ADJOINT SAMPLING: HIGHLY SCALABLE DIFFUSION SAMPLERS VIA ADJOINT MATCHING

Aaron Havens[2,†,*], **Benjamin Kurt Miller**[1,*], **Bing Yan**[1,3,*], **Carles Domingo-Enrich**[4], **Anuroop Sriram**[1], **Brandon Wood**[1], **Daniel Levine**[1], **Bin Hu**[2], **Brandon Amos**[1], **Brian Karrer**[1], **Xiang Fu**[1,*], **Guan-Horng Liu**[1,*], **Ricky T. Q. Chen**[1,*]
[1]FAIR at Meta, [2]University of Illinois [3], New York University, [4]Microsoft Research New England
[*]Core Contributors, [†]Work done during internship at FAIR

ABSTRACT

We introduce *Adjoint Sampling*, a highly scalable and efficient algorithm for learning diffusion processes that sample from unnormalized densities, or energy functions. It is the first on-policy approach that allows significantly more gradient updates than the number of energy evaluations and model samples, allowing us to scale to much larger problem settings than previously explored by similar methods. Our framework is theoretically grounded in stochastic optimal control and shares the same theoretical guarantees as Adjoint Matching, being able to train without the need for corrective measures that push samples towards the target distribution. We show how to incorporate key symmetries, as well as periodic boundary conditions, for modeling molecules in both cartesian and torsional coordinates. We demonstrate the effectiveness of our approach through extensive experiments on classical energy functions, and further scale up to neural network-based energy models where we perform amortized conformer generation across many molecular systems. To encourage further research in developing highly scalable sampling methods, we plan to open source these challenging benchmarks, where successful methods can directly impact progress in computational chemistry.

1 INTRODUCTION

Sampling from complex, high-dimensional distributions underlies many important problems in computational science, with applications spanning molecular modeling, Bayesian inference , and generative modeling. In particular, we are interested in sampling from the target distribution with only access to its unnormalized energy function E, which defines the Boltzmann distribution

$$\mu(x) = \frac{\exp\left(-\frac{1}{\tau}E(x)\right)}{Z}, \tag{1}$$

where $Z = \int_{\mathbb{R}^d} \exp\left(-\frac{1}{\tau}E(x)\right) \mathrm{d}x < \infty$ is the unknown normalization constant. The Boltzmann distribution describes the equilibrium state of many physical systems, where $E(x)$ denotes the energy of a configuration x, and $\tau > 0$ is a temperature parameter. Efficiently sampling from such distributions remains challenging, especially for high-dimensional systems with intricate energy landscapes. Additionally many energy functions are extremely computationally expensive, e.g. requiring physics simulations.

Traditional approaches, such as Markov Chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC) using well-designed Markov Chains (Neal, 2001; Neal et al., 2011; Del Moral et al., 2006), provide asymptotically unbiased samples but often suffer from slow mixing and poor scalability to high-dimensional settings. This necessitates the design of better transition densities and smarter proposal distributions. Recent works try to address this by augmenting sampling with learned proposal distribution (Albergo et al., 2019; Arbel et al., 2021; Gabrié et al., 2022) via normalizing flows (Chen et al., 2018; Rezende & Mohamed, 2015).

It may seem natural to look towards the recent explosion of diffusion and flow-based generative models (Song & Ermon, 2019; Ho et al., 2020; Lipman et al., 2023; Albergo et al., 2023). However, a naïve adaptation of these data-driven generative modeling frameworks requires access to ground

truth data. This limitation is particularly significant in applications such as molecular simulations and physics-based inference, where direct access to samples from the target distribution is often unavailable. As a result, prior attempts often require an augmentation with sequential Monte Carlo or importance-sampling (Phillips et al., 2024; De Bortoli et al., 2024; Akhound-Sadegh et al., 2024), making these methods highly inefficient in terms of energy function evaluations. For a more complete overview of related methods please see Appendix A

The connection between sampling and diffusion processes was established by Tzen & Raginsky (2019b) through using classical results of stochastic optimal control (SOC) and Schrodinger-Bridge problems (Pavon, 1989; Dai Pra, 1991; Föllmer, 2005; Chen et al., 2016). Using this formulation, Zhang & Chen (2022) show that by directly parameterizing the drift of the controlled process, one could solve an SOC problem given unnormalized density for sampling tasks. This concept is further generalized by Berner et al. (2023) and Richter & Berner (2024); however, all of these method require computationally expensive simulation of the diffusion process per gradient update. Furthermore, they require at least one—sometimes many—energy evaluations per gradient update.

To overcome these challenges, we introduce *Adjoint Sampling*, a novel and extremely efficient variational inference framework based on stochastic control of diffusion processes, which we apply at much larger scale than previous methods. Our method is built on top of Adjoint Matching (Domingo-Enrich et al., 2024), a recent method developed for solving general stochastic control problems which we specialize and improve for efficiently learning to sample.

Our contributions are as follows:

- **Efficiency**: Our method is the first on-policy approach to allow far more gradient updates per model sample and energy evaluation. This is extremely important for scaling up to difficult amortized settings.
- **Theoretically Grounded**: Our method is grounded in stochastic control, following the recent work of Adjoint Matching (Domingo-Enrich et al., 2024). We improve upon Adjoint Matching by proposing a new objective that implicitly projects the model onto a set of optimal controls. We do not require corrective measures such as importance sampling or sequential Monte Carlo for our theoretical guarantees.
- **Structure**: Graph and Euclidean symmetries can be easily incorporated. We also adopt the method to handle periodic boundary conditions for modeling torsion angle representations of molecular systems.
- **New Benchmarks**: We introduce amortized molecule sampling benchmarks that challenge new methods to be applicable at scale. Being successful at these benchmarks directly drives progress in chemistry.

2 PRELIMINARIES

In this section, we briefly introduce an optimization problem over a class of stochastic processes that samples from μ at the optimal solution at the fixed time $t = 1$. This control perspective of sample generation was observed by Tzen & Raginsky (2019b), with similar formulations found in Zhang & Chen (2022); Berner et al. (2023). A more in-depth introduction to stochastic control for sampling can be found in Appendix B.1.

2.1 OPTIMIZING DIFFUSION PROCESSES FOR SAMPLING

Consider the stochastic differential equation (SDE)

$$\mathrm{d}X_t = \sigma(t)u(X_t, t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}B_t, \quad X_0 = 0, \tag{2}$$

where $\sigma : [0, 1] \to \mathbb{R}$ is a scalar noise function, $u : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ is a learnable *control* and $(B_t)_t$ is a d-dimensional Brownian motion. Under mild conditions on σ and u, this controlled process equation 2 uniquely defines time-marginals $p_t^u(X_t)$ for all $t \in \mathbb{R}$ and we seek a drift u such that $p_1^u(X_1) = \mu(X_1)$. We denote $\boldsymbol{X} = \{X_t : 0 \leq t \leq 1\}$ a sample trajectory and $p^u(\boldsymbol{X})$ and $p^{\text{base}}(\boldsymbol{X})$ the distributions over trajectories[1] generated by the controlled process equation 2 and base

[1]Formally, path measures of stochastic processes and their KL divergences are defined by Radon-Nikodyn derivative, *e.g.* $\frac{d\mathbb{P}^u}{d\mathbb{P}^{\text{base}}}(\boldsymbol{X}) = \exp(\int_0^1 \frac{1}{2}\|u_t\|^2 dt + \int_0^1 u_t \cdot dB_t)$ with $\boldsymbol{X} \sim \mathbb{P}^u$.

process $\mathrm{d}X_t = \sigma(t)\,\mathrm{d}B_t$ (i.e. setting $u \equiv 0$ in equation 2), respectively. The drift that transports the Dirac distribution to the target density μ at time $t = 1$ is not unique; however, we can choose a particular target path density that is a *Schrödinger bridge* (Pavon, 1989; Chen et al., 2016) between the Dirac and the target distribution given by

$$p^*(\boldsymbol{X}) = p^{\text{base}}(\boldsymbol{X}|X_1)\mu(X_1), \tag{3}$$

where $p^{\text{base}}(\boldsymbol{X}|X_1)$ is the posterior distribution of the base process conditioned on arriving at X_1. This Schrödinger bridge is, among other processes that coincides with μ at $t = 1$, the one that deviates the least from the base process in KL divergence. Notably, the path KL to p^* over trajectories with respect to the control u can be expressed analytically by factorizing $p^*(\boldsymbol{X}) = p^{\text{base}}(\boldsymbol{X})\frac{\mu(X_1)}{p_1^{\text{base}}(X_1)}$ and invoking Girsanov's Theorem (Protter & Protter, 2005),

$$\mathcal{L}_{\text{SOC}}(u) = D_{\text{KL}}(p^u(\boldsymbol{X}) \parallel p^*(\boldsymbol{X})) = D_{\text{KL}}(p^u(\boldsymbol{X}) \parallel p^{\text{base}}(\boldsymbol{X})) + \mathbb{E}_{p^u} \log\left(\frac{p_1^{\text{base}}(X_1)}{\mu(X_1)}\right) \tag{4}$$

$$= \mathbb{E}_{p^u}\left[\int_0^1 \tfrac{1}{2}\|u(X_t,t)\|^2\,\mathrm{d}t + \log\left(\frac{p_1^{\text{base}}(X_1)}{\mu(X_1)}\right)\right]. \tag{5}$$

To this end, we propose to minimize the objective equation 5, which, intuitively, minimizes the control energy required to transport to μ. In fact, this criteria corresponds exactly with a minimum-energy stochastic optimal control (SOC) problem with terminal cost function given by $\log\frac{p_1^{\text{base}}(x)}{\mu(x)}$. This particular SOC problem is heavily studied in literature and has a unique optimal solution u^* satisfying (Kappen, 2005; Tzen & Raginsky, 2019a) (See Appendix B.1 for more details)

$$D_{\text{KL}}(p^{u^*}(\boldsymbol{X}) \parallel p^*(\boldsymbol{X})) = -\log \mathbb{E}_{p^{\text{base}}}\left[\frac{\mu(X_1)}{p_1^{\text{base}}(X_1)}\right] = 0. \tag{6}$$

Therefore p^* can indeed be achieved by u^* and our problem is then reduced to optimizing the SOC objective equation 5 as efficiently as possible. This is precisely what our proposed method *Adjoint Sampling* addresses.

3 Adjoint Sampling

Now that we have equated the sampling problem as the solution of an SOC problem, we will now propose a highly efficient method to solve SOC problems of the form

$$\min_u \mathbb{E}_{\boldsymbol{X}\sim p^u}\left[\int_0^1 \tfrac{1}{2}\|u(X_t,t)\|^2\,\mathrm{d}t + g(X_1)\right] \tag{7a}$$

$$\text{s.t. } \mathrm{d}X_t = \sigma(t)u(X_t,t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}B_t, \quad X_0 = 0 \tag{7b}$$

which we will use to learn a stochastic process that samples from a target distribution given only an unnormalized energy function equation 1 with $g(x) = \log p_1^{\text{base}}(x) + \frac{1}{\tau}E(x)$, which is equation 5 without the normalization constant as it does not affect the optimal solution. Naïvely, one could simulate the SDE equation 7b, and then differentiate through the objective equation 7a, often referred to as the adjoint method (Bryson & Ho, 1969), and is the method of choice for prior works (Chen et al., 2018; Zhang & Chen, 2022). However, this is very slow as it requires two simulations per iteration, one for the state X_t and another one for backpropagating through the SDE.

3.1 Adjoint Matching

Adjoint Matching (Domingo-Enrich et al., 2024) is an algorithm designed to solve a more general family of stochastic optimal control problems, being able to make use of base processes with arbitrary drift $\mathrm{d}X_t = b(X_t,t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}B_t$. Unlike standard adjoint methods (Bryson & Ho, 1969) that directly differentiate through the objective equation 7a, Adjoint Matching turns the problem into a regression formulation. We discuss Adjoint Matching in detail in Appendix B. For our optimal control objective, the Adjoint Matching loss is

$$\mathcal{L}_{\text{AM}}(u) := \mathbb{E}_{p^{\bar{u}}}\left[\frac{1}{2}\int_0^1 \left\|u(X_t,t) + \sigma(t)^\top \tilde{a}(t;\boldsymbol{X})\right\|^2 \mathrm{d}t\right], \qquad \boldsymbol{X} \sim p^{\bar{u}}, \quad \bar{u} = \texttt{stopgrad}(u), \tag{8}$$

$$\text{s.t.} \quad \frac{\mathrm{d}}{\mathrm{d}t}\tilde{a}(t;\boldsymbol{X}) = -\tilde{a}(t;\boldsymbol{X})^\top \nabla b(X_t,t), \quad \tilde{a}(1;\boldsymbol{X}) = \nabla g(X_1). \tag{9}$$

where $\tilde{a}$ is referred to as the *lean adjoint* state, and $\bar{u} = \texttt{stopgrad}(u)$ denotes a stop gradient operation of u, *i.e.*, although $\boldsymbol{X} \sim p^{\bar{u}}$ is sampled according to the controlled process.

Our first key observation is that the original Adjoint Matching algorithm can be significantly simplified in the case of our sampling formulation equation 7. In particular if we set set $b \equiv 0$, the lean adjoint state equation $\frac{\mathrm{d}}{\mathrm{d}t}\tilde{a}(t; \boldsymbol{X}) = 0, \quad \tilde{a}(1, \boldsymbol{X}) = \nabla g(X_1)$ admits the unique analytical solution $\tilde{a}(t; \boldsymbol{X}) = \nabla g(X_1)$ for all $t \in [0, 1]$. This leads to a drastically simplified regression loss requiring no additional simulation of the lean adjoint state.

$$\mathcal{L}_{\text{AM}}(u) = \mathbb{E}_{\boldsymbol{X} \sim p^{\bar{u}}} \left[\int_0^1 \frac{1}{2} \| u(X_t, t) + \sigma(t) \nabla g(X_1) \|^2 \, \mathrm{d}t \right] \tag{10}$$

Roughly speaking, Adjoint Matching offers a simple interpretation: for each intermediate state X_t, simply regress the control onto the negative gradient of the terminal cost $-\nabla g(X_1)$ for all possible X_1 that can be arrived at from X_t. Since this is a moving target, this will, over the course of optimization, slowly shift the process towards regions with small terminal cost g.

3.2 Reciprocal Adjoint Matching

Adjoint Matching still requires two computationally expensive operations at every iteration: (i) simulation of the controlled process and (ii) evaluation of the terminal cost. Simulation of an SDE is a slow iterative procedure, and the terminal cost can also be costly. For instance, in our formulation, the terminal cost involves the energy model E which in many real-world scenarios is extremely computationally expensive. Ideally, neither of these two operations should be done frequently. In this section, we propose modifications to the Adjoint Matching algorithm to make it highly scalable, removing both requirements.

Firstly, notice that the training loss only depends on sampling $(X_t, X_1) \sim p^u_{t,1}$, not the full trajectory. Hence equation 10 is equivalent to sampling pairs (X_t, X_1) from the joint distribution defined by the stochastic process p^u:

$$\mathcal{L}_{\text{AM}}(u) = \int_0^1 \mathbb{E}_{(X_t, X_1) \sim p^{\bar{u}}_{t,1}} \left[\frac{1}{2} \| u(X_t, t) + \sigma(t) \nabla g(X_1) \|^2 \right] \mathrm{d}t. \tag{11}$$

This does not yet yield any efficiency gains as the only way to sample $(X_t, X_1) \sim p^u$ is through simulating the controlled process. Our key insight is to make use of the knowledge that at the optimal solution u^* we have that $p^{u^*}_{t,1}(X_t, X_1) = p^{\text{base}}_{t|1}(X_t|X_1)\mu(X_1)$ due to equation 3. Although the current control u does not necessarily satisfy this property, we can project the path measure—known as a Reciprocal projection which we make formal in Section 3.3—onto the Schrödinger bridge that generates $p^{\bar{u}}_1$ given by $p^{\text{base}}_{t|1}(X_t|X_1)p^{\bar{u}}(X_1)$. Based on this, we propose the *Reciprocal Adjoint Matching* (RAM) objective:

$$\mathcal{L}_{\text{RAM}}(u) = \int_0^1 \lambda(t) \mathbb{E}_{X_t \sim p^{\text{base}}_{t|1}, X_1 \sim p^{\bar{u}}_1} \left[\frac{1}{2} \| u(X_t, t) + \sigma(t) \nabla g(X_1) \|^2 \right] \mathrm{d}t. \tag{12}$$

That is, we sample X_1 according to the controlled process, then sample X_t conditioned on X_1 using the posterior distribution defined by the base process. Since the base process is an SDE with zero drift (equivalent to equation 2 with $u \equiv 0$), conditionals $p^{\text{base}}_{t|1}(X_t|X_1)$ are known in closed form and can be easily sampled. This also has the effect of de-correlating samples across time because the X_t samples for different values of t are now conditionally independent given X_1, whereas in equation 11 the X_t are sampled from the same trajectory. We additionally apply a time scaling of $\lambda(t) = \frac{1}{\sigma(t)^2}$ which does not affect the optimal solution but improves numerical stability.

To design a *highly efficient* algorithm, we further decouple $p(X_1)$ from the regression problem of learning u, delaying updates to $p(X_1)$ and performing multiple iterations to train u. Practically, this leads to the following alternating algorithm which we refer to as *Adjoint Sampling*:

1. Using the current control u_i, construct a buffer $\mathcal{B} = (X_1^{(i)}, \nabla g^{(i)})$ with samples $\{X_1^{(i)}\} \overset{iid}{\sim} p^u_1(X_1)$ and $\nabla g^{(i)} = \nabla g(X_1^{(i)})$.
2. Obtain updated control u_{i+1} by optimizing equation 12 using samples $\{X_1^{(i)}, \nabla g^{(i)}\} \sim \mathcal{B}$.

Algorithm 1 Adjoint Sampling

1: **Input:** Terminal Cost: $g = \log p^{\text{base}} + \frac{1}{\tau}E$, base process: p_t^{base} given by $dX_t = \sigma(t)dB_t$, outer-loop batch size: n, inner-loop batch size m, SDE drift network: u_θ, Replay buffer $\mathcal{B}$.
2: $\mathcal{B} \leftarrow \varnothing$
3: **while** Outer-Loop **do**
4: # Euler-Maruyama with **no gradient**
5: $\{X_1^{(i)}\}_{i=1}^n \sim p_1^{\bar{u}}, \qquad \bar{u} = \texttt{stopgrad}(u_\theta)$
6: # gradient of energy is evaluated **once** per sample
7: $\nabla g^{(i)} \leftarrow \nabla g(X_1^{(i)})$
8: $\mathcal{B} \leftarrow \mathcal{B} \cup \{(X_1^{(i)}, \nabla g^{(i)})\}_{i=1}^n$
9: **while** Inner-Loop **do**
10: $\{(X_1^{(j)}, \nabla g^{(j)})\}_{j=1}^m \sim \mathcal{U}(\mathcal{B})$
11: $t^{(j)} \sim \mathcal{U}([0,1]), \quad X_t^{(j)} \sim p_{t^{(j)}|1}^{base}(x|X_1^{(j)})$
12: $\mathcal{L}_{\text{RAM}}^{(j)} \leftarrow \frac{\lambda(t)}{2}\left\| u_\theta(X_t^{(j)}, t^{(j)}) + \sigma(t)\nabla g^{(j)} \right\|^2$
13: $\theta \leftarrow \texttt{optimizer_step}(\theta, \nabla_\theta \frac{1}{m}\sum_j \mathcal{L}_{\text{RAM}}^{(j)})$
14: **Output:** SDE sampler drift u_θ.

We theoretically justify this alternating scheme in the next section as implicitly performing a projection onto a more optimal control that aids in decreasing the SOC objective. Note that optimizing equation 12 in Step 2 above is extremely cheap due to the analytical form of the forward base process. Both p_1^{base} and the posterior distribution $p_{t|1}^{\text{base}}$ can be designed as closed-form Gaussian distributions which we detail in Appendix C. This means that *many gradient updates can be carried out without needing to simulate the controlled process or evaluation of the energy model*. We provide detailed pseudo-code in Algorithm 1.

Geometric Extensions Symmetries are essential to efficiently sample from energy-based models defining physical systems. We reduce dimensionality and improve data efficiency by enforcing symmetries in our model of the controlled process using Equivariant Graph Neural Networks (EGNNs) (Satorras et al., 2021) or Tensor Field Networks (Thomas et al., 2018), implemented in `e3nn` (Geiger et al., 2022; Jing et al., 2022). See Appendix D for more details.

3.3 Adjoint Sampling Theory

In this section, we provide theoretical justification to our proposed Algorithm 1. Firstly, let us define a projection $\mathcal{P}$ of a control u onto the SOC solution where p_1^u is the target distribution:

$$\mathcal{P}(u) = \arg\min_v D_{\text{KL}}(p^v(\boldsymbol{X}) \parallel p^{\text{base}}(\boldsymbol{X}|X_1)p_1^u(X_1)). \tag{13}$$

This is a projection step onto the Schrödinger bridge that shares the same terminal distribution $p_1^u(X_1)$. In connection to existing literature, this can be understood as a combination of the Reciprocal and Markovian projections of Shi et al. (2023). In terms of minimizing the objective, the resulting control process is consistently as effective as, if not superior to, the original control:

Proposition 3.1. *After projection equation 13, Reciprocal Adjoint Matching is equivalent to Adjoint Matching ($\mathcal{L}_{RAM}(\mathcal{P}(u)) = \mathcal{L}_{AM}(\mathcal{P}(u))$) and this projection improves upon on the SOC objective,*

$$J(u) \geq J(\mathcal{P}(u)), \qquad \textit{where } J(u) := \mathbb{E}_{\boldsymbol{X}\sim p^u}\left[\int_0^1 \tfrac{1}{2}\|u(X_t,t)\|^2 \,\mathrm{d}t + g(X_1)\right]. \tag{14}$$

We defer the proof to Appendix E.1. The result of Proposition 3.1 hints that if we explicitly perform this projection equation 13 at every iteration, then we are effectively performing Adjoint Matching iterations with a more optimal control. However, performing the projection requires additional computation steps, *e.g.*, using algorithm such as Bridge Matching (Shi et al., 2023), which we want to avoid. Fortunately, it turns out that using just the RAM loss is sufficient and will *implicitly* include this exact projection.

Our main theoretical result is the observation that the alternating scheme, where the distribution of X_1 samples are fixed (step 1) and we update the control by fully converging the RAM loss (step 2),

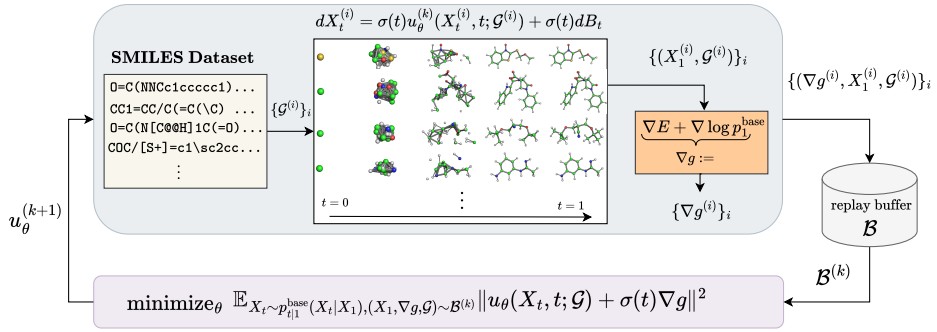


Figure 1: The iterations of Adjoint Sampling for amortized molecular conformer generation. SMILES strings condition the stochastic differential equation (SDE) on a specific molecular graph $\mathcal{G}^{(i)}$, where the final states and energy gradients and conditioning $(\nabla g^{(i)}, X_1^{(i)}, \mathcal{G}^{(i)})$ are stored into a replay buffer. The model is trained by minimizing the RAM loss using the sample buffer, progressively transforming the samples into realistic molecular conformations.

is *implicitly* performing the projection equation 13 while performing Adjoint Matching on the SOC objective equation 5. This correspondence is stated informally in the following Theorem.

Theorem 3.2 (Theoretical guarantees of Adjoint Sampling (*informal*))**.** *Starting with any control u_i, performing steps 1 and 2 to obtain u_{i+1} equivalently satisfies*

$$u_{i+1} = \mathcal{P}(u_i) - \frac{\delta \mathcal{L}_{AM}}{\delta u}(\mathcal{P}(u_i)), \tag{15}$$

where $\frac{\delta \mathcal{L}_{AM}}{\delta u}$ denotes the functional derivative with respect to the control u. Moreover, the fixed point where $u = \mathcal{P}(u)$ and $u = u - \frac{\delta \mathcal{L}_{AM}}{\delta u}(u)$ is the optimal control u^* to equation 7.

A more precise statement and proof can be found in Appendix E.1. This result provides the theoretical justification for our proposed algorithm (Algorithm 1). In practice, we differ slightly and use a replay buffer that contains samples from multiple prior steps, and we do not perform step 2 until convergence. We find this helps smoothen the optimization and improve computational efficiency.

4 Experiments

We now evaluate Adjoint Sampling on several energy functions over multi-particle systems. We propose a challenging *molecular conformer generation benchmark*, where one trains amortized models to sample conformers for a large dataset of organic molecules. For this task, we used the eSEN energy model (Fu et al., 2025). For more comparison to previous works (Zhang & Chen, 2022; Akhound-Sadegh et al., 2024) on synthetic energy benchmarks see Appendix J.1.

Table 1: Recall and precision metrics for large scale amortized conformer generation. Coverage values are for thresholds of 1.25Å. Standard deviations are computed across molecules in the test set. Results split by number of rotatable bonds can be found in Figure 11.

		SPICE				GEOM-DRUGS			
		Recall		Precision		Recall		Precision	
	Method	Cov. ↑	AMR ↓	Cov. ↑	AMR ↓	Cov. ↑	AMR ↓	Cov. ↑	AMR ↓
	RDKit ETKDG	71.35±34.07	1.03±0.52	68.23±37.80	1.13±0.64	66.37±34.85	1.09±0.73	73.55±37.49	1.03±0.80
	Torsional AdjSampling	85.66±23.90	0.85±0.29	70.55±34.51	1.05±0.53	75.21±29.93	0.95±0.56	72.40±34.57	1.02±0.68
	Cartesian AdjSampling	80.28±27.68	0.96±0.28	47.97±32.24	1.25±0.37	61.36±38.74	1.17±0.60	41.19±35.93	1.46±0.67
	Cartesian AdjSampling (+pretrain)	88.58±19.96	0.85±0.25	65.12±29.90	1.13±0.34	73.67±35.26	0.97±0.57	52.17±36.30	1.25±0.62
w/ relaxation	RDKit ETKDG	81.30±27.89	0.80±0.45	74.53±35.10	0.98±0.64	75.50±28.77	0.85±0.59	79.93±30.76	0.81±0.67
w/ relaxation	Torsional AdjSampling	87.42±22.07	0.73±0.33	74.05±33.47	0.94±0.59	79.04±27.18	0.82±0.49	76.21±32.08	0.87±0.64
w/ relaxation	Cartesian AdjSampling	91.49±18.27	0.71±0.29	60.00±29.97	1.02±0.38	80.59±29.46	0.81±0.49	51.95±32.44	1.25±0.68
w/ relaxation	Cartesian AdjSampling (+pretrain)	96.36± 8.92	0.60±0.25	76.46±26.22	0.92±0.35	87.32±23.89	0.70±0.45	68.02±31.86	1.01±0.55

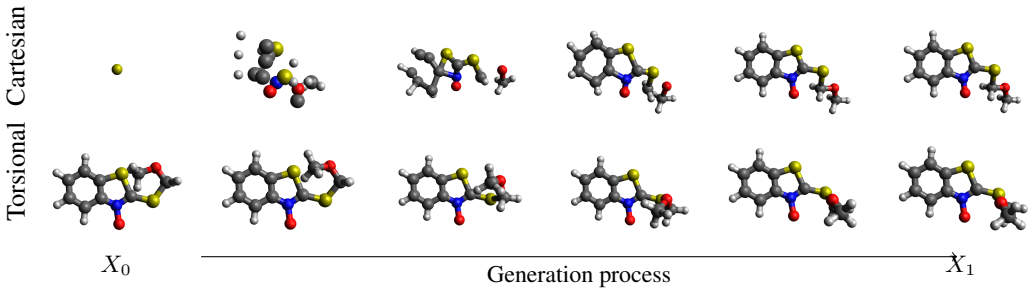


Figure 2: The figure depicts two sampled trajectories from trained Adjoint Sampling models that use either the cartesian or the torsional representations. They target conformations of the held-out SMILES string `COCSc1sc2ccccc2[n+]1[O-]`. The left frame X_0 comes from the initial Dirac distribution and the right frame X_1 is a sampled conformer.

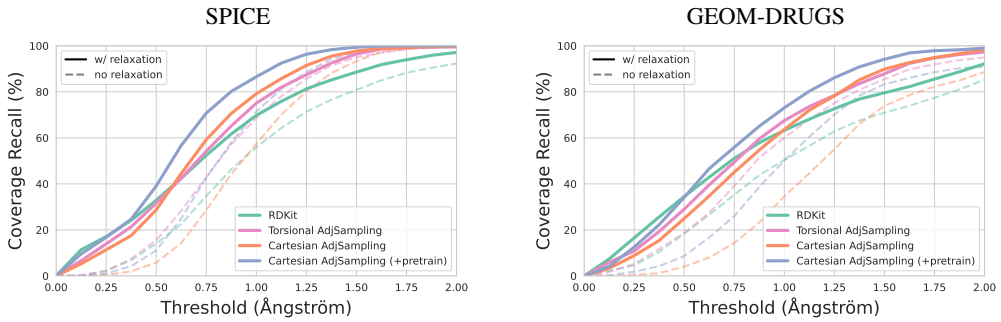


Figure 3: Recall coverage versus RMSD threshold for Adjoint Sampling variants and RDKit. We show performance both with and without relaxation.

4.1 Sampling Conformers from an Energy Function

We provide a new benchmark to encourage research into highly scalable sampling methods capable of finding *conformers* from a molecular energy model. These are local minima on the molecule's potential energy surface, *i.e.* a fixed molecular topology has a set of conformers determined by locally stable configurations of rotations around its bonds. Depending on the flexibility, a molecule can have one or many different conformers.

Sampling conformations based on quantum mechanical energy functions such as density functional theory (DFT) is an important but challenging task. Due to the high cost of DFT calculations, we use an energy model that was pretrained on the SPICE-MACE-OFF dataset (Eastman et al., 2023; Kovács et al., 2023) called *eSEN* (Fu et al., 2025). It has shown very high accuracy in predicting the DFT energy. eSEN is a transferable message passing neural network model that can predict the energy of a given molecular conformation for molecules containing {H, C, N, O, F, P, S, Cl, Br, I}. It featurizes a molecule with a sparse graph representation, encoding the chemical element in the node features and the Cartesian displacements in the edge features. The SPICE-MACE-OFF dataset comprehensively samples the energy surface in high and low energy regions, enabling eSEN to be generally applicable to many conformations of small molecules.

Using the Adjoint Sampling framework, we can sample atomic coordinates of a specific molecule by conditioning on its connectivity graph. These atomic coordinates are proposed conformations and graph conditioning enables amortization, *i.e.* learning a sampler for molecules consisting of arbitrary combinations of organic elements and bonds. We design two approaches:

1. **Cartesian Adjoint Sampling** (Appendix H.1): We sample all 3D Cartesian coordinates for each atom, conditioned on elements (nodes) and bond type (edges).

2. **Torsional Adjoint Sampling** (Appendix H.2): We sample *only* the torsion angles in the molecule, conditioned on elements, bond lengths, and bond angles (3-tuples).

In addition, we evaluate a *pretraining* step on RDKit samples using Bridge Matching (Shi et al., 2023) which aids in initial exploration of low energy regions (see Appendix F for more details). In our experiments, we scale the eSEN energy by the Boltzmann constant and a low temperature. In Cartesian Adjoint Sampling, we introduce additional energy terms to regularize bond lengths, ensuring we are targeting the correct molecule according to its connectivity. The terms do not appear in Torsional Adjoint Sampling; they are unnecessary since the torsional SDE cannot break bonds by design.

Datasets We consider two datasets of molecular structures: SPICE (Eastman et al., 2023) and GEOM-DRUGS (Axelrod & Gomez-Bombarelli, 2022). SPICE enumerates over 23,000 diverse drug-like molecules and their properties; it is designed to be utilized for machine learning. All of our models are trained on SPICE; however, we utilize *merely* the SMILES strings to extract the molecular topology. This is possible because Adjoint Sampling does not require any data during training, *i.e.*, we do *not* use the atomic configurations! We evaluate generalization capabilities by computing precision and recall on a subset of GEOM-DRUGS, see Appendix I.2. This is one the largest datasets and most relevant to pharmaceutical drug design.

We split the SPICE molecules into a train set and a 100 molecule test set, allowing us to validate our sampler's ability to extrapolate to unseen molecules. We produce reference conformers for evaluation purposes using RDKit (Landrum, 2013); CREST (Pracht et al., 2020), which is considered the gold standard in molecular chemistry; and ORCA (Neese, 2012). Because this procedure is computational expensive, we do this only for the SPICE test set. GEOM-DRUGS provides conformers for us. See Appendix I for details about the datasets.

Baselines We compare our approach to RDKit ETKDG (Riniker & Landrum, 2015), a chemistry-based conformer generation method. RDKit combines a rules-based distance matrix method, an iterative refinement algorithm, a 3D embedding, and experimental torsional preferences to generate conformers, making it a suitable domain-specific baseline.

Evaluation Our assessment focuses on reference conformer coverage *recall*, which measures the percentage of reference CREST conformers recovered by the generated samples, and coverage *precision*, which evaluates how often generated conformers closely match a low-energy reference structure. In particular, recall is a very important metric for molecular design because exploration is typically challenging and good mechanisms exist to filter candidates. The number of reference conformers varies significantly for molecules under consideration, ranging from tens to thousands. We also report Absolute Mean RMSD (AMR) as an error metric, with lower values indicating better structural fidelity. See Appendix H.3 for more details. Results are presented both with and without relaxation, where post-generation optimization helps refine molecular conformers in Table 1. While samples without relaxation are a test of the sampling algorithm's performance, testing with relaxation is ultimately what brings the most value to computational chemistry being a standard refinement process (see Appendix H).

Results As seen in Table 1, across both SPICE and GEOM-DRUGS, Adjoint Sampling (both Cartesian and Torsional) outperform the RDKit baseline in Recall. On generalizing to the GEOM-DRUGS dataset, we show worse precision than RDKit but significantly improved recall. As depicted in Figure 3, Cartesian Adjoint Sampling outperforms RDKit on almost all threshold values, with and without relaxation. While relaxation improves coverage across all methods, it particularly benefits the performance of Adjoint Sampling, making the gap between RDKit-ETKDG even larger. We believe this comes from the fact that Adjoint Sampling naturally explores more of the configuration space by being initialized as a noisy stochastic process.

Without relaxation or pretraining, the Torsional AdjSampling variant performs significantly better than Cartesian Variant in precision and recall on both datasets. However, after pretraining Cartesian AdjSampling surpasses the Torsional variant and even further exceeds in performance after relaxation, (being pretrained or not). This may be due to the Torsional representation evolving on a constrained space as seen in Figure 2, where the Cartesian representation evolves with the energy regularizer (see Appendix H.1) but is unconstrained. This state representation will affect how the controlled process explores the configuration space, suggesting that the unconstrained Cartesian representation samples are more diverse.

Finally, in Figure 11 we also compare across different number of rotatable bonds—the number of conformers increases drastically with the number of rotatable bonds (see fig. 7)—where we see that the gap between our model and RDKit increases as the difficulty increases.

REFERENCES

Tara Akhound-Sadegh, Jarrid Rector-Brooks, 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. In *Forty-first International Conference on Machine Learning*, 2024.

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

Michael S Albergo, Gurtej Kanwar, and Phiala E Shanahan. Flow-based generative models for markov chain monte carlo in lattice field theory. *Physical Review D*, 100(3):034515, 2019.

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

Michael Arbel, Alex Matthews, and Arnaud Doucet. Annealed flow transport monte carlo. In *International Conference on Machine Learning*, pp. 318–330. PMLR, 2021.

Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in neural information processing systems*, 35:11423–11436, 2022.

Richard Bellman. *Dynamic programming*. Princeton Landmarks in Mathematics. Princeton University Press, Princeton, NJ, 2010., 1957.

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.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *arXiv preprint arXiv:2211.01364*, 2023.

A. E. Bryson and Y. C. Ho. *Applied Optimal Control*. Blaisdell, 1969.

Junhua Chen, Lorenz Richter, Julius Berner, Denis Blessing, Gerhard Neumann, and Anima Anandkumar. Sequential controlled langevin diffusions. *arXiv preprint arXiv:2412.07081*, 2024.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. On the relation between optimal transport and schrödinger bridges: A stochastic control viewpoint. *Journal of Optimization Theory and Applications*, 169:671–691, 2016.

Paolo Dai Pra. A stochastic control approach to reciprocal diffusion processes. *Applied mathematics and Optimization*, 23(1):313–329, 1991.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in Neural Information Processing Systems*, 35:2406–2422, 2022.

Valentin De Bortoli, Michael Hutchinson, Peter Wirnsberger, and Arnaud Doucet. Target score matching. *arXiv preprint arXiv:2402.08667*, 2024.

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.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control, 2024. URL `https://arxiv.org/abs/2409.08861`.

Peter Eastman, Pavan Kumar Behara, David L Dotson, Raimondas Galvelis, John E Herr, Josh T Horton, Yuezhi Mao, John D Chodera, Benjamin P Pritchard, Yuanqing Wang, et al. Spice, a dataset of drug-like molecules and peptides for training machine learning potentials. *Scientific Data*, 10(1):11, 2023.

W.H. Fleming and R.W. Rishel. *Deterministic and Stochastic Optimal Control*. Stochastic Modelling and Applied Probability. Springer New York, 2012.

Hans Föllmer. An entropy approach to the time reversal of diffusion processes. In *Stochastic Differential Systems Filtering and Control: Proceedings of the IFIP-WG 7/1 Working Conference Marseille-Luminy, France, March 12–17, 1984*, pp. 156–163. Springer, 2005.

Marvin Friede, Christian Hölzer, Sebastian Ehlert, and Stefan Grimme. dxtb—an efficient and fully differentiable framework for extended tight-binding. *The Journal of Chemical Physics*, 161(6), 2024.

Xiang Fu, Brandon M Wood, Luis Barroso-Luque, Daniel S Levine, Meng Gao, Misko Dzamba, and C Lawrence Zitnick. Learning smooth and expressive interatomic potentials for physical property prediction. *arXiv preprint arXiv:2502.12147*, 2025.

Marylou Gabrié , Grant M. Rotskoff, and Eric Vanden-Eijnden. Adaptive monte carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10), mar 2022.

Octavian Ganea, Lagnajit Pattanaik, Connor Coley, Regina Barzilay, Klavs Jensen, William Green, and Tommi Jaakkola. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. *Advances in Neural Information Processing Systems*, 34:13757–13769, 2021.

Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks, 2022. URL `https://arxiv.org/abs/2207.09453`.

Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madisetti, Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel Rød, and Michael Bailey. Euclidean neural networks: e3nn, April 2022. URL `https://doi.org/10.5281/zenodo.6459381`.

Stefan Grimme, Christoph Bannwarth, and Philip Shushkov. A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (z= 1–86). *Journal of chemical theory and computation*, 13(5):1989–2009, 2017.

Majdi Hassan, Nikhil Shenoy, Jungyoon Lee, Hannes Stark, Stephan Thaler, and Dominique Beaini. Et-flow: Equivariant flow-matching for molecular conformer generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Jiajun He, Yuanqi Du, Francisco Vargas, Dinghuai Zhang, Shreyas Padhy, RuiKang OuYang, Carla Gomes, and José Miguel Hernández-Lobato. No trick, no treat: Pursuits and challenges towards simulation-free training of neural samplers. *arXiv preprint arXiv:2502.06685*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

Peter Holderrieth, Michael S Albergo, and Tommi Jaakkola. LEAPS: A discrete neural sampler via locally equivariant networks. *arXiv preprint arXiv:2502.10843*, 2025.

Mengjian Hua, Matthieu Laurière, and Eric Vanden-Eijnden. A simulation-free deep learning approach to stochastic optimal control. *arXiv preprint arXiv:2410.05163*, 2024.

Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron C Courville. Riemannian diffusion models. *Advances in Neural Information Processing Systems*, 35:2750–2761, 2022.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. *Advances in Neural Information Processing Systems*, 35:24240–24253, 2022.

H J Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11), nov 2005.

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.

Leon Klein, Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Advances in Neural Information Processing Systems*, 36, 2024a.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 36, 2024b.

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.

Jonas Köhler, Michele Invernizzi, Pim De Haan, and Frank Noé. Rigid body flows for sampling molecular crystal structures. In *International Conference on Machine Learning*, pp. 17301–17326. PMLR, 2023.

Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch-gordan nets: a fully fourier space spherical convolutional neural network, 2018. URL `https://arxiv.org/abs/1806.09231`.

Dávid Péter Kovács, J Harry Moore, Nicholas J Browning, Ilyes Batatia, Joshua T Horton, Venkat Kapil, William C Witt, Ioan-Bogdan Magdău, Daniel J Cole, and Gábor Csányi. Mace-off23: Transferable machine learning force fields for organic molecules. *arXiv preprint arXiv:2312.15211*, 2023.

Salem Lahlou, Tristan Deleu, Pablo Lemos, Dinghuai Zhang, Alexandra Volokhova, Alex Hernández-Garcıa, Léna Néhale Ezzine, Yoshua Bengio, and Nikolay Malkin. A theory of continuous generative flow networks. In *International Conference on Machine Learning*, pp. 18269–18300. PMLR, 2023.

Greg Landrum. Rdkit documentation. *Release*, 1(1-79):4, 2013.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Guan-Horng Liu, Yaron Lipman, Maximilian Nickel, Brian Karrer, Evangelos Theodorou, and Ricky T. Q. Chen. Generalized schrödinger bridge matching. In *The Twelfth International Conference on Learning Representations*, 2024.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *arXiv preprint arXiv:2201.13259*, 2023.

Alex Matthews, Michael Arbel, Danilo Jimenez Rezende, and Arnaud Doucet. Continual repeated annealed flow transport monte carlo. In *International Conference on Machine Learning*, pp. 15196–15219. PMLR, 2022.

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, 2024.

Laurence Illing Midgley, Vincent Stimper, Gregor NC Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. In *The Twelfth International Conference on Learning Representations: ICLR 2024*, 2023.

Benjamin Kurt Miller, Mario Geiger, Tess E Smidt, and Frank Noé. Relevance of rotationally equivariant convolutions for predicting molecular properties. *arXiv preprint arXiv:2008.08461*, 2020.

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

Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

F. Neese. The orca program system. *WIRES Comput. Molec. Sci.*, 2(1):73–78, 2012. doi: 10.1002/wcms.81.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

Michele Pavon. Stochastic control and nonequilibrium thermodynamical systems. *Applied Mathematics and Optimization*, 19:187–202, 1989.

Angus Phillips, Hai-Dang Dau, Michael John Hutchinson, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Particle denoising diffusion sampler. *arXiv preprint arXiv:2402.06320*, 2024.

Philipp Pracht, Fabian Bohle, and Stefan Grimme. Automated exploration of the low-energy chemical space with fast quantum chemical methods. *Physical Chemistry Chemical Physics*, 22(14): 7169–7192, 2020.

Philipp Pracht, Stefan Grimme, Christoph Bannwarth, Fabian Bohle, Sebastian Ehlert, Gereon Feldmann, Johannes Gorges, Marcel Müller, Tim Neudecker, Christoph Plett, et al. Crest—a program for the exploration of low-energy molecular chemical space. *The Journal of Chemical Physics*, 160(11), 2024.

Philip E Protter and Philip E Protter. *Stochastic differential equations*. Springer, 2005.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

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

Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024.

Sereina Riniker and Gregory A Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12): 2562–2574, 2015.

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

S.P. Sethi. *Optimal Control Theory: Applications to Management Science and Economics*. Springer International Publishing, 2018.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2023.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *arXiv preprint arXiv:1907.05600*, 2019.

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

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds, 2018. URL `https://arxiv.org/abs/1802.08219`.

James Thornton, Michael Hutchinson, Emile Mathieu, Valentin De Bortoli, Yee Whye Teh, and Arnaud Doucet. Riemannian diffusion schr\" odinger bridge. *arXiv preprint arXiv:2207.03024*, 2022.

Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent Gaussian models in the diffusion limit. *arXiv:1905.09883*, 2019a.

Belinda Tzen and Maxim Raginsky. Theoretical guarantees for sampling and inference in generative models with latent diffusions. *arXiv:1903.01608*, 2019b.

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023.

Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nusken. Transport meets variational inference: Controlled monte carlo diffusions. In *The Twelfth International Conference on Learning Representations: ICLR 2024*, 2024.

Daniel F Veber, Stephen R Johnson, Hung-Yuan Cheng, Brian R Smith, Keith W Ward, and Kenneth D Kopple. Molecular properties that influence the oral bioavailability of drug candidates. *Journal of medicinal chemistry*, 45(12):2615–2623, 2002.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data, 2018. URL `https://arxiv.org/abs/1807.02547`.

Hao Wu, Jonas Köhler, and Frank Noé. Stochastic normalizing flows. *Advances in Neural Information Processing Systems*, 33:5933–5944, 2020.

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.

Dinghuai Zhang, Ricky T. Q. Chen, Cheng-Hao Liu, Aaron Courville, and Yoshua Bengio. Diffusion generative flow samplers: Improving learning signals through partial trajectory optimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=OIsahq1UYC`.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022.

A Related Work

Learning Augmented MCMC Markov Chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC) methods have been the standard for sampling from complex distributions using a well-designed Markov-chain. Due to prohibitively long mixing times and poor scaling in high-dimensions, existing work have combined MCMC and SMC techniques with deep learning. Albergo et al. (2019), Arbel et al. (2021) and Gabrié et al. (2022) learn better MCMC proposal distributions with variational inference via normalizing flows (Chen et al., 2018). Matthews et al. (2022) proposed learning proposal distributions for improving SMC using stochastic normalizing flow (Wu et al., 2020). Albergo & Vanden-Eijnden (2024) and Holderrieth et al. (2025) starts from MCMC procedures and optimize a learnable component by minimizing the Kolmogorov equations.

MCMC-reliant Diffusion Samplers There are many works that learn diffusion processes but rely on an auxiliary sampling mechanism to obtain the correct signal for training. More recent advances leverage score-based diffusion models (Song & Ermon, 2019; Ho et al., 2020) in unnormalized sampling tasks to improve sample efficiency and scalability. Phillips et al. (2024) and De Bortoli et al. (2024) demonstrate the effectiveness of simple regression objectives, such as score matching, while Akhound-Sadegh et al. (2024) propose iterated Denoising Energy Matching (iDEM), an offline, yet biased, algorithm for learning the score from a replay buffer, thereby overcoming scalability issues of simulation. Interestingly, Phillips et al. (2024) derived a similar loss to ours but relies on an identity that holds true only at the optimum and thus performs sequential Monte Carlo (SMC) to obtain

samples for training. In general, these approaches either require access to ground truth samples via an auxiliary sampling algorithm such as MCMC, or rely on importance-weighted estimation or resampling, all of which require extensive evaluation of the energy function.

SOC-based Diffusion Samplers In contrast, stochastic optimal control (SOC) based samplers reframe sampling tasks as an optimization problem. Zhang & Chen (2022) and Vargas et al. (2023) showcase directly optimizing for controlled processes that match the desired target distribution. This framework has been generalized in many ways Berner et al. (2023); Richter & Berner (2024); Vargas et al. (2024); Chen et al. (2024). However, all of these methods are hindered by their computational requirements, including computationally expensive differentiation through the sampling procedure, computation of higher-order derivatives in constructing the training objectives, or the need for importance sampling (*i.e.* multiple energy evaluations). Adjoint Sampling overcomes these challenges by being able to make significantly more gradient updates per generated sample and energy evaluation.

Off-policy Methods Off-policy methods (Malkin et al., 2023; Richter & Berner, 2023; Akhound-Sadegh et al., 2024; Hua et al., 2024) are those which do not need to use samples from the current model. As such, off-policy methods do not inherently make use of the gradient of the energy function, as they do not differentiate through the model sample. To alleviate this, it is typical to parameterize the model using the gradient of the energy function, and it has been found that performance strongly relies on this trick (He et al., 2025). As our setting concerns computationally expensive energy functions, this parameterization is no longer feasible. Furthermore, being off-policy does not imply being more efficient. Many off-policy methods still require the full trajectory to evaluate their loss, only being able to use sub-trajectories if the time-marginals are either additionally learned (Bengio et al., 2023; Lahlou et al., 2023; Zhang et al., 2024) or prescribed (Albergo & Vanden-Eijnden, 2024; Chen et al., 2024; Holderrieth et al., 2025). In contrast, Adjoint Sampling is an on-policy training method, explicitly uses the gradient of the energy function, but benefits significantly from requiring only (X_t, X_1) pairs, being able to use a replay buffer and Reciprocal projections to train at a significantly reduced computational cost.

Molecule Conformer Generative Models To our knowledge, no deep learning-based sampling method has been applied at scale to generate molecular conformers directly from energy without data. Boltzmann generators use importance sampling to improve performance using energy information, but require offline molecular dynamics data to learn the initial models (Noé et al., 2019). Using an annealed importance sampling method, Boltzmann generators have been able to to sample small, single particle systems without data, but their results are not amortized and require many energy evaluations (Midgley et al., 2024). Several flow-based generative modeling works have learned to generate conformers and Boltzmann distributions by training on ground truth data (Köhler et al., 2020; Jing et al., 2022; Xu et al., 2022; Hassan et al., 2024; Köhler et al., 2023; Klein et al., 2024a). However, these works did not learn solely from existing energy models.

B ADDITIONAL PRELIMINARIES

B.1 STOCHASTIC OPTIMAL CONTROL

Stochastic optimal control (SOC; Bellman (1957); Fleming & Rishel (2012); Sethi (2018)) considers general optimization problems over stochastic differential equations. Specifically, a class of SOC problems can be expressed as the following optimization problem:

$$\min_{u \in \mathcal{U}} \mathbb{E}\Big[\int_0^1 \left(\frac{1}{2} \|u(X_t^u, t)\|^2 + f(X_t^u, t) \right) \mathrm{d}t + g(X_1^u) \Big], \tag{16}$$

$$\text{s.t. } \mathrm{d}X_t^u = \left(b(X_t^u, t) + \sigma(t) u(X_t^u, t) \right) \mathrm{d}t + \sigma(t) \mathrm{d}B_t, \qquad X_0^u \sim p_0 \tag{17}$$

where in equation 17, $X_t^u \in \mathbb{R}^d$ is the state of the stochastic process, $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is commonly referred to as the control, $b : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ is a base drift, and $g : [0,1] \to \mathbb{R}^{d \times d}$ is the diffusion coefficient. These jointly define the *controlled process* $\boldsymbol{X}^u \sim p^u$ that we are interested in optimizing; often both b and g are fixed and we only optimize over the control u.

In the following proposition show that for the case of $b \equiv 0$, $f \equiv 0$ and $X_0 = 0$ that the Schrodinger-Bridge $p^*(\boldsymbol{X}) = p^{\text{base}}(\boldsymbol{X}|X_1)\mu(X_1)$ can actually be achieved by $p^u(\boldsymbol{X})$.

Proposition B.1. *For the stochastic process equation 2, there exists a unique minimizer u^* to the following optimization problem*

$$\min_u D_{\text{KL}}\left(p^u(\boldsymbol{X})||p^*(\boldsymbol{X})\right) = 0, \tag{18}$$

and the optimal controlled distribution p^{u^*} satisfies:

$$p^{u^*}_{t,1}(X_t, X_1) = p^*_{t,1}(X_t, X_1) = p^{base}_{t|1}(X_t|X_1)\mu(X_1), \tag{19}$$

where $p^{base}_{t|1}(X_t|X_1)$ is the posterior distribution of the base process conditioned on X_1.

Proof. By using the definition of the KL divergence of path measures, $p^*(\boldsymbol{X}) = p^{\text{base}}(\boldsymbol{X}|X_1)\mu(X_1)$ can be factorized as

$$D_{\text{KL}}\left(p^u(\boldsymbol{X})||p^*(\boldsymbol{X})\right) = D_{\text{KL}}\left(p^u(\boldsymbol{X})||p^{\text{base}}(\boldsymbol{X})\frac{\mu(X_1)}{p^{\text{base}}_1(X_1)}\right) \tag{20}$$

$$= D_{\text{KL}}\left(p^u(\boldsymbol{X})||p^{\text{base}}(\boldsymbol{X})\right) + \mathbb{E}_{\boldsymbol{X}\sim p^u}\log\left(\frac{p^{\text{base}}(X_1)}{\mu(X_1)}\right). \tag{21}$$

By Girsanov's Theorem (Protter & Protter, 2005) to further evaluate the right-hand side as

$$D_{\text{KL}}\left(p^u(\boldsymbol{X})||p^{\text{base}}(\boldsymbol{X})\right) + \mathbb{E}_{\boldsymbol{X}\sim p^u}\log\left(\frac{p^{\text{base}}(X_1)}{p^*(X_1)}\right)$$

$$= \mathbb{E}_{\boldsymbol{X}\sim p^u}\left[\int_0^1 \tfrac{1}{2}\|u(X_t,t)\|^2\,\mathrm{d}t + \log\left(\frac{p^{\text{base}}(X_1)}{p^*(X_1)}\right)\right].$$

This criteria corresponds exactly with a minimum-energy stochastic optimal control (SOC) problem with stage cost $\frac{1}{2}\|u(X_t,t)\|^2$ and terminal cost function $g(X_1) = \log\left(\frac{p^{\text{base}}(X_1)}{p^*(X_1)}\right)$.

$$\min_u \mathbb{E}_{\boldsymbol{X}\sim p^u}\left[\int_0^1 \tfrac{1}{2}\|u(X_t,t)\|^2\,\mathrm{d}t + g(X_1)\right] \tag{22}$$

$$\text{s.t. } \mathrm{d}X_t = \sigma(t)u(X_t,t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}B_t, \quad X_0 = 0 \tag{23}$$

We can define the cost-to-go of our problem at a particular state and time t by

$$J(u;x,t) := \mathbb{E}_{\boldsymbol{X}\sim p^u}\left[\int_0^1 \tfrac{1}{2}\|u(X_t,t)\|^2\,\mathrm{d}t + g(X_1) \mid X_t = x\right] \tag{24}$$

then we know that (under mild conditions on g) the *value function* or cost-to-go under the optimal control takes a rather surprising form and that indeed the optimal control u^* is unique and takes the form of a time-varying function of the current state (Kappen, 2005):

$$V(x,t) := \min_u J(u;x,t) = -\log \mathbb{E}_{p^{\text{base}}}\left[\exp(-g(X_1)) \mid X_t = x\right], \quad u^*(x,t) = -\sigma(t)\nabla_x V(x,t) \tag{25}$$

With this fact about the optimal solution of the SOC problem, we can upper bound the aforementioned path KL divergence between the optimally controlled process and our target Schrodinger-bridge as

$$D_{\text{KL}}\left(p^{u^*}(\boldsymbol{X})||p^*(\boldsymbol{X})\right) = V(X_0,0) = -\log \mathbb{E}_{\boldsymbol{X}\sim p^{\text{base}}}\left[\exp(-g(X_1))\right] \tag{26}$$

$$= -\log\int_{\mathbb{R}^d}\mu(x)\,\mathrm{d}x = 0, \tag{27}$$

By the data processing inequality we have

$$D_{\text{KL}}\left(p^{u^*}_{t,1}(X_t,X_1)||p^*_{t,1}(X_t,X_1)\right) \le D_{\text{KL}}\left(p^{u^*}(\boldsymbol{X})||p^*(\boldsymbol{X})\right) = 0, \quad \forall t \in [0,1] \tag{28}$$

which implies that $p^{u^*}_{t,1}(X_t,X_1) = p^{\text{base}}_{1|t}(X_t|X_1)\mu(X_1)$ for all $t \in [0,1]$. This is exactly our claim and completes the proof. □

B.2 ADJOINT MATCHING

Adjoint Matching (Domingo-Enrich et al., 2024) is an algorithm designed to solve stochastic optimal control problems of the form equation 16. Unlike standard adjoint methods (Bryson & Ho, 1969) that directly differentiate through the objective, Adjoint Matching turns the problem into a regression formulation:

$$\mathcal{L}_{\mathrm{AM}}(u) := \mathbb{E}_{p^{\bar{u}}}\left[\frac{1}{2}\int_0^1 \left\|u(X_t,t) + \sigma(t)^\top \tilde{a}(t;\boldsymbol{X})\right\|^2 \mathrm{d}t\right], \qquad \boldsymbol{X} \sim p^{\bar{u}}, \quad \bar{u} = \texttt{stopgrad}(u), \tag{29}$$

$$\text{where} \quad \frac{\mathrm{d}}{\mathrm{d}t}\tilde{a}(t;\boldsymbol{X}) = -(\tilde{a}(t;\boldsymbol{X})^\top \nabla_x b(X_t,t) + \nabla_x f(X_t,t)), \tag{30}$$

$$\tilde{a}(1;\boldsymbol{X}) = \nabla_{X_1} g(X_1). \tag{31}$$

where $\tilde{a}$ is referred to as the "lean adjoint" state, and $\texttt{stopgrad}(\cdot)$ denotes a stop gradient operation, *i.e.*, although $\boldsymbol{X} \sim p^{\bar{u}}$ is sampled according to the controlled process, Adjoint Matching does not differentiate through the sampling procedure.

Note that for the general problem formulation equation 16, Adjoint Matching requires two simulations, one to sample a trajectory from the stochastic process $\boldsymbol{X} \sim p^u$ and one to solve the lean adjoint state backwards in time equation 30 from a terminal condition at $t = 1$ equation 31.

Adjoint Matching is greatly simplified for simple base processes. One of our key observations is that the Adjoint Matching algorithm greatly simplifies when the $b = 0$ and $f = 0$, since then equation 30 is zero. This means that the lean adjoint state $\tilde{a}(t,\boldsymbol{X}) = \nabla g(X_1)$ for all t. The Adjoint Matching algorithm hence reduces to

$$\mathcal{L}_{\mathrm{AM}}(u) := \mathbb{E}_{p^{\bar{u}}}\left[\frac{1}{2}\int_0^1 \left\|u(X_t,t) + \sigma(t)^\top \nabla_{X_1} g(X_1)\right\|^2 \mathrm{d}t\right], \qquad \boldsymbol{X} \sim p^{\bar{u}}, \quad \bar{u} = \texttt{stopgrad}(u). \tag{32}$$

C BASE PROCESS DERIVATIONS

C.1 EUCLIDEAN SPACE

Our base process is modeled by a stochastic differential equation ($X_t \in \mathbb{R}^n$ and $\sigma : \mathbb{R} \to \mathbb{R}^+$):

$$dX_t = \sigma(t)dB_t, \qquad X_0 = 0 \tag{33}$$

corresponding to equation 2 with the control set to zero, *i.e.*, $u_t = 0$. With this, the forward transition distributions are ($t > s$):

$$p^{\text{base}}_{t|s}(x|X_s) = \mathcal{N}(x|X_s, \nu_{t|s}I), \qquad \text{where } \nu_{t|s} = \int_s^t \sigma(s)^2 ds. \tag{34}$$

With $s = 0$ and $X_0 = 0$, we obtain the time marginals of equation 33:

$$p^{\text{base}}_{t}(x) = \mathcal{N}(x|0, \nu_t I), \qquad \text{where } \nu_t = \int_0^t \sigma(s)^2 ds. \tag{35}$$

Furthermore, the backward transition distributions are ($t > s$):

$$p^{\text{base}}_{s|t}(x|X_t) = \frac{p^{\text{base}}_s(x)p^{\text{base}}_{t|s}(X_t|x)}{p^{\text{base}}_t(X_t)} = \mathcal{N}(x|\alpha_{s|t}X_t, \overleftarrow{\nu}_{s|t}I), \tag{36}$$

$$\text{where } \alpha_{s|t} = \frac{\nu_s}{\nu_s + \nu_{t|s}}, \quad \overleftarrow{\nu}_{s|t} = \left(\frac{1}{\nu_s} + \frac{1}{\nu_{t|s}}\right)^{-1} = \alpha_{s|t}\nu_{t|s} \tag{37}$$

As such, the Reciprocal Adjoint Matching objective equation 12 becomes tractable when $\nu_{t|s}$ can be derived in closed form.

Constant noise schedule. Setting the diffusion coefficient to be a constant $\sigma(t) = \sigma$ results in

$$\nu_{t|s} = \sigma^2(t-s), \qquad \alpha_{s|t} = \tfrac{s}{t} \tag{38}$$

and hence the following terminal and backward transition distributions:

$$p_1^{\text{base}}(X_1) = \mathcal{N}(X_1; 0, \sigma^2 I), \qquad p_{t|1}^{\text{base}}(x|X_1) = \mathcal{N}(x; tX_1, \sigma^2(1-t)tI). \tag{39}$$

Geometric noise schedule. Song et al. (2021); Karras et al. (2022) propose the following geometric schedule (stated here in reverse-time)

$$\sigma(t) = \sigma_{\text{min}} \left(\tfrac{\sigma_{\text{max}}}{\sigma_{\text{min}}}\right)^{1-t} \sqrt{2 \log \tfrac{\sigma_{\text{max}}}{\sigma_{\text{min}}}} \tag{40}$$

results in

$$\nu_{t|s} = \sigma_{\text{max}}^2 \left(\left(\tfrac{\sigma_{\text{min}}}{\sigma_{\text{max}}}\right)^{2s} - \left(\tfrac{\sigma_{\text{min}}}{\sigma_{\text{max}}}\right)^{2t} \right), \qquad \alpha_{s|t} = \frac{\left(\tfrac{\sigma_{\text{min}}}{\sigma_{\text{max}}}\right)^{2s} - 1}{\left(\tfrac{\sigma_{\text{min}}}{\sigma_{\text{max}}}\right)^{2t} - 1} \tag{41}$$

and hence the following terminal and backward transition distributions:

$$p_1^{\text{base}}(X_1) = \mathcal{N}(X_1; 0, \sigma_{\text{max}}^2 \left(1 - \left(\tfrac{\sigma_{\text{min}}}{\sigma_{\text{max}}}\right)^2\right) I), \qquad p_{t|1}^{\text{base}}(X_t|X_t) = \mathcal{N}(X_t; \alpha_{t|1} X_1, \alpha_{t|1}\nu_{1|t} I). \tag{42}$$

C.2 Flat tori

Simulating the base process equation 33 in $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ produces a wrapped Gaussian distribution $\bar{p}_t^{\text{base}}$

$$\bar{p}_t^{\text{base}}(x) = \sum_{k \in \mathbb{Z}} p_t^{\text{base}}(x+k), \tag{43}$$

where p_t^{base} is the time marginal of the base process over $\mathbb{R}$ defined in equation 35 and $\mathbb{Z}$ is the set of integers. We can equivalently can interpret equation 43 as marginalizing a random variable k. That is, we view $p_t^{\text{base}}(x+k)$ as a joint distribution $p_t(x,k)$ over $x \in \mathbb{T}$ and $k \in \mathbb{Z}$,

$$p_t(x,k) := \underbrace{\frac{p_t^{\text{base}}(x+k)}{\int_{x' \in \mathbb{T}} p_t^{\text{base}}(x'+k)}}_{:= p_t(x|k)} \underbrace{\int_{x' \in \mathbb{T}} p_t^{\text{base}}(x'+k)}_{:= p_t(k)}. \tag{44}$$

Intuitively, the variable k represents which sub-interval on $\mathbb{R}$ was collapsed to obtain x. Furthermore, we can infer k given x at time t:

$$p_t(k|x) = \frac{p_t(x,k)}{\sum_{k'} p_t(x,k')}, \tag{45}$$

hence arriving at the following backward transition distribution ($s < t$) in which we marginalize k:

$$\bar{p}_{s|t}^{\text{base}}(x|X_t) = \sum_{k \in \mathbb{Z}} p_{s|t}(x|k, X_t) p_t(k|X_t) = \sum_{k \in \mathbb{Z}} \left(\sum_{k' \in \mathbb{Z}} p_{s|t}^{\text{base}}(x + k'|X_t + k) \right) \frac{p_t^{\text{base}}(X_t + k)}{\sum_{k' \in \mathbb{Z}} p_t^{\text{base}}(X_t + k')}, \tag{46}$$

where $p_{s|t}^{\text{base}}(x|X_t + k)$ is the backward transition probability in $\mathbb{R}$ defined in equation 36. Thus sampling from $\bar{p}_{t|1}^{\text{base}}(x|X_1)$ given in equation 46 can be done by the following sampling process

1. Sample $k \sim p(k)$ where $p(k) \propto p_1^{\text{base}}(X_1 + k)$.
2. Sample $Y_t \sim p_{t|1}^{\text{base}}(\cdot|X_1 + k)$.
3. Project $X_t = Y_t \bmod 1.0$.

For the n-dimensional $\mathbb{T}^n$, since the base process is factorized, we simply sample independently for each dimension.

D GEOMETRIC EXTENSIONS

Symmetries are essential to efficiently sample from energy-based models defining physical systems. Molecules are symmetric with respect to atom permutations, rotations, translations, and parity (reflection). We reduce dimensionality and improve data efficiency by enforcing symmetries in our model of the controlled process. Our model is either parameterized by E(n) Equivariant Graph Neural Networks (EGNNs) (Satorras et al., 2021) or Tensor Field Networks (Thomas et al., 2018), implemented in `e3nn` (Geiger et al., 2022; Jing et al., 2022), which incorporate graph features while respecting symmetry constraints (Kondor et al., 2018; Weiler et al., 2018; Miller et al., 2020; Geiger & Smidt, 2022). We experiment with two controlled processes: one sampling atomic positions using EGNN, and the other sampling *torsion* angles with `e3nn`. See Appendix H for details.

D.1 GRAPH-CONDITIONED $SE(3)$-INVARIANT SAMPLING

Consider the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Each node $v_i \in \mathcal{V}$ and edge $e_{ij} \in \mathcal{E}$ has feature attributes relating the topological structure of the data (*e.g.*, atom types and bond orderings), including a spatial coordinate $x_i \in \mathbb{R}^d$. We can condition an equivariant drift u_θ on the graph features such that it is equivariant to the spatial symmetries given by the group $g \in G := \text{Aut}(\mathcal{G}) \times SO(d)$ acting via $g \cdot x = (P \otimes R)x$. (*i.e.* $u_\theta(g \cdot X_t, t; \mathcal{G}) = g \cdot u_\theta(X_t, t; \mathcal{G})$). Here $P \in \text{Aut}(\mathcal{G})$ are graph automorphisms — represented as a permutation matrix that reorders the graph nodes while preserving its structure — and $R \in SO(d)$ is an d-dimensional rotation matrix.

Translation Invariance To enforce translation invariance, the system is restricted to the zero center-of-mass subspace $\mathcal{X}^{\text{CoM}} = \{x \mid \sum_{i=1}^k x^i = 0\} \subset \mathbb{R}^{kd}$. This is achieved by projecting the particle positions $y \in \mathbb{R}^{kd}$ to the zero CoM subspace using the projection operator:

$$x = \mathcal{A}y, \quad \mathcal{A} = \left(I_k - \frac{1}{k} \mathbf{1}_k \mathbf{1}_k^\top \right) \otimes I_d, \tag{47}$$

where I_k is the identity matrix, and $\mathbf{1}_k$ is a vector of ones. With this we can define a zero CoM process.

$$\mathrm{d}X_t = \sigma(t)\mathcal{A}u_\theta(X_t, t; \mathcal{G})\,\mathrm{d}t + \sigma(t)\mathcal{A}\,\mathrm{d}B_t, \quad X_0 = 0$$

Now p^{base} is a singular Gaussian $p_t^{\text{base}}(x) = \mathcal{N}(x; 0, \nu_t \mathcal{A}\mathcal{A}^\top)$, which can be sampled by first sampling the isotropic Gaussian $\mathcal{N}(x; 0, \nu_t)$ and then projecting via $\mathcal{A}$. We have $p_t^{\text{base}}(x) \propto \mathcal{N}(x; 0, \nu_t)$ for all $x \in \mathcal{X}^{\text{CoM}}$ and the posterior $p_{t|1}^{\text{base}}$ also projects onto $\mathcal{X}^{\text{CoM}}$.

Geometric Adjoint Sampling By ensuring the drift is G-equivariant and zero CoM, we ensure the model distributions are G-invariant for all $t \in [0, 1]$ (Köhler et al., 2020; Xu et al., 2022). Putting this all together, our RAM loss equation 12 can be modified to support the proposed symmetries.

$$\mathcal{L}_{\text{GeoRAM}}(\theta) = \int_0^1 \lambda(t) \mathbb{E}_{p_{t|1}^{\text{base}} p_1^{\bar{u}}} \left[\frac{1}{2} \|\mathcal{A}(u_\theta(X_t, t; \mathcal{G}) + \sigma(t)\nabla g(X_1))\|^2 \right] \mathrm{d}t \tag{48}$$

Assuming that energy E and p_1^{base} are G-invariant, the target $-\sigma(t)\mathcal{A}\nabla g(X_1)$ for our learned drift will be G-equivariant and zero CoM.

D.2 PERIODIC BOUNDARY CONDITIONS

In many cases we may want to model a state space with periodic boundary conditions. Concretely, we consider a state space that is the flat tori in n dimensions, denoted $\mathbb{T}^n = \mathbb{R}^n / \mathbb{Z}^n$. It is a quotient space resulting from identifying any point $x = (x^1, \ldots, x^i, \ldots, x^n)$ with $(x^1, \ldots, x^i + 1, \ldots, x^n)$ for all $i \in [n]$. The derivations for the SOC objective in $\mathbb{R}^n$ equation 5 can directly be extended to this quotient space and also general Riemannian manifolds (De Bortoli et al., 2022; Thornton et al., 2022; Huang et al., 2022). Let us denote by $\bar{p}^{\text{base}}$ the distribution of the SDE modeling the base process, *i.e.*, equation 2 with $u_t = 0$, that lives on $\mathbb{T}^n$. The RAM objective equation 12 requires computing $\nabla \log \bar{p}_1^{\text{base}}(\cdot)$ as part of the terminal cost, and sampling X_t from $\bar{p}_{t|1}^{\text{base}}(\cdot|X_1)$. Simulating the base process in $\mathbb{T}^n$ produces a factorized wrapped Gaussian distribution,

$$\bar{p}_t^{\text{base}}(x) = \prod_{i=1}^n \sum_{k=-\infty}^{\infty} p_t^{\text{base}}(x^i + k), \tag{49}$$

where $\bar{p}^{\text{base}}$ is the distribution of the base process in $\mathbb{R}$. In practice, we can easily compute this up to high numerical precision by truncating the summation in equation 49. Furthermore, we can sample from the backwards transition distribution independently for each dimension $i \in [n]$:

$$k^i \sim p(k^i), \qquad p(k^i) \propto p_1^{\text{base}}(X_1^i + k^i), \qquad X_t^i = Y_t^i \bmod 1.0, \qquad Y_t^i \sim p_{t|1}^{\text{base}}(Y_t^i | X_1^i + k^i). \tag{50}$$

In practice, we consider only values of k within some truncated set $\{-k_{\text{trunc}}, \dots, k_{\text{trunc}}\}$ with k_{trunc} sufficiently large to cover the region where $p_1^{\text{base}}(X_1^i + k)$ is practically non-zero. Details for the derivation can be found in Appendix C.2.

E Theory of Reciprocal Adjoint Matching

As a preliminary, let us first define precisely what is meant be *Reciprocal projection*. We take some terminology and results from Shi et al. (2023), denoting $\mathcal{M}$ the subset of Markov path measures associated with an SDE of the form $\mathrm{d}X_t = v(X_t, t)\mathrm{d}t + \sigma(t)\,\mathrm{d}B_t$, with σ, v locally Lipschitz.

Definition E.1 (Reciprocal class, Def. 3 of Shi et al. (2023))**.** A Borel probability measure $\Pi \in \mathbf{P}(C)$ is in the Reciprocal class $\mathcal{R}(\mathbb{Q})$ of $\mathbb{Q} \in M$ if $\Pi = \Pi_{0,T}\mathbb{Q}_{(0,T)|0,T}$.

The Reciprocal projection of $\mathbb{P} \in \mathbf{P}(C)$ is defined as $\Pi^* = \mathcal{P}_{\mathcal{R}(\mathbb{Q})}(\mathbb{P}) = \mathbb{P}_{0,T}\mathbb{Q}_{(0,T)|0,T}$. By construction, Π^* belongs to the Reciprocal class $\mathcal{R}(\mathbb{Q})$. The following characterization shows that the Reciprocal projection is indeed a projection, i.e. $\mathcal{P}^2_{\mathcal{R}(\mathbb{Q})} = \mathcal{P}_{\mathcal{R}(\mathbb{Q})}$.

Lemma E.2 (Reciprocal projection characterization, Prop. 4 of Shi et al. (2023))**.** *Let* $\mathbb{P} \in \mathbf{P}(C)$*,* $\Pi^* = \mathcal{P}_{\mathcal{R}(\mathbb{Q})}(\mathbb{P})$*. Then,* $\Pi^* = \arg\min_{\mathbb{P}}\{\mathrm{KL}(\mathbb{P}|\mathbb{Q}) : \Pi \in \mathcal{R}(\mathbb{Q})\}$.

Throughout the paper, we set $\mathbb{Q} = \mathbb{B}_\nu$, i.e. $\mathbb{Q}$ is the probability measure of the process $(\int_0^t \sigma(s)\,\mathrm{d}B_s)_{t\in[0,T]}$, which we denote as $\mathcal{P}_{\mathcal{R}(\mathbb{Q})}$.

Although this projection is exactly what is used by the proposed reciprocal adjoint matching objective Equation (12), we will now proceed to define a similar projection in terms of drifts which will allow us to analyze an idealized version of the adjoint sampling algorithm.

E.1 Reciprocal projections decrease the SOC objective

In the case that $p^u = \mathbb{P}$ is generated by some drift function u, we can view the reciprocal projection as a projection of drifts $\mathcal{P}$ as proposed in Theorem 3.2, noting that the unique drift generating $\mathcal{P}_{\mathcal{R}}(p^u)$ is given as the solution of the following Schrodinger bridge problem

$$\mathcal{P}(u) := \arg\min_w \mathbb{E}_{\boldsymbol{X}\sim p^w}\left[\int_0^1 \tfrac{1}{2}\|w(X_t,t)\|^2\,\mathrm{d}t + \log\left(\frac{p^{\text{base}}(X_1)}{p^u(X_1)}\right)\right]. \tag{51}$$

This projection of drifts allows us to reason about the RAM loss with respect to both the AM and SOC functionals which are described in Proposition 3.1.

Proof of Proposition 3.1. Let $v = \mathcal{P}(u)$. Because $\mathcal{P}^2(u) = \mathcal{P}(u)$, the reciprocal projection "approximation" introduced in RAM is exactly the distribution induced by v and so $\mathcal{L}_{\text{AM}}(v) = \mathcal{L}_{\text{RAM}}(v)$.

To show that $J(u) \geq J(\mathcal{P}(u))$, we simply apply the definition of $\mathcal{P}(u)$ equation 51 which yields

$$\begin{aligned}
J(u) &= \mathbb{E}_{\boldsymbol{X}\sim p^u}\left[\int_0^1 \tfrac{1}{2}\|u(X_t,t)\|^2\,\mathrm{d}t + \left(\log\frac{p^{\text{base}}}{\mu}\right)(X_1)\right] \\
&= \mathbb{E}_{\boldsymbol{X}\sim p^u}\left[\int_0^1 \tfrac{1}{2}\|u(X_t,t)\|^2\,\mathrm{d}t + \left(\log\frac{p^{\text{base}}}{p^u} + \log\frac{p^u}{\mu}\right)(X_1)\right] \\
&\geq \mathbb{E}_{\boldsymbol{X}\sim p^v}\left[\int_0^1 \tfrac{1}{2}\|v(X_t,t)\|^2\,\mathrm{d}t + \left(\log\frac{p^{\text{base}}}{p^u}\right)(X_1)\right] + \mathbb{E}_{X_1\sim p^u}\left[\log\frac{p^u}{\mu}(X_1)\right] && \text{by equation 51} \\
&= \mathbb{E}_{\boldsymbol{X}\sim p^v}\left[\int_0^1 \tfrac{1}{2}\|v(X_t,t)\|^2\,\mathrm{d}t + \left(\log\frac{p^{\text{base}}}{p^u} + \log\frac{p^u}{\mu}\right)(X_1)\right] && p_1^v(X_1) = p_1^u(X_1) \\
&= J(v) = J(\mathcal{P}(u)).
\end{aligned}$$

□

E.2 Adjoint Sampling Preserves Critical Points of Adjoint Matching

Consider a slightly more general version of the Reciprocal Adjoint Matching (RAM) objective with arbitrary fixed reference drift v,

$$\mathcal{L}(u; v) = \mathbb{E}_{\boldsymbol{X}\sim p^v}\left[\int_0^1 \frac{1}{2}\|u(X_t, t) + \sigma(t)\nabla g(X_1)\|^2 \,\mathrm{d}t\right] \tag{52}$$

$$s.t. \quad \mathrm{d}X_t = \sigma(t)u(X_t, t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}B_t, \quad X_0 = 0 \tag{53}$$

Then the RAM and AM loss can be both be equivalently restated in terms of $\mathcal{L}$

$$\mathcal{L}_{\text{RAM}}(u) = \mathcal{L}(u; \tilde{u}), \quad \mathcal{L}_{\text{AM}}(u) = \mathcal{L}(u, \bar{u}) \quad s.t. \quad \tilde{u} = \mathcal{P}(\bar{u}), \quad , \bar{u} = \texttt{stopgrad}(u),$$

where RAM is shown by first projecting via $\mathcal{P}$, and then setting this as the reference drift with stop gradient applied.

If in our main algorithm we consider sequentially minimizing the RAM loss where starting from some drift v_i we only populate the buffer $\mathcal{B}$ with samples from p^{v_i}, we can model adjoint sampling via the following update procedure:

$$v_{i+1} = \arg\min_u \mathcal{L}(u; \tilde{v}_i), \quad s.t. \quad \tilde{v}_i = \mathcal{P}(v_i) \tag{54}$$

with this iterative formulation, we can actually re-frame Adjoint Sampling in terms of the stationary condition of the Adjoint matching loss derived in Domingo-Enrich et al. (2024). To show this, we first consider the *first variation* of $\mathcal{L}$ with respect to the first argument which is defined by

$$\delta\mathcal{L}(u; v)(\eta) := \frac{\mathrm{d}}{\mathrm{d}\varepsilon}\mathcal{L}(u + \varepsilon\eta; v)|_{\varepsilon=0} = \mathbb{E}_{\boldsymbol{X}\sim p^v}\left[\int_0^1 \langle u(X_t, t) + \sigma(t)\nabla g(X_1), \eta(X_t, t)\rangle \,\mathrm{d}t\right]$$

the first-order necessary condition of an extremum of this cost functional is that $\delta\mathcal{L}(u; v) \equiv 0$ which is to say

$$\delta\mathcal{L}(u; v)(\eta) = \mathbb{E}_{\boldsymbol{X}\sim p^v}\left[\int_0^1 \langle u(X_t, t) + \sigma(t)\mathbb{E}_{p^v}\left[\nabla g(X_1) \mid X_t\right], \eta(X_t, t)\rangle \,\mathrm{d}t\right] = 0, \tag{55}$$

for all $\eta \in \mathcal{C}(\mathbb{R}^d \times [0, 1], \mathbb{R})$. We know that if p^v has support over $\mathbb{R}^d$ and $\mathbb{E}_{p^v}\left[\int_0^1 \langle \frac{\delta\mathcal{L}}{\delta u}(X_t, t), \eta(X_t, t)\rangle \,\mathrm{d}t\right] = 0$ for all η, then it must be that $\frac{\delta\mathcal{L}}{\delta u}(u; v)(x, t) = 0$ almost everywhere. This is the functional derivative with respect to u which satisfies

$$\frac{\delta\mathcal{L}}{\delta u}(u; v)(x, t) := u(x, t) + \sigma(t)\mathbb{E}_{p^v}\left[\nabla g(X_1) \mid X_t = x\right] = 0, \quad a.e. \tag{56}$$

In view of this general necessary condition, we can identify the corresponding necessary condition for $\mathcal{L}_{\text{AM}}(u)$.

$$\frac{\delta\mathcal{L}_{\text{AM}}}{\delta u}(u)(x, t) := \frac{\delta\mathcal{L}}{\delta u}(u; \bar{u})(x, t) = u(x, t) + \sigma(t)\mathbb{E}_{p^{\bar{u}}}\left[\nabla g(X_1) \mid X_t = x\right] = 0, \quad a.e. \tag{57}$$

This coincides with the functional derivative derived in Domingo-Enrich et al. (2024), where it was observed that the critical points of Adjoint Matching are exactly when equation 57 is zero. Furthermore, it was determined that the only critical point is the optimal control u^* to the SOC problem equation 7. We can now state Theorem 3.2 more precisely and provide a proof in the following Theorem.

Theorem E.3 (Adjoint Sampling)**.** *Suppose we have a feedback u_i. Then performing the iteration equation 54 to obtain u_{i+1} is equivalent to the iteration*

$$u_{i+1} = \mathcal{P}(u_i) - \frac{\delta\mathcal{L}_{AM}}{\delta u}(\mathcal{P}(u_i)), \tag{58}$$

Moreover, u is a fixed point satisfying $u = \mathcal{P}(u)$ and $u = u - \frac{\delta\mathcal{L}_{AM}}{\delta u}(u)$ if and only if u is a critical point of $\mathcal{L}_{AM}$ (i.e. it is the optimal control u^* to equation 7).

Proof. By minimizing $\mathcal{L}(u; \tilde{u}_i)$ point-wise for a fixed $\tilde{u}_i = \mathcal{P}(u_i)$, we know the optimal iterates take the form

$$u_{i+1}(x,t) = -\sigma(t)\mathbb{E}_{p^{\tilde{u}_i}}\left[\nabla g(X_1) \mid X_t = x\right]. \tag{59}$$

Recalling the Adjoint Matching loss $\mathcal{L}_{\text{AM}}(u) = \mathcal{L}(u; \bar{u})$ we can identify its functional derivative at $\mathcal{P}(u_i)$ that stems from first variation to be

$$\frac{\delta \mathcal{L}_{AM}}{\delta u}(\mathcal{P}(u_i)) = \mathcal{P}(u_i) + \sigma(t)\mathbb{E}_{p^{\mathcal{P}(u_i)}}\left[\nabla g(X_1) \mid X_t = x\right]. \tag{60}$$

Through a simple algebraic manipulation, we can arrive at a new expression for the iterate given by

$$u_{i+1}(x,t) = \mathcal{P}(u_i)(x,t) - \frac{\delta \mathcal{L}_{AM}}{\delta u}\left(\mathcal{P}(u_i)\right)(x,t).$$

We will now show that v is a fixed point of the Adjoint Sampling iteration and $\mathcal{P}(v) = v$ if and only if v is a critical point of $\mathcal{L}_{\text{AM}}$ (and hence $v = u^*$). First of all, if v is a critical point of $\mathcal{L}_{\text{AM}}$, then it must be $v = u^*$ (Domingo-Enrich et al., 2024) and $\frac{\delta \mathcal{L}_{AM}}{\delta u}(v) = 0$. Because $v = u^*$ is the optimal solution of the SOC loss, it is its own projection and so $\mathcal{P}(u^*) - \frac{\delta \mathcal{L}_{AM}}{\delta u}(\mathcal{P}(u_i))(u^*) = u^*$. Therefore v is a fixed point of the Adjoint Sampling iteration.

On the other hand, if v is a fixed-point of the Adjoint Sampling iteration and $\mathcal{P}(v) = v$, our iteration formula tell us that $\frac{\delta \mathcal{L}_{AM}}{\delta u}(v) = 0$ and so v is a critical point of $\mathcal{L}_{\text{AM}}$ and that $v = u^*$ necessarily. □

There are some important practical difference between this statement and what we do in practice. In practice, we optimize over a finite size buffer $\mathcal{B}$, where the iterations discussed here in Theorem E.3 assumes we sample from $p_1^{u_i}$. Furthermore, we do not need to exactly solve the RAM step to convergence in practice and only partially optimize for a fixed number of iterations before collecting new samples for $\mathcal{B}$. This may be desirable as it will have a smoothing affect on our functional gradient descent of Theorem E.3 which tracks a large unit step-size gradient update.

F Bridge Matching and Pretraining

There may be situations where we have some initial data distribution p^{data}, but want to improve our generated samples even further using an energy function. We can in fact pretrain our control drift u_θ on this data using *Bridge Matching* (Shi et al., 2023; Liu et al., 2024), which can be similarly formulated as a controlled SDE diffusing from $X_0 = 0$ and switch to Adjoint Sampling from the energy without changing the optimal distribution.

Specifically, given our controlled stochastic process Equation (2), we can write down the following Bridge Matching loss to regress onto

$$\int_0^1 \lambda(t)\mathbb{E}_{p^{\text{base}}_{t|1}(X_t|X_1)p^{\text{data}}(X_1)}\|\sigma(t)u_\theta(X_t,t) - \sigma(t)^2\nabla_{X_t}\log p^{\text{base}}_{1|t}(X_1|X_t)\|^2 \mathrm{d}t, \tag{61}$$

where for general base process $\mathrm{d}X_t = \sigma(t)\,\mathrm{d}B_t$, we have

$$p^{\text{base}}_{1|t}(x|X_t) = \mathcal{N}(x|X_t, \nu_{1|t}I), \quad \nu_{1|t} = \int_t^1 \sigma(s)^2\,\mathrm{d}s.$$

Therefore our score target is

$$\sigma(t)^2\nabla_{X_t}\log p^{\text{base}}_{1|t}(X_1|X_t) = \sigma(t)^2\nabla_{X_t}\left(-\frac{1}{2}\frac{(X_1 - X_t)^2}{\nu_{1|t}} + \text{const}\right) = \sigma(t)^2\frac{X_1 - X_t}{\nu_{1|t}}.$$

Additionally, we apply a scaling $\lambda(t) := \frac{\sqrt{\alpha(t)}}{\sigma(t)}$ to the loss so that the target is $(X_1 - X_t)$ which improves numerical stability, resulting in the following Bridge Matching objective.

$$\mathcal{L}_{\text{BM}}(\theta) = \int_0^1 \mathbb{E}_{p^{\text{base}}_{t|1}(X_t|X_1)p^{\text{data}}(X_1)}\left\|\frac{\nu_{1|t}}{\sigma(t)}u_\theta(X_t,t) - (X_1 - X_t)\right\|^2 \mathrm{d}t \tag{62}$$

G SYNTHETIC ENERGY EXPERIMENT DETAILS

G.1 DOUBLE WELL POTENTIAL (DW-4)

We use the same double-well potential as in iDEM (Akhound-Sadegh et al., 2024), which was originally proposed in Köhler et al. (2020). DW-4 describes a pair-wise distance potential energy for a system of 4 particles $\{x_1, x_2, x_3, x_4\}$, where each particle has 2 spatial dimensions $x_i \in \mathbb{R}^2$ $(d = 8)$. The potentials analytical form is given by:

$$E(x) = \frac{1}{\tau} \sum_{ij} a(d_{ij} - d_0) + b(d_{ij} - d_0)^2 + c(d_{ij} - d_0)^4, \quad d_{ij} = \|x_i - x_j\|_2 \tag{63}$$

where we set $a = 0$, $b = -4$, $c = 0.9$ and temperature $\tau = 1$.

G.2 LENNARD-JONES POTENTIAL (LJ-13, LJ-55)

Similarly to DW-4, the Lennard-Jones potential is also based pair-wise distances of a system with n particles, but each each particle has 3 spatial dimensions. Its analytical form is given by

$$E^{\text{LJ}}(x) = \frac{\epsilon}{\tau} \sum_{ij} \left(\left(\frac{r_m}{d_{ij}} \right)^6 - \left(\frac{r_m}{d_{ij}} \right)^{12} \right), \quad d_{ij} = \|x_i - x_j\|_2 \tag{64}$$

where r_m, τ, ϵ and c are physical constants. As in Köhler et al. (2020) and Akhound-Sadegh et al. (2024), we add the additional a harmonic potential:

$$E^{\text{osc}}(x) = \frac{1}{2} \sum_i \|x_i - x_{\text{COM}}\|^2 \tag{65}$$

where x_{COM} refers to the center of mass of the system. Therefore, the final energy is then $E^{\text{Tot}} = E^{\text{LJ}}(x) + cE^{\text{osc}}(x)$, for c the oscillator scale. As in previous work, we use $r_m = 1, \tau = 1, \epsilon = 1$ and $c = 1.0$.

G.3 ARCHITECTURES AND HYPER PARAMETERS

In this section we give a brief overview of the parameters used to train Adjoint Sampling on the synthetic energy functions. For all experiments, we an EGNN architecture similar to Akhound-Sadegh et al. (2024). Results for iDEM on all experiments are reproduced using the same configurations found in Akhound-Sadegh et al. (2024).

DW-4: For the DW-4 energy we use an EGNN with 3 layers and 128 hidden features. We train Adjoint Sampling for 1000 outer iterations, generating 512 new samples and energy evaluations per iteration into a buffer of max size 10000. In each iteration we optimize on 500 batches of batch size 512 from the replay buffer, using a learning rate of 3×10^{-4}. We use a geometric noise schedule with $\sigma_{min} = 10^{-4}$ and $\sigma_{max} = 3.0$.

LJ-13: For the LJ-13 energy we use an EGNN with 5 layers and 128 hidden features. We train Adjoint Sampling for 1000 outer iterations, generating 1024 new samples and energy evaluations per iteration into a buffer of max size 10000. In each iteration we optimize on 500 batches from the replay buffer of batch size 512, using a learning rate of 3×10^{-4}. We use a geometric noise schedule with $\sigma_{min} = 10^{-3}$ and $\sigma_{max} = 3.0$.

LJ-55: For the LJ-55 energy we use an EGNN with 5 layers and 128 hidden features. We train Adjoint Sampling for 1000 outer iterations, generating 128 new samples and energy evaluations per iteration into a buffer of max size 10000. In each iteration we optimize on 500 batches from the replay buffer of batch size 512, using a learning rate of 3×10^{-4}. We use a geometric noise schedule with $\sigma_{min} = 0.5$ and $\sigma_{max} = 3.0$.

G.4 REPORTED METRICS

Geometric $\mathcal{W}_2$ Because our energy and diffusion process is invariant to rotational and permutations symmetries, we also take these symmetries into account when measuring distance between

generated and ground truth point clouds (*e.g.*, from long run MCMC). Note this $\mathcal{W}_2$ metric is different from what is reported in Akhound-Sadegh et al. (2024), which uses the euclidean metric.

$$\mathcal{W}_2(\mu, \nu) = \left(\inf_{\pi} \int \pi(x, y) d(x, y)^2 \, \mathrm{d}x \, \mathrm{d}y \right)^{\frac{1}{2}} \tag{66}$$

where π is the transport plan with marginals constrained to μ and ν respectively. Here the distance metric d takes into account all point-cloud symmetries, and is obtained by minimizing the squared Euclidean distance over all possible combinations of rotations, reflections ($O(d)$), and permutations ($\mathbb{S}(k)$) for k particles of spatial dimension d.

$$d(x_0, x_1) = \min_{R \in O(d), P \in \mathbb{S}(k)} \|x_0 - (R \otimes P) x_1\|_2^2. \tag{67}$$

However, computing the exact minimal squared distance is computationally infeasible in practice. Therefore, we adopt the approach of Köhler et al. (2020) and approximate the minimizer by performing a sequential search

$$d(x_0, x_1) \approx \min_{R \in SO(d)} \|x_0 - (R \otimes \tilde{P}) x_1\|_2^2, \quad \tilde{P} = \arg\min_{P \in \mathbb{S}(k)} \|x_0 - P x_1\|_2^2. \tag{68}$$

Energy $\mathcal{W}_2$ ($E(\cdot)\,\mathcal{W}_2$) An informative way to assess the quality of the generated samples is to look at their energy distribution with respect to the ground truth distribution obtained from energies of long-run MCMC simulations. This shows how well the generated samples are avoiding high-energy regions. Specifically, we use the Wasserstein-2 ($\mathcal{W}_2$) distance on the 1-dimensional energy distribution on $\mathbb{R}$. This results in the standard euclidean $\mathcal{W}_2$ metric

$$\mathcal{W}_2(E_\mu, E_\nu) = \left(\inf_{\pi} \int \pi(x, y) |x - y|^2 \, \mathrm{d}x \, \mathrm{d}y \right)^{\frac{1}{2}}, \tag{69}$$

Path effective sample size (path-ESS). We use importance weights over paths $\boldsymbol{X} \sim p^u$ (Zhang & Chen, 2022),

$$w(\boldsymbol{X}) = \frac{\mathrm{d}p^*(\boldsymbol{X})}{\mathrm{d}p^u(\boldsymbol{X})} = \frac{\mu(X_1)}{p_1^{\text{base}}(X_1)} \frac{\mathrm{d}p^{\text{base}}(\boldsymbol{X})}{\mathrm{d}p^u(\boldsymbol{X})} \tag{70}$$

$$= \exp\left(-\frac{1}{2} \int_0^1 \|u(X_t, t)\|^2 \, \mathrm{d}t - \int_0^1 u_t(X_t, t)^\top \mathrm{d}B_t - g(X_1) \right). \tag{71}$$

As these are unnormalized weights, we use the normalized effective sample size:

$$\text{ESS} = \frac{1}{n \mathbb{E}_{\boldsymbol{X} \sim p^u}[w(\boldsymbol{X})]} \approx \frac{\left(\sum_{i=1}^n w_i\right)^2}{n \sum_{i=1}^n w_i^2}, \tag{72}$$

estimated over n samples $\boldsymbol{X}_i \overset{iid}{\sim} p^u$ with weights $w_i = w(\boldsymbol{X}_i)$. Additionally, this ESS is normalized by the number of samples n and takes a value in $[0, 1]$.

H Sampling Molecular Conformers

Conformer generation is a fundamental task in computational chemistry with applications from drug discovery to catalyzing particular reactions of interest. In the context of this paper, conformers represent different stable states, i.e. local minima on the potential energy surface, of the same molecule. The number of conformers a molecule has is generally proportional to the number of rotatable bonds present in the molecule. For example, large molecules with lots of rotatable bonds have many conformers. A bond is considered rotatable if there is a sufficiently low energy barrier to rotation and the bond is attached to non-terminal heavy atoms (Veber et al., 2002). We define a specific substructure to match this definition in Appendix H.2 and use it to determine the torsion angles sampled by

our torsion model. However, we apply a slightly different, more restrictive, definition to classify molecules into strata based on the standard definition from RDKit in Appendix I.

In many molecules and polymers torsion angles are particularly important degrees of freedom for sampling conformations. A molecule has $3k-6$ degrees of freedom corresponding to relevant atomic motion. This quantity subtracts global translation (3 degrees) and orientation in space (3 degrees), which are both irrelevant to the intrinsic molecule. The remaining degrees of freedom can be broken down into bond lengths (pairs of atoms), bond angles (3-tuples containing two atoms bonded to one common neighbor), and torsion angles (4-tuples containing a central bond and two atoms connected at either end). The energy required to stretch a bond or distort a bond angle is typically much higher than rotating a torsion angle. As a result, torsion angles are the most important degrees of freedom for determining the distribution of possible molecular states, or the Boltzmann distribution, and only allowing torsion angles to rotate is a common approximation used in many algorithms.

Relaxation Conformers represent local minima in the energy landscape; therefore, we are interested in descent algorithms to refine structures generated by Adjoint Sampling or RDKit to reach those minima. We apply the same energy model, eSEN, that we utilized to train our drift estimator on energy to define an optimization target. Doing gradient descent on energy using forces is called relaxation. In our relaxations, we use the LBFGS algorithm and bound the maximum force norm $F_{\max} := \max_i \|\nabla_{x_i} E(\tilde{X}_1)\|$ on any of the k atoms as a stopping criteria, where $\tilde{X}_1$ is the latest structure in the descent sequence. We stop when $F_{\max} \leq 0.0154$ eV/Å or at 200 descent iterations, whatever comes first.

H.1 Sampling the target molecule in Cartesian coordinates

Bond Structure Energy Regularization Although we condition on the molecular graph, our drift control estimator aims to approximate gradient of the energy function. During the evolution of our stochastic trajectories we may wander through energy troughs that have alternative connectivity than that of our target molecule. Since we aim to generate conformers of a specific target molecule, we need to avoid this. We do so by adding a regularizer to the energy. This regularizer assigns atoms particular bond / no bond lengths based on empirically known interatomic radii and repulsive van der Waals radii.

$$\tilde{E}(x) = \frac{1}{\tau} E(x) + \alpha E^{\text{reg}}(x) \tag{73}$$

where $\alpha > 0$ is regularizer scaling constant. The regularizer energy function $E^{\text{reg}} : \mathbb{R}^d \to [0, \infty)$ is designed to be a "flat-bottom" potential having the property that

$$E^{\text{reg}}(x) = 0 \iff x \in \mathcal{S} \tag{74}$$

where $\mathcal{S}$ is the desired molecular structure class. In practice, we slightly relax this to be $x \in \mathcal{S} \implies E^{\text{reg}}(x) = 0$, for training stability and due to our inability to exactly characterize $\mathcal{S}$.

In practice, we determine an upper-bound distance using well-established empirical covalent radii between atoms with bonds, and lower-bound distances using van der Waal radii between atoms without bonds. We then penalize any pairwise distances which violate these bounds taking into account a relaxation factor $\gamma \geq 1$. With the adjacency matrix $A = (a_{ij})$ and atom type attribute vector $H = (h_i)$, we can determinine the molecule structure class $\mathcal{S}(A, H)$. With accompanying interatomic bond limit functions $d_{\text{bond}}(h_i, h_j)$ and $d_{\text{no-bond}}(h_i, h_j)$, we compute the regularizer iterating over all edges as follows:

$$E^{\text{reg}}(x, A, H; \gamma) = \sum_{i,j \in I_{\text{bond}}} \max\left\{||x_i - x_j||_2 - \gamma \cdot d_{\text{bond}}(h_i, h_j), 0\right\} \tag{75}$$

$$+ \sum_{k,l \in I_{\text{no-bond}}} \max\left\{\frac{1}{\gamma} \cdot d_{\text{no-bond}}(h_k, h_l) - ||x_k - x_l||_2, 0\right\} \tag{76}$$

where I_{bond} and $I_{\text{no-bond}}$ are the index pairs of A corresponding to a bond or no-bond between atoms respectively. Assuming accurate interatomic radiis, our regularizer satisfies $x \in \mathcal{S}(A, H) \implies E^{reg}(x, A, H; \gamma) = 0$ for all $\gamma \geq 1$.

Figure 4: Multiple representations of the molecule *diiodoethane* with SMILES string ICCI. (Left) Molecular graph representation. (Right) 3D coordinates representation. We chose the torsion angle as I-C-C-I, where I indicates a purple iodine atom and C indicates a gray carbon atom. We identify the torsions with a 4-tuple of indices, selecting the heaviest atoms on either side of the central bond as the first and last members. Ties are broken by arbitrary atomic index. We take the zero dihedral angle to be when first and last members of the 4-tuple are close, the so-called *cis*-isomer. Following Jing et al. (2022), we adjust the dihedral angle by rotating all the atoms at one end of the torsion about the bond axis, leaving the remaining atoms in place. This can be described as a torque pointing along the bond axis. When the torque would asymmetrically act on the molecule, we define the positive direction towards the side of the molecule with more atoms.

H.2 Torsion Angles and Torques

Representing Torsion Angles and Torques The orientation around a given bond can be quantified with a torsion angle. Specifically, a torsion angle is defined between four atoms (a, b, c, d), where the torsion angle is the angle between the planes formed by a, b, c and b, c, d, which are rotated around the central bc bond. The choice of atoms a and d are typically not unique, thus a different choice of a or d can lead a different torsion angle to describe the same molecular conformation. Our convention is that a and d correspond to the heaviest atoms on either side of the central bond bc. We break any ties by arbitrary existing atomic index. Next, we represent torsion angles symbolically... Assuming that atoms $a, b, c, d \in \mathbb{R}^3$ are positions in Cartesian space, we define a few vectors that lie along undirected bond axes:

$$dx_1 \coloneqq \overrightarrow{ab} \coloneqq b - a, \qquad dx_2 \coloneqq \overrightarrow{bc} \coloneqq c - b, \qquad dx_3 \coloneqq \overrightarrow{cd} \coloneqq d - c. \tag{77}$$

Those vectors allow us to define torsion angle as follows:

$$\varphi(dx_1, dx_2, dx_3) \coloneqq \operatorname{atan2}\left(dx_2 \cdot ((dx_1 \times dx_2) \times (dx_2 \times dx_3)), |dx_2| (dx_1 \times dx_2) \cdot (dx_2 \times dx_3)\right). \tag{78}$$

We define the function $\mathcal{T}\colon \mathbb{R}^{4\times 3} \to \mathbb{R}$ that takes in four column vectors of atomic positions (a, b, c, d) and outputs a torsion angle by equation 78, *i.e.*, $\mathcal{T}\colon (a, b, c, d) \mapsto \varphi(dx_1(a, b), dx_2(b, c), dx_3(c, d))$.

Following Jing et al. (2022), we can adjust a torsion angle by rotating *all the atoms* on one side of the bc bond about the bond axis $\overrightarrow{\mathrm{bc}}$. This can be described as a torque $\mathfrak{t}$ in the standard *angle-axis representation* where the axis points along $\overrightarrow{\mathrm{bc}}$. A positive sign indicates positive rotation according to the right-hand-rule and the magnitude indicates the rotation's effect on the existing torsion angle. The action of the rotation $R(\mathfrak{t})$ on our system X_t with constituent atoms $x_i \in \mathbb{R}^3$, can be written as follows:

$$R(\mathfrak{t})x_i = \begin{cases} R'(\mathfrak{t})(x_i - b) + b & \text{if } i \in \mathfrak{C}(c) \\ x_i & \text{otherwise} \end{cases}, \tag{79}$$

where $R'(\mathfrak{t})$ is the rotation matrix defined by the axis-angle representation of $\mathfrak{t}$, and $\mathfrak{C}(c)$ denotes the connected component of c in an identical graph to the molecule *except* with bond bc *removed*! $\mathfrak{C}(c)$ therefore denotes one "half," or "side," of the molecule when hypothetically divided at bond bc.

Since every torque points along $\overrightarrow{bc}$ by design, we need a convention for ordering torsion angle 4-tuples to determine the direction of rotation. We choose atom c to sit on the side of the bc bond with more atoms (Jing et al., 2022). To be more specific, if one broke bond bc, there would be two connected components $\mathfrak{C}(b)$ and $\mathfrak{C}(c)$. We choose c such that $|\mathfrak{C}(c)| > |\mathfrak{C}(b)|$. (This is the opposite convention to the definition Jing et al. (2022) write in their paper, but the same as in their code.) Given ties, we use the current arbitrary index.

Finding torsional degrees of freedom with RDKit Sampling conformations by sampling torsion angles requires identifying the torsional degrees of freedom in our target molecule. We use the package RDKit (Landrum, 2013) to find and represent the 4-tuples for those degrees of freedom. We look for matches to the following substructure: (1) two atoms must be connected by a bond, (2) neither end of that bond can be aromatic, (3) neither end of that bond is merely connected only to a single hydrogen, and (3) only one end of the bond can be part of a ring. Such a substructure can be encoded in a SMARTS string, specifically `[!$(*#*)&!D1]-&!@[!$(*#*)&!D1]`. This matching can be done on directly on the SMILES string of a molecule.

Substructure matching identifies the torsional degrees of freedom with a list of 4-tuples containing atoms in every position. Recall, this representation is not unique, as described above! Furthermore, this search technique typically finds more torsional degrees of freedom than RDKit returns when querying the number of rotatable bonds. RDKit has a specific definition of rotatable bond as defined implicitly in `CalcNumRotatableBonds`.

Considerations for torsion angles geometry We model a stochastic differential equation in the torsion angle space where we can reason with the base distribution more practically and the dimensionality is reduced. However, this comes with a few additional considerations. For this section, consider a molecule with d_{rot} torsional degrees of freedom. Due to the periodic boundary conditions on torsion angles, the space is modeled as a flat torus $\mathbb{D} := [-\pi, \pi)^{d_{\text{rot}}}$ where the extrema are identified with zero displacement due to the torus distance function, or so-called logmap. Let $D_{\text{rot}} = \{1, \ldots, d_{\text{rot}}\}$ denote the index set for each torsion angle.

Adjoint Sampling requires a memoryless noise schedule, which we achieve using a Dirac distribution centered at $\varphi = 0$ with $\varphi \in \mathbb{D}$ at $t = 0$. This choice of origin in $\mathbb{D}$ must have a corresponding unique position in cartesian space. Given atoms (a_i, b_i, c_i, d_i) that are constituents of torsion angle $\varphi_i \in [-\pi, \pi)$ with $i \in D_{rot}$, we choose $\varphi_i = 0$ to be the state in Cartesian space where a_i, d_i are closest in terms of Euclidean distance, *i.e.*,

$$\mathcal{T}(a_i, b_i, c_i, d_i) = 0 \iff d_i = \arg\min \|a_i - d_i\|_2. \tag{80}$$

The definition is the same for all $i \in D_{rot}$. We rely on intuition and ignore the details to prove that this choice is unique, and that the distance between a_i and d_i is the only necessary component to specify the torsion angle φ_i.

We now construct the molecule X_0 in cartesian space that has $\varphi_i = 0$ for all $i \in D_{rot}$. We first embed the molecular graph in cartesian space to an arbitrary, approximate conformer using RDKit's `EmbedMolecule` (or `EmbedMultipleConfs` with `numConfs=1`). Once embedded, we can identify atoms that are constituents of torsion angles, (a_i, b_i, c_i, d_i) with $i \in D_{rot}$, using `GetSubstructMatches` and determine the current torsion angles using $\mathcal{T}$. Given any atomic configuration X created using this method, we can reach X_0 by iteratively transforming X using $R(-\mathcal{T}(a_i, b_i, c_i, d_i))X$ for every i. Iterative application of this transformation is typical (Jing et al., 2022) in other work and will bring us to $\varphi_i = 0$ for all torsion angles. This is the starting point for Torsional Adjoint Sampling! By setting the torsion angles to zero, but retaining the atom types, molecular topology, bond lengths, and bond angles, we have an initial sample to transform with stochastic control.

Our SDE evolves in $\mathbb{D}$ starting at $\varphi_0 = 0$ and ending at φ_1 with the standard behavior of an SDE on a flat torus. However, our drift estimator of the torque is a function of X_t, implying we need to update the X_t positions as φ_t evolves under the SDE. Given drift estimate $\hat{\mathfrak{t}}(X_t)$ from our model, we update positions $X_{t+1} = \Pi_{i \in D_{\text{rot}}} R^i_{X_t}(\hat{\mathfrak{t}}_i + \epsilon_i) X_t$ with $\epsilon_i \sim \mathcal{N}(0, \sigma(t)^2 \Delta t)$. Δt the increment between steps and $\sigma(t)$ is determined by the geometric noise schedule in equation 40 as a function of time.

Regression target In Cartesian Adjoint sampling, the regression target is $\frac{1}{\tau}\nabla_X g(X_1)$ where the gradient is taken in Cartesian space. For our problem, we need to compute regression target $\frac{1}{\tau}\nabla_\varphi g(X_1)$. These two quantities are related by the Jacobian $\frac{d\varphi}{dX}$, namely $\nabla_\varphi g(X_1) = \frac{d\varphi}{dX}\nabla_X g(X_1)$. We compute this vector-Jacobian product using automatic differentiation of the function $R(\varphi_1)X_0 = X_1$ applied to the $\nabla_X g(X_1)$ vector determined by our energy model. To clarify, when we clip the gradient for the torsion model, we clip $\nabla_\varphi g(X_1)$.

Torsion space posterior Adjoint sampling's innovations include fast sampling of points along the control trajectory using a posterior. In the torsion case, we sample φ_t from the posterior $\bar{p}^{base}_{t|1}(\varphi \mid \varphi_1)$. This is done by affine transformation of the coordinates $[0, 1]$ of the flat torus distributions from Section C.2 to $[-\pi, \pi)$.

Transformation of torque $\mathfrak{t}$ When we send $X \mapsto -X$, the torque is transformed $\mathfrak{t} \mapsto -\mathfrak{t}$, such quantities are called pseudovectors and their signed magnitudes are called pseudoscalars. Following Jing et al. (2022) and the implementation in Geiger et al. (2022), our networks predict d_{rot} pseudoscalars without data augmentation.

H.3 Coverage Recall and Precision Metrics

Root Mean Square Deviation (RMSD) Similarly to Ganea et al. (2021) and Jing et al. (2022), we measure the so-called Average Minimum RMSD (AMR) and Coverage (COV) for Precision (P) and Recall (R). When measuring these metrics, we generate twice as many conformers as provided by CREST reference conformers. For $K = 2L$ let $\{C_l^*\}_{l\in[1,L]}$ and $\{C_k\}_{k\in[1,K]}$ be the sets of ground truth and generated conformers respectively. In our evaluations, we use $L = \max(L', 128)$, where L' is the number of reference conformers given by CREST, taking the lowest energy conformers as a subset. The RMSD metric finds the best average distance between atoms of molecule with respect to the reference molecule, taking into account all possible symmetries.

Coverage Recall

$$\text{COV-R}(\delta) := \frac{1}{L}\left|\{l \in \{1, \dots, L\} : \exists k \in \{1, \dots, K\}, \quad \text{RMSD}(C_k, C_l^*) < \delta\}\right| \tag{81}$$

$$\text{AMR-R} := \frac{1}{L}\sum_{l\in\{1,\dots,L\}} \min_{k\in\{1,\dots,K\}} \text{RMSD}(C_k, C_l^*) \tag{82}$$

Coverage Precision

$$\text{COV-P}(\delta) := \frac{1}{K}\left|\{k \in \{1, \dots, K\} : \exists l \in \{1, \dots, L\}, \quad \text{RMSD}(C_k, C_l^*) < \delta\}\right| \tag{83}$$

$$\text{AMR-P} := \frac{1}{K}\sum_{k\in\{1,\dots,K\}} \min_{l\in\{1,\dots,L\}} \text{RMSD}(C_k, C_l^*) \tag{84}$$

where $\delta > 0$ is the coverage threshold.

H.4 Hyperparameters and Architecture Details for SPICE and GEOM-DRUGS

Cartesian Adjoint Sampling: We use an Equivariant Graph Neural Network (EGNN, Satorras et al. 2021) with 12 layers and a hidden feature dimension of 128. The model is trained for 5000 outer-loop iterations, sampling 100 batches each iteration from the replay buffer. Each GPU maintains its own buffer with max size of 64000 samples and each iterations we generate 128×8 new molecules and energy evaluations for the buffer across 8 GPUs. The model is trained with a batch size of 64 per GPU and follows a geometric noise schedule with $\sigma_{\min} = 10^{-3}$ and $\sigma_{\max} = 1$. Weight gradient clipping of 10^{20} is applied. Additionally we ues the temperature $\tau = 5 \times 10^{-3}$ and a regularization constant $\alpha = 100.0$ for the regularized energy function. The regression target, the temperature-scaled gradient of the energy function is clipped at ℓ^2 norm 150. We use a step size of $dt = 0.001$ (*i.e.* 1000 nfe) for Euler-Maruyama SDE integration during training and evaluation.

Torsional Adjoint Sampling: We use a different equivariant graph neural networked called `e3nn` (Geiger & Smidt, 2022; Geiger et al., 2022) with 6 convolutional layers with 32 scalar features and

8 vector features at each layer. There is additionally a specialized layed called a pseudotorque layer by Jing et al. (2022) that we use at output to predict the pseudoscalar torque quantities. Our edge and node embeddings have 32 scalar features. The cutoff radius is 5.0 and we construct 50 Gaussian radial basis functions to embed distances between atoms. We use 8 GPUs with 1024×8 initial molecules and each iteration we generate 128×8 new molecules and evaluate their energy (gradient), which are then added to the GPU buffers. The model is trained for 3000 outer-loop iterations, sampling 100 batches each iteration from the replay buffer with max size 64000. The model is trained with a batch size of 64 and follows a geometric noise schedule with $\sigma_{\text{min}} = \pi \times 10^{-2}$, $\sigma_{\text{max}} = \pi$, and a "number of tiles" truncation parameter of 6 to approximate p^{base} on the flat tori (Appendix H.2). Weight gradient clipping of 10^{20} is applied. Additionally we use the temperature $\tau = 5 \times 10^{-3}$ and the direct energy function without any regularization. The regression target, the temperature-scaled gradient of the energy function w.r.t. torsions is clipped at ℓ^2 norm 200. We use a step size of $dt = 0.01$ (*i.e.* 100 nfe) for Euler-Maruyama SDE integration during training and evaluation.

I DATA PREPARATION FOR THE CONFORMATION BENCHMARK

Adjoint Sampling, in this context, takes in SMILES strings and outputs conformations. Here we explain the splits of the input SMILES strings and representative coordinates samples of conformations. Additionally, we explain the processes we used to produce, relax, and deduplicate our reference atomic conformations with RDKit (Landrum, 2013), CREST (Pracht et al., 2024), and ORCA (Neese, 2012) by following an extremely similar procedure to that of GEOM-DRUGS (Axelrod & Gomez-Bombarelli, 2022).

We are releasing the data we used in this paper as a challenging *Conformation Benchmark* with the goal of fostering development of scalable, amortized sampling algorithms. The data has three parts:

- A training split of 24,477 SMILES strings from SPICE (Eastman et al., 2023). These define the molecular topology for the target molecule, but do not have any atomic coordinate information. The molecules have differing levels of flexibility with between 0 and 18 rotatable bonds[2].
- A test split[3] of 80 SMILES strings from SPICE divided into groups of 10, each with between 3 and 10 rotatable bonds. We also release 44,448 DFT annotated and geometry optimized conformations.
- A test split of 80 SMILES strings from GEOM-DRUGS divided into groups of 10, each with between 3 and and 10 rotatable bonds. We identify 7024 conformations corresponding to those SMILES strings.

The details of are described below in Appendix I.1. Since we are repackaging data from SPICE and GEOM-DRUGS, our main contributions include (a) the organization of the benchmark and (b) the computational effort of finding 44,448 conformations from our SPICE test split using a similar procedure to that of GEOM-DRUGS.

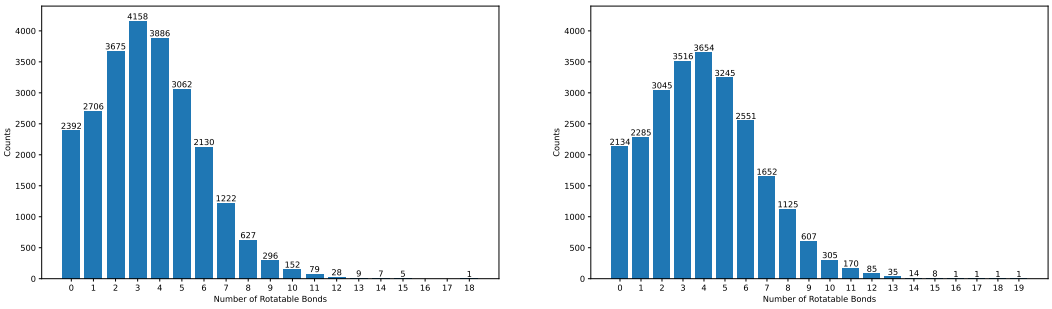
One utilizes the benchmark by applying a sampling algorithm to find conformers of a given molecular topology and determining whether the algorithm recovers the samples we provided. In Figure 3, we report this as Coverage Recall % versus RMSD. We additionally provide tabulated values in Table 1. The conformers we provide should be treated as held-out data generated using an effective and expensive standard method. We do not claim to have comprehensively sampled the conformation space for these molecules, but we argue that they are a representative sample given modern methods. Furthermore, since we only release local minima, the sample does not come from a Boltzmann distribution at a fixed temperature, *i.e.* the samples are not drawn from a probability distribution. There is no sense of a conformer being more "likely" than another in a probabilistic manner in our data. Instead, the benchmark represents a loosely defined notion of sampling by finding many low energy conformations in an attempt to cover the configuration space.

[2]Rotatable bonds are defined by `RDKit`'s function `CalcNumRotatableBonds` (Landrum, 2013).

[3]The test split is disjoint from the train split and their union is all of SPICE.

I.1 SPICE DATASET

We used the PubChem subset of the SPICE dataset (Eastman et al., 2023) to train and evaluate our models. This subset includes small, drug-like molecules comprised of between 18 and 50 atoms with elements Br, C, Cl, F, H, I, N, O, P, and S. We are interested in conformations of molecules, implying



`CalcNumRotatableBonds` `GetSubstructMatches` with SMARTS

Figure 5: Histograms of rotatable bonds and torsional degrees of freedom in the SPICE training set. Left: Number of rotatable bonds determined using RDKit's `CalcNumRotatableBonds`. Right: Number of torsional degrees of freedom determined by `GetSubstructMatches` with SMARTS string `[!$(*#*)&!D1]-&!@[!$(*#*)&!D1]`.

that we care about the rotational degrees of freedom. We identify two different sets describing these degrees of freedom: *number of rotatable bonds* and *torsional degrees of freedom*, or *torsions* for short. In our case, rotatable bonds $\subseteq$ torsional degrees of freedom. We compute the number of rotatable bonds using the RDKit package `CalcNumRotatableBonds`. We count the number of torsional degrees of freedom using the SMARTS string `[!$(*#*)&!D1]-&!@[!$(*#*)&!D1]` and `GetSubstructMatches`, also from RDKit. We group molecules into classes according to the number of rotatable bonds, but we provide the torsional degrees of freedom to our Adjoint Sampling Torsion model. The distributions of rotatable bonds and the comparison between the two sets of rotational degrees of freedom can be found in Figure 5 and Figure 6. We find fewer rotatable bonds than torsions matching our substructure query, which is expected. More information about our torsional model is in Appendix H.

We randomly subsampled 10 molecules from each class of molecules with a fixed number of rotatable bonds, in the range of 3-10, to create the test set. We performed pre-processing and CREST (Pracht et al., 2024) on the resulting 80 molecules. Our procedure follows GEOM-DRUGS (Axelrod & Gomez-Bombarelli, 2022).

During pre-processing, we first used RDKit to generate 50 conformers with `EmbedMultipleConfs` using parameters `pruneRmsThresh=0.01, maxAttempts=5, useRandomCoords=False` corresponding to a pruning threshold of similar conformers of 0.01 Å, a maximum of five embedding attempts per conformer, coordinate initialization from the eigenvalues of the distance matrix, and a random seed. If no conformers were successfully generated then `numConfs` was increased to 500. Afterwards, the conformers were optimized in the RDKit default MMFF force field. We deduplicated the conformers using RDKit's `GetBestRMS`, unless the calculation took longer than 48 hours then we switched to RDKit's `AlignMol`. (97 used `GetBestRMS`, 3 used `AlignMol`.) After removing the duplicate conformers that exhibited an RMSD $< 0.1\mathring{A}$, the ten conformers with the lowest energy were further optimized with an approximate energy model known as *extended tight binding* (xTB) (Grimme et al., 2017; Friede et al., 2024). The conformer with the lowest xTB energy was used as input to the CREST simulation. We used the default hyperparameters from CREST version 3.0.2 including a 6 kcal/mol cutoff on final conformers. Any SMILES strings containing slash or backslash indicate a cis/trans stereochemistry and were skipped since their SMILES string is not preserved during the above procedure.

The output of the above procedure was a thoroughly sampled potential energy surface by the standards of modern methods; however, the structures are optimized using the xTB level of accuracy.

We took the process a step further and performed geometry optimizations using DFT with ORCA on each conformer in the test set. Specifically, we took the output of CREST and ran ORCA relaxation with settings:

```
Opt wB97M-V def2-tzvpd RIJCOSX def2/J NoUseSym DIIS NOSOSCF NormalConv DEFGRID2
NormalPrint AllPop
%scf Convergence Tight maxiter 300 end
%elprop Dipole true Quadrupole true end
%output Print[P_ReducedOrbPopMO_L] 1 Print[P_ReducedOrbPopMO_M] 1
Print[P_BondOrder_L] 1 Print[P_BondOrder_M] 1 Print[P_Fockian] 1 Print[P_OrbEn] 2 end
%geom MaxIter 100 end
%scf THRESH 1e-12 TCUT 1e-13 end
```

Some lines were broken in the latex version of the code snippet above due to space limitations. These results are run at a very high level of DFT theory. Further geometry optimization may be desired at `DEFGRID3` with `TightOpt`; however, such an extreme precision was prohibitive given our resources available at the time.

After the geometry optimization, we utilized RDKit's `GetAllConformerBestRMS` and a threshold of 0.1 Å to identify pairwise duplicates and remove them from the dataset. The average number of reference conformers for each number of rotatable bonds is presented in the left side of Figure 7.

We investigated the reason that the 80 molecule test set from SPICE generated an order of magnitude more conformers than the 80 molecule GEOM-DRUGS test set. We found, after replicating the GEOM-DRUGS conformers, that the molecules in the SPICE test set are inherently more flexible than the ones from the GEOM-DRUGS test set. GEOM-DRUGS molecules tended to have more rings and hydrogen bonds, leading to fewer conformations. In other words, SPICE tended to have a flatter potentially energy surface. Since we followed the GEOM-DRUGS procedure, we applied the same 6 kcal/mol cutoff in CREST and kept many more conformers from SPICE. For Coverage Recall %, we capped the reference conformers per molecule at 512.

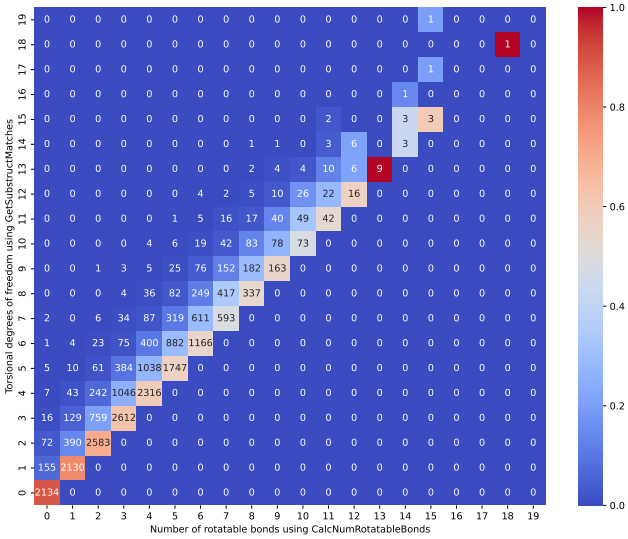
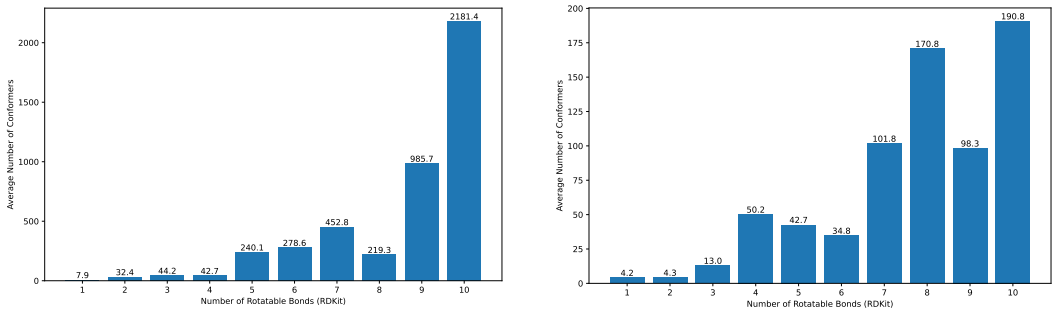
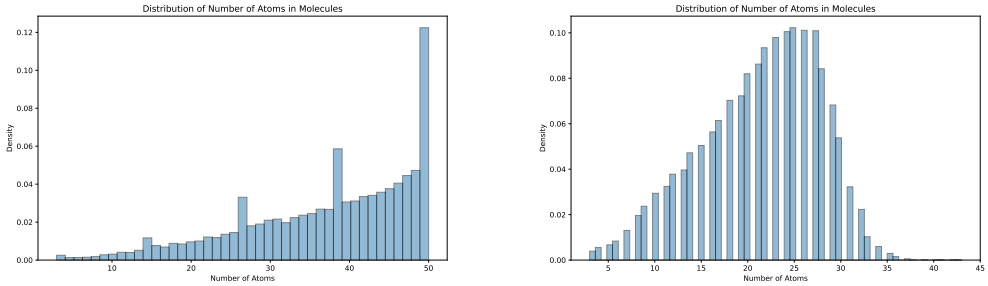


Figure 6: Correlation heatmap between the torsional degrees of freedom (y-axis) versus the number of rotatable bonds (x-axis). Torsional degrees of freedom are identified using `GetSubstructMatches` while rotatable bonds are identified using `CalcNumRotatableBonds`. There are always more or equal torsional degrees of freedom than rotatable bonds. Colors indicate a conditional probability: for each number of rotatable bonds (x-axis), the color shows the fraction of molecules with a given number of torsional degrees of freedom (y-axis). Numbers in cells show the raw counts.



SPICE GEOM-DRUGS

Figure 7: Average number of conformers in the SPICE (*left*) and GEOM-DRUGS (*right*) test sets vs number of rotatable bonds calculated by RDKit's function `CalcNumRotatableBonds`. Note the different y-axes between the figures.



SPICE: number of atoms. SPICE: number of heavy atoms (excludes Hydrogen).

Figure 8: Atom counts appearing in the SPICE dataset: (*left*): Frequency of total atom counts. (*right*): Frequency of only heavy atom counts (excluding Hydrogen atoms). Note that our conformer generation predicts positions of *all* atoms.

I.2 GEOM-DRUGS Dataset

We also evaluated our models on the GEOMDRUGS dataset (Axelrod & Gomez-Bombarelli, 2022). We took the test set from Torsional Diffusion (Jing et al., 2022), and excluded molecules that have elements other than Br, C, Cl, F, H, I, N, O, P, and S. From the remaining molecules, we randomly sampled 10 molecules for each number of rotatable bonds in the range of 3-10 (calculated by RDKit's `CalcNumRotatableBonds`). The smallest molecule was comprised of 19 atoms and the largest of 65, so slightly larger than SPICE.

We use the conformers in the GEOM-DRUGS dataset as the reference for evaluation. The conformers are represented as RDKit molecule object. To use the same format as the SPICE dataset, we extracted the Cartesian coordinates from the RDKit molecules and wrote them into `.xyz` files for each molecule. The average number of reference conformers for each number of rotatable bonds are presented in the right side of Figure 7.

J Additional Experimental Results

J.1 Synthetic Energies

Several synthetic energy functions of n-particle bodies proposed and benchmarked by previous works (Köhler et al., 2020; Midgley et al., 2023; Klein et al., 2024b; Akhound-Sadegh et al., 2024). We consider three energy functions: A 2D 4-particle Double-Well Potential (DW-4), a 3D 13-particle Lennard-Jones potential (LJ-13) and finally a a 55-particle Lennard-Jones energy (LJ-55). See Appendix G for details.

Table 2: Results for the synthetic energy function experiments. We report a geometric $\mathcal{W}_2$ metric based on on Klein et al. (2024b) and 1D energy histogram $E(\cdot)\,\mathcal{W}_2$ metric (visualized in Appendix J.1) with respect to ground truth MCMC samples. We also report a path-measure ESS when applicable. See Appendix G.4 for more details. †The values reported are per sample (*i.e.*, divided by the batch size) and according to the LJ-55 experiment hyperparameters.

	DW-4 ($d = 8$)			LJ-13 ($d = 39$)			LJ-55 ($d = 165$)			# $E(\cdot)$ evals per gradient update†	# $u_\theta(\cdot)$ evals per gradient update†
Method	path-ESS ↑	$\mathcal{W}_2$ ↓	$E(\cdot)\,\mathcal{W}_2$ ↓	path-ESS ↑	$\mathcal{W}_2$ ↓	$E(\cdot)\,\mathcal{W}_2$ ↓	path-ESS ↑	$\mathcal{W}_2$ ↓	$E(\cdot)\,\mathcal{W}_2$ ↓		
PIS (Zhang & Chen, 2022)	0.462$\pm$0.081	0.68$\pm$0.23	0.65$\pm$0.25	0.012$\pm$0.011	1.93$\pm$0.07	18.02$\pm$1.12	0.001$\pm$0.000	4.79$\pm$0.45	228.70$\pm$131.27	1	1000
DDS (Vargas et al., 2023)	0.461$\pm$0.076	0.92$\pm$0.11	0.90$\pm$0.37	0.010$\pm$0.011	1.99$\pm$0.13	24.61$\pm$8.99	0.001$\pm$0.000	4.60$\pm$0.09	173.09$\pm$18.01	1	1000
LogVariance (Richter & Berner, 2023)	0.025$\pm$0.042	1.04$\pm$0.29	1.89$\pm$0.89	—	—	—	—	—	—	1	1000
iDEM (Akhound-Sadegh et al., 2024)	—	0.70$\pm$0.06	0.55$\pm$0.14	—	1.61$\pm$0.01	30.78$\pm$24.46	—	4.69$\pm$1.52	93.53$\pm$16.31	512	3
iDEM w/ 1 MC sample	—	1.21$\pm$0.02	2.70$\pm$0.18	—	2.03$\pm$0.02	22.41$\pm$0.18	—	5.79$\pm$1.60	1e32$\pm$1e32	1	3
Adjoint Sampling w/o RP (**Ablation**)	0.448$\pm$0.110	0.63$\pm$0.11	1.03$\pm$0.23	0.159$\pm$0.068	1.68$\pm$0.01	2.91$\pm$1.39	0.094$\pm$0.025	4.50$\pm$0.10	94.48$\pm$76.12	0.002	3
Adjoint Sampling (**Ours**)	0.627$\pm$0.037	0.62$\pm$0.06	0.55$\pm$0.12	0.220$\pm$0.041	1.67$\pm$0.01	2.40$\pm$1.25	0.066$\pm$0.037	4.50$\pm$0.05	58.04$\pm$20.98	0.002	3

Baselines Along with the Adjoint Sampling ablation, we compare against the most recent state-of-the-art sampler iDEM (Akhound-Sadegh et al., 2024) who claims to be first learned sampler to scale successfully to LJ-55. We also compare against PIS (Zhang & Chen, 2022) and DDS (Vargas et al., 2023), being the most comparable method to Adjoint Sampling in its SOC / reverse KL formulation. Lastly, for completeness we present an offline variant of the PIS loss referred to as the LogVariance (Richter & Berner, 2023), which is actually gradient-free. All methods use the EGNN (Satorras et al., 2021) architecture for all experiment settings, where PIS and DDS use fewer number of layers making back-propogation through the SDE tractable.

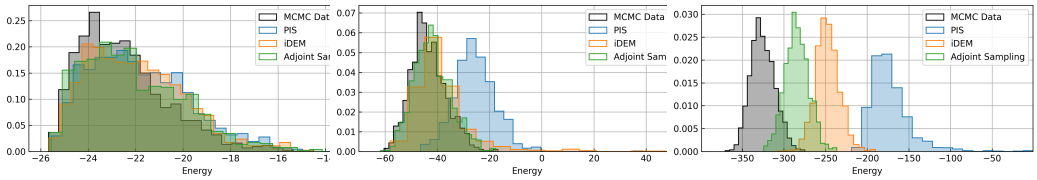
Evaluation In Table 2 we report our metrics including $\mathcal{W}_2$: the geometric 2-Wasserstien distance (taking into account symmetries by first optimizing a rigid transformation (Klein et al., 2024b) between model generated samples and the ground truth MCMC data and $E(\cdot)\mathcal{W}_2$: The 1D $\mathcal{W}_2$ distance between the distribution of energy values produced by generated samples and ground-truth and path effective sample size (ESS): Roughly measuring the proportion of informative sample trajectories with respect to each algorithms target SDE process. Since iDEM is not built on a reverse KL objective, the path-ESS metric is not meaningful and is hence omitted. Please see Appendix G.4 for more details. To demonstrate the energy sample efficiency of Adjoint Sampling we report the number of energy function evaluations per gradient update.

Results It can be seen clearly in Table 2 that Adjoint Sampling is on par with iDEM in terms of particle $\mathcal{W}$-2 distances, however iDEM struggle to remain in low energy regions as indicated by the much larger $E(\cdot)\mathcal{W}_2$ on the LJ-13 and LJ-55 experiments (visualized as energy histograms in Appendix J.1). Moreover, *iDEM uses $\mathcal{O}(100,000)$ more energy evaluations than Adjoint Sampling per gradient update* since it relies on Monte Carlo (MC) estimation of the ground truth score function. This is what makes scaling iDEM to more complex energy functions intractable. To test a more efficient version of iDEM, we also run iDEM using only a single MC sample for estimating their objective, however it proved to yield overall unstable metrics, most likely due to its biased MC score target estimator. In regards to SOC-based methods, PIS and DDS struggle to perform beyond LJ13, while the offline method LogVariance can not scale beyond DW-4. These results indicate that Adjoint Sampling is the preferred method for large and complex energy models.

Energy Histograms for Synthetic Energy Experiments We provide additional qualitative comparisons between the energy distributions generated by our synthetic energy benchmarks and those obtained via ground-truth MCMC sampling. Appendix J.1 illustrate histograms for three distinct systems — DW4, LJ13, and LJ55. These results clearly indicate that Adjoint Sampling is more effective at avoiding high-energy regions, especially in the case of LJ55, which features the highest-dimensional energy surface.

J.2 Runtime Analysis of Molecular Conformer Generation

This section further examines the computational efficiency of our proposed Adjoint Sampling technique through a detailed runtime analysis of molecular conformer generation task.



(a) DW4 (b) LJ13 (c) LJ55

Figure 9: Energy Histograms of the Synthetic energy benchmark overlayed on the ground-truth MCMC generated samples.

As illustrated in Figure 10, Adjoint Sampling overcomes two significant computational bottlenecks found in prior approaches, namely the intensive SDE simulation and the costly energy evaluation steps. By mitigating these challenges, our method is able to perform substantially more gradient updates within the same execution time, leading to enhanced overall performance in tasks with expensive energy function evaluation, such as conformer generation.

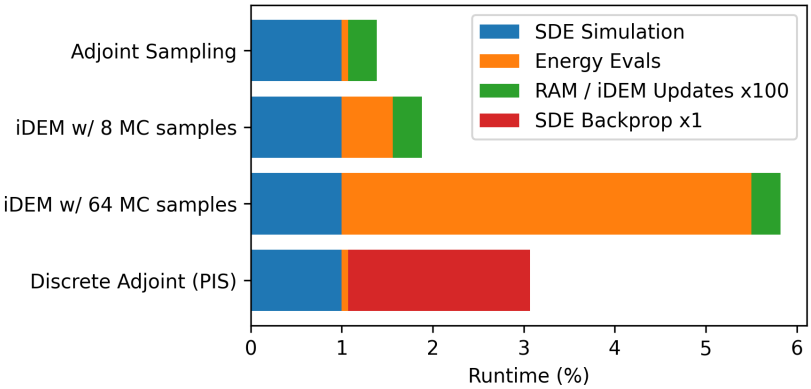


Figure 10: Conformer Generation Run-time: A run-time analysis breakdown of Adjoint Sampling compared to selected baselines at the level of gradient updates. Adjoint Sampling overcomes the two primary bottle-knecks of previous methods: SDE simulation and energy evaluation, allowing Adjoint Sampling to perform far more gradient updates in the same run-time.

J.3 Choice of energy function

Table 3: Test set MAE for eSEN-900k and MACE-OFF-23-M on the SPICE-MACE-OFF test dataset. Energy (**E**) MAE is in meV/atom. Force (**F**) MAE is in meV/Å.

	PubChem		DES370K M.		DES370K D.		Dipeptides		Sol. AA		Water		QMugs	
Model	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**	**E**	**F**
eSEN-900k	0.66	14.37	0.50	5.32	0.52	5.62	0.37	8.52	1.41	17.16	0.71	12.66	0.44	14.56
MACE-OFF-23-M	0.91	20.57	0.63	9.36	0.58	9.02	0.52	14.27	1.21	23.26	0.76	15.27	0.69	23.58

The eSEN energy model (Fu et al., 2025), used in this work, has been shown to accurately predict DFT energy and forces on the SPICE dataset. We use an eSEN network with 2 layers, $L_{\max} = 2, M_{\max} = 0$, 64 channels, and a radius cutoff of 4.5 Å. This model has around 900k trainable parameters. To demonstrate the robustness of the adjoint matching algorithm to the choice of the energy function, we conduct additional molecular conformation sampling experiments using adjoint matching on a pretrained MACE-OFF-23 energy model (Batatia et al., 2022; Kovács et al., 2023). The test-set prediction errors of eSEN-900k and the MACE-OFF-23-M (Kovács et al., 2023) model are reported in Table 3. The same energy model is used for both relaxation and for optimizing adjoint sampling, where applicable. Table 4 shows that adjoint matching with MACE-OFF-M and eSEN achieve similar performance in generating molecular conformation.

Table 4: Recall and precision metrics for conformer generation using different energy models. Coverage values are for thresholds of 1.25Å. Standard deviations are computed across molecules in the test set. We see eSEN performs slightly better at generalizing to GEOM.

		SPICE				GEOM-DRUGS			
		Recall		Precision		Recall		Precision	
	Method	Cov. ↑	AMR ↓	Cov. ↑	AMR ↓	Cov. ↑	AMR ↓	Cov. ↑	AMR ↓
	eSEN	80.28±27.68	0.96±0.28	47.97±32.24	1.25±0.37	61.36±38.74	1.17±0.60	41.19±35.93	1.46±0.67
	MACE	74.93±30.36	1.02±0.27	50.22±32.22	1.28±0.39	61.48±36.57	1.19±0.60	38.59±31.73	1.50±0.61
	eSEN (+pretrain)	88.58±19.96	0.85±0.25	65.12±29.90	1.13±0.34	73.67±35.26	0.97±0.57	52.17±36.30	1.25±0.62
	MACE (+pretrain)	89.62±19.25	0.84±0.25	67.07±29.48	1.11±0.34	71.06±35.17	1.01±0.57	53.92±37.07	1.25±0.61
w/ relaxation	eSEN	91.49±18.27	0.71±0.29	60.00±29.97	1.02±0.38	80.59±29.46	0.81±0.49	51.95±32.44	1.25±0.68
w/ relaxation	MACE	91.06±17.94	0.72±0.28	65.19±28.27	1.05±0.37	79.38±27.71	0.85±0.46	53.50±30.46	1.24±0.57
w/ relaxation	eSEN (+pretrain)	96.36± 8.92	0.60±0.25	76.46±26.22	0.92±0.35	87.32±23.89	0.70±0.45	68.02±31.86	1.01±0.55
w/ relaxation	MACE (+pretrain)	95.73±11.32	0.61±0.25	77.08±25.73	0.92±0.34	84.82±26.00	0.76±0.48	66.01±33.19	1.04±0.57

J.4 Ablating Reciprocal Projection

Here we present the same ablation of the Reciprocal Projection as performed for the Synthetic experiments in Table 2, but for the molecular conformer generation task. Recall that instead of using the Reciprocal projection (RP) to sample X_t given X_1, we simply store sample pairs (X_1, X_t) in the buffer and train on the Adjoint Matching objective equation 11. We call this *Adjoint Sampling w/o RP*, which helps demonstrate the effectiveness of the Reciprocal projection. For the purpose of this ablation, we only consider the MACE energy and do not perform any post generation relaxation.

Table 5: Recall and precision metrics for Adjoint Sampling with and without use of the Reciprocal Projection (RP). Coverage values are for thresholds of 1.25Å. Standard deviations are computed across molecules in the test set. We see that the Reciprocal Projection results in significant improvements across all metrics, especially when generalizing to the unseen dataset GEOM-DRUGS. No pretraining and no relaxation is performed post generation.

	SPICE				GEOM-DRUGS			
	Recall		Precision		Recall		Precision	
Method	Cov. ↑	AMR ↓	Cov. ↑	AMR ↓	Cov. ↑	AMR ↓	Cov. ↑	AMR ↓
Adjoint Sampling w/o RP (**Ablation**)	78.25±30.06	0.98±0.34	57.64±32.00	1.23±0.43	53.46±36.17	1.30±0.48	35.16±33.60	1.74±0.68
Adjoint Sampling (**Ours**)	80.28±27.68	0.96±0.28	47.97±32.24	1.25±0.37	61.36±38.74	1.17±0.60	41.19±35.93	1.46±0.67

J.5 Threshold Ablation for Molecular Conformation Generation

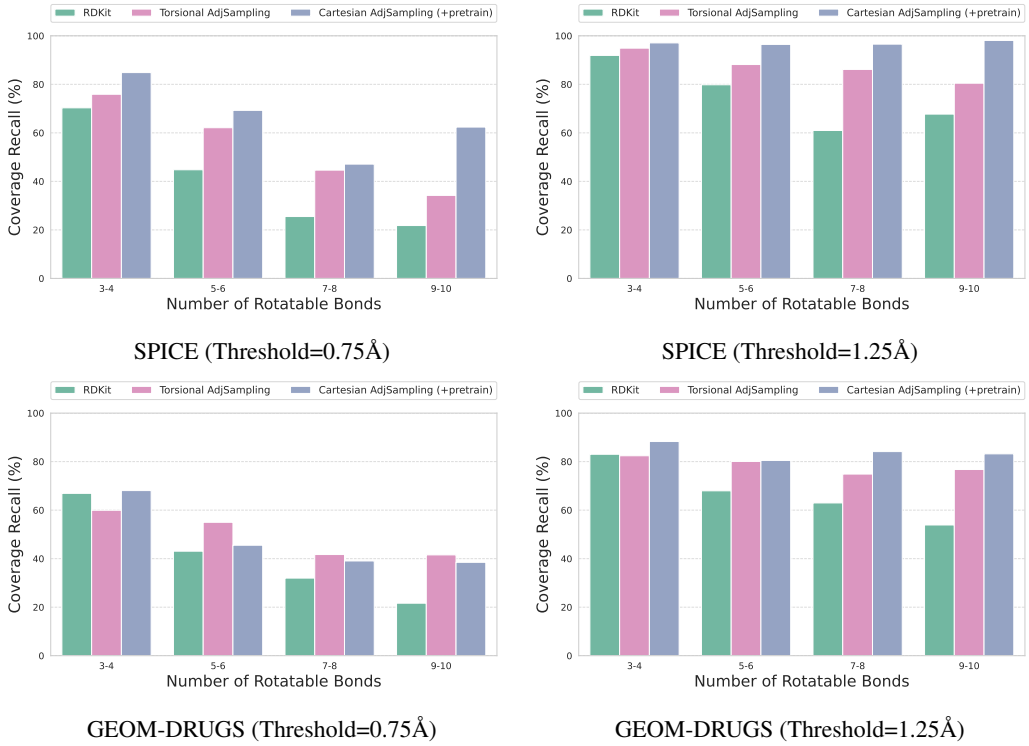


Figure 11: Coverage recall (%) after relaxation for molecules of different number of rotatable bonds. We see that the gap between our model and RDKit increases for higher number of rotatable bonds.