LEAPS: A DISCRETE NEURAL SAMPLER VIA LOCALLY EQUIVARIANT NETWORKS

Peter Holderrieth[1,*], **Michael S. Albergo**[2,3*], **Tommi Jaakkola**[1]
[1]MIT CSAIL, [2]Society of Fellows, Harvard University,
[3]Institute for Artificial Intelligence and Fundamental Interactions, *Equal contribution

ABSTRACT

We propose *LEAPS*, an algorithm to sample from discrete distributions known up to normalization by learning a rate matrix of a continuous-time Markov chain (CTMC). The method can be seen as a continuous-time formulation of annealed importance sampling and sequential Monte Carlo methods, extended so that the variance of the importance weights is offset by the inclusion of the CTMC. To derive these importance weights, we introduce a set of Radon-Nikodym derivatives of CTMCs over their path measures. Because the computation of these weights is intractable with standard neural network parameterizations of rate matrices, we devise a new compact representation for rate matrices via what we call *locally equivariant* functions. To parameterize them, we introduce a family of locally equivariant multilayer perceptrons, attention layers, and convolutional networks, and provide an approach to make deep networks that preserve the local equivariance. This property allows us to propose a scalable training algorithm such that the variance of the importance weights associated to the CTMC are minimal. We demonstrate the efficacy of our method on problems in statistical physics.

1 INTRODUCTION

A prevailing task across statistics and the sciences is to draw samples from a probability distribution whose probability density is known up to normalization. Solutions to this problem have applications in topics ranging across Bayesian uncertainty quantification (Gelfand & Smith, 1990), capturing the molecular dynamics of chemical compounds (Berendsen et al., 1984; Allen & Tildesley, 1987), and computational approaches to statistical and quantum physics (Wilson, 1974; Duane et al., 1987; Faulkner & Livingstone, 2023).

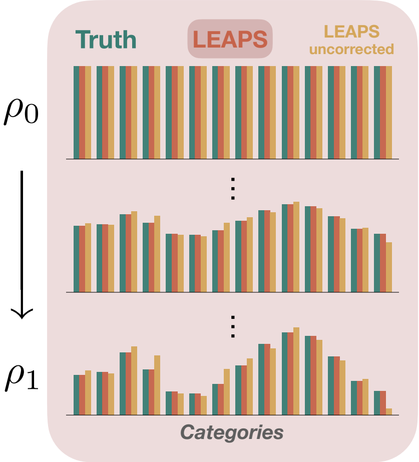

Figure 1: Illustration of the LEAPS algorithm. LEAPS allows to learn a dynamical transport of discrete distributions from $t = 0$ to $t = 1$ (blue). Samples are generated via the simulation of a Continuous-time Markov chain (yellow). Further, importance sampling weights allow to correct training errors (red).

The most salient approach to such sampling problems is Markov chain Monte Carlo (MCMC) (Metropolis et al., 1953; Robert et al., 1999), in which a randomized process is simulated whose equilibrium is the distribution of interest. While powerful and widely applied, MCMC methods can be inefficient as they suffer from slow convergence times into equilibrium, especially for distributions exhibiting multi-modality. Therefore, MCMC is often combined with other techniques that rely on non-equilibrium dynamics, e.g. via annealing from a simpler distribution with importance sampling (IS) (Kahn & Harris, 1951) or sequential Monte Carlo methods (SMC) (Neal, 2001; Doucet et al., 2001). Even then, the variance of these importance weights may be untenably large, and making sampling algorithms more efficient remains an active area of research. Inspired by the rapid progress in generative modeling, there has been extensive interest in augmenting contemporary sampling algorithms with learning (Noé et al., 2019; Albergo et al., 2019; Gabrié et al., 2022; Nicoli et al., 2020; Matthews et al., 2022).

Recently, there has been rapid progress in development of generative models using techniques from dynamical measure transport, i.e. where data from a base distribution is transformed into samples from the target distribution via flow or diffusion processes (Ho et al., 2020; Song et al., 2020; Albergo & Vanden-Eijnden, 2022; Albergo et al., 2023; Lipman et al., 2023; Liu et al., 2022). While there have been various developments on adapting these non-equilibrium dynamics for sampling in continuous state spaces (Zhang & Chen, 2022; Vargas et al., 2023; Máté & Fleuret, 2023; Tian et al., 2024; Albergo & Vanden-Eijnden, 2024; Richter & Berner, 2024; Akhound-Sadegh et al., 2024; Sun et al., 2024), there is a lack of existing literature on such approaches for discrete distributions.

However, discrete data are prevalent in various applications, such as in the study of spin models in statistical physics, protein and genomic data, and language. For generative modeling, a new family of models via continuous time Markov chains (CTMCs), commonly called ”discrete diffusion” models, for discrete state spaces (Campbell et al., 2022; Bengio et al., 2021; Austin et al., 2021; Lou et al., 2023; Gat et al.; Shaul et al., 2024; Campbell et al., 2024), have gained popularity. Here, we provide a new solution to the discrete sampling problem via CTMCs assuming access only to the unnormalized probability mass function. Our **main contributions** are:

- We introduce LEAPS, a sampler for discrete distributions via CTMCs that combines annealed importance sampling and sequential Monte Carlo with learned measure transport.
- To define the importance weights, we derive a Radon-Nikodym derivative for reverse-time CTMCs, control of which minimizes the variance of these weights.
- We show that the measure transport can be learnt and the variance of the importance weights minimized by optimizing a physics-informed neural network (PINN) loss function.
- We make the computation of the PINN objective scalable by introducing the notion of a locally equivariant network. We show how to build locally equivariant versions of common neural network architectures, including attention and convolutions.
- We experimentally verify the correctness and efficacy of the resulting LEAPS algorithm in high dimensions via simulation of the Ising model.

2 Setup and Assumptions

In this work, we are interested in the problem of sampling from a **target distribution** ρ_1 on a **finite state space** S. We refer to ρ_1 by its probability mass function (pmf) given by

$$\rho_1(x) = \frac{1}{Z_1}\exp(-U_1(x)) \quad (x \in S), \tag{1}$$

where we assume that we do not know the normalization constant Z_1 but only the function **potential** U_1. Our goal is to produce samples $X \sim \rho_1$. To achieve this goal, it is common to construct a **time-dependent probability mass function (pmf)** $(\rho_t)_{0\leq t\leq 1}$ over S which fulfils that ρ_0 has a distribution from which we can sample easily, e.g. $\rho_0 = \text{Unif}_S$, and ρ_1 is our target of interest. We write ρ_t as:

$$\rho_t(x) = \frac{1}{Z_t}\exp(-U_t(x)), \quad Z_t = \sum_{y\in S}\exp(-U_t(y)), \quad F_t = -\log Z_t \tag{2}$$

where Z_t (or equivalently F_t) is unknown. The value F_t is also called the **free energy**. Throughout, we assume that U_t is continuously differentiable in t. Note that for the case of $\rho_0 = \text{Unif}_S$ so that $U_0(x) = 0$, we get that $\rho_t \propto \exp(-tU_1(x))$ that can be considered a form of *temperature annealing*.

3 Background: Continuous-time Markov chains (CTMCs)

In this work, we seek to sample from ρ_1 using **continuous-time Markov chains (CTMC)**. A CTMC $(X_t)_{0\leq t\leq 1}$ is given by a set of random variables $X_t \in S$ $(0 \leq t \leq 1)$ whose evolution is determined by a time-dependent **rate matrix** $Q_t(y, x)$ $(0 \leq t \leq 1, x, y \in S)$ which fulfills the conditions:

$$Q_t(y; x) \geq 0 \quad (\text{for } y \neq x), \quad Q_t(x; x) = -\sum_{y\neq x} Q_t(y, x) \quad (\text{for } x \in S) \tag{3}$$

The rate matrix Q_t determines the **generator equation**

$$\mathbb{P}[X_{t+h} = y|X_t = x] = \mathbf{1}_{x=y} + hQ_t(y, x) + o(h) \tag{4}$$

for all $x, y \in S$ and $h > 0$ where $o(h)$ describes an error function such that $\lim_{h \to 0} o(h)/h = 0$. Our goal is to find a Q_t that is a solution to the **Kolmogorov forward equation (KFE)**

$$\partial_t \rho_t(x) = \sum_{y \in S} Q_t(x, y) \rho_t(y), \quad \rho_{t=0} = \rho_0. \tag{5}$$

Fulfilling the KFE is a necessary and sufficient condition to ensure that the distribution of walkers initialized as $X_0 \sim \rho_0$ and evolving according to (3) follow the prescribed path ρ_t, in particular such that $X_{t=1} \sim \rho_1$.

4 Importance Sampling with CTMCs

In general, the CTMC $(X_t)_{0 \leq t \leq 1}$ with arbitrary Q_t will have different marginals than ρ_t. To still obtain an unbiased estimator, it is common to use **importance sampling (IS)** to reweigh samples obtained while simulating X_t. Here, we introduce a time-evolving set of log-weights $A_t \in \mathbb{R}$ for $0 \leq t \leq 1$ to re-weight the distribution of X_t to a distribution μ_t defined such that for all $h : S \to \mathbb{R}$

$$\mathbb{E}_{x \sim \mu_t}[h(x)] = \frac{\mathbb{E}[\exp(A_t) h(X_t)]}{\mathbb{E}[\exp(A_t)]} \quad \Leftrightarrow \quad \mu_t(x) = \frac{\mathbb{E}[\exp(A_t) | X_t = x]}{\sum_{y \in S} \mathbb{E}[\exp(A_t) | X_t = y]},$$

where $\mathbb{E}[\cdot]$ denotes expectation over the process (X_t, A_t). Intuitively, the distribution μ_t is obtained by re-weighting samples from the current distribution of X_t. This effectively means that from a finite number of samples $(X_t^1, A_t^1), \dots, (X_t^n, A_t^n)$, we can obtain a Monte Carlo estimator of $\mathbb{E}_{x \sim \mu_t}[h(x)]$. Our goal is to find a scheme of computing A_t such that its reweighted distribution coincides with the target densities ρ_t, i.e. $\mu_t = \rho_t$ for all $0 \leq t \leq 1$.

We next derive a proposed scheme of computing weights A_t. Before we provide a formal explanation from first principles, we first provide a *heuristic* derivation of our proposed scheme in the following paragraph. Intuitively, the log-weights A_t should accumulate the deviation from the true distribution of X_t to the desired distribution ρ_t. We can rephrase this as "accumulating the error of the KFE" that one may want to write as the difference between both sides of equation (5):

$$\partial_t \rho_t(x) - \sum_{y \in S} Q_t(x, y) \rho_t(y)$$

As we do not know the normalization constant Z_t, it is intuitive to divide by $\rho_t(x)$ and use that $-\partial_t F_t + \partial_t U_t(x) = -\partial_t \log \rho_t(x) = -\partial_t \rho_t(x)/\rho_t(x)$ to obtain after dropping the constant $-\partial_t F_t$

$$\mathcal{K}_t \rho_t(x) = \frac{\partial_t \rho_t(x)}{\rho_t(x)} - \sum_{y \in S} Q_t(x, y) \frac{\rho_t(y)}{\rho_t(x)} = -\partial_t U_t(x) - \sum_{y \in S} Q_t(x, y) \frac{\rho_t(y)}{\rho_t(x)} \tag{6}$$

where we defined a new operator $\mathcal{K}_t \rho_t$. Intuitively, the operator $\mathcal{K}_t$ measures the violation from the KFE in log-space and it is intuitive to define A_t as the accumulated error of that violation, i.e. as the integral

$$A_t = \int_0^t \mathcal{K}_s \rho_s(X_s) \mathrm{d}s \quad \Leftrightarrow \quad A_{t+h} \Rightarrow A_t + h \mathcal{K}_t \rho_t(X_t) \quad (t = 0, h, 2h, 3h, \dots) \tag{7}$$

We call this the **proactive update** as the update anticipates where X_t is jumping to. We next provide a rigorous characterization of A_t defined in this manner.

5 IS via Radon-Nikodym Derivatives

Apriori, it is not clear that the log-weights A_t that we obtain via the proactive rule provide a valid IS scheme. Next, we show that there are many possible IS schemes but the proactive update rule is *optimal* among a natural family of IS schemes. The state space that we are interested in is the space $\mathcal{X}$ of CTMC trajectories defined as $\mathcal{X} = \{X : [0, 1] \to S | X_{t^-} \text{ exists and } X_{t^+} = X_t\}$, i.e. all trajectories that are continuous from the right with left limits. Such trajectories are commonly called **càdlàg** trajectories. We consider *path distributions* (or *path measures*), i.e. probability distributions over trajectories. For a CTMC $\mathbf{X} = (X_t)_{0 \leq t \leq 1}$ with rate matrix Q_t and initial distribution μ, we denote the corresponding **path distribution** as $\overrightarrow{\mathbb{P}}^{\mu, Q}$ where the arrow $\overrightarrow{\mathbb{P}}$ denotes that we go forward in time. Similarly, we denote with $\overleftarrow{\mathbb{P}}^{\nu, Q'}$ a CTMC running in reverse time initialized with ν. We present the following proposition whose proof can be found in Appendix A:

Proposition 5.1. *Let μ, ν be two initial distributions over S. Let Q_t, Q'_t be two rate matrices. Then the Radon-Nikodym derivative of the corresponding path distributions running in opposite time over the time interval $[0, t]$ is given by:*

$$\log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\nu,Q'}}{\mathrm{d}\overrightarrow{\mathbb{P}}^{\mu,Q}}(\mathbf{X}) = \log\left(\frac{\nu(X_t)}{\mu(X_0)}\right) + \int\limits_0^t Q'_s(X_s, X_s) - Q_s(X_s, X_s)\mathrm{d}s + \sum_{X_s^- \neq X_s} \log\left(\frac{Q'_s(X_s^-, X_s)}{Q_s(X_s, X_s^-)}\right)$$

where we sum over all points where X_s jumps in the last term.

Let us now revisit our goal of finding an IS scheme to sample from the target distribution ρ_1. The key idea is to construct a CTMC running in reverse-time with initial distribution ρ_t and then use the RND from Proposition 5.1. For a function $h : S \to \mathbb{R}$, we can then express its expectation under ρ_t as:

$$\mathbb{E}_{x\sim\rho_t}[h(x)] = \mathbb{E}_{\mathbf{X}\sim\overleftarrow{\mathbb{P}}^{\rho_t,Q'}}[h(X_t)] = \mathbb{E}_{\mathbf{Y}\sim\overrightarrow{\mathbb{P}}^{\rho_0,Q}}\left[h(X_t)\frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\rho_t,Q'}}{\mathrm{d}\overrightarrow{\mathbb{P}}^{\rho_0,Q}}(\mathbf{X})\right] \tag{8}$$

i.e. the RND $\frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\rho_1,Q'}}{\mathrm{d}\overrightarrow{\mathbb{P}}^{\rho_0,Q}}(\mathbf{X})$ gives a valid set of importance weights. Note that this holds for *arbitrary* Q'_t. However, to sample efficiently, it is crucial that the IS weights have low variance. Therefore, we will now derive the *optimal* IS scheme of this form. Ideally the weights will have *zero* variance - in other words the RND $\frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\rho_1,Q'}}{\mathrm{d}\overrightarrow{\mathbb{P}}^{\rho_0,Q}}(\mathbf{X})$ will be constant $= 1$. This is the case if and only if the path measures are the same, i.e. if the CTMC in reverse time is a time-reversal of the CTMC running in forward time. It is well-known that this is equivalent to $Q'_t(y, x) = Q_t(x, y)q_t(y)/q_t(x)$ for all $y \neq x$ where q_t denotes the true marginal of X_t, i.e. $X_t \sim q_t$. As we strive to make $q_t = \rho_t$, it is natural to set $q_t = \rho_t$ and define $Q'_t = \bar{Q}_t$ as

$$\bar{Q}_t(y, x) = Q_t(x, y)\frac{\rho_t(y)}{\rho_t(x)} \quad \text{for all } y \neq x, \quad \bar{Q}_t(x, x) = -\sum_{y\in S, y\neq x} Q_t(x, y)\frac{\rho_t(y)}{\rho_t(x)} \tag{9}$$

Let us now return to the proactive update that we defined in (7). We can now rigorously characterize it. Plugging in the definition of $\bar{Q}$, we can use Proposition 5.1 to obtain the main result of this section:

Theorem 5.2. *For the proactivate updates A_t as defined in (7) and $\bar{Q}_t$ as defined in (9), it holds:*

$$A_t + F_t - F_0 = \log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\rho_t,\bar{Q}}}{\mathrm{d}\overrightarrow{\mathbb{P}}^{\rho_0,Q}}(\mathbf{X})$$

This implies that we obtain a valid IS scheme fulfilling:

$$\mathbb{E}_{x\sim\rho_t}[h(x)] = \frac{\mathbb{E}[\exp(A_t)h(X_t)]}{\mathbb{E}[\exp(A_t)]} \quad (0 \leq t \leq 1) \tag{10}$$

i.e. $\mu_t = \rho_t$ for all $0 \leq t \leq 1$. Further, A_t will have zero variance for every $0 \leq t \leq 1$ if and only if $X_t \sim \rho_t$ for all $0 \leq t \leq 1$.

A proof can be bound in Appendix B. This theorem can be seen as a discrete state space equivalent of the generalized version of the Jarzynski equality (Jarzynski, 1997; Vaikuntanathan & Jarzynski, 2008) that has also recently been used for sampling in continuous spaces (Vargas et al., 2024; Albergo & Vanden-Eijnden, 2024).

6 PINN OBJECTIVE

As a next step, we introduce a learning procedure for learning an optimal rate matrix of a CTMC. For this, we denote with Q_t^θ a parameterized rate matrix with parameters θ (e.g. represented in a neural network). Our goal is to learn Q_t^θ such that (5) is fulfilled for the corresponding CTMC X_t. By Theorem 5.2 this equivalent to minimizing the variance of the IS weights. To measure the variance the weights, it is common to use the log-variance divergence (Nüsken & Richter, 2023; Richter & Berner, 2023) given by

$$\mathcal{L}^{\text{log-var}}(\theta; t) = \mathbb{V}_{\mathbf{X}\sim\mathbb{Q}}[\log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\rho_t,\bar{Q}^\theta}}{\mathrm{d}\overrightarrow{\mathbb{P}}^{\rho_0,Q^\theta}}(\mathbf{X})] = \mathbb{V}_{\mathbf{X}\sim\mathbb{Q}}[A_t + F_t - F_0] = \mathbb{V}_{\mathbf{X}\sim\mathbb{Q}}[A_t]$$

where $\mathbb{Q}$ is a reference measuring whose support covers the support of $\overleftarrow{\mathbb{P}}^{\rho_t,\bar{Q}^\theta}$ and $\overrightarrow{\mathbb{P}}^{\rho_0,Q^\theta}$ and where we used that F_0, F_t are constants. The above loss is tractable but we can bound it by a loss that is computationally more efficient. To do so, we use an auxiliary **free energy network** $F_t^\phi : \mathbb{R} \to \mathbb{R}$ with parameters ϕ. Note that F_t^ϕ is a one-dimensional function and therefore induces minimal additional computational cost.

Proposition 6.1. *For any reference measure $\mathbb{Q}$, the **PINN-objective** defined by*

$$\mathcal{L}(\theta,\phi;t) = \mathbb{E}_{s\sim Unif_{[0,t]},x_s\sim\mathbb{Q}_s}\left[|\mathcal{K}_s^\theta\rho_s(x_s) - \partial_s F_s^\phi|^2\right]$$

has a unique minimizer (θ^*,ϕ^*) such that $Q_t^{\theta*}$ satisfies the KFE and $F_t^{\phi^*} = F_t$ is the free energy. Further, this objective is an upper bound to the log-variance divergence:

$$\mathcal{L}^{log\text{-}var}(\theta;t) \leq t^2\mathcal{L}(\theta,\phi;t)$$

In particular, if $\mathcal{L}(\theta,\phi;t)=0$, then also $\mathcal{L}^{log\text{-}var}(\theta;t)=0$ and the variance of the IS weights is zero.

A proof can be found in Appendix C. Note that we can easily minimize the PINN objective via stochastic gradient descent. It is ”off-policy” as the reference distribution $\mathbb{Q}$ is arbitrary. This objective can be seen as the CTMC equivalent of that in (Albergo & Vanden-Eijnden, 2024; Tian et al., 2024; Sun et al., 2024).

7 Efficient IS and training via local equivariance

We now turn to the question of how to make the above training procedure efficient. Note that for small state spaces S we could rely on analytical solutions to the KFE (Campbell et al., 2022; Shaul et al., 2024). In many applications, though, the state space S is so large that we cannot store $|S|$ elements efficiently in a computer. Often state spaces S are of the form $S=\mathcal{T}^d$ where $\mathcal{T}=\{1,\dots,N\}$ is a set of N tokens. One then defines a notion of a **neighbor** y of x, i.e. an element $y=(y_1,\dots,y_d)$ that differs from x in at most one dimension (i.e. $y_i \neq x_i$ at most one i). We denote as $N(x)$ the set of all neighbors of x. We then restrict functional form of the rate matrices to only allow for jumps to neighbors, i.e. $Q_t^\theta(y,x)=0$ if $y\notin N(x)$. One can then use a neural network Q_t^θ represented by the function

$$Q_t^\theta : S \to (\mathbb{R}^{N-1})^d, \quad x \mapsto (Q_t^\theta(\tau,i|x))_{i=1,\dots,d,\tau\in\mathcal{T}\setminus\{x_i\}}$$

to parameterize a rate matrix defined $Q_t(y,x) = Q_t^\theta(y^j,j|x)$ if $y\in N(x)$ and $y^j\neq x^j$. This parameterization is commonly used in the context of discrete markov models (”discrete diffusion models”) (Campbell et al., 2022; 2024). With that, the operator $\mathcal{K}_t^\theta$ in (6) becomes:

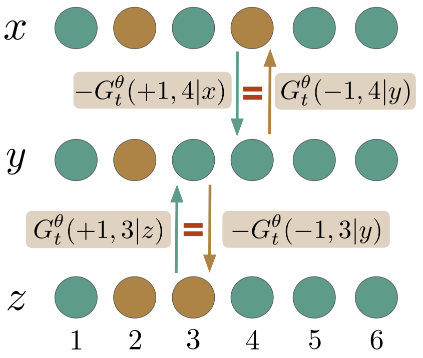

Figure 2: Visualization of local equivariance. Two tokens $\mathcal{T}=\{-1,+1\}$ and $d=6$. Local equivariance means that the *flux* to transition to a neighbor is the negative of the flux of transitioning from that neighbor back.

$$\mathcal{K}_t^\theta\rho_t(x) + \partial_t U_t(x) = \sum_{\substack{i=1,\dots,d\\ y\in N(x),\, y_i\neq x_i}}\left[Q_t^\theta(y^i,i|x) - Q_t^\theta(x^i,i|y)\frac{\rho_t(y)}{\rho_t(x)}\right]$$

The key problem with the above update is that it requires us to evaluate the neural network $|N(x)|$ times. Therefore, with the standard neural network parameterization, this update - and with that proactive IS sampling scheme and training via the PINN-objective - is very *inefficient*.

To make the computation of $\mathcal{K}_t^\theta$ efficient, we choose to build an **inductive bias** into our neural network architecture to compute $\mathcal{K}_t^\theta$ with no additional cost. Specifically, we introduce here the notion of **local equivariance**. A neural network F_t^θ represented by the function

$$F_t^\theta : S \to (\mathbb{R}^{N-1})^d, \quad x \mapsto (F_t^\theta(\tau,i|x))_{i=1,\dots,d,\tau\in\mathcal{T}\setminus\{x_i\}}$$

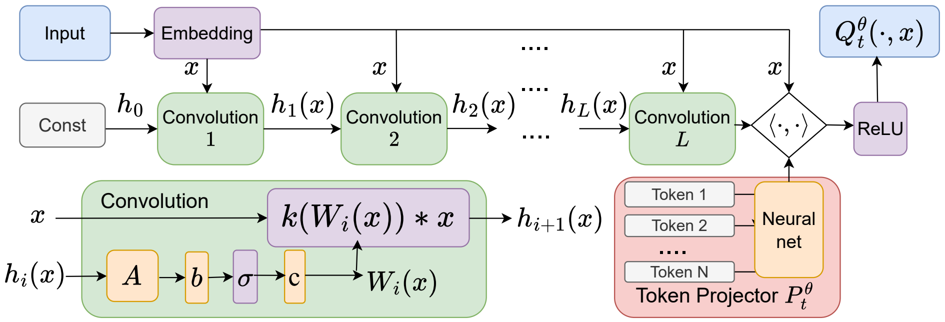

Figure 3: Overview of locally equivariant convolutional neural network architecture.

is called **locally equivariant** if the following condition holds for all $i = 1, \dots, d$:

$$F_t^\theta(\tau, i|x) = -F_t^\theta(x^i, i|\text{Swap}(x, i, \tau)) \quad \text{where} \quad \text{Swap}(x, i, \tau) = (x_1, \dots, x_{i-1}, \tau, x_{i+1}, \dots, x_d)$$

In other words, the function F_t^θ gives values for each neighbor of an element x. It is instructive to consider this value as a ”flux” going from x to each neighbor. Local equivariance says that the flux from x to its neighbor is negative the flux from the neighbor to x (see Figure 2). Therefore, every coordinate map F_j is equivariant with respect to transformations of the j-th input (“locally” equivariant). Note that we do not specify how F_i transforms for $i \neq j$ under transformations of x_j. This distinguishes it from ”full” equivariance as, for example, used in geometric deep learning (Bronstein et al., 2021; Weiler & Cesa, 2019; Thomas et al., 2018). We can use a locally equivariant neural network to parameterize a rate matrix via

$$Q_t^\theta(\tau, j|x) = [F_t^\theta(\tau, j|x)]_+, \tag{11}$$

where $[z]_+ = \max(z, 0)$ describes the ReLU operation. This representation is not a restriction (see Appendix D for a proof):

Proposition 7.1 (Universal representation theorem)**.** *For any CTMC model with marginals ρ_t, there is a corresponding CTMC with the same marginals ρ_t and a rate matrix that can be written as in (11) for a locally equivariant function F_t^θ.*

This representation allows to efficiently compute $\mathcal{K}_t^\theta$ in one forward pass of the neural network:

$$\mathcal{K}_t^\theta \rho_t(x) + \partial_t U_t(x) = \sum_{\substack{i=1,\dots,d \\ y \in N(x),\, y_i \neq x_i}} \left[[F_t^\theta(y^i, i|x)]_+ - [-F_t^\theta(y^i, i|x)]_+ \frac{\rho_t(y)}{\rho_t(x)} \right]$$

With this, we can efficiently compute the proactive IS update A_t and evaluate the PINN-objective. Therefore, this construction allows for scalable training and efficient importance sampling. We call the resulting algorithm **LEAPS** (**L**ocally **E**quivariant discrete **A**nnealed im**P**ortance **S**ampler). The acronym also highlights that we use a Markov *jump* process to sample (i.e. that takes ”leaps” through space). Finally, we note that while the sum in computing $\mathcal{K}_t^\theta$ includes values $\rho(y)$ for all neighbors y of x, this can be a considered a *discrete gradient*. For many scientific and physical models this requires often only 2× the computation compared to a single evaluation of $\rho_t(x)$.

8 Design of locally equivariant neural networks

It remains to be stated how to construct locally equivariant neural networks. We will focus on two fundamental designs used throughout deep learning: attention layers, and convolutional neural networks (in Appendix G, we discuss multilayer perceptrons). As usual, tokens are embedded as token vectors $e_\tau \in \mathbb{R}^{c_{\text{in}}}$ where c_{in} is the embedding dimension.

Locally-equivariant attention (LEA) layer. Let us consider a self-attention layer operating on keys $k_j = k_j(x_j)$, queries $q_j = q_j(x_j)$, and values $v_j = v_j(x_j)$ - each of which is a function of element x_j. We define the locally equivariant attention layer then as:

$$F_t^\theta(\tau, j|x) = (\omega_\tau - \omega_{x_j})^T \sum_{s \neq j} \frac{\exp(k_s^T q_j)}{\sum\limits_{t \neq j} \exp(k_t^T q_j)} v_s$$

It can be shown that this layer is locally equivariant if the queries q_j are independent of the sign of x_j (i.e. $q_j(x_j) = q_j(-x_j)$) which can be easily achieved. By stacking across multiple attention heads, one can create a locally equivariant MultiHeadAttention (LEA) with this.

Locally-equivariant convolutional (LEC) network. Local equivariance is different from "proper" equivariance in that *the composition of locally equivariant functions is not locally equivariant* in general. Therefore, we cannot simply compose locally equivariant neural network layers as we would do with "proper" equivariant neural networks. In particular, the MLP (see Appendix G) and the attention layers cannot simply be composed as their composition would violate the local equivariance. This fundamentally changes considerations about how to compose layers and how to construct *deep* neural networks. We will now illustrate this for the case of convolutional neural networks. To construct a deep locally equivariant convolutional neural network (LEC), we assume that our data lies on a grid. A convolutional layer is characterized by a matrix $W \in \mathbb{R}^{(2k-1)\times(2k-1)}$ and its operation is denoted via $k(W) * x$ where k denotes the convolutional kernel with weights W. Here, we set the center of W to zero: $W_{kk} = 0$ (i.e. such that corresponding location is effectively ignored). To stack such layers, we can make the output of the previous layer feed into the *weights* of the next layer:

$$h_0 =(1, \dots, 1)^T, \quad W_i = \sigma(A_i h_i + b_i) + c_j, \quad h_{i+1} = k_t(W_i) * x, \quad H_t^\theta(x) = h_L$$

where $A_i \in \mathbb{R}^{d_i \times d_i}, b_i \in \mathbb{R}^{d_i}, c_i \in \mathbb{R}^{d_i}$ are learnable tensors which operate on each coordinate independently (i.e. a 1x1 convolution) and $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function to make it non-linear. We call the resulting network $H_t^\theta(x) = (H_t^\theta(1|x), \dots, H_t^\theta(d|x)$ the **prediction head**. Combined with a small network $P_t^\theta : \mathcal{T} \to \mathbb{R}^k$ that we call **token projector**, we define the full neural network as

$$F_t^\theta(\tau, j|x) = (P_t^\theta(e_\tau) - P_t^\theta(x_j))^T H_t^\theta(j|x)$$

In Appendix E, we verify that F_t^θ defined in this way is locally equivariant. With this construction, one can stack deep highly complex convolutional neural networks. Note that this convolutional neural network has two (separate) symmetries: it is geometrically translation equivariant and locally equivariant in the sense defined in this work.

9 Related Work

CTMCs (Campbell et al., 2022) have been used for various applications in generative modeling ("discrete diffusion" models), including text and image generation (Shi et al., 2024; Shaul et al., 2024; Sahoo et al., 2024) and molecular design (Gruver et al., 2023; Campbell et al., 2024; Lisanza et al., 2024). While here we use a RND for CTMC running in *reverse* time, one recovers the loss functions of these generative models considering a RND of two *forward* time CTMCs (in Appendix A).

Over the past decade there has been continued interest in combining the statistical guarantees of MCMC and IS with learning transport maps. A non-parametric version of this is described in (Marzouk et al., 2016), and a parametric version through coupling-based normalizing flows was used to study systems in molecular dynamics and statistical physics (Noé et al., 2019; Albergo et al., 2019; Gabrié et al., 2022; Wang et al., 2022). These methods were extended to weave normalizing flows with SMC moves (Arbel et al., 2021; Matthews et al., 2022). More recent research focuses on replacing the generative model with a continuous flow or diffusion (Zhang & Chen, 2022; Vargas et al., 2023; Akhound-Sadegh et al., 2024; Sun et al., 2024). Our method is similar in spirit to the results in (Albergo & Vanden-Eijnden, 2024; Vargas et al., 2024) but takes the necessary theoretical and computational leaps to make these approaches possible for discrete distributions.

The primary alternative to what we propose is to correct using importance weights arising from the estimate of the probability density computed using an autoregressive model (Nicoli et al., 2020). However, the computational cost of producing samples in this case scales naively as $O(d)$, whereas we have no such constraint *a priori* in our case so long as the error in the Euler sampling scheme is kept small. Other work focuses on discrete formulations of normalizing flows, but the performant version reduces to an autoregressive model (Tran et al., 2019).

10 Experiments

As a demonstration of the validity of LEAPS in high dimensions, we sample the Gibbs distribution associated to a 2-dimensional Ising model. We choose the Ising model because it is a well-studied

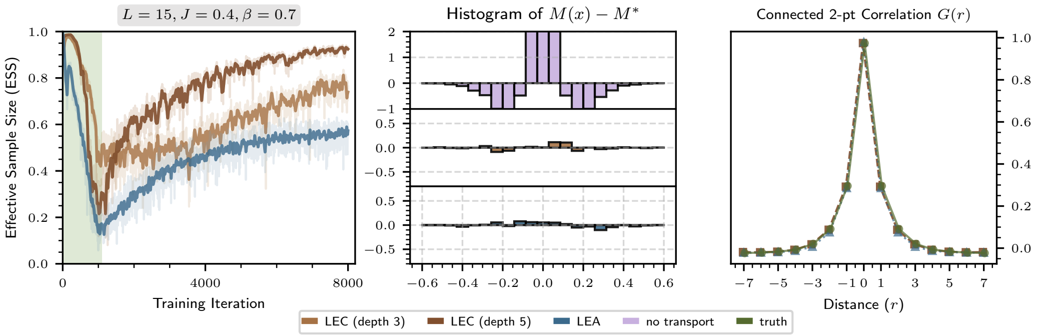

Figure 4: Performance metrics of LEAPS on the $L = 15, J = 0.4, \beta = 0.7$ Ising model. **Left:** Effective sample size of LEAPS samplers over training. Increasing the depth of LEC significantly improves performance. **Center:** Difference in the histograms of the magnetization $M(x)$ of configurations as compared to the ground truth set attained from a Glauber dynamics run of 25,000 steps, labeled as M^*. We denote by ”no transport” the case of using annealed dynamics with just the marginal preserving MCMC updates to show that the transport from Q_t is essential in our construction. **Right:** Comparison of the 2-point correlation function against the Glauber dynamics ground truth.

model. In particular, it is a solvable model, which allows us to construct a robust ground truth against which we can assess the various neural architectures underlying our algorithm. Configurations of the $L \times L$ Ising lattice follow the target distribution $\rho_1(x) = e^{-\beta H(x)+F_1}$ where β is the inverse temperature of the system, F_1 the free energy, and $H(x) : \{-1, 1\}^{L\times L} \to \mathbb{R}$ is the Hamiltonian for the model (see equation (12) for details). Neighboring spins are uncorrelated at high temperature but reach a critical correlation when the temperature drops behold a certain threshold. We use LEAPS to reproduce the statistics of the theory on a 15×15 lattice at parameters which approach this threshold, and compare our results against a ground truth of long-run Glauber dynamics, an efficient algorithm for simulation in this parameter regime. Note that this corresponds to a $d = 15 \times 15 = 225$ dimensional space. To make ρ_t time dependent, we make the parameters of J_t, μ_t, β_t linear functions of time, starting from the non-interacting case $J_0 = 0$.

Results. We compare three different realizations of our method, one using LEA, and the other two using deep LEC that vary in depth. For all generated samples, we use 100 steps of integration of (3). We benchmark them on the effective sample size (ESS), which is a standard measure of how many effective samples our model gives according to the variance of the importance weights (see details Appendix H). In addition, we compute various observables using the Ising configurations generated by our model, such as histograms of the magnetization compared to ground truth, as well as the two point connected correlation function. The latter is a measure of the dependency between spin values a distance r in lattice separation. In Figure 4, we show in the leftmost panel that the convolutional architecture outperforms the attention-based method, and the performance gap grows as we make the LEC network deeper. In the center panel, the difference in histograms of the magnetization of lattice configurations for our models as compared to ground truth samples is shown to be statistically zero, whereas relying on local MCMC alone for the same number of sampling steps (plotted in purple) illustrates that the dynamics have not converged. In the right plot, we see clear agreement between our learned sampler and the ground truth for the connected two point function. These results show that LEAPS can be an efficient simulator of complex, high dimensional target distributions.

11 Discussion

In this work, we present the LEAPS algorithm that allows to learn a non-equilibrium sampler via CTMCs parameterized by locally equivariant neural networks. A natural direction of future work will be to connect the ideas presented here with guidance or reward fine-tuning of generative CTMC models (discrete diffusion) - a problem known to be strongly tied to sampling. Further, LEAPS could easily be extended to sample across a whole family of distributions as opposed to only for a single, fixed target. Finally, we anticipate that the use of locally equivariant neural network combined with the IS scheme presented here might be useful more broadly for probabilistic models.

REFERENCES

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

M. S. Albergo, G. Kanwar, and P. E. Shanahan. Flow-based generative models for markov chain monte carlo in lattice field theory. *Phys. Rev. D*, 100:034515, Aug 2019. doi: 10.1103/PhysRevD.100.034515. URL https://link.aps.org/doi/10.1103/PhysRevD.100.034515.

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2022.

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

M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. 1987.

Michael Arbel, Alexander G. D. G. Matthews, and Arnaud Doucet. Annealed flow transport monte carlo. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 18–24 Jul 2021.

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

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics*, 81(8):3684–3690, 10 1984. ISSN 0021-9606. doi: 10.1063/1.448118. URL https://doi.org/10.1063/1.448118.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

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

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Arnaud Doucet, Nando de Freitas, and Neil J. Gordon (eds.). *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, 2001. ISBN 978-1-4419-2887-0. doi: 10.1007/978-1-4757-3437-9. URL https://doi.org/10.1007/978-1-4757-3437-9.

Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987. ISSN 0370-2693. doi: https://doi.org/10.1016/0370-2693(87)91197-X. URL https://www.sciencedirect.com/science/article/pii/037026938791197X.

Michael F. Faulkner and Samuel Livingstone. Sampling algorithms in statistical physics: a guide for statistics and machine learning, 2023. URL https://arxiv.org/abs/2208.04751.

Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. Adaptive monte carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10):e2109420119, 2022. doi: 10.1073/pnas.2109420119. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2109420119`.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Alan E. Gelfand and Adrian F. M. Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409, 1990. ISSN 01621459, 1537274X. URL `http://www.jstor.org/stable/2289776`.

Nate Gruver, Samuel Don Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=MfiK69Ga6p`.

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

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky TQ Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary markov processes. *arXiv preprint arXiv:2410.20587*, 2024.

C. Jarzynski. Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.*, 78:2690–2693, Apr 1997. doi: 10.1103/PhysRevLett.78.2690. URL `https://link.aps.org/doi/10.1103/PhysRevLett.78.2690`.

Herman Kahn and Theodore E Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards applied mathematics series*, 12:27–30, 1951.

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`.

Sidney Lyayuga Lisanza, Jacob Merle Gershon, Samuel W. K. Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel J. Hendel, Miriam K. Simma, Ge Liu, Muna Yase, Hongwei Wu, Claire D. Tharp, Xinting Li, Alex Kang, Evans Brackenbrough, Asim K. Bera, Stacey Gerben, Bruce J. Wittmann, Andrew C. McShan, and David Baker. Multistate and functional protein design using rosettafold sequence space diffusion. *Nature Biotechnology*, 2024. doi: 10.1038/s41587-024-02395-w. URL `https://doi.org/10.1038/s41587-024-02395-w`.

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

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

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. *Sampling via Measure Transport: An Introduction*, pp. 1–41. Springer International Publishing, Cham, 2016. ISBN 978-3-319-11259-6. doi: 10.1007/978-3-319-11259-6_23-1. URL `https://doi.org/10.1007/978-3-319-11259-6_23-1`.

Alex Matthews, Michael Arbel, Danilo Jimenez Rezende, and Arnaud Doucet. Continual repeated annealed flow transport Monte Carlo. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15196–15219. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/matthews22a.html`.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 06 1953. ISSN 0021-9606. doi: 10.1063/1.1699114. URL https://doi.org/10.1063/1.1699114.

Bálint Máté and François Fleuret. Learning interpolations between boltzmann densities, 2023. URL https://arxiv.org/abs/2301.07388.

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

Kim A. Nicoli, Shinichi Nakajima, Nils Strodthoff, Wojciech Samek, Klaus-Robert Müller, and Pan Kessel. Asymptotically unbiased estimation of physical observables with neural samplers. *Phys. Rev. E*, 101:023304, Feb 2020. doi: 10.1103/PhysRevE.101.023304. URL https://link.aps.org/doi/10.1103/PhysRevE.101.023304.

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. doi: 10.1126/science.aaw1147. URL https://www.science.org/doi/abs/10.1126/science.aaw1147.

Nikolas Nüsken and Lorenz Richter. Solving high-dimensional hamilton-jacobi-bellman pdes using neural networks: perspectives from the theory of controlled diffusions and measures on path space, 2023. URL https://arxiv.org/abs/2005.05409.

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

Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=h4pNROsO06.

Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky TQ Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. *arXiv preprint arXiv:2412.03487*, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=xcqSOfHt4g.

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

Jingtong Sun, Julius Berner, Lorenz Richter, Marius Zeinhofer, Johannes Müller, Kamyar Azizzadenesheli, and Anima Anandkumar. Dynamical measure transport and neural pde solvers for sampling, 2024. URL https://arxiv.org/abs/2407.07873.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

Yifeng Tian, Nishant Panda, and Yen Ting Lin. Liouville flow importance sampler. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48186–48210. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/tian24c.html.

Dustin Tran, Keyon Vafa, Kumar Krishna Agrawal, Laurent Dinh, and Ben Poole. Discrete flows: Invertible generative models of discrete data, 2019. URL https://arxiv.org/abs/1905.10347.

Suriyanarayanan Vaikuntanathan and Christopher Jarzynski. Escorted free energy simulations: Improving convergence by reducing dissipation. *Phys. Rev. Lett.*, 100:190601, May 2008. doi: 10.1103/PhysRevLett.100.190601. URL https://link.aps.org/doi/10.1103/PhysRevLett.100.190601.

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8pvnfTAbu1f.

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

Yihang Wang, Lukas Herron, and Pratyush Tiwary. From data to noise to data for mixing physics across temperatures with generative artificial intelligence. *Proceedings of the National Academy of Sciences*, 119(32):e2203656119, 2022. doi: 10.1073/pnas.2203656119. URL https://www.pnas.org/doi/abs/10.1073/pnas.2203656119.

Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in neural information processing systems*, 32, 2019.

Kenneth G. Wilson. Confinement of quarks. *Phys. Rev. D*, 10:2445–2459, Oct 1974. doi: 10.1103/PhysRevD.10.2445. URL https://link.aps.org/doi/10.1103/PhysRevD.10.2445.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: A stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_uCb2ynRu7Y.

A Proof of Proposition 5.1

Without loss of generality, we set the final time point to be $T = 1$. We compute for a bounded continuous function $\Phi : \mathcal{X} \to \mathbb{R}$:

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{X}\sim\overrightarrow{\mathbb{P}}^{\mu,Q}}[\Phi(\mathbf{X})] \\
=& \lim_{n\to\infty} \mathbb{E}_{\mathbf{X}\sim\overrightarrow{\mathbb{P}}^{\mu,Q}}[\Phi(X_0, X_{1/n}, X_{2/n}, \dots, X_{\frac{n-1}{n}}, X_1)] \\
=& \lim_{n\to\infty} \mathbb{E}_{\mathbf{X}\sim\overleftarrow{\mathbb{P}}^{\nu,Q'}}\left[\Phi(X_0, X_{1/n}, X_{2/n}, \dots, X_{\frac{n-1}{n}}, X_1)\frac{\overrightarrow{\mathbb{P}}^{\mu,Q}(X_0, X_{1/n}, \dots, X_{\frac{n-1}{n}}, X_1)}{\overleftarrow{\mathbb{P}}^{\nu,Q'}(X_0, X_{1/n}, X_{2/n}, \dots, X_{\frac{n-1}{n}}, X_1)}\right] \\
=& \lim_{n\to\infty} \mathbb{E}_{\mathbf{X}\sim\overleftarrow{\mathbb{P}}^{\nu,Q'}}\left[\Phi(X_0, X_{1/n}, X_{2/n}, \dots, X_{\frac{n-1}{n}}, X_1)\frac{\mu(X_0)}{\nu(X_1)} \prod_{s=0,1/n,2/n,\dots,\frac{n-1}{n}} \frac{\overrightarrow{\mathbb{P}}^{\mu,Q}(X_{s+h}|X_s)}{\overleftarrow{\mathbb{P}}^{\nu,Q'}(X_s|X_{s+h})}\right] \\
=& \lim_{n\to\infty} \mathbb{E}_{\mathbf{X}\sim\overleftarrow{\mathbb{P}}^{\nu,Q'}}\left[\Phi(\mathbf{X})\frac{\mu(X_0)}{\nu(X_1)} \exp\left(h \sum_{s,X_{s+h}=X_s} Q_s(X_s, X_s) - Q'_{s+h}(X_s, X_s)\right) \prod_{s,X_{s+h}\neq X_s} \frac{Q_s(X_{s+h}, X_s)}{Q'_{s+h}(X_s, X_{s+h})}\right] \\
=&\mathbb{E}_{\mathbf{X}\sim\overleftarrow{\mathbb{P}}^{\nu,Q'}}\left[\Phi(\mathbf{X})\frac{\mu(X_0)}{\nu(X_1)} \exp\left(\int_0^1 Q_s(X_s, X_s) - Q'_s(X_s, X_s)\mathrm{d}s\right) \prod_{s,X_{s^-}\neq X_s} \frac{Q_s(X_s, X_{s^-})}{Q'_s(X_{s^-}, X_s)}\right]
\end{aligned}
$$

where we used the definition of the rate matrix Q_t, Q'_t, the continuity of Q'_t in t and the fact that the left and right Riemann integral coincide. As Φ was arbitrary, the RND is given by:

$$
\log \frac{\mathrm{d}\overrightarrow{\mathbb{P}}^{\mu,Q}}{\mathrm{d}\overleftarrow{\mathbb{P}}^{\nu,Q'}}(\mathbf{X}) = \log(\mu(X_0)) - \log(\nu(X_1)) + \int_0^1 Q_s(X_s, X_s) - Q'_s(X_s, X_s)\mathrm{d}s + \sum_{s,X_s^-\neq X_s} \log\left(\frac{Q_s(X_s, X_s^-)}{Q'_s(X_s^-, X_s)}\right)
$$

B Proof of Theorem 5.2

Specifically, we use Proposition 5.1 to compute

$$
\begin{aligned}
\log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\rho_t,\bar{Q}_t}}{\mathrm{d}\overrightarrow{\mathbb{P}}^{\rho_0,Q_t}}(\mathbf{X}) =& \log(\rho_t(X_t)) - \log(\rho_0(X_0)) + \int_0^t \bar{Q}_s(X_s, X_s) - Q_s(X_s, X_s)\mathrm{d}s + \sum_{s,X_s^-\neq X_s} \log\left(\frac{\bar{Q}_s(X_s^-, X_s)}{Q_s(X_s, X_s^-)}\right) \\
=&F_t - F_0 - U_t(X_t) + U_0(X_0) + \int_0^t \bar{Q}_s(X_s, X_s) - Q_s(X_s, X_s)\mathrm{d}s + \sum_{s,X_s^-\neq X_s} \log\left(\frac{\bar{Q}_s(X_s^-, X_s)}{Q_s(X_s, X_s^-)}\right)
\end{aligned}
$$

Note that the function $t \mapsto U_t(X_t)$ is a piecewise differentiable function. Therefore, we can apply the fundamental theorem on every differentiable ”piece” and get:

$$
\begin{aligned}
U_t(X_t) - U_0(X_0) =& \int_0^t \partial_s U_t(X_t)\mathrm{d}s + \sum_{s,X_s^-\neq X_s} U_s(X_s) - U_s(X_s^-) \\
=& \int_0^t \partial_s U_s(X_s)\mathrm{d}s + \sum_{s,X_s^-\neq X_s} \log \frac{\rho_s(X_s^-)}{\rho_s(X_s)}
\end{aligned}
$$

Next, we can insert the above equation and get:

$$
\begin{aligned}
&\log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}^{\rho_t,\bar{Q}_t}}{\mathrm{d}\overrightarrow{\mathbb{P}}^{\rho_0,Q_t}}(\mathbf{X}) \\
=&F_t - F_0 - U_t(X_t) + U_0(Y_0) + \int\limits_0^t \bar{Q}_s(X_s, X_s) - Q_s(X_s, X_s)\mathrm{d}s + \sum_{s, X_s^- \neq X_s} \log\left(\frac{\bar{Q}_s(X_s^-, X_s)}{Q_s(X_s, X_s^-)}\right) \\
=&F_t - F_0 - \int\limits_0^t \partial_s U_s(X_s)\mathrm{d}s - \sum_{s, X_s^- \neq X_s} \log \frac{\rho_s(X_s^-)}{\rho_s(X_s)} + \int\limits_0^t \bar{Q}_s(X_s, X_s) - Q_s(X_s, X_s)\mathrm{d}s + \sum_{s, X_s^- \neq X_s} \log\left(\frac{\bar{Q}_s(X_s^-, X_s)}{Q_s(X_s, X_s^-)}\right) \\
=&F_t - F_0 - \int\limits_0^t \partial_s U_s(X_s)\mathrm{d}s + \int\limits_0^t \bar{Q}_s(X_s, X_s) - Q_s(X_s, X_s)\mathrm{d}s + \sum_{s, X_s^- \neq X_s} \log\left(\underbrace{\frac{\bar{Q}_s(X_s^-, X_s)}{Q_s(X_s, X_s^-)}\frac{\rho_s(X_s)}{\rho_s(X_s^-)}}_{=1}\right) \\
=&F_t - F_0 - \int\limits_0^t \partial_s U_s(X_s)\mathrm{d}s + \int\limits_0^t - \sum_{y \neq X_s} Q_s(X_s, y)\frac{\rho_t(y)}{\rho_t(X_s)} - Q_s(X_s, X_s)\mathrm{d}s + 0 \\
=&F_t - F_0 + \left[-\int\limits_0^t \partial_s U_s(X_s)\mathrm{d}s - \int\limits_0^t \sum_{y \in S} Q_s(X_s, y)\frac{\rho_t(y)}{\rho_t(X_s)}\mathrm{d}s\right] \\
=&F_t - F_0 + A_t
\end{aligned}
$$

where we used the definition of A_t in (7) and the definition of $\bar{Q}_t$ in (9). Note that for $h = 1$, we get that

$$1 = \mathbb{E}_{x\sim\rho_t}[h(x)] = \mathbb{E}[\exp(A_t + F_t - F_0)] = \mathbb{E}[\exp(A_t)]\exp(F_t - F_0)$$

because F_t, F_0 are constants. Therefore, in particular $\mathbb{E}[\exp(A_t)] = \exp(F_0 - F_t) = Z_t/Z_0$. Note that we assume that $Z_0 = 1$ as we know ρ_0. Therefore, $\mathbb{E}[\exp(A_t)] = Z_t$. This proves (10).

C Proof of Proposition 6.1

We can use the variational formulation of the variance as the minimizer of the mean squared error to derive a computationally more efficient upper bound, i.e. we can re-express for every $0 \leq t \leq 1$:

$$
\begin{aligned}
&\mathcal{L}^{\text{log-var}}(\theta; t) \\
=&\mathbb{V}_{\mathbf{X}\sim\mathbb{Q}}[A_t] \\
=&\min_{\hat{F}_t \in \mathbb{R}} \mathbb{E}_{\mathbf{X}\sim\mathbb{Q}}[|A_t - \hat{F}_t|^2] \\
=&t^2 \min_{\partial_s \hat{F}_s \in \mathbb{R}, 0 \leq s \leq t} \mathbb{E}_{\mathbf{X}\sim\mathbb{Q}}\left[|\frac{1}{t}\int\limits_0^t \mathcal{K}_s^\theta \rho_s(X_s) - \partial_s \hat{F}_s \mathrm{d}s|^2\right] \\
\leq&t^2 \min_{\partial_s \hat{F}_s \in \mathbb{R}, 0 \leq s \leq t} \mathbb{E}_{\mathbf{X}\sim\mathbb{Q}}\left[\frac{1}{t}\int\limits_0^t |\mathcal{K}_s^\theta \rho_s(X_s) - \partial_s \hat{F}_s|^2 \mathrm{d}s\right] \\
=&t^2 \min_{\partial_s \hat{F}_s \in \mathbb{R}, 0 \leq s \leq t} \mathbb{E}_{s\sim\text{Unif}_{[0,1]}, X_s\sim\mathbb{Q}_s}\left[|\mathcal{K}_s^\theta \rho_s(X_s) - \partial_s \hat{F}_s|^2\right]
\end{aligned}
$$

where we used Jensen's inequality and denote with $\mathbb{Q}_s$ the marginal of $\mathbb{Q}$ at time s. We now arrive at the result by replacing the above with the free energy network F_t^ϕ. Further, note that the above bound is tight for $\mathbb{Q}$-almost every $\mathbf{X}$:

$$\mathcal{K}_s^\theta \rho_s(X_s) - \partial_s F_s = C(\mathbf{X}_{0:t})$$

is a constant in time s. However, this constant may depend on $\mathbf{X}$. Further, note that

$$\mathbb{Q}_s(X_s) = |\mathcal{K}_s^\theta \rho_s(X_s) - \partial_s F_s|^2$$

D Proof of Proposition 7.1

Before we prove the statement, we prove an auxiliary statement about one-way rate matrices. We call a rate matrix Q_t a **one-way** rate matrix if

$$\begin{aligned} & Q_t(y,x)\neq 0 \quad \Rightarrow Q_t(x,y)=0 \quad \text{for all } x\neq y\\ \Leftrightarrow \quad & Q_t(y,x)=0 \quad \text{or} \quad Q_t(x,y)=0 \quad \text{for all } x\neq y\end{aligned}$$

Intuitively, a rate matrix Q_t is a one-way rate matrix if we can always only go from $x\to y$ or from $y\to x$. The next proposition shows that there is no problem restricting ourselves to one-way rate matrices.

Lemma D.1. *For every CTMC with rate matrix Q_t and marginals q_t, there is a one-way rate matrix $\bar{Q}_t$ such that its corresponding CTMC X_t has marginals q_t if $X_0\sim q_0$ is initialized with the same initial distribution. Furhter, if $Q_t(y,x)=0$ for $y\neq x$, then also $\bar{Q}_t(y,x)=0$.*

Proof. Let Q_t be a rate matrix defining a CTMC with marginals q_t. Then

$$\begin{aligned}
\partial_t q_t(x) =& \sum_{y\in S} Q_t(x,y)q_t(y)\\
=& \sum_{y\neq x} Q_t(x,y)q_t(y) - Q_t(y,x)q_t(x)\\
=& \sum_{y\neq x} \left[Q_t(x,y) - Q_t(y,x)\frac{q_t(x)}{q_t(y)}\right] q_t(y)\\
=& \sum_{y\neq x} \left[Q_t(x,y) - Q_t(y,x)\frac{q_t(x)}{q_t(y)}\right]_+ q_t(y) - \left[Q_t(y,x)\frac{q_t(x)}{q_t(y)} - Q_t(x,y)\right]_+ q_t(y)\\
=& \sum_{y\neq x} \left[Q_t(x,y) - Q_t(y,x)\frac{q_t(x)}{q_t(y)}\right]_+ q_t(y) - \left[Q_t(y,x) - Q_t(x,y)\frac{q_t(y)}{q_t(x)}\right]_+ q_t(x)\\
=& \sum_{y\neq x} \bar{Q}_t(x,y)q_t(y) - \bar{Q}_t(y,x)q_t(x)\\
=& \sum_{y\in S} \bar{Q}_t(x,y)q_t(y)
\end{aligned}$$

where we defined

$$\bar{Q}_t(y,x) = \begin{cases} \left[Q_t(y,x) - Q_t(x,y)\frac{q_t(y)}{q_t(x)}\right]_+ & y\neq x\\ -\sum\limits_{z\neq x} Q_t(z,x) & y=x\end{cases}$$

Note that

$$\begin{aligned}
\bar{Q}_t(y,x) >& 0\\
\Rightarrow \quad Q_t(y,x) >& Q_t(x,y)\frac{q_t(y)}{q_t(x)}\\
\Rightarrow \quad Q_t(y,x)\frac{q_t(x)}{q_t(y)} >& Q_t(x,y)\\
\Rightarrow \quad \left[Q_t(x,y) - Q_t(y,x)\frac{q_t(x)}{q_t(y)}\right]_+ =& 0\\
\Rightarrow \bar{Q}_t(x,y) =& 0
\end{aligned}$$

Therefore, we learn that $\bar{Q}_t$ fulfils the desired condition and fulfils the KFE. Therefore, we have proved that we can swap out Q_t for $\bar{Q}_t$ and we will have an CTMC with the same marginals. □

Now, let us return to the proof of Proposition 7.1. Given a rate matrix Q_t, we can now use a one-way rate matrix $\bar{Q}_t$ with the same marginals and define function:

$$F_t(\tau, i|x) = \begin{cases} \bar{Q}_t(y,x) & \text{if } Q_t(y,x) > 0 \\ -\bar{Q}_t(x,y) & \text{otherwise} \end{cases} \qquad \text{where } y = \text{Swap}(x, i, \tau)$$

By construction, it holds that $F_t(\tau, i|x)$ is locally equivariant and that $[F_t(\tau, i|x)]_+ = \bar{Q}_t(y, x)$. This finishes the proof.

E Local equivariance of ConvNet

To verify the local equivariance, one can compute

$$\begin{aligned} F_t^{\theta}(\tau, i|x) =& (P_t^{\theta}(e_\tau) - P_t^{\theta}(x_i))^T H_t^{\theta}(i|x) \\ =& -(P_t^{\theta}(x_i) - P_t^{\theta}(e_\tau))^T H_t^{\theta}(i|x) \\ =& -(P_t^{\theta}(x_i) - P_t^{\theta}(e_\tau))^T H_t^{\theta}(i|\text{Swap}(x, i, \tau)) \\ =& -F_t(x^i, i|\text{Swap}(x, i, \tau)), \end{aligned}$$

where we have used the invariance of the projection head $H_t^{\theta}(i|x)$ to changes in the i-th dimension. This shows the local equivariance.

F Recovering loss functions for CTMC models via RNDs

We discuss here in more detail how the Radon-Nikodym derivatives (RNDs) presented in Proposition 5.1 relate to the construction of loss function for CTMC generative models, also called ”discrete diffusion” models. The connection lies in the fact that the loss function of these models relies on RNDs of two CTMCs running both in forward time. We can prove the following statement:

Proposition F.1. *Let μ, ν be two initial distributions over S. Let Q_t, Q'_t be two rate matrices. Then the Radon-Nikodym derivative of the corresponding path distributions in forward time over the interval $[0, t]$ is given by:*

$$\log \frac{\mathrm{d}\overrightarrow{\mathbb{P}}^{\mu,Q}}{\mathrm{d}\overrightarrow{\mathbb{P}}^{\nu,Q'}}(\mathbf{X}) = \log \frac{\mathrm{d}\mu}{\mathrm{d}\nu}(X_0) + \int_0^t Q_s(X_s, X_s) - Q'_s(X_s, X_s) ds + \sum_{s, X_s^- \neq X_s} \log \left(\frac{Q_s(X_s, X_s^-)}{Q'_s(X_s, X_s^-)} \right)$$

where we sum over all points where X_s jumps in the last term.

The proof of the above formula is very similar to the proof of Proposition 5.1 and an analogous formula also appeared in (Campbell et al., 2024), for example. The above proposition allows us to by compute the KL-divergence:

$$\begin{aligned}
&D_{KL}(\overrightarrow{\mathbb{P}}_1^{\mu,Q}||\overrightarrow{\mathbb{P}}_1^{\nu,Q'})\\
\leq &D_{KL}(\overrightarrow{\mathbb{P}}^{\mu,Q}||\overrightarrow{\mathbb{P}}^{\nu,Q'})\\
= &\mathbb{E}_{\mathbf{X}\sim\overrightarrow{\mathbb{P}}^{\mu,Q}}\left[\log\frac{\mathrm{d}\,\overrightarrow{\mathbb{P}}^{\mu,Q}}{\mathrm{d}\,\overrightarrow{\mathbb{P}}^{\nu,Q'}}(\mathbf{X})\right]\\
= &D_{KL}(\mu||\nu)+\mathbb{E}_{\mathbf{X}\sim\overrightarrow{\mathbb{P}}^{\mu,Q}}\left[\int\limits_0^1 Q_t(X_t,X_t)-Q'_t(X_t,X_t)dt+\sum_{t,X_t^-\neq X_t}\log\left(\frac{Q_t(X_t,X_t^-)}{Q'_t(X_t,X_t^-)}\right)\right]\\
= &D_{KL}(\mu||\nu)+\mathbb{E}_{t\sim\text{Unif}_{[0,1]},x_t\sim\overrightarrow{\mathbb{P}}_t^{\mu,Q}}\left[Q_t(X_t,X_t)-Q'_t(X_t,X_t)\right]\\
&+\mathbb{E}_{\mathbf{X}\sim\overrightarrow{\mathbb{P}}^{\mu,Q}}\left[\sum_{t,X_t^-\neq X_t}\log\left(\frac{Q_t(X_t,X_t^-)}{Q'_t(X_t,X_t^-)}\right)\right]\\
= &D_{KL}(\mu||\nu)+\mathbb{E}_{t\sim\text{Unif}_{[0,1]},x_t\sim\overrightarrow{\mathbb{P}}_t^{\mu,Q}}\left[Q_t(X_t,X_t)-Q'_t(X_t,X_t)\right]\\
&+\int\limits_0^1\mathbb{E}_{X_t\sim\overrightarrow{\mathbb{P}}_t^{\mu,Q}}\left[\sum_{y\neq X_t}Q_t(y;X_t)\log\left(\frac{Q_t(y;X_t)}{Q'_t(y,X_t)}\right)\right]\mathrm{d}t\\
= &D_{KL}(\mu||\nu)+\mathbb{E}_{t\sim\text{Unif}_{[0,1]},X_t\sim\overrightarrow{\mathbb{P}}_t^{\mu,Q}}\left[\sum_{y\neq X_t}Q'_t(y,X_t)-Q_t(y,X_t)+Q_t(y;X_t)\log\left(\frac{Q_t(y;X_t)}{Q'_t(y,X_t)}\right)\right]
\end{aligned}$$

where we have used the data processing inequality in the first term. Having a parameterized model $Q'_t=Q^\theta_t$, this leads to the following loss:

$$\begin{aligned}
L(\theta)= &D_{KL}\left(\overrightarrow{\mathbb{P}}^{\mu,Q}||\overrightarrow{\mathbb{P}}^{\mu,Q_t^\theta}\right)\\
= &D_{KL}(\mu||\nu)+\mathbb{E}_{t\sim\text{Unif}_{[0,1]},X_t\sim\overrightarrow{\mathbb{P}}_t^{\mu,Q}}\left[\sum_{y\neq X_t}Q^\theta_t(y,X_t)-Q_t(y,X_t)\log\left(Q^\theta_t(y,X_t)\right)\right]+C
\end{aligned}$$

where Q_t is some reference process. The above recovers loss functions in the context of CTMC and jump generative models (Campbell et al., 2022; Gat et al.; Shaul et al., 2024; Campbell et al., 2024) and Euclidean jump models (Holderrieth et al., 2024, Section D.1.). Note that the above loss cannot be used for the purposes of sampling in a straight-forward manner because we do not have access to samples from the marginals of the ground reference $\overrightarrow{\mathbb{P}}^{\mu,Q}$.

G LOCALLY-EQUIVARIANT MULTILAYER PERCEPTRONS (MLPS)

Let us set $c_{\text{in}}=1$ in this paragraph for readability. Let $W^1,\dots,W^k\in\mathbb{R}^{d\times d}$ be a set of weight matrices with a zero diagonal, i.e. $W_{ii}=0$ for $i=1,\dots,d$. Further, let $\sigma:\mathbb{R}\to\mathbb{R}$ be an activation function and $\omega_\tau\in\mathbb{R}^k$ be a learnable projection vector for every token $\tau\in\mathcal{T}$. Then define the map:

$$F_t^\theta(\tau,j|x)=\sum_{i=1}^k(\omega^i_\tau-\omega^i_{x_j})\sigma(W^ix)_j$$

where $\sigma(W^ix)_j$ denotes the j-th element of the vector obtained by multiplying the vector x with the matrix W^i and applying the activation function a componentwise. One can easily show that this is a locally equivariant neural network corresponding to a MLP with one hidden layer.

H NUMERICAL EXPERIMENTS

Hamiltonian of Ising model. The Hamiltonian of the Ising model is given by

$$H(x)=-J\sum_{\langle i,j\rangle}x_ix_j+\mu\sum_i x_i. \tag{12}$$

Here, J is the interaction strength, $\langle i, j \rangle$ denotes summation over nearest neighbors of spins s_i, s_j, μ is the magnetic moment.

Effective Sample Size We use the self-normalized definition of the effective sample size such that, given the log weights A_t associated to N CTMC instances, the ESS at time t in the generation is given by:

$$\mathrm{ESS}_t = \frac{\left(N^{-1} \sum_{t=1}^{N} \exp\left(A_t^i\right)\right)^2}{N^{-1} \sum_{i=1}^{N} \exp\left(2A_i^i\right)} \tag{13}$$

H.1 Ising model experiments

Here we provide details of the numerical implementation of our study of the $L = 15$ Ising model. For the locally equivariant attention (LEA) mechanism, we use 40 attention heads, each with query, key, and value matrices of dimension 40x40. As such, there are about 350,000 parameters in the model. In addition, the locally equivariant convolutional net (LEC) of depth three uses kernel sizes of [5, 7, 15], while the depth five version uses [3, 5, 7, 9, 15], amounting to around 100,000 parameters.