FROM POTENTIALS TO ENSEMBLES: BOLTZMANN SAMPLING FROM LEARNED ENERGY SURFACES VIA STOCHASTIC OPTIMAL CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Equilibrium ensembles governed by the Boltzmann distribution are fundamental to understanding the thermodynamics of ligand binding and conformational landscapes. While generative models have proven effective at sampling molecular configurations, they do not inherently recover the underlying physical energy landscape. Bridging this gap between empirical density estimation and physical thermodynamics remains an open challenge. We introduce ENTROPICA, a training framework that guides flow- and diffusion-based molecular generators toward equilibrium distributions determined by physical energies and temperature. ENTROPICA identifies the optimal transport path over cumulative informational and physical cost and anchors the predicted energy to the potential energy surface (PES). Supervision therefore comes from energies and forces rather than from sample density alone. We first validate the framework on a two-dimensional prototype system: trained on its PES alone, without access to sample data points, the model recovers both the terminal energy landscape and the Boltzmann distribution it defines, reproducing the reference data. Trained on molecules and density functional theory obtained energies and forces, ENTROPICA recovers the PES more accurately than a classical force field, while simultaneously learning to sample the associated equilibrium ensemble. Generated conformations lie in low-energy regions of the learned surface and the sampled density matches the Boltzmann statistics of molecular dynamics ground truth. ENTROPICA opens a route toward physics based molecular generation via PES guidance, with potential applications from equilibrium ensemble generation to scoring of molecular interactions.

1 INTRODUCTION

At equilibrium, a molecular system populates an energetic landscape according to the Boltzmann distribution, which minimizes the free energy $F = \langle u \rangle - \tau H$, balancing the internal energy u of its microstates against the entropy H of the ensemble, with τ denoting temperature. Hence, macroscopic properties must be inferred from the entire equilibrium ensemble rather than a single microstate, but this ensemble typically spans an intractably large number of configurations. Of particular interest is the difference in free-energy $\Delta\Delta F$ between two equilibrium ensembles, which quantifies the spontaneity of transitions such as drug–receptor binding or protein conformational switching. Although molecular dynamics can sample these ensembles in principle, convergence requires long simulations and states separated by high energy barriers are visited only rarely (Cournia et al., 2017).

Generative models offer a promising approach to overcome this bottleneck. Diffusion and flow matching models (Song et al., 2021; Lipman et al., 2022) learn to transform a Gaussian prior into the distribution of molecular conformations without traversing the barriers that separate different states (Lewis et al., 2025). These particular models learn the score $\nabla_{\mathbf{x}} \ln p_t$ of the density of their training data, which is a property of the empirical sample distribution rather than of the physical system itself. The model therefore reproduces distributional biases inherited from the training distribution.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

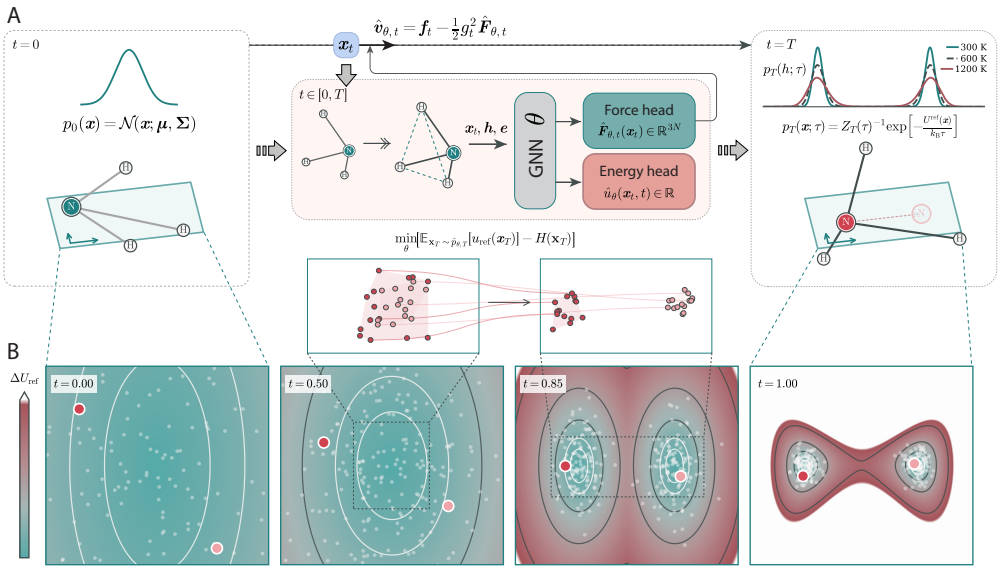


Figure 1: Overview of the ENTROPICA training framework. **(A)** Starting from a Gaussian prior $p_{\text{prior}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ at $t = 0$, a neural network processes noisy coordinates of a molecular graph with fixed atom and bond types. Output heads predict a reduced energy $\hat{u}_\theta(\boldsymbol{x}_t, t)$ and a vector field $\hat{\boldsymbol{F}}_{\theta,t}(\boldsymbol{x}_t)$ representing the score. Denoising score matching supervises the vector head to update the coordinates. At $t = T$, boundary losses anchor the predicted energy and force to physical reference energies. The target equilibrium density at temperature τ follows $p_T(\boldsymbol{x}; \tau) \propto \exp[-U^{\text{phys}}(\boldsymbol{x})/(k_{\text{B}}\tau)]$; the curves illustrate its temperature-dependent marginal. **(B)** The HJB loss leads to minimizing the expected terminal energy while maximizing entropy $\mathbb{E}_{\hat{p}_{\theta,T}}[u_{\text{ref}}] - H(\hat{p}_{\theta,T})$, where $u = U^{\text{phys}}/k_B\tau$. An ammonia nitrogen-displacement plane illustrates the transition from a smooth prior field to the two basins of the terminal potential energy surface.

To address this, we introduce ENTROPICA, an energy-based diffusion model whose network predicts a scalar energy $u_\theta(\mathbf{x}, t)$ at every point of configuration space and at every point in time. At terminal time this energy is bound to a reference PES and its forces, and minimizing the Kullback–Leibler (KL) divergence to the Boltzmann target yields a Hamilton–Jacobi–Bellman (HJB) residual that carries this physical anchor to every intermediate time. Instead of learning a negative log-density solely bound to the distribution of the training data, ENTROPICA instead learns a potential anchored to a quantum-mechanical reference and the transport toward its equilibrium ensemble (Fig. 1).

Contributions. Our main contributions in this work are the following:

- We derive a HJB training objective for flow- and diffusion-based generative models directly from the KL divergence to a Boltzmann target, phrasing it as an optimal control problem over cumulative informational and physical cost (Section 3.2), interpreting our model as Neural Partial Differential Equation (NeuralPDE) and impose physical boundary conditions that tie the terminal value function and score to a reference PES. Supervision then derives from physical energies and forces additionally than from sample density alone (Section 3.2).
- We show for a two-dimensional prototype that a model trained on a PES alone recovers the corresponding Boltzmann distribution and, when conditioned on temperature, accurately learns the PES across different temperatures (Section 4.1).
- We train a score-based, SE(3)-equivariant diffusion model on Density Functional Theory (DFT) energies and forces of small molecules together with the HJB objective, recovering the physical PES (to within 9–11 meV on rMD17) and the equilibrium mean energy and effective temperature, even when trained on data resampled to a higher temperature (Sections 4.2–4.3).
- Trained on the ANI-1x dataset, that forms no Boltzmann ensemble at any temperature, ENTROPICA generates conformers that concentrate far closer to the energetical minimum than denoising score matching trained on the same data, while its energy head reaches force-field-level accuracy on those molecules (Section 4.4).

2 Background and Related Work

Boltzmann generators aim to solve the problem of long simulation runs and insufficient conformational exploration by learning to sample according to the Boltzmann distribution directly. The prototype Boltzmann generator was constructed as a normalizing flow from a Gaussian prior to the target density, trained by minimizing the reverse KL divergence against a known energy function (Noé et al., 2019). Subsequent work has expanded this approach by flow matching in Cartesian coordinates, transferability across chemical space (Klein & Noé, 2024), temperature-conditioned variants (Dibak et al., 2022), temperature-annealing training regimes (Akhound-Sadegh et al., 2026) and autoregressive models (Rehman et al., 2026).

Diffusion models (Ho et al., 2020; Song et al., 2021) offer a powerful generative approach for high-dimensional molecular data, from structure prediction and protein design to docking and equilibrium ensembles (Abramson et al., 2024; Watson et al., 2023; Lewis et al., 2025). They define a stochastic process that gradually corrupts data with Gaussian noise and learn to invert it, guided at every step by the learned time-dependent score $\nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$, which is trained by denoising score matching. Flow matching (Lipman et al., 2022) instead regresses a deterministic velocity field along a prescribed probability path. Energy-based diffusion models combine both by learning a time-conditioned energy $U_\theta(\mathbf{x}, t)$ and obtaining the score as its negative gradient (LeCun et al., 2006; Song & Kingma, 2021). Trained on samples alone, however, the learned energy remains the negative log-density of the training ensemble rather than a physical potential (Arts et al., 2023).

Machine-learning interatomic potentials (MLIPs) are trained on a reference PES (typically obtained by DFT calculations) of a molecular system, regressing a scalar energy, from which the forces follow as $\mathbf{F} = -\nabla_{\mathbf{x}} U$ by automatic differentiation (Behler & Parrinello, 2007; Smith et al., 2017; Schütt et al., 2018). However, while being fast and accurate, sampling the corresponding equilibrium ensemble still requires an external simulator and therefore inherits the sampling problems of MD (Section 1).

The Hamilton–Jacobi–Bellman equation is the infinitesimal form of Bellman's principle of optimality and characterizes the value function of a deterministic or stochastic control problem (Bellman, 1957). Berner et al. (2022) obtain an optimal-control equation for diffusion models by Hopf–Cole transformation of the Fokker–Planck equation. The resulting terminal condition is the intractable marginal log-density of the inference process, which they note is unavailable in practice. Related transport formulations fix both endpoints, as in Schrödinger bridges (De Bortoli et al., 2021; Shi et al., 2023). Closest to our setting, Fokker–Planck consistency has been imposed on diffusion models as an auxiliary regularizer, at the level of the score (Lai et al., 2023) and of a learned energy for molecular systems (Plainer et al., 2026), motivated by a self-consistency property that trained models empirically violate.

3 Method

3.1 Problem Formulation

We seek to derive a physics-informed loss for generative models whose samples follow the Boltzmann distribution of a potential energy surface $U^{\mathrm{phys}}(\mathbf{x})$ at temperature τ,

$$p(\mathbf{x}) = Z^{-1} e^{-u(\mathbf{x})}, \qquad u(\mathbf{x}) = \frac{U^{\mathrm{phys}}(\mathbf{x})}{k_B \tau}, \tag{1}$$

where u is the dimensionless reduced potential, k_B the Boltzmann constant and Z the partition function. Throughout, U^{phys} denotes energies in physical units and u their reduced counterpart. Additionally, the model's predicted energy landscape for clean data should match the true physical PES U^{phys}, and its score match the gradient of the PES $-\nabla_{\mathbf{x}} u = -\nabla_{\mathbf{x}} U^{\mathrm{phys}} / k_B \tau$, which is the true reduced force.

3.2 THE HAMILTON–JACOBI–BELLMAN LOSS

We define a continuous-time transformation ϕ_t over the interval $t \in [0, T]$, where $t = 0$ corresponds to the Gaussian prior $p_{\text{prior}}(\mathbf{x}) = \mathcal{N}(\mathbf{x} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $t = T$ corresponds to the generated model distribution p_θ. In the flow matching paradigm (Lipman et al., 2022), the dynamics are governed by a time-dependent velocity field $\mathbf{v}_{\theta,t}(\mathbf{x}) = \frac{d}{dt}\phi_{\theta,t}(\mathbf{x})$; for diffusion models, the generative process is characterized by the score function $\mathbf{s}_{\theta,t}(\mathbf{x}) = \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$. Requiring the terminal density to be of the Boltzmann form, $p_T(\mathbf{x}) = Z_T^{-1} \exp(-u(\mathbf{x}))$, fixes the terminal score to the reduced force $\mathbf{F}(\mathbf{x}) = -\nabla_{\mathbf{x}} u(\mathbf{x})$,

$$\mathbf{s}_T(\mathbf{x}) = \nabla_{\mathbf{x}} \ln p_T(\mathbf{x}) = -\nabla_{\mathbf{x}} u(\mathbf{x}) = \mathbf{F}(\mathbf{x}), \tag{2}$$

so that the learned score field at terminal time coincides with the true conservative force field of the physical system.

Following Noé et al. (2019), we minimize the KL divergence between the Gaussian prior and the density $q_{0,\theta}$ obtained by evolving the target Boltzmann distribution backward along the model trajectory to $t = 0$, accounting for the change of volume induced by the flow. Integrating the instantaneous change of variables formula (Lipman et al., 2022) over $[0, T]$ and inserting the Boltzmann form of p_T yields

$$D_{\text{KL}}[p_{\text{prior}} \parallel q_{0,\theta}] = -H_{\text{prior}} + \ln Z_T + \int_{\mathbf{x}\sim p_{\text{prior}}} p_{\text{prior}}(\mathbf{x}) \left[u(\phi_{\theta,T}(\mathbf{x})) - \int_0^T \text{div}(\mathbf{v}_{\theta,t}(\phi_{\theta,t}(\mathbf{x}))) \, dt \right] d\mathbf{x}, \tag{3}$$

where H_{prior} is the entropy of the prior, u the reduced potential of equation 1 at terminal time T, and Z_T is the partition function at T. Neither H_{prior} nor $\ln Z_T$ depends on θ, so both drop out of the gradient and the intractable partition function never has to be evaluated.

The minimization of the loss function in equation 3 can be formally framed as a problem in optimal control over the interval $[0, T]$. We define the value function $V(\mathbf{x}, t)$ as the minimum cumulative cost to transition from a state $\mathbf{x}$ at time t to a terminal configuration. Specifically, the learning objective corresponds to seeking a set of model parameters θ that minimize

$$V(\mathbf{x}_0, 0) = \min_\theta \left\{ \int_0^T C(\phi_{\theta,t}(\mathbf{x}_0)) \, dt + D(\phi_{\theta,T}(\mathbf{x}_0)) \right\}, \tag{4}$$

where $\mathbf{x}_0$ is a sample from the prior distribution. In our specific physical context, we identify:

- The **cost rate function** $C(\phi_{\theta,t}(\mathbf{x}_0)) = -\text{div}(\mathbf{v}_{\theta,t}(\phi_{\theta,t}(\mathbf{x}_0)))$, which penalizes the local compression or expansion of the probability density along the trajectory.
- The **terminal cost function** $D(\phi_{\theta,T}(\mathbf{x}_0)) = u(\phi_{\theta,T}(\mathbf{x}_0))$, which enforces that the final states reside in low-energy regions of the PES.
- The **terminal condition** $V(\phi_T(\mathbf{x}_0), T) = u(\phi_T(\mathbf{x}_0))$, which relates to the true reduced potential.

We write $u_\theta(\mathbf{x}_t, t)$ for the parametric ansatz of this value function, i.e. the cumulative energetic and informational cost accrued from state $\mathbf{x}_t$ at time t to the terminal time. Applying Bellman's principle of optimality (Bellman, 1957) over an infinitesimal step dt, expanding to first order, and identifying $\partial u_{\theta,t}/\partial \mathbf{x} = -\mathbf{F}_{\theta,t}$ and $\partial \mathbf{x}_t/\partial t = \mathbf{v}_{\theta,t}$ gives, in the limit $dt \to 0$,

$$\frac{\partial u_\theta(\mathbf{x}_t, t)}{\partial t} - \mathbf{F}_{\theta,t}(\mathbf{x}_t)^\top \mathbf{v}_{\theta,t}(\mathbf{x}_t) - \text{div}(\mathbf{v}_{\theta,t}(\mathbf{x}_t)) = 0 \tag{5}$$

with the velocity $\mathbf{v}_t$ defined as

$$\begin{aligned} \mathbf{v}_{\theta,t}(\mathbf{x}_t) &= \mathbf{f}_t(\mathbf{x}_t) - \tfrac{1}{2} g_t^2 \nabla \ln p_{\theta,t}(\mathbf{x}_t) \\ &= \mathbf{f}_t(\mathbf{x}_t) - \tfrac{1}{2} g_t^2 \mathbf{F}_{\theta,t}(\mathbf{x}_t), \end{aligned} \tag{6}$$

(Song et al., 2021). The full derivation is given in Appendix A.1. This Hamilton–Jacobi–Bellman (HJB) partial differential equation (Bellman & Lee, 1978) characterizes the evolution of the value function along the optimal trajectory, linking the local flow dynamics to the global energy minimization objective. The learning problem it entails is that of identifying the optimal transport path minimizing the total physical and informational cost required to generate Boltzmann-aligned configurations.

We define the *HJB loss* as the expected absolute value of the residual of equation 5,

$$\mathcal{L}_{\mathrm{HJB}}(\theta) = \mathbb{E}_{t,\,\mathbf{x}_t}\left|\frac{\partial u_\theta(\mathbf{x}_t,t)}{\partial t} - \mathbf{F}_{\theta,t}(\mathbf{x}_t)^\top \mathbf{v}_{\theta,t}(\mathbf{x}_t) - \mathrm{div}\big(\mathbf{v}_{\theta,t}(\mathbf{x}_t)\big)\right|, \tag{7}$$

minimized over the model parameters θ.

Imposing the value $u_\theta(\cdot, T) = u$ and its spatial gradient $\mathbf{F}_{\theta,T} = -\nabla_{\mathbf{x}} u$ as terminal conditions pins the solution of the HJB equation to the physical PES.

3.3 Fokker–Planck Consistency

Proposition 1 (HJB–Fokker–Planck equivalence)**.** *The HJB residual of equation 5 equals, up to an additive function of t alone,*

$$\mathcal{R}_\theta(\mathbf{x},t) \;=\; \mathcal{L}[\mathbf{s}_{\theta,t}](\mathbf{x}) - \partial_t \ln p_{\theta,t}(\mathbf{x}), \quad \text{where} \quad \mathcal{L}[\mathbf{s}] := \tfrac{1}{2} g_t^2 \big(\mathrm{div}(\mathbf{s}) + \|\mathbf{s}\|_2^2\big) - \langle \mathbf{f}, \mathbf{s}\rangle - \mathrm{div}(\mathbf{f}). \tag{8}$$

Consequently $\nabla_{\mathbf{x}} \mathcal{R}_\theta = -\,\epsilon[\mathbf{s}_\theta]$ is the score Fokker–Planck residual of Lai et al. (2023), and $\mathcal{L}_{\mathrm{HJB}} = \|\mathcal{R}_\theta\|_1$ is its energy-level counterpart.

The proof is given in Appendix A.2. Since $\epsilon[\mathbf{s}_\theta]$ is the spatial gradient of $\mathcal{R}_\theta$, the two residuals differ only by a spatially constant function of time $\ln Z_t$, which cancelled from the KL objective in equation 3.

Fokker–Planck regularization as the local Boltzmann-generator objective. Existing work interprets the residual as a self-consistency property of the ground-truth score that trained models empirically violate, and adds it to a denoising score matching loss with a tunable weight (Lai et al., 2023). The same construction has been applied at the level of the energy (Plainer et al., 2026). By Proposition 1, the Fokker–Planck residual is the differential form of the Boltzmann-generator KL objective (Noé et al., 2019).

3.4 Parameterization and Training Objective

Our algorithms follow the flow matching time convention of Section 3.2 with $T = 1$ ($t = 0$: Gaussian prior, $t = 1$: clean data). We use a variance-exploding kernel $\mathbf{x}_t = \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, with drift and squared diffusion coefficient expressed in flow time, $\mathbf{f}_t \equiv \mathbf{0}$ and $g_t^2 = \mathrm{d}\sigma_t^2/\mathrm{d}t < 0$, where σ_t decreases from $\sigma_{\max}$ to $\sigma_{\min}$ along the generative direction. The negative sign of g_t^2 carries the time reversal, so that equation 6 and equation 5 hold in the generative direction.

The value function and its gradient are parameterized by one equivariant graph neural network, adapted from Schneuing et al. (2025), to act on a molecular graph with fixed atom and bond types. Conditioned on the flow time, it returns a graph-level energy $u_\theta(\mathbf{x}, t)$ and an equivariant vector field $\mathbf{F}_{\theta,t}(\mathbf{x})$ from two separate heads (Appendix A.3). Learning $\mathbf{F}_{\theta,t}$ explicitly rather than as $-\nabla_{\mathbf{x}} u_\theta$ spares every evaluation of the HJB residual a third derivative of u_θ; a consistency loss enforces $\mathbf{F}_{\theta,t} \approx -\nabla_{\mathbf{x}} u_\theta$ instead. For Sections 4.2–4.3 this network is trained on all objectives at once (Algorithm 3). For Section 4.4 it is split into a frozen score-pretrained branch and a trained correction branch. The pretrained and correction energies and forces are summed to a total energy and force respectively, as in equation 28.

Both heads are supervised by DFT energies and forces, normalized as is standard for MLIPs (Wood et al., 2026): per-element isolated-atom energies are removed by a least-squares fit, and the per-atom residual energies and the forces are standardized (Appendix A.4). Inverting this normalization gives physical energies and forces U_θ^{phys} and $\mathbf{F}_\theta^{\mathrm{phys}}$, whereas the score and the HJB residual operate in reduced units,

$$u_\theta = \frac{U_\theta^{\mathrm{phys}}}{k_B \tau}, \qquad \mathbf{F}_\theta = \frac{\mathbf{F}_\theta^{\mathrm{phys}}}{k_B \tau}. \tag{9}$$

Temperature enters only through this conversion. Sampling integrates the reverse dynamics of Algorithm 2 with $\mathbf{F}_\theta$ as the score, so the generated conformations target $p_\tau(\mathbf{x}) \propto \exp\big(-U_\theta^{\mathrm{phys}}(\mathbf{x})/(k_B\tau)\big)$ at whichever temperature τ is chosen.

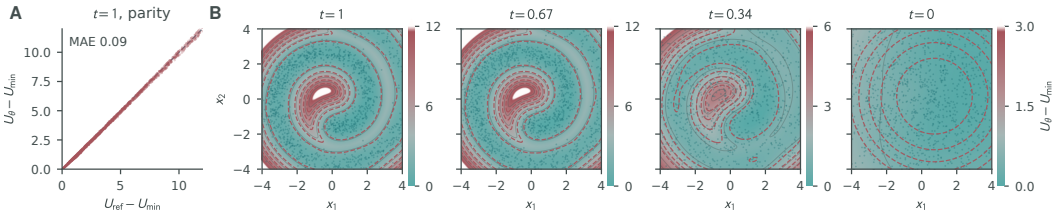


Figure 2: Recovery of the Boltzmann distribution from a PES alone on the two-dimensional Swiss-roll prototype. **(A)** Energy parity plot to the reference PES. The $t = 1$ PES is obtained through kernel density estimation of the reference Swiss-roll point cloud. **(B)** Intermediate t PES. Teal corresponds to lower energies and higher probabilities of finding a data point at this position. We show the learned PES $u_\theta(\mathbf{x}, t)$ alongside reference contour lines and generated samples.

The training objective combines the physical constraints with denoising score matching,

$$\mathcal{L} = \lambda_{\text{DSM}}\mathcal{L}_{\text{DSM}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}} + \lambda_U\mathcal{L}_U + \lambda_F\mathcal{L}_F + \lambda_{\nabla U}\mathcal{L}_{\nabla U} + \lambda_{\text{HJB}}\mathcal{L}_{\text{HJB}}. \tag{10}$$

The boundary terms act on clean frames at $t = 1$. $\mathcal{L}_U$ fits the energy to the DFT label. $\mathcal{L}_F$ and $\mathcal{L}_{\nabla U}$ fit the force head and the energy gradient to the DFT forces. $\mathcal{L}_{\text{HJB}}$ is the residual equation 40. $\mathcal{L}_{\text{DSM}}$ is denoising score matching on the force head and $\mathcal{L}_{\text{cons}}$ the consistency penalty between the two heads. All terms are defined in Appendix A.5. On rMD17 all terms are trained jointly from the start under the SOAP optimizer (Vyas et al., 2025), with a reduced HJB weight during the first epochs; on ANI-1x the score is pretrained and frozen, and a correction branch is trained on the physical terms (Appendix B.3).

4 Experiments

4.1 Recovering a Boltzmann Distribution from a Potential Alone

We first test on a two-dimensional prototype whether the PES alone suffices to recover its Boltzmann distribution (Fig. 2). A Swiss-roll point cloud defines a reference data distribution, from which we construct a PES by kernel density estimation (KDE) $u(\mathbf{x}) = -\ln p_{\text{KDE}}(\mathbf{x})$. EnTropica was trained solely on this PES, equipped with an energy and force head, supervised by the HJB-, energy- and consistency-loss (eqs. 40, 35, and 38), omitting supervision by denoising score matching loss. It aims to reproduce the PES and to generate samples following the ground-truth data distribution. For comparison, intermediate t smoothed PES were calculated from convolving the reference density with the Gaussian kernel of the noise schedule, $u_t = -\log\left(p_1 * \mathcal{N}(0, \sigma_t^2)\right)$ (Akhound-Sadegh et al., 2024). Algorithm 1 describes the training in this setting. Scaling the reference energy to an effective energy by the dimensionless temperature $u_\tau(\mathbf{x}) = u(\mathbf{x})/\tau$ and $p_\tau \propto e^{-u_\tau}$, the model learns temperature conditioned PES and samples data points accordingly. Sampling is conducted via numerically solving the reverse stochastic differential equation (SDE) (Algorithm 2, A.6). Because the target PES is obtained by KDE with a bandwidth larger than the intrinsic noise of the reference point cloud, the resulting PES is less sharply peaked, and the generated distribution is correspondingly more spread out than the reference data. Temperature-conditioned training and sampling at lower temperatures (Fig. 6, C.1) leads to a sharper distribution $p_\tau(\mathbf{x}) \propto \exp(\frac{-u(\mathbf{x})}{\tau})$. Sampling at higher temperatures leads in general to more spread out distributions. A flow matching implementation of the same task is given in C.2.

4.2 Comparison against Molecular Dynamics Ensembles on revised MD17

To test our setup on molecular systems, we trained the model on the structures, energies and forces of the revised MD17 (rMD17) dataset (Christensen & Von Lilienfeld, 2020). We chose three molecules that span the dataset's size range (9–24 atoms), exhibiting a torsion ϕ_1 whose density the sampled ensemble can be compared against, and with chemically distinct contributions to the PES: intramolecular O–H···O hydrogen bond for malonaldehyde, π-conjugation across the azo bridge with rotatable aryl torsions for azobenzene and hydrogen bonding between the carboxyl and phenolic hydroxyl groups for salicylic acid.

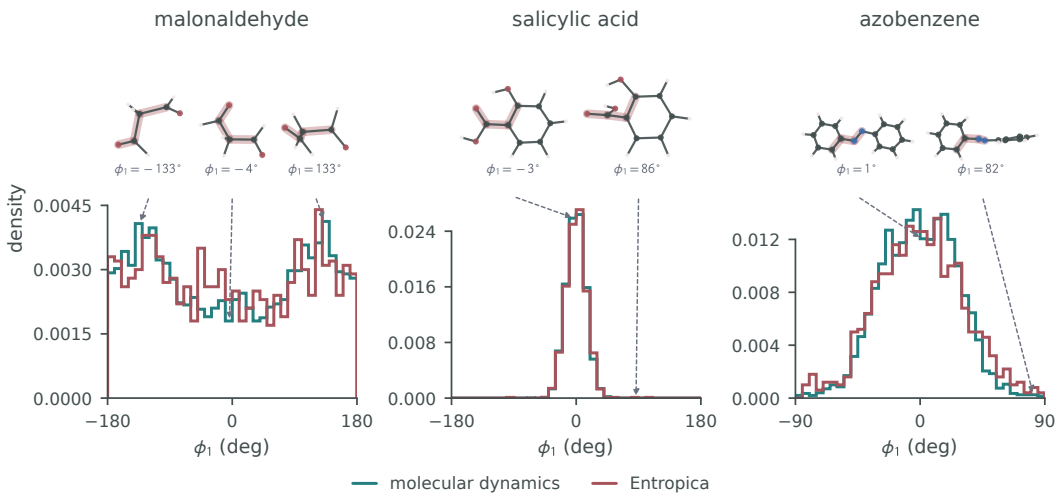


Figure 3: Sampled torsion densities on rMD17 for malonaldehyde, salicylic acid, and azobenzene. The torsion marginal $p(\phi_1)$ of the generated ensemble against the MD trajectory at $\tau = 500$ K. Samples are drawn by the reverse SDE (Algorithm 2), 1000 conformers per molecule against 4000 reference frames. Conformations corresponding to high- and low-likelyhood torsions are shown above.

Because configuration space in the molecular case is vastly larger than in the two-dimensional prototype, training on the HJB loss with PES boundary conditions alone caused the predicted vector field to collapse to trivial solutions. On Boltzmann distributed data, the denoising score matching (DSM) vector already approximates the target score, and we train it jointly with the HJB residual and the boundary conditions (equation 10, Algorithm 3). To account for the complex PES learning task, we utilize isolated-atomic-reference-energy and Z-Score normalization techniques as defined in Appendix A.4. Figure 3 reports for three molecules the generated torsion densities, matching the reference MD data. The corresponding Jensen-Shannon divergences alongside energy accuracy are reported in table 6, Appendix C.3, reaching a best range of 0.0077–0.0223 nats against a sampling floor of 0.0032–0.0127 nats, and centred energy MAEs of 9.5–11.4 meV across the three molecules, and corresponding 2D density plots along ϕ_1 and ϕ_2 are shown in figure 8, Appendix C.3.

4.3 Training without Equilibrium Data

The agreement with equilibrium MD in Fig. 3 leaves open whether the model learns the Boltzmann ensemble from the PES or reproduces the distribution of its training frames. To distinguish these possibilities, we resample the rMD17 azobenzene frames from the 500 K ensemble to a 600 K Boltzmann ensemble while keeping their geometries and DFT labels fixed. This raises the mean energy by 0.38 eV. We train Entropica on both datasets and sample every model at $\tau = 500$ K, aiming to recover the equilibrium at $\tau = 500$ K. The retained fraction is the change in sampled mean energy, as evaluated by a MACE model (Batatia et al., 2022) trained on 950 equilibrium DFT frames (Appendix B.4), divided by the imposed shift (Appendix C.4).

Score matching alone retains $95 \pm 3\,\%$ of the shift (Fig. 4, Table 7). Adding only the HJB residual, the Fokker–Planck regularization baseline (Plainer et al., 2026), retains $115 \pm 4\,\%$, and adding only the boundary conditions $104 \pm 3\,\%$. Entropica, with the HJB residual and the boundary conditions together, is the only model variant whose sampled mean energy and effective temperature return to the 500 K values, retaining $11 \pm 2\,\%$, whereas score matching reproduces the shifted training density. The resampled data also broaden the ϕ_1 marginal, by $\Delta 3.3°$ for score matching ($27.8° \to 31.1°$), $\Delta 4.5°$ for the Fokker-–Planck-regularized baseline ($29.1° \to 33.6°$), but only $\Delta 1.3°$ for Entropica ($32.1° \to 33.4°$), with the JS divergence to the 500 K data following the same order ($\Delta = 0.006, 0.011, 0.003$ nats). Entropica's torsion marginal is thus the least sensitive to the training density, though already broader than the reference on equilibrium data. The physical terms improve recovery of the true energetical distribution and improves robustness in Boltzmann distribution recovery but do not fully correct it (Appendix C.4).

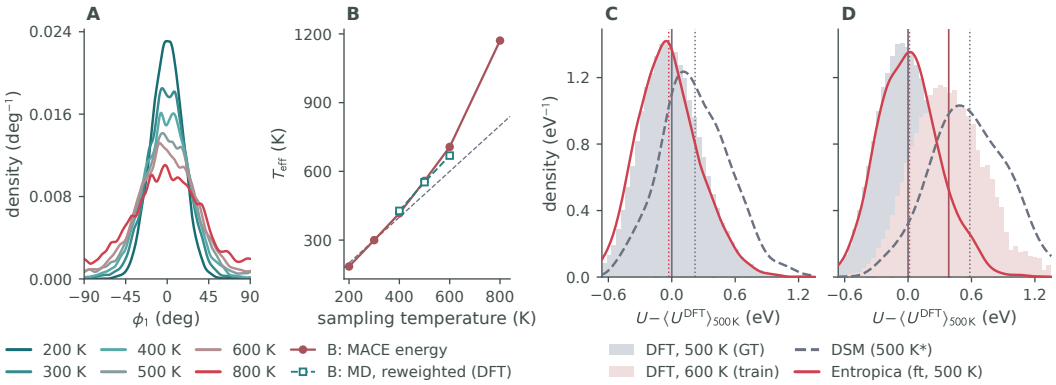


Figure 4: Temperature control and recovery of equilibrium on rMD17 azobenzene. Energies of generated conformations are evaluated by one MACE referee trained on equilibrium DFT frames. **(A)** Torsion ϕ_1 of a model trained at 500 K and sampled at 200–800 K. The sampling temperature τ enters through the effective force of the reverse SDE (equation 9, Algorithm 2). **(B)** Effective temperature of the sampled ensembles against the sampling temperature, from a harmonic density-of-states fit to the MACE referee energies (circles). Squares apply the same fit to DFT energies of the MD trajectory Boltzmann-reweighted to each temperature, and show the estimator's own offset. **(C)** Recovery from biased data: MACE referee energies of generated conformations on the 500 K frames and sampled at $\tau = 500$ K, over the DFT energies of those frames (grey). **(D)** The same models trained on the frames resampled to a 600 K Boltzmann ensemble (red), whose mean energy is 0.38 eV higher, and again sampled at 500 K. Solid vertical lines mark data means and dotted lines sampled means.

4.4 Simultaneous PES Learning and Sampling on ANI-1x

The azobenzene experiment tests whether physical supervision can counter a biased training distribution for one molecule. We next test whether the same model improves sampling of low energy conformations and learns DFT energies and forces across a diverse dataset of non-equilibrium structures of small molecules. ANI-1x (Smith et al., 2020) contains approximately five million conformers of small organic molecules, labelled with ωB97X/6-31G(d) energies and forces. These conformers do not form a Boltzmann ensemble at any single temperature. We split molecules by constitution and evaluate on 1,631 held-out molecules comprising 143,668 conformers (Appendix C.5).

On these molecules, the model achieves a per-molecule-centred energy mean absolute error of 222 meV (Table 8 and Fig. 5A). Fig. 5B relates the energy error to each held-out molecule's similarity to the training set. As a generator, the same network samples the held-out molecules at 300 K with an exponential energy tail. While the recovery of the approximate true Boltzmann ensemble could not be observed, the Entropica training setup leads to generated conformations much stronger distributed in low energy wells compared to standard DSM trained on the same data or the test-set conformers themselves (Fig. 5C).

5 Discussion and Limitations

On the two-dimensional prototype, the HJB residual with PES boundary conditions is sufficient to learn the smoothed intermediate variants of the PES and to sample the ground-truth distribution without any denoising score matching term. For high dimensional systems, in our case molecules, the PDE collapses to a trivial solution and needs further guidance via additional DSM loss as initial guess. Trained on molecular dynamics data, the generated densities by the HJB-PES guidance approach reproduce the ground truth MD densities and are comparably accurate with densities obtained by a DSM-fokker-planck-regularized model trained on the same data. PES guidance anchors the generative process to the physical PES, leading to a more robust model which is no longer relying solely on training data densities: When trained on artificially perturbed sample densities (resampling the training MD distribution at higher temperature), the HJB-PES guided model still reproduces the mean energy and effective temperature of the unperturbed reference ensemble while non-PES anchored models inherit this bias. We also discovered that all tested PDE-regularized models do

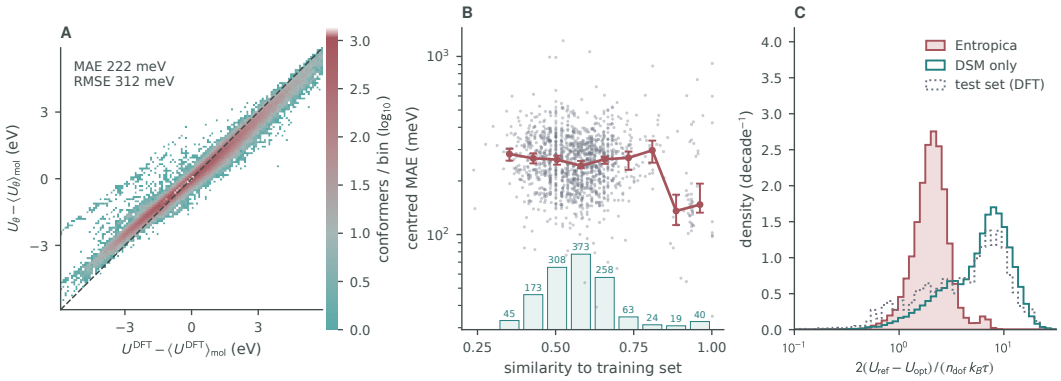


Figure 5: Energy prediction on 1,631 held-out ANI-1x molecules. **(A)** Predicted versus DFT energies, centred per molecule, for the 128,570 scored conformers (Table 8); color indicates the density of conformers. **(B)** Per-molecule centred energy MAE against the molecule's highest Tanimoto similarity to training molecules, line indicates median per similarity bin, with bootstrap 95% intervals, and each bin's molecule count. **(C)** Energies of 500 conformers generated for each of 64 held-out molecules by ENTROPICA, its DSM-only ablation, and of the test-set DFT conformers. Conformers that fail the geometry-validity check of Appendix B.5 are dropped (median 11 % per molecule for ENTROPICA and 4 % for DSM only). Valid conformers are scored by ANI-2x (E_{ref}), and reduced by each molecule's ANI-2x minimum E_{opt}, its vibrational degrees of freedom $n_{\mathrm{dof}} = 3N - 6$ and $k_B T$ at $T = 300$ K.

not uniformly outperform DSM-only baselines (not exclusively applying to the ENTROPICA set up). Satisfying a PDE residual is harder than regression onto a score since it requires backpropagation through derivatives of the network, admits trivial solutions, and has a more complex loss landscape in high dimensions (Krishnapriyan et al., 2021). As an interatomic potential at the clean boundary, ENTROPICA shows larger errors than domain-specific, state-of-the-art MLIPs, predicting DFT energies to within 9.5–11.4 meV on the evaluated rMD17 molecules, against 0.9–1.1 meV for MACE trained specifically on the same molecules which we attribute to the multi-objective and PDE learning task. While approximate equilibrium training data is still needed to recover the true Boltzmann distribution in sampling, and HJB-PES anchoring cannot reweight this strong imbalance in the training data, when trained on a dataset sampled far from equilibrium, such as ANI-1x, ENTROPICA generates conformations concentrated much more strongly in low-energy regions on the PES than pure denoising score matching.

6 CONCLUSION

We derived the HJB loss as differential form of the KL divergence Boltzmann generator loss (Noé et al., 2019) and connected it to the physical PES as boundary conditions. We showed that this is equivalent to the Fokker-Planck regularization propagating the target density defined by its energy through the intermediate time steps, extending its use case beyond density consistency regularization. Our approach yields a generative model anchored to a DFT-level PES, reproducing equilibrium distributions accurately and more robust than score matching and Fokker–Planck regularization. Furthermore, our model can be interpreted as an MLIP for time-smoothed variants of the PES, recovering the physical PES at the clean-data boundary. This could allow the sampling density to be trained on cheap approximate ensembles, such as force-field MD, while the generative model remains anchored to a high-resolution PES. Beyond sampling, our formulation gives direct access to thermodynamic quantities: the average internal energy as the expectation of the predicted potential energy over generated samples and the entropy from the expected model log-density, obtained by integrating the divergence of the velocity field along the probability-flow ODE. Together they give access not only to free energies but to their decomposition into energetic and entropic contributions, helping to answer the question whether a conformational change or a binding is enthalpy- or entropy-driven.

AI USE STATEMENT

In this work, we used generative AI tools for cleaning and reformatting datasets, providing feedback on research methodology and experiments, implementing methods, assisting with translation and supporting qualitative and thematic data analysis. In the formulation of mathematical claims, providing critical ingredients for proving mathematical claims and assisting in the writing of proofs, AI was exclusively applied for Appendix A.2.

Additionally, we used generative AI tools for creating and editing software code, sourcing and searching for information, identifying relevant literature, formatting references and editing the research paper to improve readability. Furthermore, we used AI tools to help typeset and phrase our equations.

We have not used generative AI tools for generating synthetic data sets, developing theoretical models or conceptual frameworks, proposing or refining hypotheses or interpreting results.

We have reviewed all AI-assisted work. Specifically, all AI-implemented code and data cleaning scripts were manually audited and tested for correctness by the authors. AI-identified literature and formatted references were individually verified to ensure relevance and prevent hallucinations and AI-assisted methodological feedback was critically evaluated against existing literature before adoption. Furthermore, all AI-typeset equations were strictly cross-checked against our manual derivations to ensure fidelity to our manually-derived math.

We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The derivation of the HJB objective from the Boltzmann-generator KL divergence is given in full in Appendix A.1, and the proof of Proposition 1 in Appendix A.2. The network architecture, diffusion schedule and optimization hyperparameters are given in Appendix A.3 and Tables 2–5, the normalization of DFT energies and forces in Appendix A.4, all loss terms in Appendix A.5, pseudocode for training and sampling in Appendix A.6. All molecular experiments use the publicly available rMD17 and ANI-1x datasets. The ANI-1x split and the resampling procedure of the non-equilibrium experiment are described in appendices C.5 and C.4. Code and trained models will be released upon publication.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Yim, Jiaming Zhang, Meg Bileschi, Tom Spring, Alexander Brik, Michael Bodo, Augustin Zídek, Joseph Cahill, Ioana Bica, Katharina Tunyasuvunakool, Aaron Senter, Ross Murchison, Russell Koh, John M. Jumper, and Demis Hassabis. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016): 493–500, 2024. doi: 10.1038/s41586-024-07487-w.

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Joey Bose, Jackson Rector-Brooks, Alexander Tong, Marcin Sendera, Leo Rector-Brooks, Nikolay Malkin, Pablo Lemos, Max Rector-Brooks, Yoshua Bengio, and Alex Hernández-García. Iterated denoising energy matching for sampling from Boltzmann densities. *arXiv preprint arXiv:2402.06121*, 2024.

Tara Akhound-Sadegh, Jungyoon Lee, Avishek Joey Bose, Valentin De Bortoli, Arnaud Doucet, Michael M. Bronstein, Dominique Beaini, Siamak Ravanbakhsh, Kirill Neklyudov, and Alexander Tong. Progressive inference-time annealing of diffusion models for sampling from Boltzmann densities. In *Advances in Neural Information Processing Systems*, volume 38, pp. 63218–63251, 2026.

Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zügner, Marco Le, Frank Noé, Victor Garcia Satorras, Michiel Hermans, and Max Welling. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, 2023.

Ilyes Batatia, David P. Kovacs, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11423–11436, 2022.

Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters*, 98(14):146401, 2007.

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

Richard Bellman and E. Stanley Lee. Functional equations in dynamic programming. *Aequationes Mathematicae*, 17(1):1–18, 1978. doi: 10.1007/BF01818535.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *arXiv preprint arXiv:2211.01364*, 2022.

Anders S. Christensen and O. Anatole Von Lilienfeld. On the role of gradients for machine learning of molecular energies and forces. *Machine Learning: Science and Technology*, 1(4):045018, 2020.

Zoe Cournia, Bryce Allen, and Woody Sherman. Relative binding free energy calculations in drug discovery: recent advances and practical considerations. *Journal of Chemical Information and Modeling*, 57(12):2911–2937, 2017.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17695–17709, 2021.

Christian Devereux, Justin S Smith, Kate K Huddleston, Kipton Barros, Roman Zubatyuk, Olexandr Isayev, and Adrian E Roitberg. Extending the applicability of the ani deep learning molecular potential to sulfur and halogens. *Journal of chemical theory and computation*, 16(7):4192–4202, 2020.

Manuel Dibak, Leon Klein, Andreas Krämer, and Frank Noé. Temperature steerable flows and Boltzmann generators. *Physical Review Research*, 4(4):L042005, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

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

Ilia Igashov, Arne Schneuing, Adrian W Dobbelstein, Irina Morozova, Rebecca M Neeser, Kara Zielinski, Luciano Andres Abriata, Aaron S Petruzzella, David R Pavel Iosub, Olivia Gampp, et al. A unified 3d generative model for synthesizable structure-based drug design. *bioRxiv*, pp. 2026–09, 2026.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=1YLJDvSx6J4`.

Leon Klein and Frank Noé. Transferable boltzmann generators. In *Advances in Neural Information Processing Systems*, volume 37, pp. 45281–45314, 2024.

Aditi S. Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M. Kirby, and Michael W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26548–26560, 2021.

Chieh-Hsin Lai, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. FP-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18365–18398. PMLR, 2023.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. In Gökhan Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, and Ben Taskar (eds.), *Predicting Structured Data*, chapter 1. MIT Press, 2006.

S. Lewis, T. Hempel, J. Jiménez-Luna, M. Gastegger, Y. Xie, A. Y. Foong, V. G. Satorras, O. Abdin, B. S. Veeling, I. Zaporozhets, Y. Chen, S. Yang, A. E. Foster, A. Schneuing, J. Nigam, F. Barbero, S. Vincent, A. Campbell, J. Yim, M. Lienen, Y. Shi, S. Zheng, H. Schulz, U. Munir, R. Sordillo, R. Tomioka, C. Clementi, and F. Noé. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, pp. eadv9817, 2025.

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

Jun S Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and computing*, 6(2):113–119, 1996.

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.

Michael Plainer, Hao Wu, Leon Klein, Stephan Günnemann, and Frank Noé. Consistent sampling and simulation: Molecular dynamics with energy-based diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 24460–24505, 2026.

Danyal Rehman, Charlie B. Tan, Yoshua Bengio, Avishek Joey Bose, and Alexander Tong. Autoregressive boltzmann generators. *arXiv preprint arXiv:2606.27361*, 2026.

Arne Schneuing, Ilia Igashov, Adrian W. Dobbelstein, Thomas Castiglione, Michael M. Bronstein, and Bruno Correia. Multi-domain distribution learning for de novo drug design. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=g3VCIM94ke.

Kristof T. Schütt, Huziel E. Sauceda, Pieter-Jan Kindermans, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet – A deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger bridge matching. In *Advances in Neural Information Processing Systems*, volume 36, pp. 62183–62223, 2023.

Justin S. Smith, Olexandr Isayev, and Adrian E. Roitberg. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chemical Science*, 8(4):3192–3203, 2017.

Justin S. Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian E. Roitberg, Olexandr Isayev, and Sergei Tretiak. The ANI-1ccx and ANI-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific Data*, 7(1):134, May 2020. ISSN 2052-4463. doi: 10.1038/s41597-020-0473-z. URL https://doi.org/10.1038/s41597-020-0473-z.

Yang Song and Diederik P. Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

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*, 2021.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M. Kakade. SOAP: Improving and stabilizing shampoo using adam for language modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IDxZhXrpNf.

Joseph L. Watson, David Juergens, Nathaniel R. Bennett, Brian L. Trippe, Jason Yim, Helen E. Eisenach, Woody Ahern, Andrew J. Borst, Robert J. Ragotte, Lukas F. Milles, Basile I. M. Wicky, Niki Hanikel, Samuel J. Pellock, Alexis Courbet, William Sheffler, Jue Wang, Preetham Venkatesh, Isaac Sinding, Siguird Vázquez Torres, Philip Borst, Nathan De Witt, Sanjib Bera, Neil P. King, and David Baker. De novo design of protein structure and function with RFdiffusion. *Nature*, 620 (7976):1089–1100, 2023. doi: 10.1038/s41586-023-06415-8.

Brandon Wood, Yi-Lun Liao, Abhishek Das, Tess Smidt, et al. UMA: A family of universal models for atoms. In *Advances in Neural Information Processing Systems*, volume 38, pp. 129391–129427, 2026.

A Extended methods

A.1 Full Derivation of the HJB Objective

This appendix gives in full the derivation summarized in Section 3.2: from the KL divergence to the Boltzmann target, through the optimal control formulation, to the HJB equation equation 5 whose residual defines the training loss equation 40.

From the KL Divergence to the Control Objective We define a continuous-time transformation ϕ_t over the interval $t \in [0, T]$, where $t = 0$ corresponds to the Gaussian prior $p_{\text{prior}}(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $t = T$ corresponds to the generated model distribution p_θ. In the flow matching paradigm, the dynamics are governed by a time-dependent velocity field $\mathbf{v}_t(\mathbf{x}) = \frac{d}{dt}\phi_t(\mathbf{x})$. For diffusion models, the generative process is characterized by the score function $\mathbf{s}_t(\mathbf{x}) = \nabla_{\mathbf{x}} \ln p_t(\mathbf{x})$.

Under the assumption of physical consistency, we align the score of the target distribution at $t = T$ with the equilibrium properties of the system. Given the Boltzmann distribution $p_T(\mathbf{x}) = Z_T^{-1} \exp(-u(\mathbf{x}))$, the score is derived as

$$\nabla_{\mathbf{x}} \ln p_T(\mathbf{x}) = \nabla_{\mathbf{x}} \left(-u(\mathbf{x}) - \ln Z_T\right) = -\nabla_{\mathbf{x}} u(\mathbf{x}). \tag{11}$$

Since the physical force $\mathbf{F}(\mathbf{x})$ is defined as the negative gradient of the PES, we obtain the direct correspondence $\mathbf{s}_T(\mathbf{x}) = \mathbf{F}(\mathbf{x})$ of equation 2. This identity ensures that the generative model's learned score field at the terminal time coincides with the true conservative force field of the physical system.

The objective is to minimize the Kullback–Leibler (KL) divergence between the distribution resulting from the backward evolution of the target density through the model's change of volume of the distribution (Noé et al., 2019). We consider the density $q_{0,\theta}(\mathbf{x})$ produced by evolving the target Boltzmann distribution $p_T(\mathbf{x})$ backward along the trajectory to $t = 0$, accounting for the change in volume induced by the model's flow. The objective is to minimize the divergence between this backward-evolved density and the Gaussian prior:

$$\begin{aligned} D_{\text{KL}}[p_{\text{prior}}(\mathbf{x}) \parallel q_{0,\theta}(\mathbf{x})] &= \int_{\mathbf{x}\sim p_{\text{prior}}} p_{\text{prior}}(\mathbf{x}) \left[\ln p_{\text{prior}}(\mathbf{x}) - \ln q_{0,\theta}(\mathbf{x})\right] d\mathbf{x} \\ &= -H_{\text{prior}} - \int_{\mathbf{x}\sim p_{\text{prior}}} p_{\text{prior}}(\mathbf{x}) \ln q_{0,\theta}(\phi_0(\mathbf{x})) d\mathbf{x}, \end{aligned} \tag{12}$$

where H_{prior} denotes the entropy of the Gaussian prior distribution.

Integrating the instantaneous change of variables formula over the trajectory $[0, T]$ gives

$$\ln p_\theta(\phi_T(\mathbf{x})) - \ln q_{0,\theta}(\phi_0(\mathbf{x})) = -\int_0^T \text{div}(\mathbf{v}_t(\phi_t(\mathbf{x}))) \, dt, \tag{13}$$

where $\mathbf{v}_t$ represents the velocity field estimated by the model (Lipman et al., 2022). Substituting this relation into equation 12, we obtain the expanded objective

$$D_{\text{KL}} = -H_{\text{prior}} - \int_{\mathbf{x}\sim p_{\text{prior}}} p_{\text{prior}}(\mathbf{x}) \left[\ln p_\theta(\phi_T(\mathbf{x})) + \int_0^T \text{div}(\mathbf{v}_t(\phi_t(\mathbf{x}))) dt\right] d\mathbf{x}. \tag{14}$$

By substituting $p_\theta(\phi_T(\mathbf{x}))$ with the target Boltzmann distribution $p_T(\mathbf{x}) = Z_T^{-1} \exp(-u(\phi_T(\mathbf{x})))$, we arrive at the following formulation:

$$\begin{aligned} D_{\text{KL}} &= -H_{\text{prior}} - \int_{\mathbf{x}\sim p_{\text{prior}}} p_{\text{prior}}(\mathbf{x}) \left[-u(\phi_T(\mathbf{x})) - \ln Z_T + \int_0^T \text{div}(\mathbf{v}_t(\phi_t(\mathbf{x}))) dt\right] d\mathbf{x} \\ &= -H_{\text{prior}} + \ln Z_T + \int_{\mathbf{x}\sim p_{\text{prior}}} p_{\text{prior}}(\mathbf{x}) \left[u(\phi_T(\mathbf{x})) - \int_0^T \text{div}(\mathbf{v}_t(\phi_t(\mathbf{x}))) dt\right] d\mathbf{x}, \end{aligned} \tag{15}$$

which is equation 3 of the main text. Here u denotes the dimensionless effective energy (reduced potential) defined as $u_{\text{physical}} \cdot (k_B \tau)^{-1}$ at terminal time T, and Z_T represents the partition function of the configuration space at T. Both H_{prior} and $\ln Z_T$ are independent of θ and therefore vanish under differentiation; the intractable partition function never enters the training signal.

Optimal Control Formulation Inserting the cost rate C, the terminal cost D and the terminal condition identified in Section 3.2 into the value function equation 4, and writing $u_\theta(\mathbf{x}_t, t)$ for its parametric ansatz, gives the explicit form

$$u_\theta(\mathbf{x}_0, 0) = \min_\theta \left(u(\phi_T(\mathbf{x}_0)) - \int_0^T \operatorname{div}(\mathbf{v}_t(\phi_t(\mathbf{x}_0)))dt \right), \tag{16}$$

so that $u_\theta(\mathbf{x}_t, t)$ represents the cumulative total (energetic and informational) cost at time t.

Principle of Optimality and the HJB Equation In the following step we apply the principle of optimality (Bellman, 1957), which states that optimality of the whole trajectory is reached by optimality for a time step dt and optimality of the rest of the trajectory:

$$u_\theta(\mathbf{x}_t, t) = \min_\theta \left(u_\theta(\mathbf{x}_{t+dt}, t+dt) - \int_t^{t+dt} \operatorname{div}(\mathbf{v}_s(\phi_s(\mathbf{x}_t)))\, ds \right), \tag{17}$$

where $u_\theta(\mathbf{x}_{t+dt}, t+dt)$ represents the remaining total cost from time $t+dt$ to the terminal time T.

The first term on the right-hand side can be evolved using a Taylor expansion:

$$u_\theta(\mathbf{x}_{t+dt}, t+dt) = u_\theta(\mathbf{x}_t, t) + \frac{\partial u_\theta(\mathbf{x}_t, t)}{\partial t} dt + \frac{\partial u_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}_t}{\partial t} dt + \mathcal{O}(dt^2). \tag{18}$$

Substituting equation 18 in equation 17 and replacing $\frac{\partial u_\theta(\mathbf{x}_t, t)}{\partial \mathbf{x}}$ by $-\mathbf{F}_{\theta,t}(\mathbf{x}_t)$ and $\frac{\partial \mathbf{x}_t}{\partial t}$ by $\mathbf{v}_t(\mathbf{x}_t)$, we obtain

$$u_\theta(\mathbf{x}_t, t) = \min_\theta \left(u_\theta(\mathbf{x}_t, t) + \frac{\partial u_\theta(\mathbf{x}_t, t)}{\partial t} dt - \mathbf{F}_{\theta,t}(\mathbf{x}_t)^\top \mathbf{v}_t(\mathbf{x}_t)\, dt + \mathcal{O}(dt^2) - \int_t^{t+dt} \operatorname{div}(\mathbf{v}_s(\phi_s(\mathbf{x}_t)))\, ds \right). \tag{19}$$

By subtracting $u_\theta(\mathbf{x}_t, t)$ from both sides and dividing by dt, we obtain equation 5,

$$\frac{\partial u_\theta(\mathbf{x}_t, t)}{\partial t} - \mathbf{F}_{\theta,t}(\mathbf{x}_t)^\top \mathbf{v}_t(\mathbf{x}_t) - \operatorname{div}(\mathbf{v}_t(\mathbf{x}_t)) = 0. \tag{20}$$

This Hamilton-Jacobi-Bellman PDE (Bellman & Lee, 1978) characterizes the evolution of the value function along the optimal trajectory, linking the local flow dynamics to the global energy minimization objective. The HJB loss in equation 40 is the norm of its residual, minimized over the model parameters θ.

A.2 Proof of Proposition 1

The HJB equation of equation 5 reads

$$\partial_t u_\theta(\mathbf{x}, t) - \mathbf{F}_{\theta,t}(\mathbf{x})^\top \mathbf{v}_{\theta,t}(\mathbf{x}) - \operatorname{div}\big(\mathbf{v}_{\theta,t}(\mathbf{x})\big) = 0, \tag{21}$$

the velocity field of equation 6 is $\mathbf{v}_{\theta,t} = \mathbf{f}_t - \frac{1}{2} g^2 \mathbf{F}_{\theta,t}$, and the energy and force are defined as

$$u_\theta(\mathbf{x}, t) = -\ln p_{\theta,t}(\mathbf{x}) + c(t), \qquad \mathbf{F}_{\theta,t} = -\nabla_{\mathbf{x}} u_\theta = \nabla_{\mathbf{x}} \ln p_{\theta,t} = \mathbf{s}_{\theta,t}, \tag{22}$$

with $c(t) = -\ln Z_t$.

From equation 22,

$$\partial_t u_\theta = -\partial_t \ln p_{\theta,t} + \dot{c}(t). \tag{23}$$

Using $\mathbf{F}_{\theta,t} = \mathbf{s}_{\theta,t}$ and the velocity field,

$$\mathbf{F}_{\theta,t}^\top \mathbf{v}_{\theta,t} = \big\langle \mathbf{s}_{\theta,t},\, \mathbf{f}_t - \tfrac{1}{2} g^2 \mathbf{s}_{\theta,t} \big\rangle = \langle \mathbf{f}_t, \mathbf{s}_{\theta,t} \rangle - \tfrac{1}{2} g^2 \|\mathbf{s}_{\theta,t}\|_2^2, \tag{24}$$

and, since g does not depend on $\mathbf{x}$,

$$\operatorname{div}(\mathbf{v}_{\theta,t}) = \operatorname{div}(\mathbf{f}_t) - \tfrac{1}{2} g^2 \operatorname{div}(\mathbf{s}_{\theta,t}). \tag{25}$$

Substituting the three expressions into equation 21 and grouping the terms that depend on $\mathbf{s}_{\theta,t}$,

$$\begin{aligned} &-\partial_t \ln p_{\theta,t} + \dot{c}(t) - \langle \mathbf{f}_t, \mathbf{s}_{\theta,t} \rangle + \tfrac{1}{2} g^2 \|\mathbf{s}_{\theta,t}\|_2^2 - \operatorname{div}(\mathbf{f}_t) + \tfrac{1}{2} g^2 \operatorname{div}(\mathbf{s}_{\theta,t}) \\ &\quad = \underbrace{\tfrac{1}{2} g^2 \big(\operatorname{div}(\mathbf{s}_{\theta,t}) + \|\mathbf{s}_{\theta,t}\|_2^2\big) - \langle \mathbf{f}_t, \mathbf{s}_{\theta,t} \rangle - \operatorname{div}(\mathbf{f}_t)}_{= \mathcal{L}[\mathbf{s}_{\theta,t}]} - \partial_t \ln p_{\theta,t} \; + \; \dot{c}(t) \\ &\quad = \mathcal{R}_\theta(\mathbf{x}, t) + \dot{c}(t), \end{aligned} \tag{26}$$

which is equation 8, up to the spatially constant function of time $\dot{c}(t)$. $\mathcal{L}$ is exactly the one appearing in equation (7) of Lai et al. (2023), so $\mathcal{R}_\theta \equiv 0$ is the log-Fokker–Planck equation $\partial_t \ln p_{\theta,t} = \mathcal{L}[\mathbf{s}_{\theta,t}]$. The HJB residual and $\mathcal{R}_\theta$ therefore agree modulo $\dot{c}(t)$, an additive function of time alone.

Applying $\nabla_{\mathbf{x}}$ to $\mathcal{R}_\theta$ annihilates $\dot{c}(t)$ and, by the commutation assumption,

$$\nabla_{\mathbf{x}} \mathcal{R}_\theta = \nabla_{\mathbf{x}} \mathcal{L}[\mathbf{s}_{\theta,t}] - \partial_t \nabla_{\mathbf{x}} \ln p_{\theta,t} = \nabla_{\mathbf{x}} \mathcal{L}[\mathbf{s}_{\theta,t}] - \partial_t \mathbf{s}_{\theta,t} = -\epsilon[\mathbf{s}_\theta], \tag{27}$$

with $\epsilon[\mathbf{s}_\theta] := \partial_t \mathbf{s}_\theta - \nabla_{\mathbf{x}} \mathcal{L}[\mathbf{s}_\theta]$ the score Fokker–Planck residual of Lai et al. (2023), their equation (9).

Remark. *The interchange $\partial_t \nabla_{\mathbf{x}} = \nabla_{\mathbf{x}} \partial_t$ is Schwarz's theorem. Lai et al. (2023) invoke the same step for the ground-truth density between their Eqs. (7) and (8).*

A.3 Network architecture

A single GVP-GNN backbone (Jing et al., 2021) adapted from DrugFlow (Schneuing et al., 2025) produces both the energy and the force head. We discard the heterogeneous pocket and residue node and edge types and the atom- and bond-type denoising of DrugFlow, keeping a single ligand node type on a molecular graph with fixed atom and bond types. Its coordinate-velocity readout is replaced by the two physical heads described below.

Inputs and featurization. A molecule is a graph of N atoms with atomic numbers in $\{\mathrm{H}, \mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{F}\}$ (one-hot) and coordinates $\mathbf{x} \in \mathbb{R}^{N \times 3}$. Atom types are embedded by a two-layer SiLU MLP into 128 scalar node features; nodes carry no input vector features, so geometry enters only through the edges. The flow time t is mapped by a sinusoidal embedding (dimension 64) and a two-layer SiLU MLP and concatenated to the node scalars.

Graph and edge features. Message passing runs on the covalent-bond graph: the molecule's bonds form the (bidirectional) edges and their bond-type one-hot encoding is used as edge scalar features. Each edge additionally carries a radial-basis expansion of its interatomic distance (16 Gaussians on $[0, d_{\max}]$, $d_{\max} = 50\,\text{Å}$) and the unit direction $\hat{\mathbf{r}}_{ij}$ as an edge vector feature, together with a normalized cross-product feature so that enantiomers are distinguished.

Energy and force heads. The energy head is a two-layer SiLU MLP applied to each atom's SO(3)-invariant scalar features, whose per-atom outputs are averaged over the molecule to give the SO(3)-invariant graph-level scalar $u_\theta(\mathbf{x}, t) \in \mathbb{R}$, while the force head returns the SO(3)-equivariant vector $\mathbf{F}_{\theta,t}(\mathbf{x})_i = \mathbf{u}_i$. Both raw head outputs are divided by $\max(\sigma_t, \sigma_{\min})$ (Section 3.4). Conservativity, $\mathbf{F}_\theta \approx -\nabla_{\mathbf{x}} u_\theta$, is imposed by the consistency penalty equation 38 rather than by construction, as this would require triple differentiation through the energy head.

Diffusion process and optimization. Coordinates follow the variance-exploding perturbation kernel $\mathbf{x}_t = \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We use generative time, with $t = 0$ at the Gaussian prior and $t = 1$ at clean data, so σ_t decreases from $\sigma_{\max} = 5.0$ to $\sigma_{\min} = 0.01$ over 1000 steps. In the velocity and HJB equations, g_t^2 denotes the signed flow-time coefficient $\mathrm{d}\sigma_t^2/\mathrm{d}t < 0$, rather than the nonnegative squared diffusion amplitude of the forward noising process. The VE drift is $\mathbf{f}_t \equiv \mathbf{0}$, and the sign of g_t^2 accounts for the reversal from noising time to generative time. Atom types and bonds are held fixed.

Hyperparameters of every run are listed in Tables 2–5 (Appendix B.3).

Composition of the two branches. As an alternative to single-branch training, the backbone above can be instantiated twice: a score-pretrained branch is frozen and a correction branch is trained on

Table 1: Backbone and training hyperparameters for the single-branch energy–force model.

Parameter	Value
Backbone (GVP-GNN)	
GVP convolution layers L	3
Node hidden dim (scalar, vector)	$(64,\ 32)$
Edge hidden dim (scalar, vector)	$(64,\ 32)$
Scalar node feature width	128
Scalar activation / vector gating	SiLU / yes
Dropout	0.0
Time embedding	sinusoidal, dim 64
Distance RBFs / $d_{\max}$	16 / 50 Å
Edges	covalent bonds, bond-type one-hot
Atom types	H, C, N, O, F
Diffusion (variance exploding)	
$\sigma_{\min}$ / $\sigma_{\max}$	0.01 / 5.0
Discretization steps	1000
Atom / bond diffusion	off (fixed)
Optimization	
Optimizer	AdamW (score matching) $\rightarrow$ SOAP (physics + HJB)
Learning rate	5×10^{-4} (AdamW); $10^{-3} \rightarrow 5 \times 10^{-4}$ (SOAP)
SOAP β / precond. freq.	$(0.95, 0.95)$ / 10
Batch size / gradient clip	128 / 10
HJB divergence estimator	Hutchinson, 4 probes

top of it (Algorithm 4), their energies and forces added as follows. The total predicted energy and force field are the sums of the frozen pretrained branch and the correction branch (Algorithm 4),

$$u_\theta = u_\theta^{\mathrm{pre}} + u_\theta^{\mathrm{corr}}, \qquad \mathbf{F}_\theta = \mathbf{F}_\theta^{\mathrm{pre}} + \mathbf{F}_\theta^{\mathrm{corr}}. \tag{28}$$

The two branches are natively expressed in different units and converted before being added. The pretrained branch is trained by denoising score matching and is therefore dimensionless: its raw outputs are divided by the noise level, the energy-level analogue of the ϵ-parameterization, which yields the reduced quantities directly and the physical ones by multiplication with $k_B\tau$,

$$u_\theta^{\mathrm{pre}} = \frac{U_\theta^{\mathrm{pre,raw}}}{\sigma_t}, \quad \mathbf{F}_\theta^{\mathrm{pre}} = \frac{\mathbf{F}_\theta^{\mathrm{pre,raw}}}{\sigma_t}, \qquad U_\theta^{\mathrm{pre,phys}} = k_B\tau\, u_\theta^{\mathrm{pre}}, \quad \mathbf{F}_\theta^{\mathrm{pre,phys}} = k_B\tau\, \mathbf{F}_\theta^{\mathrm{pre}}. \tag{29}$$

The correction branch is regressed against DFT labels and is natively physical: its raw outputs are mapped back through the normalization equation 32–equation 33, and the reduced quantities follow by division by $k_B\tau$,

$$U_\theta^{\mathrm{corr,phys}} = \sigma_E\, U_\theta^{\mathrm{corr,raw}} + \sum_Z n_Z\, e_Z + N\mu, \qquad \mathbf{F}_\theta^{\mathrm{corr,phys}} = \sigma_F\, \mathbf{F}_\theta^{\mathrm{corr,raw}}, \tag{30}$$

$$u_\theta^{\mathrm{corr}} = \frac{U_\theta^{\mathrm{corr,phys}}}{k_B\tau}, \qquad \mathbf{F}_\theta^{\mathrm{corr}} = \frac{\mathbf{F}_\theta^{\mathrm{corr,phys}}}{k_B\tau}. \tag{31}$$

The HJB residual equation 40 is evaluated on the effective quantities.

A.4 Reference energies and normalization

Molecule i consists of N_i atoms with atomic numbers $Z_{i,a}$ and positions $\mathbf{x}_{i,a} \in \mathbb{R}^3$; we write $n_{i,Z} = |\{a : Z_{i,a} = Z\}|$ for its element counts and E_i^{DFT}, $\mathbf{F}_{i,a}^{\mathrm{DFT}}$ for the reference energy and forces in eV and eV Å^{-1}. Total DFT energies are dominated by the atomic self-energies e_Z, which carry no conformational information, so we first remove the best element-wise linear fit over the M training molecules,

$$\mathbf{e}_{\mathrm{iso}} = \arg\min_{\mathbf{e}\in\mathbb{R}^{|\mathcal{Z}|}} \sum_{i=1}^{M} \Big(E_i^{\mathrm{DFT}} - \sum_{Z\in\mathcal{Z}} n_{i,Z}\, e_Z\Big)^2, \qquad \hat{E}_i = \frac{1}{N_i}\Big(E_i^{\mathrm{DFT}} - \sum_Z n_{i,Z}\, e_Z\Big), \tag{32}$$

and standardize the per-atom residual energies and the forces,

$$\mu = \frac{1}{M}\sum_i \hat{E}_i, \qquad \sigma_E = \Big(\frac{1}{M}\sum_i \hat{E}_i^2 - \mu^2\Big)^{1/2}, \qquad \sigma_F = \Big(\frac{1}{3\sum_i N_i}\sum_{i,a,c}\big(F^{\mathrm{DFT}}_{i,a,c}\big)^2\Big)^{1/2}, \tag{33}$$

where c denotes the vector components. The network's DFT-normalized outputs are mapped back to physical units by inverting this normalization,

$$U_\theta^{\mathrm{phys}} = \sigma_E\, U_\theta^{\mathrm{raw}} + \sum_Z n_Z\, e_Z + N\mu, \qquad \mathbf{F}_\theta^{\mathrm{phys}} = \sigma_F\, \mathbf{F}_\theta^{\mathrm{raw}}, \tag{34}$$

and the reduced quantities follow by division through $k_B\tau$ (Eq. equation 9).

A.5 Loss terms

Write $\mathcal{B}$ for a minibatch of graphs and $\mathcal{B}_1 \subset \mathcal{B}$ for the subset pinned at the clean boundary $t = 1$, and $\langle\cdot\rangle_a$ for the mean over the atoms of a graph and their Cartesian components. Vector-field terms are averaged with $\langle\cdot\rangle_a$; scalar terms are taken per graph. The terms of the objective equation 10 are

$$\mathcal{L}_U = \frac{1}{|\mathcal{B}_1|}\sum_{b\in\mathcal{B}_1} \mathrm{huber}_\delta\big(U_\theta^{\mathrm{phys}}(\mathbf{x}^{(b)},1),\, U^{\mathrm{phys}}(\mathbf{x}^{(b)})\big), \tag{35}$$

$$\mathcal{L}_F = \frac{1}{|\mathcal{B}_1|}\sum_{b\in\mathcal{B}_1} \Big\langle \big(\mathbf{F}_\theta^{\mathrm{phys}}(\mathbf{x}^{(b)},1) - \mathbf{F}^{\mathrm{phys}}(\mathbf{x}^{(b)})\big)^2\Big\rangle_a, \tag{36}$$

$$\mathcal{L}_{\nabla U} = \frac{1}{|\mathcal{B}_1|}\sum_{b\in\mathcal{B}_1} \Big\langle \big(-\nabla_{\mathbf{x}} U_\theta^{\mathrm{phys}}(\mathbf{x}^{(b)},1) - \mathbf{F}^{\mathrm{phys}}(\mathbf{x}^{(b)})\big)^2\Big\rangle_a, \tag{37}$$

$$\mathcal{L}_{\mathrm{cons}} = \frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}} \Big\langle \bar\sigma^2_{t^{(b)}}\big(\mathbf{F}_\theta(\mathbf{x}_t^{(b)},t^{(b)}) + \nabla_{\mathbf{x}} u_\theta(\mathbf{x}_t^{(b)},t^{(b)})\big)^2\Big\rangle_a, \qquad \bar\sigma_t = \max(\sigma_t,\sigma_{\mathrm{floor}}), \tag{38}$$

$$\mathcal{L}_{\mathrm{DSM}} = \frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}} \Big\langle \big(-\sigma_{t^{(b)}}\mathbf{F}_\theta(\mathbf{x}_t^{(b)},t^{(b)}) - \boldsymbol{\epsilon}^{(b)}\big)^2\Big\rangle_a, \tag{39}$$

$$\mathcal{L}_{\mathrm{HJB}} = \frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}} \Big| \partial_t u_\theta(\mathbf{x}_t^{(b)},t^{(b)}) + \tfrac{1}{2} g^2_{t^{(b)}}\Big(\big\|\mathbf{F}_\theta(\mathbf{x}_t^{(b)},t^{(b)})\big\|_2^2 + \mathrm{div}\,\mathbf{F}_\theta(\mathbf{x}_t^{(b)},t^{(b)})\Big)\Big|. \tag{40}$$

The two force boundary terms constrain different quantities. $\mathcal{L}_F$ fits the force head, which the sampler integrates. $\mathcal{L}_{\nabla U}$ fits the gradient of the energy head, in physical units and through its own derivative rather than the consistency term's, so the energy itself is a conservative potential with the DFT forces. Since $\mathbf{f}_t \equiv \mathbf{0}$ under variance exploding schedule, throughout used in our set up, the drift terms $\langle \mathbf{F}_{\theta,t}, \mathbf{f}_t\rangle$ and $\mathrm{div}(\mathbf{f}_t)$ of the general residual vanish. equation 40 is therefore the ℓ_1 norm of the per-graph residual equation 5 with $\mathbf{v}_{\theta,t}$ substituted from equation 6. Its divergence is the trace over all $3N$ coordinates, estimated with K Rademacher probes (Hutchinson, 1989),

$$\widehat{\mathrm{div}}(\mathbf{F}_{\theta,t}) = \frac{1}{K}\sum_{k=1}^{K} \big\langle \mathbf{z}_k,\ \nabla_{\mathbf{x}}\langle \mathbf{F}_{\theta,t}, \mathbf{z}_k\rangle\big\rangle, \qquad \mathbf{z}_k \sim \mathrm{Rademacher}. \tag{41}$$

A.6 ALGORITHMS

Algorithm 1 Pretraining-free training on the Swiss-roll prototype: supervision through energy only (Section 4.1). *(optional)* Temperature conditioning; setting $\tau \equiv 1$ recovers the unconditioned model.

Require: reference point cloud $\{\mathbf{y}_j\}_{j=1}^{M} \subset \mathbb{R}^2$ (Swiss roll) and KDE bandwidth h, defining the PES $u(\mathbf{x}) = -\ln p_{\mathrm{KDE}}(\mathbf{x})$ with $p_{\mathrm{KDE}}(\mathbf{x}) = \frac{1}{M}\sum_j \mathcal{N}(\mathbf{x}; \mathbf{y}_j, h^2\mathbf{I})$
Require: domain $\Omega = [-4, 4]^2$; batch size B; number of clean-boundary points $B_1 < B$; noise schedule σ_t; weights $\lambda_U, \lambda_{\mathrm{cons}}, \lambda_{\mathrm{HJB}}$
Require: *(optional)* temperature range $[\tau_{\min}, \tau_{\max}]$, sampled in log-space; the reduced potential at temperature τ is $u_\tau(\mathbf{x}) = u(\mathbf{x})/\tau$, so that $p_\tau \propto e^{-u_\tau}$
while not converged **do**
  draw B points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(B)}$ uniformly at random from Ω ▷ each point is one single-node graph
  draw $t^{(b)} \sim \mathcal{U}[0, 1]$ independently for $b = 1, \ldots, B$
  (optional) draw $\ln \tau^{(b)} \sim \mathcal{U}[\ln \tau_{\min}, \ln \tau_{\max}]$ independently for $b = 1, \ldots, B$
  set $t^{(b)} \leftarrow 1$ for the first B_1 points, and call these the clean-boundary set $\mathcal{B}_1$
  $u_\theta(\mathbf{x}, t, \tau),\ \mathbf{F}_\theta(\mathbf{x}, t, \tau) \leftarrow \mathrm{net}_\theta(\mathbf{x}, t, \tau)$ ▷ one forward pass; τ is embedded like t and omitted without temperature conditioning
  $\nabla_{\mathbf{x}} u_\theta \leftarrow \partial\big(\sum_b u_\theta\big)/\partial \mathbf{x}, \quad \partial_t u_\theta \leftarrow \partial\big(\sum_b u_\theta\big)/\partial t$ ▷ automatic differentiation
  $\mathrm{div}(\mathbf{F}_{\theta,t}) \leftarrow \mathrm{tr}\, \nabla_{\mathbf{x}} \mathbf{F}_{\theta,t}$
  evaluate the energies $u(\mathbf{x}^{(b)})$, $b \in \mathcal{B}_1$, from the KDE surface, and rescale them to $u_{\tau^{(b)}}(\mathbf{x}^{(b)}) = u(\mathbf{x}^{(b)})/\tau^{(b)}$ ▷ *(optional)*; without conditioning the target is $u(\mathbf{x}^{(b)})$
  $\mathcal{L} \leftarrow \lambda_U \mathcal{L}_U + \lambda_{\mathrm{cons}} \mathcal{L}_{\mathrm{cons}} + \lambda_{\mathrm{HJB}} \mathcal{L}_{\mathrm{HJB}}$ ▷ equation 35, equation 38, equation 40
  $\theta \leftarrow \mathrm{AdamOptimizerStep}(\theta, \nabla_\theta \mathcal{L})$
end while

Algorithm 2 Sampling and energy evaluation (reverse SDE). *(optional)* applies temperature-conditioned sampling; setting $\tau \equiv 1$ recovers the unconditioned sampler.

Require: trained energy head u_θ and vector head $\mathbf{F}_\theta$; molecular graph (atom types and bonds) or, for the two-dimensional prototype, a single-node graph
Require: number of integration steps N
Require: *(optional)* number of Langevin corrector steps M per integration step ▷ $M = 0$ recovers the pure predictor
Require: *(optional)* target temperature $\tau \in [\tau_{\min}, \tau_{\max}]$, held fixed along the trajectory ▷ the sample is drawn from $p_\tau \propto e^{-u/\tau}$
$\Delta t \leftarrow 1/N$
$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$, then subtract the centre of mass (do not subtract for Swiss-roll prototype) ▷ Gaussian prior at $t = 0$, independent of τ
for $i = 0, 1, \ldots, N-1$ **do**
  $t \leftarrow i\,\Delta t$
  $\Delta\sigma^2 \leftarrow \sigma_t^2 - \sigma_{t+\Delta t}^2, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x} \leftarrow \mathbf{x} + \Delta\sigma^2\, \mathbf{F}_\theta(\mathbf{x}, t, \tau) + \sqrt{\Delta\sigma^2}\, \mathbf{z}$ ▷ reverse SDE, VE discretization; for molecules $\mathbf{F}_\theta = \mathbf{F}_\theta^{\mathrm{phys}}/k_B\tau$ (Eq. 9)
  if not Swiss-roll prototype: subtract the centre of mass from $\mathbf{x}$
  for $j = 1, \ldots, M$ **do** ▷ *(optional)* Langevin corrector at $t + \Delta t$
    $\epsilon \leftarrow \Delta\sigma^2/M, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x} \leftarrow \mathbf{x} + \epsilon\, \mathbf{F}_\theta(\mathbf{x}, t + \Delta t, \tau) + \sqrt{2\epsilon}\, \mathbf{z}$
    if not Swiss-roll prototype: subtract the centre of mass from $\mathbf{x}$
  end for
end for
return conformation $\mathbf{x}$, its reduced energy $u_\theta(\mathbf{x}, 1, \tau)$ and forces $\mathbf{F}_{\theta,1}(\mathbf{x}, \tau)$

Algorithm 3 ENTROPICA single-branch training (default): one network trained jointly on the score-matching, boundary and HJB objectives (Sections 4.2 and 4.3). The two-branch variant used for ANI-1x is Algorithm 4.

Require: dataset $\mathcal{D}$ of conformations $\mathbf{x} \in \mathbb{R}^{N\times 3}$ with atom types and bonds, DFT energies $U^{\mathrm{phys}}(\mathbf{x})$ and forces $\mathbf{F}^{\mathrm{phys}}(\mathbf{x})$
Require: batch size B; clean-boundary count $B_1 < B$; noise schedule σ_t; Hutchinson probes K; loss weights $\lambda_{\mathrm{DSM}}, \lambda_{\mathrm{cons}}, \lambda_U, \lambda_F, \lambda_{\mathrm{HJB}}$; optimizer $\mathcal{O} \in \{\mathrm{AdamW}, \mathrm{SOAP}\}$
while not converged **do**
  draw B triples $\big(\mathbf{x}^{(b)}, u(\mathbf{x}^{(b)}), \mathbf{F}(\mathbf{x}^{(b)})\big)$ from $\mathcal{D}$ and times $t^{(b)} \sim \mathcal{U}[0,1]$
  set $t^{(b)} \leftarrow 1$ for the first B_1 molecules, the clean-boundary set $\mathcal{B}_1$
  $\boldsymbol{\epsilon}^{(b)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathbf{x}_t^{(b)} \leftarrow \mathbf{x}^{(b)} + \sigma_{t^{(b)}}\boldsymbol{\epsilon}^{(b)}$ ▷ $\mathbf{x}_t^{(b)} = \mathbf{x}^{(b)}$ for $b \in \mathcal{B}_1$
  $u_\theta(\mathbf{x}_t, t),\ \mathbf{F}_\theta(\mathbf{x}_t, t) \leftarrow \mathrm{net}_\theta(\mathbf{x}_t, t)$ ▷ single network, both heads
  $\nabla_{\mathbf{x}} u_\theta \leftarrow \partial\big(\sum_b u_\theta\big)/\partial \mathbf{x}_t, \quad \partial_t u_\theta \leftarrow \partial\big(\sum_b u_\theta\big)/\partial t$
  $\widehat{\mathrm{div}}(\mathbf{F}_{\theta,t}) \leftarrow$ Hutchinson estimate equation 41 with K Rademacher probes
  convert the heads to physical and reduced units, equation 34 and equation 9
  $\mathcal{L} \leftarrow \lambda_{\mathrm{DSM}}\mathcal{L}_{\mathrm{DSM}} + \lambda_{\mathrm{cons}}\mathcal{L}_{\mathrm{cons}} + \lambda_U\mathcal{L}_U + \lambda_F\mathcal{L}_F + \lambda_{\mathrm{HJB}}\mathcal{L}_{\mathrm{HJB}}$ ▷ equation 35–equation 40
  $\theta \leftarrow \mathcal{O}\text{-step}(\theta, \nabla_\theta \mathcal{L})$ ▷ AdamW for score matching; SOAP once the physical boundary/HJB terms are active
end while

Algorithm 4 ENTROPICA: score pretraining followed by physical correction (Section 4.4.)

Require: dataset $\mathcal{D}$ of molecular conformations $\mathbf{x} \in \mathbb{R}^{N\times 3}$ with atom types and bonds, DFT energies $u(\mathbf{x})$ and DFT forces $\mathbf{F}(\mathbf{x})$
Require: batch size B; number of clean-boundary molecules $B_1 < B$; noise schedule σ_t; Hutchinson probes K; loss weights
Stage A — score pretraining. Train the base network θ_{pre} on conformations only; the DFT energies and forces are not used.
while not converged **do**
  draw B conformations $\mathbf{x}^{(b)}$ from $\mathcal{D}$ and times $t^{(b)} \sim \mathcal{U}[0,1]$
  $\boldsymbol{\epsilon}^{(b)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathbf{x}_t^{(b)} \leftarrow \mathbf{x}^{(b)} + \sigma_{t^{(b)}}\boldsymbol{\epsilon}^{(b)}$
  $u_{\mathrm{pre}}(\mathbf{x}_t, t),\ \mathbf{F}_{\mathrm{pre}}(\mathbf{x}_t, t) \leftarrow \mathrm{net}_{\theta_{\mathrm{pre}}}(\mathbf{x}_t, t)$
  $\nabla_{\mathbf{x}} u_{\mathrm{pre}} \leftarrow \partial\big(\sum_b u_{\mathrm{pre}}\big)/\partial \mathbf{x}_t$
  $\mathcal{L} \leftarrow \lambda_{\mathrm{DSM}}\mathcal{L}_{\mathrm{DSM}} + \lambda_{\mathrm{cons}}\mathcal{L}_{\mathrm{cons}}$ ▷ equation 39, equation 38
  $\theta_{\mathrm{pre}} \leftarrow \mathrm{AdamWOptimizerStep}(\theta_{\mathrm{pre}}, \nabla_{\theta_{\mathrm{pre}}}\mathcal{L})$
end while
Stage B — physical correction. Freeze θ_{pre} and train the correction network θ_{corr} on top of it.
while not converged **do**
  draw B triples $\big(\mathbf{x}^{(b)}, u(\mathbf{x}^{(b)}), \mathbf{F}(\mathbf{x}^{(b)})\big)$ from $\mathcal{D}$ and times $t^{(b)} \sim \mathcal{U}[0,1]$
  set $t^{(b)} \leftarrow 1$ for the first B_1 molecules, and call these the clean-boundary set $\mathcal{B}_1$
  $\boldsymbol{\epsilon}^{(b)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad \mathbf{x}_t^{(b)} \leftarrow \mathbf{x}^{(b)} + \sigma_{t^{(b)}}\boldsymbol{\epsilon}^{(b)}$ ▷ $\mathbf{x}_t^{(b)} = \mathbf{x}^{(b)}$ for $b \in \mathcal{B}_1$
  $u_{\mathrm{pre}}, \mathbf{F}_{\mathrm{pre}} \leftarrow \mathrm{net}_{\theta_{\mathrm{pre}}}(\mathbf{x}_t, t)$ ▷ frozen weights, no gradient
  $u_{\mathrm{corr}}, \mathbf{F}_{\mathrm{corr}} \leftarrow \mathrm{net}_{\theta_{\mathrm{corr}}}(\mathbf{x}_t, t)$
  convert both branches to reduced units, equation 29 and equation 31
  $u_\theta \leftarrow u_{\mathrm{pre}} + u_{\mathrm{corr}}, \quad \mathbf{F}_\theta \leftarrow \mathbf{F}_{\mathrm{pre}} + \mathbf{F}_{\mathrm{corr}}$ ▷ total energy and force field
  $\nabla_{\mathbf{x}} u_\theta \leftarrow \partial\big(\sum_b u_\theta\big)/\partial \mathbf{x}_t, \quad \partial_t u_\theta \leftarrow \partial\big(\sum_b u_\theta\big)/\partial t$
  $\widehat{\mathrm{div}}(\mathbf{F}_{\theta,t}) \leftarrow$ Hutchinson estimate equation 41 with K Rademacher probes
  $\mathcal{L} \leftarrow \lambda_U\mathcal{L}_U + \lambda_F\mathcal{L}_F + \lambda_{\nabla U}\mathcal{L}_{\nabla U} + \lambda_{\mathrm{cons}}\mathcal{L}_{\mathrm{cons}} + \lambda_{\mathrm{HJB}}\mathcal{L}_{\mathrm{HJB}}$ ▷ equation 35–equation 38, equation 40
  if optimizer is SOAP **then**
    $\theta_{\mathrm{corr}} \leftarrow \mathrm{SOAPOptimizerStep}(\theta_{\mathrm{corr}}, \nabla_{\theta_{\mathrm{corr}}}\mathcal{L})$
  else
    $\theta_{\mathrm{corr}} \leftarrow \mathrm{AdamOptimizerStep}(\theta_{\mathrm{corr}}, \nabla_{\theta_{\mathrm{corr}}}\mathcal{L})$
  end if
end while

Algorithm 5 Pretraining-free flow matching training on the Swiss-roll prototype: supervision through energy only (Section 4.1). The network predicts a velocity field instead of a score. *(optional)* Temperature conditioning; setting $\tau \equiv 1$ recovers the unconditioned model.

Require: reference point cloud $\{\mathbf{y}_j\}_{j=1}^{M} \subset \mathbb{R}^2$ (Swiss roll) and KDE bandwidth h, defining the PES $u(\mathbf{x}) = -\ln p_{\mathrm{KDE}}(\mathbf{x})$ with $p_{\mathrm{KDE}}(\mathbf{x}) = \frac{1}{M}\sum_j \mathcal{N}(\mathbf{x}; \mathbf{y}_j, h^2\mathbf{I})$
Require: domain $\Omega = [-4, 4]^2$; batch size B; number of clean-boundary points $B_1 < B$; Hutchinson probes K; weights $\lambda_U, \lambda_{\mathrm{cons}}, \lambda_{\mathrm{HJB}}$
Require: *(optional)* temperature range $[\tau_{\min}, \tau_{\max}]$, sampled in log-space; the reduced potential at temperature τ is $u_\tau(\mathbf{x}) = u(\mathbf{x})/\tau$, so that $p_\tau \propto e^{-u_\tau}$
while not converged **do**
  draw B points $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(B)}$ uniformly at random from Ω ▷ each point is one single-node graph
  draw $t^{(b)} \sim \mathcal{U}[0, 1]$ independently for $b = 1, \ldots, B$
  (optional) draw $\ln \tau^{(b)} \sim \mathcal{U}[\ln \tau_{\min}, \ln \tau_{\max}]$ independently for $b = 1, \ldots, B$
  set $t^{(b)} \leftarrow 1$ for the first B_1 points, and call these the clean-boundary set $\mathcal{B}_1$
  $u_\theta(\mathbf{x}, t, \tau),\ \mathbf{v}_{\theta,t}(\mathbf{x}, \tau) \leftarrow \mathrm{net}_\theta(\mathbf{x}, t, \tau)$ ▷ one forward pass; τ is embedded like t and omitted without temperature conditioning
  $\nabla_{\mathbf{x}} u_\theta \leftarrow \partial\big(\sum_b u_\theta\big)/\partial \mathbf{x}, \quad \partial_t u_\theta \leftarrow \partial\big(\sum_b u_\theta\big)/\partial t$ ▷ automatic differentiation
  $\widehat{\mathrm{div}}(\mathbf{v}_{\theta,t}) \leftarrow$ Hutchinson estimate equation 41 with K Rademacher probes
  evaluate the energies $u(\mathbf{x}^{(b)})$, $b \in \mathcal{B}_1$, from the KDE surface, and rescale them to $u_{\tau^{(b)}}(\mathbf{x}^{(b)}) = u(\mathbf{x}^{(b)})/\tau^{(b)}$ ▷ *(optional)*; without conditioning the target is $u(\mathbf{x}^{(b)})$
  $\mathcal{L} \leftarrow \lambda_U \mathcal{L}_U + \lambda_{\mathrm{cons}} \mathcal{L}_{\mathrm{cons}} + \lambda_{\mathrm{HJB}} \mathcal{L}_{\mathrm{HJB}}$ ▷ equation 35, equation 42, equation 43
  $\theta \leftarrow \mathrm{AdamOptimizerStep}(\theta, \nabla_\theta \mathcal{L})$
end while

Algorithm 6 Flow-matching sampling and energy evaluation (probability-flow ODE). *(optional)* applies temperature-conditioned sampling; setting $\tau \equiv 1$ recovers the unconditioned sampler.

Require: trained energy head u_θ and velocity head $\mathbf{v}_\theta$; molecular graph (atom types and bonds) or, for the two-dimensional prototype, a single-node graph
Require: number of integration steps N
Require: *(optional)* target temperature $\tau \in [\tau_{\min}, \tau_{\max}]$, held fixed along the trajectory ▷ the sample is drawn from $p_\tau \propto e^{-u/\tau}$
$\Delta t \leftarrow 1/N$
$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$, then subtract the centre of mass (do not subtract for Swiss-roll prototype) ▷ Gaussian prior at $t = 0$, independent of τ
for $i = 0, 1, \ldots, N-1$ **do**
  $t \leftarrow i\,\Delta t$
  $\mathbf{x} \leftarrow \mathbf{x} + \Delta t\, \mathbf{v}_\theta(\mathbf{x}, t, \tau)$ ▷ explicit Euler step of the probability-flow ODE; τ is dropped without temperature conditioning
  if not Swiss-roll prototype: subtract the centre of mass from $\mathbf{x}$
end for
return conformation $\mathbf{x}$, its reduced energy $u_\theta(\mathbf{x}, 1, \tau)$ and forces $-\nabla_{\mathbf{x}} u_\theta(\mathbf{x}, 1, \tau)$

B Experimental details

B.1 Noise schedule and sampler

In noise time $s = 1 - t$ the VE schedule is geometric, $\sigma(s) = \sigma_{\min}(\sigma_{\max}/\sigma_{\min})^s$, with $\sigma_{\min} = 0.01$ and $\sigma_{\max} = 5$ for molecules and $\sigma_{\min} = 0.001$ and $\sigma_{\max} = 15$ for the Swiss roll. Generation integrates the reverse SDE of Algorithm 2 with N Euler–Maruyama steps, each followed by M Langevin corrector steps $\mathbf{x} \leftarrow \mathbf{x} + \epsilon\, \mathbf{F}_\theta + \sqrt{2\epsilon}\, \mathbf{z}$ with $\epsilon = \Delta\sigma^2/M$ (Table 4).

B.2 Data and splits

Each rMD17 molecule (Christensen & Von Lilienfeld, 2020) is trained separately on its 100,000-frame trajectory, 5 % of which are held out and serve as the test set used for the tables and figures. Bonds are annotated once per molecule and held fixed for all frames. Energies and forces are

Table 2: Architecture. The two-branch model instantiates the backbone twice, as the frozen pretrained branch and as the correction. Hidden dimensions are those of nodes and edges alike.

	single-branch (rMD17, ANI-1x)	two-branch, per branch (ANI-1x)
GVP convolution layers	3	4
Hidden dim (scalar, vector)	$(64, 32)$	$(128, 32)$
Parameters	0.59 M	1.64 M (3.28 M in total)
Scalar node feature width	128	
Activation / vector gating	SiLU / yes	
Dropout	0	
Time embedding	sinusoidal, 64	
Distance RBFs / $d_{\max}$	16 / 50 Å	
Edges	covalent bonds, bond-type one-hot	
Atom types	H, C, N, O, F	
Head outputs divided by	$\max(\sigma_t, \sigma_{\min})$	

Table 3: Training of the molecular models. Epochs give the maximum and, in parentheses, the kept checkpoint. Shared by all runs: VE schedule $\sigma \in [0.01, 5]$ with 1000 training steps, Huber $\delta = 0.5$ eV on $\mathcal{L}_U$, $\sigma_{\text{floor}} = 0.2$, gradient clipping at 10 and a constant learning rate. SOAP uses $\beta = (0.95, 0.95)$, preconditioning every 10 steps and weight decay 10^{-12}; AdamW uses AMSGrad, $\beta = (0.9, 0.999)$ and weight decay 10^{-12}. *rMD17* covers Sections 4.2 and 4.3, whose non-equilibrium models use a batch of 120. The correction's epochs count on from its warm-up, and both phases were stopped by hand. *DSM only* is the score-only ablation of Section 4.4. The rMD17 finetune (*ft*) continues from the kept checkpoint for 100 epochs with $\lambda_U = \lambda_F = \lambda_{\nabla U} = 2$ and $\lambda_{\text{HJB}} = 0.025$ (warm-up 0.005 for 4 epochs), otherwise as in the rMD17 column. The correction branch is freshly initialized on the frozen pretrained branch; the HJB warm-up gives weight and epochs.

	single-branch		two-branch (ANI-1x)	
	rMD17	ANI-1x, DSM only	pretrain	correction
Temperature τ (K)	500	300	300	300
Initialization	fresh	fresh	fresh	on pretrain
Optimizer	SOAP	AdamW	AdamW	AdamW
Learning rate	10^{-3}	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$
Batch (graphs)	240	128	128	32
Epochs	150	500 (40)	100 (68)	$10 + 20$ (25)
$\lvert\mathcal{B}_1\rvert/\lvert\mathcal{B}\rvert$	0.33	–	–	0.33
λ_{DSM}	1	1	1	0
λ_{cons}	5	0	1	5
$\lambda_U, \lambda_F, \lambda_{\nabla U}$	$1, 1, 1$	$0, 0, 0$	$0, 0, 0$	$1, 2, 1$
λ_{HJB}	0.005	0	0	0.1
HJB warm-up	0.001, 4	–	–	0.001, 10
Hutchinson probes K	4	–	–	3

converted from kcal mol^{-1} to eV. The non-equilibrium training sets are resampled from the training pool only (Appendix C.4). ANI-1x is described in Appendix C.5.

B.3 Training protocol

Tables 2–5 list every setting. A fixed fraction $|\mathcal{B}_1|/|\mathcal{B}|$ of each batch is pinned at $t = 1$ and carries the boundary terms; the rest is noised at $t \sim \mathcal{U}[0, 1]$. The kept checkpoint has the lowest validation loss evaluated at the final loss weights, excluding the HJB warm-up epochs. *ft* finetunes that checkpoint with λ_{HJB} raised five-fold and the boundary weights two-fold. Ablations set the following weights to zero: *no HJB* sets $\lambda_{\text{HJB}} = 0$, *no PES* sets $\lambda_U = \lambda_F = \lambda_{\nabla U} = 0$, and *DSM only* both. On ANI-1x the *DSM only* ablation is a single-branch model trained on score matching alone under AdamW. The two-branch model pretrains the score, freezes it, and trains the correction first with $\lambda_{\text{HJB}} = 0.001$ and then with $\lambda_{\text{HJB}} = 0.1$. All ablations were trained to convergence at the weights of Table 3.

Table 4: Sampling. Each of the N predictor steps, uniform in t, is followed by M Langevin corrections (Appendix B.1); the flow-matching prototype integrates the probability-flow ODE instead. The references are the KDE density, 4000 held-out MD frames, the 500 K training data (non-equilibrium) or the MD trajectory reweighted to each τ (sweep), and the held-out DFT conformers.

	Swiss roll	rMD17	non-eq. / τ sweep	ANI-1x
Predictor steps N	200 (100, FM)	100	120	1000
Corrector steps M	0	3	5	2
Samples per molecule	1500	1000	2000	500
Molecules	–	3	1	64
Temperature	0.1–2 (reduced)	500 K	500 / 200–800 K	300 K
Seed	0	0	0	42

Table 5: Two-dimensional prototype. Steps are optimizer steps, each on a fresh batch of collocation points.

Reference	2000 Swiss-roll points (noise 0.2), KDE bandwidth $h = 0.4$
Collocation	uniform on $[-4, 4]^2$; batch 1024, 25 % pinned at $t = 1$
VE schedule	$\sigma \in [0.001, 15]$
Network	MLP, 4 time-conditioned residual blocks of width 128; time embedding 64
Divergence	exact (two dimensions)
Loss weights	$\lambda_U = \lambda_{\text{cons}} = \lambda_{\text{HJB}} = 1$; Huber $\delta = 1$
Optimizer	Adam, lr $= 5 \cdot 10^{-4}$
Steps	30,000; 150,000 temperature-conditioned
Temperature conditioning	τ log-uniform on $[0.1, 2]$; boundary only for the first 10 % of steps, full residual from 80 %

B.4 Referee model

Energies of generated rMD17 conformers are scored by a MACE model (Batatia et al., 2022) trained separately for each molecule on 950 frames of its equilibrium training pool, with 50 frames for early stopping, following the published rMD17 recipe: 2 interactions, 256 channels, $L = 2$, correlation 3, $r_{\max} = 5$ Å, float64, 1000 epochs with stochastic weight averaging from epoch 750. On the held-out frames it reaches 0.9–1.1 meV energy MAE, and 1.30 meV on the 600 K set of Appendix C.4. ANI-1x conformers are scored by ANI-2x (Devereux et al., 2020).

B.5 Evaluation metrics

Torsion densities are compared by the Jensen–Shannon divergence of histograms (60 bins for malonaldehyde and 120 for salicylic acid and azobenzene, over periods of 360°, 360° and 180°) and by the circular Wasserstein-1 distance. Each floor is the same quantity between the full reference and a random subsample of itself of the sample size. The effective temperature is $k_B T$ of a harmonic density-of-states fit $p(E) \propto (E - E_0)^{n/2-1} e^{-E/k_B T}$ with $n = 3N - 6$ to referee energies, excluding the highest 1 %; errors are bootstrap standard errors. We follow Igashov et al. (2026) to compute geometric validity of generated samples, and exclude extreme geometric outliers from energy evaluation: A conformer is valid if every bond length, bond angle and ring of its annotated topology is likely under kernel densities fitted to 451,134 ANI-1x training conformers (lowest $q > 1.78 \cdot 10^{-5}$, aromatic rings within 0.332 Å of planar); the thresholds pass 99.0–99.6 % of held-out ANI-1x conformers and every held-out rMD17 frame. For the equilibrium recovery experiments, we compute the retained fraction as $\Delta\langle E_{\text{MACE}}\rangle_{\text{samples}} / \Delta\langle E_{\text{DFT}}\rangle_{\text{data}}$ between the two training sets.

B.6 Sampling-temperature sweep

The full-objective rMD17 azobenzene model before finetuning, trained at 500 K, is sampled at $\tau \in \{200, 300, 400, 500, 600, 800\}$ K, with τ entering only through the reduced force. The reference at each τ is 10,000 frames of the 500 K MD trajectory Boltzmann-reweighted by $\exp[-U^{\text{DFT}}(1/k_B\tau - 1/k_B \cdot 500\,\text{K})]$; where its Kish effective sample size (Liu, 1996) falls below 1 % (at 200, 300 and 800 K), no reference is shown. The model's effective temperature tracks the reference through 500 K (417 vs. 428 K at 400 K, 559 vs. 553 K at 500 K) and runs hot beyond it (707 vs. 669 K at 600 K; 1171 K at 800 K).

C Additional Results

C.1 Temperature Conditioning of Potential-Alone Prototype System

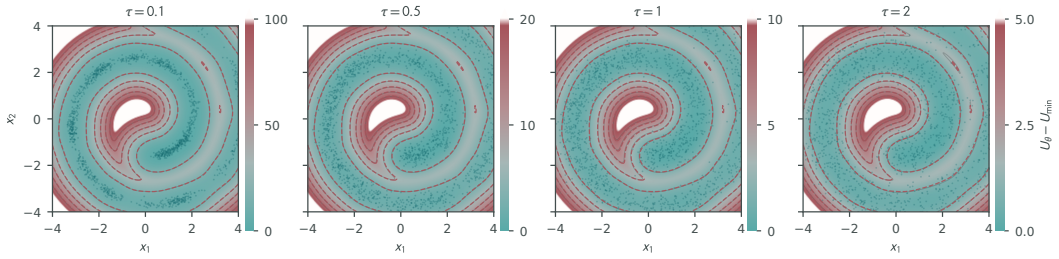


Figure 6: Temperature-conditioned HJB training. The target reference energy is scaled by a temperature scalar $u_\tau(\mathbf{x}) = u(\mathbf{x})/\tau$. A small temperature scalar leads to sharper distributions and vice versa.

C.2 Flow Matching Implementation of the Potential-Alone Prototype

We also tested the HJB loss equation 7 in the potential-alone setup of Section 4.1 for flow matching. Here, the consistency equation 38 and HJB equation 7 losses are rearranged to the following objectives

$$\mathcal{L}_{\text{cons}} = \frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}}\left\|\bar{\sigma}_{t^{(b)}}\left(\mathbf{v}_{\theta,t}(\mathbf{x}_t^{(b)}) + \tfrac{1}{2}g_t^2\,\nabla_{\mathbf{x}}u_\theta(\mathbf{x}^{(b)},t^{(b)})\right)\right\|^2, \qquad \bar{\sigma}_t = \max(\sigma_t, \sigma_{\text{floor}}), \quad (42)$$

and

$$\mathcal{L}_{\text{HJB}} = \frac{1}{B}\sum_{b=1}^{B}\left|\left(\partial_t u_\theta(\mathbf{x}^{(b)},t^{(b)}) + \mathbf{v}_{\theta,t}(\mathbf{x}_t^{(b)})\cdot\nabla_{\mathbf{x}}u_\theta(\mathbf{x}^{(b)},t^{(b)}) - \nabla\cdot\mathbf{v}_{\theta,t}(\mathbf{x}_t^{(b)})\right)\right|. \quad (43)$$

The training and sampling procedures are stated in Algorithms 5 and 6. Fig. 7 shows the results of the training. The flow matching model learns the intermediate PES as well and recovers the sampled data point distributions from it.

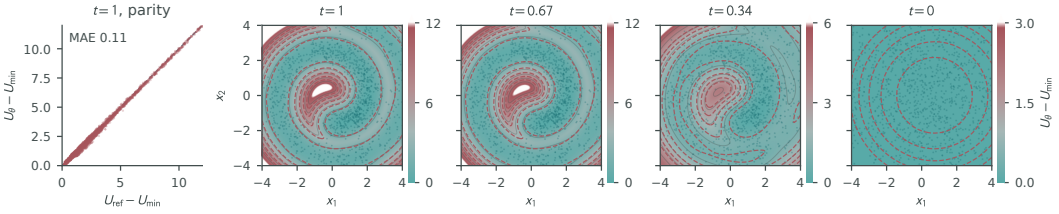


Figure 7: Recovery of the Boltzmann distribution from a PES alone on the two-dimensional Swiss-roll prototype for flow matching model. The reference PES and figure layout is identical to 2A-B. The same energy-only training under flow matching, with the objectives rearranged to equation 42 and equation 43 and sampling by the probability-flow ODE (Algorithm 6). Darker teal corresponds to lower energies and higher probabilities of finding a data point at this position. Shown are the learned PES $u_\theta(\mathbf{x}, t)$ of the flow-matching model alongside reference contour lines from the reference PES and generated samples for $t = 1$ are obtained by Algorithm 6.

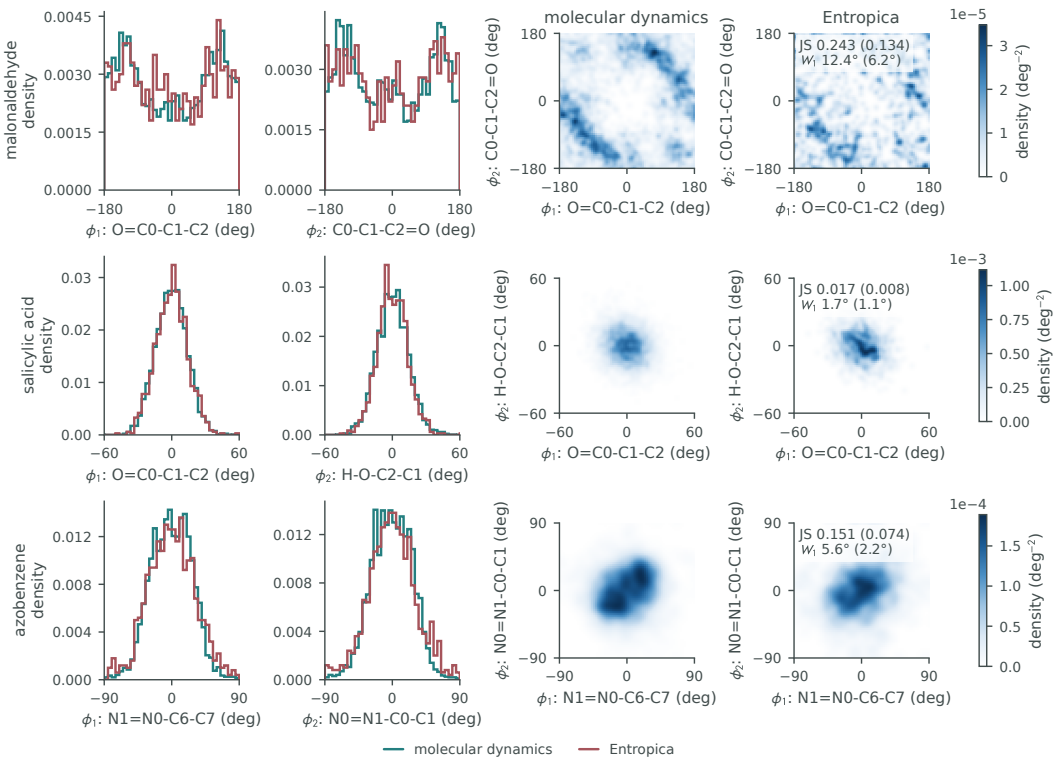


Figure 8: Torsion distributions on rMD17. For each molecule, the marginal densities of its two torsions ϕ_1 and ϕ_2 from molecular dynamics (teal; 4000 held-out frames) and from 1000 ENTROPICA samples (red), followed by their joint densities. Axis labels give the chemical torsion; Jensen—Shannon divergence and circular W_1 between the joint densities are indicated. The floor is indicated in brackets: MD against a same-size subsample of itself.

C.3 ADDITIONAL rMD17 RESULTS

Table 6: Ablations of ENTROPICA on rMD17. *Scoring*: held-out frames at 500 K, centred energy MAE and R^2 against DFT; MACE (trained on 950 frames of the training pool) and MMFF94 (typed from the bond annotation) score. *Sampling*: *valid*, the samples passing the geometry-validity check of Appendix B.5; JS divergence and circular W_1 of the sampled ϕ_1 density to MD, ϕ_1 being O=C–C–C in malonaldehyde and salicylic acid and N=N–C–C in azobenzene; and the effective temperature of the valid samples under the MACE referee (density-of-states fit, top 1% of energies trimmed). The *MD* row is the floor: the reference against a same-size subsample of itself, and the estimator's reading of the MD ensemble. ENTROPICA *(ft)* finetunes the *no finetune* model with 5× HJB and 2× PES weight; *no HJB* keeps the Dirichlet and Neumann conditions and drops the residual, *no PES* the energy and force boundary terms, *DSM only* both. Without the PES term the energy is defined only up to an affine map, so the MAE against DFT measures a gauge the objective never fixed as well as a modelling error; the slope of the fit against DFT falls from 0.96–0.98 to 0.62–0.74.

	scoring		sampling			
	MAE (meV)	R^2	valid (%)	JS (nats)	W_1 (deg)	T_{eff} (K)
malonaldehyde						
ENTROPICA (ft)	11.4	0.995	100.0	0.0107	4.3	539
no finetune	10.9	0.995	100.0	0.0097	2.2	563
no HJB	8.1	0.997	100.0	0.0074	2.3	673
no PES	58.6	0.898	100.0	0.0111	2.7	792
DSM only	64.3	0.859	100.0	0.0079	2.2	766
MACE	1.1	0.9999	–	–	–	–
MMFF94	107.7	0.694	–	–	–	–
MD	–	–	–	0.0054	2.1	608
salicylic acid						
ENTROPICA (ft)	8.4	0.999	97.7	0.0146	2.7	592
no finetune	11.3	0.998	99.8	0.0077	0.5	591
no HJB	6.9	0.999	99.9	0.0072	0.6	661
no PES	68.2	0.913	100.0	0.0047	0.4	780
DSM only	70.3	0.918	100.0	0.0067	0.5	711
MACE	1.0	0.9999	–	–	–	–
MMFF94	128.2	0.635	–	–	–	–
MD	–	–	–	0.0032	0.5	585
azobenzene						
ENTROPICA (ft)	9.5	0.999	100.0	0.0223	3.0	548
no finetune	9.7	0.999	100.0	0.0263	2.7	549
no HJB	9.0	0.999	100.0	0.0281	3.3	616
no PES	84.6	0.888	99.9	0.0284	3.3	778
DSM only	73.8	0.931	100.0	0.0209	2.5	656
MACE	0.9	>0.9999	–	–	–	–
MMFF94	118.8	0.776	–	–	–	–
MD	–	–	–	0.0127	0.9	551

C.4 TRAINING WITHOUT EQUILIBRIUM DATA: DETAILS

Data. An equilibrium validation split (5 %) is held out first, and only the training pool is resampled, with weights $w(\mathbf{x}) \propto \exp[-U^{\mathrm{DFT}}(\mathbf{x})(1/k_B\tau' - 1/k_B\tau)]$ and $\tau' = 600$ K. A reweighting exponential in the energy is a change of temperature, and it is the only reweighting that still gives a Boltzmann ensemble of the same PES. The resampled set lies $8.9\, k_B\tau$ above the equilibrium set in mean DFT energy.

Referee and readout. All ensembles are scored by the MACE referee of Appendix B.4. The imposed shift is the difference of the mean DFT energies of the two training pools, $+0.385$ eV at

Table 7: Recovery of the 500 K equilibrium ensemble of rMD17 azobenzene from training data resampled to 600 K; every model is sampled at $\tau = 500$ K and ϕ_1 pools both phenyl twists. *valid*: samples, and data frames, passing the geometry check of Appendix B.5, the only ones scored. $\Delta\langle E\rangle$: mean MACE-referee energy relative to the mean DFT energy of the 500 K data. T_{eff}: density-of-states fit to the referee energies (top 1% trimmed), with its bootstrap error. JS divergence and circular W_1 of ϕ_1 to the 500 K data. *retained*: the change of $\langle E\rangle$ between the two training sets as a fraction of the 0.38 eV shift of the data; 0% recovers equilibrium. The *DFT* rows are the data themselves: the 500 K row gives the estimators' reading of the ground truth and its floor at the sample size, the 600 K row what reproducing the shifted training data would score. ENTROPICA (FT) finetunes ENTROPICA from its kept checkpoint with 5× the HJB weight and 2× the PES boundary weights, which lowers the relative weight of denoising score matching.

	valid (%)	$\Delta\langle E\rangle$ (meV)	T_{eff} (K)	ϕ_1 width (deg)	JS (nats)	W_1 (deg)	retained (%)
trained on 500 K data (ground truth)							
DSM	100.0	+220	639 ± 9	27.8	0.001	0.9	–
DSM + HJB	100.0	+450	761 ± 15	29.1	0.004	1.6	–
DSM + boundary	100.0	+150	612 ± 11	29.6	0.004	2.0	–
ENTROPICA	100.0	+11	575 ± 11	31.8	0.010	3.4	–
ENTROPICA (ft)	100.0	-29	565 ± 10	32.1	0.011	4.0	–
DFT, 500 K (GT)	100.0	+0	557	27.1	0.001	0.5	–
trained on 600 K data							
DSM	100.0	+584	744 ± 11	31.1	0.007	3.0	95 ± 3
DSM + HJB	99.9	+892	913 ± 18	33.6	0.015	4.8	115 ± 4
DSM + boundary	100.0	+552	739 ± 13	34.8	0.017	5.8	104 ± 3
ENTROPICA	100.0	+158	616 ± 11	34.6	0.018	5.5	38 ± 3
ENTROPICA (ft)	100.0	+13	569 ± 9	33.4	0.014	4.7	11 ± 2
DFT, 600 K (train)	100.0	+385	672	31.4	0.005	2.8	100

600 K. The retained fraction is the change in a model's mean referee energy between the two training sets, divided by this shift; errors are standard errors over 2,000 samples.

Ablation. At matched weights the retained fractions form a 2×2 design: score matching alone retains 95 ± 3 %, with the HJB residual 115 ± 4 %, with the boundary conditions 103 ± 3 %, and with both 38 ± 3 %. The interaction, -86 ± 6 percentage points, accounts for the entire recovery. Raising the HJB weight five-fold and the boundary weights two-fold, warm-started from the converged full model, gives 11 ± 2 % (ENTROPICA (ft)). Fitting the density of states to the sampled energies gives effective temperatures of 744 ± 11 K for the DSM model and 569 ± 9 K for ENTROPICA (ft), against 557 K for the equilibrium frames themselves.

C.5 ADDITIONAL ANI-1X RESULTS

ANI-1x groups conformers by molecular formula, which merges distinct constitutions. We reassign every conformer to a molecule by bond perception, split by constitution and keep only connected molecules. This leaves 51,008 training molecules and 1,631 held-out molecules (143,668 conformers), and no held-out molecule appears in training in any conformation. Energies are centred per molecule: absolute ANI-1x energies are dominated by the atomic reference, so any model that identifies the composition reaches $R^2 \approx 1$. Similarity to training is the maximum Tanimoto similarity over Morgan fingerprints (radius 2, 2048 bits).

Table 8: Energy and force accuracy on 128,570 conformers of 1,631 held-out ANI-1x molecules, without the 15,098 on which MMFF94 diverges. Energies are centred per molecule, or per molecular formula, which mixes isomers that share a composition but not an electronic energy. *DSM only* trains the score alone, so its energy head is untrained.

	energy, centred (meV)			forces	
	per molecule		per formula		
model	MAE	median	MAE	MAE (eV/Å)	cos median
Entropica	222	169	300	0.247	0.9933
ablation					
DSM only	1335	1086	1489	1.150	0.9579
conventional force field					
MMFF94	255	161	809	0.519	0.9784

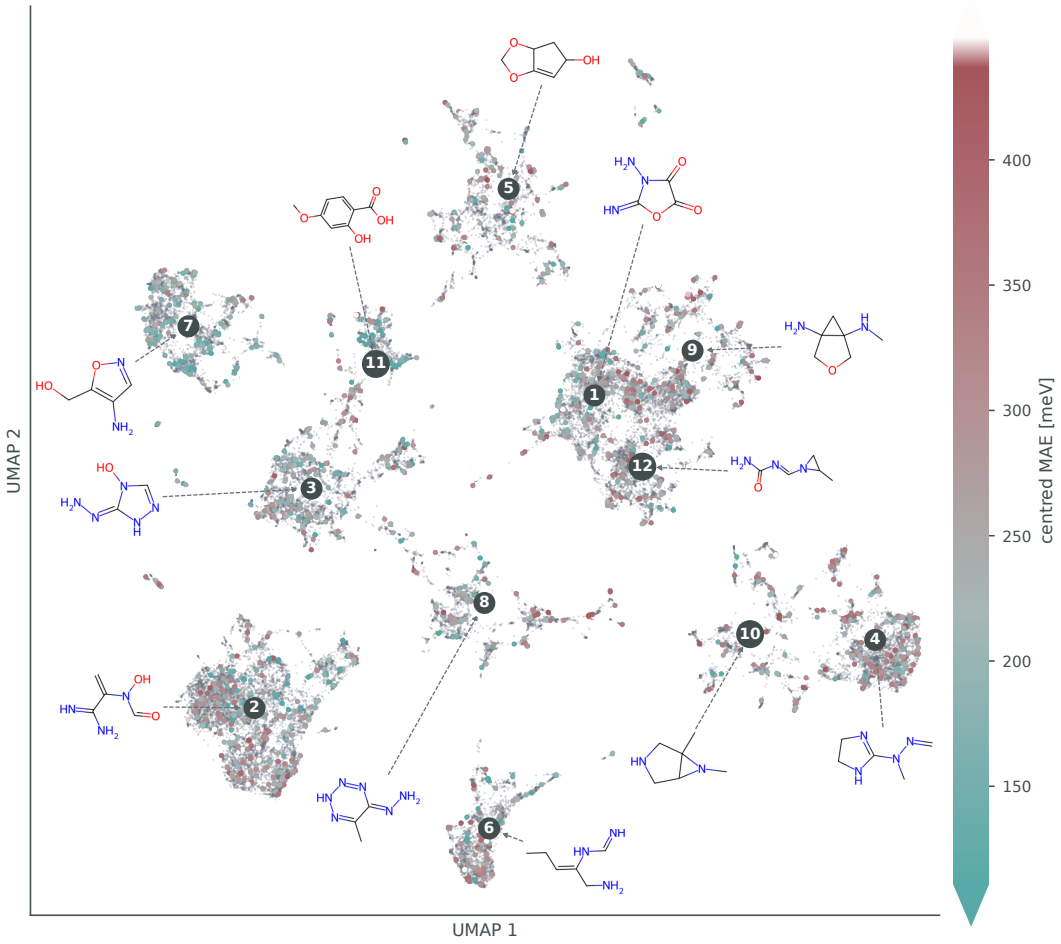


Figure 9: Held-out error across ANI-1x chemical space. Molecules are embedded with ChemNet descriptors (512 dimensions), reduced by PCA to 100 components and projected with UMAP (15 neighbours, minimum distance 0.1). Small grey points: 51,008 training molecules. Colored points: held-out molecules, by per-molecule centred energy MAE. Numbered discs mark twelve regions from k-means on the UMAP coordinates, annotated with representative molecules. Each group is characterized by its most enriched functional groups: aromatic heterocycles (3, 7, 8), amides (1, 2, 12), ethers (5, 9), amines (4, 6, 10) and aromatic carbocycles (11). High errors are spread over multiple regions, although low-error clusters exist (11).

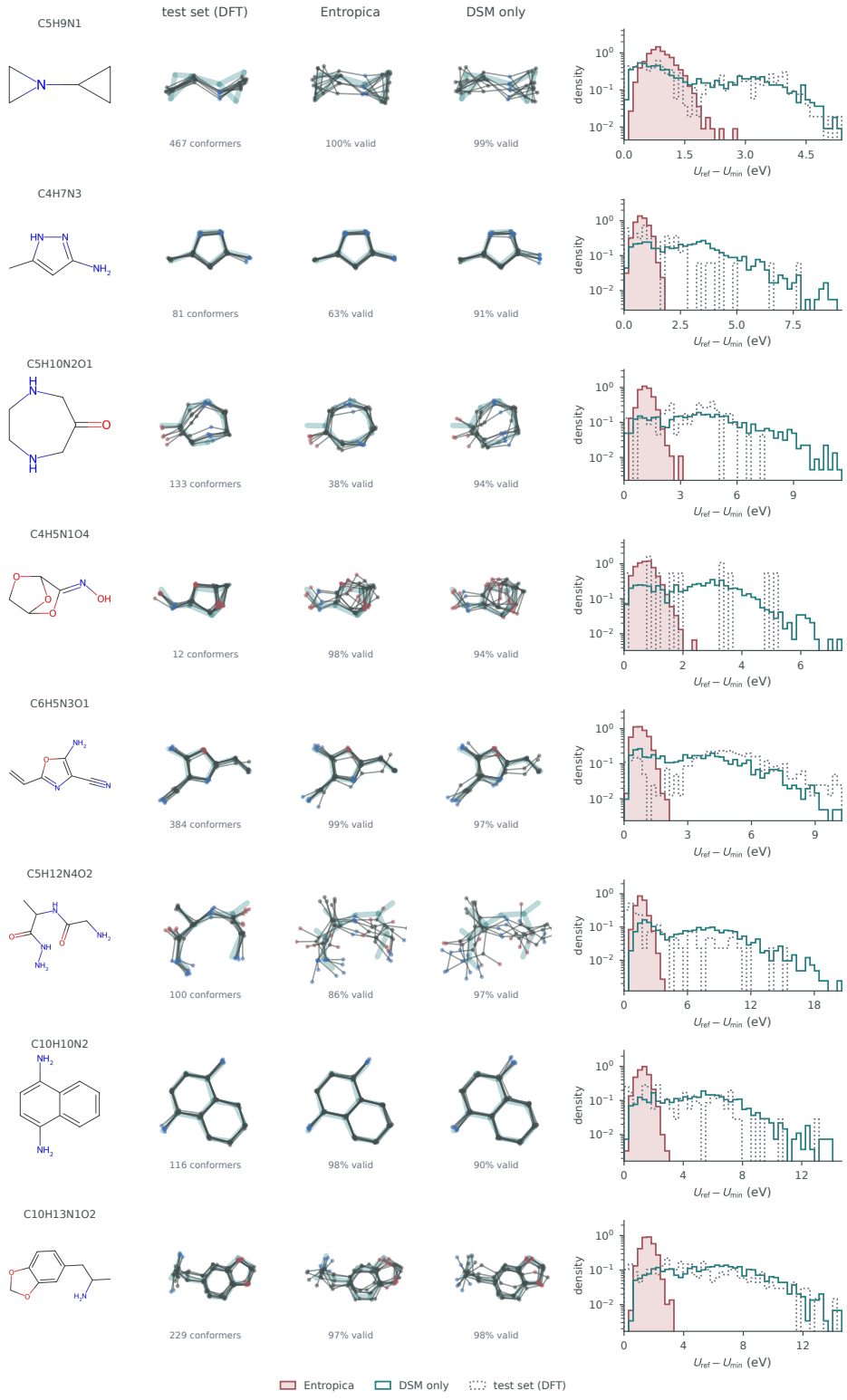


Figure 10: Generated conformers of eight exemplary held-out ANI-1x molecules. For each of the molecules, its structure; six randomly drawn conformers from the test set (DFT), and six valid conformers (Appendix B.5) from ENTROPICA and from the DSM-only ablation, with the valid fraction beneath; heavy atoms only, superposed on the held-out conformer the models were conditioned on (pale teal); and the energy distributions of the valid generated and test-set conformers (dotted), scored by ANI-2x (E_{ref}) relative to the lowest of them.