposition 3.1 (First-order optimality condition)**.** *Assuming mild regularity properties and concavity of $\mathcal{R}$, the maximiser in (2) is unique and satisfies*

$$\mu^*(\mathrm{d}x) = \frac{1}{Z} e^{\Psi(x, \mu^*)} \, p_1(x) \, \mathrm{d}x, \tag{9}$$

for normalising constant $Z = \int_{\mathbb{R}^d} e^{\Psi(x, \mu^*)} \, p_1(x) \, \mathrm{d}x$.

The proof, essentially setting the functional derivative of the objective $\mathcal{R}(\mu) - \mathrm{KL}(\mu \| p_1)$ to zero, can be found in Appendix D.1. In contrast to (4), the tilt (9) is *implicit*, as the target measure μ^* appears on both sides of the equation and the tilting potential $\Psi(x, \mu^*)$ is itself measure-dependent. Theorem 3.1 therefore places the problem in a *mean-field regime*—the optimality condition is a self-consistency relation, in which the target measure is characterised by a potential that depends on the measure itself.

As in the standard case from Section 2, let us fix a family of measure-defined reward functions $\mathcal{R}_t : \mathcal{P}(\mathbb{R}^d) \to \mathbb{R}$ interpolating from $\mathcal{R}_0 = 0$ to $\mathcal{R}_1 = \mathcal{R}$. Extending (9), we can define the curve of distributions

$$\mu_t^*(\mathrm{d}x) = \frac{1}{Z_t} e^{\Psi_t(x, \mu_t^*)} \, p_t(x) \, \mathrm{d}x, \qquad t \in [0, 1], \tag{10}$$

tracing from p_0 to μ_1^*. As in (2), the measures μ_t^* are defined implicitly and self-consistently through the corresponding measure-dependent potentials $\Psi_t(x, \mu_t)$, similarly defined as functional derivatives of the time-dependent rewards $\mathcal{R}_t$. We note that these intermediate measures μ_t^* satisfy analogous variational principles, namely $\mu_t^* = \arg\max_\mu \left\{ \mathcal{R}_t(\mu) - \mathrm{KL}(\mu \| p_t) \right\}$. Following (6), in this work we will take $\mathcal{R}_t = t \, \mathcal{R}(\hat{x}_1(\cdot, t) \# \mu)$, using the denoiser estimate $\hat{x}_1$.

3.2 Idealised Dynamics

Distributionally-Steered Dynamics We will use this mean-field framework to develop a probabilistically-principled inference-time steering procedure that traces out this distributional path μ_t^* during the generation process. As in the standard setting described in Section 2, it is again intuitive to include a gradient-steering term to 'push' the generated distribution in the direction of higher reward during the inference procedure. To do so, one can consider the modified dynamics,

$$\mathrm{d}X_t = \left[b_t(X_t) + \varepsilon_t s_t(X_t) + \varepsilon_t \nabla_x \Psi_t(X_t; \mu_t) \right] \mathrm{d}t + \sqrt{2\varepsilon_t} \, \mathrm{d}B_t. \tag{11}$$

In fact, such gradient-steering resembles many existing works for batch-level reward steering that incorporate gradient terms into the inference dynamics, for example for repulsive potentials (Corso et al., 2024; Lam et al., 2026), ensemble likelihoods given experimental data (Maddipatla et al., 2025), expected information gain (Saravanan et al., 2026), and maximum mean discrepancy to a reference dataset (Sani et al., 2026).

Corrective Mean-Field Reweighting Just as in the standard case, merely incorporating a gradient steering term into the dynamics does not in general follow the target marginals μ_t^*, nor come with any strong theoretical understanding of the generated distribution. It is therefore again necessary to

Algorithm 1: Mean-Field Inference-Time Steering for Diffusion Models (Single Batch)

Input: Number of particles N, distributional reward function $\mathcal{R}$ and corresponding first variation Ψ_t, time increment δt or alternative time schedule.
Initialisation: Sample $X_0^{1:N} \sim p_0^{\otimes N}$, initialise log-weights $A_0^i = 0$.
For $t \in \{0, \delta t, 2\delta t, \dots, 1-\delta t\}$:

- **Positional updates:**
  - Perform discretisation step for positional updates,
    $$X_{t+\delta t}^i = X_t^i + \big[b_t(X_t^i) + \varepsilon_t s_t(X_t^i) + \varepsilon_t \nabla_x \Psi_t(X_t^i; \hat{\mu}_t^N)\big]\delta t + \sqrt{2\varepsilon_t}\,\delta B_t^i,$$
    where $\hat{\mu}_t^N = \sum_{j=1}^N w_t^j \delta_{X_t^j}$ for $w_t^j \propto e^{A_t^j}$.
- **Log-weight updates:**
  - Compute $\frac{\mathrm{d}}{\mathrm{d}t}\Psi_t(\cdot; \hat{\mu}_t^N)$ at points X_t^i using Algorithm 3 or Algorithm 4.
  - Compute mean-field FK potentials $g_t^i = b_t(X_t^i) \cdot \nabla_x \Psi_t(X_t^i; \hat{\mu}_t^N) + \frac{\mathrm{d}}{\mathrm{d}t}\Psi_t(X_t^i; \hat{\mu}_t^N)$.
  - Perform discretisation step for log-weight updates, $A_{t+\delta t}^i = A_t^i + g_t^i \delta t$.
- **Resample:** Optionally resample according to current weights $w_t^i \propto e^{A_t^i}$.

Return: Weighted particles (X_1^i, w_1^i), where $w_1^i \propto e^{A_1^i}$.

incorporate a Feynman-Kac style reweighting scheme to correct for this discrepancy. This setting differs from the standard pointwise case, as the particles now *interact* during the inference dynamics due to the dependence of the steering term on the evolving distribution. Fortunately, by appealing to our mean-field framework we can derive the corrective potentials under this interacting particle behaviour. This yields our main result, in which we construct mean-field type, or McKean-Vlasov, dynamics whose coefficients are chosen so that the corresponding law follows the desired path μ_t^*.

Theorem 3.2 (Weighted McKean-Vlasov Dynamics). *Consider the dynamics*

$$\mathrm{d}X_t = \big[b_t(X_t) + \varepsilon_t s_t(X_t) + \varepsilon_t \nabla_x \Psi_t(X_t; \mu_t)\big]\mathrm{d}t + \sqrt{2\varepsilon_t}\,\mathrm{d}B_t,\ X_0 \sim p_0 \tag{12a}$$

$$\mathrm{d}A_t = \big[b_t(X_t) \cdot \nabla_x \Psi_t(X_t; \mu_t) + \dot{\Psi}_t(X_t)\big]\mathrm{d}t,\ A_0 = 0 \tag{12b}$$

with the weighted measure $\mu_t \propto \mathrm{Law}(X_t)e^{A_t}$, *and* $\dot{\Psi}_t(x) := \frac{\mathrm{d}}{\mathrm{d}t}\Psi_t(x; \mu_t)$. *Then, under mild conditions,* $\mu_t = \mu_t^*$, *for all* $t \in [0, 1]$.

Note that these dynamics strongly resemble the pointwise setting (7). However, a key difference is that now the time-derivative term $\partial_t r(\cdot)$ has been replaced with the term $\dot{\Psi}_t(\cdot) = \frac{\mathrm{d}}{\mathrm{d}t}\Psi_t(\cdot, \mu_t)$; here, the position x is kept fixed but the time-dependence now arises through both Ψ_t and the measure evolution μ_t. Again in contrast to the standard case, this induces *implicit* behaviour in the log-weight updates—their dynamics involve the change in the measure μ_t, but the measure evolution is itself dependent on the change in the weights. We remark that the term $\dot{\Psi}_t$ can be equivalently characterised as solving the following implicit expression, which we will later use in our inference procedure.

Proposition 3.3. *Define the second variation of the reward functional,* $\Phi_t(x, y, \mu) = \frac{\delta^2 \mathcal{R}_t}{\delta \mu^2}(x, y, \mu)$, *and a mean-centred version* $\tilde{\Phi}_t(x, y, \mu) = \Phi_t(x, y, \mu) - \int \Phi_t(x, z, \mu)\,\mu(\mathrm{d}z)$. *Then, under the target flow* (10)*, the time derivatives* $\dot{\Psi}_t$ *satisfy the following implicit equality:*

$$\dot{\Psi}_t(x) = \partial_t \Psi_t(x, \mu_t^*) + \int_{\mathbb{R}^d} \Big(\tilde{\Phi}_t(x, y, \mu_t^*)\dot{\Psi}_t(y) + b_t(y) \cdot \nabla_y \tilde{\Phi}_t(x, y, \mu_t^*) + \tilde{\Phi}_t(x, y, \mu_t^*)\, b_t(y) \cdot \nabla_y \Psi_t(y, \mu_t^*)\Big)\,\mathrm{d}\mu_t^*(y). \tag{13}$$

3.3 Method

To turn the idealised dynamics of Theorem 3.2 into a finite-particle algorithm, we approximate the McKean-Vlasov system (12) with an ensemble $(X_t^i, w_t^i)_{i=1}^N$, where the weighted empirical measure $\hat{\mu}_t = \sum_{j=1}^N w_t^j \delta_{X_t^j}$ approximates μ_t. To do so, we need the following terms.

Gradient-Steering Term $\nabla_x \Psi_t(x; \mu_t)$ The gradient-steering term $\nabla_x \Psi_t(x; \mu_t)$ in the positional updates can be easily computed with autodifferentiation, evaluated on the current empirical measure $\hat{\mu}_t = \sum_{j=1}^N w_t^j \delta_{X_t^j}$ with weights $w_t^j \propto e^{A_t^j}$. When one can analytically compute the first variation (as is often the case; see Appendix A.2), then one can use a finite-sample approximation $\Psi_t(X_t^i, \hat{\mu}_t)$ and then differentiate with respect to the first argument. Alternatively, one could approximate the steering term using $(w_t^i)^{-1} \nabla_{X_t^i} \mathcal{R}_t(\hat{\mu}_t)$, by directly evaluating the reward on the empirical measure and differentiating with respect to the positions X_t^i (see Appendix B).

Time-Derivative Term $\dot{\Psi}_t(x)$ The more challenging term in the dynamics is the time-derivative $\dot{\Psi}_t(x) = \frac{\mathrm{d}}{\mathrm{d}t}\big(\Psi_t(x, \mu_t)\big)$ due to its implicit behaviour. We provide two different approaches for computing this term. The first involves solving a linear system based on the implicit expression in (13), which involves only tractable terms, and the second solves for $\dot{\Psi}_t(x)$ through a lightweight Picard iteration scheme; we describe these in Algorithms 3 and 4 respectively. We remark that the cost of such procedures is low, and the computational cost is often dominated by the network evaluations rather than these lightweight solvers (see Appendix C.4). We found that the choice of solver for $\dot{\Psi}_t(x)$ had little effect on the empirical performance of the method and we therefore primarily used the fixed-point iterative solver for simplicity, but we remark that the first approach may be useful in certain settings.

SMC Resampling As in the pointwise-reward Feynman-Kac steering approaches (Singhal et al., 2025; Skreta et al., 2025), we use an SMC-style resampling scheme to avoid weight degeneracy. Any such scheme could be used; in our experiments we use residual resampling at fixed intervals.

Algorithm We outline the inference procedure for a single batch of N particles in Algorithm 1, which will approximate the target tilted measure μ^* as the batch size N goes to infinity. In practice, the batch size is limited by computational constraints; we therefore also provide a multi-batch procedure in Algorithm 2 in Appendix B.

Convergence in N Under mild assumptions on the dynamics and the $\dot{\Psi}_t$ solver (see Assumption D.4), we can show convergence of the finite-particle system to the idealised dynamics (12). In particular, for δ_N (defined in (43)) quantifying the approximation error when solving for $\dot{\Psi}_t$, we have the following convergence result.

Theorem 3.4 (Convergence of the Weighted Particle System). *Under mild assumptions, we have that*

$$\mathbb{E}\left[\sup_{t\in[0,1]} \left|\hat{\mu}_t^N(\phi) - \mu_t^*(\phi)\right|\right] \longrightarrow 0, \tag{14}$$

for every $\phi \in C_b^2(\mathbb{R}^d)$, as $N \to \infty$.

Generalising the Pointwise Setting We remark that the pointwise-reward setting is recovered from our framework by choosing $\mathcal{R}(\mu) = \int r(x)\,\mathrm{d}\mu(x)$. In this case, the first variation is $\Psi(x, \mu) = r(x)$, so our mean-field steering procedure recovers the standard procedure from Section 2.

4 Experiments

In this section, we empirically examine the behaviour of our proposed framework, aiming to verify that the resulting samples are drawn from the prescribed reward-tilted distribution. We first consider low-dimensional settings where the reward-tilted target can be tractably computed, allowing us to reliably quantify the sampling accuracy. Following this, we consider larger-scale illustrative case studies on proteins, where we observe similar behaviour in higher-dimensional, realistic settings.

4.1 Low-dimensional experiments

We first consider a flow model that samples from a 1-dimensional bimodal Gaussian mixture p_1 with means $(-1.0, 1.0)$, both modes with unit variance, and mode weights $(1, 3)$ respectively. To construct

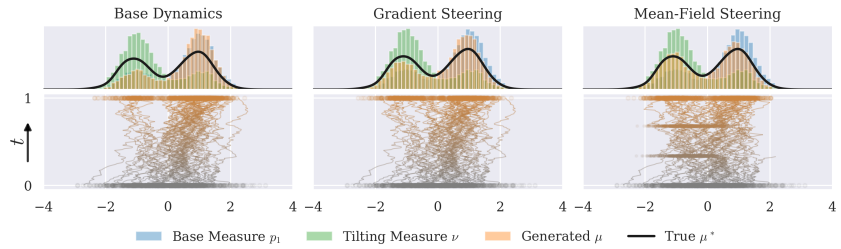


Figure 2: Trajectories and final measures μ_1 for the base, gradient-steered, and mean-field dynamics, alongside an approximation of the true reward-tilted target μ^* for comparison.

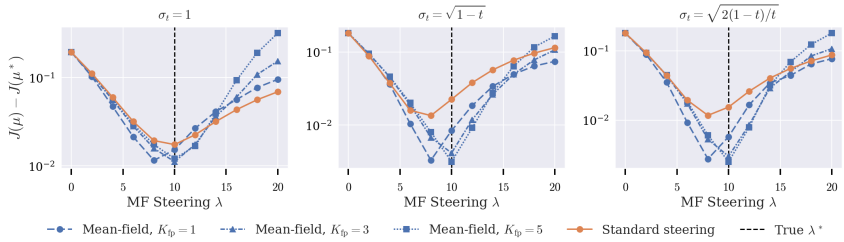


Figure 3: Objective gaps $J(\mu) - J(\mu^*)$ for mean-field sampling and gradient steering in the MMD example with $\lambda^* = 10$. We compare fixed-point iterations $K_{FP} = 1, 3, 5$ for our method against gradient steering with the steering strength varied linearly over $[0, 4]$.

a distributional reward, we consider tilting the output distribution towards a similar Gaussian mixture ν but now with respective weights $(3, 1)$ using a squared Maximum Mean Discrepancy (MMD) reward (similar to Sani et al. (2026)). We aim to sample from

$$\mu^* = \arg\min_{\mu} \left\{ \mathrm{KL}(\mu \| p_1) + \lambda^* \, \mathrm{MMD}^2(\mu, \nu) \right\}. \tag{15}$$

Trajectory Visualisation In Figure 2, we qualitatively compare trajectories for the base dynamics, the gradient-steered dynamics (representative of existing approaches such as Corso et al. (2024); Maddipatla et al. (2025); Lam et al. (2026); Saravanan et al. (2026); Sani et al. (2026)), and our mean-field steering approach. We use a constant noise schedule $\sigma_t = 1.0$ and $\lambda^* = 10$, and include an approximation to the true tilted solution μ^* for reference. As expected, the base dynamics sample the original measure p_1. We see that standard gradient steering does push the particles towards the tilting measure ν, but does not target μ^*. In contrast, incorporating the mean-field reweighting procedure accurately samples from the tilted target μ^*.

Quantitative Analysis of Optimality We quantitatively assess how accurately each sampler targets the tilted distribution μ^*. For the objective $J(\mu) = \mathrm{KL}(\mu \| p_1) + \lambda^* \mathrm{MMD}^2(\mu, \nu)$, we report the optimality gap $J(\mu) - J(\mu^*)$. In Figure 3, we compare gradient steering with our mean-field sampler using fixed-point iterations to compute $\dot{\Psi}_t$. Results are shown for three noise schedules: fixed $\sigma_t = 1.0$, decaying $\sigma_t = \sqrt{1-t}$, and the memoryless schedule $\sigma_t = \sqrt{2(1-t)/t}$ used in Domingo-Enrich et al. (2025).

In all cases, the objective gap is minimised when using mean-field steering with the correct λ value, confirming that our procedure correctly targets the prescribed distribution μ^*. The behaviour is consistent across noise schedules, indicating that the targeted distribution is invariant to these hyperparameter choices. Gradient-steering, by contrast, samples from a measure whose form depends on the steering strength and which does not in general coincide with μ^*. The mean-field procedure thus provides a sampler whose target distribution is explicitly characterised by λ, rather than being indirectly induced by the choice of steering parameters.

We also note that running $K_{FP} = 1$ in the fixed-point $\dot{\Psi}_t$ solver appears to give a slight bias as expected, but running $K_{FP} = 3, 5$ are highly accurate and perform very similarly. We provide full experimental details and additional results, including for different tilting strengths λ, different measure-defined rewards $\mathcal{R}$, and a more detailed analysis of the fixed-point solver, in Appendix C.1.

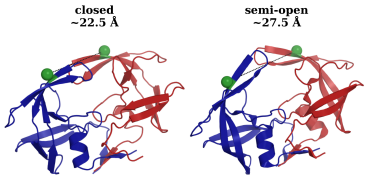


Figure 4: HIV-1 Protease exhibits two main conformations: closed *(left)* and semi-open *(right)*.

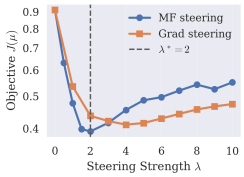


Figure 5: The objective proxy is minimised at the correct value for the steering coefficient λ.

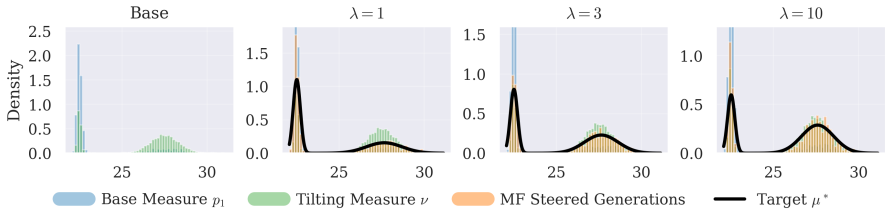


Figure 6: Distributions of the residue-55 distances in the generated conformations in the HIV-1 protease example. Increasing the strength λ increases the tilt towards the experimentally-observed distribution ν.

4.2 Tilting Protein Conformational Ensembles Towards Experimental Observations

In the previous section, we verified that our approach provides a principled correction mechanism for distributional steering in controlled low-dimensional settings. We now investigate to what extent this holds in higher-dimensional generative modelling tasks. Diffusion models are widely used for complex modalities such as images, videos, and proteins; among these, protein generation is a particularly natural application for the distributional steering setting, since the generated ensemble is often itself the object of interest and experimental observations frequently provide indirect, distribution-level information. This has motivated recent work on steering generated protein ensembles using distributional reward functions derived from experimental observations (Maddipatla et al., 2025; 2026).

4.2.1 Distributions Over Residue Distances

We first consider two settings in which we steer conformational ensembles generated by Boltz-2 (Passaro et al., 2025; Wohlwend et al., 2024) towards experimentally observed distributions over lower-dimensional features, again comparing relative to gradient-steering. We remark that our aim is not necessarily to improve over gradient-steering; indeed, this is highly effective at steering towards desired distributional behaviour. Rather, we aim to understand to what extent the improved consistency and predictability afforded by our theoretically grounded correction mechanism extends to these higher-dimensional, challenging settings.

Protein Conformation Generation Proteins typically occupy ensembles of conformational states rather than a single fixed structure, and characterising these ensembles is crucial for understanding disease. Recent diffusion models can sample plausible conformations conditioned on an amino-acid sequence, but their induced conformational distributions may be miscalibrated, for example collapsing onto certain modes or assigning incorrect mode proportions. Experimental methods can probe conformational ensembles, but they often provide indirect, distribution-level information rather than individual measurements. Such data can nevertheless reveal approximate conformational state proportions; this motivates combining experimental data with generative modelling, by 'tilting' the base model toward empirically observed distributional information.

HIV-1 Protease We first consider HIV-1 protease; this protein exhibits a 'flap region' that opens and closes between different conformational states (Figure 4), which can be inferred from the residue-55 $C_\alpha - C_\alpha$ distances. Under the base Boltz-2 model, the states are sampled with proportions 79% closed and 21% semi-open. The double electron-electron resonance (DEER) method can be used to infer state proportions; such analysis in Liu et al. (2016) suggests that these relative proportions are 25% and 75%. Based on these findings, we will aim to tilt generations towards a Gaussian mixture

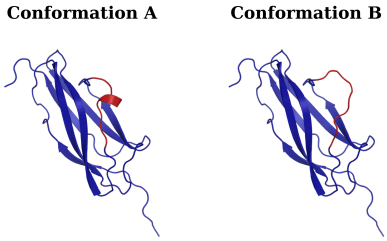


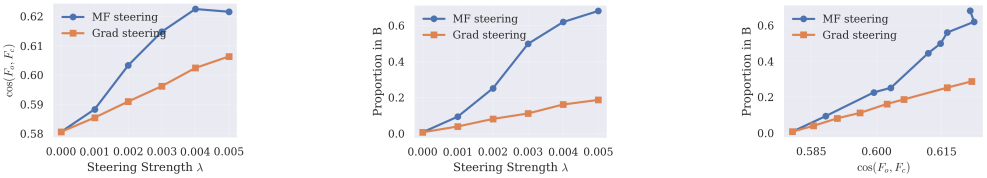
Figure 7: Two conformations of 4OLE, with the region of interest (residues 423-431) highlighted in red.



(a) Base distribution p_1.

(b) Steered distribution μ_1.

Figure 8: X-ray crystallography experiment and figure inspired by Maddipatla et al. (2025), showing the sampled ROI and the target electron density. The base diffusion model samples nearly entirely conformation A. Distributional steering can cover both modes, giving a better fit to the observed density.



(a) Cosine alignment between generated and target electron densities, as steering strength increases.

(b) Proportion of generations closer to conformation B, as steering strength increases.

(c) For the same cosine alignment, mean-field steering placed more samples closer to conformation B.

Figure 9: Comparing gradient-only and mean-field steering, when steering 4OLE generations according to the observed electron density.

ν over residue-55 distances with these proportions, using the MMD objective (15) but here instead evaluated on the 1-dimensional residue distances (that is, $\mathcal{R}(\mu) = -\lambda\,\mathrm{MMD}^2(T\#\mu, \nu)$, where T is the projection onto the 1 dimensional space). We emphasise that the diffusion process itself takes place in the full 4608-dimensional space.

Results In Figure 6 we display histograms of the residue-55 distance for different tilt-strengths λ using our method. For reference, we also plot a proxy for the target μ^* similarly to before; as the ambient-space $\mathrm{KL}(\mu\|p_1)$ term is intractable, we instead evaluate KL on the 1-dimensional residue distances (that is, we report an finite-sample approximation to $\mathrm{KL}(T\#\mu\|T\#p_1)$ where T is the projection onto the 1 dimensional space; for details, see Appendix C.2). We see that, as expected, the generations are pulled towards the observed data as λ increases, and the induced distribution of residue distances accurately matches the 'ground-truth' proxy μ^* (we give similar visualisations for gradient-steering in Appendix C.2). As before, in Figure 5 we see that the objective is minimised at the correct λ value when using mean-field steering, demonstrating that our principled approach remains effective even in this challenging, high-dimensional setting.

In Appendix C.3, we also consider another Boltz-2 example in which generations of adenylate kinase are steered to match a two-dimensional distribution over bond angles obtained from molecular dynamics simulations.

4.2.2 Crystallographic Electron Densities

To conclude, we consider a more challenging distributional-reward steering example, in which we steer the generations of a diffusion model to fit an electron density observed using X-ray crystallography. This technique provides a spatial and temporal average of the density over molecules in a crystal lattice, therefore constituting a population-level observation requiring a distribution-level reward. Here, we consider an example from Maddipatla et al. (2025), who steer the Protenix model (ByteDance et al., 2025) so that the generated ensemble's density is guided towards the empirical density. Here, we investigate how this steering mechanism is altered by including our additional mean-field correction.

We consider the PDB structure 4OLE, which is known to display heterogeneous conformations within the same crystal structure around a small region of interest (ROI) (Figure 7). The base model generates almost entirely from the helical conformation A, which does not explain the bimodal electron density in this region. By steering according to a log-likelihood reward function, Maddipatla et al. (2025) show that both modes can be recovered, giving an improved fit to the empirical electron density (Figure 8).

Inspired by this experiment, in Figure 9 we compare the behaviour of standard gradient-only steering (similar to the steering used in Maddipatla et al. (2025)) with the additional inclusion of our mean-field reweighting scheme. For full details of the experimental setup, see Appendix C.5. In the absence of a known ground-truth, we instead focus on discussing the qualitative differences between the results. We first observe that for the same steering strength, mean-field steering appears to result in a stronger reward tilt in this setting, giving a larger cosine alignment and more samples closer to conformation B. We also observe that mean-field steering generally placed more samples in the alternate conformation for the same cosine alignment.

5 Discussion

Limitations The primary limitation of our approach is its reliance on resampling. While similar schemes are used in the pointwise-reward steering setting too, this can be a limitation in certain modalities (particularly images) as the final generations may include versions of similar images. For settings such as conformation generation, this is typically less of an issue. We anticipate that this can be mitigated using methods to reduce weight variance (Ren et al., 2026). We also require a differentiable reward function and an additional computational cost for the $\dot{\Psi}_t$ solver. Finally, we remark that exact minimisation of a distributional objective is often not strictly necessary in practice, and gradient-steering alone may suffice in such cases; our framework can be seen as providing a principled foundation underlying such approaches.

Conclusion and Future Directions In this work, we have studied distributional reward steering under a mean-field framework, deriving a Feynman-Kac reweighting scheme that exactly targets the prescribed reward-tilted measure μ^*. Our framework provides a principled counterpart to existing batch-level gradient-steering methods, and recovers existing FK-weighted steering for pointwise rewards as a special case. We empirically verified that our proposed framework targets the true tilted distribution in low-dimensional settings where the target is tractable, and illustrated the extent to which this behaviour transfers to higher-dimensional protein conformation tasks. We hope that future work can build upon the framework developed here, drawing on domain expertise to formulate more expressive distributional steering objectives inspired by practical applications.

Acknowledgments and Disclosure of Funding

SH is supported by the EPSRC CDT in Modern Statistics and Statistical Machine Learning [EP/S023151/1]. The authors acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR) (McIntosh-Smith et al., 2024). Isambard-AI is operated by the University of Bristol and is funded by the UK Government's Department for Business, Innovation, Science and Trade (DBIST) via UK Research and Innovation; and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023]. The authors have no competing interests to declare.

References

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J., et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023.

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions, 2023.

Azangulov, I., Zhang, L., and Ashouritaklimi, K. Variance-tilted diffusion models for diverse sampling. In *ICML 2026 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2026.

Bansal, A., Chu, H.-M., Schwarzschild, A., Sengupta, R., Goldblum, M., Geiping, J., and Goldstein, T. Universal guidance for diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Beckstein, O., Denning, E. J., Perilla, J. R., and Woolf, T. B. Zipping and unzipping of adenylate kinase: Atomistic insights into the ensemble of open-closed transitions. *Journal of Molecular Biology*, 394(1):160–176, 2009. ISSN 0022-2836.

Black, K., Janner, M., Du, Y., Kostrikov, I., and Levine, S. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

ByteDance, A. A. T., Chen, X., Zhang, Y., Lu, C., Ma, W., Guan, J., Gong, C., Yang, J., Zhang, H., Zhang, K., Wu, S., Zhou, K., Yang, Y., Liu, Z., Wang, L., Shi, B., Shi, S., and Xiao, W. Protenix - advancing structure prediction through a comprehensive AlphaFold3 reproduction. *bioRxiv*, 2025. doi: 10.1101/2025.01.08.631967.

Carmona, R. and Delarue, F. *Probabilistic Theory of Mean Field Games with Applications I: Mean Field FBSDEs, Control, and Games*, volume 83 of *Probability Theory and Stochastic Modelling*. Springer International Publishing, Cham, 2018.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

Clark, K., Vicol, P., Swersky, K., and Fleet, D. J. Directly fine-tuning diffusion models on differentiable rewards. In *The Twelfth International Conference on Learning Representations*, 2024.

Corso, G., Xu, Y., Bortoli, V. D., Barzilay, R., and Jaakkola, T. S. Particle guidance: non-i.i.d. diverse sampling with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Del Moral, P. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, 2004.

Del Moral, P. and Miclo, L. Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering. In *Seminaire de probabilites XXXIV*, pp. 1–145. Springer, 2007.

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

Domingo-Enrich, C., Drozdzal, M., Karrer, B., and Chen, R. T. Q. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025.

Douc, R., Cappé, O., and Moulines, E. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA 2005)*, pp. 64–69. IEEE, 2005. doi: 10.1109/ISPA.2005.195385.

Fadini, A., Li, M., McCoy, A. J., Banjara, S., Okumura, H., Napier, E., Fontana, P., Khan, A. R., Jovine, L., Terwilliger, T. C., Read, R. J., Hekstra, D. R., and AlQuraishi, M. AlphaFold as a prior: experimental structure determination conditioned on a pretrained neural network. *Nature Methods*, 23(4):785–795, 2026. ISSN 1548-7105. doi: 10.1038/s41592-026-03047-4.

Genevay, A., Peyré, G., and Cuturi, M. Learning generative models with Sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Giesecke, K., Spiliopoulos, K., and Sowers, R. B. Default clustering in large portfolios: Typical events. *The Annals of Applied Probability*, 23(1):348–385, 2013. doi: 10.1214/12-AAP845.

Havens, A. J., Miller, B. K., Yan, B., Domingo-Enrich, C., Sriram, A., Levine, D. S., Wood, B. M., Hu, B., Amos, B., Karrer, B., Fu, X., Liu, G.-H., and Chen, R. T. Q. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. In *International Conference on Machine Learning*, 2025.

He, J., Hernández-Lobato, J. M., Du, Y., and Vargas, F. RNE: plug-and-play diffusion inference-time control and energy-based training. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022.

Hofbauer, J. and Sigmund, K. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

Holderrieth, P., Chen, D., Eyring, L., Shah, I., Anantharaman, G., He, Y., Akata, Z., Jaakkola, T., Boffi, N. M., and Simchowitz, M. Diamond maps: Efficient reward alignment via stochastic flow maps. In *Forty-third International Conference on Machine Learning*, 2026.

Howard, S., Nüsken, N., and Pidstrigach, J. Control consistency losses for diffusion bridges. In *Forty-third International Conference on Machine Learning*, 2026.

Jakubowski, A. On the skorokhod topology. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 22(3):263–285, 1986.

Joint Center for Structural Genomics (JCSG). Crystal structure of a neighbor of BRCA1 gene 1 (NBR1) from Homo sapiens at 2.52 A resolution. Protein Data Bank (PDB ID: 4OLE), 2014. URL `https://www.rcsb.org/structure/4OLE`.

Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, Sep 1976.

Kirchhof, M., Thornton, J., Béthune, L., Ablin, P., Ndiaye, E., and Cuturi, M. Shielded diffusion: Generating novel and diverse images using sparse repellency. In *Forty-second International Conference on Machine Learning*, 2025.

Lam, H., Pujalte Ojeda, S., Brezinova, M., Hanke, J., Ong, X. E., Mu, Y., and Vendruscolo, M. Metadiffusion: inference-time meta-energy biasing of biomolecular diffusion models. *bioRxiv*, 2026. doi: 10.64898/2026.02.10.704873.

Levy, A., Chan, E. R., Fridovich-Keil, S., Poitevin, F., Zhong, E. D., and Wetzstein, G. Solving inverse problems in protein space using diffusion-based priors, 2024.

Li, M., Han, J., Cossio, P., and Wu, L. Robust inference-time steering of protein diffusion models via embedding optimization, 2026.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Liu, G.-H., Choi, J., Chen, Y., Miller, B. K., and Chen, R. T. Q. Adjoint Schrödinger bridge sampler. In *Conference on Neural Information Processing Systems*, 2025a.

Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Liu, Y., Yu, Z., Lindsay, R. J., Lin, G., Chen, M., Sahoo, A., and Hanson, S. M. Exendiff: An experiment-guided diffusion model for protein conformational ensemble generation. *PRX Life*, 3: 023013, Jun 2025b. doi: 10.1103/PRXLife.3.023013.

Liu, Z., Casey, T. M., Blackburn, M. E., Huang, X., Pham, L., de Vera, I. M. S., Carter, J. D., Kear-Scott, J. L., Veloro, A. M., Galiano, L., and Fanucci, G. E. Pulsed epr characterization of hiv-1 protease conformational sampling and inhibitor-induced population shifts. *Phys. Chem. Chem. Phys.*, 18:5819–5831, 2016. doi: 10.1039/C5CP04556H.

Maddipatla, A., Sellam Bojan, N., Bojan, M., Masalitin, V., Vedula, S., Schanda, P., Marx, A., and Bronstein, A. M. Experiment-guided AlphaFold3 resolves measurement-consistent protein ensembles. *Nature Biotechnology*, 2026. ISSN 1546-1696.

Maddipatla, S. A., Bojan, N., Bojan, M., Vedula, S., Schanda, P., Marx, A., and Bronstein, A. Inverse problems with experiment-guided AlphaFold. In *Forty-second International Conference on Machine Learning*, 2025.

McIntosh-Smith, S., Alam, S. R., and Woods, C. Isambard-AI: A leadership class supercomputer optimised specifically for artificial intelligence, 2024.

Neal, R. M. Annealed importance sampling. *Statistics and Computing*, 11:125–139, 2001.

Passaro, S., Corso, G., Wohlwend, J., Reveiz, M., Thaler, S., Somnath, V. R., Getz, N., Portnoi, T., Roy, J., Stark, H., Kwabi-Addo, D., Beaini, D., Jaakkola, T., and Barzilay, R. Boltz-2: Towards accurate and efficient binding affinity prediction. *bioRxiv*, 2025.

Potaptchik, P., Saravanan, A., Mammadov, A., Prat, A., Albergo, M. S., and Teh, Y. W. Meta flow maps enable scalable reward alignment. In *Forty-third International Conference on Machine Learning*, 2026.

Ren, Y., Gao, W., Ying, L., Rotskoff, G. M., and Han, J. Driftlite: Lightweight drift control for inference-time scaling of diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Sabour, A., Albergo, M. S., Domingo-Enrich, C., Boffi, N. M., Fidler, S., Kreis, K., and Vanden-Eijnden, E. Test-time scaling of diffusions with flow maps, 2025.

Sani, M. M., Jamali, N., Jalali, M., and Farnia, F. Training-free distribution adaptation for diffusion models via maximum mean discrepancy guidance, 2026.

Santambrogio, F. *Optimal Transport for Applied Mathematicians*. Birkhäuser/Springer, 2015.

Santi, R. D., Vlastelica, M., Hsieh, Y.-P., Shen, Z., He, N., and Krause, A. Flow density control: Generative optimization beyond entropy-regularized fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Saravanan, A., Knowles, R., Kerrigan, G., and Rainforth, T. DiffBED: Scaling bayesian experimental design to high-dimensions. In *The Fourteenth International Conference on Learning Representations*, 2026.

Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., and Ranganath, R. A general framework for inference-time scaling and steering of diffusion models. In *Forty-second International Conference on Machine Learning*, 2025.

Skreta, M., Akhound-Sadegh, T., Ohanesian, V., Bondesan, R., Aspuru-Guzik, A., Doucet, A., Brekelmans, R., Tong, A., and Neklyudov, K. Feynman-Kac correctors in diffusion: Annealing, guidance, and product of experts. In *Forty-second International Conference on Machine Learning*, 2025.

So, O., Karrer, B., Fan, C., Chen, R. T. Q., and Liu, G.-H. Discrete adjoint matching. In *The Fourteenth International Conference on Learning Representations*, 2026.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Thornton, J., Béthune, L., Zhang, R., Bradley, A., Nakkiran, P., and Zhai, S. Controlled generation with distilled diffusion energy models and sequential Monte Carlo. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

Vinograd, G., Achituve, I., and Fetaya, E. Diverse sampling in diffusion models with marginal preserving particle guidance, 2026.

Wang, Z., Santi, R. D., Mo, X., Zavlanos, M. M., Krause, A., and Johansson, K. H. Efficient tail-aware generative optimization via flow model fine-tuning. In *ICLR 2026 Workshop on Foundation Models for Science: Real-World Impact and Science-First Design*, 2026.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., and Baker, D. De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100, 2023. ISSN 1476-4687.

Wohlwend, J., Corso, G., Passaro, S., Getz, N., Reveiz, M., Leidal, K., Swiderski, W., Atkinson, L., Portnoi, T., Chinn, I., Silterra, J., Jaakkola, T., and Barzilay, R. Boltz-1: Democratizing biomolecular interaction modeling. *bioRxiv*, 2024.

Wu, L., Trippe, B. L., Naesseth, C. A., Cunningham, J. P., and Blei, D. Practical and asymptotically exact conditional sampling in diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., and Dong, Y. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Conference on Neural Information Processing Systems*, 2023.

Appendix

The Appendix is structured in the following way. In Appendix A, we discuss related work and required background information. In Appendix B, we provide additional details regarding implementation of the mean-field steering procedure. In Appendix C, we provide details of the experimental settings, along with additional experimental results. In Appendix D, we provide the proofs of the results stated in the main body. Finally, Appendix E includes the licenses for assets used.

A Background

A.1 Related Work

Pointwise-Reward Gradient-Steering Methods Since the development of diffusion and flow-based generative models, there has been an extensive literature concerning incorporating additional steering terms into the inference dynamics to encourage desirable behaviour. Important examples include classifier-guidance (Dhariwal & Nichol, 2021), diffusion posterior sampling (Chung et al., 2023), and universal guidance (Bansal et al., 2024) to name a few. While effective, simply incorporating gradient terms comes with few theoretical guarantees on the generated distribution, so a more recent line of works has incorporated Feynman-Kac reweighting procedures to correct the dynamics to trace out a prescribed path of measures (Wu et al., 2023; Thornton et al., 2025; Singhal et al., 2025; Skreta et al., 2025; He et al., 2026; Sabour et al., 2025). Our presentation largely follows that in Sabour et al. (2025), though they focus on using a flow-map as their $\hat{x}_1$ prediction. Recent works Potaptchik et al. (2026); Holderrieth et al. (2026) also consider inference-time steering according to pointwise rewards, but sidestep the issues of requiring reweighting schemes by using a stochastic flow map to estimate the value function under a stochastic optimal control framework.

Batch-Reward Gradient-Steering Methods Recently there have been a number of works that have used steering terms that incorporate particle interactions in order to encourage desirable distributional properties in the generated samples. Similarly to the pointwise setting, these approaches are effective but largely heuristic in nature, and come with a weak understanding of the sampled distribution. In fact, many of these methods use gradient-steered dynamics; our framework can thus be interpreted as providing a theoretical grounding for such methods through our derived reweighting scheme.

Many of these works focus on steering to encourage diversity in the generated samples. For a sequence of kernels k_t, the *Particle Guidance* method of Corso et al. (2024) modifies the inference dynamics to be

$$\mathrm{d}X_t^i = \big[b_t(X_t^i) + \varepsilon_t s_t(X_t^i) - \varepsilon_t \nabla_{X_t^i} \sum_{j=1}^N k_t(X_t^i, X_t^j)\big]\mathrm{d}t + \sqrt{2\varepsilon_t}\mathrm{d}B_t^i. \tag{16}$$

which corresponds to gradient-steering for a batch-reward function $\mathcal{R}_t^N(x^{1:N}) = -\frac{1}{2N}\sum_{i,j=1}^N k_t(x^i, x^j)$. As commented in Corso et al. (2024), this does not in general sample from μ_1^* as the dynamics do not correspond to a reversed diffusion process. To correct for this, Corso et al. (2024) also introduce a *learned* potential steering approach, but this requires additional training and thus falls under the fine-tuning paradigm (see below) rather than the inference-time steering paradigm that we consider in this work. Recent work by Lam et al. (2026) from the computational biology literature similarly incorporates the gradient of biasing potentials into the inference dynamics, including a diversity-encouraging potential $E_i = \frac{k}{N-1}\sum_{i\neq j}\exp\big(-\frac{(\mathrm{CV}_i - \mathrm{CV}_j)^2}{2\sigma^2}\big)$ defined on chosen collective variables. Kirchhof et al. (2025) include a *sparse-repellency* term in the inference dynamics, that steers the generations out of shielded areas. While similar in spirit to particle guidance (in that the base dynamics are modified to include a repulsive interaction term), the control terms cannot be directly interpreted as gradients of a potential. Their steering terms are 'sparse' in the sense that the base dynamics are only modified where necessary to avoid shielded areas. Finally, we also note the concurrent works Azangulov et al. (2026); Vinograd et al. (2026) that have also targeted principled techniques for distributional steering, under the particular setting of encouraging diversity. While our proposed framework does provide a theoretically-principled approach for such diversity steering, we remark that our specific reweighting procedure is somewhat ill-suited to the diversity setting due to its reliance on resampling, as whenever a particle is duplicated then this will generally

be bad for diversity. We anticipate that combining with approaches to mitigate weight changes, such as Ren et al. (2026), can help to address this, and we believe this is a valuable direction for future work.

Batch-level gradient steering has also been considered beyond the diversity setting. Such works include Maddipatla et al. (2025; 2026), who consider steering the Protenix diffusion model (ByteDance et al., 2025) according to experimental observations that provide distributional information. These works formulate the task as an inverse problem in which one aims to sample from the ensemble's posterior distribution $p(\mathcal{X}|a, y) \propto p(y|\mathcal{X}, a) \cdot p(\mathcal{X}|a)$; here, $\mathcal{X} = \{X^1, ..., X^N\}$ is the sampled ensemble, $p(y|\mathcal{X}, a)$ denotes a chosen likelihood function, and the prior $p(\mathcal{X}|a)$ is given by the base diffusion model (conditioned on amino acid sequence a). This corresponds to choosing a log-likelihood reward function $\mathcal{R}(\mu) = \log p(y|\mu, a)$ in our framework. Sani et al. (2026) consider steering according to an MMD-based reward $\mathcal{R}(\hat{\mu}) = -\text{MMD}^2(\mu, \nu)$ for a reference dataset ν, similar to several of our experimental settings. DiffBED (Saravanan et al., 2026) steers according to the expected information gain $\mathcal{R}(\hat{\mu}) = \text{EIG}(\hat{\mu})$, for applications in Bayesian experimental design.

The aim of this present work is to show how gradient-based steering for distributional rewards (like in the examples discussed above) can be placed on a theoretically grounded footing using a Feynman-Kac reweighting procedure, analogous to how FK-approaches have placed pointwise gradient steering on a principled foundation. We summarise this positioning of our work within the surrounding pointwise-reward and distributional-reward literature in Table 1.

	Heuristic Gradient Steering	**Principled Targeting of μ^***
Pointwise Reward $r(X_1)$	$\nabla r(X_t)$ steering (e.g. Dhariwal & Nichol (2021); Chung et al. (2023); Bansal et al. (2024))	Reweighting approaches (e.g. Wu et al. (2023); Singhal et al. (2025); Skreta et al. (2025); He et al. (2026); Sabour et al. (2025))
Distributional Reward $\mathcal{R}(\mu)$	$\nabla_{X_i} \mathcal{R}(\hat{\mu})$ steering (e.g. Corso et al. (2024); Maddipatla et al. (2025); Sani et al. (2026); Lam et al. (2026); Saravanan et al. (2026))	**Mean-Field Steering (Ours)**

Table 1: Positioning of our work within the pointwise-reward and distributional-reward steering literatures.

Fine-tuning Flow-based Generative Models In this work, we consider the inference-time steering literature for controlling diffusion models. A separate but related literature concerns *fine-tuning* diffusion models. Similarly to inference-time steering, such methods aim to improve the quality of the generations according to a chosen reward function. However, unlike inference-time steering methods, these approaches involve additional training of the generative model, and must be re-trained for different reward functions.

For the pointwise-reward case, such methods are now widespread; prominent examples of such methods include DRaFT (Clark et al., 2024), ReFL (Xu et al., 2023), and DDPO (Black et al., 2024), to name a few. Adjoint Matching (Domingo-Enrich et al., 2025) provides a scalable method for finetuning diffusion models from the perspective of stochastic optimal control (SOC), and has subsequently seen extensions to sampling, discrete settings, and wider SOC problems (Havens et al., 2025; Liu et al., 2025a; So et al., 2026; Howard et al., 2026). A recent line of works Santi et al. (2026); Wang et al. (2026) have considered fine-tuning flow-based models according to measure-valued rewards, thus considering a similar setting to the one studied in this work but under the fine-tuning paradigm rather than inference-time steering.

A.2 Functional derivatives

In this section, we briefly recall the notion of a functional derivative with respect to a probability measure. For a complete and rigorous reference, we refer to Santambrogio (2015) for an accessible introduction.

Table 2: Examples of first variations of functionals on probability measures. Functional derivatives on probability spaces are defined up to additive constants.

Name	**Functional** $\mathcal{F}(\mu)$	**Functional derivative** $\frac{\delta \mathcal{F}}{\delta \mu}$
Linear reward	$\int f(x)\,\mu(dx)$	$f(x)$
Non-linear moment reward	$F\left(\int \varphi(x)\,\mu(dx)\right)$	$F'\left(\int \varphi(y)\,\mu(dy)\right)\varphi(x)$
Quadratic moment reward	$\left(\int \varphi(x)\,\mu(dx)\right)^2$	$2\left(\int \varphi(y)\,\mu(dy)\right)\varphi(x)$
Squared Maximum Mean Discrepancy (MMD^2)	$\iint k(x,y)\,(\mu-\nu)(dx)(\mu-\nu)(dy)$	$2\int k(x,y)\,(\mu-\nu)(dy)$
Entropy	$\int \rho_\mu(x)\log\rho_\mu(x)\,dx$	$\log\rho_\mu(x)+1$
Quadratic Wasserstein cost to fixed ν	$\frac{1}{2}W_2^2(\mu,\nu)$	Kantorovich potential $\phi(x)$ from μ to ν for cost $\frac{1}{2}\|x-y\|^2$
Entropic transport cost to fixed ν	$\mathrm{OT}_\varepsilon(\mu,\nu)$	Schrödinger potential $\varphi^\varepsilon(x)$

Let $\mathcal{F}:\mathcal{P}(\mathbb{R}^d)\to\mathbb{R}$ be a functional on probability measures. We say that a measurable function

$$\frac{\delta \mathcal{F}}{\delta \mu}(x,\mu):\mathbb{R}^d\times\mathcal{P}(\mathbb{R}^d)\to\mathbb{R}$$

is a *functional derivative* (or *first variation*) of $\mathcal{F}$ at μ if, for measures ν close to μ,

$$\mathcal{F}(\nu)-\mathcal{F}(\mu)=\int_{\mathbb{R}^d}\frac{\delta \mathcal{F}}{\delta \mu}(x,\mu)\,(\nu-\mu)(dx)+o(\nu-\mu). \tag{17}$$

Informally, $\frac{\delta \mathcal{F}}{\delta \mu}(x,\mu)$ measures the first-order change in $\mathcal{F}$ when one adds a small amount of mass near the point x. Since $\nu-\mu$ has total mass zero, the functional derivative is only defined up to an additive constant depending on μ. In most formulas this ambiguity is harmless. A convenient equivalent characterisation is obtained by considering perturbations of the form

$$\mu_\varepsilon=(1-\varepsilon)\mu+\varepsilon\nu,$$

for $\varepsilon>0$ small enough. Then $\frac{\delta \mathcal{F}}{\delta \mu}$ is characterised by

$$\frac{d}{d\varepsilon}\mathcal{F}(\mu_\varepsilon)\bigg|_{\varepsilon=0}=\int_{\mathbb{R}^d}\frac{\delta \mathcal{F}}{\delta \mu}(x,\mu)\,(\nu-\mu)(dx). \tag{18}$$

Examples In Table 2 we record the functional derivatives of common measure-defined functionals $\mathcal{F}$ that could be used under our framework (typically one would use $\mathcal{R}=-\mathcal{F}$ as a reward).

B Implementation Details

In this section, we provide additional implementation details for the mean-field steering procedure. Recall from Section 3.3 that we make a finite-particle approximation to the idealised weighted McKean-Vlasov dynamics (12). For N particles, letting $\hat{\mu}_t=\sum_{i=1}^N w_t^i\delta_{X_t^i}$ for weights $w_t^i\propto e^{A_t^i}$, these dynamics are given by

$$\mathrm{d}X_t^i=\left[b_t(X_t^i)+\varepsilon_t s_t(X_t^i)+\varepsilon_t\nabla_x\Psi_t(X_t^i;\hat{\mu}_t)\right]\mathrm{d}t+\sqrt{2\varepsilon_t}\,\mathrm{d}B_t^i, \qquad X_0^i\sim p_0 \tag{19a}$$

$$\mathrm{d}A_t^i=\left[b_t(X_t^i)\cdot\nabla_x\Psi_t(X_t^i;\hat{\mu}_t)+\dot{\Psi}_t(X_t^i)\right]\mathrm{d}t, \qquad A_0^i=0. \tag{19b}$$

We now provide additional details regarding computation of the steering terms.

Algorithm 2: Mean-Field Inference-Time Steering for Diffusion Models (Multi-Batch)

Input: Number of batches M, number of particles per batch N, distributional reward function $\mathcal{R}$ and corresponding first variation Ψ_t, time increment δt or alternative time schedule.
Initialisation: Initialise an empty set of completed samples and log-weights $\mathcal{H}_0 = \emptyset$.
For $m \in \{1, \dots, M\}$:

Initialise batch: Sample $X_0^{m,1:N} \sim p_0^{\otimes N}$, initialise log-weights $A_0^{m,i} = 0$.
For $t \in \{0, \delta t, 2\delta t, \dots, 1 - \delta t\}$:

- **Construct combined terminal measure:**
  - Let $\mathcal{H}_{m-1} = \{(\bar{X}_1^\ell, \bar{A}_1^\ell)\}_{\ell=1}^{(m-1)N}$ denote the retained samples and log-weights from previous batches.
  - Define $\hat{\mu}_1^{m,N} = \sum_{\ell=1}^{(m-1)N} \bar{w}_1^\ell \delta_{\bar{X}_1^\ell} + \sum_{j=1}^{N} w_t^{m,j} \delta_{\hat{x}_1(X_t^{m,j}, t)}$, where $\bar{w}_1^\ell, w_t^{m,j} \propto e^{\bar{A}_1^\ell}, e^{A_t^{m,j}}$, with normalisation over both previously sampled and denoised current particles.
- **Positional updates:**
  - Perform discretisation step for positional updates,
    $$X_{t+\delta t}^{m,i} = X_t^{m,i} + \big[b_t(X_t^{m,i}) + \varepsilon_t s_t(X_t^{m,i}) + \varepsilon_t \nabla_x \Psi_t(X_t^{m,i}; \hat{\mu}_t^{m,N})\big]\delta t + \sqrt{2\varepsilon_t}\, \delta B_t^{m,i},$$
    where, by slight abuse of notation, using $\hat{\mu}_t^{m,N}$ denotes using the combined terminal measure approximation induced by $\hat{\mu}_1^{m,N}$ when evaluating the reward terms.
- **Log-weight updates:**
  - Compute $\frac{\mathrm{d}}{\mathrm{d}t}\Psi_t(\cdot; \hat{\mu}_t^{m,N})$ at points $X_t^{m,i}$.
  - Compute mean-field FK potentials
    $$g_t^{m,i} = b_t(X_t^{m,i}) \cdot \nabla_x \Psi_t(X_t^{m,i}; \hat{\mu}_t^{m,N}) + \frac{\mathrm{d}}{\mathrm{d}t}\Psi_t(X_t^{m,i}; \hat{\mu}_t^{m,N}).$$
  - Perform discretisation step for log-weight updates, $A_{t+\delta t}^{m,i} = A_t^{m,i} + g_t^{m,i}\delta t$.
- **Resample:** Optionally resample according to current weights $w_t^{m,i} \propto e^{A_t^{m,i}}$.

Update completed set: Update completed set with the completed batch samples and corresponding log-weights
$$\mathcal{H}_m = \mathcal{H}_{m-1} \cup \{(X_1^{m,i}, A_1^{m,i})\}_{i=1}^{N}.$$

Return: Weighted particles (X_1^i, w_1^i).

Gradient-Steering Term $\nabla_x \Psi_t(x; \mu_t)$ As explained in the main text, the gradient-steering term $\nabla_x \Psi_t(x; \mu_t)$ appearing in the positional updates can be computed directly with autodifferentiation, evaluated on the current empirical measure $\hat{\mu}_t$. When one can analytically compute the first variation (as one can do, for example, for functionals in Table 2), then one can use a finite-sample approximation $\Psi_t(X_t^i, \hat{\mu}_t)$ and then differentiate with respect to the first argument. Alternatively, one can directly differentiate $\mathcal{R}_t(\hat{\mu}_t)$ evaluated on the empirical measure with respect to the positions X_t^i and rescale according to the particle weights, giving $(w_t^i)^{-1}\nabla_{X_t^i}\mathcal{R}_t(\hat{\mu}_t)$ as the i-th steering term. Note that in the first case, the derivative with respect to the position is passed solely through the position x with the measure μ kept fixed, while in the second the dependence on x is passed through both—this is an algorithmic choice, as under standard smoothness assumptions these both converge to the desired mean-field steering term $\nabla_x \Psi_t(x; \mu_t)$ in the large-N limit – this follows from arguments similar to those in the proof of Theorem D.5. In practice we found both choices to behave very similarly, and so use the first approach in our experiments.

Time-Derivative Term $\dot{\Psi}_t(x)$ The more challenging term in the dynamics is the time-derivative $\dot{\Psi}_t(x) = \frac{\mathrm{d}}{\mathrm{d}t}\big(\Psi_t(x, \mu_t)\big)$, in which the position x is kept fixed but the time-dependence arises through both Ψ_t and the measure evolution μ_t (note that the dependence of $\dot{\Psi}_t(x)$ on μ_t is suppressed for notational convenience). In contrast to the standard case, this induces *implicit* behaviour in the log-weight updates—their dynamics involve the change in the measure μ_t, but the measure evolution is itself dependent on the change in the weights. We provide two different methods to overcome this.

The first is mathematically more explicit, building a tractable linear system of size N which can be solved cheaply. The second considers a simple fixed-point iterative procedure to approximate the weight updates. We outline both below.

For the implicit expression in (13), all terms are generally tractable (in the proof of Theorem 3.3, we use integration-by-parts to remove troublesome divergence terms on the score function). In many of the most standard cases (such as MMD, and diversity kernel rewards, for example) the second variation has an analytical expression. In other cases, one can take derivatives with respect to the particle weights for the finite sample approximations (which is cheap, as these are scalar). Note that the integral in (13) is linear in $\dot{\Psi}_t(y)$, so when evaluated using the current empirical measure this gives a linear system for the terms $\dot{\Psi}_t(X_t^i)$ with tractable coefficients, which can be solved cheaply. The resulting procedure is outlined in Algorithm 3.

As a simpler alternative, one can instead use a small fixed-point iterative procedure based on finite differences. For example, one can first approximate using $\dot{\Psi}_t(x) \approx \frac{\Psi_{t+\delta t}(x, \tilde{\mu}_{t+\delta t}(w_t)) - \Psi_t(x, \mu_t)}{\delta t}$ where $\tilde{\mu}_{t+\delta t}(w_t) = \sum_j w_t^j \delta_{X_{t+\delta t}^j}$, then use this to approximate the weights $w_{t+\delta t}^j$. These new weights can be used as a new approximation to the next measure $\mu_{t+\delta t}$, and repeating the iterations a few times quickly gives accurate weights updates. In most settings this iterative procedure adds little computational overhead, as the expensive parts of the reward computations are usually independent of the weights (indeed, the expensive part is usually evaluating the denoiser estimates). For strong tilt strengths, we sometimes found it necessary to used damped Picard iterations, though for more reasonable tilt-strengths this was not necessary. We outline this fixed-point procedure in Algorithm 4.

We found that the choice of solver for $\dot{\Psi}_t(x)$ had little effect on the empirical performance of the method, which we verify in Appendix C.1. We therefore used the fixed-point iterative solver in our higher-dimensional experiments for simplicity, but note that practitioners might find the first approach useful in certain settings.

Details on the finite-difference computation As mentioned above, the simplest finite difference implementation would be to use $\dot{\Psi}_t(x) \approx \frac{\Psi_{t+\delta t}(x, \tilde{\mu}_{t+\delta t}(w_t)) - \Psi_t(x, \mu_t)}{\delta t}$. While simple, this does come with the downside that it involves evaluating the network on X_t at the time $t + \delta t$. It is in fact possible to avoid any such time-offset by using a combination of autodifferentiation and finite-differences, as we now outline.

We are considering rewards of the form $\mathcal{R}_t(\mu_t) = \alpha(t)\mathcal{R}(\hat{x}_1(t, \cdot)\#\mu)$, which involves projecting to the current predicted endpoint. We can therefore write the first variation of the form $\Psi_t(x, \mu) = \alpha(t)\, \psi\big(\hat{x}_1(t, x), \hat{x}_1(t, \cdot)\#\mu\big)$ for some ψ. Using the shorthand notation $\theta_i = \hat{x}_1(t, X_t^i)$ and $\hat{\pi}_t = \hat{x}_1(t, \cdot)\#\mu$, we can write the following approximation

$$\dot{\Psi}_t(X_t^i) \approx \frac{\alpha(t+\delta t)\, \psi\big(\theta_i, \hat{\pi}_{t+\delta t}^{(k)}\big) - \alpha(t)\, \psi(\theta_i, \hat{\pi}_t)}{\delta t} + \alpha(t)\, \frac{d}{ds}\, \psi\big(\hat{x}_1(s, x), \hat{\pi}_t\big)\Big|_{s=t} \tag{20}$$

In this expression, the first term is a finite-difference approximation that includes the dependence on the time-evolution of μ_t, and the second term can be implemented easily using autodifferentiation. By implementing in this manner, one avoids evaluating the network at any points with a time-offset. We use this implementation in the protein experiments.

Multi-Batch Algorithm We outline the inference procedure for a single batch of N particles in Algorithm 1, which will approximate the target tilted measure μ^* as the batch size N goes to infinity. In practice, the batch size is limited by computational constraints; we therefore also provide a multi-batch procedure in Algorithm 2. Specifically, one can simulate multiple batches of N particles, and retain the generated samples along with the obtained weights. For each subsequent batch simulation, we evaluate the steering terms using the combined measure of the previously generated samples (with their respective weights) along with the particles currently being simulated. The procedure for updating the positions and log-weights of the current simulated particles remains unchanged.

Additional Steering Terms We remark here that one can also include additional steering terms in the positional updates, if one accounts for these in the log-weight dynamics. In particular, if one includes an additional $u_t(X_t; \mu_t)$ term as follows,

$$\mathrm{d}X_t = \big[b_t(X_t) + \varepsilon_t s_t(X_t) + \varepsilon_t \nabla_x \Psi_t(X_t; \mu_t) + u_t(X_t; \mu_t)\big]\mathrm{d}t + \sqrt{2\varepsilon_t}\, \mathrm{d}B_t, \tag{21}$$

then the necessary log-weight dynamics become

$$\begin{aligned}\mathrm{d}A_t = \big[&b_t(X_t)\cdot\nabla_x\Psi_t(X_t;\mu_t) + \dot{\Psi}_t(X_t) + \nabla_x\cdot u_t(X_t;\mu_t)\\ &+ u_t(X_t;\mu_t)\cdot(s_t(X_t)+\nabla_x\Psi_t(X_t;\mu_t))\big]\mathrm{d}t.\end{aligned} \tag{22}$$

In particular, a natural choice might be to add in additional gradient steering by taking $u_t(X_t;\mu_t) = \chi_t\nabla_x\Psi_t(X_t;\mu_t)$ for a coefficient function χ_t (similar to the presentation in Sabour et al. (2025)). In this case, the dynamics become

$$\mathrm{d}X_t = \big[b_t(X_t) + \varepsilon_t s_t(X_t) + \varepsilon_t\nabla_x\Psi_t(X_t;\mu_t) + \chi_t\nabla_x\Psi_t(X_t;\mu_t)\big]\mathrm{d}t + \sqrt{2\varepsilon_t}\,\mathrm{d}B_t, \tag{23a}$$

$$\begin{aligned}\mathrm{d}A_t = \big[&b_t(X_t)\cdot\nabla_x\Psi_t(X_t;\mu_t) + \dot{\Psi}_t(X_t)\\ &+ \chi_t\big(\|\nabla_x\Psi_t(X_t;\mu_t)\|^2 + \Delta_x\Psi_t(X_t;\mu_t) + s_t(X_t)\cdot\nabla_x\Psi_t(X_t;\mu_t)\big)\big]\mathrm{d}t.\end{aligned} \tag{23b}$$

These dynamics are again analogous to those in the corresponding pointwise case (see Sabour et al. (2025)). Note too that there is no additional implicit dependence on the measure evolution; this is all captured by the terms already present in the standard dynamics (12).

In this work, we used the dynamics (12) in our experiments. We note, however, that a possible limitation of this approach is that it can only redistribute mass among regions populated by existing particles. If the target tilt requires adding mass in an unpopulated area, this would not be possible by reweighting alone. To mitigate this issue, one could include additional steering into the positional updates using the above dynamics, to help ensure that there are particles covering the support of the target measures that can then be reweighted appropriately.

Optional Poisson Equation Solver To combat weight degeneracy in Feynman-Kac steering inference schemes, recent work Ren et al. (2026) subsequently approximately solves the Poisson equation $\nabla\cdot(\rho_t u_t) = -\rho_t g_t$ for an additional drift term u_t. By solving this Poisson equation, the resulting update offloads the changes in log-weights into changes in position. Ren et al. (2026) parameterise u_t with a small set of basis functions, and approximately solve the Poisson equation by solving a resulting linear system. We remark here that this method can also be directly applied in our setting. The exact parameterisation can be chosen by the user; for example, given that we are already considering an interacting particle system, one could consider parameterising the additional drift to be of form $u_t(x) = \sum_j \alpha_j\nabla_x k(x, X_t^j)$, for a suitably chosen kernel k. We leave investigating combining our approach with that of Ren et al. (2026) to future work; in particular, we anticipate that reducing the changes in the weights could enable applications for diversity steering, for which a resampling procedure is not well-suited.

Relation to Pointwise Rewards Recall from Section 3.3 that our framework recovers the standard pointwise steering approach outlined in Section 2 when using the reward $\mathcal{R}(\mu) = \int r(x)\mathrm{d}\mu(x)$. Given this, one might ask whether a pointwise reward could simply be used in place of a distributional one. We emphasise that the two settings solve fundamentally different problems. As an illustration, consider what happens as the reward strength λ grows towards infinity. In the pointwise case, the tilted distribution becomes increasingly concentrated on the single point that maximises r. In the distributional case, the tilt pushes μ towards the maximiser of the reward functional; for a divergence reward $D(\mu,\nu)$, this means μ approaches the entire target measure ν rather than any individual point. More fundamentally, matching a prescribed target measure ν via a pointwise reward would require choosing $r(x) = \log(\nu(x)/p_1(x))$, and hence require explicit access to the base density p_1 which is typically not available. Distributional rewards therefore enable forms of control that are not naturally expressible in the pointwise setting.

C Experimental Details and Additional Results

C.1 Low-dimensional experiments

Experimental setting For the base model, we train a simple flow-matching model to sample from a bimodal Gaussian mixture base distribution p_1, with means $(-1.0, 1.0)$, both modes with unit variance, and mode weights $(1, 3)$ respectively. To construct a distributional reward, we consider tilting the model's output distribution towards a similar Gaussian mixture ν but now with respective

Algorithm 3: Implicit Linear System $\dot{\Psi}_t$ Solver

Input: Weighted particles $(X_t^i, w_t^i)_{i=1}^N$, functions b_t, $\partial_t \Psi_t$, $\nabla_y \Psi_t$, $\tilde{\Phi}_t$, $\nabla_y \tilde{\Phi}_t$.
Output: Values $\dot{\psi}_i \approx \dot{\Psi}_t(X_t^i)$.
Set $\hat{\mu}_t = \sum_{i=1}^N w_t^i \delta_{X_t^i}$.
For each $i, j \in \{1, \dots, N\}$, set $\phi_{ij} = \tilde{\Phi}_t(X_t^i, X_t^j, \hat{\mu}_t)$, $g_{ij} = \nabla_y \tilde{\Phi}_t(X_t^i, X_t^j, \hat{\mu}_t)$, $b_j = b_t(X_t^j)$, and $h_j = \nabla_y \Psi_t(X_t^j, \hat{\mu}_t)$.
For each $i, j \in \{1, \dots, N\}$, set $A_{ij} = \delta_{ij} - w_t^j \phi_{ij}$.
For each $i \in \{1, \dots, N\}$, set $r_i = \partial_t \Psi_t(X_t^i, \hat{\mu}_t) + \sum_{j=1}^N w_t^j \big(b_j \cdot g_{ij} + \phi_{ij} b_j \cdot h_j\big)$.
Solve $A\dot{\psi} = r$ for $\dot{\psi} = (\dot{\psi}_1, \dots, \dot{\psi}_N)^\top$.
return $(\dot{\psi}_i)_{i=1}^N$.

Algorithm 4: Fixed-Point $\dot{\Psi}_t$ Solver

Input: Particles and log-weights $(X_t^i, A_t^i)_{i=1}^N$, next-step particles $(X_{t+\delta t}^i)_{i=1}^N$, b_t, $\nabla_x \Psi_t$, step size δt, iterations K_{FP}, optional damping parameter $\alpha \in (0, 1]$.
Output: Values $\dot{\psi}_i \approx \dot{\Psi}_t(X_t^i)$ and updated weights $w_{t+\delta t}^i$.
Set $\hat{\mu}_t = \sum_{i=1}^N w_t^i \delta_{X_t^i}$, and $a_i^{(0)} = A_t^i$, for $i = 1, \dots, N$
for $k = 0, \dots, K_{FP} - 1$ **do**
    Construct candidate weighted next-step measure $\hat{\mu}_{t+\delta t}^{(k)} = \sum_{j=1}^N w_j^{(k)} \delta_{X_{t+\delta t}^j}$.
    Compute finite differences $\dot{\psi}_i^{(k)} = \big(\Psi_{t+\delta t}(X_t^i, \hat{\mu}_{t+\delta t}^{(k)}) - \Psi_t(X_t^i, \hat{\mu}_t)\big)/\delta t$ (or using (20)).
    Update candidate log-weights updates $\tilde{a}_i^{(k+1)} = A_t^i + \delta t \big[\dot{\psi}_i^{(k)} + b_t(X_t^i) \cdot \nabla_x \Psi_t(X_t^i, \hat{\mu}_t)\big]$.
    Update next-step weights as $a_i^{(k+1)} = (1 - \alpha) a_i^{(k)} + \alpha\, \tilde{a}_i^{(k+1)}$.
    Construct updates next-step weights $w_i^{(k+1)} \propto e^{a_i^{(k+1)}}$.
end for
Set $\dot{\psi}_i = \dot{\psi}_i^{(K_{FP}-1)}$ and $w_{t+\delta t}^i = w_i^{(K_{FP})}$, for $i = 1, \dots, N$.
Return: Final iterations $(\dot{\psi}_i, w_{t+\delta t}^i)_{i=1}^N$.

weights $(3, 1)$. For the results in the main paper, we use the squared MMD reward

$$\mu^* = \arg\min_{\mu} \big\{ \mathrm{KL}(\mu \| p_1) + \lambda^* \, \mathrm{MMD}^2(\mu, \nu) \big\}. \tag{24}$$

We also demonstrate similar results for other reward functions below. By optimising a Gaussian mixture model, we also obtain a highly accurate approximation to the true tilted solution μ^* from which we aim to sample; the resulting density is plotted in Figure 2 to give a visual indication of how well μ^* has been targeted.

For the reward function, we use a Gaussian radial basis kernel with bandwidth 5.0 (we note that results were robust across different kernel widths, provided they were sufficiently wide to give meaningful gradients). The results in Figure 2 are for $\lambda^* = 10.0$, and again we note that similar behaviour was observed across λ values. We generated 4,000 samples with 100 steps, and for mean-field sampling we resampled using residual resampling every 34 steps for the visualisation in Figure 2, and every 10 steps in the other examples (we observed very similar performance for different resampling schemes). For the results in Figures 2 and 3 we used the fixed-point iterations with no damping, and used 3 fixed-point iterations for the visualisation in Figure 2. To evaluate the KL divergence, we used a leave-one-out estimator on pointclouds, using a Gaussian RBF kernel with bandwidth 0.2. The low-dimensional experiments were run on a single NVIDIA GeForce RTX 2080Ti GPU.

C.1.1 Additional Results

Objective results for different λ values In Figure 3, we showed how the value of the objective optimality gap $J(\mu) - J(\mu^*)$ changes for different steering strengths, for standard gradient steering and mean-field steering. Here, $J(\mu) = \lambda^* \, \mathrm{MMD}^2(\mu, \nu) + \mathrm{KL}(\mu \| p_1)$, and we took $\lambda^* = 10.0$. In

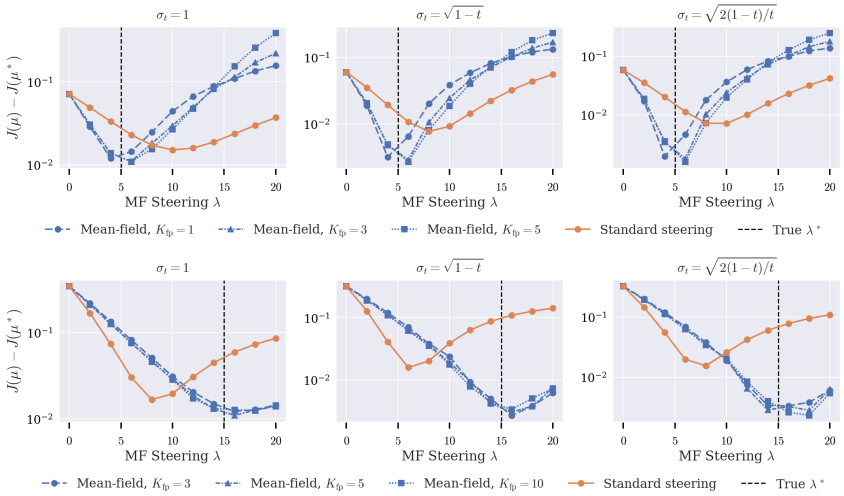


Figure 10: **Low-dimensional MMD Example:** Objective optimality gap plots, similar to Figure 3, for additional λ^* values 5.0 *(top)* and 15.0 *(bottom)*. As in Figure 3, the objective gap is minimised when using mean-field steering with the correct value of λ.

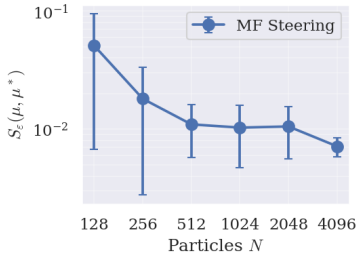


Figure 11: Sinkhorn divergence $\text{SD}_\varepsilon(\mu, \mu^*)$ relative to approximated 'ground-truth' μ^*, for the generated mean-field steering samples using different batch sizes N (mean$\pm$std over 8 runs, using entropic regularisation $\varepsilon = 0.1$).

particular, we saw that the optimum steering strength for mean-field steering coincided with the true λ^*, and that this was consistent across different σ_t schedules, indicating that the mean-field sampling is correctly targeting the tilting measure μ^*. In contrast, the best choice of steering strength for standard gradient steering did not correspond to the λ^* in the objective, and thus would need searching over. In Figure 10, we report the same results for different values of λ^*, and observe a similar pattern.

Comparison of different $\dot{\Psi}_t$ solvers In Section 3.3, we provide two possible approaches for computing the term $\dot{\Psi}_t$, namely a fixed-point procedure (Algorithm 4), and also an implicit linear solver Algorithm 3. In the majority of our experiments, we use the fixed-point solver for simplicity, however here we provide a comparison and discussion regarding the choice of solver.

We generally found the fixed-point solver to converge quickly and accurately. When the tilt strength is not too high, damping in Algorithm 4 is not required, however for large tilt strengths we found it helped to ensure the iterations converged. The implicit linear solver worked reliably in all cases, though its implementation is more involved. We visualise results for standard and large tilt strengths $\lambda \in \{10.0, 40.0\}$ in Figure 12, which reflect the discussion here.

Dependence on N To assess the dependence on the batch size N, we plot the Sinkhorn divergence relative to approximated 'ground-truth' μ^* for different values of N in Figure 11. As expected, the approximation becomes increasingly accurate as N increases, and for large N is very consistent between runs.

Grad	$K_{\text{FP}} = 1$	$K_{\text{FP}} = 4$	$K_{\text{FP}} = 10$	Implicit Linear
1.6	3.9	3.9	3.9	5.8

Table 3: Runtime comparison across steering and solver variants, in the MMD example. Time taken for 100 step simulations (s).

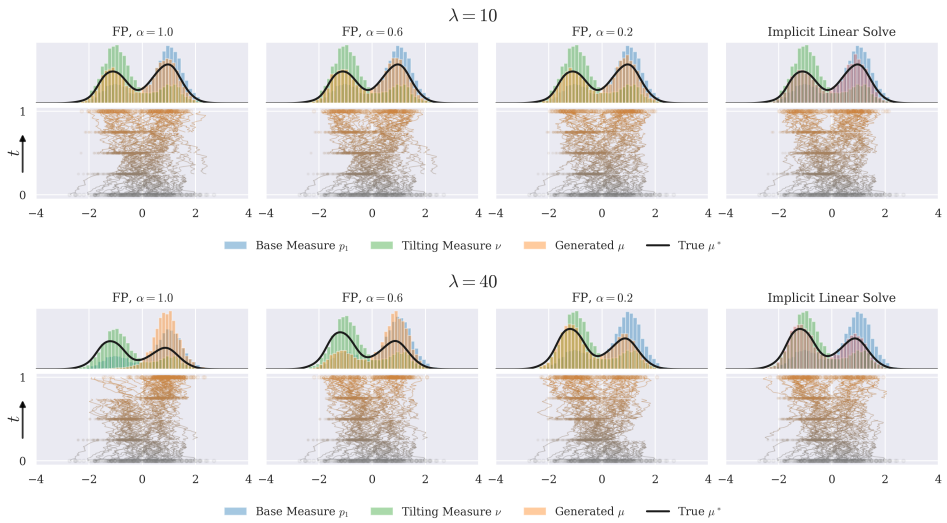


Figure 12: **Comparison of $\dot{\Psi}_t$ solvers, low-dimensional MMD example:** *(Top)* For smaller tilt strength $\lambda = 10.0$, the fixed-point procedure converges accurately for all damping levels. *(Bottom)* A stronger tilt strength necessitates using a larger damping factor in order to converge. *(Right)* The implicit linear solver computes the solution accurately in both cases.

Runtime Analysis In Table 3, we also report runtimes for the different approaches in the MMD example. In this low-dimensional setting, the main additional cost comes from the kernel evaluations required to compute $\dot{\Psi}_t$, which explains the increased inference time relative to gradient steering. However, we can re-use these kernel evaluations across the fixed-point iterations, meaning that subsequent fixed-point iterations do not noticeably increase the sampling time. Generally, we found the implicit linear solver to be slower than the fixed-point approach. We note that in the higher-dimensional cases, the bottleneck increasingly becomes the network evaluation rather than such reward-based computations, meaning the cost of the $\dot{\Psi}_t$ solver becomes less pronounced.

C.1.2 Different Reward Functions

To verify that the same behaviour holds in other settings, we also consider the same experimental setup as above but for some alternative choices of the reward function $\mathcal{R}(\mu)$.

Sinkhorn Divergence The Sinkhorn Divergence SD_ε (Genevay et al., 2018) is a smooth, debiased approximation to the Wasserstein distance between two measures, regularised by an entropic parameter ε. As it is differentiable, we can steer to minimise the objective

$$J(\mu) = \lambda \, \text{SD}_\varepsilon(\mu, \nu) + \text{KL}(\mu \| p_1). \tag{25}$$

We display obtained trajectories and the corresponding objective optimality gaps for this choice of reward in Figure 13, using $\varepsilon = 0.1$ as the regularisation value. Again, $J(\mu)$ is generally minimised by mean-field steering with $\lambda = \lambda^*$, confirming that our procedure is targeting the correct μ^* in this setting too. This is also confirmed visually in the histogram plots.

Entropy We also do similarly using the entropy $\mathcal{R}(\mu) = -\int \log \mu \, \mathrm{d}\mu$. Rather than steering towards a 'tilting' measure ν, such a reward encourages the particles to spread out. We plot the same results as above for this new reward in Figure 14, and observe similar behaviour in this setting too.

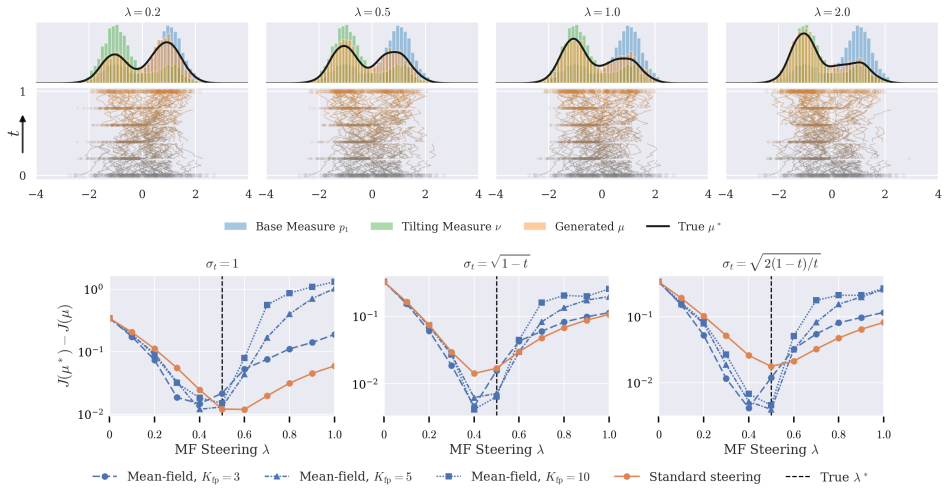


Figure 13: **Low-dimensional Sinkhorn Divergence Example:** *(Top)* Comparison of trajectories obtained using mean-field steering, for increasing λ. The $\dot{\Psi}_t$ use iterations $\{3, 3, 10, 20\}$ with damping parameters $\{1.0, 1.0, 0.6, 0.2\}$ respectively. *(Bottom)* Comparison of the optimality gap $J(\mu) - J(\mu^*)$ for $\lambda^* = 0.5$, for mean-field and gradient-only steering. The gradient-steering coefficient is varied linearly in $[0, 3.0]$.

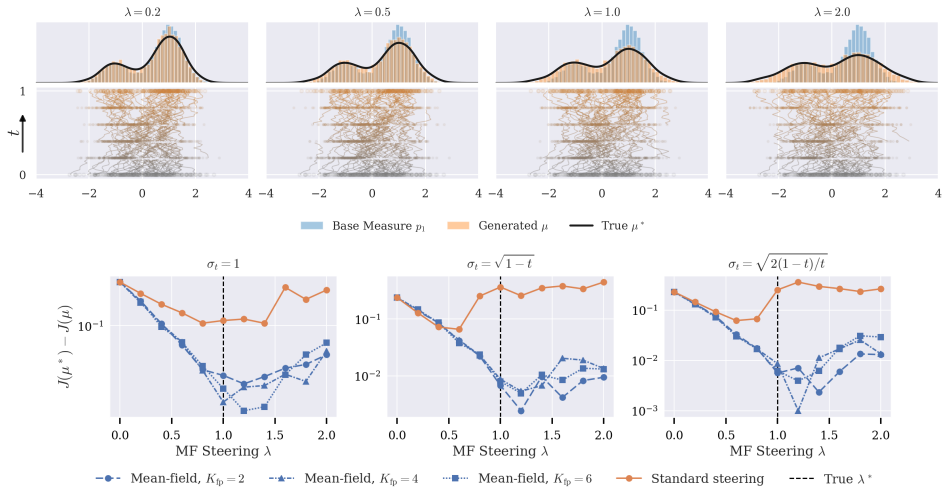


Figure 14: **Low-dimensional Entropy Example:** *(Top)* Comparison of trajectories obtained using mean-field steering, for increasing λ. The $\dot{\Psi}_t$ use iterations $\{3, 3, 5, 10\}$ with damping parameter $\alpha \in \{1.0, 1.0, 1.0, 0.5\}$ respectively.*(Bottom)* Comparison of the optimality gap $J(\mu) - J(\mu^*)$ for $\lambda^* = 1.0$, for mean-field and gradient-only steering.

C.1.3 Gaussian Ground-Truth Validation in Higher Dimensions

Above, we validated that our mean-field steering procedure correctly targets the true μ^* in several $1d$ settings, in which the ground-truth is not available in closed form but can nevertheless be accurately approximated in a tractable manner. We now provide an additional example to validate our method in higher dimensions. Numerical optimisation to obtain an approximation to the ground-truth μ^* is no longer tractable, so we consider a setting in which μ^* is available in closed form. Specifically, we consider a setting where the base diffusion samples a standard Gaussian $p_1 = \mathcal{N}(0, I_d)$, and the reward is defined based on the mean of μ as $\mathcal{R}(\mu) = -\frac{\lambda}{2}\|\mathbb{E}_\mu[X] - a\|_2^2$ for some $a \in \mathbb{R}^d$. In this setting, the ground-truth can be computed analytically and is given in closed form as $\mu^* = \mathcal{N}(\frac{\lambda}{1+\lambda}a, I_d)$.

In Table 4 below, we report the Bures-Wasserstein distance to the ground-truth when $\lambda^* = 1.0$ and $a = \mathbf{1}$. We run the procedure for different values of λ, and as in the existing examples we again see that the value is minimised for the true value $\lambda = \lambda^*$, indicating that our approach again targets the correct μ^* (mean $\pm$ std, over 8 runs). The experiments are run with 100 steps, and 10,000 samples.

Steering λ:	**0.6**	**0.8**	**$\lambda^* = 1.0$**	**1.2**	**1.4**
$d = 5$	0.305 ± 0.009	0.196 ± 0.025	0.150 ± 0.013	0.192 ± 0.017	0.245 ± 0.014
$d = 10$	0.515 ± 0.022	0.379 ± 0.018	0.346 ± 0.014	0.366 ± 0.018	0.429 ± 0.018
$d = 20$	0.830 ± 0.011	0.664 ± 0.021	0.610 ± 0.021	0.625 ± 0.014	0.675 ± 0.028

Table 4: Bures-Wasserstein distance to the ground-truth μ^* corresponding to a true tilting strength $\lambda^* = 1.0$ in the Gaussian example, for different steering strengths λ. The distance is minimised for the correct value $\lambda = \lambda^*$, indicating that μ^* is being targeted correctly.

We also report the dependence on the number of particles N in Table 5, for the same Gaussian example with $d = 10$.

$N = 1{,}000$	**$N = 3{,}000$**	**$N = 10{,}000$**	**$N = 30{,}000$**
0.742 ± 0.073	0.478 ± 0.023	0.330 ± 0.018	0.287 ± 0.016

Table 5: Bures-Wasserstein distance to the ground-truth μ^* corresponding to a true tilting strength $\lambda^* = 1.0$ in the Gaussian example, for different numbers of particles N.

C.2 HIV-1 Protease

Recently, there have been many works that have considered steering diffusion models for protein conformation generation according to experimental observations (see Fadini et al. (2026); Liu et al. (2025b); Levy et al. (2024); Maddipatla et al. (2025; 2026); Li et al. (2026), to name a few). In Section 4.2.1, we first considered an example in which we steer generations for HIV-1 protease according to an experimentally observed distribution over residue distances (Liu et al., 2016). HIV-1 protease is a symmetric homodimer with 99-residues in each monomer, resulting in generating a 1536 atom system giving a full dimensionality of 4608. The system exhibits an important 'flap region', which opens and closes between different conformational states (Figure 4). The double electron-electron resonance (DEER) method can be used to infer proportions across different states by attaching spin-labels to selected residues and approximately measuring the distance between them, aggregated over the conformational ensemble. We will consider the v6 variant of the protease as studied in Liu et al. (2016); this is known from X-ray crystallography and MD modelling to exhibit four different states: *closed*, *semi-open*, *wide-open*, and *tucked*. The DEER data analysed by Liu et al. (2016) suggests that these states are occupied by proportions 21% closed, 61% semi-open, 8% wide-open, and 10% tucked respectively. However, samples from Boltz-2 are strongly biased towards the closed state by comparison, with the states being sampled with proportions 79% closed, 21% semi-open with no tucked or wide-open generations (Figure 6, left). Following the analysis from Liu et al. (2016), we will aim to tilt towards a Gaussian mixture ν defined over residue 55 distances, with relative proportions 25% and 75% (adjusted for removing the tucked and wide-open states), and optimise the objective

$$\mu^* = \arg\min_{\mu} \left\{ \mathrm{KL}(\mu \| p_1) + \lambda \, \mathrm{MMD}^2(T\#\mu, \nu) \right\}, \tag{26}$$

where T denotes the map to the residue-55 $C_\alpha - C_\alpha$ distance. For the MMD computation, we used a Gaussian RBF kernel on this residue-distance space, with bandwidth 2.5.

Experimental Setting We implement our mean-field steering method in Boltz-2, performing the appropriate modifications for the variance-exploding formulation used by the model. We use 50 steps, and slightly adjust the time schedule to spend longer in the middle of the generative procedure (according to a sigmoid-shaped CDF), as this is where the bifurcation into the distinct states appears to occur. We set the step-scale parameter of Boltz-2 to 1.5, which is a common value to control the

temperature from which the distribution is sampled. In all Boltz-2 experiments, we use a sample size of 64 and generate 1024 samples, using the multi-batch procedure in Algorithm 2. We use residual resampling at every step (though we also obtained similar results with less resampling). We observed occasional spikes in the derivative through the denoiser, so for stability we clipped the maximum change in the log-weights to 1.0 at each step, and clipped the steering term $\nabla\Psi_t(X_t^i, \hat{\mu}_t)$ so that its norm did not exceed 10% of the norm of the standard drift $b_t + \varepsilon_t s_t$. As the cost of the Picard iterations is negligible (see Table 6), we used $K_{FP} = 50$ fixed-point iterations with a damping factor of $\alpha = 0.5$ to be confident of convergence (we remark that this is likely very conservative). The Boltz-2 experiments were run on a single NVIDIA GH200 GPU.

Note that when evaluating the objective $J(\mu)$, it is intractable to evaluate the $\mathrm{KL}(\mu\|p_1)$ term in the full ambient space, so instead we compute a proxy on the $1d$ projection onto the residue-55 distances. As the data exhibits two prominent modes with different scales, we use a bandwidth of 0.2 for the left mode and 0.4 for the right mode when computing the KL divergence proxy; other than this, we follow the same procedure as in the low-dimensional experiments.

C.2.1 Additional Results

In Figure 15, we provide similar histogram plots to the ones in Figure 6 for the standard gradient-steering approach, which corresponds to the mean-field steering but without the corrective weights, with the mean-field steering results plotted again for comparison. We see that the standard gradient-steering, while steering towards the targets, results in a generated distribution that is slightly different in shape to μ^*, with a stronger peak. In contrast, mean-field steering exhibits an accurate fit to μ^*.

In Figure 16, we also plot the objective curves for values $\lambda^* \in \{1, 3\}$, in addition to $\lambda^* = 2$ also reported in Figure 5. For each, we again see that the best choice of λ indeed coincides with the true value λ^*, indicating that our method is correctly targeting μ^*. In contrast, using strength $\lambda = \lambda^*$ is not the best choice of steering strength when using gradient-only steering.

C.3 Adenylate Kinase

Another route towards observing a protein's conformational landscape is through molecular dynamics (MD) simulations. Atomistic MD is accurate but very expensive; this has motivated cheaper coarse-grained simulations to infer information about the higher-level structure. Such simulations could be used to infer approximate distributions over certain collective variables, such as important angles or residue distances. Given such information, one may then wish to 'tilt' the full-atom generations of a diffusion model towards such approximate distributional properties of the higher-level structure.

As an illustrative example of this idea, we consider another example using Boltz-2 in which we sample the conformational ensemble of adenylate kinase (AdK). This consists of 1664 atoms giving a dimensionality of 4992. AdK is known to transition between open and closed states through motions that are well-characterised through two collective variables: the LID-CORE angle and the NMP-CORE angle (Figure 17). From MD simulations, one can observe the joint distribution over these two angles. In our experiment, we consider tilting according to data from Beckstein et al. (2009), which is displayed as the underlying heatmap in Figure 20.

We generate the tilting data by sampling 4,000 points according to the densities provided by Beckstein et al. (2009). We again optimise for an MMD objective as in (26), in which the MMD is now evaluated on the 2-dimensional angle space using an isotropic kernel bandwidth of 10.0. We stop the reweighting procedure at time 0.75 so that the duplicated particles become sufficiently dispersed in the remaining generation time. For the plot in Figure 18, we classify a generation as in the 'upper-left' region if the NMP and LID angles are in the ranges $(40°, 55°)$ and $(112°, 150°)$ respectively. We remark that we observed negligible change in the proportion of steric clashes in the generations for the steering strengths we considered. Across both steering mechanisms, the mean clash rate per 1000 heavy atoms varied between 5.9 and 6.3 across the different strengths. It is unfortunately difficult to reliably evaluate a $2d$ proxy for the KL divergence in this example, as p_1 does not place sufficient mass in the upper-left region. In Figure 19 we plot just the squared MMD value, which similarly to Figures 18 and 20 demonstrate that mean-field steering produces a stronger shift towards ν.

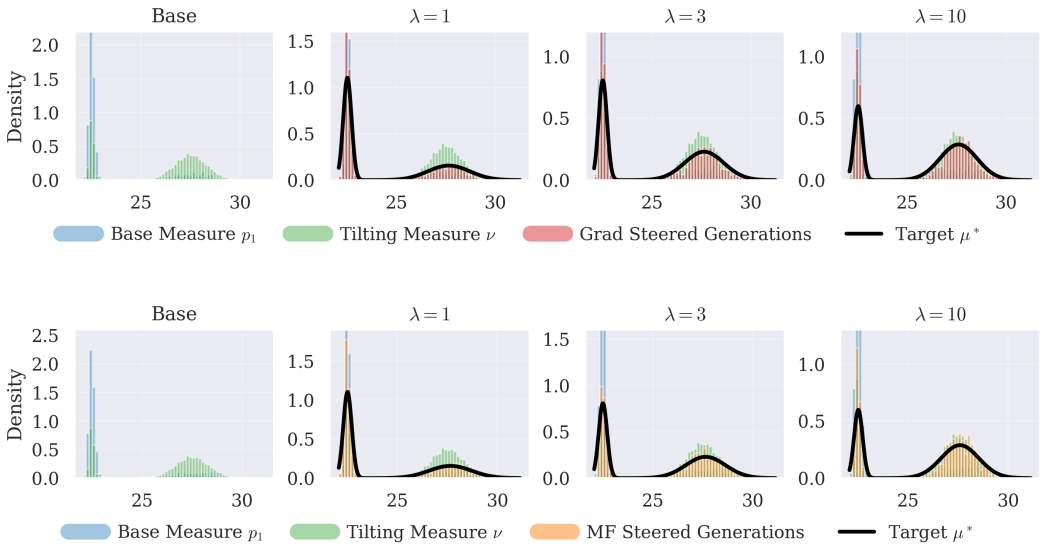
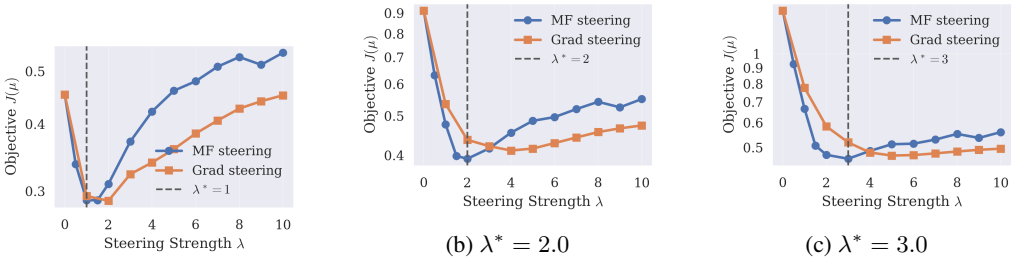


Figure 15: **HIV-1 Protease Histogram Comparison:** Comparison of the histograms generated for using gradient-only steering *(top)*, and the same steering but now using the mean-field reweighting *(bottom)*.



(a) $\lambda^* = 1.0$ (b) $\lambda^* = 2.0$ (c) $\lambda^* = 3.0$

Figure 16: **HIV-1 Protease Objective:** Values of $J(\mu) = \lambda^* \, \mathrm{MMD}^2(\mu, \nu) + \mathrm{KL}(\mu \| p_1)$, for the different steering mechanisms with varying steering strengths.

C.4 Analysis of Computational Cost

We now examine the additional computational cost of our approach compared to standard gradient-only distributional steering. The cost of the interactions between the particles in the *positional* updates is the same as in standard gradient-only steering; the additional overhead incurred by our approach is limited to the solving of the 1-dimensional log-weight dynamics. We note that as network size increases, the cost is dominated by the network evaluations. Generally, our approach will require an additional $\sim$50% of runtime and no additional peak memory usage compared to gradient-only steering; we explain this by breaking down the cost of each operation below.

Our approach requires 3 different types of network evaluation: the evaluation at X_t (a forward pass), the evaluation of $\nabla_x \Psi_t$ and $\partial_t \Psi_t$ (a backward pass), and the drift evaluation at the next step $X_{t+\delta t}$ for the finite-difference computation of $\dot{\Psi}_t$ (a forward pass). Gradient-only steering already requires the first two passes; our approach incurs only the final forward pass at the next step as an additional cost, leading to an approximately 50% increase in runtime per step. The peak memory requirements are unchanged because the final additional forward pass can be computed without gradient tracking so it incurs comparatively little memory overhead. The cost of the Picard iterations is negligible compared to the cost of the network evaluations.

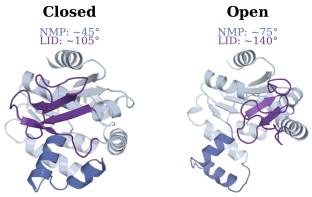


Figure 17: Open and closed conformations of Adenylate Kinase.

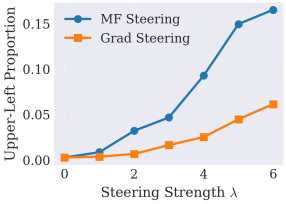


Figure 18: The proportion of generations in the 'upper-left' region, for different λ.

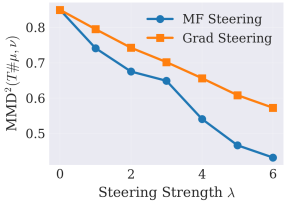
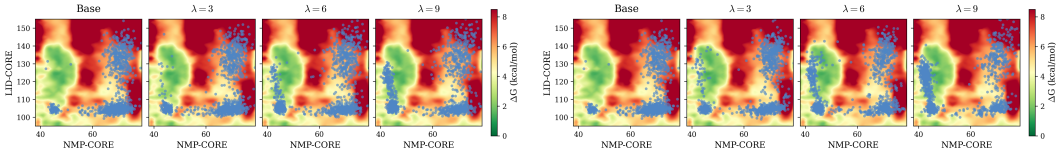


Figure 19: $\mathrm{MMD}^2(T\#\mu, \nu)$ for increasing steering strength λ.



(a) Gradient-only steering.

(b) Mean-field steering.

Figure 20: Scatter plots for the LID and NMP angles for the generated data, with overlaid empirical heatmap from Beckstein et al. (2009).

Sampler operation	1 Picard iteration	5 iterations	10 iterations	Peak memory usage
Network forward at X_t	32.41%	32.38%	32.34%	38.4 GB
Backward for $\nabla_x \Psi$ and $\partial \Psi_t$	35.56%	35.33%	35.29%	37.47 GB
Network forward at $X_{t+\delta t}$	32.19%	32.16%	32.12%	4.42 GB
Ψ evaluation and Picard iterations	0.036%	0.128%	0.244%	2.01 GB

Table 6: Breakdown of computational costs for the different operations performed in each sampling step, in the HIV-1 protease experiment, measured on an NVIDIA L40S GPU. Note that the first two operations are required for gradient-only steering too.

	Grad	$K_{\mathrm{FP}} = 1$	$K_{\mathrm{FP}} = 4$	$K_{\mathrm{FP}} = 10$
HIV-1	44s	69s	69s	69s
AdK	48s	75s	75s	75s

Table 7: Runtime comparison across steering and solver variants, in the protein examples. Time taken for 50 step simulations, with a batch size of 64, measured on an NVIDIA L40S GPU.

These observations are confirmed by the profiling of the per-step computational cost (reported in Table 6) in the HIV-1 Protease example, which reports the proportion of per-step runtime and the peak memory usage of each of these operations (similar results are observed in the other experiments too). As expected, increasing the number of Picard iterations adds negligible computational cost. In Table 7, we also report sampling runtimes for both steering mechanisms in the two protein examples, which are consistent with the runtime discussion above. As expected, the proportional overhead of the $\dot{\Psi}_t$ solver is lower compared to the timings in Table 3, as the cost of evaluating the network is now much higher. Again, the results confirm that the additional cost of further fixed point iterations is negligible.

C.5 Tilting Towards X-ray Crystallographic Electron Densities

X-Ray Crystallography Another approach for probing the conformational behaviour of proteins is X-ray crystallography. This process involves crystallising a purified sample of the protein, and analysing the resulting X-ray diffraction pattern to produce an experimentally-observed 3-dimensional electron density. This electron density is an average over all molecules in the crystal, so when the

molecules exhibit heterogeneous conformations in the structure then the density cannot be explained by a single conformation prediction alone. Aligning a generative model towards such experimentally-observed information therefore again requires using a distributional reward function.

Motivated by the fact that many experimental techniques give ensemble-level information, recent works Maddipatla et al. (2025; 2026) have considered steering Protenix (ByteDance et al., 2025) (an open-source implementation of AlphaFold3 (Abramson et al., 2024)) using the gradient of distribution-level reward functions to guide the generation. For an amino acid sequence a, and experimental observation y dependent on the ensemble $\mathcal{X} = \{X^1, ..., X^n\}$, they aim to sample an ensemble from the posterior $p(\mathcal{X}|a, y)$, which by Bayes' rule satisfies $p(\mathcal{X}|a, y) \propto p(y|\mathcal{X}, a) \cdot p(\mathcal{X}|a)$. Here, $p(y|\mathcal{X}, a)$ denotes a chosen likelihood function, and the prior $p(\mathcal{X}|a)$ is given by the base diffusion model. This corresponds to a finite-sample version of our framework, using the log-likelihood reward function $\mathcal{R}(\mu) = \log p(y|\mu, a)$. Steering according to X-ray crystallographic electron densities is an important experimental setting considered in these works.

As with other examples of gradient-only steering, steering with the gradient alone will not sample from the exact posterior distribution μ^*. We therefore here compare standard gradient-steering, akin to the approach in Maddipatla et al. (2025; 2026), along with our mean-field steering procedure, to investigate how the inclusion of the FK weights affects the behaviour.

Reward Function We largely follow the experimental settings of Maddipatla et al. (2025; 2026), with a few modifications to make suitable for our setting. The reward function is chosen as

$$\mathcal{R}(\hat{\mu}) = -\lambda \left\| F_o - \tfrac{1}{N} \sum_{i=1}^{N} F_c(X^i) \right\|_2^2, \tag{27}$$

where F_o denotes the experimentally-observed density, $F_c(X^i)$ denotes a constructed density for the generated structure X^i, and λ is a strength parameter. We remark that Maddipatla et al. (2025; 2026) used an L_1 distance rather than L_2 considered here; we chose to use L_2 here as we found it gave more consistent behaviour between experiments. Following Maddipatla et al. (2025), we steer only the atoms that are in the region of interest (ROI) that exhibits the variations in structure. To do so, the electron density $F_c(X^i)$ is constructed using the current generation's ROI atoms combined with non-ROI atoms from the known reference structure in the PDB dataset. For 4OLE, the ROI consists of the residues 423-431 (highlighted in red in Figure 7).

As the densities will not necessarily be on the same scale, we centre and normalise both densities (we stop the gradient with respect to atom positions through the standardisation constants). Similar to Maddipatla et al. (2025), the density $F_c(X^i)$ for the generated samples is constructed as a mixture of Gaussian kernels,

$$F_c(\xi) \;=\; \sum_{i=1}^{m} \sum_{j=1}^{6} a_{ij} \left(\frac{4\pi}{b_{ij} + B_i} \right)^{3/2} \exp\!\left(-\frac{4\pi^2}{b_{ij} + B_i} \left\| \mathbf{x}_i - \xi \right\|_2^2 \right). \tag{28}$$

Here, $\xi \in \mathbb{R}^3$ is the point where the density is being evaluated, and $\mathbf{x}_i \in \mathbb{R}^3$ is the 3d position of the i-th atom, and a_{ij}, b_{ij} are per-atom constants. The B_i factors additionally determine the width of the kernels; the work Maddipatla et al. (2025) sets this to $B = 4/N$, but we choose to use the B factors from the PDB data as we simulate larger ensembles. As we are interested in the behaviour of the atoms in the ROI, we construct this density using the predicted ROI atoms, along with non-ROI atom positions from the known reference structure. Therefore the steering acts only on the ROI atoms. Note that before evaluating the reward, the generations are aligned to the reference structure using the non-ROI atoms using the Kabsch algorithm (Kabsch, 1976). We use the $2mF_\mathrm{o} - DF_\mathrm{c}$ map of 4OLE as the target density, whose values are given on the FFT grid at approximately 0.84Å spacing. We evaluate the reward using the 13,247 points on the grid within 5Å of the ROI atoms.

The implementation in Maddipatla et al. (2025) also includes an additional substructure-conditioning term, which steers the non-ROI atoms towards their known positions in the reference structure. We omit this steering in our implementation, as it would mean that the base generative process would not follow the trajectory of a trained flow model which is required for us to have access to the score of the marginals. In practice, we found the base diffusion process already had good alignment with the structure outside of the ROI, so the generations were reasonable without requiring this additional steering term.

Sampling We sample 16 batches of 64 samples, using our multi-batch sampling procedure. We note that the sampled distribution became more consistent as the number of batches increased, which we anticipate may be due to using more terminal samples in the empirical approximation of $\hat{\mu}_1$ which gives a better approximation (as the terminal samples are more realistic than the Tweedie estimates). We used 100 sampling steps, using residual sampling at each reweighting step. Similarly to in the Boltz-2 examples, we used a middle-heavy schedule to spend more time in the period in which the modes were determined. We used $K_{FP} = 200$ Picard iterations with damping factor $\alpha = 0.1$ to be confident of convergence (this is again likely very conservative, but adds negligible cost). We stopped applying the reweighting procedure after time 0.75 to allow the particles to become sufficiently dispersed in the final generated ensemble (since in the Protenix sampling procedure, the noise becomes increasingly small towards the terminal time). The experiments were run on a single NVIDIA GH200 GPU.

Results We report results in Figure 9. In Figure 9a, we report how the cosine alignment between the target density F_o and the constructed density $\sum_i F_c(X^i)$ of the generated ensemble changes as the steering strength increases, which reflects changes in the reward $\mathcal{R}(\mu)$. In Figure 9b we do likewise for the proportion of the generations closer to conformation B. In Figure 9c, we also plot this same proportion against the cosine-alignment obtained using different steering strengths; the displayed results use strengths $\lambda \in \{0.0, 0.001, 0.002, 0.003, 0.004, 0.005, 0.0075, 0.01\}$ for gradient-only steering, and $\lambda \in \{0.0, 0.001, 0.0015, 0.002, 0.0025, 0.003, 0.0035, 0.004, 0.005\}$ for mean-field steering. The visualisation of the experimental setting in Figure 8 is inspired by Figure 3 in Maddipatla et al. (2025); Figure 8a displays three samples from the base diffusion model, and Figure 8b shows samples obtained using mean-field steering with strength $\lambda = 0.003$. For clear visualisation, we display three samples close to conformation A and the three samples closest to conformation B.

In the absence of a known ground-truth, we instead focus on discussing the qualitative differences between the results. We first note that for the same steering strength, mean-field steering appears to result in a stronger reward tilt, suggesting that gradient-only steering understeers relative to the true inverse-problem target μ^* in this instance. We also observe that mean-field steering placed more samples in the alternate conformation (which appeared more difficult for gradient-only steering). We note that the accuracy of these results is likely limited by using a simple KDE approximation for the generated electron density; as noted by Maddipatla et al. (2025), we anticipate these results could be improved by using a more accurate electron density model.

D Proofs

In this section, we provide the proofs of the theoretical results in the paper.

D.1 Proof of Proposition 3.1

We establish the result under the following (mild) regularity assumption on $\mathcal{R}$:

Assumption D.1. *The reward functional $\mathcal{R}$ admits a functional derivative, is upper semicontinuous with respect to weak convergence and bounded from above.*

Proof of Proposition 3.1. Denote the objective by

$$\mathcal{J}(\mu) := \mathcal{R}(\mu) - \mathrm{KL}(\mu \| p_1).$$

1) We first show that $\mathcal{J}$ admits a maximiser. Since $\mathcal{R}$ is bounded from above (say $\sup_{\mu \in \mathcal{P}(\mathbb{R}^d)} \mathcal{R}(\mu) \leq C_{\mathcal{R}}$) and $\mathrm{KL}(\mu \| p_1) \geq 0$, we have $\sup_{\mu \in \mathcal{P}(\mathbb{R}^d)} \mathcal{J}(\mu) < \infty$.

Let $(\mu_n)_{n \geq 1}$ be a maximising sequence. Then, for n large enough, we have that $\mathcal{J}(\mu_n) \geq \mathcal{R}(p_1) - 1$, because $\mathrm{KL}(p_1 \| p_1) = 0$. It follows that

$$\mathrm{KL}(\mu_n \| p_1) = \mathcal{R}(\mu_n) - \mathcal{J}(\mu_n) \leq C_{\mathcal{R}} - \mathcal{R}(p_1) + 1,$$

and thus the sequence $(\mathrm{KL}(\mu_n \| p_1))_{n \geq 1}$ is bounded. The sublevel sets of the relative entropy are weakly compact, so, after passing to a subsequence, we may assume that $\mu_n \Rightarrow \mu^*$, weakly for some candidate maximiser $\mu^* \in \mathcal{P}(\mathbb{R}^d)$.

Since $\mathcal{R}$ is upper semicontinuous with respect to weak convergence, and since $\mathrm{KL}(\cdot\|p_1)$ is lower semicontinuous, we have

$$\mathcal{J}(\mu^*) = \mathcal{R}(\mu^*) - \mathrm{KL}(\mu^*\|p_1) \geq \limsup_{n\to\infty} \left(\mathcal{R}(\mu_n) - \mathrm{KL}(\mu_n\|p_1)\right).$$

Hence, μ^* is indeed a maximiser. This maximiser is also unique by concavity of $\mathcal{R}$ and strict convexity of $\mathrm{KL}(\cdot\|p_1)$.

2) We now derive the first-order optimality condition (9). For an arbitrary perturbation $\nu \in \mathcal{P}(\mathbb{R}^d)$, set

$$\mu_\varepsilon = (1-\varepsilon)\mu^* + \varepsilon\nu.$$

By optimality of μ^*, the derivative of $\varepsilon \mapsto \mathcal{J}(\mu_\varepsilon)$ at $\varepsilon = 0$ vanishes. By definition, we have

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{R}(\mu_\varepsilon)\bigg|_{\varepsilon=0} = \int_{\mathbb{R}^d} \Psi(x,\mu^*)\,(\nu-\mu^*)(\mathrm{d}x).$$

We also compute the functional derivative of the KL-regulariser:

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathrm{KL}(\mu_\varepsilon\|p_1)\bigg|_{\varepsilon=0} = \int_{\mathbb{R}^d} \left(\log\frac{\mathrm{d}\mu^*}{\mathrm{d}p_1}(x) + 1\right)(\nu-\mu^*)(\mathrm{d}x).$$

As a consequence,

$$\int_{\mathbb{R}^d} \left[\Psi(x,\mu^*) - \log\frac{\mathrm{d}\mu^*}{\mathrm{d}p_1}(x) - 1\right](\nu-\mu^*)(\mathrm{d}x) = 0.$$

Since ν is arbitrary, and $\nu - \mu^*$ has total mass zero, we see that

$$\log\frac{\mathrm{d}\mu^*}{\mathrm{d}p_1}(x) = \Psi(x,\mu^*) - C,$$

for some constant $C \in \mathbb{R}$. Equivalently,

$$\mu^*(\mathrm{d}x) = e^{-C}e^{\Psi(x,\mu^*)}p_1(x)\,\mathrm{d}x,$$

leading to (9) after enforcing normalisation. □

D.2 Proof of Proposition D.2

In order to construct a particle system that realises the interpolating measures defined in (10), we need the following intermediate result about μ_t^*.

Proposition D.2. *The targets* (10) *satisfy the evolution equation*

$$\partial_t\mu_t^*(x) = \mu_t^*(x)\left(h_t(x) - \int_{\mathbb{R}^d} h_t(y)\mu_t^*(y)\,dy\right), \tag{29}$$

where we define the log-growth rate $h_t(x) := \frac{\mathrm{d}}{\mathrm{d}t}\left(\Psi_t(x,\mu_t^*)\right) - \nabla\cdot b_t(x) - \nabla\log p_t(x)\cdot b_t(x)$*, which is measure-dependent.*

Remark D.1. *As described by* (29)*, the target μ_t^* is instantaneously reweighted by h_t (relative to its mean). This type of equation appears in several related settings, including SMC and Feynman-Kac type flows (Del Moral, 2004), tempering and annealed importance sampling (Neal, 2001), and evolutionary replicator dynamics (Hofbauer & Sigmund, 1998). In our setting, much like $\Psi_t(x;\mu_t^*)$ in* (10)*, h_t itself depends on the targets μ_t^*.*

Proof of Proposition D.2. We take logarithms on both sides of (10), leading to

$$\log\mu_t^*(x) = \Psi_t(x,\mu_t^*) + \log p_t(x) - \log Z_t.$$

Differentiating in time yields

$$\frac{\partial_t\mu_t^*(x)}{\mu_t^*(x)} = \frac{\mathrm{d}}{\mathrm{d}t}\Psi_t(x,\mu_t^*) + \partial_t\log p_t(x) - \partial_t\log Z_t.$$

The time marginals associated to the base dynamics (3) satisfy the continuity equation

$$\partial_t p_t + \nabla \cdot (p_t b_t) = 0,$$

and thus we obtain

$$\partial_t \log p_t(x) = \frac{\partial_t p_t(x)}{p_t(x)} = -\frac{\nabla \cdot (p_t b_t)(x)}{p_t(x)} = -\nabla \cdot b_t(x) - \nabla \log p_t(x) \cdot b_t(x).$$

Therefore, by the definition of h_t,

$$\frac{\partial_t \mu_t^*(x)}{\mu_t^*(x)} = h_t(x) - \partial_t \log Z_t. \tag{30}$$

It remains to identify $\partial_t \log Z_t$. Differentiating under the integral sign gives

$$\partial_t Z_t = \int_{\mathbb{R}^d} e^{\Psi_t(y,\mu_t^*)} p_t(y) \left[\frac{\mathrm{d}}{\mathrm{d}t} \Psi_t(y, \mu_t^*) + \partial_t \log p_t(y) \right] \mathrm{d}y = \int_{\mathbb{R}^d} e^{\Psi_t(y,\mu_t^*)} p_t(y) h_t(y) \, \mathrm{d}y,$$

using the definition of h_t in the second equality. Therefore, we have

$$\partial_t \log Z_t = \frac{\partial_t Z_t}{Z_t} = \int_{\mathbb{R}^d} h_t(y) \, \mu_t^*(y) \, \mathrm{d}y,$$

and combining with (30) gives (29). □

D.3 Proof of Theorem 3.2

We establish the result under the following assumption:

Assumption D.3. *The McKean-Vlasov dynamics* (12) *are well posed: for the initial condition* $X_0 \sim p_0$*, the solution exists and is unique.*

This assumption holds true under mild growth and regularity conditions on the base drift b_t and the reward $\mathcal{R}$ (see, for example, Del Moral & Miclo (2007, Section 3) or Carmona & Delarue (2018, Section 4.2)). We highlight the following subtlety: to convert (12) into standard form covered by those references, it is necessary to *define* $\dot{\Psi}$ via the linear equation (13) (and verify appropriate assumptions on the associated solution operator), because time derivatives with respect to the measure μ_t are non-standard in this context.

Proof of Theorem 3.2. For convenience, let us introduce the short-hands

$$B_t(x, \mu) := b_t(x) + \varepsilon_t s_t(x) + \varepsilon_t \nabla_x \Psi_t(x; \mu), \qquad g_t(x, \mu) := b_t(x) \cdot \nabla_x \Psi_t(x; \mu) + \dot{\Psi}_t(x), \tag{31}$$

and define the unnormalised weighted law

$$\nu_t(\phi) := \mathbb{E}[\phi(X_t) e^{A_t}].$$

By Itô's formula, ν_t satisfies the Feynman–Kac equation (Del Moral & Miclo, 2007, Section 3.1)

$$\partial_t \nu_t = (L_t^{\mu_t})^\dagger \nu_t + g_t(\cdot, \mu_t) \nu_t,$$

where

$$L_t^\mu \phi = B_t(\cdot, \mu) \cdot \nabla \phi + \varepsilon_t \Delta \phi \tag{32}$$

is the μ-dependent generator of (12a), and $(L_t^{\mu_t})^\dagger$ is its adjoint in $L^2(\mathbb{R}^d)$. Since $\mu_t = \nu_t / \nu_t(1)$, the normalised law satisfies

$$\partial_t \mu_t = (L_t^{\mu_t})^\dagger \mu_t + (g_t(\cdot, \mu_t) - \mu_t(g_t(\cdot, \mu_t))) \, \mu_t. \tag{33}$$

We now check that that path of target measures μ_t^* satisfies the same equation. Since $\mu_t^*(x) = \frac{1}{Z_t} e^{\Psi_t(x;\mu_t^*)} p_t(x)$ and $s_t = \nabla \log p_t$, we can write the score of the target measures as

$$s_t + \nabla_x \Psi_t(\cdot; \mu_t^*) = \nabla \log \mu_t^*.$$

Consequently, looking at the expression for $(L_t^{\mu_t^*})^*\mu_t^*$, the ε_t-terms cancel,

$$\begin{aligned}(L_t^{\mu_t^*})^*\mu_t^* &= -\nabla\cdot\left((b_t+\varepsilon_t\nabla\log\mu_t^*)\,\mu_t^*\right)+\varepsilon_t\Delta\mu_t^*\\ &= -\nabla\cdot(b_t\mu_t^*)-\varepsilon_t\nabla\cdot(\mu_t^*\nabla\log\mu_t^*)+\varepsilon_t\Delta\mu_t^*\\ &= -\nabla\cdot(b_t\mu_t^*).\end{aligned}$$

Expanding this divergence, we have

$$\begin{aligned}-\nabla\cdot(b_t\mu_t^*) &= -(\nabla\cdot b_t)\,\mu_t^*-b_t\cdot\nabla\mu_t^*\\ &= -(\nabla\cdot b_t)\,\mu_t^*-b_t\cdot\big(\mu_t^*(s_t+\nabla\Psi_t))\big)\\ &= \big(-\nabla\cdot b_t-b_t\cdot s_t-b_t\nabla\Psi_t\big)\,\mu_t^*\end{aligned}$$

Noting that $g_t(x,\mu_t^*)\mu_t^*=\big(b_t\cdot\nabla\Psi_t+\dot\Psi_t\big)\mu_t^*$, we therefore have

$$\begin{aligned}(L_t^{\mu_t^*})^*\mu_t^*+g_t(\cdot,\mu_t^*)\mu_t^* &= \big(\dot\Psi_t-\nabla\cdot b_t-b_t\cdot s_t\big)\,\mu_t^*\\ &= h_t\,\mu_t^*,\end{aligned}$$

where h_t is the function from Proposition D.2. Integrating this expression, and noting that the first term integrates to 0, we see that $\mu_t^*(g_t(\cdot,\mu_t^*))=\mu_t^*(h_t)$. After subtracting this, we have

$$(L_t^{\mu_t^*})^*\mu_t^*+\left(g_t(\cdot,\mu_t^*)-\mu_t^*(g_t(\cdot,\mu_t^*))\right)\mu_t^*=\mu_t^*\left(h_t-\mu_t^*(h_t)\right).$$

By Proposition D.2, the right-hand side equals $\partial_t\mu_t^*$. Thus μ_t^* solves the same normalised Feynman–Kac equation (33) as μ_t. Since the initial conditions agree and the equation is unique by assumption, we conclude that

$$\mu_t=\mu_t^*,\qquad t\in[0,T].$$

□

D.4 Proof of Proposition 3.3

Proof. We use the definition $\dot\Psi_t(x)=\frac{\mathrm{d}}{\mathrm{d}t}\big(\Psi_t(x,\mu_t)\big)$ and the chain rule to compute

$$\begin{aligned}\dot\Psi_t(x) &= \partial_t\Psi_t(x,\mu_t)+\int_{\mathbb{R}^d}\Phi(x,y,\mu_t)\,\partial_t\mu_t(\mathrm{d}y)\\ &= \partial_t\Psi_t(x,\mu_t)+\int_{\mathbb{R}^d}\Phi(x,y,\mu_t)\left(h_t(y)-\int_{\mathbb{R}^d}h_t(z)\mu_t(z)\,\mathrm{d}z\right)\mu_t(\mathrm{d}y)\\ &= \partial_t\Psi_t(x,\mu_t)+\int_{\mathbb{R}^d}\tilde\Phi(x,y,\mu_t)\,h_t(y)\,\mu_t(\mathrm{d}y)\\ &= \partial_t\Psi_t(x,\mu_t)+\int_{\mathbb{R}^d}\tilde\Phi(x,y,\mu_t)\left[\dot\Psi_t(y)-\nabla\cdot b_t(y)-\nabla\log p_t(y)\cdot b_t(y)\right]\mu_t(\mathrm{d}y),\end{aligned}$$

where we have used Proposition D.2 in the second line. From a computational standpoint, the problematic term here is $\int_{\mathbb{R}^d}\tilde\Phi(x,y,\mu_t)\nabla\cdot b_t(y)\mu_t(\mathrm{d}y)$, as this divergence is required to be computed in the data space which could be large. Fortunately, we can use integration by parts to obtain a more tractable expression. For any test function $\phi(y)$, we have

$$\begin{aligned}\int\phi(x)\nabla\cdot b_t(x)\,\mu_t(\mathrm{d}x) &= -\int b_t(x)\cdot\nabla\big(\phi(x)\mu_t(x)\big)\,\mathrm{d}x\\ &= -\int b_t(x)\cdot\big[\mu_t(x)\nabla\phi(x)+\phi(x)\nabla\mu_t(x)\big]\,\mathrm{d}x\\ &= -\int b_t(x)\cdot\mu_t(x)\nabla\phi(x)\,\mathrm{d}x-\int\phi(x)b_t(x)\cdot\nabla\log\mu_t(x)\,\mu_t(\mathrm{d}x)\\ &= -\int b_t(x)\cdot\mu_t(x)\nabla\phi(x)\mathrm{d}x-\int\phi(x)\,b_t(x)\cdot\nabla\Psi_t(x,\mu_t)\,\mu_t(\mathrm{d}x)\\ &\quad-\int\phi(x)\,b_t(x)\cdot\nabla\log p_t(x)\,\mu_t(\mathrm{d}x),\end{aligned}$$

where we used the log-derivative trick $\nabla\mu_t=(\nabla\log\mu_t)\mu_t$, and in the final line used the fact that $\nabla\log\mu_t=\nabla\Psi_t(\cdot,\mu_t)+\nabla\log p_t$ (which follows from (10)). Taking $\phi(y)=\tilde\Phi_t(x,y,\mu_t)$ and noting that the terms involving the score $\nabla\log p_t$ cancel, we obtain (13). □

D.5 Proof of Theorem 3.4

Consider the dynamics

$$\mathrm{d}X_t^{i,N} = \Big[b_t(X_t^{i,N}) + \varepsilon_t s_t(X_t^{i,N}) + \varepsilon_t \nabla_x \Psi_t(X_t^{i,N}; \hat{\mu}_t^N)\Big]\,\mathrm{d}t + \sqrt{2\varepsilon_t}\,\mathrm{d}B_t^i, \qquad X_0^{i,N} \sim p_0, \tag{34a}$$

$$\mathrm{d}A_t^{i,N} = \Big[b_t(X_t^{i,N}) \cdot \nabla_x \Psi_t(X_t^{i,N}; \hat{\mu}_t^N) + \dot{\Psi}_t^{(N)}(X_t^{i,N})\Big]\,\mathrm{d}t, \qquad A_0^{i,N} = 0. \tag{34b}$$

where the weighted empirical measure is

$$\hat{\mu}_t^N := \sum_{i=1}^N w_t^{i,N} \delta_{X_t^{i,N}}, \tag{35}$$

with normalised weights

$$w_t^{i,N} := \frac{\exp(A_t^{i,N})}{\sum_{j=1}^N \exp(A_t^{j,N})}. \tag{36}$$

In (34b), $\dot{\Psi}_t^{(N)}$ denotes a computable approximation of $\dot{\Psi}_t = \frac{\mathrm{d}}{\mathrm{d}t}\Psi_t(\cdot;\mu_t)$, in the following sense: For an arbitrary probability measure μ, define $\dot{\Psi}_t^{\mathrm{exact}}(\cdot,\mu)$ as the solution to (13) where all occurrences of μ_t^* are replaced by μ. In other words, $\dot{\Psi}_t^{\mathrm{exact}}(\cdot,\mu)$ is defined to be the solution to

$$\dot{\Psi}_t^{\mathrm{exact}}(x,\mu) = \partial_t \Psi_t(x,\mu) \tag{37}$$

$$+ \int_{\mathbb{R}^d} \Big(\tilde{\Phi}_t(x,y,\mu)\dot{\Psi}_t^{\mathrm{exact}}(y,\mu) + b_t(y)\cdot\nabla_y\tilde{\Phi}_t(x,y,\mu) + \tilde{\Phi}_t(x,y,\mu)\, b_t(y)\cdot\nabla_y\Psi_t(y,\mu)\Big)\,\mathrm{d}\mu(y). \tag{38}$$

In particular, for the weighted empirical measure $\hat{\mu}_t^N = \sum_{j=1}^N w_t^{j,N}\delta_{X_t^{j,N}}$, this equation reads

$$\dot{\Psi}_t^{\mathrm{exact}}(x,\hat{\mu}_t^N) = \partial_t\Psi_t(x,\hat{\mu}_t^N) + \sum_{j=1}^N w_t^{j,N}\Big[\tilde{\Phi}_t(x,X_t^{j,N},\hat{\mu}_t^N)\dot{\Psi}_t^{\mathrm{exact}}(X_t^{j,N},\hat{\mu}_t^N) + b_t(X_t^{j,N})\cdot\nabla_y\tilde{\Phi}_t(x,X_t^{j,N},\hat{\mu}_t^N) + \tilde{\Phi}_t(x,X_t^{j,N},\hat{\mu}_t^N)\, b_t(X_t^{j,N})\cdot\nabla_y\Psi_t(X_t^{j,N},\hat{\mu}_t^N)\Big]; \tag{39}$$

evaluating this at $x = X_t^{i,N}$ leads to an $N\times N$ linear system that forms the basis of Algorithm 3.

The implemented quantity $\dot{\Psi}_t^{(N)}$ used in (34b) may instead only be approximately equal to $\dot{\Psi}_t^{\mathrm{exact}}(\cdot,\hat{\mu}_t^N)$, for example coming from a finite number of Picard iterations as in Algorithm 4. The solver-consistency condition in the third item of Assumption D.4 quantifies the resulting numerical error.

Assumption D.4 (Regularity, well-posedness and solver consistency)**.** *Assume the following.*

1. *The McKean-Vlasov dynamics (12) are well posed, in the sense that the corresponding Feynman-Kac equation (33) has a unique solution in $C([0,T],\mathcal{P}(\mathbb{R}^d))$.*
2. *The drift coefficients of (12) (denoted by B_t and g_t in (31)) are bounded and Lipschitz in (x,μ), uniformly for $t\in[0,T]$. More precisely, for some constant $L>0$,*

$$|B_t(x,\mu) - B_t(y,\nu)| \le L\big(|x-y| + d_{\mathrm{BL}}(\mu,\nu)\big), \tag{40}$$

$$|g_t(x,\mu) - g_t(y,\nu)| \le L\big(|x-y| + d_{\mathrm{BL}}(\mu,\nu)\big), \tag{41}$$

and

$$|B_t(x,\mu)| + |g_t(x,\mu)| \le L. \tag{42}$$

We also assume that $\sup_{t\in[0,T]}\varepsilon_t < \infty$.

3. *The approximation $\dot{\Psi}_t^{(N)}$ is consistent in the sense that*

$$\delta_N := \mathbb{E}\left[\int_0^T \sum_{i=1}^N w_t^i \left|\dot{\Psi}_t^{(N)}(X_t^i) - \dot{\Psi}_t^{\text{exact}}(X_t^i, \hat{\mu}_t^N)\right| \mathrm{d}t\right] \longrightarrow 0 \qquad \text{as } N \to \infty, \quad (43)$$

where $\dot{\Psi}_t^{\text{exact}}(\cdot, \hat{\mu}_t^N)$ denotes the exact solution for $\dot{\Psi}$ to equation (13) in which every occurrence of μ_t^* is replaced by $\hat{\mu}_t^N$, and $t \in [0, T]$ is fixed.

Under this assumption, the weighted empirical measure in (35) converges to the target μ_t^*:

Theorem D.5 (**Convergence of the Weighted Particle System**). *Suppose Assumption D.4 holds. Then*

$$\mathbb{E}\left[\sup_{t\in[0,T]} \left|\hat{\mu}_t^N(\phi) - \mu_t^*(\phi)\right|\right] \longrightarrow 0 \qquad \text{as } N \to \infty, \quad (44)$$

for every $\phi \in C_b^2(\mathbb{R}^d)$,

With this in place, a similar result holds if the particle system (34) is rejuvenated by resampling:

Corollary D.5.1 (Convergence with residual resampling). *Suppose the assumptions of Theorem D.5 hold, and suppose that residual resampling is performed at a fixed finite collection of times*

$$0 < \tau_1 < \cdots < \tau_K \leq T,$$

where the maximum number of resampling steps K is uniformly bounded in N. After each resampling event, the particles are assigned equal weights $\frac{1}{N}$, i.e. $A_{\tau_i}^j = 0$ for $j = 1, \ldots, N$ and $i = 1, \ldots, K$. Then, for every $\phi \in C_b^2(\mathbb{R}^d)$,

$$\mathbb{E}\left[\sup_{t\in[0,T]} \left|\hat{\mu}_t^N(\phi) - \mu_t^*(\phi)\right|\right] \longrightarrow 0 \qquad \text{as } N \to \infty.$$

Before proceeding to the proof, we need the following auxiliary result:

Lemma D.6. *Defining the squared sum of weights $S_t^N := \sum_{i=1}^N (w_t^{i,N})^2$, we have that*

$$\mathbb{E}\left[\sup_{t\in[0,T]} S_t^N\right] \leq e^{4LT}\left(\frac{1}{N} + 4\delta_N\right) \longrightarrow 0, \qquad \text{as } N \to \infty. \quad (45)$$

Proof. For each particle, define the shorthand

$$g_t^{i,N} := g_t\left(X_t^{i,N}, \widehat{\mu}_t^N\right)$$

and the solver error

$$e_t^{i,N} := \dot{\Psi}_t^{(N)}\left(X_t^{i,N}\right) - \dot{\Psi}_t^{\text{exact}}\left(X_t^{i,N}, \widehat{\mu}_t^N\right).$$

We also define the weighted averages

$$\overline{g}_t^N := \sum_{j=1}^N w_t^{j,N} g_t^{j,N}, \qquad \overline{e}_t^N := \sum_{j=1}^N w_t^{j,N} e_t^{j,N}.$$

Differentiating (36), and using (34b), gives

$$\frac{\mathrm{d}}{\mathrm{d}t} w_t^{i,N} = w_t^{i,N}\left(g_t^{i,N} - \overline{g}_t^N + e_t^{i,N} - \overline{e}_t^N\right).$$

Consequently,

$$\frac{\mathrm{d}}{\mathrm{d}t} S_t^N = 2\sum_{i=1}^N \left(w_t^{i,N}\right)^2 \left(g_t^{i,N} - \overline{g}_t^N\right) + 2\sum_{i=1}^N \left(w_t^{i,N}\right)^2 \left(e_t^{i,N} - \overline{e}_t^N\right). \quad (46)$$

We proceed by estimating the right-hand side of (46): By the boundedness assumption (42), we have $\left|g_t^{i,N}\right| \leq L$ and $\left|\overline{g}_t^N\right| \leq \sum_{j=1}^N w_t^{j,N} \left|g_t^{j,N}\right| \leq L$, and hence

$$2\sum_{i=1}^N \left(w_t^{i,N}\right)^2 \left(g_t^{i,N} - \overline{g}_t^N\right) \leq 4LS_t^N.$$

By the definition (36), the weights $w_t^{i,N}$ are nonnegative and sum to one, and so $\left(w_t^{i,N}\right)^2 \leq w_t^{i,N}$ and $S_t^N \leq 1$. Therefore,

$$2\sum_{i=1}^N \left(w_t^{i,N}\right)^2 \left(e_t^{i,N} - \overline{e}_t^N\right) \leq 2\sum_{i=1}^N \left(w_t^{i,N}\right)^2 \left|e_t^{i,N}\right| + 2S_t^N \left|\overline{e}_t^N\right| \leq 4\sum_{i=1}^N w_t^{i,N} \left|e_t^{i,N}\right|.$$

Combining the preceding estimates with (46) gives

$$\frac{\mathrm{d}}{\mathrm{d}t} S_t^N \leq 4LS_t^N + 4\sum_{i=1}^N w_t^{i,N} \left|e_t^{i,N}\right|.$$

Since the weights are initialised uniformly, $w_0^{i,N} = 1/N$, we have $S_0^N = \sum_{i=1}^N \frac{1}{N^2} = \frac{1}{N}$. Grönwall's inequality therefore yields

$$\sup_{t\in[0,T]} S_t^N \leq e^{4LT}\left[\frac{1}{N} + 4\int_0^T \sum_{i=1}^N w_t^{i,N} \left|e_t^{i,N}\right| \,\mathrm{d}t\right].$$

Taking expectations then implies the claim. □

Proof of Theorem D.5. We fix $\phi \in C_b^2(\mathbb{R}^d)$ and derive the evolution equation for the empirical measure $\hat{\mu}_t^N(\phi) = \sum_{i=1}^N w_t^{i,N}\phi(X_t^{i,N})$. Itô's formula gives

$$\hat{\mu}_t^N(\phi) = \hat{\mu}_0^N(\phi) + \int_0^t \mathcal{F}_s^\phi(\hat{\mu}_s^N)\,\mathrm{d}s + E_t^{N,\phi} + M_t^{N,\phi}, \tag{47}$$

where

$$\mathcal{F}_t^\phi(\mu) := \mu\left(B_t(\cdot,\mu)\cdot\nabla\phi + \varepsilon_t\Delta\phi\right) + \mu(g_t(\cdot,\mu)\phi) - \mu(g_t(\cdot,\mu))\mu(\phi) \tag{48}$$

refers to the integrated right-hand side of (33), the solver error is encoded in

$$E_t^{N,\phi} := \int_0^t R_s^{N,\phi}\,\mathrm{d}s, \qquad R_t^{N,\phi} := \sum_{i=1}^N w_t^{i,N} e_t^{i,N}\left(\phi(X_t^{i,N}) - \hat{\mu}_t^N(\phi)\right), \tag{49}$$

and

$$M_t^{N,\phi} := \int_0^t \sqrt{2\varepsilon_s}\sum_{i=1}^N w_s^{i,N}\nabla\phi(X_s^{i,N})\cdot\mathrm{d}B_s^i \tag{50}$$

is the martingale part. Notice that $|R_t^{N,\phi}| \leq 2\|\phi\|_\infty \sum_{i=1}^N w_t^{i,N}\left|e_t^{i,N}\right|$; the assumption in (43) therefore implies

$$\mathbb{E}\left[\sup_{t\in[0,T]} |E_t^{N,\phi}|\right] \leq 2\|\phi\|_\infty\delta_N \longrightarrow 0. \tag{51}$$

The quadratic variation of the martingale part can be estimated as

$$\langle M^{N,\phi}\rangle_T \leq 2\|\varepsilon\|_\infty T\|\nabla\phi\|_\infty^2 \sup_{t\in[0,T]} S_t^N.$$

Itô's isometry, Doob's L^2 maximal inequality, and Lemma D.6 therefore imply

$$\mathbb{E}\left[\sup_{t\in[0,T]} |M_t^{N,\phi}|^2\right] \leq C_T\|\nabla\phi\|_\infty^2\left(\frac{1}{N} + \delta_N\right) \longrightarrow 0, \tag{52}$$

where the constant C_T is independent of N. Formally, we thus conclude that (47) converges term-by-term to a weak form of the Feynman-Kac equation (33) which governs the target law μ_t^*. To make this argument precise, we argue as follows:

Tightness. We show that the family $\{\mathrm{Law}(\hat{\mu}^N) : N \geq 1\}$ is tight in $C([0,T], \mathcal{P}(\mathbb{R}^d))$. To this end, fix $\phi \in C_c^\infty(\mathbb{R}^d)$. By the boundedness assumption (42) and $\sup_{t\in[0,T]} \varepsilon_t < \infty$, we have that $\sup_{t,\mu} |\mathcal{F}_t^\phi(\mu)| < \infty$. Therefore, the drift paths $t \mapsto \int_0^t \mathcal{F}_s^\phi(\hat{\mu}_s^N)\,\mathrm{d}s$ have a common deterministic Lipschitz constant, while (51) and (52) show that the remaining error $E_t^{N,\phi} + M_t^{N,\phi}$ in (47) vanishes uniformly in probability. It now follows from Arzela–Ascoli that the family of real-valued processes $(\hat{\mu}^N(\phi))_{N\geq 1}$ is tight in $C([0,T],\mathbb{R})$. To obtain tightness of $\{\mathrm{Law}(\hat{\mu}^N) : N \geq 1\}$, we verify the conditions of Jakubowski's tightness criterion (Jakubowski, 1986, Theorem 3.1(i)): indeed, we may choose a countable convergence-determining family $\mathcal{G} \subset C_c^\infty(\mathbb{R}^d)$; compact containment follows from Lemma D.7 below. Finally, since every $\hat{\mu}^N$ has continuous paths, and the Skorokhod topology restricted to continuous paths is the uniform topology (Jakubowski, 1986, Proposition 1.6(i)), the laws $\{\mathrm{Law}(\hat{\mu}^N) : N \geq 1\}$ are indeed tight in $C([0,T], \mathcal{P}(\mathbb{R}^d))$.

Subsequential limits. We conclude by showing that every subsequence of $\hat{\mu}^N$ has a further subsequence whose law converges weakly to the Dirac measure at μ^* in $\mathcal{P}(C([0,T], \mathcal{P}(\mathbb{R}^d)))$: Let $(N_j) \subset \mathbb{N}$ be an arbitrary subsequence. By tightness, it has a further subsequence, not relabelled, such that

$$\hat{\mu}^{N_j} \Longrightarrow \overline{\mu} \qquad \text{in } C([0,T], \mathcal{P}(\mathbb{R}^d)).$$

For every $\phi \in C_c^\infty(\mathbb{R}^d)$, Assumption D.4 implies the Lipschitz property

$$|\mathcal{F}_t^\phi(\mu) - \mathcal{F}_t^\phi(\nu)| \leq C_\phi d_{\mathrm{BL}}(\mu, \nu), \tag{53}$$

uniformly in t. Together with the facts that $E_t^{N,\phi} + M_t^{N,\phi}$ converges to zero in probability and that the initial empirical measures converge to p_0 by the law of large numbers, the continuous mapping theorem and Slutsky's lemma imply that, almost surely,

$$\overline{\mu}_t(\phi) = p_0(\phi) + \int_0^t \mathcal{F}_s^\phi(\overline{\mu}_s)\,\mathrm{d}s$$

for every rational t and every ϕ in a countable dense subset of $C_c^\infty(\mathbb{R}^d)$. Continuity in time and density extend this identity to every $t \in [0,T]$ and every $\phi \in C_c^\infty(\mathbb{R}^d)$. As a consequence, $\overline{\mu}$ is a weak solution of the Feynman–Kac equation (33). Uniqueness in Assumption D.4 gives $\overline{\mu} = \mu^*$ almost surely.

Every subsequence therefore has a further subsequence converging in law to the same deterministic path μ^*. Hence the full sequence converges in probability to μ^* in $C\big([0,T], \mathcal{P}(\mathbb{R}^d)\big)$. Since every $\phi \in C_b^2(\mathbb{R}^d)$ is bounded and Lipschitz, it follows that $\sup_{t\in[0,T]} \big|\hat{\mu}_t^N(\phi) - \mu_t^*(\phi)\big| \longrightarrow 0$ in probability. Moreover,

$$\sup_{t\in[0,T]} \big|\hat{\mu}_t^N(\phi) - \mu_t^*(\phi)\big| \leq 2\|\phi\|_\infty$$

shows that the sequence is uniformly integrable, and thus the claim follows. □

The following technical lemma is used in the tightness step of the preceding proof.

Lemma D.7 (Compact containment)**.** *For every $\eta > 0$, there exists a weakly compact set $\mathsf{K}_\eta \subset \mathcal{P}(\mathbb{R}^d)$ such that*

$$\liminf_{N\to\infty} \mathbb{P}\left(\hat{\mu}_t^N \in \mathsf{K}_\eta \text{ for every } t \in [0,T]\right) \geq 1 - \eta.$$

Proof. For any radius $r > 0$, we denote the open ball in $\mathbb{R}^d$ by $B_r := \{x \in \mathbb{R}^d : |x| < r\}$. Choose a radial, nondecreasing function $\chi \in C^\infty(\mathbb{R}^d; [0,1])$ such that $\chi = 0$ on B_1 and $\chi = 1$ on B_2^c, and set

$$\chi_R(x) := \chi(x/R), \qquad R > 0.$$

Notice the following properties,

$$\mathbf{1}_{B_{2R}^c} \leq \chi_R \leq \mathbf{1}_{B_R^c}, \qquad \|\nabla\chi_R\|_\infty \leq \frac{C}{R}, \qquad \|\Delta\chi_R\|_\infty \leq \frac{C}{R^2}, \tag{54}$$

for some constant $C < \infty$, as well as

$$|\mathrm{Cov}_\mu\left(g_t(\cdot,\mu),\chi_R\right)| \le 2L\mu(\chi_R), \tag{55}$$

which follows directly from $0 \le \xi_R \le 1$ and (42).

We now apply (47) with $\phi = \chi_R$, take the supremum over time and expectations. Using the estimates (51), (52), (54), as well as Cauchy-Schwarz, we arrive at

$$\begin{aligned}\mathbb{E}\left[\sup_{r\in[0,t]} \hat{\mu}_r^N(\chi_R)\right] &\le p_0(\chi_R) + C_T\left(\frac{1}{R} + \frac{1}{R^2}\right) + 2\delta_N \\ &\quad + \frac{C_T}{R}\left(\frac{1}{N} + \delta_N\right)^{1/2} + 2L\int_0^t \mathbb{E}\left[\sup_{u\in[0,s]} \hat{\mu}_u^N(\chi_R)\right] \mathrm{d}s,\end{aligned}$$

with a constant $C_T < \infty$ that may depend on T, but not on N or R. Grönwall's inequality then yields

$$\mathbb{E}\left[\sup_{t\in[0,T]} \hat{\mu}_t^N(\chi_R)\right] \le C_T\left[p_0(\chi_R) + \frac{1}{R} + \delta_N + \frac{1}{R}\left(\frac{1}{N} + \delta_N\right)^{1/2}\right], \tag{56}$$

with a different constant $C_T < \infty$, denoted by the same symbol. Since $p_0(\chi_R) \to 0$ as $R \to \infty$ and $\delta_N \to 0$ as $N \to \infty$, it follows that

$$\lim_{R\to\infty} \limsup_{N\to\infty} \mathbb{E}\left[\sup_{t\in[0,T]} \hat{\mu}_t^N(\chi_R)\right] = 0.$$

This uniform control of the probability mass at spatial infinity yields the claim by a relatively standard countable-tail construction; see, for example, Giesecke et al. (2013, Lemma 6.1). □

Proof of Corollary. Between resampling times, the proof of Theorem D.5 is unchanged. Moreover, resampling resets all weights to $\frac{1}{N}$, and hence

$$\sum_{i=1}^N \left(w_{\tau_k}^{i,N}\right)^2 = \frac{1}{N} \le \sum_{i=1}^N \left(w_{\tau_k-}^{i,N}\right)^2,$$

where $w_{\tau_k-}^{i,N}$ are the weights just before, and $w_{\tau_k}^{i,N}$ are the weights just after resampling. As a consequence, the conclusion of Lemma D.6 remains valid.

For a bounded test function ϕ, define the jump induced by the k-th resampling step by

$$\Delta_k^{N,\phi} := \hat{\mu}_{\tau_k}^N(\phi) - \hat{\mu}_{\tau_k-}^N(\phi).$$

In (47), the effect of resampling is the appearance of an additional jump term

$$J_t^{N,\phi} := \sum_{k:\,\tau_k \le t} \Delta_k^{N,\phi}. \tag{57}$$

Residual resampling is conditionally unbiased and its conditional variance is bounded by that of multinomial resampling; see Douc et al. (2005, Sections 2.2 and 3.2). Consequently,

$$\mathbb{E}\left[\Delta_k^{N,\phi} \,\middle|\, \mathcal{F}_{\tau_k^N-}\right] = 0, \qquad \mathbb{E}\left[\left|\Delta_k^{N,\phi}\right|^2 \,\middle|\, \mathcal{F}_{\tau_k^N-}\right] \le \frac{\|\phi\|_\infty^2}{N}.$$

Therefore, (57) is a martingale, and we have the estimate

$$\mathbb{E}\left[\sup_{t\in[0,T]} \left|J_t^{N,\phi}\right|^2\right] \le \frac{CK\|\phi\|_\infty^2}{N} \longrightarrow 0,$$

where the limit statement rests on the fact that K is uniformly bounded. The tightness and subsequential-identification arguments proceed exactly as in the proof of Theorem D.5, yielding the result. □

E Licenses

The following assets were used in this work.

- Boltz-2 (Passaro et al., 2025)
  MIT License
  `https://github.com/jwohlwend/boltz`
- Protenix (ByteDance et al., 2025)
  Apache-2.0 License
  `https://github.com/bytedance/Protenix`
- Guided Protein Structure Prediction (Maddipatla et al., 2025; 2026)
  `https://github.com/sai-advaith/guided_alphafold`
- Adenylate Kinase Potential of Mean Force (Beckstein et al., 2009)
  Creative Commons Attribution-ShareAlike 4.0 International License
  `https://becksteinlab.physics.asu.edu/research/52/adk-apo-pmf`
- 4OLE PDB entry (Joint Center for Structural Genomics (JCSG), 2014)
  CC0 1.0 Universal (CC0 1.0) Public Domain Dedication
  `https://www.rcsb.org/structure/4OLE`

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In our paper, we provide a mean-field framework for distributional control of diffusion models, and empirically evaluate its behaviour. Our proposed framework and methodology is outlined in Section 3, with experimental validation in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a dedicated limitations section in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Proofs of our theoretical results are included in Appendix D.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Detailed descriptions of method implementation details are given in Appendix B, and detailed descriptions of the experimental settings are provided in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include code for the low-dimensional experiments in Section 4.1 in the supplementary material. For our protein experiments, we adapt this codebase to the Boltz-2 codebase.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We include all experimental details in Appendix C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We report error bars in some plots (such as Figure 11), but it is not computationally feasible to include in all our plots. We also note in the Appendix that most of the metrics are very consistent across runs due to the large batch size.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include these details in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the Code of Ethics and confirm that our work conforms to these guidelines. We see no potential harmful consequences of our work, and include extensive reproducibility details in Appendix C.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work is mostly theory and methodology-focused. We do not see any immediate societal impacts, though certain applications may share similar societal consequences as in flow-based generative models upon which our approach is based.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the works that we use, and include the licenses of code and datasets in Appendix E.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.