

DRIFTLITE: LIGHTWEIGHT DRIFT CONTROL FOR INFERENCE-TIME SCALING OF DIFFUSION MODELS

Yinuo Ren¹, Wenhao Gao², Lexing Ying^{3,1}, Grant M. Rotskoff^{2,1}, Jiequn Han^{4*}

¹ Institute for Computational and Mathematical Engineering, Stanford University

² Department of Chemistry, Stanford University

³ Department of Mathematics, Stanford University

⁴ Center for Computational Mathematics, Flatiron Institute

{yinuoren, whgao, lexing, rotskoff}@stanford.edu

*Correspondence to: Jiequn Han <jhan@flatironinstitute.org>

ABSTRACT

We study inference-time scaling for diffusion models, where the goal is to adapt a pre-trained model to new target distributions without retraining. Existing guidance-based methods are simple but introduce bias, while particle-based corrections suffer from weight degeneracy and high computational cost. We introduce *DriftLite*, a lightweight, training-free particle-based approach that steers the inference dynamics on the fly with provably optimal stability control. DriftLite exploits a fundamental degree of freedom in the Fokker-Planck equation between the drift and particle potential, and yields two practical instantiations: *Variance- and Energy-Controlling Guidance (VCG/ECG)* for approximating the optimal drift with modest and scalable overhead. Across Gaussian mixture models, particle systems, and large-scale protein-ligand co-folding problems, DriftLite consistently reduces variance and improves sample quality over pure guidance and sequential Monte Carlo baselines. These results highlight a principled, efficient route toward scalable inference-time adaptation of diffusion models. Our source code is publicly available at <https://github.com/yinuoren/DriftLite>.

1 INTRODUCTION

Diffusion- (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song & Ermon, 2019; Song et al., 2020) and flow-based (Zhang et al., 2018; Lipman et al., 2022; Albergo & Vanden-Eijnden, 2022; Liu et al., 2022; Ren et al., 2025b) models have revolutionized generative modeling, achieving state-of-the-art performance in domains ranging from creative media synthesis (Rombach et al., 2022; Le et al., 2023; Ho et al., 2022; Austin et al., 2021) to fundamental scientific discovery (Xu et al., 2022; Watson et al., 2023; Duan et al., 2023; Gao et al., 2024; Zhu et al., 2024; Zeni et al., 2025; Duan et al., 2025). They typically rely on a neural network to approximate a time-dependent vector field, which guides a stochastic process from noises to a complex target. However, training is resource-intensive, making it impractical to retrain from scratch for every new setting. This renders a lightweight adaptation of pre-trained models to target distributions that is both compelling and essential.

To this end, a spectrum of adaptation methods has emerged. At one end are guidance-based techniques, the most popular and straightforward inference-time techniques, which inject new information into the drift term, such as classifier (Dhariwal & Nichol, 2021) or classifier-free guidance (Ho & Salimans, 2022) and its many variants (Chung et al., 2022; Trippe et al., 2022; Bansal et al., 2023; Song et al., 2023a;b; He et al., 2023; Guo et al., 2024; Zheng et al., 2024; Rojas et al., 2025). While simple and effective for many tasks, these methods are often heuristic and introduce uncontrolled bias (Chidambaram et al., 2024; Wu et al., 2024a), a significant drawback for scientific applications where sampling accuracy is paramount. On the opposite are methods that resort to extra training, such as fine-tuning (Fan & Lee, 2023; Fan et al., 2023; Black et al., 2023; Clark et al., 2023; Wallace et al., 2024) as in the LLM context (Ouyang et al., 2022; Rafailov et al., 2023), learning within a stochastic control framework (Domingo-Enrich et al., 2024a; Uehara et al., 2024; Thornton et al., 2025), similar to learning-based samplers (Zhang & Chen, 2021; Vargas et al., 2023a;b;

Domingo-Enrich et al., 2024b; Richter & Berner, 2023; Zhu et al., 2025), or adding additional training objectives (Venkatraman et al., 2024; Thornton et al., 2025), but this shifts the computational burden back to retraining, forfeiting the efficiency of inference-time approaches.

Between these ends lies a middle ground of training-free but more sophisticated inference-time approaches. A promising direction formulates the problem in Bayesian and Monte Carlo sampling frameworks (Du et al., 2023; Xu & Chi, 2024; Wu et al., 2024b; Coeurdoux et al., 2024; Bruna & Han, 2024; Zheng et al., 2025). In particular, Sequential Monte Carlo (SMC) methods (Del Moral et al., 2006; Doucet et al., 2000) have been recently introduced to correct for the bias of guidance by simulating the target dynamics with weighted particles (Wu et al., 2023; Cardoso et al., 2023; Skreta et al., 2025; Chen et al., 2025; Singhal et al., 2025; Lee et al., 2025). Despite their strong theoretical grounding and asymptotic guarantees, these particle-based methods face a critical practical bottleneck: *weight degeneracy*. As the simulation progresses, the weights of a few particles grow exponentially while the rest decay, causing the effective sample size to collapse. To mitigate this, one may increase the number of particles, raising computational cost, or use fewer particles, resulting in instability and degraded sample quality.

Our work introduces *DriftLite*, a lightweight, training-free inference-time scaling approach that resolves the inherent instability of particle-based methods without sacrificing mathematical rigor. By exploiting a fundamental degree of freedom in the Fokker-Planck equation, we actively control particle drift on the fly. This proactive steering mechanism absorbs sources of weight variation, preventing the weight collapse common in passive reweighting schemes and dramatically improving stability. The method’s modest and scalable computational overhead requiring only the solution of a small linear system per step, makes it fundamentally lightweight, especially compared with methods that require retraining or extra training of additional control networks. Unlike learning-based control (Vargas et al., 2023b; Richter & Berner, 2023) or heuristic control frameworks (He et al., 2025), DriftLite is a **training-free** solution derived directly from the principle of variance reduction. Comparing with computationally intensive PDE solvers (Albergo & Vanden-Eijnden, 2024) and trajectory balance-based GFlowNet methods (Bengio et al., 2023; Sendera et al., 2024; Venkatraman et al., 2024), our method is applied **on the fly** without any parameter updating or backpropagating through the diffusion model. It is designed to scalably match an entire target distribution in high-dimensional, continuous systems, more rigorous than targeting sample-focused metrics (Ma et al., 2025) or solving problems in simpler discrete settings (Chertkov et al., 2025).

Our Contributions. Building on this insight, our work makes the following contributions:

- We formulate and exploit a fundamental degree of freedom in the Feynman-Kac-type Fokker-Planck equation (2.5), establishing a principled trade-off between the particle drift and the reweighting potential, and show that it can be directly exploited to actively minimize particle weight variance.
- We introduce *DriftLite*, a lightweight and training-free framework that computes a control drift on-the-fly to stabilize the sampling dynamics. We derive two practical instantiations, Variance-Controlling Guidance (VCG) and Energy-Controlling Guidance (ECG), which are computationally efficient and require solving only a small additional linear system at each time step.
- We conduct extensive experiments on challenging benchmarks, including high-dimensional Gaussian mixture models, molecular particle systems, and large-scale protein-ligand co-folding. Our results demonstrate that DriftLite substantially reduces weight variance, stabilizes the Effective Sample Size (ESS), and improves final sample quality over current baselines.

2 PRELIMINARIES

In this section, we establish the problem setting, including the fundamentals of diffusion models and the inference-time scaling tasks central to our study.

2.1 DIFFUSION MODELS

We begin with a *pre-trained* diffusion or flow-matching model, to which we refer as the base model. This model defines both a forward process $(\mathbf{x}_s)_{s \in [0, T]}$ governed by the following stochastic differential equation (SDE) and Fokker-Planck (FP) equation:

$$d\mathbf{x}_s = \mathbf{u}_s(\mathbf{x}_s)ds + U_s d\mathbf{w}_s \quad (\text{SDE}), \quad \partial_s p_s(\mathbf{x}) = -\nabla \cdot [p_s(\mathbf{x})\mathbf{u}_s(\mathbf{x})] + \frac{U_s^2}{2} \Delta p_s(\mathbf{x}) \quad (\text{FP}), \quad (2.1)$$

where \mathbf{u}_s is the forward drift, p_s is the marginal distribution at time s , and $(\mathbf{w}_s)_{s \geq 0}$ is a Wiener process. p_0 represents the data distribution, and p_T is a simple prior, typically a standard Gaussian.

Generative modeling is performed using the backward process. Letting $t = T - s$ be the reverse time and denote $\hat{*}_t = *_{T-t}$, the backward process $(\hat{\mathbf{x}}_t)_{t \in [0, T]}$ is then described by:

$$d\hat{\mathbf{x}}_t = \mathbf{v}_t(\hat{\mathbf{x}}_t)dt + V_t d\mathbf{w}_t \text{ (SDE)}, \quad \partial_t \hat{p}_t(\mathbf{x}) = -\nabla \cdot [\hat{p}_t(\mathbf{x})\mathbf{v}_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta \hat{p}_t(\mathbf{x}) \text{ (FP)},$$

where \mathbf{v}_t is the backward drift. The process starts from the noise distribution $\hat{p}_0 \approx p_0$ and recovers the data distribution $\hat{p}_T = p_0$. In traditional diffusion models, the backward drift \mathbf{v}_t is related to the forward drift $\mathbf{u}_s(\mathbf{x}_s) = -F_s \mathbf{x}_s$ via the score function $\nabla \log \hat{p}_t$:

$$\mathbf{v}_t(\mathbf{x}) = -\hat{\mathbf{u}}_t(\mathbf{x}) + \frac{\hat{U}_t^2 + V_t^2}{2} \nabla \log \hat{p}_t(\mathbf{x}). \quad (2.2)$$

The word “pre-trained” signifies that we have access to the forward drift \mathbf{u}_s and a reliable NN approximation of the score $\nabla \log \hat{p}_t$, which in turn defines the backward drift \mathbf{v}_t .

2.2 INFERENCE-TIME SCALING

Our goal is to adapt the generative process of a pre-trained model to new, related tasks at inference time. This approach avoids the significant computational cost and data requirements of retraining from scratch, making it desirable to leverage existing models. We focus on two primary scenarios:

- Annealing: Given a factor γ , the goal is to sample from $q_T \propto p_0^\gamma$. This is common in physics for generating low-temperature samples concentrated around primary modes of a distribution (Karczewski et al., 2024), using a model trained on easier-to-obtain high-temperature data.
- Reward-Tilting: Given a reward function $r(\mathbf{x})$, the goal is to sample from $q_T \propto p_0 \exp(r)$. This can be interpreted as posterior sampling with p_0 being the prior and the reward r being the posterior likelihood. It is widely used in applications, such as inverse design (Chung et al., 2022), where the reward function encodes the desired properties of the generated samples.

In this work, we mainly focus on the settings where the reward r is twice-differentiable. In scientific applications, it is typically a physics-based energy with analytic gradients (e.g., Passaro et al. (2025)), which guarantees cheap access to derivatives at inference time. For black-box rewards, one could in principle approximate derivatives using stochastic estimators.

Distribution Path Selection. We can unify both scenarios by defining the target compactly as

$$q_T(\mathbf{x}) \propto \hat{p}_T(\mathbf{x})^\gamma \exp(r(\mathbf{x})) = p_0(\mathbf{x})^\gamma \exp(r(\mathbf{x})).$$

To sample from q_T , we define a modified backward process that evolves along a path of distributions $(q_t)_{t \in [0, T]}$ that smoothly connects from initial noise to our target q_T . Following recent works (Skreta et al., 2025; Chen et al., 2025), we adopt a both conceptually and computationally simple path:

$$q_t(\mathbf{x}) \propto \hat{p}_t(\mathbf{x})^\gamma \exp(r_t(\mathbf{x})),$$

where the reward r_t interpolates from an initial state r_0 chosen such that q_0 is easy to sample from, to the final reward $r_T = r$. While more complex paths can be learned via optimal control (Liu et al., 2025), we focus on such pre-defined paths to maintain a training-free framework.

Guidance-Based Dynamics. A common and intuitive approach, to which we refer as *pure guidance* (Nichol et al., 2021; Ho & Salimans, 2022), is to inject the new information directly into the drift term by replacing the original score $\nabla \log \hat{p}_t$ with a heuristic score $\nabla \log q_t$ corresponding to the marginal q_t , leading to the following Fokker-Planck equation:

$$\partial_t q_t(\mathbf{x}) = -\nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x})q_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}), \quad (2.3)$$

where the modified drift $\tilde{\mathbf{v}}_t$ is defined below (cf., Eqn. (2.2)):

$$\tilde{\mathbf{v}}_t(\mathbf{x}) = -\hat{\mathbf{u}}_t(\mathbf{x}) + \frac{\hat{U}_t^2 + V_t^2}{2} (\gamma \nabla \log \hat{p}_t(\mathbf{x}) + \nabla r_t(\mathbf{x})). \quad (2.4)$$

However, this method is known to be intrinsically biased because it fails to account for the changing normalization constant of q_t over time (Chidambaram et al., 2024). To correct this bias, the true dynamics must include a self-normalizing reweighting term, as formalized below.

Proposition 2.1 (Guidance-Based Dynamics). *The exact time evolution of the density $(q_t)_{t \in [0, T]}$ follows the following Feynman-Kac-type Fokker-Planck equation:*

$$\partial_t q_t(\mathbf{x}) = -\nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) + q_t(\mathbf{x}) g_t(\mathbf{x}), \quad (2.5)$$

where $\tilde{\mathbf{v}}_t$ is the same drift as in pure guidance (2.4), and the reweighting potential $g_t(\mathbf{x}) = G_t(\mathbf{x}) - \mathbb{E}_{q_t}[G_t(\cdot)]$ is given by:

$$G_t = \dot{r}_t - (1-\gamma) \nabla \cdot \tilde{\mathbf{u}}_t + \frac{\tilde{U}_t^2}{2} (\Delta r_t - \gamma(1-\gamma) \|\nabla \log \tilde{p}_t\|^2) + \nabla r_t^\top \left(-\tilde{\mathbf{u}}_t + \gamma \tilde{U}_t^2 \nabla \log \tilde{p}_t + \frac{\tilde{U}_t^2}{2} \nabla r_t \right).$$

We refer readers to App. A.1 for the proof. The PDE describes dynamics that diffuse with the guidance drift $\tilde{\mathbf{v}}_t$, while densities continuously reweight according to the centered potential g_t .

Weighted Particle Method. The corrected PDE (2.5) can be simulated using Sequential Monte Carlo (SMC) (Doucet et al., 2000; Del Moral et al., 2006), where the density q_t is approximated by an empirical distribution formed by an ensemble of N weighted particles $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i \in [N]}$:

$$\begin{cases} d\mathbf{x}_t^{(i)} = \tilde{\mathbf{v}}_t(\mathbf{x}_t^{(i)}) dt + V_t d\mathbf{w}_t^{(i)}, & i \in [N], \\ d\log w_t^{(i)} = \hat{g}_t(\mathbf{x}_t^{(i)}) := G_t(\mathbf{x}_t^{(i)}) - \sum_{i=1}^N w_t^{(i)} G_t(\mathbf{x}_t^{(i)}), & i \in [N]. \end{cases} \quad (2.6)$$

We refer to this baseline as Guidance-SMC (G-SMC) (Skreta et al., 2025; Chen et al., 2025). This method is provably convergent, with the KL divergence to the target scaling as $\mathcal{O}(N^{-1})$ in the diffusion context (Andrieu et al., 2018; Huggins & Roy, 2019; Domingo-Enrich et al., 2020; Cardoso et al., 2023; Chen et al., 2025). A brief justification of this method is given in App. A.2.

3 METHOD: LIGHTWEIGHT DRIFT CONTROL

While the principled dynamics outlined in Prop. 2.1 offer a path to unbiased sampling, their reliance on weighted particles introduces the critical vulnerability of weight degeneracy. As the simulation progresses, the exponential dependency of the weights w on the potential g_t leads to rapid weight degeneracy and collapse of the effective sample size. This instability makes the standard Guidance-SMC approach computationally inefficient, especially with a limited number of particles.

This section introduces our solution: *DriftLite*, a lightweight, training-free framework that actively controls the drift to stabilize the weights. We develop in three steps: (1) we formulate a fundamental degree of freedom in the governing Fokker-Planck equation (2.5), (2) we exploit this freedom to formulate an objective for minimizing the variance of the reweighting potential g_t , and (3) we derive two practical, computationally efficient algorithms (VCG and ECG) for achieving this control.

3.1 DEGREE OF FREEDOM IN THE FOKKER-PLANCK EQUATION

Our key insight is that we can dynamically modify the particle SDE to counteract the sources of weight variance. Instead of passively reweighting particles, we can proactively steer them by “off-loading” the problematic parts of the potential g_t into a new, corrective drift term. This is enabled by a degree of freedom within the Fokker-Planck equation, which we formalize below.

Proposition 3.1 (Degree of Freedom). *For any control drift $\mathbf{b}_t(\mathbf{x})$, the Feynman-Kac-type Fokker-Planck equation (2.5) is equivalent to:*

$$\partial_t q_t(\mathbf{x}) = -\nabla \cdot [(\tilde{\mathbf{v}}_t(\mathbf{x}) + \mathbf{b}_t(\mathbf{x})) q_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) + q_t(\mathbf{x}) \phi_t(\mathbf{x}), \quad (3.1)$$

where the residual potential is $\phi_t(\mathbf{x}) = g_t(\mathbf{x}) + h_t(\mathbf{x}; \mathbf{b}_t(\mathbf{x}))$ with control potential h_t being:

$$h_t(\mathbf{x}; \mathbf{b}_t) = (\gamma \nabla \log \tilde{p}_t(\mathbf{x}) + \nabla r_t(\mathbf{x})) \cdot \mathbf{b}_t(\mathbf{x}) + \nabla \cdot \mathbf{b}_t(\mathbf{x}).$$

Proof Sketch. The core of the proof is detailed in App. A.3. Briefly, we have $-\nabla \cdot (\mathbf{b}_t(\mathbf{x}) q_t(\mathbf{x})) + q_t(\mathbf{x}) h_t(\mathbf{x}; \mathbf{b}_t) = 0$, since $h_t(\mathbf{x}; \mathbf{b}_t)$ is constructed using $\nabla \log q_t = \gamma \nabla \log \tilde{p}_t + \nabla r_t$. An important property is that the correction term has zero expectation under q_t , i.e., $\mathbb{E}_{q_t}[h_t(\cdot; \mathbf{b}_t)] = 0$. \square

This proposition provides a powerful tool: we can introduce any control drift \mathbf{b}_t to alter the dynamics, as long as it is compensated by an extra control potential $h_t(\cdot; \mathbf{b}_t)$. Since a large variance in

the potential g_t is the direct cause of weight degeneracy, our goal is to choose \mathbf{b}_t strategically to minimize the variance of the new residual potential ϕ_t . An ideal control would make ϕ_t constant, completely stabilizing the particle weights. In fact, a perfect, variance-eliminating control always exists for any base potential g_t , as shown in the following proposition:

Proposition 3.2 (Optimal Control, Informal Version). *There exists a unique curl-free control $\mathbf{b}_t^*(\mathbf{x}) = \nabla A_t^*(\mathbf{x})$ such that $\phi_t^*(\mathbf{x}) = g_t(\mathbf{x}) + h_t(\mathbf{x}; \mathbf{b}_t^*) = 0$ for all \mathbf{x} , where the optimal scalar potential $A_t^*(\mathbf{x})$ is the solution to the following Poisson equation:*

$$\nabla \cdot (q_t(\mathbf{x}) \nabla A_t^*(\mathbf{x})) = -q_t(\mathbf{x}) g_t(\mathbf{x}). \quad (3.2)$$

The proof and further discussion are provided in App. A.4. Intuitively, Eqn. (3.2) follows by noticing that the control potential satisfies $q_t(\mathbf{x}) h_t(\mathbf{x}; \mathbf{b}_t) = \nabla \cdot (q_t(\mathbf{x}) \mathbf{b}_t(\mathbf{x}))$ from its definition.

This mechanism is closely related to twisted proposals in SMC (Briers et al., 2010; Whiteley & Lee, 2014; Heng et al., 2020) and its recent applications (Lawson et al., 2022; Zhao et al., 2024; Lu & Wang, 2024) and Prop. 3.1 can be viewed as a diffusion formulation of this degree of freedom.

3.2 IN SEARCH OF OPTIMAL CONTROL

While Prop. 3.2 guarantees a perfect solution, solving the high-dimensional PDE in (3.2) at every time step is computationally intractable. In contrast to learning-based methods that uses neural networks to approximate the optimal control via backpropagation (Albergo & Vanden-Eijnden, 2024; Vargas et al., 2023b), we propose two training-free, practical methods that approximate this optimal control by balancing effectiveness with efficiency. Both methods share a core strategy: restricting the search for the control drift \mathbf{b}_t to a finite-dimensional subspace. This simplification is key, as it transforms the complex problem of minimizing the residual potential ϕ_t into solving a small linear system. This reduction from an intractable PDE to a tractable linear solve makes the control truly lightweight, hence the name *DriftLite*.

Variance-Controlling Guidance (VCG). The most direct approach is to find a control \mathbf{b}_t that explicitly minimizes the variance of the residual potential:

$$\min_{\mathbf{b}_t} \text{Var}_{\mathbf{x} \sim q_t} [\phi_t(\mathbf{x})] = \text{Var}_{\mathbf{x} \sim q_t} [g_t(\mathbf{x}) + h_t(\mathbf{x}; \mathbf{b}_t)]. \quad (3.3)$$

Instead of parameterizing \mathbf{b}_t with a neural network (Albergo & Vanden-Eijnden, 2024), we seek a lightweight solution by approximating it as a linear combination of basis functions.

Ansatz 3.3 (Linear Control Drift). *The optimal control drift $\mathbf{b}_t^*(\mathbf{x})$ is approximated as $\mathbf{b}_t(\mathbf{x}) = \sum_{i=1}^n \theta_t^i \mathbf{s}_i(\mathbf{x})$, where $\{\mathbf{s}_i(\mathbf{x})\}_{i \in [n]}$ are pre-defined vector bases and $\boldsymbol{\theta}_t = (\theta_t^1, \dots, \theta_t^n)^\top$ are the coefficients to be found.*

Under this ansatz, the residual potential becomes $\phi_t(\mathbf{x}) = g_t(\mathbf{x}) + \sum_{i=1}^n \theta_t^i h_t^i(\mathbf{x})$, where $h_t^i(\mathbf{x}) = h_t(\mathbf{x}; \mathbf{s}_i)$. The objective (3.3) corresponds to a standard least-square problem, whose solution is obtained by solving an $n \times n$ linear system $\mathbf{A}_t \boldsymbol{\theta}_t = \mathbf{c}_t$, where $\mathbf{A}_{ij} = \mathbb{E}_{q_t} [h_t^i h_t^j]$ and $\mathbf{c}_i = -\mathbb{E}_{q_t} [g_t h_t^i]$.

Energy-Controlling Guidance (ECG). An alternative approach directly targets the curl-free optimal control \mathbf{b}_t^* in Prop. 3.2 by variationally solving the Poisson equation (3.2). As shown by Yu & E (2018), this equation is the Euler-Lagrange equation for the following energy functional:

$$\min_{A_t} \mathcal{E}_t[A_t] = \int \left(\frac{1}{2} q_t(\mathbf{x}) \|\nabla A_t(\mathbf{x})\|^2 - q_t(\mathbf{x}) g_t(\mathbf{x}) A_t(\mathbf{x}) \right) d\mathbf{x}. \quad (3.4)$$

We can efficiently find an approximate minimizer using the Ritz method for the scalar potential A_t .

Ansatz 3.4 (Linear Control Potential). *The optimal scalar potential $A_t^*(\mathbf{x})$ is approximated as $A_t(\mathbf{x}) = \sum_{i=1}^n \theta_t^i s_t^i(\mathbf{x})$, where $\{s_t^i(\mathbf{x})\}_{i \in [n]}$ are scalar bases. The control drift is then given by $\mathbf{b}_t(\mathbf{x}) = \nabla A_t(\mathbf{x}) = \sum_{i=1}^n \theta_t^i \nabla s_t^i(\mathbf{x})$.*

Substituting into the energy functional (3.4) again yields a linear system of equations $\mathbf{A}_t \boldsymbol{\theta}_t = \mathbf{c}_t$, where $\mathbf{A}_{ij} = \mathbb{E}_{q_t} [\nabla s_t^i \top \nabla s_t^j]$ and $\mathbf{c}_i = \mathbb{E}_{q_t} [g_t s_t^i]$.

3.3 PRACTICAL IMPLEMENTATION

Choice of Bases. The effectiveness of VCG and ECG depends on the choice of suitable basis functions. While the formal solution for the optimal control b_t^* is intractable (*cf.*, App. A.4), its structure reveals that the ideal control is a function of temporally locally available quantities like the score $\nabla \log \tilde{p}_t$, the reward gradient ∇r_t , and the potential g_t (containing the forward drift \tilde{u}_t and higher-order terms). This suggests that a natural low-rank approximation is obtained by projecting the intractable optimal control onto the span of locally available vector fields, as defined in the following.

- Variance-Controlling Guidance (VCG): We use the following vector basis functions:

$$s_1(\mathbf{x}) = \nabla r_t(\mathbf{x}), \quad s_2(\mathbf{x}) = \nabla \log \tilde{p}_t(\mathbf{x}), \quad s_3(\mathbf{x}) = \tilde{u}_t(\mathbf{x}).$$

Note that using s_2 requires computing the Laplacian $\Delta \log \tilde{p}_t(\mathbf{x})$, which can be approximated efficiently with Hutchinson’s trace estimator (Hutchinson, 1989) in high dimensions.

- Energy-Controlling Guidance (ECG): We use the corresponding scalar potentials:

$$s_1(\mathbf{x}) = r_t(\mathbf{x}), \quad s_2(\mathbf{x}) = \log \tilde{p}_t(\mathbf{x}), \quad s_3(\mathbf{x}) = \tilde{U}_t(\mathbf{x}),$$

where \tilde{U}_t is a potential such that $\nabla \tilde{U}_t = \tilde{u}_t$. This method is especially convenient when the log-likelihood $\log \tilde{p}_t$ is readily available from upstream training tasks (Akhound-Sadegh et al., 2025; Guth et al., 2025; Thornton et al., 2025). If not, approximations or alternative bases may be used, such as the score norm $\|\nabla \log \tilde{p}_t\|^2$ or random projections of the score $\nabla \log \tilde{p}_t \cdot \xi$ for random ξ .

For annealing tasks, reward-based bases (s_1 and s_3) are automatically dismissed.

Weighted Particle Simulation. As discussed in Sec. 2.2, we simulate the Feynman-Kac-type Fokker-Planck equation (3.1) using the SMC/weighted particle method detailed in Alg. 1. The key difference from G-SMC Eqn. (2.6) is the use of the controlled drift $\tilde{v}_t + b_t$ and the residual potential $\phi_t = g_t + h_t(\cdot; b_t)$. To prevent weight collapse, particles are resampled when the Effective Sample Size (ESS) drops below a threshold τ . These principled versions with resampling are denoted VCG-SMC/ECG-SMC. For high-dimensional problems where resampling introduces additional stochastic instability, we also consider simpler variants, denoted VCG/ECG, which use the low-variance residual potential ϕ_t , retain continuous path-level weights, but omit resampling steps.

Our method adds modest computational overhead. The linear solve itself is negligible, while the dominant additional cost comes from evaluating basis functions and building a small $n \times n$ linear system at each time step, where n is the number of bases, typically $n \leq 3$ in our experiments. The components of this system (A_t and c_t) are computed as expectations over the current weighted particles, reusing terms like the score $\nabla \log \tilde{p}_t$ and the reward gradient ∇r_t that are already computed for the base guidance drift. While accurate evaluation of the score Laplacian $\Delta \log \tilde{p}_t$ can improve control quality, efficiency is preserved with stochastic approximations, and thus the per-step overhead remains constant in dimension and fully parallelizable across particles, resulting in moderate runtime increase compared to the pure guidance baseline (*cf.*, empirical results in Tabs. 5 and 8).

4 EXPERIMENTS

In this section, we empirically test the performance of DriftLite by designing a series of challenging annealing and reward-tilting tasks, comparing our DriftLite methods (VCG and ECG with and without SMC) against two key baselines: Pure Guidance (PG) (2.3) (Ho & Salimans, 2022), Guidance-SMC (G-SMC) (2.5) (Skreta et al., 2025; Chen et al., 2025). Our implementation uses JAX (Bradbury et al., 2018) to ensure efficient, parallelized computation on GPUs. Our source code is publicly available at <https://github.com/yinuoren/DriftLite>.

4.1 GAUSSIAN MIXTURE MODEL

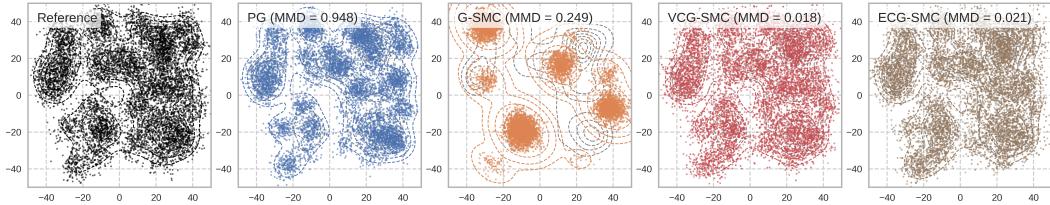
We begin with a 30-dimensional Gaussian Mixture Model (GMM) (*cf.*, App. B.1 for detailed settings), a controlled environment where the exact score $\nabla \log p_t$ and the potential $\log p_t$ are known analytically, allowing us to isolate and evaluate the performance of the sampling algorithms themselves, free from any confounding errors of a learned score network. We evaluate the methods with multiple metrics, including the Negative Log-Likelihood difference (ΔNLL), Maximum Mean Discrepancy (MMD), and Sliced Wasserstein Distance (SWD) (*cf.*, App. B.5).

Algorithm 1: DriftLite-VCG/ECG-SMC Implementation

Input: Original drift path \mathbf{v}_t , original potential path g_t , time steps $\{t_k\}_{k=0}^M$, reward $r(\mathbf{x})$, schedule β_t , basis functions, number of particles N , ESS threshold τ .

- 1 Initialize particles $\mathbf{x}_0^{(i)} \sim \tilde{p}_0$ and weights $w_0^{(i)} \leftarrow \frac{1}{N}$ for $i = 1, \dots, N$;
- 2 **for** $k \leftarrow 0$ to $M - 1$ **do**
- 3 Form weighted estimates of \mathbf{A}_{t_k} and \mathbf{c}_{t_k} using $\{(\mathbf{x}_{t_k}^{(i)}, w_{t_k}^{(i)})\}_{i \in [N]}$;
- 4 Solve $\mathbf{A}_{t_k} \boldsymbol{\theta}_{t_k} = \mathbf{c}_{t_k}$ to obtain the control drift $\mathbf{b}_{t_k}(\cdot)$;
- 5 $\mathbf{v}_{t_k}(\cdot) \leftarrow \mathbf{v}_{t_k}(\cdot) + \mathbf{b}_{t_k}(\cdot)$, $g_{t_k}(\cdot) \leftarrow g_{t_k}(\cdot) + h_{t_k}(\cdot; \mathbf{b}_{t_k})$;
- 6 $\log w_{t_{k+1}}^{(i)} \leftarrow \log w_{t_k}^{(i)} + g_{t_k}(\mathbf{x}_{t_k}^{(i)})(t_{k+1} - t_k)$, $\mathbf{w}_{t_{k+1}} \leftarrow \text{softmax}(\mathbf{w}_{t_k})$;
- 7 $\mathbf{x}_{t_{k+1}}^{(i)} \leftarrow \mathbf{x}_{t_k}^{(i)} + \mathbf{v}_{t_k}(\mathbf{x}_{t_k}^{(i)})(t_{k+1} - t_k) + V_{t_k} \sqrt{t_{k+1} - t_k} \mathbf{z}^{(i)}$, where $\mathbf{z}^{(i)} \sim \mathcal{N}(0, \mathbf{I})$;
- 8 **if** $ESS(\mathbf{w}_{t_{k+1}}) < \tau$ **or periodically then**
- 9 Resample $\{\mathbf{x}_{t_{k+1}}^{(i)}\}_{i \in [N]}$ according to $\{w_{t_{k+1}}^{(i)}\}_{i \in [N]}$ and reset $w_{t_{k+1}}^{(i)} \leftarrow \frac{1}{N}$ for all i ;

Output: Final samples $\{\mathbf{x}_T^{(i)}, w_T^{(i)}\}_{i \in [N]}$ from the last completed pass.

Figure 1: Qualitative comparison of sampling methods on the GMM annealing task ($\gamma = 2.5$).

Annealing. We first test the ability to sharpen the GMM’s modes by annealing, which tests each method’s ability to maintain the correct relative mode weights. As shown in Fig. 1, the pure guidance (PG) method produces visibly biased samples, while G-SMC suffers from mode collapse, a direct consequence of the weight degeneracy that our work aims to solve. In contrast, our methods (VCG and ECG) accurately sample from the correct modes, also corroborated with quantitative comparisons in Tab. 3. A closer look at the ESS and potential variance evolution during the inference dynamics Fig. 2 reveals why DriftLite succeeds. Our control mechanism reduces the variance of the reweighting potential by several orders of magnitude compared to G-SMC. This directly prevents weight degeneracy, leading to a stable Effective Sample Size (ESS) throughout the simulation and superior final sample quality. Notably, ECG, while not directly minimizing variance, achieves a similar stabilizing effect, validating the energy-based control perspective. Fig. 3 shows the performance of all methods as the number of particles varies. It indicates that our methods not only outperform the baselines but also converge more efficiently, achieving better results with fewer particles.

Reward-Tilting. The results of the reward-tilting task where the distribution is shifted towards a region defined by a quadratic reward (Figs. 4, 9 and 10 and Tab. 4) confirm our findings from the annealing task. We refer to App. C.1 for further experimental results.

Learning-Based Baseline. To contrast DriftLite’s training-free inference-time control with amortized drift-learning, we additionally implement a training-based baseline, *Neural Controlling Guidance (NCG)*, which parameterizes the control drift by a neural network and optimizes the variance objective (3.3) via backpropagation. See App. C.1 for full setup details and quantitative results.

Iterative Refinement. Furthermore, we introduce an iterative refinement procedure, where the learned control drift $\tilde{\mathbf{v}}_t + \mathbf{b}_t$ and potential $\phi_t = g_t + h_t(\cdot; \mathbf{b}_t)$ from one full pass are used as the base dynamics for the next. As further discussed in App. C.4, this process progressively reduces variance and stabilizes ESS over multiple rounds (*cf.*, Figs. 17 and 18), further enhancing sample quality (*cf.*, Tabs. 15 and 16). The monotonic decrease in variance across rounds also acts as a proxy for the reduction of approximation error in the linear control ansatz (Thms. 3.3 and 3.4).

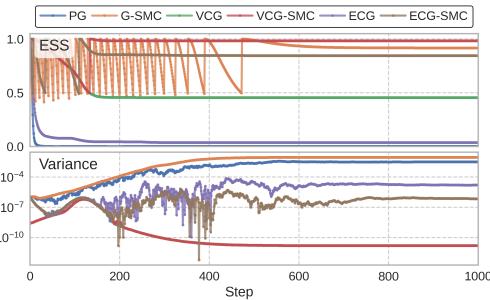


Figure 2: Evolution of ESS and potential variance during inference on the GMM annealing task ($\gamma = 2.2$). Our methods (VCG/ECG) substantially reduce variance and stabilize ESS.

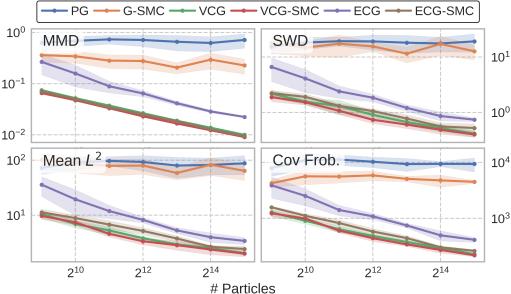


Figure 3: Performance metrics versus number of particles for the GMM annealing task ($\gamma = 2.0$). Our methods consistently outperform baselines and show strong scaling.

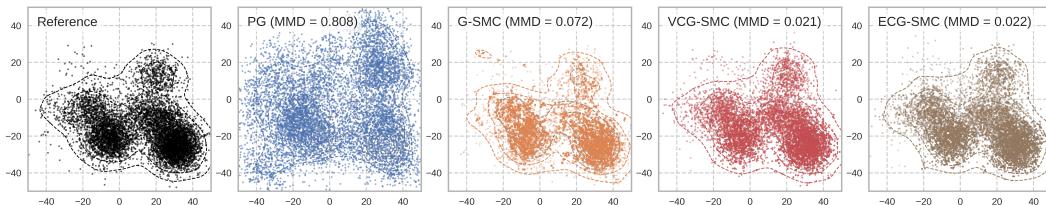


Figure 4: Qualitative comparison of sampling methods on the GMM reward-tilting task ($\sigma = 200.0$).

4.2 PARTICLE SYSTEMS

Next, we move to more realistic scientific benchmarks where the score is approximated by an NN trained on finite data. We evaluate on two standard systems with complex, multimodal energy landscapes: a 2D 4-particle Double-Well (DW-4) and a 3D 13-particle Leonard-Jones system (LJ-13), both widely used as benchmarks (Klein et al., 2023; Akhoud-Sadegh et al., 2024; 2025; Liu et al., 2025; Skreta et al., 2025; Zhang et al., 2025).

The score is obtained by training an $E(n)$ -Equivariant Graph Neural Network (EGNN) (Satorras et al., 2021b) (cf., App. B.2). For all particle systems, ground-truth reference is obtained from under-damped Langevin dynamics simulations with BAOAB splitting scheme (Leimkuhler & Matthews, 2013) (cf., App. B.3). The EDM framework (Karras et al., 2022) is adopted for both training and inference (cf., App. B.4). We measure performance using additional metrics that capture physical correctness, including the Radial Distribution Function (RDF) for structure and the energy distribution for thermodynamics (cf., App. B.5). Based on the GMM results showing VCG’s superior performance over ECG and the lack of pre-trained log-likelihood, we proceed with only the VCG variants of DriftLite in the following experiments.

Double-Well-4 (DW-4). We first consider the DW-4 system (cf., App. B.1). This system features two energy minima separated by a barrier. The annealing task requires the sampler to correctly populate both modes, even when they are sharpened at low temperatures. As shown in Fig. 13, VCG-SMC achieves a nearly perfect match with the ground-truth RDF and energy distribution. This demonstrates that by using variance reduction to maintain an ensemble of high-quality particles, DriftLite effectively leverages global information to navigate challenging energy landscapes where baselines fail to do so.

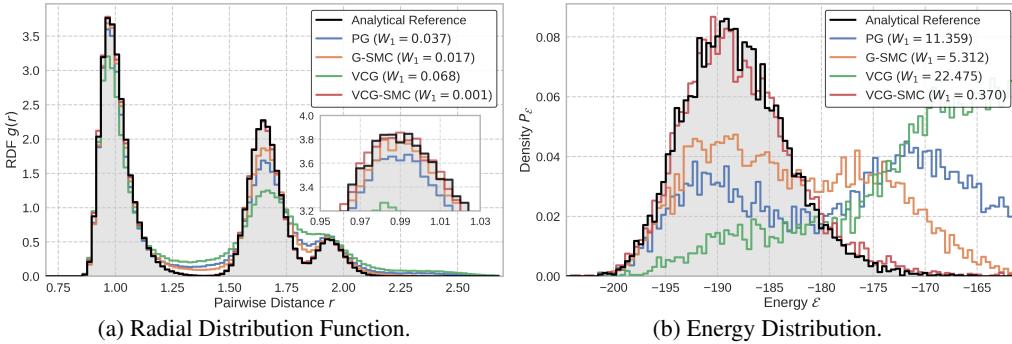
Motivated by Schebek et al. (2024), we consider applying an additional harmonic potential as a reward, and the reward-tilted distribution corresponds to another DW-4 system with a different configuration. The quantitative results in Tab. 1 confirm that our methods consistently outperform baselines by a large margin across all metrics. The ESS/potential variance plot in Fig. 11 confirms the stabilizing effect of our method on ESS. An ablation study in Fig. 12 demonstrates that our method converges as the number of particles increases across metrics.

Lennard-Jones-13 (LJ-13). We conclude with a highly challenging annealing task on the LJ-13 system (cf., App. B.1), a complex benchmark known for its rugged energy landscape and singular

Table 1: Performance comparison on Particle Systems (DW-4 and LJ-13). Results are mean \pm std over 5 runs. Best results per column are in bold.

| Method | DW-4, Annealing ($T = 2.0, \gamma = 2.0$) | | | | | DW-4, Reward-Tilting ($T = 2.0, \lambda' = 0.5$) | | | | |
|---------|---|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ |
| PG | 0.159 ± 1.232 | 0.400 ± 0.168 | 1.088 ± 0.384 | 0.208 ± 0.008 | 0.551 ± 0.009 | 0.867 ± 1.437 | 0.771 ± 0.085 | 1.714 ± 0.232 | 0.627 ± 0.003 | 1.837 ± 0.013 |
| G-SMC | 0.038 ± 0.338 | 0.365 ± 0.058 | 1.012 ± 0.253 | 0.208 ± 0.146 | 0.190 ± 0.080 | 0.329 ± 0.016 | 0.087 ± 0.039 | 0.194 ± 0.082 | 0.118 ± 0.004 | 0.330 ± 0.016 |
| VCG | -0.043 ± 0.022 | 0.014 ± 0.001 | 0.037 ± 0.008 | 0.043 ± 0.002 | 0.663 ± 0.015 | 0.699 ± 1.905 | 0.614 ± 0.139 | 1.692 ± 0.438 | 0.161 ± 0.033 | 0.461 ± 0.094 |
| VCG-SMC | -0.032 ± 0.009 | 0.014 ± 0.001 | 0.035 ± 0.002 | 0.060 ± 0.006 | 0.031 ± 0.007 | 0.296 ± 0.016 | 0.021 ± 0.001 | 0.048 ± 0.002 | 0.107 ± 0.005 | 0.296 ± 0.016 |

| Method | LJ-13, Annealing ($T = 2.0, \gamma = 2.5$) | | | | | LJ-13, Reward-Tilting ($T = 2.0, \lambda' = 0.8$) | | | | |
|---------|--|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|---|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ |
| PG | 13.58 ± 16.73 | 0.603 ± 0.095 | 0.598 ± 0.087 | 0.037 ± 0.000 | 11.40 ± 0.129 | 4.975 ± 5.159 | 0.719 ± 0.021 | 0.797 ± 0.058 | 0.070 ± 0.001 | 5.430 ± 0.074 |
| G-SMC | 13.64 ± 9.949 | 0.616 ± 0.012 | 0.523 ± 0.060 | 0.040 ± 0.030 | 13.48 ± 9.626 | 1.783 ± 0.202 | 0.081 ± 0.019 | 0.084 ± 0.032 | 0.023 ± 0.002 | 1.784 ± 0.202 |
| VCG | -1.084 ± 0.931 | 0.136 ± 0.032 | 0.144 ± 0.044 | 0.069 ± 0.001 | 22.86 ± 0.177 | 7.629 ± 12.37 | 0.695 ± 0.127 | 0.702 ± 0.174 | 0.036 ± 0.020 | 2.790 ± 1.531 |
| VCG-SMC | -0.699 ± 0.189 | 0.102 ± 0.050 | 0.098 ± 0.044 | 0.002 ± 0.000 | 0.286 ± 0.100 | 1.734 ± 0.106 | 0.015 ± 0.001 | 0.015 ± 0.001 | 0.022 ± 0.001 | 1.735 ± 0.106 |

Figure 5: Comparison of generated distributions for the LJ-13 annealing task ($\gamma = 2.5$). VCG-SMC is the only method that successfully recovers all three peaks in the (a) RDF and closely matches the (b) Energy Distribution. Insets provide a zoomed-in view.

behaviors at short distances. Fig. 5 presents the result of a demanding inference-time annealing task from $T = 1.0$ to 0.4 with $\gamma = 2.5$. The target distribution exhibits a third peak in its RDF corresponding to a structural feature almost absent at the initial temperature (Fig. 8). In a powerful demonstration of its capabilities, VCG-SMC is the only method that successfully discovers and samples from all modes, matching both the RDF and energy distribution with high precision. Metrics in Tab. 1 further confirm a significant performance gap over the baselines in this complex setting.

We refer readers to App. C.2 for additional experimental results and visualization on DW-4 and LJ-13 systems, with results with varying base temperatures T , annealing factor γ , constraint strength λ' , and number of particles N .

4.3 PROTEIN-LIGAND CO-FOLDING

Lastly, we apply DriftLite to the protein-ligand co-folding problem (Abramson et al., 2024; Bryant et al., 2024), a central task of structural biology and drug discovery. The goal is to generate 3D protein structures and their binding partners (ligands, particularly small molecules) simultaneously and in a mutually dependent manner, given the protein sequence and the ligand identity. This problem extends the classical protein folding problem (Jumper et al., 2021; Baek et al., 2021) and is crucial for elucidating protein-ligand interactions. Despite the recent progress achieved by diffusion models, notably AlphaFold3 (Abramson et al., 2024), Protenix (Team et al., 2025), and Boltz-2 (Passaro et al., 2025), a persistent challenge is that purely data-driven generative approaches tend to overemphasize global structural similarity while often producing conformations that violate basic physical constraints (Buttenschoen et al., 2024; Masters et al., 2024). Recent studies demonstrated that incorporating physics-based steering potentials can help mitigate this limitation (Passaro et al., 2025).

We adopt and follow the experimental setup of Boltz-2 (Passaro et al., 2025), an open-weight diffusion model, as the base model, and apply VCG-SMC to steer the generation of protein-ligand struc-

Table 2: Performance comparison on steering the physical validity of protein-ligand co-folding. Results are mean \pm std over 3 runs. Best results per column are in bold.

| Method | Valid Fraction \uparrow | Clash Free Fraction \uparrow | Bond Length \downarrow | Bond Angle \downarrow | Internal Clash \downarrow | Chiral Atom \downarrow | Chain Clashes \downarrow |
|---------|-------------------------------------|-------------------------------------|------------------------------------|-------------------------------------|------------------------------------|-----------------------------------|------------------------------------|
| Base | 0.374 ± 0.003 | 0.490 ± 0.007 | 55.00 ± 3.61 | 133.00 ± 7.00 | 138.67 ± 4.04 | 118.33 ± 12.74 | 398.67 ± 4.16 |
| FKS | 0.379 ± 0.014 | 0.490 ± 0.007 | 52.67 ± 2.89 | 127.33 ± 5.69 | 140.33 ± 2.08 | 126.33 ± 5.51 | 377.00 ± 20.30 |
| G-SMC | 0.838 ± 0.008 | 0.945 ± 0.005 | 42.33 ± 13.05 | 98.00 ± 23.07 | 31.33 ± 4.93 | 2.33 ± 0.58 | 31.67 ± 1.53 |
| VCG-SMC | 0.856 ± 0.008 | 0.950 ± 0.003 | 24.33 ± 9.29 | 61.00 ± 19.08 | 32.33 ± 4.16 | 1.00 ± 1.00 | 30.00 ± 1.00 |

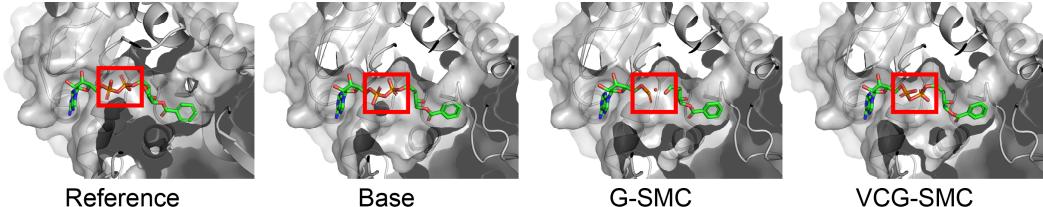


Figure 6: The reference and predicted complex structure of Hst2 bound to 2'-O-benzoyl ADP ribose. The reference corresponds to the experimentally determined crystal structure (PDB ID: 7F51). The unsteered base prediction inverted a chiral center in the ligand (highlighted with a red box). G-SMC failed to correct this issue and even broke the bonding, whereas VCG-SMC successfully guided the generation toward the correct chirality and preserved a chemically meaningful structure.

tures toward physically valid conformations using a physics-based potential as reward. We compare our method with two additional baselines: the unsteered model (Base) and Feynman-Kac Steering (FKS) (Singhal et al., 2025). We assess physical validity using the widely adopted PoseBuster V2 benchmark (Buttenschoen et al., 2024). Results are summarized in Tab. 2. In this task, evaluating the physics-based reward and its gradient adds relatively little overhead. VCG-SMC exhibits the strongest performance with fewer or without rule violations, improving the quality of partially valid structures, and increasing the proportion of fully valid ones. This underscores its effectiveness in a complex real-world setting. An example highlighting these improvements is shown in Fig. 6. Implementation details are provided in App. B. Additional experimental results on the ESS evolution and runtime comparisons are provided in App. C.3.

5 CONCLUSION

We introduce DriftLite, a lightweight, training-free framework that resolves a critical trade-off in the inference-time scaling of pre-trained diffusion models. By identifying and exploiting a fundamental degree of freedom in the Fokker-Planck equation, DriftLite actively controls the sampling drift, thereby mitigating the weight degeneracy that plagues previous particle-based methods. Our practical instantiations, VCG and ECG, impose modest and scalable overhead while dramatically improving the stability and accuracy of inference-time scaling. Experiments further confirm their effectiveness and strong scaling with the number of particles, and we observe that the VCG variant is generally more robust, while the ECG holds promise in several specific scenarios. Across particle and protein systems, our approach consistently produces higher-quality samples and handles complex distributions more robustly compared to existing inference-time scaling baselines.

While DriftLite proves effective, its reliance on a fixed set of linear basis functions presents a potential limitation. Future work could explore more expressive yet still efficient representations for the control drift, such as compact neural networks or adaptive basis sets, including those involving the posterior mean (Chung et al., 2022). Our current instantiations also assume twice-differentiability of the reward r and extending DriftLite to less smooth rewards is an open direction. We anticipate that future work will extend DriftLite to more complex physical systems, including LJ-55, ALDP, by pairing our inference-time scaling framework with continued advances in generative modeling. Furthermore, we have focused on annealing and reward-tilting tasks with non-heuristic targets and accuracy demands, but the DriftLite framework is broadly applicable beyond these tasks. Extending it to other generative problems, such as product-of-experts models or conditional generation, is a promising direction for future research. It is also interesting to study the inference-time scaling problem in the discrete settings under both the uniform (Campbell et al., 2022; Ren et al., 2024; 2025a) and the absorbing (Lou et al., 2023; Shi et al., 2024) frameworks.

ACKNOWLEDGMENTS

This work was conducted during Yinuo Ren’s internship and visit at Center for Computational Mathematics, Flatiron Institute, an internal research division of the Simons Foundation. We thank the Scientific Computing Core, Flatiron Institute, for the computing facilities and support. This material is based upon work supported by the National Science Foundation under Award No. CHE-2441297 to Grant M. Rotskoff. Lexing Ying acknowledges support by the National Science Foundation under Award No. DMS-2208163.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Tara Akhoud-Sadegh, Jarrid Rector-Brooks, Avishek Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, et al. Iterated denoising energy matching for sampling from boltzmann densities. *arXiv preprint arXiv:2402.06121*, 2024.

Tara Akhoud-Sadegh, Jungyoon Lee, Avishek Joey Bose, Valentin De Bortoli, Arnaud Doucet, Michael M Bronstein, Dominique Beaini, Siamak Ravanbakhsh, Kirill Neklyudov, and Alexander Tong. Progressive inference-time annealing of diffusion models for sampling from boltzmann densities. *arXiv preprint arXiv:2506.16471*, 2025.

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Michael S Albergo and Eric Vanden-Eijnden. Nets: A non-equilibrium transport sampler. *arXiv preprint arXiv:2410.02711*, 2024.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 72(3):269–342, 2010.

Christophe Andrieu, Anthony Lee, and Matti Vihola. Uniform ergodicity of the iterated conditional smc and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, pp. 842–872, 2018.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 843–852, 2023.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/jax-ml/jax>.

Mark Briers, Arnaud Doucet, and Simon Maskell. Smoothing algorithms for state-space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89, 2010.

Joan Bruna and Jiequn Han. Posterior sampling with denoising oracles via tilted transport. *arXiv preprint arXiv:2407.00745*, 2024.

Patrick Bryant, Atharva Kelkar, Andrea Guljas, Cecilia Clementi, and Frank Noé. Structure prediction of protein-ligand complexes from sequence information with umol. *Nature Communications*, 15(1):4536, 2024.

Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *Chemical Science*, 15(9):3130–3139, 2024.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Gabriel Cardoso, Sylvain Le Corff, Eric Moulines, et al. Monte carlo guided denoising diffusion models for bayesian linear inverse problems. In *The Twelfth International Conference on Learning Representations*, 2023.

Haoxuan Chen, Yinuo Ren, Martin Renqiang Min, Lexing Ying, and Zachary Izzo. Solving inverse problems via diffusion-based priors: An approximation-free ensemble sampling approach. *arXiv preprint arXiv:2506.03979*, 2025.

Michael Chertkov, Sungsoo Ahn, and Hamidreza Behjoo. Sampling decisions. *arXiv preprint arXiv:2503.14549*, 2025.

Muthu Chidambaram, Khashayar Gatmiry, Sitan Chen, Holden Lee, and Jianfeng Lu. What does guidance do? a fine-grained analysis in a simple setting. *arXiv preprint arXiv:2409.13074*, 2024.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.

Florentin Coeurdoux, Nicolas Dobigeon, and Pierre Chainais. Plug-and-play split Gibbs sampler: embedding deep generative priors in bayesian inference. *IEEE Transactions on Image Processing*, 2024.

DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem, 2020. URL <http://github.com/google-deepmind>.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

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

Carles Domingo-Enrich, Samy Jelassi, Arthur Mensch, Grant Rotskoff, and Joan Bruna. A mean-field analysis of two-player zero-sum games. *Advances in neural information processing systems*, 33:20215–20226, 2020.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024a.

Carles Domingo-Enrich, Jiequn Han, Brandon Amos, Joan Bruna, and Ricky TQ Chen. Stochastic optimal control matching. *Advances in Neural Information Processing Systems*, 37:112459–112504, 2024b.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pp. 8489–8510. PMLR, 2023.

Chenru Duan, Yuanqi Du, Haojun Jia, and Heather J Kulik. Accurate transition state generation with an object-aware equivariant elementary reaction diffusion model. *Nature computational science*, 3(12):1045–1055, 2023.

Chenru Duan, Aditya Nandy, Shyam Chand Pal, Xin Yang, Wenhao Gao, Yuanqi Du, Hendrik Kraß, Yeonghun Kang, Varinia Bernales, Zuyang Ye, et al. The rise of generative ai for metal-organic framework design and synthesis. *arXiv preprint arXiv:2508.13197*, 2025.

Lawrence C Evans. *Partial differential equations*, volume 19. American mathematical society, 2022.

Ying Fan and Kangwook Lee. Optimizing ddpm sampling with shortcut fine-tuning. *arXiv preprint arXiv:2301.13362*, 2023.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023.

Wenhao Gao, Shitong Luo, and Connor W Coley. Generative artificial intelligence for navigating synthesizable chemical space. *arXiv preprint arXiv:2410.03494*, 2024.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. *arXiv preprint arXiv:2404.14743*, 2024.

Florentin Guth, Zahra Kadkhodaie, and Eero P Simoncelli. Learning normalized image densities via dual score matching. *arXiv preprint arXiv:2506.05310*, 2025.

Jiajun He, José Miguel Hernández-Lobato, Yuanqi Du, and Francisco Vargas. Rne: a plug-and-play framework for diffusion density estimation and inference-time control. *arXiv preprint arXiv:2506.05668*, 2025.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. *arXiv preprint arXiv:2311.16424*, 2023.

Jeremy Heng, Adrian N Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential monte carlo. *The Annals of Statistics*, 48(5):2904–2929, 2020.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Jonathan H Huggins and Daniel M Roy. Sequential monte carlo as approximate sampling: bounds, adaptive resampling via ∞ -ESS, and an application to particle Gibbs. *Bernoulli*, 25(1):584–622, 2019.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.

Rafał Karczewski, Markus Heinonen, and Vikas Garg. Diffusion models as cartoonists: The curious case of high density regions. *arXiv preprint arXiv:2411.01293*, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36:59886–59910, 2023.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

Daniel Lacker. Mean field games and interacting particle systems. *preprint*, 2018.

Dieterich Lawson, Allan Raventós, Andrew Warrington, and Scott Linderman. Sixo: Smoothing inference with twisted objectives. *Advances in Neural Information Processing Systems*, 35:38844–38858, 2022.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. *Advances in neural information processing systems*, 36:14005–14034, 2023.

Cheuk Kit Lee, Paul Jeha, Jes Frellsen, Pietro Lio, Michael Samuel Albergo, and Francisco Varillas. Debiasing guidance for discrete diffusion with sequential monte carlo. *arXiv preprint arXiv:2502.06079*, 2025.

Benedict Leimkuhler and Charles Matthews. Rational construction of stochastic numerical methods for molecular sampling. *Applied Mathematics Research eXpress*, 2013(1):34–56, 2013.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin Kurt Miller, and Ricky TQ Chen. Adjoint Schrödinger bridge sampler. *arXiv preprint arXiv:2506.22565*, 2025.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Jianfeng Lu and Yuliang Wang. Guidance for twisted particle filter: a continuous-time perspective. *arXiv preprint arXiv:2409.02399*, 2024.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

Matthew R Masters, Amr H Mahmoud, and Markus A Lill. Do deep learning models for co-folding learn the physics of protein-ligand interactions? *bioRxiv*, pp. 2024–06, 2024.

Laurence Midgley, Vincent Stimper, Javier Antorán, Emile Mathieu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Se (3) equivariant augmented coupling flows. *Advances in Neural Information Processing Systems*, 36:79200–79225, 2023.

Pierre Del Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer, 2004.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Saro Passaro, Gabriele Corso, Jeremy Wohlwend, Mateo Reveiz, Stephan Thaler, Vignesh Ram Somnath, Noah Getz, Tally Portnoi, Julien Roy, Hannes Stark, et al. Boltz-2: Towards accurate and efficient binding affinity prediction. *BioRxiv*, pp. 2025–06, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Yinuo Ren, Haoxuan Chen, Grant M Rotskoff, and Lexing Ying. How discrete and continuous diffusion meet: Comprehensive analysis of discrete diffusion models via a stochastic integral framework. *arXiv preprint arXiv:2410.03601*, 2024.

Yinuo Ren, Haoxuan Chen, Yuchen Zhu, Wei Guo, Yongxin Chen, Grant M Rotskoff, Molei Tao, and Lexing Ying. Fast solvers for discrete diffusion models: Theory and applications of high-order algorithms. *arXiv preprint arXiv:2502.00234*, 2025a.

Yinuo Ren, Grant M Rotskoff, and Lexing Ying. A unified approach to analysis and design of denoising markov models. *arXiv preprint arXiv:2504.01938*, 2025b.

Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. *arXiv preprint arXiv:2307.01198*, 2023.

Kevin Rojas, Ye He, Chieh-Hsin Lai, Yuta Takida, Yuki Mitsufuji, and Molei Tao. Theory-informed improvements to classifier-free guidance for discrete diffusion models. *arXiv preprint arXiv:2507.08965*, 2025.

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

Victor Garcia Satorras, Emiel Hoogeboom, Fabian Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows. *Advances in Neural Information Processing Systems*, 34:4181–4192, 2021a.

Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021b.

Maximilian Schebek, Michele Invernizzi, Frank Noé, and Jutta Rogal. Efficient mapping of phase diagrams with conditional boltzmann generators. *Machine Learning: Science and Technology*, 5 (4):045045, 2024.

Marcin Sendera, Minsu Kim, Sarthak Mittal, Pablo Lemos, Luca Scimeca, Jarrid Rector-Brooks, Alexandre Adam, Yoshua Bengio, and Nikolay Malkin. Improved off-policy training of diffusion samplers. *Advances in Neural Information Processing Systems*, 37:81016–81045, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Marta Skreta, Tara Akhoud-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. *arXiv preprint arXiv:2503.02819*, 2025.

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

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023a.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pp. 32483–32498. PMLR, 2023b.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Alain-Sol Sznitman. Topics in propagation of chaos. *Ecole d’été de probabilités de Saint-Flour XIX—1989*, 1464:165–251, 1991.

ByteDance AML AI4Science Team, Xinshi Chen, Yuxuan Zhang, Chan Lu, Wenzhi Ma, Jiaqi Guan, Chengyue Gong, Jincai Yang, Hanyu Zhang, Ke Zhang, et al. Protenix-advancing structure prediction through a comprehensive alphafold3 reproduction. *BioRxiv*, pp. 2025–01, 2025.

James Thornton, Louis Béthune, Ruixiang Zhang, Arwen Bradley, Preetum Nakkiran, and Shuangfei Zhai. Composition and control with distilled energy diffusion models and sequential monte carlo. *arXiv preprint arXiv:2502.12786*, 2025.

Brian L Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.

Francisco Vargas, Will Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. *arXiv preprint arXiv:2302.13834*, 2023a.

Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. Transport meets variational inference: Controlled monte carlo diffusions. *arXiv preprint arXiv:2307.01050*, 2023b.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *Advances in neural information processing systems*, 37:76080–76114, 2024.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisebach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

Nick Whiteley and Anthony Lee. Twisted particle filters. *The Annals of Statistics*, 42(1):115–141, 2014.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

Yuchen Wu, Minshuo Chen, Zihao Li, Mengdi Wang, and Yuting Wei. Theoretical insights for diffusion guidance: A case study for gaussian mixture models. *arXiv preprint arXiv:2403.01639*, 2024a.

Zihui Wu, Yu Sun, Yifan Chen, Bingliang Zhang, Yisong Yue, and Katherine Bouman. Principled probabilistic imaging using diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 37:118389–118427, 2024b.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Xingyu Xu and Yuejie Chi. Provably robust score-based diffusion posterior sampling for plug-and-play image reconstruction. *arXiv preprint arXiv:2403.17042*, 2024.

Bing Yu and Weinan E. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, 639(8055):624–632, 2025.

Fengzhe Zhang, Laurence I Midgley, and José Miguel Hernández-Lobato. Efficient and unbiased sampling from boltzmann distributions via variance-tuned diffusion models. *arXiv preprint arXiv:2505.21005*, 2025.

Linfeng Zhang, Weinan E, and Lei Wang. Monge-Ampère flow for generative modeling. *arXiv preprint arXiv:1809.10188*, 2018.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling. *arXiv preprint arXiv:2111.15141*, 2021.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential monte carlo. *arXiv preprint arXiv:2404.17546*, 2024.

Hongkai Zheng, Wenda Chu, Austin Wang, Nikola Kovachki, Ricardo Baptista, and Yisong Yue. Ensemble kalman diffusion guidance: A derivative-free method for inverse problems. *arXiv preprint arXiv:2409.20175*, 2024.

Hongkai Zheng, Wenda Chu, Bingliang Zhang, Zihui Wu, Austin Wang, Berthy T Feng, Caifeng Zou, Yu Sun, Nikola Kovachki, Zachary E Ross, et al. Inversebench: Benchmarking plug-and-play diffusion priors for inverse problems in physical sciences. *arXiv preprint arXiv:2503.11043*, 2025.

Yuchen Zhu, Tianrong Chen, Evangelos A Theodorou, Xie Chen, and Molei Tao. Quantum state generation with structure-preserving diffusion model. *arXiv preprint arXiv:2404.06336*, 2024.

Yuchen Zhu, Wei Guo, Jaemoo Choi, Guan-Horng Liu, Yongxin Chen, and Molei Tao. Mdns: Masked diffusion neural sampler via stochastic optimal control. *arXiv preprint arXiv:2508.10684*, 2025.

A PROOFS

In this section, we present the omitted proofs of several propositions and additional discussions of the theoretical claims in the main content.

A.1 PROOF OF PROPOSITION 2.1

To aid reading, we reiterate Prop. 2.1 below.

Proposition A.1 (Guidance-Based Dynamics). *The exact time evolution of the density $(q_t)_{t \in [0, T]}$ follows the following Fokker-Planck equation:*

$$\partial_t q_t(\mathbf{x}) = -\nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) + q_t(\mathbf{x}) g_t(\mathbf{x}),$$

where $\tilde{\mathbf{v}}_t$ is given by

$$\begin{aligned} \tilde{\mathbf{v}}_t(\mathbf{x}) &= -\bar{\mathbf{u}}_t(\mathbf{x}) + \frac{\bar{U}_t^2 + V_t^2}{2} (\gamma \nabla \log \bar{p}_t(\mathbf{x}) + \nabla r_t(\mathbf{x})) \\ &= -(1 - \gamma) \bar{\mathbf{u}}_t(\mathbf{x}) + \gamma \mathbf{v}_t(\mathbf{x}) + \frac{\bar{U}_t^2 + V_t^2}{2} \nabla r_t(\mathbf{x}), \end{aligned}$$

and the reweighting potential $g_t(\mathbf{x}) = G_t(\mathbf{x}) - \mathbb{E}_{q_t}[G_t(\cdot)]$ is given by:

$$\begin{aligned} G_t &= \dot{r}_t - (1 - \gamma) \nabla \cdot \bar{\mathbf{u}}_t + \frac{\bar{U}_t^2}{2} (\Delta r_t - \gamma(1 - \gamma) \|\nabla \log \bar{p}_t\|^2) \\ &\quad + \nabla r_t^\top \left(-\bar{\mathbf{u}}_t + \gamma \bar{U}_t^2 \nabla \log \bar{p}_t + \frac{\bar{U}_t^2}{2} \nabla r_t \right). \end{aligned}$$

Proof. We begin with the definition of the target density $q_t(\mathbf{x})$:

$$\log q_t(\mathbf{x}) = \gamma \log \bar{p}_t(\mathbf{x}) + r_t(\mathbf{x}) - Z_t,$$

where we define the log-partition function

$$Z_t = \log \int \bar{p}_t(\mathbf{y})^\gamma \exp(r_t(\mathbf{y})) d\mathbf{y},$$

and taking the time derivative gives:

$$\partial_t \log q_t(\mathbf{x}) = \gamma \partial_t \log \bar{p}_t(\mathbf{x}) + \dot{r}_t(\mathbf{x}) - \partial_t Z_t. \quad (\text{A.1})$$

Recall that the Fokker-Planck equation for the backward process marginals \bar{p}_t is as follows:

$$\partial_t \bar{p}_t(\mathbf{x}) = -\nabla \cdot (\mathbf{v}_t(\mathbf{x}) \bar{p}_t(\mathbf{x})) + \frac{V_t^2}{2} \Delta \bar{p}_t(\mathbf{x}),$$

which can be expressed in terms of the log-density as:

$$\begin{aligned} \partial_t \log \bar{p}_t(\mathbf{x}) &= -\bar{p}_t^{-1}(\mathbf{x}) \nabla \cdot (\mathbf{v}_t(\mathbf{x}) \bar{p}_t(\mathbf{x})) + \frac{V_t^2}{2} \bar{p}_t^{-1}(\mathbf{x}) \Delta \bar{p}_t(\mathbf{x}) \\ &= -\nabla \cdot \mathbf{v}_t(\mathbf{x}) - \mathbf{v}_t(\mathbf{x}) \cdot \nabla \log \bar{p}_t(\mathbf{x}) + \frac{V_t^2}{2} (\Delta \log \bar{p}_t(\mathbf{x}) + \|\nabla \log \bar{p}_t(\mathbf{x})\|^2). \end{aligned} \quad (\text{A.2})$$

We posit that the time evolution of q_t is governed by the Fokker-Planck equation with an additional reweighting term:

$$\partial_t q_t(\mathbf{x}) = -\nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) + q_t(\mathbf{x}) g_t(\mathbf{x}),$$

and dividing both sides by $q_t(\mathbf{x})$, we can solve for the potential $g_t(\mathbf{x})$:

$$g_t(\mathbf{x}) = \partial_t \log q_t(\mathbf{x}) + q_t^{-1}(\mathbf{x}) \nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})] - \frac{V_t^2}{2} q_t^{-1}(\mathbf{x}) \Delta q_t(\mathbf{x}).$$

Since we have

$$\int \partial_t q_t(\mathbf{x}) d\mathbf{x} = \frac{d}{dt} \int q_t(\mathbf{x}) d\mathbf{x} = 0,$$

and

$$\int \left(-\nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) \right) d\mathbf{x} = 0$$

by applying the divergence theorem and assuming suitable boundary conditions, the reweighting term must have zero expectation, *i.e.*,

$$\mathbb{E}_{\mathbf{x} \sim q_t} [g_t(\mathbf{x})] = \int q_t(\mathbf{x}) g_t(\mathbf{x}) = 0.$$

Thus, we can write $g_t(\mathbf{x}) = G_t(\mathbf{x}) - \mathbb{E}_{q_t} [G_t(\cdot)]$, where we define $G_t(\mathbf{x})$ by absorbing the spatially independent term $\partial_t Z_t$:

$$\begin{aligned} G_t(\mathbf{x}) &= g_t(\mathbf{x}) + \partial_t Z_t \\ &= \partial_t \log q_t(\mathbf{x}) + \partial_t Z_t + q_t^{-1}(\mathbf{x}) \nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})] - \frac{V_t^2}{2} q_t^{-1}(\mathbf{x}) \Delta q_t(\mathbf{x}) \\ &= \dot{r}_t(\mathbf{x}) + \gamma \partial_t \log \tilde{p}_t(\mathbf{x}) + q_t^{-1}(\mathbf{x}) \nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})] - \frac{V_t^2}{2} q_t^{-1}(\mathbf{x}) \Delta q_t(\mathbf{x}) \\ &= \dot{r}_t(\mathbf{x}) + \gamma \left(-\nabla \cdot \mathbf{v}_t(\mathbf{x}) - \mathbf{v}_t(\mathbf{x})^\top \nabla \log \tilde{p}_t(\mathbf{x}) + \frac{V_t^2}{2} (\Delta \log \tilde{p}_t(\mathbf{x}) + \|\nabla \log \tilde{p}_t(\mathbf{x})\|^2) \right) \\ &\quad + \nabla \cdot \tilde{\mathbf{v}}_t(\mathbf{x}) + \tilde{\mathbf{v}}_t(\mathbf{x})^\top \nabla \log q_t(\mathbf{x}) - \frac{V_t^2}{2} (\Delta \log q_t(\mathbf{x}) + \|\nabla \log q_t(\mathbf{x})\|^2) \\ &= \dot{r}_t(\mathbf{x}) - \underbrace{\gamma \nabla \cdot \mathbf{v}_t(\mathbf{x}) + \nabla \cdot \tilde{\mathbf{v}}_t(\mathbf{x})}_{(I)} - \underbrace{\gamma \mathbf{v}_t(\mathbf{x})^\top \nabla \log \tilde{p}_t(\mathbf{x}) + \tilde{\mathbf{v}}_t(\mathbf{x})^\top \nabla \log q_t(\mathbf{x})}_{(II)} \\ &\quad + \underbrace{\frac{\gamma V_t^2}{2} (\Delta \log \tilde{p}_t(\mathbf{x}) + \|\nabla \log \tilde{p}_t(\mathbf{x})\|^2) - \frac{V_t^2}{2} (\Delta \log q_t(\mathbf{x}) + \|\nabla \log q_t(\mathbf{x})\|^2)}_{(III)}, \end{aligned}$$

where the second equality is due to Eqn. (A.1), and the second-to -last is due to Eqn. (A.1).

We now substitute the expressions for $\partial_t \log \tilde{p}_t$ and $\log q_t$ and simplify term by term.

• (I) Divergence Terms:

$$\begin{aligned} (I) &= -\gamma \nabla \cdot \mathbf{v}_t(\mathbf{x}) + \nabla \cdot \tilde{\mathbf{v}}_t(\mathbf{x}) = \nabla \cdot (\tilde{\mathbf{v}}_t(\mathbf{x}) - \gamma \mathbf{v}_t(\mathbf{x})) \\ &= \nabla \cdot \left(-(1 - \gamma) \tilde{\mathbf{u}}_t(\mathbf{x}) + \frac{\tilde{U}_t^2 + V_t^2}{2} \nabla r_t(\mathbf{x}) \right) = -(1 - \gamma) \nabla \cdot \tilde{\mathbf{u}}_t(\mathbf{x}) + \frac{\tilde{U}_t^2 + V_t^2}{2} \Delta r_t(\mathbf{x}). \end{aligned}$$

• (II) Inner Product Terms:

$$\begin{aligned} (II) &= -\gamma \mathbf{v}_t(\mathbf{x})^\top \nabla \log \tilde{p}_t(\mathbf{x}) + \tilde{\mathbf{v}}_t(\mathbf{x})^\top \nabla \log q_t(\mathbf{x}) \\ &= -\gamma \mathbf{v}_t(\mathbf{x})^\top \nabla \log \tilde{p}_t(\mathbf{x}) + \tilde{\mathbf{v}}_t(\mathbf{x})^\top (\gamma \nabla \log \tilde{p}_t(\mathbf{x}) + \nabla r_t(\mathbf{x})) \\ &= (\tilde{\mathbf{v}}_t(\mathbf{x}) - \mathbf{v}_t(\mathbf{x}))^\top (\gamma \nabla \log \tilde{p}_t(\mathbf{x}) + \tilde{\mathbf{v}}_t(\mathbf{x})^\top \nabla r_t(\mathbf{x})) \\ &= \left(-(1 - \gamma) \tilde{\mathbf{u}}_t(\mathbf{x}) + \frac{\tilde{U}_t^2 + V_t^2}{2} \nabla r_t(\mathbf{x}) \right)^\top \gamma \nabla \log \tilde{p}_t(\mathbf{x}) \\ &\quad + \left(-(1 - \gamma) \tilde{\mathbf{u}}_t(\mathbf{x}) + \gamma \mathbf{v}_t(\mathbf{x}) + \frac{\tilde{U}_t^2 + V_t^2}{2} \nabla r_t(\mathbf{x}) \right)^\top \nabla r_t(\mathbf{x}) \\ &= \nabla r_t^\top \left(-\tilde{\mathbf{u}}_t(\mathbf{x}) + \gamma \tilde{U}_t^2 \nabla \log \tilde{p}_t(\mathbf{x}) + \frac{\tilde{U}_t^2 + V_t^2}{2} \nabla r_t(\mathbf{x}) \right) - \frac{\tilde{U}_t^2 + V_t^2}{2} \gamma (1 - \gamma) \|\nabla \log \tilde{p}_t(\mathbf{x})\|^2. \end{aligned}$$

- (III) Laplacian Terms:

$$\begin{aligned}
(\text{III}) &= \frac{\gamma V_t^2}{2} (\Delta \log \tilde{p}_t(\mathbf{x}) + \|\nabla \log \tilde{p}_t(\mathbf{x})\|^2) - \frac{V_t^2}{2} (\Delta \log q_t(\mathbf{x}) + \|\nabla \log q_t(\mathbf{x})\|^2) \\
&= \frac{V_t^2}{2} (\gamma \Delta \log \tilde{p}_t(\mathbf{x}) - \Delta(\gamma \log \tilde{p}_t(\mathbf{x}) + r_t(\mathbf{x})) + \gamma \|\nabla \log \tilde{p}_t(\mathbf{x})\|^2 - \|\gamma \nabla \log \tilde{p}_t(\mathbf{x}) + \nabla r_t(\mathbf{x})\|^2) \\
&= -\frac{V_t^2}{2} \Delta r_t(\mathbf{x}) - \frac{V_t^2}{2} (\gamma(\gamma-1) \|\nabla \log \tilde{p}_t(\mathbf{x})\|^2 + 2\gamma \nabla \log \tilde{p}_t(\mathbf{x}) \cdot \nabla r_t(\mathbf{x}) + \|\nabla r_t(\mathbf{x})\|^2).
\end{aligned}$$

Combining all the simplified terms, we arrive at the expression for G_t stated in the proposition. \square

Remark A.2. This proof is similar to the proof in related works (Skreta et al., 2025; Chen et al., 2025), where Fokker-Planck equations are derived for each task-specific setting. While being more general, our approach also omits the computation of the time derivative of the log-partition function $\partial_t Z_t$, since we notice that it cancels out in the final expression for G_t . This simplification makes the proof more straightforward and concise.

A.2 JUSTIFICATION OF WEIGHTED PARTICLE SYSTEM (2.6)

Proposition A.3 (Weighted Particle Simulation). Let $q_t : \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$ solve

$$\partial_t q_t(\mathbf{x}) = -\nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) + q_t(\mathbf{x}) g_t(\mathbf{x}).$$

Then this PDE can be simulated in the mean-field limit $N \rightarrow \infty$ by the weighted particle system

$$\begin{cases} d\mathbf{x}_t^{(i)} = \tilde{\mathbf{v}}_t(\mathbf{x}_t^{(i)}) dt + V_t d\mathbf{w}_t^{(i)}, & i \in [N], \\ d \log w_t^{(i)} = \hat{g}_t(\mathbf{x}_t^{(i)}) dt, & i \in [N], \end{cases}$$

where the weights satisfy the normalization $\sum_{i=1}^N w_t^{(i)} = 1$, and the empirical centered version of g_t :

$$\hat{g}_t(\mathbf{x}) = G_t(\mathbf{x}) - \sum_{j=1}^N w_t^{(j)} G_t(\mathbf{x}_t^{(j)})$$

ensures mass preservation.

The proof of this argument is standard (Moral, 2004; Doucet et al., 2000; Andrieu et al., 2010) under typical assumptions, including local Lipschitz continuity and linear growth of the drift $\tilde{\mathbf{v}}_t$, boundedness of the diffusion coefficient V_t , moderate growth of \hat{g}_t , sufficient regularity of q_t to justify integration by parts, and either fast decay at infinity or no-flux boundary conditions to eliminate boundary terms. We provide a proof sketch below for the reader's convenience. We also point out that similar arguments also apply to the weighted particle simulation for the controlled dynamics in Prop. 3.1, which we omit for simplicity.

Proof Sketch. The main steps are as follows:

- Step 1 (Empirical measure). Define the weighted empirical distribution

$$\mu_t^N = \sum_{i=1}^N w_t^{(i)} \delta_{\mathbf{x}_t^{(i)}}.$$

- Step 2 (Test function evolution). For $\varphi \in C_c^\infty(\mathbb{R}^d)$, consider

$$\langle \varphi, \mu_t^N \rangle = \sum_{i=1}^N w_t^{(i)} \varphi(\mathbf{x}_t^{(i)}).$$

Apply Itô's lemma to $\varphi(\mathbf{x}_t^{(i)})$ under the SDE and combine with the weight dynamics via the product rule. This yields

$$d\langle \varphi, \mu_t^N \rangle = \left\langle \tilde{\mathbf{v}}_t^\top \nabla \varphi + \frac{V_t^2}{2} \Delta \varphi + \hat{g}_t \varphi, \mu_t^N \right\rangle dt + dM_t^N,$$

where M_t^N is a martingale term.

- *Step 3 (Limit $N \rightarrow \infty$).* By law of large numbers and propagation of chaos (Sznitman, 1991; Lacker, 2018), we have the weak convergence:

$$\mu_t^N \Rightarrow q_t(\mathbf{x})d\mathbf{x}, \quad \text{as } N \rightarrow \infty,$$

while the martingale term vanishes. Passing to the limit gives the weak form of the PDE:

$$\frac{d}{dt} \int \varphi(\mathbf{x}) q_t(\mathbf{x}) d\mathbf{x} = \int \left[\tilde{\mathbf{v}}_t(\mathbf{x})^\top \nabla \varphi(\mathbf{x}) + \frac{V_t^2}{2} \Delta \varphi(\mathbf{x}) + \hat{g}_t(\mathbf{x}) \varphi(\mathbf{x}) \right] q_t(\mathbf{x}) d\mathbf{x}.$$

- *Step 4 (Integration by parts).* Using divergence theorem identities, we have

$$\frac{d}{dt} \int \varphi(\mathbf{x}) q_t(\mathbf{x}) d\mathbf{x} = \int \varphi(\mathbf{x}) \left[-\nabla \cdot (\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})) + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) + \hat{g}_t(\mathbf{x}) q_t(\mathbf{x}) \right] d\mathbf{x},$$

for all test functions φ .

Hence, we have

$$\partial_t q_t(\mathbf{x}) = -\nabla \cdot (\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})) + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) + \hat{g}_t(\mathbf{x}) q_t(\mathbf{x}),$$

and the proof is complete. \square

A.3 PROOF OF PROPOSITION 3.1

The following proposition is the same as Prop. 3.1, but with a more detailed proof.

Proposition A.4 (Degree of Freedom). *For any control drift $\mathbf{b}_t(\mathbf{x})$, the Fokker-Planck equation*

$$\partial_t q_t(\mathbf{x}) = -\nabla \cdot [\tilde{\mathbf{v}}_t(\mathbf{x}) q_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) + q_t(\mathbf{x}) g_t(\mathbf{x}),$$

is equivalent to the following one with an additional control drift term $\mathbf{b}_t(\mathbf{x})$:

$$\partial_t q_t(\mathbf{x}) = -\nabla \cdot [(\tilde{\mathbf{v}}_t(\mathbf{x}) + \mathbf{b}_t(\mathbf{x})) q_t(\mathbf{x})] + \frac{V_t^2}{2} \Delta q_t(\mathbf{x}) + q_t(\mathbf{x}) \phi_t(\mathbf{x}),$$

where the residual potential is $\phi_t(\mathbf{x}) = g_t(\mathbf{x}) + h_t(\mathbf{x}; \mathbf{b}_t)$ with the control potential $h_t(\cdot; \mathbf{b}_t)$ defined as:

$$h_t(\mathbf{x}; \mathbf{b}_t) = (\gamma \nabla \log \tilde{p}_t(\mathbf{x}) + \nabla r_t(\mathbf{x})) \cdot \mathbf{b}_t(\mathbf{x}) + \nabla \cdot \mathbf{b}_t(\mathbf{x}).$$

Proof. The terms added to the right-hand side of Eqn. (2.5) to obtain Eqn. (3.1) are:

$$-\nabla \cdot (\mathbf{b}_t(\mathbf{x}) q_t(\mathbf{x})) + q_t(\mathbf{x}) (h_t(\mathbf{x}; \mathbf{b}_t) - \mathbb{E}_{q_t} [h_t(\mathbf{x}; \mathbf{b}_t)]).$$

First, we prove that $\mathbb{E}_{q_t} [h_t(\mathbf{x}; \mathbf{b}_t)] = 0$:

$$\begin{aligned} \mathbb{E}_{q_t} [h_t(\mathbf{x}; \mathbf{b}_t)] &= \int q_t(\mathbf{x}) [(\nabla r_t(\mathbf{x}) + \gamma \nabla \log \tilde{p}_t(\mathbf{x})) \cdot \mathbf{b}_t(\mathbf{x}) + \nabla \cdot \mathbf{b}_t(\mathbf{x})] d\mathbf{x} \\ &= \int \nabla \cdot (q_t(\mathbf{x}) \mathbf{b}_t(\mathbf{x})) d\mathbf{x} = 0, \end{aligned}$$

where the last equality follows from the divergence theorem, assuming appropriate boundary conditions (e.g., $q_t \mathbf{b}_t$ vanishes at infinity).

Then, we show that the remaining added terms cancel each other out:

$$\begin{aligned} & -\nabla \cdot (\mathbf{b}_t(\mathbf{x})q_t(\mathbf{x})) + q_t(\mathbf{x})h_t(\mathbf{x}; \mathbf{b}_t) \\ & = -\nabla \cdot (\mathbf{b}_t(\mathbf{x})q_t(\mathbf{x})) + q_t(\mathbf{x})[\nabla \log q_t(\mathbf{x}) \cdot \mathbf{b}_t(\mathbf{x}) + \nabla \cdot \mathbf{b}_t(\mathbf{x})] \\ & = -\nabla \cdot (\mathbf{b}_t(\mathbf{x})q_t(\mathbf{x})) + \nabla \cdot (q_t(\mathbf{x})\mathbf{b}_t(\mathbf{x})) = 0. \end{aligned}$$

Since the added terms sum to zero (given $\mathbb{E}_{q_t}[h_t(\mathbf{x}; \mathbf{b}_t)] = 0$), the two PDEs are equivalent. \square

A.4 FORMAL SOLUTION FOR THE OPTIMAL CONTROL DRIFT

In Prop. 3.2, we claim that there exists a uniform optimal control drift as we rewrite its mathematical rigorous version in the following proposition.

Proposition A.5 (Optimal Control). *Let $\Omega \subset \mathbb{R}^d$ be a bounded Lipschitz domain. Assume that $q_t : \Omega \rightarrow \mathbb{R}$ is measurable and uniformly elliptic: there exist constants $0 < \lambda \leq \Lambda < \infty$ such that*

$$\lambda \leq q_t(x) \leq \Lambda, \quad \text{for a.e. } x \in \Omega,$$

and suppose $g_t \in L^2(\Omega)$. Then there exists a unique potential $A_t^ \in H_0^1(\Omega)$ solving*

$$-\nabla \cdot (q_t(x)\nabla A_t^*(x)) = q_t(x)g_t(x) \quad \text{in } H^{-1}(\Omega).$$

Defining the control $\mathbf{b}_t^(x) := \nabla A_t^*(x) \in L^2(\Omega; \mathbb{R}^d)$, one has*

$$\phi_t^*(x) = g_t(x) + h_t(x; \mathbf{b}_t^*) = g_t(x) + \frac{1}{q_t(x)}\nabla \cdot (q_t(x)\mathbf{b}_t^*(x)) = 0 \quad \text{in } H^{-1}(\Omega).$$

In particular, \mathbf{b}_t^ is the unique curl-free control that drives ϕ_t to zero.*

Proof. This follows directly from the Lax–Milgram theorem. The bilinear form

$$a(u, v) := \int_{\Omega} q_t \nabla u \cdot \nabla v dx, \quad u, v \in H_0^1(\Omega),$$

is bounded and coercive, while the linear functional

$$L(v) := \int_{\Omega} q_t g_t v dx$$

is continuous on $H_0^1(\Omega)$. By the Lax–Milgram theorem, there exists a unique $A_t^* \in H_0^1(\Omega)$ satisfying (3.2). The existence and uniqueness of weak solutions to such elliptic PDEs are standard results; see, for example, Evans (2022, Chapter 6). \square

While the existence and the uniqueness of the solution to the Poisson equation (3.2) are shown above, we present a formal solution for the control drift \mathbf{b}_t^* in the following, leading to our choice of basis functions in Sec. 3.3.

Let $\mathbf{f}_t(\mathbf{x}) = q_t(\mathbf{x})\mathbf{b}_t^*(\mathbf{x})$. By the Helmholtz–Hodge theorem, any sufficiently smooth vector field \mathbf{f}_t can be decomposed into a curl-free component (the gradient of a scalar potential A) and a divergence-free component \mathbf{u}_{\perp} , so that

$$\mathbf{f}_t(\mathbf{x}) = \nabla A(\mathbf{x}) + \mathbf{u}_{\perp}(\mathbf{x}),$$

where $\nabla \cdot \mathbf{u}_{\perp} \equiv 0$.

Substituting this decomposition into the equation gives:

$$\nabla \cdot (\nabla A(\mathbf{x}) + \mathbf{u}_{\perp}(\mathbf{x})) = \Delta A(\mathbf{x}) = -q_t(\mathbf{x})g_t(\mathbf{x}).$$

This is a standard Poisson equation for the scalar potential A . The solution for A can be expressed formally using the Green’s function for the Laplacian in d dimensions, $G_d(\mathbf{x} - \mathbf{y})$:

$$A(\mathbf{x}) = - \int G_d(\mathbf{x} - \mathbf{y})q_t(\mathbf{y})g_t(\mathbf{y}) d\mathbf{y}.$$

The desired control is then given by $\mathbf{b}_t(\mathbf{x}) = q_t(\mathbf{x})^{-1} \nabla A(\mathbf{x})$. Taking the gradient of $A(\mathbf{x})$ with respect to \mathbf{x} and using integration by parts with the property $\nabla_{\mathbf{x}} G_d(\mathbf{x} - \mathbf{y}) = -\nabla_{\mathbf{y}} G_d(\mathbf{x} - \mathbf{y})$, we get the following formal solution:

$$\begin{aligned}\nabla A(\mathbf{x}) &= - \int \nabla_{\mathbf{x}} G_d(\mathbf{x} - \mathbf{y}) q_t(\mathbf{y}) g_t(\mathbf{y}) d\mathbf{y} \\ &= \int G_d(\mathbf{x} - \mathbf{y}) \nabla_{\mathbf{y}} (q_t(\mathbf{y}) g_t(\mathbf{y})) d\mathbf{y} \\ &= \int G_d(\mathbf{x} - \mathbf{y}) (g_t(\mathbf{y}) \nabla_{\mathbf{y}} \log q_t(\mathbf{y}) + \nabla_{\mathbf{y}} g_t(\mathbf{y})) q_t(\mathbf{y}) d\mathbf{y},\end{aligned}$$

where the term ∇g_t introduces higher-order derivatives of the reward function:

$$\nabla g_t(\mathbf{x}) = \beta_t \left[\nabla^2 r(\mathbf{x}) \left(-\bar{\mathbf{u}}_t(\mathbf{x}) - \frac{1}{2} \beta_t \nabla r(\mathbf{x}) \right) - \nabla \bar{\mathbf{u}}_t(\mathbf{x}) \nabla r(\mathbf{x}) \right] - \frac{1}{2} \beta_t \nabla \Delta r(\mathbf{x}) + \dot{\beta}_t \nabla r(\mathbf{x}).$$

This formal solution is computationally infeasible as it requires integrating over all space \mathbf{y} , weighted by the unknown density $q_t(\mathbf{y})$ that we are trying to simulate. However, its structure motivates our choice of local basis functions: the reward gradient $\nabla r_t(\mathbf{x})$, the score $\nabla \log \hat{p}_t(\mathbf{x})$, and the forward drift $\bar{\mathbf{u}}_t(\mathbf{x})$, as we discard higher-order derivatives.

B ADDITIONAL IMPLEMENTATION DETAILS

In this section, we provide additional implementation details for the experiments, including the problem setup with parameters, a detailed description of the network architecture, training, inference, and evaluation procedures.

B.1 PROBLEM SETTINGS

Gaussian Mixture Model (GMM). In the Gaussian Mixture Model (GMM) task, the data distribution is set as

$$p_0(\mathbf{x}) = \frac{1}{40} \sum_{i=1}^{40} \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, 50\mathbf{I}),$$

where each component mean $\boldsymbol{\mu}_i$ is sampled from $\text{Unif}([-40, 40])$.

For the annealing task where the target distribution is $q_T(\mathbf{x}) \propto p_0(\mathbf{x})^\gamma$, the reference samples are obtained by rejection sampling with the following proposal distribution:

$$q_0^{\text{proposal}}(\mathbf{x}) = \frac{1}{40} \sum_{i=1}^{40} \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}_i, \frac{50}{\gamma} \mathbf{I}\right).$$

For the reward-tilting task, the quadratic reward function is given by

$$r(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}),$$

where the target mean $\boldsymbol{\mu}$ is sampled from $\mathcal{N}(\boldsymbol{\mu}; \mathbf{0}, 100\mathbf{I})$ and the covariance matrix $\Sigma = \sigma \mathbf{I}$ with σ being a parameter controlling the spread of the reward. The posterior distribution corresponds to another Gaussian mixture model, in which each component acquires an updated mean and weight after incorporating the quadratic reward:

$$q_0(\mathbf{x}) = \sum_{i=1}^{40} \tilde{w}_i \mathcal{N}\left(\mathbf{x}; \tilde{\boldsymbol{\mu}}_i, \tilde{\Sigma}\right),$$

with posterior covariance

$$\tilde{\Sigma} = (\Sigma^{-1} + (50\mathbf{I})^{-1})^{-1},$$

posterior means

$$\tilde{\boldsymbol{\mu}}_i = \tilde{\Sigma} \left((50\mathbf{I})^{-1} \boldsymbol{\mu}_i + \Sigma^{-1} \boldsymbol{\mu} \right),$$

and mixture weights reweighted according to the evidence,

$$\tilde{w}_i \propto w_i \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top (\Sigma + 50\mathbf{I})^{-1}(\boldsymbol{\mu}_i - \boldsymbol{\mu})\right).$$

For all GMM experiments presented in Sec. 4.1, we set the number of particles $N = 2^{13}$ and perform resampling whenever ESS drops below 0.9. All plots in the GMM experiments are plotted by projecting onto the first two dimensions.

Double-Well-4 (DW-4). For both DW-4 and LJ-13 systems, the target is a Boltzmann distribution of the following form:

$$p_0(\mathbf{x}) \propto \exp(-\mathcal{E}(\mathbf{x})) = \exp\left(-\frac{1}{T}\left(H(\mathbf{x}) + \frac{\lambda}{2} \sum_{i=1}^n \|\mathbf{r}_i - \bar{\mathbf{r}}\|^2\right)\right), \quad (\text{B.1})$$

with the potential $H(\mathbf{x})$ system-specific, and a harmonic potential of strength λ imposed as a physical constraint.

In the reward-tilting task, we consider the quadratic reward:

$$r(\mathbf{x}) = -\frac{\lambda'}{2} \sum_{i=1}^n \|\mathbf{r}_i - \bar{\mathbf{r}}\|^2,$$

and thus the reward-tilted distribution is

$$\begin{aligned} p_0(\mathbf{x}) &\propto \exp\left(-\frac{1}{T}\left(H(\mathbf{x}) + \frac{\lambda}{2} \sum_{i=1}^n \|\mathbf{r}_i - \bar{\mathbf{r}}\|^2\right) - \frac{\lambda'}{2} \sum_{i=1}^n \|\mathbf{r}_i - \bar{\mathbf{r}}\|^2\right) \\ &= \exp\left(-\frac{1}{T}\left(H(\mathbf{x}) + \frac{\lambda + \lambda' T}{2} \sum_{i=1}^n \|\mathbf{r}_i - \bar{\mathbf{r}}\|^2\right)\right), \end{aligned}$$

i.e., another Boltzmann distribution of the same temperature but with a different constraint strength $\lambda + \lambda' T$. This reward thus corresponds physically to strengthening the harmonic confinement around the center of mass, *i.e.*, sampling the same particle system at the same temperature but with an increased trap stiffness.

First introduced by (Köhler et al., 2020), the double-well potential is defined on a system of 4 particles in the two-dimensional space ($\mathbf{x} \in \mathbb{R}^{4 \times 2}$). The potential energy function is given as:

$$H_{\text{DW}}(\mathbf{x}) = \frac{1}{2} \sum_{i < j} [a(d_{ij} - d_0) + b(d_{ij} - d_0)^2 + c(d_{ij} - d_0)^4],$$

where \mathbf{r}_i is the coordinate of particle i and $d_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$ denotes the distance between particles i and j . We use standard parameters: $a = 0.0$, $b = -4.0$, $c = 0.9$, and $d_0 = 4.0$. This would yield a double-well potential with two minima at $d_{ij} = 4 \pm \frac{2}{3}\sqrt{5}$. For the harmonic potential constraint in the Boltzmann distribution Eqn. (B.1), we set the constraint strength as $\lambda = 0.05$. For all DW-4 experiments presented in Sec. 4.2, we set the number of particles $N = 2^{15}$ and perform resampling every $K = 100$ steps.

Lennard-Jones-13 (LJ-13). The Lennard-Jones potential is a widely used model for simulating interatomic interactions with both repulsive and attractive components. In our case, it is defined on a system of 13 particles in three-dimensional space ($\mathbf{x} \in \mathbb{R}^{13 \times 3}$). The potential energy function is given as:

$$H_{\text{LJ}}(\mathbf{x}) = \frac{\epsilon}{2T} \sum_{i < j} \left[\left(\frac{\sigma}{d_{ij}} \right)^{12} - 2 \left(\frac{\sigma}{d_{ij}} \right)^6 \right].$$

We use the following parameters: $\sigma = 1.0$ and $\epsilon = 2.0$. For the harmonic potential constraint in the Boltzmann distribution Eqn. (B.1), we set the constraint strength as $\lambda = 1.0$. For all LJ-13 experiments presented in Sec. 4.2, we set the number of particles $N = 2^{13}$ and perform resampling every $K = 50$ steps.

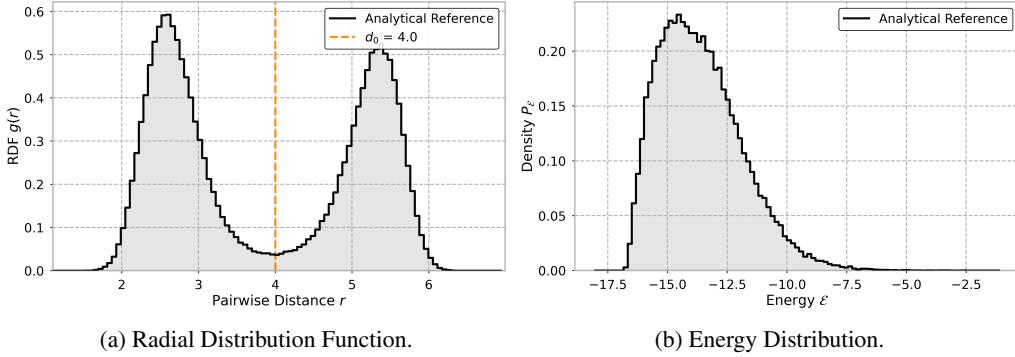


Figure 7: Reference distributions for the DW-4 system at the base temperature $T = 1.5$: (a) Radial Distribution Function (RDF) and (b) Energy Distribution.

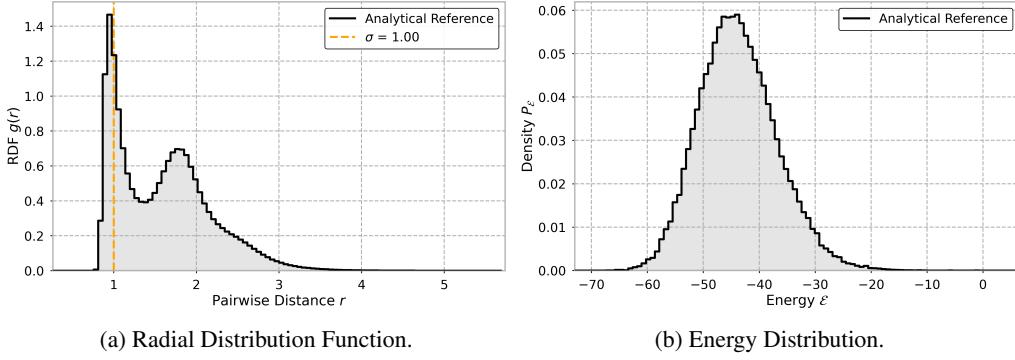


Figure 8: Reference distributions for the LJ-13 system at the base temperature $T = 1.0$: (a) Radial Distribution Function (RDF) and (b) Energy Distribution.

Protein-ligand Co-folding. The protein-ligand co-folding problem is an extension of the classical protein folding problem: rather than predicting only the structure of the protein itself, the task is to simultaneously predict the structure of its interacting counterpart, a ligand that is typically a small molecule. This problem lies at the core of structural biology and is essential for understanding how proteins and ligands interact, which in turn underpins the elucidation of biological pathways and the design of new drug molecules to modulate biological activity. In this work, we focus on addressing the issue of physical validity in protein-ligand co-folding. Diffusion-based models often overemphasize global structural similarity while occasionally generating unphysical conformations. To mitigate this issue, we incorporate physics-based potentials to steer the generative process, effectively modifying the data distribution $p_\theta(\mathbf{x})$ with a physics-based potential function:

$$p_{\text{target}}(\mathbf{x}) \propto p_\theta(\mathbf{x}) \exp(r(\mathbf{x})).$$

Following Passaro et al. (2025), we use a physics reward that penalizes local constraint violations defined in the PoseBuster V2 benchmark. Let \mathbf{x} denote all atomic coordinates, and let $\mathcal{E}_{\text{phys}}(\mathbf{x})$ be a weighted sum of potentials:

$$\mathcal{E}_{\text{phys}}(\mathbf{x}) = \alpha_{\text{bond}} U_{\text{bond}}(\mathbf{x}) + \alpha_{\text{angle}} U_{\text{angle}}(\mathbf{x}) + \alpha_{\text{chir}} U_{\text{chir}}(\mathbf{x}) + \alpha_{\text{int}} U_{\text{int-clash}}(\mathbf{x}) + \alpha_{\text{chain}} U_{\text{chain-clash}}(\mathbf{x}),$$

with nonnegative weights α_{\bullet} . The steering reward is then defined as

$$r(\mathbf{x}) = -\lambda \mathcal{E}_{\text{phys}}(\mathbf{x}).$$

For each covalent bond, bond angle, and chiral center, we apply a flat-bottom potential function that penalizes deviations from the corresponding physical rules while permitting small fluctuations within an acceptable tolerance. This ensures that generated structures remain physically plausible without being overly constrained. Further implementation details of these potentials are provided in

Passaro et al. (2025). Since we also adopt the pre-trained model weights from Passaro et al. (2025), we do not describe the model architecture or training process in details here for which we refer readers to the original work. In this experiment, we use the reward gradient $\nabla r(\mathbf{x})$ as a single basis, and its Laplacian $\Delta r(\mathbf{x})$ is obtained through auto-differentiation.

B.2 NETWORK ARCHITECTURES

The score function $\nabla \log \hat{p}_t$ in both particle experiments (DW-4 and LJ-13) is approximated by an $E(n)$ Equivariant Graph Neural Network (EGNN) (Satorras et al., 2021b;a; Köhler et al., 2020; Klein et al., 2023; Midgley et al., 2023). The network maps a time scalar t and particle coordinates $\mathbf{x} = \{\mathbf{r}_1, \dots, \mathbf{r}_n\}$ to an updated set of coordinates representing the score. All MLPs use Swish activations.

Initialization. Input coordinates are first centered. The scalar time t is encoded using sinusoidal positional embeddings ϕ_t . The resulting vector is then passed through an MLP to create the initial node features $\mathbf{h}_i^{(0)}$, which are identical for all nodes.

$$\mathbf{r}_i^{(0)} = \mathbf{r}_i - \frac{1}{n} \sum_{j=1}^n \mathbf{r}_j, \quad \mathbf{h}_i^{(0)} = \text{MLP}(\phi_t(t)).$$

Equivariant Layers. The network consists of L sequential Equivariant Graph Convolutional Layers (EGCL). For each layer $l \in \{0, \dots, L-1\}$, an initial message $\mathbf{m}_{ij}^{(l)}$ is computed for each edge (i, j) using an MLP $\phi_e^{(l)}$:

$$\mathbf{m}_{ij}^{(l)} = \phi_e^{(l)} \left(\left[\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \|\mathbf{r}_i^{(l)} - \mathbf{r}_j^{(l)}\|^2 \right] \right)$$

where $[\dots]$ denotes concatenation. An attention mechanism then refines each message by multiplying it with a learned gating coefficient:

$$\mathbf{m}_{ij}^{(l)} \leftarrow \mathbf{m}_{ij}^{(l)} \cdot \sigma \left(\phi_{\text{att}}^{(l)}(\mathbf{m}_{ij}^{(l)}) \right)$$

where $\phi_{\text{att}}^{(l)}$ is a single-layer MLP and σ is the sigmoid function. These refined messages are used to produce equivariant updates for the coordinates and invariant updates for the node features via MLPs $\phi_x^{(l)}$ and $\phi_h^{(l)}$. Both updates employ residual connections.

$$\begin{cases} \mathbf{r}_i^{(l+1)} = \mathbf{r}_i^{(l)} + \sum_{j \neq i} \frac{\mathbf{r}_i^{(l)} - \mathbf{r}_j^{(l)}}{\|\mathbf{r}_i^{(l)} - \mathbf{r}_j^{(l)}\| + C} \phi_x^{(l)}(\mathbf{m}_{ij}^{(l)}), \\ \mathbf{h}_i^{(l+1)} = \mathbf{h}_i^{(l)} + \phi_h^{(l)} \left(\left[\mathbf{h}_i^{(l)}, \sum_{j \neq i} \mathbf{m}_{ij}^{(l)} \right] \right), \end{cases}$$

where C is a normalization constant for the coordinate update.

Final Output. The final output of the network is the set of coordinates from the last layer, $\mathbf{r}^{(L)}$, which is re-centered to guarantee overall translation invariance. For each particle, the output is given by centering the coordinates:

$$\mathbf{r}_i^{\text{out}} = \mathbf{r}_i^{(L)} - \frac{1}{n} \sum_{j=1}^n \mathbf{r}_j^{(L)}.$$

For the DW-4 experiment, we use $L = 5$ layers with a hidden dimension of 64, and for the LJ-13 experiment, we use $L = 5$ layers with a hidden dimension of 128. The normalization constant C is set to 1.0 for both experiments.

B.3 SAMPLING DETAILS

To generate configuration samples, we simulate the underdamped Langevin dynamics for a system of N particles in D dimensions. The dynamics target a canonical Boltzmann distribution:

$$\pi(\mathbf{x}, \mathbf{v}) \propto \exp \left(-\frac{1}{T} \left(U(\mathbf{x}) + \frac{\|\mathbf{v}\|^2}{2} \right) \right),$$

where $U(\mathbf{x})$ is the target potential energy ($U(\mathbf{x}) = H(\mathbf{x}) + \frac{\lambda}{2} \sum_{i=1}^n \|\mathbf{r}_i - \bar{\mathbf{r}}\|^2$ in our case), \mathbf{v} are the particle velocities, T is the temperature, and we assume unit mass. The corresponding stochastic differential equations (SDEs) are:

$$\begin{cases} d\mathbf{x}_t &= \mathbf{v}_t dt \\ d\mathbf{v}_t &= \mathbf{F}(\mathbf{x}_t) dt - \gamma \mathbf{v}_t dt + \sqrt{2\gamma T} d\mathbf{w}_t, \end{cases}$$

where $\mathbf{F}(\mathbf{x}) = -\nabla U(\mathbf{x})$ is the force, γ is the friction coefficient, and $(\mathbf{w}_t)_{t \geq 0}$ represents a standard Wiener process. For both experiments, we use $\gamma = 0.5$.

Initialization. Particle positions \mathbf{x}_0 are initialized on a perturbed lattice separated by the characteristic length of each system, and the center of mass is removed. Initial velocities \mathbf{v}_0 are drawn from the equilibrium Maxwell-Boltzmann distribution, with each component sampled independently from $\mathcal{N}(0, T)$.

Numerical Integration. The Langevin SDEs are numerically integrated using the BAOAB splitting scheme (Leimkuhler & Matthews, 2013), which is a highly accurate and stable method for thermostatted systems. For a discrete time step Δt , the update from state $(\mathbf{x}_n, \mathbf{v}_n)$ to $(\mathbf{x}_{n+1}, \mathbf{v}_{n+1})$ is performed in five sequential steps:

$$\begin{aligned} \text{B: } \mathbf{v}_{n+1/2} &= \mathbf{v}_n + \frac{\Delta t}{2} \mathbf{F}(\mathbf{x}_n), \\ \text{A: } \mathbf{x}_{n+1/2} &= \mathbf{x}_n + \frac{\Delta t}{2} \mathbf{v}_{n+1/2}, \\ \text{O: } \hat{\mathbf{v}}_{n+1/2} &= e^{-\gamma \Delta t} \mathbf{v}_{n+1/2} + \sqrt{T(1 - e^{-2\gamma \Delta t})} \xi_n, \\ \text{A: } \mathbf{x}_{n+1} &= \mathbf{x}_{n+1/2} + \frac{\Delta t}{2} \hat{\mathbf{v}}_{n+1/2}, \\ \text{B: } \mathbf{v}_{n+1} &= \hat{\mathbf{v}}_{n+1/2} + \frac{\Delta t}{2} \mathbf{F}(\mathbf{x}_{n+1}), \end{aligned}$$

where ξ_n is a vector of independent standard normal random variables, i.e., $\xi_n \sim \mathcal{N}(0, \mathbf{I})$.

Sampling Protocol. The simulation begins with a burn-in phase, during which the system is evolved for N_{burn} steps to allow it to equilibrate with the stationary distribution. Following this, the sampling phase begins. To ensure that the collected samples are approximately independent, the system evolves for N_{interval} steps between each saved configuration. This process is repeated until the desired number of samples is obtained.

B.4 TRAINING AND INFERENCE DETAILS

We train our score-network using the Elucidating the Design Space of Diffusion Models (EDM) framework (Karras et al., 2022). This approach frames the learning problem as a denoising task, where a single neural network is trained to remove noise from corrupted data across a continuous range of noise levels.

Network Preconditioning. The core component is a neural network $\mathbf{F}(\cdot; \sigma)$ that is wrapped with a set of preconditioning functions dependent on the noise level σ . This design ensures numerical stability and improves performance across all noise scales. The final denoiser, $\mathbf{D}(\mathbf{x}_{\text{noisy}}; \sigma)$, which predicts a clean sample from a noisy one, is defined as:

$$\mathbf{D}(\mathbf{x}_{\text{noisy}}; \sigma) = c_{\text{skip}}(\sigma) \mathbf{x}_{\text{noisy}} + c_{\text{out}}(\sigma) \mathbf{F}(c_{\text{in}}(\sigma) \mathbf{x}_{\text{noisy}}, c_{\text{noise}}(\sigma)). \quad (\text{B.2})$$

The functions c_{skip} , c_{in} , and c_{out} provide scaling and a skip connection, while $c_{\text{noise}}(\sigma)$ creates a time-like embedding from the noise level.

We adopt the preconditioning functions as follows:

$$c_{\text{skip}}(\sigma) = \frac{\sigma_{\text{data}}^2}{\sigma^2 + \sigma_{\text{data}}^2}, \quad c_{\text{in}}(\sigma) = \frac{1}{\sqrt{\sigma^2 + \sigma_{\text{data}}^2}}, \quad c_{\text{out}}(\sigma) = \frac{\sigma \sigma_{\text{data}}}{\sqrt{\sigma^2 + \sigma_{\text{data}}^2}}, \quad c_{\text{noise}}(\sigma) = \frac{1}{4} \log \sigma,$$

where σ_{data} is a hyperparameter varying with the task. We set $\sigma_{\text{data}} = 1.8$ for the DW-4 experiment and $\sigma_{\text{data}} = 0.68$ for the LJ-13 experiment.

Training Objective. The network is trained by corrupting clean data samples $\mathbf{x}_{\text{clean}}$ with Gaussian noise of a given standard deviation σ , creating noisy samples $\mathbf{x}_{\text{noisy}} = \mathbf{x}_{\text{clean}} + \sigma \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. The training objective is to minimize the weighted mean squared error between the denoiser’s prediction and the original clean data. The loss function is given by:

$$\mathcal{L}_{\text{EDM}} = \mathbb{E}_{\mathbf{x}_{\text{clean}} \sim p_0, \sigma, \epsilon \sim \mathcal{N}(0, \mathbf{I})} \left[w(\sigma) \|\mathbf{D}(\mathbf{x}_{\text{noisy}}; \sigma) - \mathbf{x}_{\text{clean}}\|^2 \right],$$

where the distribution of σ and the weighting function $w(\sigma)$ will be specified later.

Following the EDM framework, the noise level σ is sampled from a log-normal distribution:

$$\log \sigma \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2),$$

with $P_{\text{mean}} = -1.2$ and $P_{\text{std}} = 1.2$. To avoid numerical instability, we clip the noise level to be within the range $[\sigma_{\min}, \sigma_{\max}]$ with $\sigma_{\min} = 0.002$ and $\sigma_{\max} = 80$. The weighting function is set as

$$w(\sigma) = \frac{\sigma^2 + \sigma_{\text{data}}^2}{\sigma^2 \sigma_{\text{data}}^2}.$$

Inference Process. While the model is trained as a denoiser, sampling requires the score function, $\nabla_{\mathbf{x}} \log p(\mathbf{x}_{\text{noisy}}; \sigma)$. The trained denoiser \mathbf{D} is converted to the score during inference using the following exact relation:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}_{\text{noisy}}; \sigma) = \frac{\mathbf{D}(\mathbf{x}_{\text{noisy}}; \sigma) - \mathbf{x}_{\text{noisy}}}{\sigma^2}.$$

The noise level σ coincides with the forward time s in our setting in Eqn. (2.1) and we have $T = \sigma'_{\max}$. The time discretization is chosen as

$$\{t_k\}_{k=0}^M = \left\{ \sigma'_{\max} - \left(\sigma'^{1/\rho}_{\max} + \frac{k}{M} \left(\sigma'^{1/\rho}_{\min} - \sigma'^{1/\rho}_{\max} \right) \right)^{\rho} \right\}_{k=0}^M,$$

where we choose a smaller range of noise levels for the parameters $\sigma'_{\min} = 0.005$ and $\sigma'_{\max} = 50$ to avoid the boundary part of the noise level, which may be less accurate due to the lack of training and singularity. We use $\rho = 7$ to control the spacing of the discretization. We use $M = 500$ for GMM experiments and $M = 1000$ time steps for particle system experiments.

Training Settings. For the neural network $\mathbf{F}(\cdot; \sigma)$ in the denoiser (B.2), we adopt the EGNN architecture described in App. B.2. For the training process, we use the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 5×10^{-4} and a batch size of 1024. We train the network for 10^6 steps, where we sample a fresh dataset of 10^6 data points with the same protocol as described in App. B.3 for every 2×10^5 steps. The warm-up period is only performed once before the first batch. All training is performed on a single NVIDIA A100 GPU. The training code is based on both the JAX library (Bradbury et al., 2018) and the Optax gradient processing and optimization library (DeepMind et al., 2020).

B.5 EVALUATION METRICS

To quantitatively assess the quality of the samples generated by each method against a ground-truth reference distribution, we employ a suite of five complementary metrics. Let the generated samples be a weighted set $\{\mathbf{x}^{(i)}, w^{(i)}\}_{i \in [N]}$ and the reference samples be $\{\mathbf{x}_{\text{ref}}^{(j)}, w_{\text{ref}}^{(j)}\}_{j \in [N_{\text{ref}}]}$.

Negative Log-Likelihood Difference (ΔNLL). The Negative Log-Likelihood (NLL) measures how well a set of samples fits the target probability distribution $p(\mathbf{x})$. For a weighted set of samples, it is estimated as the weighted average of the negative log-probabilities:

$$\text{NLL} = - \sum_{i=1}^N w^{(i)} \log p(\mathbf{x}^{(i)}).$$

We report ΔNLL , which is the difference between the NLL of the samples generated by a method and the NLL of the reference samples: $\Delta \text{NLL} = \text{NLL}_{\text{method}} - \text{NLL}_{\text{ref}}$. A lower absolute value indicates a better fit to the target distribution.

Maximum Mean Discrepancy (MMD). Maximum Mean Discrepancy (MMD) is an integral probability metric defined in a Reproducing Kernel Hilbert Space (RKHS) \mathcal{H} with a characteristic kernel $k(\cdot, \cdot)$. Any probability distribution P admits a unique *mean embedding* $\mu_P \in \mathcal{H}$. The squared MMD between two distributions P and Q is the squared RKHS distance between their embeddings:

$$\text{MMD}^2(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}}^2 = \|\mathbb{E}_{\mathbf{x} \sim P}[k(\mathbf{x}, \cdot)] - \mathbb{E}_{\mathbf{y} \sim Q}[k(\mathbf{y}, \cdot)]\|_{\mathcal{H}}^2.$$

Using the kernel trick, this definition can be expressed entirely in terms of kernel evaluations, avoiding explicit computation of the embeddings. For two weighted sample sets, the unbiased empirical estimator is

$$\begin{aligned} \text{MMD}^2 &= \|\mathbb{E}_{\mathbf{x} \sim P}[k(\mathbf{x}, \cdot)] - \mathbb{E}_{\mathbf{y} \sim Q}[k(\mathbf{y}, \cdot)]\|_{\mathcal{H}}^2 \\ &= \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim P}[k(\mathbf{x}, \mathbf{x}')] + \mathbb{E}_{\mathbf{y}, \mathbf{y}' \sim Q}[k(\mathbf{y}, \mathbf{y}')] - 2\mathbb{E}_{\mathbf{x} \sim P, \mathbf{y} \sim Q}[k(\mathbf{x}, \mathbf{y})] \\ &\approx \sum_{i, j=1}^N w^{(i)} w^{(j)} k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) + \sum_{i, j=1}^{N_{\text{ref}}} w_{\text{ref}}^{(i)} w_{\text{ref}}^{(j)} k(\mathbf{x}_{\text{ref}}^{(i)}, \mathbf{x}_{\text{ref}}^{(j)}) - 2 \sum_{i=1}^N \sum_{j=1}^{N_{\text{ref}}} w^{(i)} w_{\text{ref}}^{(j)} k(\mathbf{x}^{(i)}, \mathbf{x}_{\text{ref}}^{(j)}). \end{aligned}$$

We use the RBF kernel

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma_k^2}\right),$$

with $\sigma_k = 20$ for the GMM experiment and $\sigma_k = 5$ for the DW-4 and LJ-13 experiments.

Direct evaluation of MMD requires $\mathcal{O}(N^2)$ kernel evaluations, which becomes expensive for large sample sets. To scale up computation, we use *Random Fourier Features (RFF)* to approximate the RBF kernel with an explicit low-dimensional feature map $\mathbf{z}(\mathbf{x}) \in \mathbb{R}^f$, reducing the complexity to $\mathcal{O}(NN_f)$.

Bochner’s theorem states that a shift-invariant kernel can be expressed as the expectation of a product of complex Fourier features with respect to its spectral density $p(\boldsymbol{\omega})$:

$$k(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{\boldsymbol{\omega} \sim p(\boldsymbol{\omega})} \left[e^{i\boldsymbol{\omega}^\top \mathbf{x}} e^{-i\boldsymbol{\omega}^\top \mathbf{y}} \right].$$

For the RBF kernel, $p(\boldsymbol{\omega})$ is Gaussian, $\mathcal{N}(0, \sigma_k^{-2}\mathbf{I})$. The RFF approximation replaces the expectation with a Monte Carlo average over $f/2$ sampled frequencies $\boldsymbol{\omega}_k \sim \mathcal{N}(0, \sigma_k^{-2}\mathbf{I})$ and random phases $b_k \sim \text{Unif}[0, 2\pi]$, yielding the explicit feature map

$$\mathbf{z}(\mathbf{x}) = \sqrt{\frac{2}{f}} \left(\cos(\boldsymbol{\omega}_1^\top \mathbf{x} + b_1), \dots, \cos(\boldsymbol{\omega}_{f/2}^\top \mathbf{x} + b_{f/2}), \sin(\boldsymbol{\omega}_1^\top \mathbf{x} + b_1), \dots, \sin(\boldsymbol{\omega}_{f/2}^\top \mathbf{x} + b_{f/2}) \right)^\top.$$

This ensures that $\mathbf{z}(\mathbf{x})^\top \mathbf{z}(\mathbf{y}) \approx k(\mathbf{x}, \mathbf{y})$. The MMD can then be efficiently approximated as the squared Euclidean distance between the mean feature vectors:

$$\text{MMD}^2 \approx \|\mathbb{E}_{\mathbf{x} \sim P}[\mathbf{z}(\mathbf{x})] - \mathbb{E}_{\mathbf{y} \sim Q}[\mathbf{z}(\mathbf{y})]\|_2^2 \approx \left\| \sum_{i=1}^N w^{(i)} \mathbf{z}(\mathbf{x}^{(i)}) - \sum_{j=1}^{N_{\text{ref}}} w_{\text{ref}}^{(j)} \mathbf{z}(\mathbf{x}_{\text{ref}}^{(j)}) \right\|_2^2.$$

In our experiments, we use 2048 random features ($f = 2048$) for all experiments.

Sliced-Wasserstein Distance (SWD). The Wasserstein distance measures the minimum cost to transport mass from one distribution to another. As computing exact 2-Wasserstein distance between high-dimensional empirical distributions with up to 2^{15} particles in our experiments is computationally prohibitive, we report Sliced-Wasserstein Distance (SWD) as a computationally efficient approximation that involves projecting the high-dimensional distributions onto a series of random 1D lines, calculating the 1D Wasserstein-2 distance for each projection, and averaging the results:

$$\text{SWD}_2(P, Q)^2 = \int_{\mathbb{S}^{D-1}} W_2^2(\pi_\theta(P), \pi_\theta(Q)) d\theta \approx \frac{1}{p} \sum_{i=1}^p W_2^2(\pi_{\theta_i}(P), \pi_{\theta_i}(Q)),$$

where π_θ denotes the projection onto a line with direction $\theta \in \mathbb{S}^{D-1}$, p is the number of projections, and $\theta_i \sim \text{Unif}(\mathbb{S}^{D-1})$. In our experiments, we use $p = 10$ for all experiments.

Radial Distribution Function Wasserstein-1 Distance (W_1^{RDF}). The Radial Distribution Function (RDF), $g(r)$, is a fundamental descriptor of the spatial arrangement of particles in a system, measuring the probability of finding a particle at a distance r from another. For simplicity and following the literature (Akhound-Sadegh et al., 2024; Skreta et al., 2025), we define it as the ensemble-averaged interatomic distance distribution:

$$g(r) = \frac{2}{N(N-1)} \sum_{i < j} \langle \delta(r - \|\mathbf{r}_i - \mathbf{r}_j\|) \rangle,$$

where $\langle \cdot \rangle$ denotes the ensemble average over all configurations. With slight abuse of notation, we still refer to $g(r)$ as the RDF without the shell factor and density normalization.

To assess the structural accuracy of our samples, we compute the RDF for both the generated and reference configurations, yielding two 1D distributions, $g_{\text{method}}(r)$ and $g_{\text{ref}}(r)$. The W_1^{RDF} metric is the 1-Wasserstein distance between these two distributions, $W_1(g_{\text{method}}, g_{\text{ref}})$, which quantifies the difference in the learned physical structure.

Energy Wasserstein-1 Distance ($W_1^{\mathcal{E}}$). The energy of a configuration \mathbf{x} is its negative log-probability, $\mathcal{E}(\mathbf{x}) = -\log q(\mathbf{x})$. Due to the unknown normalization constant for the reward-tilted and annealed distribution $q(\mathbf{x}) \propto p^\gamma(\mathbf{x})e^{r(\mathbf{x})}$, we report the energy up to a constant, *i.e.*,

$$\mathcal{E}(\mathbf{x}) = -\gamma \log p(\mathbf{x}) - r(\mathbf{x}).$$

By evaluating the energy for every sample, we obtain two 1D distributions of energies, $P_{\mathcal{E}}^{\text{method}}$ and $P_{\mathcal{E}}^{\text{ref}}$. The $W_1^{\mathcal{E}}$ metric is the 1-Wasserstein distance, $W_1(P_{\mathcal{E}}^{\text{method}}, P_{\mathcal{E}}^{\text{ref}})$, between these energy distributions. This metric evaluates how well a method captures the correct energy landscape and the relative probabilities of different configurations.

Physical Validity. To evaluate the physical plausibility of generated protein-ligand structures, we adopt the validity metrics from PoseBusters (Buttenschoen et al., 2024), which assess adherence to fundamental physical and chemical rules.

- Valid Fraction: The fraction of generated structures that satisfy all validity checks simultaneously, serving as an overall measure of correctness.
- Clash-Free Fraction: The proportion of structures without severe steric clashes, *i.e.*, unphysical overlaps between atoms.
- Bond Length and Bond Angle Violations: Counts of deviations from standard covalent geometry. Bond length violations occur when bonds are too short or too long, while bond angle violations correspond to unrealistic angular geometries.
- Internal Clashes: The number of steric overlaps within the same ligand molecule, reflecting poor internal consistency.
- Chiral Atom Errors: The number of stereocenters incorrectly assigned, such as inversions of chirality.
- Chain Clashes: The number of steric overlaps between ligand atoms and protein atoms, indicating violations of intermolecular packing constraints.

C ADDITIONAL EXPERIMENTAL RESULTS

This appendix provides supplementary results that further explore the performance of our methods, DriftLite-VCG(-SMC) and DriftLite-ECG(-SMC), under various conditions for both the Gaussian Mixture Model (GMM) and the particle-based systems (DW-4 and LJ-13).

C.1 ADDITIONAL EXPERIMENTAL RESULTS OF GMM

We first present a more extensive ablation study on the GMM. Tab. 3 details the performance of all methods across a wider range of annealing factors, $\gamma \in \{1.5, 2.0, 2.5, 3.0\}$. The higher the annealing factor γ becomes, the more challenging the inference-time scaling task is, as it accentuates

the modes of the distribution and increases the energy barriers between them. The results reinforce the conclusions from the main text: as γ grows, the performance gap between our DriftLite methods and the baselines widens considerably. VCG-SMC, in particular, consistently achieves the best or near-best performance across all metrics, showcasing its robustness.

Similarly, Tab. 4 extends the reward-tilting experiments to different reward strengths ($\sigma \in \{100, 200, 300, 400\}$). For the definition of the parameter σ , we refer to the problem settings in App. B.1. Again, our methods demonstrate superior stability and accuracy compared to Pure Guidance (PG) and G-SMC, which degrade significantly as the reward becomes more peaked (smaller reward covariance scale σ).

Table 3: Performance ablation for the GMM annealing task with varying annealing factor γ . Results are mean \pm std over 5 configurations. Best results per column (within each γ block) are in bold.

| Method | Annealing ($\gamma = 1.5$) | | | | | Annealing ($\gamma = 2.0$) | | | | |
|---------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| | Δ NLL | MMD $\times 10^{-1}$ | SWD | Mean L_2 | Cov F $\times 10^3$ | Δ NLL | MMD $\times 10^{-1}$ | SWD | Mean L_2 | Cov F $\times 10^3$ |
| PG | -1.196 \pm 0.621 | 4.471 \pm 2.452 | 11.90 \pm 6.391 | 58.98 \pm 33.92 | 5.274 \pm 0.660 | -3.674 \pm 0.774 | 6.576 \pm 2.205 | 18.06 \pm 6.494 | 80.10 \pm 33.03 | 9.347 \pm 2.428 |
| G-SMC | 0.441 \pm 0.099 | 0.834 \pm 0.070 | 5.749 \pm 1.368 | 29.15 \pm 2.890 | 3.097 \pm 0.389 | -0.527 \pm 0.262 | 2.057 \pm 0.497 | 11.44 \pm 3.625 | 58.71 \pm 14.71 | 5.097 \pm 0.575 |
| ECG | 0.257 \pm 0.083 | 0.185 \pm 0.003 | 0.622 \pm 0.097 | 2.982 \pm 0.331 | 0.350 \pm 0.033 | 0.324 \pm 0.145 | 0.415 \pm 0.037 | 1.209 \pm 0.164 | 5.323 \pm 0.688 | 0.750 \pm 0.069 |
| ECG-SMC | 0.219 \pm 0.053 | 0.162\pm0.004 | 0.605 \pm 0.139 | 2.667 \pm 0.687 | 0.335 \pm 0.036 | 0.198 \pm 0.073 | 0.185 \pm 0.009 | 0.779 \pm 0.131 | 3.782 \pm 0.476 | 0.446 \pm 0.045 |
| VCG | 0.222 \pm 0.045 | 0.166\pm0.002 | 0.590\pm0.058 | 2.661\pm0.172 | 0.335\pm0.018 | 0.204 \pm 0.052 | 0.188 \pm 0.001 | 0.672 \pm 0.093 | 2.971 \pm 0.175 | 0.379 \pm 0.030 |
| VCG-SMC | 0.203\pm0.061 | 0.162\pm0.002 | 0.638 \pm 0.055 | 2.852 \pm 0.058 | 0.346 \pm 0.021 | 0.192\pm0.063 | 0.166\pm0.005 | 0.606\pm0.094 | 2.866\pm0.543 | 0.344\pm0.027 |
| Method | Annealing ($\gamma = 2.5$) | | | | | Annealing ($\gamma = 3.0$) | | | | |
| | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ |
| PG | -5.016 \pm 1.280 | 0.725 \pm 0.208 | 20.27 \pm 6.796 | 92.04 \pm 30.59 | 9.290 \pm 2.339 | -4.950 \pm 1.342 | 0.758 \pm 0.158 | 19.55 \pm 4.731 | 93.39 \pm 21.64 | 8.654 \pm 1.850 |
| G-SMC | -0.801 \pm 0.204 | 0.327 \pm 0.073 | 13.88 \pm 2.770 | 78.11 \pm 18.39 | 5.829 \pm 1.041 | -0.692 \pm 0.414 | 0.493 \pm 0.071 | 18.31 \pm 3.203 | 102.7 \pm 9.285 | 4.973 \pm 1.255 |
| ECG | -0.427 \pm 1.185 | 0.248 \pm 0.255 | 6.443 \pm 6.144 | 30.08 \pm 28.89 | 3.135 \pm 2.090 | -1.201 \pm 1.155 | 0.353 \pm 0.192 | 8.967 \pm 4.501 | 39.24 \pm 21.08 | 4.858 \pm 1.636 |
| ECG-SMC | 0.169\pm0.070 | 0.021 \pm 0.002 | 1.002 \pm 0.196 | 4.886 \pm 1.149 | 0.564 \pm 0.088 | 0.184 \pm 0.079 | 0.031 \pm 0.003 | 1.672 \pm 2.14 | 7.795 \pm 1.164 | 0.850 \pm 0.107 |
| VCG | 0.204 \pm 0.058 | 0.023 \pm 0.001 | 0.717 \pm 0.108 | 3.351 \pm 0.222 | 0.420 \pm 0.037 | 0.209 \pm 0.080 | 0.029 \pm 0.002 | 0.859 \pm 0.147 | 4.071 \pm 0.507 | 0.505 \pm 0.055 |
| VCG-SMC | 0.179 \pm 0.065 | 0.018\pm0.001 | 0.613\pm0.109 | 2.867\pm0.364 | 0.380\pm0.051 | 0.174\pm0.073 | 0.019\pm0.001 | 0.691\pm0.149 | 3.319\pm0.593 | 0.415\pm0.048 |

Table 4: Performance ablation for the GMM reward-tilting task with varying reward strength σ . Results are mean \pm std over 5 runs. Best results per column (within each σ block) are in bold.

| Method | Reward-Tilting ($\sigma = 100$) | | | | | Reward-Tilting ($\sigma = 200$) | | | | |
|---------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ |
| PG | 21.24 \pm 3.955 | 0.903 \pm 0.091 | 13.57 \pm 6.615 | 85.34 \pm 32.89 | 8.159 \pm 4.769 | 5.454 \pm 2.418 | 0.825 \pm 0.048 | 12.77 \pm 6.515 | 73.51 \pm 32.29 | 5.845 \pm 3.506 |
| G-SMC | 0.439 \pm 1.184 | 0.249 \pm 0.077 | 2.625 \pm 0.970 | 15.42 \pm 7.339 | 0.683 \pm 0.630 | 0.422 \pm 0.414 | 0.086 \pm 0.025 | 1.072 \pm 0.490 | 5.735 \pm 2.680 | 0.463 \pm 0.226 |
| ECG | 0.854 \pm 0.901 | 0.119 \pm 0.148 | 1.120 \pm 1.445 | 4.496 \pm 5.837 | 0.160 \pm 0.124 | 0.777 \pm 1.021 | 0.115 \pm 0.086 | 1.306 \pm 0.868 | 5.613 \pm 3.550 | 0.287 \pm 0.113 |
| ECG-SMC | 0.309 \pm 0.067 | 0.020\pm0.002 | 0.234\pm0.098 | 0.996 \pm 0.436 | 0.065 \pm 0.049 | 0.304 \pm 0.076 | 0.025 \pm 0.002 | 0.360 \pm 0.113 | 1.795 \pm 0.837 | 0.115 \pm 0.052 |
| VCG | 0.262\pm0.101 | 0.032 \pm 0.004 | 0.284 \pm 0.052 | 1.118 \pm 0.276 | 0.059\pm0.025 | 0.256\pm0.099 | 0.035 \pm 0.005 | 0.394 \pm 0.057 | 1.601\pm0.303 | 0.100\pm0.043 |
| VCG-SMC | 0.338 \pm 0.133 | 0.020 \pm 0.003 | 0.236 \pm 0.120 | 0.931\pm0.569 | 0.061 \pm 0.046 | 0.348 \pm 0.094 | 0.020\pm0.001 | 0.352\pm0.141 | 1.636 \pm 0.647 | 0.113 \pm 0.058 |
| Method | Annealing ($\sigma = 300$) | | | | | Annealing ($\sigma = 400$) | | | | |
| | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ |
| PG | 1.838 \pm 1.657 | 0.660 \pm 0.072 | 11.06 \pm 5.357 | 58.23 \pm 26.81 | 5.238 \pm 2.293 | 0.618 \pm 1.201 | 0.497 \pm 0.083 | 9.187 \pm 3.991 | 45.60 \pm 20.95 | 3.952 \pm 1.483 |
| G-SMC | 0.207\pm0.148 | 0.047 \pm 0.007 | 0.664 \pm 0.312 | 3.447 \pm 1.402 | 0.284 \pm 0.158 | 0.253\pm0.140 | 0.036 \pm 0.004 | 0.814 \pm 0.272 | 4.917 \pm 1.998 | 0.377 \pm 0.134 |
| ECG | 0.664 \pm 0.899 | 0.132 \pm 0.089 | 1.572 \pm 0.813 | 7.141 \pm 3.575 | 0.434 \pm 0.117 | 0.463 \pm 0.498 | 0.122 \pm 0.084 | 1.611 \pm 0.829 | 7.590 \pm 3.780 | 0.535 \pm 0.138 |
| ECG-SMC | 0.661 \pm 0.538 | 0.048 \pm 0.037 | 0.740 \pm 0.302 | 3.965 \pm 2.116 | 0.258 \pm 0.103 | 0.327 \pm 0.195 | 0.039 \pm 0.012 | 0.888 \pm 0.182 | 4.433 \pm 0.965 | 0.394 \pm 0.098 |
| VCG | 0.227 \pm 0.109 | 0.034 \pm 0.007 | 0.498 \pm 0.080 | 1.913 \pm 0.383 | 0.157 \pm 0.037 | 0.290 \pm 0.121 | 0.033 \pm 0.006 | 0.550 \pm 0.056 | 2.238 \pm 0.332 | 0.203 \pm 0.035 |
| VCG-SMC | 0.315 \pm 0.082 | 0.022\pm0.001 | 0.355\pm0.075 | 1.544\pm0.306 | 0.150\pm0.024 | 0.292 \pm 0.055 | 0.022\pm0.001 | 0.454\pm0.107 | 2.072\pm0.476 | 0.184\pm0.046 |

Finally, Fig. 9 and Fig. 10 complement the figures in Sec. 4.1. Fig. 9 illustrates the evolution of ESS and variance for a milder reward-tilting task ($\sigma = 50.0$), showing that even in less challenging scenarios, our control mechanism actively stabilizes the particle weights. Fig. 10 shows the performance of all methods as a function of the number of particles for a strong reward-tilting task ($\sigma = 400.0$). It clearly indicates that our VCG and ECG methods not only outperform the baselines but also converge more efficiently, achieving better results with fewer particles.

As shown in Tab. 5, for the 30-dimensional GMM problem, the relative runtime for our VCG-SMC and ECG-SMC methods is only around 5 to 6 times that of the standard Guidance-SMC (G-SMC) baseline. This modest increase in computation time is largely due to the effective parallelization of

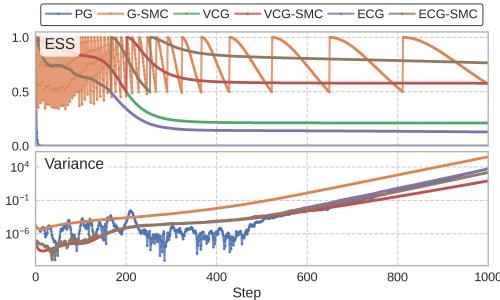


Figure 9: Evolution of ESS and potential variance for the GMM reward-tilting task ($\sigma = 50.0$).

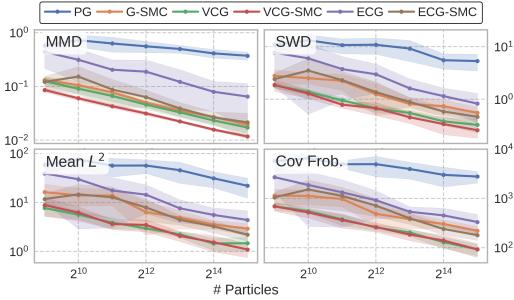


Figure 10: Performance metrics versus number of particles for the GMM reward-tilting task ($\sigma = 400.0$).

Table 5: Elapsed time comparison for different sampling methods. Results are $\text{mean}_{\pm \text{std}}$ over 5 runs. The relative runtime, including propagated standard deviation, is computed with respect to the G-SMC method.

| Method | Annealing ($\gamma = 2.5$) | | Reward-Tilting ($\sigma = 100$) | |
|---------|------------------------------|--------------------------|-----------------------------------|--------------------------|
| | Runtime (s) | Relative Runtime | Runtime (s) | Relative Runtime |
| G-SMC | $6.44_{\pm 3.28}$ | $1.00_{\pm 0.72} \times$ | $6.90_{\pm 3.01}$ | $1.00_{\pm 0.62} \times$ |
| VCG-SMC | $39.78_{\pm 0.93}$ | $6.18_{\pm 3.15} \times$ | $40.22_{\pm 0.49}$ | $5.83_{\pm 2.54} \times$ |
| ECG-SMC | $39.11_{\pm 0.50}$ | $6.07_{\pm 3.09} \times$ | $39.33_{\pm 0.59}$ | $5.70_{\pm 2.49} \times$ |

the core algorithms. These results underscore the controlled additional cost required to implement the variance reduction technique, making them commonly practical. A more optimized implementation could reduce this runtime gap even further by exploiting more GPU resources.

We further conducted GMM experiments on the annealing task ($\gamma = 2.5$) and the quadratic reward-tilting task ($\sigma = 100$) under a matched wall-clock budget (Tab. 6). In these runs, the baselines (PG, G-SMC) used 8192 particles and 2000 diffusion steps, while DriftLite variants (VCG/ECG, with or without SMC) used 1024 particles and 1000 diffusion steps. Under this configuration, the wall-clock times of DriftLite methods are comparable to, and in some cases slightly lower than, those of PG and G-SMC, despite using substantially fewer particles and steps. At the same time, DriftLite variants, especially VCG-SMC and ECG-SMC, achieve markedly lower MMD, SWD, and mean- ℓ_2 errors, while maintaining ΔNLL close to zero. These results indicate that the added control drift substantially improves sample quality at essentially fixed computational cost, mitigating the variance and weight degeneracy that limit the uncontrolled SMC baseline.

Comparison with a training-based neural drift controller (NCG). To contextualize the trade-off between training-free inference-time control and amortized drift-learning approaches, we implemented a prototype training-based baseline, which we refer to as *Neural Controlling Guidance (NCG)*.

In NCG, the control drift is parameterized by a neural network $b_{t,\theta}(x)$ taking time and state as inputs, in contrast to DriftLite which solves for the control coefficients via small linear systems under a linear ansatz (cf. Eq. (3.3) and Ansatz 3.3). Concretely, we train $b_{t,\theta}$ by minimizing a regularized version of the same variance objective:

$$\mathcal{L}_t(\theta) = \text{Var}_{x \sim q_t} [g_t(x) + h_t(x; b_{t,\theta})] + \lambda \|b_{t,\theta}\|_2^2,$$

with $\lambda = 0.1$. We use a U-Net architecture with encoder feature dimensions [128, 256, 512] and decoder feature dimensions [256, 128]. For training, we maintain an ensemble of $N = 8192$ particles (matching the inference-only baselines), and after each time step, we backpropagate $\nabla_\theta \mathcal{L}_t(\theta)$ and update parameters using a learning rate 5×10^{-4} . We repeat this training–inference procedure over multiple refinement rounds, carrying the updated dynamics from the previous round as the base for the next.

Table 6: Performance with matched wall-clock budgets for the GMM tasks. Results are mean \pm std over 5 runs. Best results per column are in bold.

| Method | Annealing ($\gamma = 2.5$) | | | | | Reward-Tilting ($\sigma = 100$) | | | | |
|----------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|----------------------------------|-----------------------------------|---------------------------------|---------------------------------|---------------------------------|----------------------------------|
| | ΔNLL | MMD | SWD | Mean L2 | Runtime (s) | ΔNLL | MMD | SWD | Mean L2 | Runtime (s) |
| PG | -5.04 \pm 1.31 | 0.73 \pm 0.22 | 18.52 \pm 5.00 | 90.58 \pm 29.89 | 17.28 \pm 0.49 | 37.92 \pm 6.06 | 0.96 \pm 0.06 | 19.20 \pm 7.74 | 112.27 \pm 26.70 | 15.98 \pm 0.51 |
| G-SMC | -0.56 \pm 0.85 | 0.46 \pm 0.07 | 18.87 \pm 3.69 | 102.17 \pm 20.40 | 18.45 \pm 0.84 | 1.23 \pm 1.39 | 0.25 \pm 0.08 | 2.81 \pm 1.40 | 17.13 \pm 9.97 | 16.74 \pm 0.31 |
| VCG | 0.11 \pm 0.20 | 0.06 \pm 0.00 | 1.85 \pm 0.36 | 7.76 \pm 1.55 | 15.12 \pm 0.14 | 0.36 \pm 0.21 | 0.09 \pm 0.01 | 0.83 \pm 0.12 | 3.19 \pm 0.73 | 14.96 \pm 0.01 |
| VCG-SMC | 0.04\pm0.23 | 0.05\pm0.00 | 1.75\pm0.41 | 7.33\pm1.61 | 15.77 \pm 0.13 | 0.25 \pm 0.19 | 0.05\pm0.01 | 0.59\pm0.17 | 2.57\pm0.90 | 15.55 \pm 0.10 |
| ECG | -0.75 \pm 1.27 | 0.40 \pm 0.08 | 10.22 \pm 2.29 | 50.45 \pm 9.10 | 13.92\pm0.18 | 0.39 \pm 0.76 | 0.23 \pm 0.10 | 1.80 \pm 0.64 | 8.49 \pm 3.84 | 14.73\pm0.09 |
| ECG-SMC | 0.08 \pm 0.25 | 0.07 \pm 0.01 | 2.69 \pm 0.61 | 14.29 \pm 4.05 | 14.83 \pm 0.19 | 0.01\pm0.12 | 0.06 \pm 0.01 | 1.03 \pm 0.58 | 5.93 \pm 3.52 | 15.30 \pm 0.05 |

Table 7: Comparison with training-based Neural Controlling Guidance (NCG) on the 30D GMM. NCG-SMC- r denotes the r -th refinement/training round. *Inference Only* evaluates the learned drift from round 20 without backpropagation. Lower is better for all metrics. *Best-performing NCG round among those reported.

| Method | MMD | SWD | Mean L_2 | ΔNLL | Runtime (s) |
|-----------------------------|-------|-------|------------|--------------------|-------------|
| PG | 0.825 | 21.86 | 66.09 | -0.65 | 4.59 |
| G-SMC | 0.143 | 6.45 | 24.94 | 0.04 | 5.53 |
| VCG-SMC | 0.017 | 0.52 | 1.23 | 0.14 | 7.83 |
| ECG-SMC | 0.019 | 0.68 | 1.83 | 0.14 | 6.99 |
| NCG-SMC-1 | 0.437 | 12.29 | 51.44 | 0.39 | 70.79 |
| NCG-SMC-5 | 0.135 | 3.53 | 13.67 | 0.34 | 307.34 |
| NCG-SMC-9* | 0.091 | 2.45 | 9.11 | 0.05 | 543.77 |
| NCG-SMC-12 | 0.223 | 13.35 | 37.96 | 0.01 | 721.23 |
| NCG-SMC-16 | 0.143 | 7.61 | 21.73 | 0.16 | 957.72 |
| NCG-SMC-20 | 0.199 | 9.36 | 24.02 | 0.08 | 1194.31 |
| NCG-SMC-20 (Inference Only) | 0.190 | 8.91 | 29.04 | 0.26 | 240.88 |

Overall, NCG improves substantially during early refinement rounds but exhibits non-monotonic behavior as training continues, highlighting the optimization sensitivity of variance-based drift learning. Moreover, even without training (Inference Only), evaluating a high-capacity neural controller incurs significant inference-time overhead relative to DriftLite’s lightweight linear solve.

C.2 ADDITIONAL EXPERIMENTAL RESULTS OF PARTICLE SYSTEMS

Here, we provide additional ablation studies for the DW-4 and LJ-13 particle systems, demonstrating the robustness of our findings.

ESS and Potential Variance Evolution. In Fig. 12, we present a similar visualization as in the GMM example (Figs. 2 and 9), where similar trends are observed: pure guidance leads to a much rapid drop in ESS during the initial stage of the inference process, while the variance-controlling guidance postpones the drop when the resampling kicks in. We also observe that the time when the curve of the potential variance $\text{Var}_{q_t}[g_t]$ of all methods meets coincides with the drop in ESS of the variance-controlling guidance, which may correspond to the splitting of modes, and resampling may be crucial to handle.

Elapsed Time Comparison. The results for the DW4 system show that the performance scaling of the advanced methods is even more favorable than what was observed for the GMM task (Tab. 5). From Tab. 8, we can see that the relative runtime of the VCG-SMC methods is only about twice that of the G-SMC baseline. This remarkable efficiency stems from the nature of the problem itself; the total computational cost is dominated by repeated, computationally heavy score evaluations. Consequently, the fixed algorithmic overhead from our DriftLite variance reduction methods becomes negligible as a fraction of the total runtime, underscoring their practicality and efficiency for computationally intensive systems.

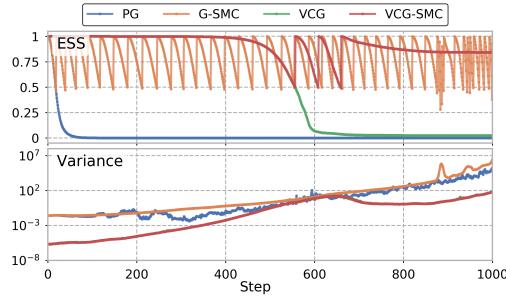


Figure 11: Evolution of ESS and potential variance for the DW-4 annealing task ($\gamma = 2.0$).

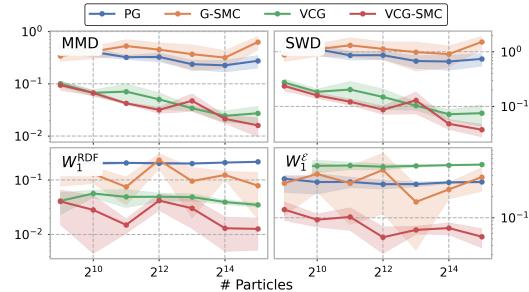


Figure 12: Performance metrics versus number of particles for the DW-4 annealing task ($\gamma = 2.0$).

Table 8: Elapsed time comparison for different sampling methods on the DW4 System. Results are mean \pm std over 5 runs. The relative runtime, including propagated standard deviation, is computed with respect to the G-SMC method.

| Method | Annealing ($\gamma = 2.0$) | | Reward-Tilting ($\lambda' = 0.5$) | |
|----------------|------------------------------|--------------------------|-------------------------------------|--------------------------|
| | Runtime (s) | Relative Runtime | Runtime (s) | Relative Runtime |
| G-SMC | 281.68 \pm 9.69 | 1.00 \pm 0.05 \times | 284.12 \pm 8.03 | 1.00 \pm 0.04 \times |
| VCG-SMC | 674.32 \pm 6.89 | 2.39 \pm 0.09 \times | 674.12 \pm 1.75 | 2.37 \pm 0.07 \times |

Ablation on Number of Particles N . Next, we present the ablation study on the number of particles on the DW-4 system as in the GMM example (Fig. 12). Despite a more complex problem nature and high-level evaluation metrics (RDF and energy distribution), our methods still present robust scaling in MMD and SWD, and show promising decay in W_1^{RDF} and $W_1^{\mathcal{E}}$, highlighting the effectiveness and robustness of our methods.

Ablation on Base Model Temperature T . In Tab. 9, we investigate the effect of the base model’s training temperature ($T \in \{1.0, 1.5\}$) on an annealing task for the DW-4 system in addition to the base temperature $T = 2.0$ as used in Tab. 1. The results show that our methods, particularly VCG-SMC, are effective regardless of the starting temperature. They successfully anneal the system to the target low-temperature state, consistently outperforming the baselines and confirming that the control mechanism adapts well to different initial dynamics.

Table 9: Ablation results on Annealing with $\gamma = 2.0$ from Base Temperature $T \in \{1.0, 1.5\}$. Results are mean \pm std over 5 runs. Best results per column (within each T block) are in bold.

| Method | $T = 1.0$ | | | | | $T = 1.5$ | | | | |
|----------------|------------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ |
| PG | -0.219 \pm 0.782 | 0.343 \pm 0.195 | 0.833 \pm 0.405 | 0.194 \pm 0.006 | 0.054\pm0.007 | 0.546 \pm 0.731 | 0.259 \pm 0.084 | 0.690 \pm 0.226 | 0.217 \pm 0.006 | 0.320 \pm 0.013 |
| G-SMC | 2.571 \pm 1.979 | 0.359 \pm 0.163 | 0.973 \pm 0.445 | 0.147 \pm 0.030 | 0.201 \pm 0.121 | 2.645 \pm 1.065 | 0.639 \pm 0.185 | 1.681 \pm 0.527 | 0.116 \pm 0.133 | 0.325 \pm 0.182 |
| VCG | -0.046 \pm 0.093 | 0.026 \pm 0.009 | 0.076 \pm 0.031 | 0.045\pm0.007 | 0.127 \pm 0.018 | 0.083 \pm 0.121 | 0.032 \pm 0.015 | 0.091 \pm 0.048 | 0.034 \pm 0.004 | 0.558 \pm 0.031 |
| VCG-SMC | -0.044\pm0.029 | 0.019\pm0.005 | 0.048\pm0.013 | 0.053 \pm 0.012 | 0.053\pm0.022 | 0.077\pm0.010 | 0.015\pm0.007 | 0.035\pm0.011 | 0.011\pm0.008 | 0.078\pm0.010 |

Ablation on Annealing Factor γ . We further test the methods with varying annealing strengths ($\gamma \in \{1.5, 2.0, 2.5\}$) for both the DW-4 and LJ-13 systems. For the DW-4 system (Figs. 13 and 14 and Tab. 10), the visual and quantitative results confirm that VCG and VCG-SMC maintain high accuracy even as γ increases. For the more complex LJ-13 system (Fig. 15 and Tab. 11), the challenge is greater. While all methods struggle with the most challenging annealing tasks, VCG-SMC consistently provides the most physically plausible results, capturing the structural features (RDF) and energy distributions far more accurately than competing methods. This highlights its superior performance in complex, high-dimensional energy landscapes.

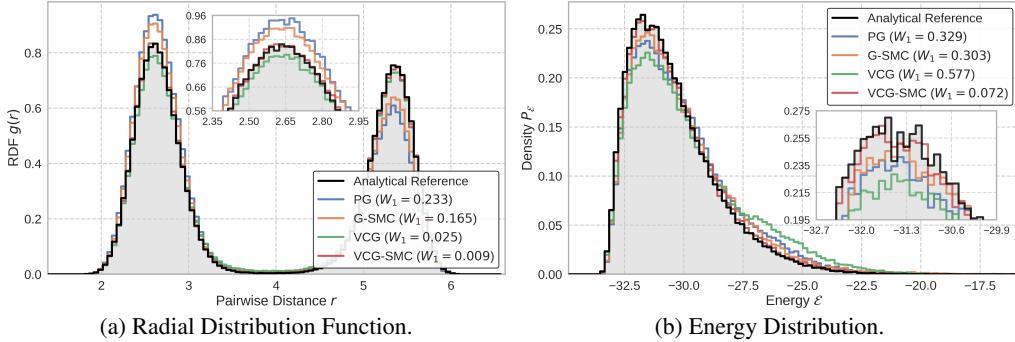


Figure 13: Comparison of generated distributions for the DW-4 annealing task ($\gamma = 2.0$): (a) Radial Distribution Function and (b) Energy Distribution.

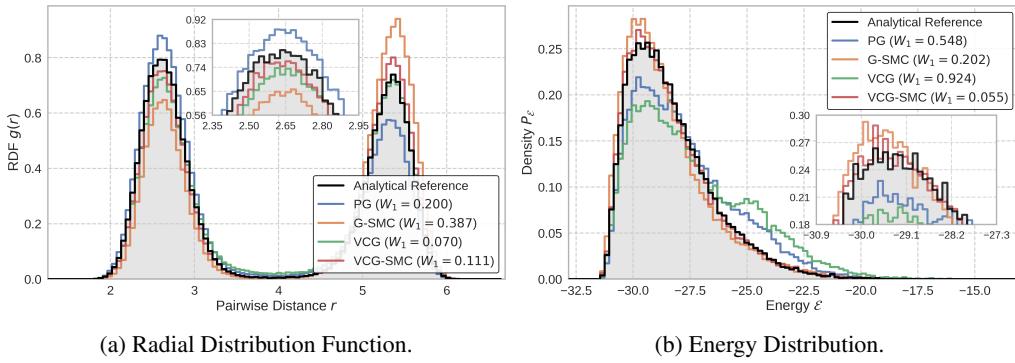


Figure 14: Comparison of generated distributions for the DW-4 annealing task ($\gamma = 2.5$): (a) Radial Distribution Function and (b) Energy Distribution.

Table 10: Performance ablation for the DW-4 annealing task ($T = 2.0$) with varying annealing factor γ . Results are mean \pm std over 5 runs. Best results per column (within each γ block) are in bold.

| Method | Annealing ($\gamma = 1.5$) | | | | | Annealing ($\gamma = 2.5$) | | | | |
|---------|--------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ |
| PG | -0.114 ± 0.339 | 0.198 ± 0.152 | 0.527 ± 0.396 | 0.132 ± 0.002 | 0.179 ± 0.023 | 0.159 ± 1.232 | 0.400 ± 0.168 | 1.088 ± 0.384 | 0.208 ± 0.008 | 0.551 ± 0.009 |
| G-SMC | -0.009 ± 0.041 | 0.053 ± 0.009 | 0.143 ± 0.031 | 0.038 ± 0.023 | 0.026 ± 0.011 | 0.038 ± 0.338 | 0.365 ± 0.058 | 0.012 ± 0.253 | 0.208 ± 0.146 | 0.190 ± 0.080 |
| VCG | -0.009 ± 0.018 | 0.010 ± 0.001 | 0.024 ± 0.003 | 0.041 ± 0.003 | 0.307 ± 0.020 | -0.065 ± 0.041 | 0.024 ± 0.003 | 0.061 ± 0.013 | 0.065 ± 0.008 | 0.931 ± 0.024 |
| VCG-SMC | -0.013 ± 0.017 | 0.009 ± 0.001 | 0.022 ± 0.002 | 0.037 ± 0.004 | 0.020 ± 0.006 | -0.060 ± 0.008 | 0.023 ± 0.001 | 0.056 ± 0.006 | 0.119 ± 0.008 | 0.059 ± 0.008 |

Table 11: Performance ablation for the LJ-13 annealing task ($T = 1.0$) with varying annealing factor γ . Results are mean \pm std over 5 runs. Best results per column (within each γ block) are in bold.

| Method | Annealing ($\gamma = 1.5$) | | | | | Annealing ($\gamma = 2.0$) | | | | |
|---------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ | ΔNLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ |
| PG | 1.224 ± 3.148 | 0.718 ± 0.021 | 0.806 ± 0.056 | 0.026 ± 0.001 | 1.855 ± 0.050 | 15.75 ± 10.56 | 0.622 ± 0.056 | 0.639 ± 0.150 | 0.053 ± 0.001 | 12.73 ± 0.193 |
| G-SMC | 3.289 ± 1.461 | 0.220 ± 0.112 | 0.190 ± 0.071 | 0.022 ± 0.007 | 3.517 ± 1.085 | 0.927 ± 2.384 | 0.416 ± 0.142 | 0.423 ± 0.171 | 0.017 ± 0.014 | 4.253 ± 3.053 |
| VCG | 1.642 ± 0.316 | 0.027 ± 0.007 | 0.029 ± 0.005 | 0.058 ± 0.001 | 8.848 ± 0.130 | 1.191 ± 1.499 | 0.124 ± 0.067 | 0.135 ± 0.076 | 0.088 ± 0.001 | 21.28 ± 0.127 |
| VCG-SMC | 2.221 ± 0.353 | 0.024 ± 0.004 | 0.024 ± 0.005 | 0.018 ± 0.002 | 3.006 ± 0.307 | 0.734 ± 0.490 | 0.091 ± 0.023 | 0.092 ± 0.021 | 0.007 ± 0.002 | 1.958 ± 0.610 |

Ablation on Reward Strength λ' . Finally, we perform an ablation on the reward strength ($\lambda' \in \{0.2, 0.5, 0.8\}$) for the reward-tilting task on both particle systems. The results, shown in Tab. 12 for DW-4 and Tab. 13 for LJ-13, are consistent with previous findings. As the reward strength increases, making the target distribution more distinct from the base distribution, the performance of the baseline methods deteriorates rapidly. In contrast, VCG-SMC maintains excellent performance,

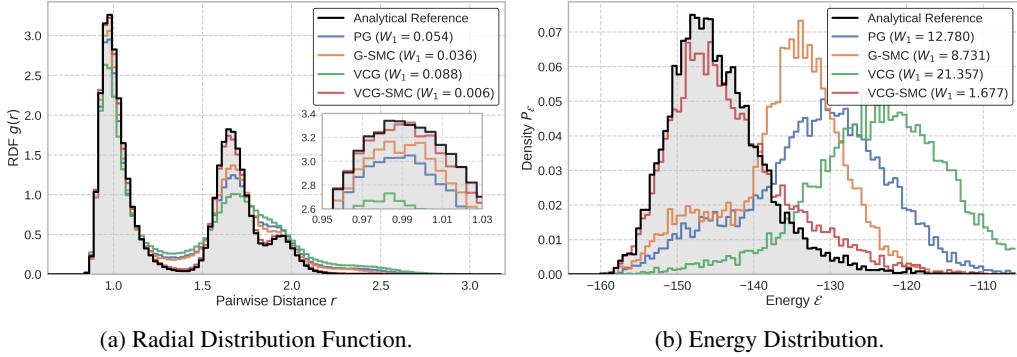


Figure 15: Comparison of generated distributions for the LJ-13 annealing task ($\gamma = 2.0$): (a) Radial Distribution Function and (b) Energy Distribution.

demonstrating its capability to accurately steer the particle distribution toward a sharply defined target region. This confirms the effectiveness and robustness of our drift control mechanism across a wide range of inference-time scaling challenges.

Table 12: Performance ablation for the DW-4 reward-tilting task ($T = 2.0$) with varying reward strength λ' . Results are mean \pm std over 5 runs. Best results per column (within each λ' block) are in bold.

| Method | Reward-Tilting ($\lambda' = 0.2$) | | | | | Reward-Tilting ($\lambda' = 0.8$) | | | | |
|---------|-------------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-------------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| | Δ NLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ | Δ NLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ |
| PG | -0.182 \pm 0.846 | 0.768 \pm 0.078 | 1.808 \pm 0.223 | 0.305 \pm 0.006 | 0.368 \pm 0.008 | 2.480 \pm 2.35 | 0.775 \pm 0.088 | 1.647 \pm 0.239 | 0.852 \pm 0.004 | 3.867 \pm 0.010 |
| G-SMC | 0.094\pm0.015 | 0.020 \pm 0.003 | 0.046 \pm 0.007 | 0.060 \pm 0.004 | 0.094 \pm 0.019 | 0.534 \pm 0.088 | 0.079 \pm 0.025 | 0.160 \pm 0.043 | 0.150 \pm 0.022 | 0.536 \pm 0.088 |
| VCG | 0.386 \pm 1.000 | 0.403 \pm 0.098 | 1.122 \pm 0.359 | 0.095 \pm 0.008 | 0.138 \pm 0.012 | 1.050 \pm 2.865 | 0.671 \pm 0.087 | 1.768 \pm 0.302 | 0.282 \pm 0.121 | 1.162 \pm 0.567 |
| VCG-SMC | 0.096\pm0.017 | 0.014\pm0.001 | 0.032\pm0.004 | 0.059\pm0.004 | 0.093\pm0.017 | 0.518\pm0.020 | 0.028\pm0.001 | 0.065\pm0.004 | 0.149\pm0.004 | 0.520\pm0.020 |

Table 13: Performance ablation for the LJ-13 reward-tilting task ($T = 1.0$) with varying reward strength λ' . Results are mean \pm std over 5 runs. Best results per column (within each λ' block) are in bold.

| Method | Reward-Tilting ($\lambda' = 0.2$) | | | | | Reward-Tilting ($\lambda' = 0.5$) | | | | |
|---------|-------------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-------------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| | Δ NLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ | Δ NLL | MMD | SWD | W_1^{RDF} | $W_1^{\mathcal{E}}$ |
| PG | 1.404 \pm 4.812 | 0.718 \pm 0.021 | 0.806 \pm 0.056 | 0.026 \pm 0.001 | 1.855 \pm 0.050 | 3.253 \pm 5.013 | 0.718 \pm 0.020 | 0.800 \pm 0.059 | 0.050 \pm 0.001 | 3.705 \pm 0.085 |
| G-SMC | 0.832 \pm 0.072 | 0.031 \pm 0.009 | 0.030 \pm 0.013 | 0.010\pm0.001 | 0.834\pm0.072 | 1.277 \pm 0.072 | 0.055 \pm 0.006 | 0.049 \pm 0.008 | 0.016\pm0.001 | 1.278 \pm 0.072 |
| VCG | 0.106\pm3.619 | 0.559 \pm 0.127 | 0.588 \pm 0.169 | 0.024 \pm 0.015 | 1.869 \pm 0.1046 | 7.470 \pm 10.167 | 0.669 \pm 0.132 | 0.675 \pm 0.167 | 0.024 \pm 0.015 | 1.869 \pm 0.1046 |
| VCG-SMC | 0.765 \pm 0.117 | 0.012\pm0.001 | 0.015\pm0.002 | 0.016 \pm 0.001 | 1.219 \pm 0.079 | 1.218\pm0.079 | 0.013\pm0.001 | 0.015\pm0.002 | 0.016\pm0.001 | 1.219\pm0.079 |

C.3 ADDITIONAL EXPERIMENTAL RESULTS OF PROTEIN-LIGAND CO-FOLDING

In Fig. 16, we show representative trajectories of the normalized Effective Sample Size (ESS) for three protein-ligand complexes (PDB IDs 7n7h, 7zu2, and 8d39). Even with a relatively small particle budget, our method (VCG-SMC) maintains substantially higher ESS than the baseline G-SMC throughout the middle and late stages of the trajectory, indicating that the controlled drift keeps a much larger fraction of particles effective. We expect this advantage to become even more pronounced as the number of particles increases.

Table 14 shows that both particle-based methods are about $3.2\times$ slower than the unsteered base sampler, reflecting the cost of propagating and reweighting a small ensemble of particles. Importantly, VCG-SMC incurs essentially no additional wall-clock overhead compared with the G-SMC baseline (31.78s vs 31.67s per complex), while delivering consistently higher ESS and improved physical validity on PoseBusters. Thus, the stability gains from DriftLite come at negligible extra cost beyond a standard SMC correction.

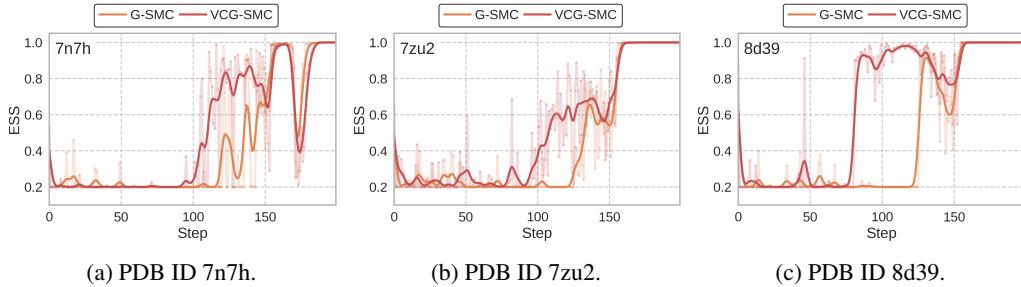


Figure 16: Evolution of ESS during inference on the protein–ligand co-folding task. The curves are smoothed using a Gaussian kernel over time, with the unsmoothed traces shown in the background for reference.

Table 14: Elapsed time per generated protein–ligand complex for different sampling methods, averaged over a random subset of 20 structures. The relative runtime is computed with respect to the G-SMC method.

| Method | Runtime (s) | Relative Runtime |
|----------------|-------------|------------------|
| Base | 10.05 | 0.32× |
| G-SMC | 31.67 | 1.00× |
| VCG-SMC | 31.78 | 1.00× |

C.4 ITERATIVE REFINEMENT

While a single pass of inference-time scaling is effective, its corrective power is finite; once the simulation reaches the terminal time, any residual mismatch between the particle and target distributions cannot be further addressed. The key insight behind our iterative approach is that each pass produces an improved sampling dynamic. The updated drift and potential:

$$\mathbf{v}_t^{\text{eff}}(\mathbf{x}) = \mathbf{v}_t(\mathbf{x}) + \mathbf{b}_t(\mathbf{x}), \quad g_t^{\text{eff}}(\mathbf{x}) = g_t(\mathbf{x}) + h_t(\mathbf{x}; \mathbf{b}_t),$$

encode richer information about the target distribution. Our iterative refinement procedure leverages this accumulated knowledge rather than discarding it. By iterating the procedure, each round builds upon the refined dynamics of the previous one, creating a virtuous cycle that progressively sharpens the sampling path.

This iterative process can also be viewed as a practical method for approaching the optimal control drift in Prop. 3.2. Since the control \mathbf{b}_t is computed at each step under a linear ansatz (Thms. 3.3 and 3.4), it provides an approximation of the true optimal control. This approximation becomes increasingly accurate as the underlying dynamics and particle distribution are improved in each round, allowing the linear model to operate on a better-conditioned problem.

Algorithm. The iterative refinement algorithm, detailed in Alg. 2, transforms the single-pass method (Alg. 1) into a multi-stage process of progressive improvement. The fundamental difference lies in the cumulative application of control. Whereas the base algorithm applies a calculated control drift just once, the iterative method repeats the entire simulation K times.

The central mechanism is the permanent absorption of the learned control into the system’s dynamics. After each full trajectory simulation (a “round”), the control terms are folded into an “effective” drift $\mathbf{v}_t^{\text{eff}}$ and potential g_t^{eff} . While particle positions are reset to the initial noise distribution at the start of a new round, these effective dynamics are preserved and carried forward. The refined dynamics from round j thus serve as the improved baseline for round $j + 1$. This process iteratively sharpens the sampling path, guiding particles more efficiently in subsequent rounds without any external training or global optimization.

Results. Our empirical results decisively validate the iterative refinement strategy on a Gaussian Mixture Model (GMM) target across both annealing and reward-tilting tasks.

Algorithm 2: Iterative Refinement for Inference-Time Scaling (cumulative drift/potential updates)

Input: Original drift path \mathbf{v}_t , original potential path g_t , time steps $\{t_k\}_{k=0}^M$, reward $r(\mathbf{x})$, schedule β_t , basis functions, number of refinement rounds K , number of particles N , ESS threshold τ .

```

1 Initialize effective drift  $\mathbf{v}_t^{\text{eff}}(\cdot) \leftarrow \mathbf{v}_t(\cdot)$ , and potential  $g_t^{\text{eff}}(\cdot) \leftarrow g_t(\cdot)$ ;
2 for  $j \leftarrow 1$  to  $K$  do
3   Initialize particles  $\mathbf{x}_{t_0}^{(i)} \sim \tilde{p}_0$  and  $w_{t_0}^{(i)} \leftarrow \frac{1}{N}$  for  $i = 1, \dots, N$ ;
4   for  $k \leftarrow 0$  to  $M - 1$  do
5     Form weighted estimates of  $\mathbf{A}_{t_k}$  and  $\mathbf{c}_{t_k}$  using  $\{(\mathbf{x}_{t_k}^{(i)}, w_{t_k}^{(i)})\}_{i \in [N]}$ ;
6     Solve  $\mathbf{A}_{t_k} \theta_{t_k} = \mathbf{c}_{t_k}$  to obtain the control drift  $\mathbf{b}_{t_k}(\cdot)$ ;
7      $\mathbf{v}_{t_k}^{\text{eff}}(\cdot) \leftarrow \mathbf{v}_{t_k}^{\text{eff}}(\cdot) + \mathbf{b}_{t_k}(\cdot)$ ,  $g_{t_k}^{\text{eff}}(\cdot) \leftarrow g_{t_k}^{\text{eff}}(\cdot) + h_{t_k}(\cdot; \mathbf{b}_{t_k})$ ;
8      $\log w_{t_{k+1}}^{(i)} \leftarrow \log w_{t_k}^{(i)} + g_{t_k}^{\text{eff}}(\mathbf{x}_{t_k}^{(i)})(t_{k+1} - t_k)$ ,  $\mathbf{w}_{t_{k+1}} \leftarrow \text{softmax}(\mathbf{w}_{t_{k+1}})$ ;
9      $\mathbf{x}_{t_{k+1}}^{(i)} \leftarrow \mathbf{x}_{t_k}^{(i)} + \mathbf{v}_{t_k}^{\text{eff}}(\mathbf{x}_{t_k}^{(i)})(t_{k+1} - t_k) + V_{t_k} \sqrt{t_{k+1} - t_k} \mathbf{z}^{(i)}$ , where  $\mathbf{z}^{(i)} \sim \mathcal{N}(0, \mathbf{I})$ ;
10    if  $\text{ESS}(\mathbf{w}_{t_{k+1}}) < \tau$  or periodically then
11      Resample  $\{\mathbf{x}_{t_{k+1}}^{(i)}\}_{i \in [N]}$  according to  $\{w_{t_{k+1}}^{(i)}\}_{i \in [N]}$ ;
12      Reset  $w_{t_{k+1}}^{(i)} \leftarrow \frac{1}{N}$  for all  $i$ ;

```

Output: Final samples $\{(\mathbf{x}_T^{(i)}, w_T^{(i)})\}_{i \in [N]}$ from the K -th refinement round.

The evolution of the sampler’s internal state, shown in Figs. 17 and 18, provides clear evidence of the algorithm’s success. With each successive refinement round (progressing from blue to red), the variance of the control estimates exhibits a striking, monotonic decrease. This variance reduction directly leads to a dramatically more stable ESS, which is maintained at near-optimal levels throughout the later stages of the simulation. This provides direct confirmation that the refined dynamics act as a substantially more efficient proposal distribution, a trend that holds robustly across both experimental setups.

The final sample quality metrics, reported in Tabs. 15 and 16, complete the picture. While the improvement in downstream metrics, such as MMD and SWD, is not always strictly monotonic with every iteration, the refinement process consistently uncovers solutions that are superior to those from the initial pass. The best-performing configurations (highlighted in bold) are frequently found in later iterations. This confirms that multiple refinement passes are invaluable for navigating the optimization landscape to find higher-quality final samples.

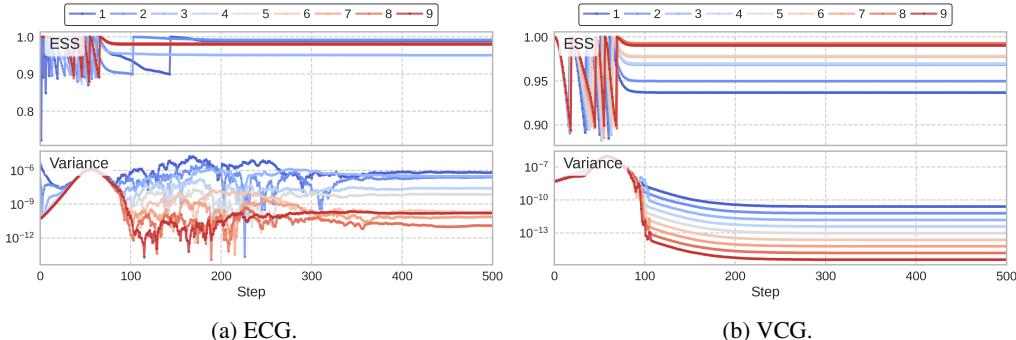


Figure 17: Evolution of ESS and potential variance across multiple refinement rounds for the GMM annealing task ($\gamma = 2.5$). Later rounds (red) show monotonically decreasing variance and more stable ESS.

Table 15: Iterative refinement performance on the GMM annealing task ($\gamma = 3.0$). Results are mean \pm std over 5 runs. Best result per column is in bold.

| Iter. | VCG-SMC | | | | | ECG-SMC | | | | |
|-------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ |
| 1 | 0.174 \pm 0.073 | 0.019 \pm 0.001 | 0.691 \pm 0.149 | 3.319 \pm 0.593 | 0.415 \pm 0.048 | 0.184 \pm 0.079 | 0.031 \pm 0.003 | 1.672 \pm 0.214 | 7.795 \pm 1.164 | 0.850 \pm 0.107 |
| 2 | 0.159 \pm 0.068 | 0.019 \pm 0.001 | 0.797 \pm 0.151 | 3.906 \pm 0.806 | 0.450 \pm 0.055 | 0.178 \pm 0.058 | 0.028 \pm 0.002 | 1.105 \pm 0.129 | 5.748 \pm 0.805 | 0.787 \pm 0.091 |
| 3 | 0.176 \pm 0.022 | 0.019 \pm 0.002 | 0.599\pm0.142 | 3.083\pm0.794 | 0.406 \pm 0.078 | 0.166 \pm 0.015 | 0.027 \pm 0.002 | 1.201 \pm 0.267 | 5.990 \pm 0.866 | 0.762 \pm 0.090 |
| 4 | 0.152\pm0.078 | 0.019 \pm 0.001 | 0.720 \pm 0.061 | 3.346 \pm 0.545 | 0.414 \pm 0.066 | 0.168 \pm 0.059 | 0.029 \pm 0.003 | 1.696 \pm 0.455 | 6.653 \pm 1.728 | 0.816 \pm 0.129 |
| 5 | 0.165 \pm 0.052 | 0.018\pm0.001 | 0.694 \pm 0.167 | 3.124 \pm 0.765 | 0.386\pm0.061 | 0.158\pm0.058 | 0.025\pm0.002 | 1.096\pm0.386 | 4.854\pm0.929 | 0.693\pm0.082 |
| 6 | 0.163 \pm 0.073 | 0.019 \pm 0.001 | 0.744 \pm 0.077 | 3.378 \pm 0.409 | 0.412 \pm 0.022 | 0.164 \pm 0.074 | 0.031 \pm 0.007 | 1.494 \pm 0.572 | 7.135 \pm 2.767 | 0.908 \pm 0.302 |
| 7 | 0.161 \pm 0.050 | 0.019 \pm 0.001 | 0.736 \pm 0.095 | 3.343 \pm 0.285 | 0.408 \pm 0.027 | 0.160 \pm 0.055 | 0.031 \pm 0.005 | 1.477 \pm 0.354 | 7.010 \pm 1.843 | 0.874 \pm 0.151 |
| 8 | 0.180 \pm 0.055 | 0.019 \pm 0.001 | 0.766 \pm 0.120 | 3.407 \pm 0.269 | 0.420 \pm 0.043 | 0.181 \pm 0.052 | 0.029 \pm 0.003 | 1.406 \pm 0.184 | 6.748 \pm 1.192 | 0.806 \pm 0.119 |
| 9 | 0.184 \pm 0.050 | 0.019 \pm 0.001 | 0.670 \pm 0.117 | 3.238 \pm 0.371 | 0.395 \pm 0.046 | 0.187 \pm 0.046 | 0.027 \pm 0.004 | 1.137 \pm 0.247 | 5.669 \pm 1.519 | 0.739 \pm 0.201 |

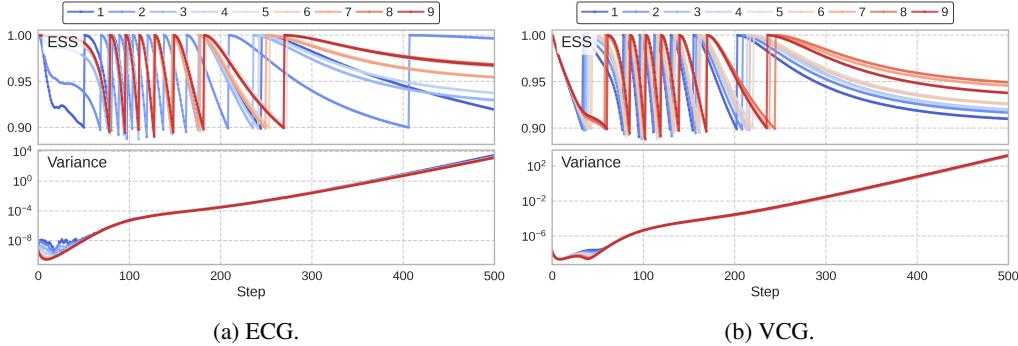


Figure 18: Evolution of ESS and potential variance across multiple refinement rounds for the GMM reward-tilting task ($\sigma = 100.0$). Later rounds (red) show monotonically decreasing variance.

Table 16: Iterative refinement performance on the GMM reward-tilting task ($\sigma = 100.0$). Results are mean \pm std over 5 runs. Best result per column is in bold.

| Iter. | VCG-SMC | | | | | ECG-SMC | | | | |
|-------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ | Δ NLL | MMD | SWD | Mean L_2 | Cov F $\times 10^3$ |
| 1 | 0.338 \pm 0.133 | 0.020 \pm 0.003 | 0.236 \pm 0.120 | 0.931 \pm 0.569 | 0.061 \pm 0.046 | 0.309 \pm 0.067 | 0.020 \pm 0.002 | 0.234 \pm 0.098 | 0.996 \pm 0.436 | 0.065 \pm 0.049 |
| 2 | 0.294 \pm 0.077 | 0.020 \pm 0.002 | 0.249 \pm 0.076 | 1.240 \pm 0.626 | 0.058 \pm 0.031 | 0.284 \pm 0.081 | 0.020 \pm 0.002 | 0.214\pm0.070 | 0.851\pm0.228 | 0.054 \pm 0.028 |
| 3 | 0.244\pm0.068 | 0.019 \pm 0.001 | 0.225 \pm 0.076 | 0.983 \pm 0.380 | 0.053 \pm 0.034 | 0.273\pm0.072 | 0.020 \pm 0.002 | 0.256 \pm 0.095 | 1.310 \pm 0.699 | 0.062 \pm 0.036 |
| 4 | 0.322 \pm 0.089 | 0.020 \pm 0.002 | 0.268 \pm 0.109 | 1.386 \pm 0.732 | 0.065 \pm 0.035 | 0.288 \pm 0.078 | 0.020 \pm 0.001 | 0.267 \pm 0.095 | 1.276 \pm 0.837 | 0.063 \pm 0.031 |
| 5 | 0.280 \pm 0.061 | 0.019\pm0.001 | 0.259 \pm 0.122 | 1.087 \pm 0.661 | 0.057 \pm 0.030 | 0.278 \pm 0.102 | 0.020 \pm 0.002 | 0.263 \pm 0.160 | 1.364 \pm 0.965 | 0.055 \pm 0.035 |
| 6 | 0.319 \pm 0.090 | 0.020 \pm 0.002 | 0.213 \pm 0.071 | 0.862 \pm 0.229 | 0.053 \pm 0.029 | 0.333 \pm 0.094 | 0.020 \pm 0.001 | 0.255 \pm 0.092 | 1.292 \pm 0.741 | 0.069 \pm 0.046 |
| 7 | 0.311 \pm 0.109 | 0.020 \pm 0.002 | 0.270 \pm 0.129 | 1.404 \pm 0.836 | 0.058 \pm 0.037 | 0.322 \pm 0.074 | 0.020 \pm 0.002 | 0.285 \pm 0.127 | 1.504 \pm 0.913 | 0.060 \pm 0.037 |
| 8 | 0.365 \pm 0.122 | 0.020 \pm 0.002 | 0.263 \pm 0.108 | 1.277 \pm 0.783 | 0.057 \pm 0.030 | 0.338 \pm 0.094 | 0.020 \pm 0.002 | 0.216 \pm 0.054 | 0.976 \pm 0.477 | 0.049 \pm 0.023 |
| 9 | 0.338 \pm 0.102 | 0.019 \pm 0.002 | 0.202\pm0.061 | 0.715\pm0.182 | 0.046\pm0.021 | 0.321 \pm 0.099 | 0.020\pm0.002 | 0.218 \pm 0.093 | 1.007 \pm 0.401 | 0.047\pm0.030 |