Energy Loss Functions for Physical Systems

Sékou-Oumar Kaba*, **Kusha Sareen***, **Daniel Levy, Siamak Ravanbakhsh**
McGill University
Mila - Quebec Artificial Intelligence Institute

Abstract

Effectively leveraging prior knowledge of a system's physics is crucial for applications of machine learning to scientific domains. Previous approaches mostly focused on incorporating physical insights at the architectural level. In this paper, we propose a framework to leverage physical information directly into the loss function for prediction and generative modeling tasks on systems like molecules and spins. We derive *energy loss functions* assuming that each data sample is in thermal equilibrium with respect to an approximate energy landscape. By using the reverse KL divergence with a Boltzmann distribution around the data, we obtain the loss as an energy difference between the data and the model predictions. This perspective also recasts traditional objectives like MSE as energy-based, but with a physically meaningless energy. In contrast, our formulation yields physically grounded loss functions with gradients that better align with valid configurations, while being architecture-agnostic and computationally efficient. The energy loss functions also inherently respect physical symmetries. We demonstrate our approach on molecular generation and spin ground-state prediction and report significant improvements over baselines. Code is available at `https://github.com/kushasareen/energy_loss`.

1 Introduction

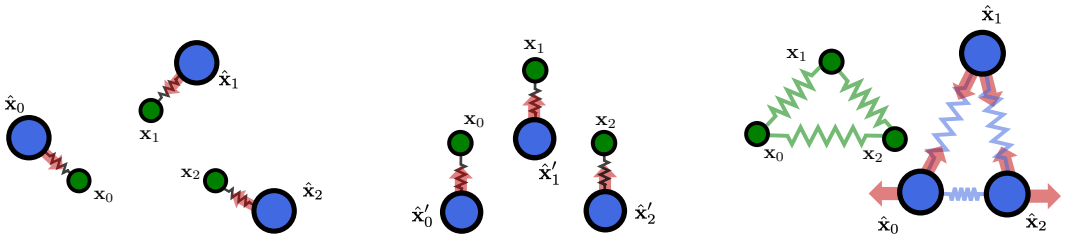
A key challenge in applications of machine learning to the physical sciences is that data can often be scarce and expensive to generate. However, we often have some prior knowledge of the physics of the system of interest, which can be used to design useful inductive biases. A common learning problem involves training a machine learning model to predict configurations of physical systems based on data collected close to equilibrium such as protein folding [Noé et al., 2020, Jumper et al., 2021, Abramson et al., 2024], crystal structure prediction [Ryan et al., 2018, Jiao et al., 2023, Zeni et al., 2025], calculation of ground states given Hamiltonian parameters [Carrasquilla and Melko, 2017], or generative modeling of physical systems [Gómez-Bombarelli et al., 2018, Sanchez-Lengeling and Aspuru-Guzik, 2018]. A significant body of work has focused on implementing physical inductive biases, such as equivariance at the level of architectures (see e.g. Zhang et al. [2023] for a review).

This work explores a complementary direction: embedding physical principles directly into the loss function. The fundamental question we ask is: can loss functions grounded in physical principles provide more effective training signals and yield models that better reflect physically valid configurations compared to generic losses such as the mean-squared error (MSE) and the cross-entropy loss?

As a response, we propose a framework for deriving *energy loss functions* tailored for physical systems in the thermal equilibrium regime. This is motivated by the fact that loss functions can be obtained from a distribution representing the uncertainty around each prediction or data sample. For physical systems in thermal equilibrium, the sensible choice is the Boltzmann distribution. Employing the reverse Kullback-Leibler (KL) divergence leads to loss functions that take the form of approximate

* Equal contribution. Correspondence: kabaseko@mila.quebec.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) MSE loss

(b) Configuration with same MSE loss as (a)

(c) Energy loss

Figure 1: Energy interpretation of loss functions. Ground truth positions are denoted in green and predictions in blue. **(a)** The MSE loss function for particle positions corresponds to quadratic potential energy centered on the data. **(b)** This choice is however physically unsound and leads to penalizing the model for configurations that are correct, i.e. related by rigid motion to the target. **(c)** A more accurate choice would be to use a loss function based on physically sound energy, which would not suffer from the aforementioned problem.

energy differences between data and predictions. This allows for a more principled quantification of the errors made by the model, which we hypothesize provides better gradients for learning.

Our framework is general in the sense that it encompasses many existing loss functions and allows us to interpret them as energies. The energy loss functions also naturally capture relevant symmetries if the underlying energy approximation does. Specifically, they make it so that no loss is incurred by the model for predicting configurations that are related to the data by symmetry. Loss functions that have this property have been suggested for atomistic systems, but they require expensive alignment or minimization procedures [Klein et al., 2023], which our framework does not require. This framework finds broad applicability to systems in thermal equilibrium, from direct regression tasks to generative modeling with diffusion models [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song et al., 2021]. Note that we consider tasks that can be framed as regression and classification problems with data, not sampling problems where we are given a ground-truth energy (like for example in Boltzmann generators [Noé et al., 2019]).

Contributions: **(i)** Methodology for deriving loss functions grounded in physical principles by minimizing the reverse KL divergence between a prediction and a Boltzmann distribution centered around data **(ii)** Instantiation of this framework for atomistic systems that yield distance-based loss functions and an analysis of the invariance properties of these losses **(iii)** Applications to diffusion models and analysis of the resulting score estimator **(iv)** Instantiation of this framework for spin systems **(v)** Empirical evaluation on a range of tasks, showing consistent improvement over baselines.

2 Background

2.1 Forward and reverse KL loss functions

We first consider a regression setting. Consider the empirical distribution $p_{\mathcal{D}}$ associated with the IID dataset $\mathcal{D} = \left\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\right\}_{i\in[N]}$, and a parametric model $f_\theta : \mathbb{R}^d \to \mathbb{R}^k, \mathbf{x} \mapsto \hat{\mathbf{y}}$ associated with the family of conditional distributions $p\left(\mathbf{y} \mid f_\theta\left(\mathbf{x}\right)\right)$. We take $f_\theta\left(\mathbf{x}\right)$ to be the model prediction of the target; the usual assumption is that conditional distribution is parametrized by a location parameter (like the mean for a Gaussian), and the model is trained to maximize the likelihood of the data $\mathcal{L}\left(\theta\right) = -\sum_i^N \log p\left(\mathbf{y}^{(i)} \mid f_\theta\left(\mathbf{x}^{(i)}\right)\right)$. The Gaussian assumption for the model results in the Mean Squared Error (MSE) loss function. For n-way classification, the model predicts the logits of a categorical distribution and maximum likelihood yields the cross-entropy loss function. Maximizing the likelihood is equivalent to minimizing the Kullback-Leibler (KL) divergence.

In this work, we will consider a reverse KL divergence objective. In the regression case, this amounts to taking the model as deterministic and instead accounting for the uncertainty at the level of the data samples. For regression, we then have $p_{\mathcal{D}}\left(\mathbf{x}, \mathbf{y}\right) = \sum_i^N \frac{1}{N}\delta\left(\mathbf{x} - \mathbf{x}_i\right) p\left(\mathbf{y} \mid \mathbf{y}^{(i)}\right)$ and $q\left(\mathbf{x}, \mathbf{y}\right) = \sum_i^N \frac{1}{N}\delta\left(\mathbf{x} - \mathbf{x}^{(i)}\right)\delta\left(\mathbf{y} - f_\theta\left(\mathbf{x}^{(i)}\right)\right)$, where $p\left(\mathbf{y} \mid \mathbf{y}^{(i)}\right)$ is a conditional distribution that specifies the uncertainty around each target. The reverse KL objective is then

$$D_{\mathrm{KL}}(q \parallel p_{\mathcal{D}}) = \mathbb{E}_q\left[\log q\left(\mathbf{x}, \mathbf{y}\right) - \log p_{\mathcal{D}}\left(\mathbf{x}, \mathbf{y}\right)\right] = -\sum_i^N \log p\left(f_\theta\left(\mathbf{x}^{(i)}\right) \mid \mathbf{y}^{(i)}\right) \tag{1}$$

For classification, the model distribution is still a categorical distribution parametrized by logits to ensure differentiability, but the distribution associated with data samples can be general.

In general, the reverse KL divergence is not equal to the forward KL divergence. Instead, it gives the likelihood of the prediction given an uncertainty model for targets. However, it is exactly equal when the sample point and the parameter can be swapped in the distribution p. It is for example the case when p is chosen as Gaussian. Our general goal will be to define more appropriate distributions $p(\hat{\mathbf{y}} \mid \mathbf{y})$ for the loss function. As we will see, the reverse KL formulation is convenient since it enables defining these distributions only around each data sample.

2.2 Diffusion models

We also consider generative modeling with diffusion models as another use case for more informed energy loss functions. This class of generative models has proven powerful, as they can efficiently learn interpolations between a prior distribution and the data distribution [Albergo et al., 2023]. The objective is typically formulated as a noise prediction task [Ho et al., 2020]

$$\mathcal{J}(\theta) = \int_0^1 \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}), \boldsymbol{\epsilon}_t \sim p(\boldsymbol{\epsilon}_t)} \left[w_t \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_\theta\|^2 \right] \mathrm{d}t \tag{2}$$

where the noise prediction is the output of a neural network $\hat{\boldsymbol{\epsilon}}_\theta = f_\theta(\mathbf{x}_t, t)$, $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$, σ_t, α_t define the noise schedule and w_t is a weighting factor. In practice, the expectation is estimated by Monte Carlo. The objective also admits an interpretation as denoising score matching [Vincent, 2011], with the optimal noise prediction satisfying $\epsilon^*(\mathbf{x}_t, t) = -\sigma_t \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$.

The loss can be equivalently seen as prediction of the data sample, with appropriate reweighting, see e.g. Kingma and Gao [2024]. With the sample prediction defined as $\hat{\mathbf{x}}_\theta \equiv \frac{(\mathbf{x}_t - \sigma_t \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t))}{\alpha_t}$, we have:

$$\mathcal{J}(\theta) = \int_0^1 \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}), \boldsymbol{\epsilon}_t \sim p(\boldsymbol{\epsilon}_t)} \left[\frac{w_t \alpha_t^2}{\sigma_t^2} \|\mathbf{x} - \hat{\mathbf{x}}_\theta\|^2 \right] \mathrm{d}t \tag{3}$$

yielding a regression-type objective with the MSE loss.

3 Energy Loss Functions

In Section 2.1, we saw that loss functions can be obtained through a reverse KL formulation with respect to a conditional distribution $p(\hat{\mathbf{y}} \mid \mathbf{y})$ centered on the data. Importantly, the conditional distribution p is always an uncertainty model; as such, there is not necessarily a *true* one.

We will define the conditional distribution p as a Boltzmann distribution

$$p(\hat{\mathbf{y}} \mid \mathbf{y}) = \frac{\exp\left(-E(\hat{\mathbf{y}}, \mathbf{y})/T\right)}{Z(\mathbf{y}, T)} \tag{4}$$

where $E : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ is related to the *physical* potential energy of the system around the data point $\mathbf{y}$, T is the temperature and $Z(\mathbf{y}, T)$ is the partition function. We assume the system is observed in physically likely configurations; hence, each data point $\mathbf{y}$ is modeled as an approximate local minimum in the energy landscape. The use of Boltzmann distributions to model the uncertainty around such configurations is natural and can be motivated from first principles [Jaynes, 1957, Pathria, 2017]. It is the steady-state distribution of a system undergoing stochastic dynamics in contact with a reservoir at temperature T (see derivation in Appendix A.2).

Assuming a general Boltzmann distribution, the reverse KL divergence Equation (1) loss function we obtain for the continuous case is

$$\mathcal{J}(\theta) = -\sum_i^N \log p\left(\hat{\mathbf{y}}_\theta^{(i)} \mid \mathbf{y}^{(i)}\right) = \sum_i^N \frac{E(\hat{\mathbf{y}}_\theta^{(i)}, \mathbf{y}^{(i)})}{T} + \log Z(\mathbf{y}^{(i)}, T), \tag{5}$$

where the log-partition function does not depend on the parameters. The model is penalized for errors by an amount given by the approximate increase in energy with respect to the data sample.

This picture allows for obtaining a physical interpretation of different conditional distributions and loss functions depending on the choice of energy $E(\hat{\mathbf{y}}, \mathbf{y})$. The Gaussian conditional distribution is

obtained with $T = 2\sigma^2$ and isotropic harmonic potential energy centered around $\mathbf{y}$:

$$E(\hat{\mathbf{y}}, \mathbf{y}) = \|\hat{\mathbf{y}} - \mathbf{y}\|^2. \tag{6}$$

We can now justify our choice of the reverse KL estimation over maximum likelihood estimation. First, in the reverse KL case, the partition function, which could be challenging to evaluate for some energies, does not depend on the model parameters θ. Second, we only need to define potential functions around each data sample $\mathbf{y}^{(i)}$, rather than around each prediction. This is a significant advantage, as we can expect some predictions to be poor, leading to configurations that are not approximate equilibria and to nonsensical energies.

3.1 Desiderata for energy functions

There is considerable freedom in the choice of the energy function. One fundamental criterion is agreement with the system's underlying physics, but this is not the only one. An appropriate energy should, in addition, satisfy the following desiderata:

1. **Minimized at the data and symmetries:** The minimizer of the energy function $E(\hat{\mathbf{y}}, \mathbf{y})$ should be the data $\mathbf{y}$ (and its symmetry equivalents). Many tasks require regressing to the data even if it is not the minimum of the true energy landscape.
2. **Optimization stability:** The gradient of the energy function $\nabla_{\hat{\mathbf{y}}} E(\hat{\mathbf{y}}, \mathbf{y})$ should be smooth and bounded to ensure well-behaved optimization with gradient-based methods.
3. **Fast evaluation:** Evaluation of the energy and its derivative should be efficient and compatible with automatic differentiation.

Based on this, we argue that one *should not* often use the true energy function, even if it is known, since it may violate all the criteria. The energy landscapes of systems of interest typically admit multiple local minima and are highly rugged [Mézard et al., 1987, Frauenfelder et al., 1991, Wales et al., 2000]. The cost of evaluating the energy can also be prohibitive [Schuch and Verstraete, 2009].

4 Energies for Atomistic Systems

We consider energies associated with the positions of n atoms in d dimensions, such that $\hat{\mathbf{y}}, \mathbf{y} \in \mathbb{R}^{n \times d}$. The potential energy Equation (6) leading to the Gaussian distribution is poorly motivated from the physical point of view. It describes the effect of an external force bringing back particles to position $\mathbf{y}$. However, a realistic potential energy should model *interactions* between particles (see Figure 1c).

Many approximations exist for the potential energies of physical systems around equilibrium. For atomic systems, the Morse potential [Morse, 1929] and the Lennard-Jones potential [Lennard-Jones, 1931] are examples of popular models. However, using these potentials for the loss Equation (5) can pose challenges for optimization, as they have highly nonlinear gradients that can explode or vanish. A simple approximation that avoids this issue and that is much more principled than the MSE potential is to use a quadratic pair potential of the form

$$E(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i,j}^{n} \frac{1}{2} k_{ij}(\mathbf{y}) \left(\|\mathbf{y}_i - \mathbf{y}_j\| - \|\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j\| \right)^2 \tag{7}$$

where the indices i, j are taken over particles. This is the general form of a second-order Taylor approximation in pairwise distances of an interaction potential (see Appendix A.3). Motivated by the fact that coordinate regression can lead to poor realism due to inconsistencies with bond lengths, this type of distance-dependent loss function has been used heuristically as a regularizer in some applications [Peng et al., 2023, Yang and Gómez-Bombarelli, 2023, Abramson et al., 2024], but to the best of our knowledge, not as a primary objective. Note that this is different from directly predicting the distances [Simm and Hernández-Lobato, 2019, Nesterov et al., 2020, Xu et al., 2021].

There is significant freedom in the choice of the coefficients $k_{ij}(\mathbf{y})$. We propose simple heuristics to set these coefficients. First, we consider setting the coefficients are set to a constant value $k_{ij}(\mathbf{y}) = k$. Note that this can be obtained from Taylor approximation of the Morse potential (see Appendix A.3). Second, we consider setting the coefficient as a decreasing function of the distance between two atoms $k_{ij}(\mathbf{y}) = f(\|\mathbf{y}_i - \mathbf{y}_j\|)$ to capture the fact that interactions between particles decrease at long

range. We consider inverse, inverse-squared, and exponential decay dependence of f on the distance. Taylor approximation of the Lennard-Jones potential yields inverse squared distance dependence (see Appendix A.3). Other possibilities can be considered: in general, given an interaction potential between particles, the coefficients can be obtained by a second-order Taylor expansion.

4.1 Invariance properties

An important property of energy loss functions is that they respect the symmetries of the associated physical energy. We formalize this in the following way:

Definition 4.1 (Invariant loss function)**.** A loss function $l : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$ between a prediction and a target is invariant to the action of the group G on $\mathbb{R}^k$ if

$$l\left(g \cdot \hat{\mathbf{y}}, \mathbf{y}\right) = l\left(\hat{\mathbf{y}}, g \cdot \mathbf{y}\right) = l\left(\hat{\mathbf{y}}, \mathbf{y}\right), \quad \forall g \in G, \hat{\mathbf{y}}, \mathbf{y} \in \mathbb{R}^k \tag{8}$$

An invariant loss function essentially compares input and targets up to transformations in G. An example of a common $SE(3)$-invariant loss function is to apply the Kabsch algorithm [Kabsch, 1976] to find an optimal alignment between a predicted structure and a target, and to use the MSE after applying the alignment [Klein et al., 2023]. It has been shown that in cases where there are multiple possible symmetry-related predictions for a given input– so-called symmetry-breaking predictions [Smidt et al., 2021, Kaba and Ravanbakhsh, 2023]– non-invariant loss functions exhibit pathological behaviour [Xie and Smidt, 2024, Jing et al., 2024, Lawrence et al., 2025]. For example, the MSE is minimized when the prediction is the mean of the possible targets rather than for any of them. There are multiple ways to define these invariant losses, which are analogous to the different ways in which invariant neural networks can be designed (see Appendix A.4 for more discussion).

It is easy to see that the energy loss Equation (5) is invariant to $G = E(d)$ if $k_{ij}\left(\mathbf{y}\right)$ is invariant, since it then only depends on invariant distances. This is analogous to how invariant functions can be built from scalars Villar et al. [2021]. These, however, are not the only symmetries of the loss. The energy loss function is additionally invariant to permutations that correspond to the symmetries of the ground-truth distance matrix. Denote the distance matrix of the data by $\Delta y_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|$ and the automorphism group of a matrix $m \in \mathbb{R}^{n \times n}$ as $\mathrm{Aut}\left(m\right) \subseteq S_n$ where the automorphisms act on the matrix by conjugation. We then have the following:

Proposition 4.2. *The loss function Equation* (7) *is invariant to the group*

$$G = E(d) \times \left(\mathrm{Aut}\left(k\left(\mathbf{y}\right)\right) \cap \mathrm{Aut}\left(\Delta y\right)\right). \tag{9}$$

All the proofs follow in Appendix A.1. This allows us to characterize the family of loss minimizers:

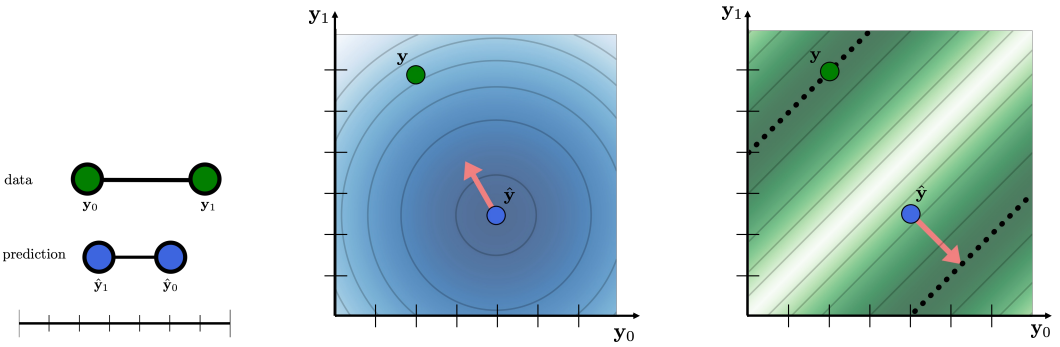
Corollary 4.3. *For any* $\mathbf{y} \in \mathbb{R}^{n \times d}$ *and* $k_{ij}\left(\mathbf{y}\right) > 0$*,*

$$\underset{\hat{\mathbf{y}} \in \mathbb{R}^{n \times d}}{\arg\min}\, E(\hat{\mathbf{y}}, \mathbf{y}) = \left\{g \cdot \mathbf{y} \;\middle|\; g \in G\right\}. \tag{10}$$

The loss landscape, therefore, presents a family of global minimizers associated with symmetries. We hypothesize that the symmetry is beneficial for learning, since it allows the model to regress to any target that is equivalent to the data by symmetry, as shown in the example in Figure 2. As our experimental results show, the benefits of invariance in the loss function are different and complementary to that of equivariance of the architecture. Equivariance guarantees that the output changes predictably under transformations of the input. However, it does not guarantee that the correct output will be learned. Invariant loss functions make the learning task easier by allowing to regress to any symmetry related configurations, which equivariance with a non-invariant loss does not allow.

4.2 Diffusion models with distance-based loss functions

The training objective of diffusion models involves the prediction of a data sample from a noisy latent one. We suggest that the energy loss functions can be used as a straightforward replacement for the MSE in these objectives. Similar distance-based objectives have been previously used in the context of diffusion models [Yang and Gómez-Bombarelli, 2023, Abramson et al., 2024, Cognolato et al., 2025]. However, it is not immediately clear that using such objectives results in learning correct score estimates. Here, we show that this is indeed the case under some conditions.



(a) MSE loss **(b)** Energy loss

Figure 2: Loss landscapes. The model has to predict the positions of two particles in one dimension. The prediction for the first particle $\hat{\mathbf{y}}_0$ is closer to the ground-truth for the second particle $\mathbf{y}_1$ and vice-versa. **(a)** The MSE minimizes the forward KL divergence between a Gaussian model distribution (blue) and the data distribution (green). It does not capture the symmetry. **(b)** The energy loss is obtained via the reverse KL with the pair energy and admits a family of minimizers associated with symmetries. It results in a gradient that points towards the closest correct configuration.

Consider the energy loss function Equation (7) with constant coefficients $k_{ij}(\mathbf{x}) = k$. The loss function computes the MSE between distance matrices for the data and the sample prediction, given by $\Delta x_{ij} = d(\mathbf{x}) = \|\mathbf{x}_i - \mathbf{x}_j\|$. Denote the Jacobian of this function by $J(\mathbf{x}) = \nabla_{\mathbf{x}} d(\mathbf{x}) \in \mathbb{R}^{dn \times n^2}$. We will seek to characterize the minimizers of a diffusion model trained using this loss,

$$\hat{\boldsymbol{\epsilon}}^* \in \underset{\hat{\boldsymbol{\epsilon}} \in \mathbb{R}^{n \times d}}{\arg\min}\ \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}_0), \boldsymbol{\epsilon}_t \sim p(\boldsymbol{\epsilon}_t)} \left[E\left(\frac{\mathbf{x}_t - \sigma_t \hat{\boldsymbol{\epsilon}}(\mathbf{x}_t, t)}{\alpha_t}, \mathbf{x} \right) \right] \tag{11}$$

The following result allows us to obtain an approximation of this set, valid for small noise scales:

Proposition 4.4. *Let $p(\mathbf{x}_t)$ be a continuously differentiable, $SE(d)$-invariant density. Assume $|\hat{\boldsymbol{\epsilon}}|$ is bounded. For small σ_t,*

$$\hat{\boldsymbol{\epsilon}}^* \approx -\sigma_t \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \mathbf{v}, \quad \mathbf{v} \in ker\left(J(\mathbf{x}_t)\right) \tag{12}$$

In addition, the minimum norm minimizer $\hat{\boldsymbol{\epsilon}}^*_{dist}$ is given by

$$\hat{\boldsymbol{\epsilon}}^*_{dist} \approx -\sigma_t \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \tag{13}$$

The set of minimizers is therefore given by the true score, up to a translation in the direction of rigid motions. The second fact follows since for an invariant measure, the score is orthogonal to the Lie algebra generators. We can also show that due to its invariance, the distance-based loss function offers a reduction in variance with respect to the MSE:

Proposition 4.5. *Let $p(\mathbf{x}_t)$ be a continuously differentiable, $SE(d)$-invariant density. Denote by $\hat{\boldsymbol{\epsilon}}^*_{dist}$ and $\hat{\boldsymbol{\epsilon}}^*_{MSE}$ the minimum norm minimizers of the Monte-Carlo estimators of the energy loss and MSE loss, respectively. For small σ_t,*

$$Bias\left[\hat{\boldsymbol{\epsilon}}^*_{dist}\right] \approx 0, \qquad Var\left[\hat{\boldsymbol{\epsilon}}^*_{dist}\right] \lesssim Var\left[\hat{\boldsymbol{\epsilon}}^*_{MSE}\right] \tag{14}$$

Note that surprisingly, even though these results are in principle only valid for the energy loss function with constant coefficient $k_{ij}(\mathbf{x})$, in our results of Section 6.2, the variants using more physically motivated coefficients still performed better empirically. We hypothesize that this is because the evaluation of the model assesses the physical plausibility of the samples rather than the agreement between the learned and data distributions. Incorporating physical information in the loss function can therefore be beneficial, even though (or because) it biases the score estimate.

4.3 Linear scaling and rigidity theory

One potential downside of using the energy loss of Equation (7) is that it has a quadratic number of terms in the number of particles N, in contrast to the linear number of terms in typical losses such as MSE loss. While in many architectures– such as transformers or densely connected graph neural networks– the quadratic cost of operations in the network makes this a non-issue, the feasibility of linear scaling may prove valuable for applications involving a large number of particles, e.g., modelling macromolecules or crystals with large unit cells.

Fortunately, a solution is provided by rigidity theory. Results in rigidity theory [Laman, 1970, Asimow and Roth, 1978] provide the conditions for recovering the coordinates of a point cloud from a *linear* number of pairwise distances. In this work, we consider a construction for sparse rigid graphs, reducing the computational cost of energy loss without affecting its global optima (see Appendix A.5 for more background and Appendix C.4 for wall-times of different loss calculations).

5 Energy Loss for Discrete Systems

The energy loss formulation can be leveraged for other types of systems. We derive a version for the discrete case, which can replace the cross-entropy loss function. Denote logits predictions as $\mathbf{z}_{\theta,j}^{(i)}$ and the associated categorical distribution as $q\left(\hat{\mathbf{y}} \mid \mathbf{z}_{\theta}^{(i)}\right)$. The reverse KL between the model distribution and a Boltzmann distribution around the data Equation (4) is given by

$$\mathcal{J}(\theta) = \frac{1}{T}\sum_{i}^{N}\left[\mathbb{E}_{q\left(\hat{\mathbf{y}}|\mathbf{z}_{\theta}^{(i)}\right)}\left[E\left(\hat{\mathbf{y}}, \mathbf{y}^{(i)}\right)\right] - TS\left[q\left(\hat{\mathbf{y}} \mid \mathbf{z}_{\theta}^{(i)}\right)\right] + T\log Z\left(\mathbf{y}^{(i)}, T\right)\right] \tag{15}$$

where $\mathbf{z}_{\theta,j}^{(i)}$ is the model prediction for the logits associated with class j and $S[q]$ is the entropy of q. The loss is therefore proportional to the free energy difference. The last term is the negative free energy at the data, and does not depend on the parameters. The loss function, therefore, simply reduces to the variational free energy of the prediction.

5.1 Application to spin systems

We consider modeling systems of spins as an application of the discrete formulation. Predicting configurations of these systems with machine learning models is a problem of high interest in physics [Carrasquilla and Melko, 2017, Pahng and Brenner, 2020] and in combinatorial optimization [Fu and Anderson, 1986]. We will be interested specifically in systems on a square lattice Λ such that $\hat{\mathbf{y}}, \mathbf{y} \in \{1, -1\}^{\Lambda}$. We consider Ising-type Hamiltonians of the form $E(\mathbf{y}) = -\frac{1}{2}\sum_{ij}^{\Lambda} J_{ij}\mathbf{y}_i\mathbf{y}_j$ where the coupling $-1 \leq J_{ij} \leq 1$ is non-zero only for neighboring sites in the lattice Λ, but does not necessarily exhibit any symmetry. Systems with unstructured couplings are known as spin glasses [Mézard et al., 1987] and often exhibit a large number of local energy minima.

Energy loss functions of the form Equation (15) can be used for classification of spin configurations. We suggest to use an approximate local energy around the data defined as

$$E(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i}^{\Lambda} h_i^{\mathrm{LF}}(\mathbf{y})\,\hat{\mathbf{y}}_i \tag{16}$$

where the local field is given by $h_i^{\mathrm{LF}}(\mathbf{y}) = \sum_{j}^{\Lambda}(J_{ij} + h^0)\mathbf{y}_j$. The local field energy captures the change in energy from flipping a spin in the configuration $\mathbf{y}$. It therefore provides an appropriate way to quantify deviations from that configuration: large values of local field are associated with spins that result in large increases of energy and that should be weighted more importantly in the loss.

An alternative would be to use the true energy instead. The objective Equation (15) would then be interpreted as entropy-regularized energy minimization. This would be expected to perform well in strict terms of minimizing the energy. However, the true energy is not a classification objective, since it does not make use of the data. In addition, it can exhibit a large number of local minima. By contrast, the local energy loss Equation (16) is convex, due to the linear dependence in $\hat{\mathbf{y}}$. It also admits the data point $\mathbf{y}$ as its unique minimum if $h^0 > 4$ (see Appendix C.3). If the data is a ground state of the true energy, $h^0 > 0$ is sufficient. The energy loss is therefore a proper classification objective.

6 Experiments

6.1 Regular shape prediction

Experiment Setup. To develop an understanding of energy loss functions, we propose a simple task where the goal is to generate regular shapes in two dimensions. Given a radius, a model is tasked

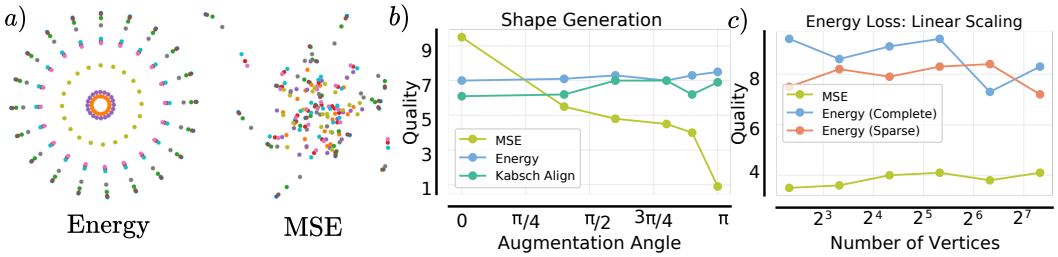


Figure 3: Regular shape prediction results. **(a)** Typical samples from optimal models trained with MSE and energy loss when $\theta_{aug} = \pi$. **(b)** The impact of θ_{aug} on sample quality. We can see as θ_{aug} increases, MSE performance deteriorates but the invariant losses (Energy and Kabsch Align) remain performant. **(c)** As the number of shape vertices scales, a sparse version of the energy loss remains equally performant as a complete-edge energy loss using only $O(N)$ operations.

with predicting the N vertices of a regular polygon of that radius. The dataset is constructed by sampling regular polygons of a radius $r \sim U[0.3, 5]$ and then applying an augmentation by randomly rotating the shape by an angle in $U[-\theta_{\text{aug}}, \theta_{\text{aug}}]$. Prediction is performed using two hidden-layer MLP. We compare standard MSE loss with the atomic energy loss using exponential coefficients, an SE(2)-invariant loss using the Kabsch algorithm to align points, and a version of the energy loss using sparse rigid graphs. We empirically confirm that the sparse graphs are globally rigid w.h.p. in Appendix A.5. To evaluate, we introduce a quality metric based on the regularity of the angular differences and the radial variation in a given shape. Intuitively, for a regular shape the angular difference variation $\sigma_{\Delta_{angle}}$ and the radial variation σ_{radius} across points should be small. A full definition follows in Appendix C.1.

Results. Figure 3 shows that the energy loss and other invariant losses continue to produce high-quality shapes when rotation augmentation is applied whereas MSE fails. Additionally, the sparse energy loss maintains nearly the same performance as the number of vertices N increases, while reducing computation by $O(N)$ operations. Interestingly, models trained with an invariant loss automatically learn to produce canonical orientations of shapes.

6.2 Molecule generation

Table 1: Metrics for GDM-aug on GEOM-Drugs.

Loss	Mol. stab. (%)	Atom stab. (%)	Valid. (%)	Unique (%)
MSE	0.8	85.6	**94.8**	100
Energy	**24.6**	**96.0**	89.7	100

Experiment Setup. First, we train diffusion models to unconditionally generate molecules in the QM9 dataset [Ramakrishnan et al., 2014]. We evaluate the performance of the energy loss when training EGNN diffusion models (EDM) [Hoogeboom et al., 2022], GNN diffusion models with and without data augmentation (GDM and GDM-aug) and near state-of-the-art joint 2D & 3D diffusion models (JODO) Huang et al. [2023]. As baselines, we compare the convergence properties to models trained with MSE and a Kabsch-aligned MSE [Kabsch, 1976]. Exponential coefficients are chosen for the energy loss. Additionally, we compare with a version of the energy loss using sparse rigid graphs.

Table 2: Evaluation metrics for GDM-aug on QM9.

Loss	Molecule stability (%)	Atom stablity (%)	Validity (%)	Uniqueness (%)
GDM-aug				
MSE	83.7 ± 2.3	98.3 ± 0.004	93.6 ± 1.7	100.0 ± 0.0
Kabsch align	82.3 ± 0.5	97.8 ± 0.004	90.8 ± 2.0	100.0 ± 0.0
Energy	**89.8** ± 2.8	**99.3** ± 0.3	**97.7** ± 1.4	99.9 ± 0.002
Energy (sparse)	89.1 ± 0.9	99.0 ± 0.1	97.4 ± 2.5	100 ± 0.0
EDM				
MSE	82.4 ± 3.4	98.8 ± 1.7	93.0 ± 2.5	99.89 ± 0.32

We also generate large molecules with GDM and GDM-aug using the GEOM-Drugs dataset [Axelrod and Gomez-Bombarelli, 2022], comparing the MSE and energy loss. A similar evaluation setup to [Satorras et al., 2022, Hoogeboom et al., 2022] is used for GDM and EDM while JODO uses a

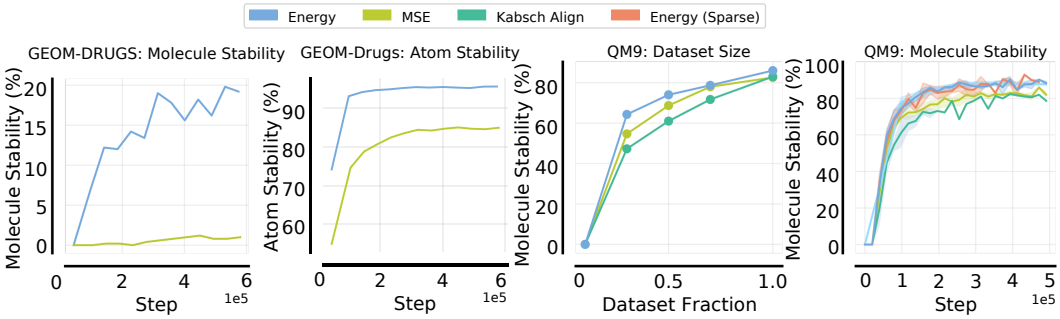


Figure 4: Molecule generation results. **(Left)** We observe a dramatic improvement on stability metrics for the GEOM-Drugs dataset, demonstrating the scalability of our approach. **(Right)** On QM9, energy loss improves metrics over all baselines. This is especially present in the low data regime where energy loss gives +10% molecule stability over MSE.

Table 3: 3D and alignment metrics for JODO variants.

	Metric-3D					Metric-Align		
Model	At. stab. (%)	Mol. stab. (%)	Val. (%)	Compl. (%)	FCD ↓	Bond ↓	Angle ↓	Dihedral ↓
JODO (paper)	99.2	93.4	—	—	0.885	0.1475	0.0121	6.29e-4
JODO (ours)	99.2	92.8	95.6	95.5	**0.854**	0.1218	0.0110	5.91e-4
JODO + Energy (Inv.)	99.4	94.3	97.1	97.0	0.892	0.1125	**0.0046**	**4.95e-4**
JODO + Energy (Exp.)	**99.6**	**96.6**	**98.4**	**98.4**	1.495	**0.0928**	0.0142	4.97e-3

broader set of 3D and align metrics Huang et al. [2023]. Since the MSE is no longer the optimization objective, we no longer report the ELBO, which depends on the MSE. Instead, we evaluate the method using several desirable features of generated molecules relevant to the drug discovery pipeline: atom stability, molecule stability, validity, and uniqueness. Additionally, for JODO, we compute the Maximum Mean Discrepancy (MMD) for bond lengths, bond angles and dihedral angles against the data distribution as well as the Fréchet ChemNet Distance (FCD) Preuer et al. [2018]. For all settings, we conduct exhaustive sweeps for learning rate and the weighting between the loss on positions and atom types. All comparisons are compute-matched.

Results. Figure 4 shows the energy loss results in faster convergence and better optima over baselines[*]. In addition, we observe that energy loss is much more data efficient than baselines, allowing for the training of capable molecular generative models, producing over 75% stable molecules using only 50% of the training set (50K samples). Table 1 contains results on the GEOM-Drugs data. Table 2 shows results on the QM9 data with GDM-aug model and and its equivariant variant EDM, with comprehensive results in Appendix C.2. Importantly, Table 2 shows that energy loss with a non-equivariant architecture results in more improvement than using an equivariant architecture, at negligible computational cost.

The results using the JODO model are reported in Table 3. We observe that using energy loss with JODO is able to improve all align metrics, and nearly all 3D metrics, with comparable FCD, compared to the default Kabsch-aligned loss. This suggests that energy loss can push the state of the art and offers complementary benefits to equivariant architectures.

Ablation. We conduct an ablation over the form of the spring coefficients $k_{ij}(\mathbf{y})$ in the energy loss (Table 4). We consider the following functional forms: constant, inverse distance, inverse square distance, and exponential decay. A thorough sweep over learning rates was conducted. With EDM/GDM, we find exponential decay to give the best empirical results. However, inverse distance coefficients work well for JODO and Appendix C.4 shows a less stark decay works better for the

Table 4: Ablation for coefficients $k_{ij}(\mathbf{y})$.

Coeff.	Mol. stab. (%)	Atom stab. (%)	Valid. (%)
Exp. Dist.	**89.8** $\pm$ 2.8	**99.3** $\pm$ 0.3	**97.7** $\pm$ 1.4
Inv. Sq. Dist.	84.6 $\pm$ 1.8	98.9 $\pm$ 0.2	96.6 $\pm$ 1.5
Inv. Dist.	84.5 $\pm$ 2.1	98.7 $\pm$ 0.2	95.0 $\pm$ 1.5
Constant	83.6 $\pm$ 1.5	98.7 $\pm$ 0.1	93.6 $\pm$ 0.7

[*] We note that our results with MSE are better than those reported in Hoogeboom et al. [2022] and attribute this to exhaustive learning rate tuning.

sparse loss function on large molecules. This suggests it is necessary to ablate these coefficients on new tasks.

6.3 Spin ground state prediction

Experimental Setup We consider the task of predicting the ground states of the spin Hamiltonian and compare the effectiveness of different loss functions. We construct a dataset of 10,000 training and test spin-glass Hamiltonians, each with couplings uniformly sampled from $[-1, 1]$. We consider grids of size 16×16, which offer a challenging problem. The target ground-states are obtained by solving the associated integer linear program [Billionnet and Elloumi, 2007]. A convolutional neural network (CNN) is trained to take as input the coefficients J_{ij} and predict the ground state configuration $\hat{\mathbf{y}}_i$. More details on the architecture and training setup are provided in Appendix C.3. We compare training with the energy loss function of Equation (16) to the cross-entropy loss function and the margin loss function, another commonly used loss function for classification. The evaluation metric we consider is the energy of the predicted configuration. We also compare with direct minimization of the true energy as a baseline, despite it not being a classification objective.

Table 5: Results on ground-state prediction.

Loss	Cross-entropy	Margin loss	Local energy	True energy
Test energy	58.8 ± 0.8	49.87 ± 1.5	45.6 ± 1.6	14.6 ± 0.3

Results The results in Table 5 show that using the local energy leads to lower configuration energies than the cross-entropy loss function and the margin loss function. As expected, minimizing the true energy still leads to lower overall energy, despite not using the data. The local-field loss also requires fewer training epochs to converge than the cross-entropy loss. The results support the hypothesis that directly embedding physical insights through the local-field formulation effectively guides the learning process toward physically meaningful predictions.

7 Conclusion

We demonstrated a new approach to designing loss functions for machine learning tasks in physical systems based on the system's energy. When applied to both continuous and discrete settings, we found that replacing a simple MSE or cross-entropy loss with our energy loss functions leads to improved predictions across experiments. We further demonstrate the suitability of this loss for diffusion models and analyze its symmetry-invariance properties and scalability.

Limitations and future work Some limitations remain, which also point to directions for future work. First, when energy loss functions are used for diffusion models, the correct score is recovered at low noise levels; exact recovery at higher noise levels would require an explicit correction, which we leave for later study. Second, while we offer a more principled approach to designing loss functions, some choices are still ad hoc. Looking ahead, richer surrogate energies that capture torsional angles could be investigated. The approach could also potentially be fruitfully extended to a broader class of systems, including crystalline materials and proteins.

Acknowledgments

We are thankful to Mohsin Hasan, Simon Blackburn, Bruno Rousseau and Simon Verret for helpful discussions. This research was supported by the CIFAR AI Chairs program, Intel AI Labs and NSERC Discovery. S.-O. K.'s research is also supported by IVADO and FRQNT, and D.L. research is additionally supported by FRQNT. Mila and Compute Canada provided computational resources.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.

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

Leonard Asimow and Ben Roth. The rigidity of graphs. *Transactions of the American Mathematical Society*, 245:279–289, 1978.

Simon Axelrod and Rafael Gomez-Bombarelli. Geom: Energy-annotated molecular conformations for property prediction and molecular generation, 2022. URL `https://arxiv.org/abs/2006.05531`.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis M Kochmann. Physics-informed diffusion models. *arXiv preprint arXiv:2403.14404*, 2024.

Alain Billionnet and Sourour Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical programming*, 109:55–68, 2007.

Juan Carrasquilla and Roger G. Melko. Machine learning phases of matter. *Nature Physics*, 13(5): 431–434, 2017. doi: 10.1038/nphys4035. URL `https://doi.org/10.1038/nphys4035`.

Subrahmanyan Chandrasekhar. Stochastic problems in physics and astronomy. *Reviews of modern physics*, 15(1):1, 1943.

Samuel Cognolato, Davide Rigoni, Marco Ballarini, Luciano Serafini, Stefano Moro, Alessandro Sperduti, et al. D4: Distance diffusion for a truly equivariant molecular design. In *ESANN 2025-Proceedings, 33rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 265–270, 2025.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

Sean Dewar. graph-rigidity-checker library. `https://github.com/dewar28/graph-rigidity-checker/`, 2025.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in neural information processing systems*, 32, 2019.

Albert Einstein. On the motion of small particles suspended in liquids at rest required by the molecular-kinetic theory of heat. *Annalen der physik*, 17(549-560):208, 1905.

Hans Frauenfelder, Stephen G Sligar, and Peter G Wolynes. The energy landscapes and motions of proteins. *Science*, 254(5038):1598–1603, 1991.

Yaotian Fu and Philip W Anderson. Application of statistical mechanics to np-complete problems in combinatorial optimisation. *Journal of Physics A: Mathematical and General*, 19(9):1605, 1986.

Crispin W Gardiner. Handbook of stochastic methods for physics, chemistry and the natural sciences. *Springer series in synergetics*, 1985.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Majdi Hassan, Nikhil Shenoy, Jungyoon Lee, Hannes Stärk, Stephan Thaler, and Dominique Beaini. Et-flow: Equivariant flow-matching for molecular conformer generation. *Advances in Neural Information Processing Systems*, 37:128798–128824, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d, 2022. URL https://arxiv.org/abs/2203.17003.

Han Huang, Leilei Sun, Bowen Du, and Weifeng Lv. Learning joint 2d 3d diffusion models for complete molecule generation, 2023. URL https://arxiv.org/abs/2305.12347.

Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal structure prediction by joint equivariant diffusion. *Advances in Neural Information Processing Systems*, 36:17464–17497, 2023.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845*, 2024.

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

Sékou-Oumar Kaba and Siamak Ravanbakhsh. Symmetry breaking and equivariant neural networks. *arXiv preprint arXiv:2312.09016*, 2023.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *International Conference on Machine Learning*, pages 15546–15566. PMLR, 2023.

Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5):922–923, 1976.

Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36, 2024.

Leon Klein and Frank Noé. Transferable boltzmann generators, 2025. URL https://arxiv.org/abs/2406.14426.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching, 2023. URL https://arxiv.org/abs/2306.15030.

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

Hendrik Anthony Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *physica*, 7(4):284–304, 1940.

Michael Krivelevich, Alan Lew, and Peleg Michaeli. Rigid partitions: from high connectivity to random graphs. *arXiv preprint arXiv:2311.14451*, 2023.

Jerome M Kurtzberg. On approximation methods for the assignment problem. *Journal of the ACM (JACM)*, 9(4):419–439, 1962.

Gerard Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering mathematics*, 4(4):331–340, 1970.

Paul Langevin. On the theory of brownian motion. *CR Acad. Sci. Paris*, 146(530-533):530, 1908.

Hannah Lawrence, Vasco Portilheiro, Yan Zhang, and Sékou-Oumar Kaba. Improving equivariant networks with probabilistic symmetry breaking. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZE6lrLvATd.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

J E Lennard-Jones. Cohesion. *Proceedings of the Physical Society*, 43(5):461, 1931. doi: 10.1088/0959-5309/43/5/301. URL https://dx.doi.org/10.1088/0959-5309/43/5/301.

Alan Lew, Eran Nevo, Yuval Peled, and Orit E. Raz. Sharp threshold for rigidity of random graphs. *Bulletin of the London Mathematical Society*, 55(1):490–501, 2023. doi: https://doi.org/10.1112/blms.12740.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Marc Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, volume 9. World Scientific Publishing Company, 1987.

Philip M Morse. Diatomic molecules according to the wave mechanics. ii. vibrational levels. *Physical review*, 34(1):57, 1929.

Vitali Nesterov, Mario Wieser, and Volker Roth. 3dmolnet: a generative network for molecular structures. *arXiv preprint arXiv:2010.06477*, 2020.

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.

Frank Noé, Gianni De Fabritiis, and Cecilia Clementi. Machine learning for protein folding and dynamics. *Current opinion in structural biology*, 60:77–84, 2020.

Seong Ho Pahng and Michael P Brenner. Predicting ground state configuration of energy landscape ensemble using graph neural network. *arXiv preprint arXiv:2008.08227*, 2020.

Raj Kumar Pathria. *Statistical Mechanics: International Series of Monographs in Natural Philosophy*, volume 45. Elsevier, 2017.

Yuval Peled. Sharp threshold for rigidity of random graphs. Presented at IPAM at UCLA, 2024.

Xingang Peng, Jiaqi Guan, Qiang Liu, and Jianzhu Ma. Moldiff: Addressing the atom-bond inconsistency problem in 3d molecule diffusion generation. *arXiv preprint arXiv:2305.07508*, 2023.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, 2018. doi: 10.1021/acs.jcim.8b00234.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1(1):140022, 2014. ISSN 2052-4463. doi: 10.1038/sdata.2014.22. URL https://doi.org/10.1038/sdata.2014.22.

Kevin Ryan, Jeff Lengyel, and Michael Shatruk. Crystal structure prediction via deep learning. *Journal of the American Chemical Society*, 140(32):10158–10168, 2018.

Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

Kusha Sareen, Daniel Levy, Arnab Kumar Mondal, Sékou-Oumar Kaba, Tara Akhound-Sadegh, and Siamak Ravanbakhsh. Symmetry-aware generative modeling through learned canonicalization, 2025. URL https://arxiv.org/abs/2501.07773.

Victor Garcia Satorras, Emiel Hoogeboom, Fabian B. Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows, 2022. URL https://arxiv.org/abs/2105.09016.

Norbert Schuch and Frank Verstraete. Computational complexity of interacting electrons and fundamental limitations of density functional theory. *Nature physics*, 5(10):732–735, 2009.

Gregor NC Simm and José Miguel Hernández-Lobato. A generative model for molecular distance geometry. *arXiv preprint arXiv:1909.11459*, 2019.

Tess E Smidt, Mario Geiger, and Benjamin Kurt Miller. Finding symmetry breaking order parameters with euclidean neural networks. *Physical Review Research*, 3(1):L012002, 2021.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

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. URL https://openreview.net/forum?id=PxTIG12RRHS.

M.F. Thorpe and P.M. Duxbury. *Rigidity Theory and Applications*. Fundamental Materials Research. Springer US, 1999. ISBN 9780306461156. URL https://books.google.ca/books?id=3XgykKbZymkC.

Soledad Villar, David W Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

David J Wales, Jonathan PK Doye, Mark A Miller, Paul N Mortenson, and Tiffany R Walsh. Energy landscapes: from clusters to biomolecules. *Advances in Chemical Physics*, 115:1–111, 2000.

Shih-Hsin Wang, Yuhao Huang, Justin M. Baker, Yuan-En Sun, Qi Tang, and Bao Wang. A theoretically-principled sparse, connected, and rigid graph representation of molecules. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=OIvg3MqWX2.

YuQing Xie and Tess Smidt. Equivariant symmetry breaking sets. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=tHKH4DNSR5.

Minkai Xu, Shitong Luo, Yoshua Bengio, Jian Peng, and Jian Tang. Learning neural generative dynamics for molecular conformation generation. *arXiv preprint arXiv:2102.10240*, 2021.

Soojung Yang and Rafael Gómez-Bombarelli. Chemically transferable generative backmapping of coarse-grained proteins. *arXiv preprint arXiv:2303.01569*, 2023.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, pages 1–3, 2025.

Leo Zhang, Kianoosh Ashouritaklimi, Yee Whye Teh, and Rob Cornish. Symdiff: Equivariant diffusion via stochastic symmetrisation. *arXiv preprint arXiv:2410.06262*, 2024.

Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*, 2023.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: As stated in the introduction, we introduce a new perspective on loss functions for physical systems (Section 3), analyze their properties (Section 4.1, Section 4.2), and demonstrate the benefits of this approach via multiple experiments (Section 6).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss assumptions made in theoretical claims, and in the conclusion Section 7 we point out further limitations that point to future research directions.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: Proofs are included in Appendix A.1.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide experimental details in the paper, and further details are provided in Appendix C.

   Guidelines:

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

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

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

Answer: [No]

Justification: We intend to make the code public upon acceptance.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Basic details about training and testing are included in Section 6, with full details included in Appendix C.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We explain our use of error bars in Appendix C, based on multiple seeds.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Compute resource details are included in Appendix C

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The paper conforms with the Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: Our contributions are meant to be general enough to apply to a very wide class of problems within the field of AI for science, including chemistry, physics, and biology. We consider these applications to be too broad to discuss any particular social impacts, positive or negative.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: No new data or pretrained models are released in this paper.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite sources for the datasets that we train our models on, and the models that we compare against.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

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

13. **New assets**

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

    Answer: [NA]

    Justification: No assets are released.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No crowdsourcing or human subjects were used.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: No crowdsourcing or human subjects were used.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: LLMs were not used as an important, original, or non-standard component of the core methods of this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Additional Theory

A.1 Proofs

A.1.1 Proof of Proposition 4.2

Proof. For any $(g_1, g_2) \in E(d) \times (\mathrm{Aut}(k(\mathbf{y})) \cap \mathrm{Aut}(\Delta y))$ where $g_1 \in E(d)$ and $g_2 \in (\mathrm{Aut}(k(\mathbf{y})) \cap \mathrm{Aut}(\Delta y))$, we have

$$E((g_1, g_2) \cdot \hat{\mathbf{y}}, \mathbf{y}) = \sum_{i,j}^{n} \frac{1}{2} k_{ij}(\mathbf{y}) (\|\mathbf{y}_i - \mathbf{y}_j\| - \|(g_1, g_2) \cdot \hat{\mathbf{y}}_i - (g_1, g_2) \cdot \hat{\mathbf{y}}_j\|)^2 \tag{17}$$

By linearity of the actions, we have

$$E((g_1, g_2) \cdot \hat{\mathbf{y}}, \mathbf{y}) = \sum_{i,j}^{n} \frac{1}{2} k_{ij}(\mathbf{y}) (\|\mathbf{y}_i - \mathbf{y}_j\| - \|g_1 \cdot (g_2 \cdot \hat{\mathbf{y}}_i - g_2 \cdot \hat{\mathbf{y}}_j)\|)^2 \tag{18}$$

Since the Euclidean norm of a difference is $E(d)$-invariant we have

$$E((g_1, g_2) \cdot \hat{\mathbf{y}}, \mathbf{y}) = \sum_{i,j}^{n} \frac{1}{2} k_{ij}(\mathbf{y}) (\|\mathbf{y}_i - \mathbf{y}_j\| - \|(g_2 \cdot \hat{\mathbf{y}}_i - g_2 \cdot \hat{\mathbf{y}}_j)\|)^2 \tag{19}$$

We then have

$$E((g_1, g_2) \cdot \hat{\mathbf{y}}, \mathbf{y}) = \sum_{i,j}^{n} \frac{1}{2} k_{ij}(\mathbf{y}) \left(\|\mathbf{y}_i - \mathbf{y}_j\| - \|\left(\hat{\mathbf{y}}_{g_2^{-1} \cdot i} - \hat{\mathbf{y}}_{g_2^{-1} \cdot j}\right)\|\right)^2 \tag{20}$$

Using the fact that $g_2 \in (\mathrm{Aut}(k(\mathbf{y})) \cap \mathrm{Aut}(\Delta y))$,

$$E((g_1, g_2) \cdot \hat{\mathbf{y}}, \mathbf{y}) = \sum_{i,j}^{n} \frac{1}{2} k_{g_2^{-1} \cdot i, g_2^{-1} \cdot j}(\mathbf{y}) \left(\|\mathbf{y}_{g_2^{-1} \cdot i} - \mathbf{y}_{g_2^{-1} \cdot j}\| - \|\left(\hat{\mathbf{y}}_{g_2^{-1} \cdot i} - \hat{\mathbf{y}}_{g_2^{-1} \cdot j}\right)\|\right)^2 \tag{21}$$

$$E((g_1, g_2) \cdot \hat{\mathbf{y}}, \mathbf{y}) = \sum_{i,j}^{n} g_2 \cdot \frac{1}{2} \left[k_{ij}(\mathbf{y}) (\|\mathbf{y}_i - \mathbf{y}_j\| - \|(\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j)\|)^2\right] \tag{22}$$

Since the sum is permutation invariant, we have

$$E((g_1, g_2) \cdot \hat{\mathbf{y}}, \mathbf{y}) = \sum_{i,j}^{n} \frac{1}{2} \left[k_{ij}(\mathbf{y}) (\|\mathbf{y}_i - \mathbf{y}_j\| - \|(\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j)\|)^2\right] \tag{23}$$

$$E((g_1, g_2) \cdot \hat{\mathbf{y}}, \mathbf{y}) = E(\hat{\mathbf{y}}, \mathbf{y}) \tag{24}$$

For the second argument, we similarly have

$$E(\hat{\mathbf{y}}, (g_1, g_2) \cdot \mathbf{y}) = E(\hat{\mathbf{y}}, \mathbf{y}), \tag{25}$$

due to the $E(d)$-invariance of the Euclidean norm and the automorphism of $\mathbf{y}$.

This completes the proof.

□

A.1.2 Proof of Corollary 4.3

Proof. First,

$$E(\mathbf{y}, \mathbf{y}) = 0 = \min_{\hat{\mathbf{y}} \in \mathbb{R}^{n \times d}} E(\hat{\mathbf{y}}, \mathbf{y}) \tag{26}$$

Then, given

$$E(\mathbf{y}, \mathbf{y}) = \sum_{i,j}^{n} \frac{1}{2} k_{ij}(\mathbf{y}) (\|\mathbf{y}_i - \mathbf{y}_j\| - \|\hat{\mathbf{y}}_i - \hat{\mathbf{y}}_j\|)^2 \tag{27}$$

we see that for any $k_{ij}(\mathbf{y}) > 0$ the loss is minimized when each term of the sum is zero which implies that $\Delta y = \Delta \hat{y}$ when the loss is minimized.

This implies

$$\underset{\hat{\mathbf{y}} \in \mathbb{R}^{n \times d}}{\arg\min} E(\hat{\mathbf{y}}, \mathbf{y}) = \{g \cdot \mathbf{y} \mid g \in E(d)\}. \tag{28}$$

Since the action of the group $\mathrm{Aut}(k(\mathbf{y})) \cap \mathrm{Aut}(\Delta y)$ on $\mathbf{y}$ is by definition trivial, the desired result follows.

□

A.1.3 Proof of Proposition 4.4

Proof. We first establish some preliminaries. Since $SE(d)$ is not compact, we do not assume that the density $p(\mathbf{x}_t)$ is normalized. This is not an issue since the results depend on the score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$, which is independent of the normalization.

We also parametrize the score in terms of a noise prediction to align with practice. The diffusion objective for time t is proportional to (multiplication by constants do not change the minimizer)

$$\mathcal{J}_t(\theta) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})} \left[\left\| \frac{1}{\alpha_t} \mathbf{x}_t - \frac{\sigma_t}{\alpha_t} \hat{\boldsymbol{\epsilon}}_\theta - \mathbf{x} \right\|^2 \right] \tag{29}$$

where $\hat{\boldsymbol{\epsilon}}_\theta = f_\theta(\mathbf{x}_t, t)$, $\mathbf{x}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$ and $p(\boldsymbol{\epsilon}) = \mathcal{N}(0, \mathbf{I})$. The sample prediction is given by $\hat{\mathbf{x}}_\theta = \frac{(\mathbf{x}_t - \sigma_t \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{x}_t, t))}{\alpha_t}$.

The expectation can be rewritten as

$$\mathcal{J}_t(\theta) = \mathbb{E}_{\mathbf{x}_t \sim p(\mathbf{x}_t), \mathbf{x} \sim p(\mathbf{x}|\mathbf{x}_t)} \left[\left\| \frac{1}{\alpha_t} \mathbf{x}_t - \frac{\sigma_t}{\alpha_t} \hat{\boldsymbol{\epsilon}}_\theta - \mathbf{x} \right\|^2 \right] \tag{30}$$

$$\mathcal{J}_t(\theta) = \mathbb{E}_{\mathbf{x}_t \sim p(\mathbf{x}_t)} \left[\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{x}_t)} \left[\left\| \frac{1}{\alpha_t} \mathbf{x}_t - \frac{\sigma_t}{\alpha_t} \hat{\boldsymbol{\epsilon}}_\theta - \mathbf{x} \right\|^2 \right] \right] \tag{31}$$

$$\mathcal{J}_t(\theta) = \mathbb{E}_{\mathbf{x}_t \sim p(\mathbf{x}_t)} \left[\mathcal{J}_t(\mathbf{x}_t, \theta) \right] \tag{32}$$

The objective is minimized if for all $\mathbf{x}_t$, we have the following noise prediction

$$\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}} = \underset{\hat{\boldsymbol{\epsilon}}_\theta \in \mathbb{R}^{n \times d}}{\arg\min} \mathcal{J}_t(\mathbf{x}_t, \theta) = \underset{\hat{\boldsymbol{\epsilon}}_\theta \in \mathbb{R}^{n \times d}}{\arg\min} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{x}_t)} \left[\left\| \frac{1}{\alpha_t} \mathbf{x}_t - \frac{\sigma_t}{\alpha_t} \hat{\boldsymbol{\epsilon}}_\theta - \mathbf{x} \right\|^2 \right] \tag{33}$$

Since expectation and minimization commute for the MSE, we have

$$\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}} = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{x}_t)} \left[\underset{\hat{\boldsymbol{\epsilon}}_\theta \in \mathbb{R}^{n \times d}}{\arg\min} \left\| \frac{1}{\alpha_t} \mathbf{x}_t - \frac{\sigma_t}{\alpha_t} \hat{\boldsymbol{\epsilon}}_\theta - \mathbf{x} \right\|^2 \right] \tag{34}$$

$$\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}} = \frac{1}{\sigma_t} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{x}_t)} \left[\mathbf{x}_t - \alpha_t \mathbf{x} \right] \tag{35}$$

Using Tweedie's formula, we obtain the usual score matching relationship:

$$\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}} = -\sigma_t \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t). \tag{36}$$

To prove our result, we consider the energy loss objective,

$$\mathcal{J}_t(\mathbf{x}_t, \theta) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|\mathbf{x}_t)} \left[E\left(\frac{1}{\alpha_t} \mathbf{x}_t - \frac{\sigma_t}{\alpha_t} \hat{\boldsymbol{\epsilon}}_\theta, \mathbf{x} \right) \right] \tag{37}$$

where

$$E(\hat{\mathbf{x}}, \mathbf{x}) = \sum_{i,j}^{n} \left(\|\mathbf{x}_i - \mathbf{x}_j\| - \|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\| \right)^2 = \sum_{i,j}^{n} \left(d(\mathbf{x})_{ij} - d(\hat{\mathbf{x}})_{ij} \right)^2 \tag{38}$$

and $d : \mathbb{R}^{n\times d} \to \mathbb{R}^{n^2}$ computes the distance matrix.

We perform a Taylor approximation of $d(\mathbf{x})$ and $d(\hat{\mathbf{x}})$ around $\mathbf{x}_t$

$$d(\mathbf{x}) \approx d(\mathbf{x}_t) + J(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t) \tag{39}$$

where $J(\hat{\mathbf{y}})$ is the Jacobian of d.

This approximation is valid when $|\mathbf{x} - \mathbf{x}_t| \ll 1$ and $|\mathbf{x} - \hat{\mathbf{x}}_t| \ll 1$ which corresponds to $|\sigma_t| \ll \frac{1}{\boldsymbol{\epsilon}}$ and $|\sigma_t| \ll \frac{1}{\hat{\boldsymbol{\epsilon}}}$ respectively. Assuming that $\boldsymbol{\epsilon}$ follows a standard normal distribution, and that $\hat{\boldsymbol{\epsilon}}$ is bounded, for usual diffusion schedules (with monotonically increasing noise) the approximation will be valid for small values of t.

This yields

$$E(\hat{\mathbf{x}}, \mathbf{x}) = \|J(\mathbf{x}_t)(\mathbf{x} - \hat{\mathbf{x}})\|^2 \tag{40}$$

Replacing in Equation (37), we have

$$\mathcal{J}_t(\mathbf{x}_t, \theta) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{x}_t)}\left[\left\|J(\mathbf{x}_t)\left(\frac{1}{\alpha_t}\mathbf{x}_t - \frac{\sigma_t}{\alpha_t}\hat{\boldsymbol{\epsilon}}_\theta - \mathbf{x}\right)\right\|^2\right] \tag{41}$$

Since $\mathcal{J}_t$ is a convex function of $\hat{\boldsymbol{\epsilon}}$, its minimization with respect to $\hat{\boldsymbol{\epsilon}}_\theta$ can be performed by solving for vanishing gradient, which leads to the minimizer $\hat{\boldsymbol{\epsilon}}^*$ satisfying

$$J(\mathbf{x}_t)^T J(\mathbf{x}_t)\left(\hat{\boldsymbol{\epsilon}}^* - \frac{1}{\sigma_t}\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{x}_t)}[\mathbf{x}_t - \alpha_t\mathbf{x}]\right) = 0 \tag{42}$$

$$J(\mathbf{x}_t)^T J(\mathbf{x}_t)(\hat{\boldsymbol{\epsilon}}^* - \sigma_t\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)) = 0 \tag{43}$$

Because of the $SE(d)$ invariance of d, $J(\mathbf{x}_t)$ does not have full-rank. Denoting the kernel space of $J(\mathbf{x}_t)$ as $\ker(J(\mathbf{x}_t))$, we obtain

$$\hat{\boldsymbol{\epsilon}}^* = -\mathbf{P}_{J(\mathbf{x}_t)}\frac{1}{\sigma_t}\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t) + \mathbf{v}, \quad \mathbf{v} \in \ker(J(\mathbf{x}_t)) \tag{44}$$

where $\mathbf{P}_{J(\mathbf{x}_t)}$ is the orthogonal projector onto $\mathrm{range}(J(\mathbf{x}_t))$.

Using the assumption that the measure $p(\mathbf{x}_t)$ is $SE(d)$-invariant, the score lies in $\mathrm{range}(J(\mathbf{x}_t))$ and is orthogonal to any $\mathbf{v} \in \ker(J(\mathbf{x}_t))$. The norm of $\hat{\boldsymbol{\epsilon}}^*$ is therefore minimized when it is equal to the score. This completes the proof.

□

A.1.4 Proof of Proposition 4.5

Proof. We assume the same approximation regime as in Proposition 4.4.

The Monte-Carlo estimators associated with the two losses (for the distance loss, we assume the minimum norm minimizer) are given by

$$\hat{\boldsymbol{\epsilon}}^*_{\text{dist}} = -\mathbf{P}_{J(\mathbf{x}_t)}\frac{1}{\sigma_t}\frac{1}{N}\sum_i^N \mathbf{x}_t - \alpha_t\mathbf{x}^{(i)} \qquad \hat{\boldsymbol{\epsilon}}^*_{\text{MSE}} = -\frac{1}{\sigma_t}\frac{1}{N}\sum_i^N \mathbf{x}_t - \alpha_t\mathbf{x}^{(i)} \tag{45}$$

where the N samples $\mathbf{x}^{(i)}$ are drawn i.i.d. from $p(\mathbf{x} \mid \mathbf{x}_t)$.

Bias $[\hat{\boldsymbol{\epsilon}}^*_{\text{dist}}] = 0$:

This simply follows from Equation (36) and the fact that the true score for an $SE(d)$-invariant density lies in $\mathrm{range}(J(\mathbf{x}_t))$, so that $-\frac{1}{\sigma_t}\mathbf{P}_{J(\mathbf{x}_t)}\nabla_{\mathbf{x}_t}p(\mathbf{x}_t) = -\frac{1}{\sigma_t}\nabla_{\mathbf{x}_t}p(\mathbf{x}_t)$.

Var $[\hat{\boldsymbol{\epsilon}}^*_{\text{dist}}] \lesssim$ Var $[\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}}]$:

We have

$$\text{Cov}[\hat{\boldsymbol{\epsilon}}^*_{\text{dist}}] = \mathbf{P}_{J(\mathbf{x}_t)}\text{Cov}[\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}}]\,\mathbf{P}^T_{J(\mathbf{x}_t)} \tag{46}$$

We can then obtain,

$$\mathrm{Tr}\left(\mathrm{Cov}\left[\hat{\boldsymbol{\epsilon}}^*_{\text{dist}}\right]\right) = \mathrm{Tr}\left(\mathbf{P}_{J(\mathbf{x}_t)}\mathrm{Cov}\left[\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}}\right]\mathbf{P}^T_{J(\mathbf{x}_t)}\right) \tag{47}$$

$$\mathrm{Tr}\left(\mathrm{Cov}\left[\hat{\boldsymbol{\epsilon}}^*_{\text{dist}}\right]\right) = \mathrm{Tr}\left(\mathbf{P}_{J(\mathbf{x}_t)}\mathrm{Cov}\left[\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}}\right]\right) \tag{48}$$

which compared to

$$\mathrm{Tr}\left(\mathrm{Cov}\left[\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}}\right]\right) = \mathrm{Tr}\left(\mathbf{P}_{J(\mathbf{x}_t)}\mathrm{Cov}\left[\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}}\right]\right) + \mathrm{Tr}\left(\left(\mathbf{I} - \mathbf{P}_{J(\mathbf{x}_t)}\right)\mathrm{Cov}\left[\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}}\right]\right) \tag{49}$$

since $\left(\mathbf{I} - \mathbf{P}_{J(\mathbf{x}_t)}\right)$ and $\mathbf{P}_{J(\mathbf{x}_t)}$ are both positive semi-definite, we conclude

$$\mathrm{Tr}\left(\mathrm{Cov}\left[\hat{\boldsymbol{\epsilon}}^*_{\text{dist}}\right]\right) \leq \mathrm{Tr}\left(\mathrm{Cov}\left[\hat{\boldsymbol{\epsilon}}^*_{\text{MSE}}\right]\right) \tag{50}$$

which completes the proof. □

At a high-level, the projector $\mathbf{P}_{J(\mathbf{x}_t)}$ removes the components associated with rigid motions from the estimated score. The minimum norm assumption amounts to the network denoinsing to the closet sample to $\mathbf{x}_t$ in the orbit of $\mathbf{x}$.

A.2 The Boltzmann Distribution is the Steady-state Distribution

We consider the potential energy $E(\hat{\mathbf{y}}, \mathbf{y})$ around a configuration $\mathbf{y}$. We assume $\mathbf{y}$ in an approximate equilibrium configuration of the system, and therefore a local minimum of the potential. We model the evolution of the system using Newton's equation (in Hamiltonian form), with an additional term modeling the noise. This gives the SDEs

$$\mathrm{d}\hat{\mathbf{y}} = \frac{\partial K}{\partial \mathbf{P}}\mathrm{d}t, \qquad \mathrm{d}\mathbf{P} = -\frac{\partial E}{\partial \hat{\mathbf{y}}}\mathrm{d}t - \mathbf{P}\mathrm{d}t + \sqrt{2T}\mathrm{d}\mathbf{W}. \tag{51}$$

where $\mathbf{W}$ is generically taken as standard Brownian noise and K is the kinetic energy of the system. The term proportional to $\mathbf{P}$ in the second equation represents the effect of friction and has the effect of bringing the system back towards the equilibrium configuration. This is the Langevin equation, introduced by Einstein [1905], Langevin [1908] to study Brownian motion. It describes the evolution of a system in a heat bath at temperature T; this is the uncertainty model we consider.

We suppose momentum variables are not of interest. We therefore take the kinetic energy (or masses) as negligible, obtaining the *overdamped* limit Kramers [1940]

$$\mathrm{d}\mathbf{y} = -\frac{\partial E}{\partial \hat{\mathbf{y}}}\mathrm{d}t + \sqrt{2T}\mathrm{d}\mathbf{W} \tag{52}$$

The probability density of $\mathbf{y}$ evolves according to the Fokker-Plank equation Chandrasekhar [1943]

$$\frac{\partial p(\hat{\mathbf{y}}, t \mid \mathbf{y})}{\partial t} = T\nabla^2_{\mathbf{y}} p(\hat{\mathbf{y}}, t \mid \mathbf{y}) + \nabla_{\mathbf{y}} \cdot [p(\hat{\mathbf{y}}, t \mid \mathbf{y}))\nabla_{\mathbf{y}} E(\hat{\mathbf{y}}, \mathbf{y})] \tag{53}$$

For anything but the simplest potentials, this equation admits no known closed-form solution. However, under some growth conditions on the potential $V(\mathbf{y})$ (see e.g. Gardiner [1985]), the Fokker-Plank equation admits the Boltzmann distribution as a unique steady-state solution when $t \to \infty$:

$$p(\hat{\mathbf{y}} \mid \mathbf{y}) = \frac{1}{Z(\mathbf{y}, T)} \exp\left(-E(\hat{\mathbf{y}}, \mathbf{y})/T\right) \tag{54}$$

where $Z(\mathbf{y}, T)$ is the partition function.

A.3 Second-order Taylor Approximations of Potential Energies

Below we justify the two practical choices for the coefficients $k_{ij}(\mathbf{y})$ (constant and inverse–squared decay) by performing second–order Taylor expansions of two standard pair potentials around their equilibrium bond length r. Throughout we write the equilibrium distance between atoms $r = \|\mathbf{y}_1 - \mathbf{y}_2\|$ and the difference between the equilibrium distance and the prediction $\delta r = \hat{r} - r$.

Morse potential The Morse potential (typically used to describe bonded pairs) is given

$$E_{\text{Morse}}(\hat{r}, r) = D\Big[1 - \exp\big(-a(\hat{r} - r)\big)\Big]^2. \tag{A.2}$$

Expanding the exponential for small δr gives $\exp(-ax) = 1 - ax + \frac{1}{2}a^2\delta r^2 + \mathcal{O}(\delta r^3)$. The quadratic approximation is

$$E_{\text{Morse}}(\hat{r}, r) = Da^2\delta r^2 \ = \tfrac{1}{2}, k^{(\text{M})}\delta r^2 \tag{55}$$

with

$$k^{(\text{M})} = 2Da^2 \tag{A.4}$$

Because D and a are fixed per bond type, $k^{(\text{M})}$ is constant in r.

Lennard–Jones potential The 12–6 Lennard–Jones potential (commonly used for non-bonding interactions) can be written in terms of its equilibrium separation $r = 2^{1/6}\sigma$:

$$E_{\text{LJ}}(\hat{r}, r) = \varepsilon\Big[\big(\tfrac{r}{\hat{r}}\big)^{12} - 2\big(\tfrac{r}{\hat{r}}\big)^{6}\Big]. \tag{A.5}$$

Expanding for small δr and keeping terms up to second order,

$$E_{\text{LJ}}(\hat{r}, r) = -\varepsilon + \frac{1}{2}\, k^{(\text{LJ})}\delta r^2 + \mathcal{O}(\delta r^3), \tag{56}$$

$$k^{(\text{LJ})} = \left.\frac{\partial^2 E_{\text{LJ}}}{\partial \hat{r}^2}\right|_{\hat{r}=r} = \frac{72\,\varepsilon}{r^2}. \tag{A.6}$$

Because ε is fixed for a given atom-pair type, the resulting spring constant

$$k^{(\text{LJ})} \propto r^{-2}$$

decays with the *inverse square* of the equilibrium distance.

A.4 Invariance and Invariant Loss Functions

Physical systems frequently have inherent symmetry, and machine learning models built for such systems benefit from handling these symmetries. Two common symmetries are the $E(3)$ symmetry of Cartesian coordinates, and the $S(n)$ symmetry of exchangeable objects (such as atoms in an atomistic systems). In the context of generative models, we note three approaches to designing models that respect the symmetry of the underlying distribution:

Invariant Distribution If we decide that the probability distribution we want to generate is *invariant* to symmetry transformations, a common strategy is to first sample from a distribution that is invariant to the group of interest, and then applying a function that is equivariant to the group [Köhler et al., 2020]. For example, Hoogeboom et al. [2022] samples from a Gaussian distribution that is invariant to rotations in $SO(3)$, and uses an $E(3)$-equivariant neural network to parametrize a denoising diffusion model. The probability distribution of the resulting structures generated by their model is therefore invariant to rotations.

Alignment In many situations, we want predictions from a neural network to match ground truth data, *up to some symmetry transformation*. This is often the case in generative models for proteins and molecules, where there is symmetry to SE(3) and S(n). Recent works such as Hassan et al. [2024], Klein et al. [2023] handle this in a flow-matching context by performing an alignment procedure between samples from a prior and data samples, finding an element of S(n) and SO(3) that minimizes the distance between them. While costly, this optimal alignment can be found by using the Hungarian algorithm [Kurtzberg, 1962] and the Kabsch algorithm [Kabsch, 1976]. Similarly, Sareen et al. [2025] use an equivariant network to learn alignments that brings data samples into canonical representatives [Kaba et al., 2023], which are then fed into a generative model. Zhang et al. [2024] use a similar symmetrization method to achieve equivariance. This technique obviates the need to use an expensive equivariant network for the generative model.

Invariance-based loss Another method to handle symmetry is to use a loss based only on invariant features. Works such as Xu et al. [2021], Simm and Hernández-Lobato [2019] and Nesterov et al. [2020] learn to model interatomic distance matrices rather than atomic coordinates, and convert from distance matrices into coordinates as a post-processing step. Because distances are invariant under E(3) transformations, the resulting method is invariant. Our proposed loss function Equation (5) falls into this last category.

A.5 More on Rigidity Theory

Background on theory We begin by describing the setting of rigidity theory and the necessary properties for our energy loss to scale linearly in the number of vertices.

Rigidity theory defines a *framework* as a graph $G = (V, E)$ and a map $\phi : V \to \mathbb{R}^d$ which can be interpreted as the physical coordinates of a given vertex. We call a framework *globally rigid* if every $\phi' : V \to \mathbb{R}^d$ that yields the same distances between adjacent vertices is obtained from ϕ by an isometry. A framework is *rigid* if there are no *non-trivial* continuous motions of vertices starting from ϕ that preserves the distance between adjacent vertices. Trivial motions, in this case, correspond to group elements of $E(d)$. The central problem of rigidity theory is to determine under which conditions different families of frameworks are rigid or globally rigid [Peled, 2024] [Thorpe and Duxbury, 1999].

In our application, it suffices to neglect modeling some interactions between atoms (terms in the sum in Equation (7)), as long as the minimum loss configuration is unique and thus corresponds to the data. In other words, we require the edges describing this sum are *globally rigid*.

Due to the simplicity of their construction, we mention a few recent results on the rigidity of random graphs. The first result is that a random k-regular graph is rigid with high probability (w.h.p.) in D dimensions for $k \geq D^2$. It has been conjectured that $k \geq 2D$ should be enough for rigidity, but the existing proof is limited to $D = 2$ [Krivelevich et al., 2023] [Peled, 2024]. Alternatively, if using Erdos-Renyi random graphs, in D dimensions, one could keep adding edges at random until the minimum node degree becomes D, at which point the graph becomes rigid w.h.p. [Lew et al., 2023]. Note that a promising construction for rigid graphs was also recently proposed in the context of machine learning [Wang et al., 2025].

$2D$-regular graphs are globally rigid w.h.p. To construct a sparse version of the energy loss, we require that the edges over which the pairwise distances differences are summed in Equation (7) make up a globally rigid graph. For simplicity, we use random $2D$-regular graphs in the sparse version of the energy loss since the number of edges scales linearly in N.

We empirically confirm the conjecture for D and N in ranges relevant for our setting. We construct 1000 random 6-regular graphs and check the fraction of the graphs that are globally rigid using rigidity checking code from Dewar [2025]. This code implements a rank check of the rigidity matrix and a random stress test. The key result is depicted in Figure 5.

These graphs are used in the sparse energy loss for shape prediction. For molecules, we use a symmetrized version of these graphs according to molecule symmetries. In both settings, 100 random graphs are pre-generated for each number of vertices and a random graph from this pool is chosen when computing the loss.

Symmetrization procedure for random graphs We notice that Proposition 4.2 does not necessarily hold for Equation (7) when the edges are not complete. This means atoms symmetric under $\mathrm{Aut}\,(\Delta y)$ will not necessarily obtain the same gradients. To remedy this, we introduce a symmetrization procedure for a given k-regular graph and molecule.

We can select k random edges for a representative node in each orbit and symmetrize the adjacency matrix over the orbits via

$$A_{sym} = \sum_{g \in \mathrm{Aut}(\Delta y)} g \cdot A \cdot g^{-1}. \tag{57}$$

Notice that the total number of edges we get by this procedure is

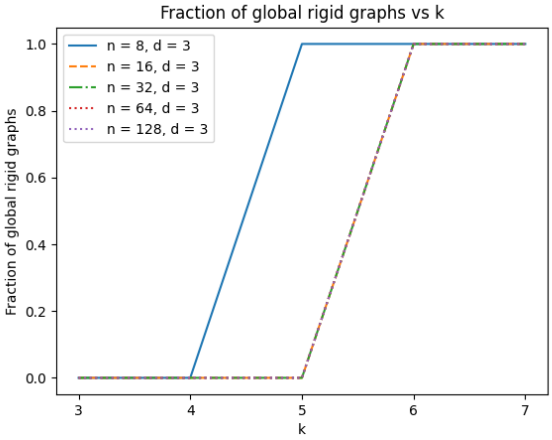


Figure 5: Global rigidity testing of random k-regular graphs. Here, n denotes the number of vertices and d the dimension.

$$|E_{sym}| = O\left(\sum_{uv \in E} |uG||vG|\right), \tag{58}$$

which can be quadratic in $|V|$ if there are large orbits (ie. size $O(|V|)$) directly connected by edges but is linear otherwise. It may be possible to ensure this is linear w.h.p. by choosing the k edges in a way that depends on G but we leave this for future work. We empirically verify these edges are sparse in our setting.

A.6 More on Spin Energy

For an Ising configuration $\mathbf{y} \in \{-1, +1\}^{\Lambda}$ the energy change caused by flipping spin y_i is $\Delta E_i = 2\mathbf{y}_i h_i^{\mathrm{LF}}(\mathbf{y})$ with $h_i^{\mathrm{LF}}(\mathbf{y}) = \sum_j J_{ij}\mathbf{y}_j$. Setting $h_0 = 0$ therefore makes

$$E_{\mathrm{LF}}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i \in \Lambda} h_i^{\mathrm{LF}}(\mathbf{y})\, \hat{\mathbf{y}}_i,$$

proportional to this exact spin-flip energy around the ground state. The local energy we propose is therefore a sensible linear approximation to the true energy.

To obtain a convex loss, we make sure the local field weighting is always positive, For this, we add a global offset $h_0 > 0$ in $h_i^{\mathrm{LF}}(\mathbf{y}) = \sum_j (J_{ij} + h_0)\mathbf{y}_j$. Since each site has at most four neighbours with $|J_{ij}| \leq 1$, we have $-4 \leq h_i^{\mathrm{true}} \leq 4$; choosing a single $h_0 > 4$ ensures weight $(J_{ij} + h_0) > 0$ and penalises energetically costly errors more heavily.

A.7 Extension to Flow Matching

The energy loss can be extended to Gaussian flow matching [Lipman et al., 2023]. We show hereafter the correspondence for the conditional vector field. The noisy sample is given by the interpolation

$$\mathbf{x}_t = (1 - t)\mathbf{x} + t\boldsymbol{\epsilon}.$$

The flow matching objective aims at regressing the vector field:

$$\mathbf{u} = \frac{\mathbf{x}_t - \mathbf{x}}{t}.$$

Given a vector field prediction, the corresponding sample prediction is

$$\mathbf{x} = \mathbf{x}_t - t\mathbf{u}_\theta(\mathbf{x}_t).$$

The correspondence between MSE on the vector field prediction and on sample prediction is therefore:

$$\|\mathbf{u} - \mathbf{u}_\theta\|^2 = \frac{1}{t^2}\|\mathbf{x} - \mathbf{x}_\theta\|^2.$$

Therefore, the associated energy objective is obtained by replacement of the regression MSE:

$$\frac{1}{t^2}E(\mathbf{x} - \mathbf{x}_\theta).$$

Our theoretical results relating to score estimation properties also transfer to flow matching. This is because Gaussian flow matching also implicitly provides a method for score estimation similar to diffusion models. Given the optimal vector field, $\mathbf{u}^*(\mathbf{x}_t)$, the score is given by:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) = -\frac{(1-t)\mathbf{u}^*(\mathbf{x}_t) + \mathbf{x}_t}{t}.$$

B Related works

Several different lines of research have also incorporated a concept of energy into a machine learning framework. In this section, we distinguish our framework from distinct but related areas of research.

Energy-based models Traditional energy-based models approach learning as shaping an energy landscape, where observed configurations correspond to low-energy states [LeCun et al., 2006]. Deep counterparts and their connection to discriminative training have also been extensively explored in many recent works (e.g., Du and Mordatch [2019], Grathwohl et al. [2019]). A key distinction of the existing literature on energy-based models and our energy loss approach is that, because they minimize the forward KL (max-likelihood or alternatives such as contrastive and large-margin losses), they need to deal with the minimization of the partition function or its surrogates – i.e., the energy of arbitrary points in the domain must be high. In contrast, our treatment remains close to supervised learning losses and avoids the partition function altogether.

Physics-informed neural networks (PINNs) [Raissi et al., 2019] has proposed PINNs as a way to learn PDEs by penalizing residuals directly in the loss, enforcing solutions consistent with physical constraints. They have recently been used in the context of diffusion models [Bastek et al., 2024]. In contrast to our approach, these models do not rely on training data and models are instead learned to satisfy known differential equations on randomly generated points from the domain. Another family of physics-informed losses appears in Hamiltonian Neural Networks (HNN) [Greydanus et al., 2019] and Lagrangian Neural Networks (LNN) [Cranmer et al., 2020].

Energy Sampling and Boltzmann Generators A separate line of research incorporating energies and generative modeling has been in Boltzmann Generators [Noé et al., 2019, Köhler et al., 2020, Klein and Noé, 2025]. These models are designed to sample physical configurations according to a Boltzmann distribution stemming from a known energy function. While our framework is also based on an assumption of data belonging to a Boltzmann distribution, ours is instead simply an approximation of the local landscape around each data point and does not assume the existence of a callable energy function.

C Experimental Details and Additional Results

C.1 Regular shape generation

Experimental details and hyperparameters We use a 2 hidden-layer MLP with hidden dimension 64 for this task. We conducted a sweep over hidden dimension and find behaviour is relatively consistent. Models are trained in parallel on an Nvidia Quadro RTX 8000 using the Adam optimizer. The dataset size is 100K randomly generated samples and we train all models for 50 epochs. A sweep over dataset size showed fairly consistent results. We conduct thorough sweeps for learning rate for each loss, shape degree and augmentation angle. For each setting, the model giving the highest quality is chosen.

Table 6: Best hyperparameters for GDM.

Loss	Coefficient	Learning Rate	Positional Loss Weight
Energy	Constant	9e-4	0.05
	Inv. Dist.	7e-4	0.05
	Inv. Sq. Dist.	6e-4	0.05
	Exp. Dist.	4e-4	0.05
MSE	-	1e-3	1.5
MAE	-	1e-3	0.8
Kabsch Align	-	8e-4	0.8

Table 7: Best hyperparameters for EDM.

Loss	Coefficient	Learning Rate	Positional Loss Weight
Energy	Constant	1e-4	0.05
	Inv. Dist.	1e-4	0.05
	Inv. Sq. Dist.	1e-4	0.1
	Exp. Dist.	1e-4	0.05
MSE	-	3e-4	1.0
MAE	-	3e-4	1.0
Kabsch Align	-	3e-4	0.8

Evaluation To evaluate shape quality, we introduce a metric that captures how regular the angular differences and radial distances are across the shape. In a well-formed, regular shape, we expect both the variation in angular differences ($\sigma_{\Delta_{angle}}$) and the variation in radial distances (σ_{radius}) to be small. In particular, we choose Quality $:= -\ln\left(\frac{\sigma_{\Delta_{angle}}}{2\pi} + \frac{\sigma_{radius}}{r}\right)$. This is, of course, a design choice used to map a shape to a single number. We record both $\sigma_{\Delta_{angle}}$ and σ_{radius} to ensure both terms are well-represented in the quality and find that visually this metric is a good reflection of the visual regularity of a generated shape. For reference, above a quality of 5-6, shapes look nearly visually perfect, as in Figure 3a. For quality below this, they become slightly irregular and below 2 they look very disordered as in Figure 3b.

C.2 Molecule generation

C.2.1 QM9 Dataset

Experimental details and hyperparameters On QM9, we match the setup in Hoogeboom et al. [2022] as closely as possible. We train GDMs and EDMs with with 9 layers and 256 node features on 100k samples from the dataset. The diffusion process has 1000 diffusion steps with polynomial noise schedule and precision 1×10^{-5}. An L2 denoising loss is used with mini-batch size 512 on GDM and 400 on EDM. We use the Adam optimizer. An EMA decay of 0.9999 is used. Runs were conducted on single 48G GPUs mainly on the Nvidia Quadro RTX 8000, A6000 and L40S. A full run of 3000 epochs takes 2-4 days on a single GPU.

We conduct extensive sweeps for learning rate and positional loss weight for all losses. We tune the positional loss weight to ensure there is balance between loss on positions and atom-type for all losses. Learning rates were searched for in broadly in the range [1e-5, 1e-2] before narrowing the range to [2e-3, 4e-4]. For the positional loss weight, we choose values in $[0.05, 0.1, 0.5, 0.8, 1.0, 1.5]$. We find final performance is not very sensitive to the positional loss weight. Tuned hyperparameters are summarized in Table 6 and Table 7.

Additional results Here, we include results for all settings for GDM, GDM-aug and EDM. The results follow in Table 8. Results are averaged across seeds.

Table 8: Complete results on QM9.

Loss	Mol. stab. (%)	Atom stab. (%)	Valid. (%)	Unique (%)
GDM				
MSE	81.7 ± 3.3	98.3 ± 0.3	93.3 ± 1.7	99.98 ± 0.04
MAE	76.3 ± 2.0	97.7 ± 0.3	91.1 ± 1.2	99.96 ± 0.05
Kabsch Align	81.7 ± 2.2	98.4 ± 0.2	93.1 ± 1.2	99.93 ± 0.13
Energy (Sparse)	**86.1** ± 2.3	**99.0** ± 0.1	96.2 ± 1.4	100.0 ± 0.0
Energy	**86.2** ± 2.1	98.9 ± 0.2	**96.6** ± 1.3	100.0 ± 0.0
GDM-aug				
MSE	83.7 ± 2.3	98.3 ± 0.004	93.6 ± 1.7	100.0 ± 0.0
MAE	76.4 ± 0.9	98.1 ± 0.3	92.6 ± 1.2	99.99 ± 0.02
Kabsch Align	82.3 ± 0.5	97.8 ± 0.004	90.8 ± 2.0	100.0 ± 0.0
Energy (Sparse)	89.1 ± 0.9	99.0 ± 0.1	97.4 ± 2.5	100.0 ± 0.0
Energy	**89.8** ± 2.8	**99.3** ± 0.3	**97.7** ± 1.4	99.99 ± 0.002
EDM				
MSE	82.4 ± 3.4	98.8 ± 1.7	93.0 ± 2.5	99.89 ± 0.32
MAE	74.8 ± 1.7	97.8 ± 0.3	88.6 ± 0.7	99.96 ± 0.07
Kabsch Align	80.6 ± 3.0	98.3 ± 3.0	92.5 ± 3.0	99.91 ± 0.07
Energy	**86.6** ± 1.6	**99.0** ± 0.20	**96.8** ± 1.1	99.96 ± 0.06

Table 9: Complete results for GDM and GDM-aug on GEOM-Drugs.

Loss	Mol. stab. (%)	Atom stab. (%)	Valid. (%)	Unique (%)
GDM				
MSE	0.3	84.7	**93.8**	100
Energy	**21.1**	**95.8**	89.6	100
GDM-aug				
MSE	0.8	85.6	**94.8**	100
Energy	**24.6**	**96.0**	89.7	100

C.2.2 GEOM-Drugs Dataset

Experimental details and hyperparameters On GEOM-Drugs, we use a similar setting to QM9 but now train models with 4 layers and 256 node features, following Hoogeboom et al. [2022]. We train the model for 13 epochs. Training is distributed across 4 80G Nvidia A100l GPUs and a single run takes roughly 2.5 days. We use a batch size of 128 with the Adam optimizer.

We start with optimal learning rate and positional loss weight from QM9 and do a sweep over learning rates [5e-4, 1e-3, 2e-3] for MSE and [1e-4, 4e-4, 1e-3] for energy loss. The hyperparameters in Table 6 gave the best results. We use exponential coefficients for the energy loss.

Additional results We additionally report the performance of MSE and energy losses with GDM-aug in Table 9.

C.3 Spin ground state prediction

Experimental details and hyperparameters The CNN we use for the spin prediction task is a 6 layer ResNet type architecture with 256 hidden layer size. All networks are trained with a learning rate of 1×10^{-3} until convergence. We use the Adam optimizer with batch size 256. Temperature is set to $T = 0.1$ for the local energy loss. Training takes around 5 hours on Nvidia V100 GPUs.

C.4 More on sparse energy loss

C.4.1 Timing

Our objective in including the sparse energy loss is to demonstrate our method can efficiently generalize to systems with many particles where loss calculation may contribute significantly to running time (e.g. very large point clouds). This is not the case for molecules, where the neural network (GNN or Transformer) is typically fully connected and thus scales as N^2. As Table 10 shows, the most expensive loss calculation is less than 1% of the total backward and forward time.

Table 10: Wall-times for loss computation on QM9 on an NVIDIA L40S.

Component	Loss Type	Time (ms)
Loss computation	MSE	0.18 ± 0.01
	Energy	0.51 ± 0.02
	Sparse Energy	0.57 ± 0.02
	Kabsch Align	1.14 ± 0.03
Forward pass	–	74 ± 16
Backward pass	–	94 ± 3
Optimizer step	–	1.43 ± 0.01

To better understand the scale at which this becomes a relevant consideration and the utility of the sparse energy loss, see the following wall clock times from the shape generation setting in Table 11.

Table 11: Runtime for different loss functions as the number of nodes increases.

# Nodes	Energy (ms)	Sparse Energy (ms)	MSE (ms)	Kabsch Align (ms)
30	0.240 ± 0.0021	0.255 ± 0.0012	0.058 ± 0.0004	0.807 ± 0.0027
300	0.245 ± 0.0011	0.257 ± 0.0024	0.059 ± 0.0004	0.804 ± 0.0048
3000	20.120 ± 0.0055	0.275 ± 0.0018	0.0779 ± 0.0008	0.944 ± 0.0028
30000	–	0.293 ± 0.0029	0.0740 ± 0.0011	3.242 ± 0.1452
300000	–	2.652 ± 0.0050	0.131 ± 0.0004	24.381 ± 0.0272

At 30000+ nodes, the energy loss requires too much memory to compute. Importantly, the sparse energy is cheaper than the Kabsch Align by a factor 5-10x. Note that all losses have some constant cost that does not scale with N contributing to the wall-clock time. This explains why for QM9 (avg. 28 atoms) energy loss is marginally faster than sparse energy and why in the scaling table wall-clock times start to increase with N only after a certain point.

Interestingly, using an equivariant network with EDM takes 129.71 ± 0.044 ms for the forward pass and 180.07 ± 0.070 ms for the backward pass. Using the energy loss imparts a 0.3% increase on one backward pass through the model, while using an equivariant architecture imparts a 94% increase, while providing inferior benefits. The energy loss with a non-equivariant architecture results in more improvement than using an equivariant architecture, at negligible computational cost, which we think is a significant finding.

C.5 Sparse energy loss on larger systems

Table 12: Sparse energy loss with GDM on GEOM-Drugs.

Loss	Mol. stab. (%)	Atom stab. (%)	Valid. (%)	Unique (%)
MSE	0.3	84.7	93.8	100
Energy	**21.1**	**95.8**	89.6	100
Sparse Energy (Inv. Dist)	7.4	91.9	92.6	100

We include the sparse energy results on Geom-Drugs in Table 12. We found using a more gradual distance decay in the coefficient worked better when the edges are sparse and random. These results highlight a potential compute-performance tradeoff for this version of the loss on larger graphs. This result can likely be improved by being more intentional in the selection of sparse edges. We leave this to future work.