FIXED-POINT NEURAL SAMPLERS IN DISCRETE SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Sampling from discrete, unnormalized distributions without access to data is a challenging problem. Neural samplers offer a promising approach by training generative models from density evaluations directly. Despite recent progress, existing discrete neural samplers are prone to mode collapse, come without convergence guarantees when trained via fixed-point iterations, and are often tied to a specific reference process such as masked or uniform diffusion. In this work, we introduce Discrete Gibbs Iterative Neural Sampler, a fixed-point neural sampler that addresses these limitations, enabling efficient, scalable learning, substantially reducing mode collapse in practice. Our framework builds on masked diffusion and also extends to transport between pairs of distributions. We demonstrate that the resulting method scales effectively to high-dimensional systems, supports amortized sampling across different conditions, and enables accurate estimation of alloy phase diagrams.

1 INTRODUCTION

Sampling from discrete distributions is a fundamental problem in physics, chemistry, and machine learning, yet remains computationally intractable in many settings of practical interest. Consider configurations $x = (x^1, \dots, x^D) \in \mathcal{X} = \mathcal{A}^D$, where D is the number of sites and each x^d takes values in a finite alphabet $\mathcal{A}$. Given an unnormalized density $\tilde{p} : \mathcal{X} \to \mathbb{R}$, the target distribution is

$$\pi(x) = \frac{\tilde{p}(x)}{Z}, \qquad Z = \sum_{x \in \mathcal{X}} \tilde{p}(x). \tag{1}$$

This formulation encompasses spin configurations, protein sequences, and distributions over solutions to combinatorial problems. Although the unnormalized density of an individual configuration can be evaluated, the exponentially large state space generally precludes enumeration. Markov chain Monte Carlo (MCMC) (Brooks et al., 2011) avoids evaluating Z, but can mix prohibitively slowly when high probability regions are separated by high energy barriers or narrow transition pathways.

The continuous counterpart of this problem has seen substantial progress through *neural samplers*, generative models trained on a target distribution specified by its energy function without precurated data. Boltzmann generators pioneered this idea for molecular equilibrium sampling with normalizing flows but suffered from scaling issues to systems of practical interest (Noé et al., 2019). More recent methods (Blessing et al., 2026c; Havens et al., 2026) have developed scalable training procedures for diffusion models based on fixed-point iterations by exploiting gradient information of the target density combined with efficient reuse of evaluated configurations. These advances have enabled neural samplers to tackle realistic materials systems, with substantial improvements in efficiency compared to classical Monte Carlo methods (Cheng et al., 2026).

In this work, we introduce Discrete Gibbs Iterative Neural Sampler (DIG-IN), a fixed-point neural sampler that is inspired by the principles underlying the successes of continuous neural samplers. At the core of our approach are *Gibbs labels*, i.e., normalized distributions over single-site perturbations of a discrete configuration that provide a discrete analogue of gradient. In summary, DIG-IN *i)* incorporates these Gibbs labels which provide a strong learning signal and help to avoid mode collapse (Section 3.1), *ii)* achieves a balance between variance and bias through tempered importance weighting, and enjoys convergence guarantees for certain setting (Section 3.2), and *iii)* is extendable to allow for a transport between any two pairs of distributions, whereas existing discrete constructions are often tied to specific reference distributions or corruption processes (Section 3.3).

Experiments on Ising and Potts models show that DIG-IN is the only method that avoids mode collapse in the ordered phase while giving accurate $\ln Z$ estimates. It scales to amortized sampling

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

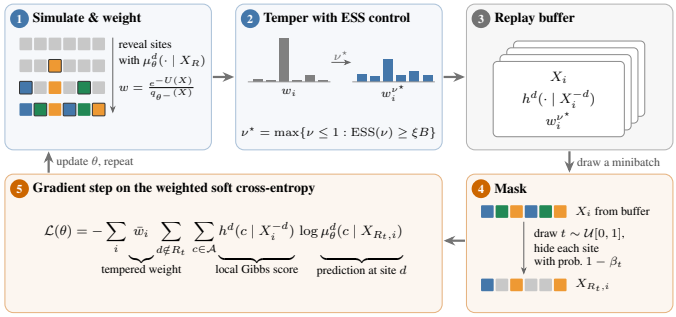


Figure 1: **DIG-IN iteration.** One training iteration consists of data collection, importance weight smoothing using ESS control, and parameter updates using a weighted cross-entropy loss.

Algorithm 1 DIG-IN

Input: target $\tilde{p}$, initial model, training loss; buffer size B, refill size b, minibatch size m, inner steps K, ESS fraction ξ.

Repeat until training is complete:

1: Generate b samples from current sampler, and append them to buffer.
2: Store their sample and importance weights w; keep the newest B buffer samples.
3: **for** $k = 1, \ldots, K$ **do**
4: Draw m buffer entries uniformly.
5: Mask sites randomly for each sample.
6: Calculate Gibbs labels for unmasked sites.
7: Choose $\nu^\star$ by ESS control (ξ).
8: Update model with $\bar{w}$-weighted cross-entropy loss with Gibbs label.
9: **end for**

Return: the trained sampler.

over thermodynamic conditions on lattices with up to $D = 2500$ sites, and it accurately recovers the phase diagrams of two alloy systems (Section 5).

2 Background: CTMCs and Masked Diffusion Models

This section introduces the basics of CTMCs and masked diffusion. These models can be formulated and trained in several ways; here, we focus on a simple background for our method.

Throughout, we write $N = |\mathcal{A}|$ for the alphabet size, x^{-d} for all coordinates except site d, and x_R for the values at a set of sites R. The notation $x^{d \leftarrow c}$ means replacing x^d by c.

Continuous-time Markov chains (CTMCs). A CTMC evolves on a discrete state space by remaining in its current state for a random time and then jumping to another. For a process $(X_t)_{t \in [0,1]}$, let $P_{t,t+h}(x, y) = \mathbb{P}(X_{t+h} = y \mid X_t = x)$ denote its transition probabilities over an interval of length $h > 0$. The *generator*, or rate matrix, describes their instantaneous change, $Q_t = \lim_{h \downarrow 0}(P_{t,t+h} - I)/h$, where I is the identity matrix. This definition gives the small-time expansion

$$\mathbb{P}(X_{t+h} = y \mid X_t = x) = \mathbf{1}_{\{y=x\}} + hQ_t(x, y) + o(h). \tag{2}$$

For $y \neq x$, the entry $Q_t(x, y) \geq 0$ is the rate of jumping from x to y. The diagonal entry $Q_t(x, x) = -\sum_{y \neq x} Q_t(x, y)$ is the negative total rate of leaving x, ensuring that each row sums to zero.

The process run backward in time is also a CTMC. Write $p_t(x) = \mathbb{P}(X_t = x)$ for the marginal distribution. Its reverse generator $\bar{Q}_t$, indexed by the same t but moving backward in time, satisfies

$$\bar{Q}_t(x, y) = Q_t(y, x)p_t(y)/p_t(x), \qquad y \neq x, \quad p_t(x) > 0. \tag{3}$$

The marginal density ratio is known as *concrete scores* (Lou et al., 2024) This is the discrete analogue of Nelson's time-reversal relation (Campbell et al., 2022; Sun et al., 2023). For generation, it is often sufficient to simulate the process in a single time direction. In contrast, path-space importance reweighting and free-energy estimators generally require evaluating both the forward and reverse path probabilities, and therefore access to both generators (He et al., 2025; 2026).

As is common in discrete generative modeling, we consider processes that change one site at each jump. For a state x with $x^d = a$, we denote the rate of replacing a by $c \neq a$ by

$$r_t^d(a, c \mid x^{-d}) := Q_t\big(x, x^{d \leftarrow c}\big). \tag{4}$$

The reverse single-site rates $\bar{r}_t^d$ are defined analogously using $\bar{Q}_t$. Together, the single-site rates determine when the process jumps, which site changes, and its new symbol.

Masked diffusion. A commonly used construction for CTMC in generative modeling is masked diffusion, which adds a new mask symbol $\mathtt{M} \notin \mathcal{A}$ to the alphabet. Masked diffusion then learns to reverse a process that progressively masks symbols until all tokens are fully masked. We first introduce

a decreasing schedule α_t, with $\alpha_0 = 1$ and $\alpha_1 = 0$. Given a clean sample $X_0 \sim \pi$, each coordinate of X_t retains its original symbol with probability α_t and is masked otherwise, independently across sites. Once masked, a site stays masked. Writing $\dot{\alpha}_t = \mathrm{d}\alpha_t/\mathrm{d}t$, the instantaneous masking rate is $r_t^d(c, \mathtt{M} \mid x^{-d}) = -\dot{\alpha}_t/\alpha_t$ for $c \in \mathcal{A}$ and $t < 1$.

By Equation (3), reversing these masking jumps requires the concrete scores. For masking, it suffices to learn a categorical denoiser. Let $R = \{d : x^d \neq \mathtt{M}\}$ denote the revealed sites of a partially masked state x. For $d \notin R$ and $c \in \mathcal{A}$, the desired predictor is $\mu_*^d(c \mid x_R) = \mathbb{P}_{X\sim\pi}(X^d = c \mid X_R = x_R)$. This recovers the time-reversal of a masking process as the concrete score is represented implicitly:

$$\frac{p_t(x^{d\leftarrow c})}{p_t(x)} = \frac{\alpha_t}{1-\alpha_t}\mu_*^d(c \mid x_R) \tag{5}$$

As we can see, even the reverse process requires a time-dependent generator, a time-independent denoiser suffices. Consequently, the generation can also be done by revealing one site at a time in a random order, yielding any-order autoregressive generation (Hoogeboom et al., 2022).

Given target samples, the predictor can be learned by letting them reconstruct masked tokens (Sahoo et al., 2024; Shi et al., 2024): draw $X \sim \pi$ and $t \sim \mathrm{Unif}(0, 1)$, then mask each site independently with probability $1 - \alpha_t$ and write R_t for the set of sites that remain revealed. Then we train the predictor by the cross-entropy loss

$$\mathcal{L}_{\mathrm{CE}}(\theta) = -\mathbb{E}_{X,t,R_t}[\sum_{d\notin R_t} \log \mu_\theta^d(X^d \mid X_{R_t})] = -\mathbb{E}_{X,t,R_t}[\sum_{d\notin R_t}\sum_{c\in\mathcal{A}} \mathbf{1}_{\{X^d=c\}} \log \mu_\theta^d(c \mid X_{R_t})], \tag{6}$$

where $\mathbf{1}_{\{X^d=c\}}$ is the one-hot label for the observed symbol at site d.

3 Discrete Gibbs Iterative Neural Sampler

Here, we introduce Discrete Gibbs Iterative Neural Sampler (DIG-IN). We first derive Gibbs labels from single-site energy differences and show that they provide a lower-variance training signal for the masked denoiser (Section 3.1). Since target samples are unavailable, we then turn this into a fixed-point iteration on self-generated samples, corrected by ESS-tempered importance weights, and state convergence analysis (Section 3.2). Finally, we extend the construction beyond masked diffusion to general discrete interpolants between two distributions (Section 3.3).

3.1 From energy differences to Gibbs soft labels

In continuous spaces, the score function $\nabla_x \log \pi(x)$ describes how a configuration x changes to increase its density. This information is crucial, as the unnormalized density only tells if a sample is good or bad, while the gradient tells how to change the sample, which provides stronger supervision signal for the neural sampler. For discrete configurations, the analogue is to compare the energies obtained by replacing the symbol at one site while holding the others fixed. These comparisons determine the target's conditional distribution over possible replacements, which we will use as additional learning signal for training a neural sampler as outlined below.

The gradient is an infinitesimal perturbation in continuous space. The discrete counterpart is a single-site replacement. For a complete configuration x, replacing the symbol at site d by c gives the energy difference, and plays the role of a gradient (Meng et al., 2022):

$$\nabla_{d,c}U(x) := -\log \pi(x^{d\leftarrow c}) + \log \pi(x). \tag{7}$$

Normalizing the corresponding Boltzmann weights gives

$$h^d(c \mid x^{-d}) = \frac{\exp[-\nabla_{d,c}U(x)]}{\sum_{a\in\mathcal{A}} \exp[-\nabla_{d,a}U(x)]} = \mathbb{P}_{X\sim\pi}(X^d = c \mid X^{-d} = x^{-d}). \tag{8}$$

We call this distribution a *Gibbs soft label*, since it is exactly the conditional distribution used by Gibbs sampling. Instead of asking the predictor to reconstruct only the sampled symbol, this label specifies the relative probabilities of all its alternatives under the target.

We use these labels to replace the one-hot targets in Equation (6). Assume for now that we have samples from our target distribution π. We can then train our network with the cross-entropy loss:

$$\mathcal{L}_{\mathrm{Gibbs}}(\theta) = -\mathbb{E}_{X\sim\pi,t,R_t}[\sum_{d\notin R_t}\sum_{c\in\mathcal{A}} h^d(c \mid X^{-d}) \log \mu_\theta^d(c \mid X_{R_t})]. \tag{9}$$

This has the same optimum as Equation (6) yet with lower per-site variance as stated below.

Proposition 3.1 (Consistency and per-site variance reduction)**.** *Let $X \sim \pi$. Fix a reveal set R, a masked site $d \notin R$, and predictor parameters θ. Define*

$$\ell^d_{\mathrm{CE}} = -\log \mu^d_\theta(X^d \mid X_R), \quad \ell^d_{\mathrm{Gibbs}} = -\sum_{c\in\mathcal{A}} h^d(c \mid X^{-d}) \log \mu^d_\theta(c \mid X_R). \tag{10}$$

If ℓ^d_{CE} has finite second moment, then

$$\mathbb{E}[\ell^d_{\mathrm{Gibbs}} \mid X_R] = \mathbb{E}[\ell^d_{\mathrm{CE}} \mid X_R], \quad \mathrm{Var}(\ell^d_{\mathrm{Gibbs}} \mid X_R) \le \mathrm{Var}(\ell^d_{\mathrm{CE}} \mid X_R). \tag{11}$$

This results follow from tower property and Rao-Blackwellization, since $\ell^d_{\mathrm{Gibbs}} = \mathbb{E}[\ell^d_{\mathrm{CE}} \mid X^{-d}]$.

3.2 From fixed-point consistency to fixed-point convergence

The previous construction leaves one important issue unresolved. Proposition 3.1 assumes that the completed configurations are distributed according to π, whereas obtaining such samples is precisely the problem we aim to solve. A similar obstacle arises in continuous-space neural samplers (Havens et al., 2025; Liu et al., 2025a; Shaul et al., 2025; Blessing et al., 2026c). These methods address it through a self-improving bootstrapping procedure: samples are first drawn from the current sampler, then used to construct supervision as if they were drawn from the target, and the sampler is subsequently refitted using this supervision. Repeating this procedure refines the sampler closer and closer to the target. Crucially, when the sampler already matches the target distribution, the resulting update leaves it unchanged; hence, the target distribution is a fixed point of the iteration. However, fixed-point consistency alone does not imply fixed-point convergence. Starting from an imperfect sampler, the distribution of training contexts can be biased, and repeatedly fitting exact local labels under this biased distribution may not recover the target. In this subsection, we aim to address it.

A straightforward idea is to fix this mismatch using directly importance sampling: let q_{θ^-} denote the frozen sampler that generates the current replay buffer. For a generated configuration x, define the unnormalized importance weight $w(x) = e^{-U(x)}/q_{\theta^-}(x)$. Here, $q_{\theta^-}(x)$ denotes the model likelihood assigned by the generation procedure to the sampled configuration. For masked diffusion, this probability is available from the categorical probabilities used during generation; see Section A.2 for further details. Applying the importance weighting to the Gibbs-label objective recovers the population loss that would have been obtained from target samples:

$$\mathcal{L}_{\mathrm{IW}}(\theta;\theta^-) \propto -\mathbb{E}_{X\sim q_{\theta^-}}\left[w(X)\mathbb{E}_{t,R_t}\left[\sum_{d\notin R_t}\sum_{c\in\mathcal{A}} h^d(c \mid X^{-d}) \log \mu^d_\theta(c \mid X_{R_t})\right]\right]. \tag{12}$$

By Theorem 3.1, this objective has the target masked conditional μ_* as its optimum.

In practice, however, when the current sampler is far from the target, the importance weights can become highly concentrated, leading to prohibitively high variance. This is a well-known challenge in importance-weighted learning, and prior work has addressed it through trust-region constraints (Schulman et al., 2015; Blessing et al., 2026a), proximal updates (Guo et al., 2026a), or smoothing (Cheng et al., 2026). Following these principles, we adopt *tempered importance weighting*, which introduces a controlled bias at each iteration in exchange for substantially reduced variance and more stable learning. Specifically, for a tempering exponent $\nu \in [0,1]$, we replace the importance weight $w(X)$ in Equation (12) by $\bar{w}(X) = w(X)^\nu$ and denote the corresponding loss as $\mathcal{L}_{\mathrm{FP}}(\theta;\theta^-,\nu)$. We choose ν adaptively according to the effective sample size of the B buffered configurations $x_1,\dots,x_B$, i.e., $\mathrm{ESS}(\nu,\theta^-) = \left(\sum_{i=1}^B \bar{w}(x_i)\right)^2 / \sum_{i=1}^B \bar{w}(x_i)^2$. Given a desired fraction $\xi \in (0,1)$, we use the largest exponent that maintains a sufficiently large effective sample size,

$$\nu^\star(\theta^-) = \max\left\{\nu \in [0,1] : \mathrm{ESS}(\nu,\theta^-) \ge \xi B\right\}. \tag{13}$$

Hence, when the current sampler already has good overlap with the target, $\nu^\star$ approaches one and the update uses nearly exact importance correction. When the overlap is poor, the correction is automatically softened to prevent a small number of configurations from dominating the optimization. Hence, starting from some θ_0, the fixed point iteration is given by

$$\nu_i = \nu^\star(\theta_i), \qquad \theta_{i+1} = \arg\min_\theta \ \mathcal{L}_{\mathrm{FP}}(\theta;\theta_i,\nu_i) \tag{14}$$

Under certain settings and idealized conditions stated below, we provide convergence guarantees for the population version of this tempered fixed-point iteration; proofs are in Section B.

Intuitively, tempering creates a tradeoff within each outer iteration. Full importance correction directly reproduces the target training distribution but may concentrate almost all weight on a few samples. Softer correction produces a more stable, but biased, training distribution. The key question is whether repeatedly applying these biased updates still moves the sampler toward the target.

To formalize this question, let $\Phi_\nu(g)$ denote one idealized outer iteration with tempering exponent ν: generate complete configurations using the current predictor g, form the corresponding tempered Gibbs-label objective, and return its exact population minimizer. Thus the population iteration is

$$g_{i+1} = \Phi_\nu(g_i). \tag{15}$$

Theorem 3.2 (Convergence of tempered fixed-point training)**.** *Suppose the state space is finite, the target and categorical predictors are strictly positive, and sampling and population fitting are exact. Assume also that the reveal policy is independent of the generated values and that training covers every predictor head queried during generation. Then, tempering changes how quickly and under what conditions the iteration approaches the target, but not its destination:*

(i) The target predictor μ_* is a fixed point of Φ_ν for every $\nu \in [0, 1]$. At $\nu = 1$, one population update recovers μ_* from any positive initialization.

(ii) With one fixed reveal order, the iteration converges globally to μ_* for every fixed $\nu \in [0, 1]$. It also converges when the exponent varies across iterations but remains bounded away from zero, which includes the ESS-adaptive rule in the idealized population setting.

(iii) With a general value-independent distribution over reveal orders, global convergence is guaranteed under the sufficient condition

$$2D(1 - \nu) < 1.$$

The fixed-point iteration, including pseudocode, is illustrated in Figure 1.

3.3 Beyond masked diffusion

We next extend the construction beyond masked diffusion to general discrete interpolants. Unlike the masked case where we have specified the masking process and learn the unmasking process with a single conditional predictor, for a general interpolant, however, we seek two transition dynamics, a forward and a backward process, that are exact time reversals of one another.

This mirrors the role of stochastic interpolants (Albergo et al., 2025) in continuous spaces, where a prescribed probability path can be associated with mutually time-reversed forward and backward dynamics. We proceed in two steps. First, we establish a general discrete interpolant construction that jointly defines forward and backward generators satisfying the time-reversal relation. We then show how the quantities defining these generators can be learned directly from endpoint energy functions, extending the Gibbs-supervision principle developed above beyond masked diffusion. We include detailed derivation in Section C and present experimental verification in Section F.1.

Constructing the Discrete interpolants. In summary, we first define an interpolant path between samples from the two endpoint distributions. A simple choice is to interpolate the endpoint samples $X_0 \sim p_0$ and $X_1 \sim p_1$ through the coordinate-factorized interpolant

$$K_t(x \mid x_0, x_1) = \prod_d \kappa_t^d(x^d \mid x_0^d, x_1^d), \qquad p_t(x) = \mathbb{E}[K_t(x \mid X_0, X_1)]. \tag{16}$$

Then, we seek to learn a pair of forward and backward CTMCs that are (1) time-reversal and (2) shares the same marginal density p_t along the time steps. This leads to the following requirements, derived from Nelson's relation and the Kolmogorov equations. Here, following the convention in discrete diffusion, we restrict the dynamics to single-site transitions:

$$\textbf{Nelson's:} \quad \bar{r}_t^d(a, c \mid x^{-d}) = r_t^d(c, a \mid x^{-d}) \frac{p_t(x^d = c, x^{-d})}{p_t(x^d = a, x^{-d})} = r_t^d(c, a \mid x^{-d}) \frac{p_t(x^d = c \mid x^{-d})}{p_t(x^d = a \mid x^{-d})}; \tag{17}$$

$$\textbf{Kolmogorov:} \quad \partial_t p_t(x) = \sum_d \sum_{c \neq a} r_t^d(c, a \mid x^{-d})\, p_t(x^d = c, x^{-d}) - r_t^d(a, c \mid x^{-d})\, p_t(x^d = a, x^{-d}). \tag{18}$$

Equation (17) suggests a natural symmetric parameterization of the forward and backward generators:

$$r_t^d(c, a|x^{-d}) = m_t^d(c, a|x^{-d}) p_t(x^d = a|x^{-d}); \tag{19}$$

$$\bar{r}_t^d(a, c|x^{-d}) = m_t^d(c, a|x^{-d}) p_t(x^d = c|x^{-d}), \tag{20}$$

and we only need to figure out the shared component m_t^d satisfying Equation (18). Since our interpolant is coordinate-wise, the time derivative of marginal density can be decomposed into summation of local terms: $\partial_t p_t = \sum_d \partial_t^{(d)} p_t$ with $\partial_t^{(d)} p_t(x^{-d}) = 0$. Plugging in these symmetric construction and the local decomposition, a sufficient condition for the Kolmogorov equation to hold can be imposed coordinate-wise as

$$\partial_t^{(d)} p_t(x^d = a \mid x^{-d}) = \sum_{c \neq a} \underbrace{p_t(x^d = a \mid x^{-d}) p_t(x^d = c \mid x^{-d}) \left[m_t^d(c, a \mid x^{-d}) - m_t^d(a, c \mid x^{-d}) \right]}_{\text{local flux } J_t^d(c,a|x^{-d})}. \tag{21}$$

Define the temporal potential: $\phi_t^d(a \mid x^{-d}) = \partial_t^{(d)} \log p_t(x^d = a \mid x^{-d})$ and we can obtain from Equation (21) that $\phi_t^d(a \mid x^{-d}) = \sum_c p_t(x^d = c \mid x^{-d}) \left[m_t^d(c, a \mid x^{-d}) - m_t^d(a, c \mid x^{-d}) \right]$. This shows that the Kolmogorov equation constrains only the antisymmetric part of m_t^d. Recall that Nelson's relation is satisfied by construction for any choice of m_t^d as well. We therefore can have the freedom to take the simplest circulation-free choice,

$$m_t^d(c, a \mid x^{-d}) - m_t^d(a, c \mid x^{-d}) = \phi_t^d(a \mid x^{-d}) - \phi_t^d(c \mid x^{-d}). \tag{22}$$

A non-negative realization is $m_t^d(c, a \mid x^{-d}) = \left[\phi_t^d(a \mid x^{-d}) - \phi_t^d(c \mid x^{-d}) \right]_+$, which yields

$$r_t^d(c, a \mid x^{-d}) = p_t(x^d = a \mid x^{-d}) \left[\phi_t^d(a \mid x^{-d}) - \phi_t^d(c \mid x^{-d}) \right]_+, \tag{23}$$

$$\bar{r}_t^d(a, c \mid x^{-d}) = p_t(x^d = c \mid x^{-d}) \left[\phi_t^d(a \mid x^{-d}) - \phi_t^d(c \mid x^{-d}) \right]_+. \tag{24}$$

In other words, we only need to parameterize and learn $p_t^d(\cdot \mid x^{-d})$ and $\phi_t^d(\cdot \mid x^{-d})$.

Minimum kinetic energy local flux This choice corresponding to an antisymmetric local flux:

$$J_t^d(c, a|x^{-d}) = p_t(x^d = c|x^{-d}) \partial_t^{(d)} p_t(x^d = a|x^{-d}) - p_t(x^d = a|x^{-d}) \partial_t^{(d)} p_t(x^d = c|x^{-d}) \tag{25}$$

Notably, this constructed flux uniquely minimizes the probability-weighted kinetic energy: at t with $p_t > 0$, define the kinetic energy for a local flux j as

$$\mathcal{E}_t(j) = \frac{1}{2} \sum_d \sum_{x^{-d}} p_t(x^{-d}) \sum_a \sum_{c \neq a} \frac{j_t^d(a, c \mid x^{-d})^2}{p_t(x^d = a|x^{-d}) p_t(x^d = c|x^{-d})} \tag{26}$$

The unique minimizer that satisfies Kolmogorov is given by Equation (25). Similar construction was also derived and explored by Shaul et al. (2025).

Learning the discrete interpolants. Finally, we consider learning p_t^d and ϕ_t^d from energy and buffer samples. In our setting, samples from p_0 are directly available or easy to obtain, while p_1 is specified only through the energy U_1. For a generated endpoint X_1, we replace its value at site d by the Gibbs soft label $h_1^d(a \mid X_1^{-d}) \propto \exp[-U_1(X_1^{d \leftarrow a})]$. Under independent coupling $p_0 \otimes p_1$, define

$$\ell_{t,d}(c) = \sum_a h_1^d(a \mid X_1^{-d}) \kappa_t^d(c \mid X_0^d, a), \qquad \dot{\ell}_{t,d}(c) = \sum_a h_1^d(a \mid X_1^{-d}) \partial_t \kappa_t^d(c \mid X_0^d, a). \tag{27}$$

The two fields can then be learned with

$$\mathcal{L}_q(\theta) = -\mathbb{E} \sum_{d,c} \ell_{t,d}(c) \log p_{\theta,t}^d(c \mid X_t^{-d}), \quad \mathcal{L}_\phi(\psi) = \mathbb{E} \sum_{d,c} \left[\frac{1}{2} \ell_{t,d}(c) \phi_{\psi,t}^d(c \mid X_t^{-d})^2 - \dot{\ell}_{t,d}(c) \phi_{\psi,t}^d(c \mid X_t^{-d}) \right] \tag{28}$$

The intermediate state X_t is sampled directly from $K_t(\cdot \mid X_0, X_1)$, so the same endpoint pair and its energy-based labels can be reused across many draws of t and X_t. As in the masked case, because X_1 is generated by the current sampler rather than drawn from p_1, we weight each endpoint pair by the tempered importance weight $\bar{w}$ with ESS control. The two objectives above are therefore replaced in practice by their weighted versions. Detailed form and derivation can be found in Section C.2.

Since both $p_t^d(\cdot \mid x^{-d})$ and $\phi_t^d(\cdot \mid x^{-d})$ must not depend on the queried value x^d, we parameterize them with a hollow (or blind-spot) architecture (Laine et al., 2019; Sun et al., 2022; Zhao et al., 2026), whose architecture prevents the d-th output from accessing x^d while retaining the full context x^{-d}.

Table 1: **Ising and Potts lattice spin models.** We consider the Ising and Potts model at three different inverse temperatures β corresponding to disordered, critical and ordered phases. We consider three evaluation criteria: The effective sample size (ESS), the error in normalization constant estimation $|\Delta \ln Z|$ and the total variation distance between the empirical distribution over the q modes of the reference target density, denoted as $\mathrm{TVD}_{\mathrm{mode}}$. The table reports Mean±standard deviation over 3 seeds. In each column, blue shading is used to indicate methods whose pessimistic value lies within one standard deviation of the best method. Table entries with dashes indicate that the method does not support computing $\ln Z$ and ESS.

	disordered			critical			ordered		
Method	$\lvert\Delta \ln Z\rvert$ $\times 10^3 \downarrow$	ESS $\uparrow$	$\mathrm{TVD}_{\mathrm{mode}}$ $\downarrow$	$\lvert\Delta \ln Z\rvert$ $\times 10^3 \downarrow$	ESS $\uparrow$	$\mathrm{TVD}_{\mathrm{mode}}$ $\downarrow$	$\lvert\Delta \ln Z\rvert$ $\times 10^3 \downarrow$	ESS $\uparrow$	$\mathrm{TVD}_{\mathrm{mode}}$ $\downarrow$
Ising, 25×25, $q = 2$									
MDNS	$1.40_{\pm 0.79}$	$.961_{\pm .002}$	$.012_{\pm .004}$	$1.93_{\pm 1.00}$	$.925_{\pm .006}$	$.016_{\pm .006}$	$693_{\pm 0}$	$.983_{\pm .007}$	$.500_{\pm .000}$
PDNS	$2.86_{\pm 1.86}$	$.956_{\pm .001}$	$.014_{\pm .006}$	$1.47_{\pm 0.61}$	$.917_{\pm .008}$	$.016_{\pm .008}$	$693_{\pm 0}$	$.988_{\pm .003}$	$.500_{\pm .000}$
DASBS	–	–	$.007_{\pm .004}$	–	–	$.042_{\pm .028}$	–	–	$.447_{\pm .075}$
DASBS, memoryless	$601_{\pm 159}$	$.002_{\pm .002}$	$.010_{\pm .004}$	$821_{\pm 116}$	$.002_{\pm .001}$	$.360_{\pm .198}$	$825_{\pm 413}$	$.003_{\pm .003}$	$.500_{\pm .000}$
LEAPS	$660_{\pm 1}$	$.985_{\pm .001}$	$.007_{\pm .004}$	$2536_{\pm 146}$	$.006_{\pm .003}$	$.027_{\pm .014}$	$10680_{\pm 2968}$	$.000_{\pm .000}$	$.023_{\pm .007}$
DIG-IN (ours)	$1.07_{\pm 0.35}$	$.995_{\pm .000}$	$.006_{\pm .004}$	$1.63_{\pm 0.27}$	$.980_{\pm .001}$	$.015_{\pm .004}$	$0.75_{\pm 0.45}$	$.979_{\pm .011}$	$.014_{\pm .007}$
Potts, 16×16, $q = 3$									
MDNS	$1.40_{\pm 0.92}$	$.957_{\pm .004}$	$.008_{\pm .003}$	$11_{\pm 2}$	$.874_{\pm .007}$	$.017_{\pm .009}$	$1089_{\pm 2}$	$.944_{\pm .041}$	$.667_{\pm .000}$
PDNS	$1.33_{\pm 0.86}$	$.957_{\pm .005}$	$.008_{\pm .004}$	$8.31_{\pm 4.32}$	$.879_{\pm .003}$	$.013_{\pm .009}$	$628_{\pm 325}$	$.964_{\pm .008}$	$.444_{\pm .157}$
DASBS	–	–	$.005_{\pm .003}$	–	–	$.129_{\pm .026}$	–	–	$.441_{\pm .223}$
DASBS, memoryless	$91_{\pm 40}$	$.016_{\pm .002}$	$.004_{\pm .004}$	$187_{\pm 87}$	$.006_{\pm .002}$	$.290_{\pm .163}$	$814_{\pm 334}$	$.004_{\pm .002}$	$.584_{\pm .118}$
LEAPS	$163_{\pm 0}$	$.999_{\pm .000}$	$.007_{\pm .003}$	$1411_{\pm 366}$	$.004_{\pm .003}$	$.019_{\pm .011}$	$7693_{\pm 832}$	$.000_{\pm .000}$	$.032_{\pm .011}$
DIG-IN (ours)	$1.03_{\pm 0.28}$	$.997_{\pm .000}$	$.005_{\pm .002}$	$13_{\pm 2}$	$.971_{\pm .000}$	$.006_{\pm .003}$	$9.44_{\pm 1.34}$	$.974_{\pm .007}$	$.014_{\pm .011}$

4 Related work

Neural samplers on discrete spaces. Discrete neural samplers include AR models (Wu et al., 2019; Nicoli et al., 2020), GFlowNets (Bengio et al., 2021; Malkin et al., 2022), and CTMCs including LEAPS, MDNS, PDNS, Discrete ASBS, JANUS (Holderrieth et al., 2025a; Zhu et al., 2025; Guo et al., 2026b;a; Blessing et al., 2026b). Ours combines Gibbs soft-label supervision with tempered importance weighting, achieving a stable and scalable training and also mitigating mode collapsing.

Discrete interpolant and flow models. Our method supports construction for general interpolants between two arbitrary distributions. Discrete flow and generator matching also support learning dynamics between general distributions (Gat et al., 2024; Holderrieth et al., 2025b). However, they typically do not learn a pair of time-reversed forward and backward dynamics. Recently, He et al. (2026) develop a framework for learning mutually time-reversed forward and backward dynamics. However, they all consider data-based training, while our method learns the local conditional distribution and temporal potential from endpoint energies.

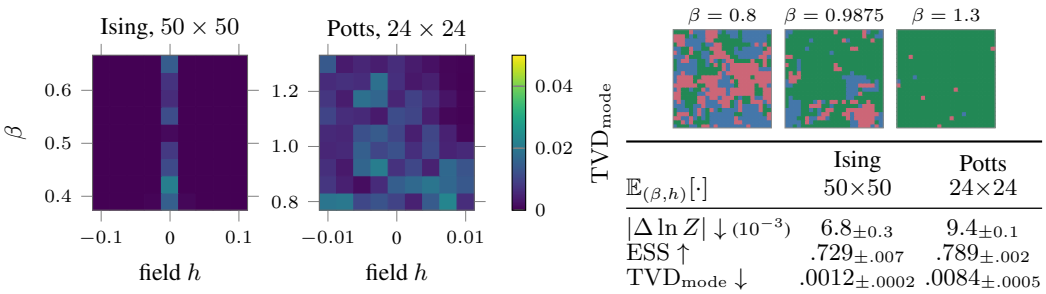
Appendix D provides a more detailed discussion on the related works.

5 Experiments

We perform several experiments on the Ising and Potts lattice spin models at different stages of the phase transition (Section 5.1). Next, we test our method on larger lattices, where we train an amortized sampler that is conditioned on two thermodynamic parameters (Section 5.2). Lastly, we consider two alloys, where the quantity of interest is a phase diagram (Section 5.3). We refer to Section E for details on the experimental setup and to Section F for further experiments, including a component ablation and a bridge between two Boltzmann distributions.

5.1 Ising and Potts lattice spin models

We use the Ising model on a 25×25 lattice ($D = 625$) and the Potts model on a 16×16 lattice ($D = 256$). Both place a q-state spin $\sigma_i \in \{1, \dots, q\}$ ($q = 2$ for Ising and $q = 3$ for Potts) on every site of a periodic square lattice and favor equal neighbors through the energies, i.e., $E(\sigma) = -\sum_{\langle ij \rangle} \delta_{\sigma_i \sigma_j}$ for Potts and $E(\sigma) = -\sum_{\langle ij \rangle} s_i s_j$ with spins $s_i = \pm 1$ for Ising with the convention $\delta_{\sigma_i \sigma_j} = (1 + s_i s_j)/2$. Hence, the target is $p_\beta(\sigma) \propto e^{-\beta E(\sigma)}$ at inverse temperature β. Each model is sampled at a disordered, a critical and an ordered β (Section E.1.1). Above β_c the target has q



$\mathbb{E}_{(\beta,h)}[\cdot]$	Ising 50×50	Potts 24×24
$\lvert\Delta \ln Z\rvert \downarrow (10^{-3})$	$6.8_{\pm 0.3}$	$9.4_{\pm 0.1}$
ESS ↑	$.729_{\pm .007}$	$.789_{\pm .002}$
$\mathrm{TVD}_{\mathrm{mode}} \downarrow$	$.0012_{\pm .0002}$	$.0084_{\pm .0005}$

Figure 2: **Amortized training across (β, h) of DIG-IN.** *Left:* a low $\mathrm{TVD}_{\mathrm{mode}}$ across all (β, h) indicates that our method avoids mode collapse and accurately learns the relative mode weights, which change under an external field h. *Right, top:* Potts samples of the amortized model at $h = 0$ in the disordered phase, near the critical point and in the ordered phase. *Right, bottom:* average ESS, $|\Delta \ln Z|$ and $\mathrm{TVD}_{\mathrm{mode}}$ over the 9×9 grid of (β, h) values, all computed with the importance weights. The results show mean $\pm$ standard deviation over three seeds.

symmetry-related modes. We compare against MDNS (Zhu et al., 2025), PDNS (Guo et al., 2026a), DASBS (Guo et al., 2026b) and LEAPS (Holderrieth et al., 2025a). All methods are trained for the same number of gradient steps and are evaluated identically. Besides the effective sample size (ESS) of the importance weights and the error $|\Delta \ln Z|$ of the normalization-constant estimate, we report the mode TVD ($\mathrm{TVD}_{\mathrm{mode}}$). Each draw is assigned to one of the q modes and $\mathrm{TVD}_{\mathrm{mode}}$ is the total variation distance between the resulting categorical distribution over the q modes and that of the reference. A detailed overview of computational costs is given in Table 3.

Away from the ordered phase the interpolant-based samplers, DIG-IN, MDNS and PDNS are all close to the exact $\ln Z$ on Ising and Potts, with DIG-IN leading in every $|\Delta \ln Z|$ column except critical Potts, where MDNS and PDNS are marginally better. Memoryless DASBS and LEAPS are one to three orders of magnitude further off. However, the ordered phase, which is multimodal, separates the samplers, where all methods except for DIG-IN and LEAPS suffer from mode collapse as shown by $\mathrm{TVD}_{\mathrm{mode}}$. The high $\ln Z$ error from LEAPS indicates that training is unstable leaving DIG-IN as the only approach that does not suffer from mode collapse while giving accurate $\ln Z$ estimates.

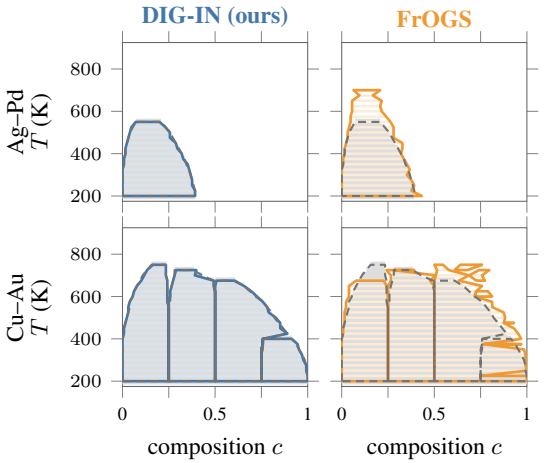
5.2 Amortized training

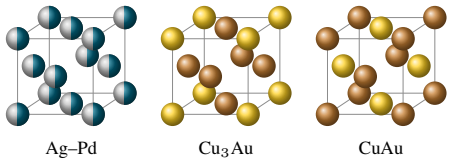
We train an amortized version of DIG-IN conditioned on (β, h) for the Ising and Potts model, where h is an external field. The external field results in a new target density given as $\log \tilde{p}(\sigma|\beta, h) = \beta(\sum_{\langle ij \rangle} s_i s_j + h \sum_i s_i)$ for Ising and $\log \tilde{p}(\sigma|\beta, h) = \beta(\sum_{\langle ij \rangle} \delta_{\sigma_i \sigma_j} + h\, N_0(\sigma))$ for Potts, where $N_0(\sigma)$ counts the sites in state 0. For the Ising model, we use a 50×50 lattice and therefore $D = 2500$. We consider $\beta \in [0.39, 0.65]$, and $|h| \leq 0.1$. For the Potts model we use a 24×24 lattice ($D = 576$) and $\beta \in [0.8, 1.3]$, $|h| \leq 0.01$. In this setting, a single sampler has to represent the disordered phase, criticality and the ordered phase. We evaluate DIG-IN on a 9×9 grid of (β, h).

When amortizing across multiple conditions and a phase transition in high dimensions, DIG-IN is able to achieve high ESS values and low $\ln Z$ error. Moreover, under an external field h, the q different modes do not have equal relative weight, meaning that the neural sampler not only has to avoid mode collapse but also needs to be able to accurately learn the relative weights of the modes. Figure 2 shows that DIG-IN is indeed able to achieve both, mode coverage and correct relative mode weighting, as indicated by the low $\mathrm{TVD}_{\mathrm{mode}}$ across all thermodynamic conditions (β, h).

5.3 Alloy phase diagrams

Mapping the phase diagram of an alloy requires sampling atomic configurations across a plane of chemical conditions $(T, \Delta\mu)$. We train one amortized sampler per alloy, conditioned on $(T, \Delta\mu)$, and compare DIG-IN with FrOGS (Min et al., 2026), which places a conditioned autoregressive prior (Du et al., 2026b) in front of learned LEAPS transport. The systems are those of Damewood et al. (2022) and Min et al. (2026), i.e., Ag–Pd on a $5{\times}5{\times}5$ fcc cell ($D = 125$), with a single miscibility gap, and Cu–Au on an $8{\times}4{\times}4$ cell ($D = 128$), with three ordered phases. The energy is a DFT-fitted cluster expansion and the target is semi-grand canonical over $T \in [200, 900]$ K and $\Delta\mu \in [-0.4, 0.4]$ eV.





$\mathbb{E}_{(T,\Delta\mu)}[\cdot]$	FrOGS	DIG-IN (ours)
Ag–Pd, $D = 125$		
$\lvert\Delta c\rvert \downarrow (10^{-3})$	5.25 ± 5.80	0.48 ± 0.00
$\lvert\Delta \ln Z\rvert/D \downarrow (10^{-4})$	27.0 ± 26.5	0.62 ± 0.01
ESS $\uparrow$	0.210 ± 0.067	0.973 ± 0.001
Cu–Au, $D = 128$		
$\lvert\Delta c\rvert \downarrow (10^{-2})$	2.35 ± 0.61	1.10 ± 0.01
$\lvert\Delta \ln Z\rvert/D \downarrow (10^{-4})$	112 ± 14	0.82 ± 0.04
ESS $\uparrow$	0.098 ± 0.055	0.874 ± 0.006

Figure 3: **Alloy phase diagrams.** *Left:* Ag–Pd ($D = 125$, top) and Cu–Au ($D = 128$, bottom). The grey dashed lines are the reference phase boundaries and the colored lines are the phase boundaries the model predicted. *Right, top:* example structures of different phases: Ag–Pd is a random fcc alloy, shown with mixed occupancy, and separates into Ag-rich and Pd-rich fcc below its miscibility gap; Cu–Au orders into Cu_3Au, CuAu and $CuAu_3$. *Right, bottom:* average composition error $|\Delta c|$, $|\Delta \ln Z|/D$ and ESS over the $(T, \Delta\mu)$ values, which are the thermodynamic conditions the model was trained on. The results show mean $\pm$ standard deviation over three seeds. The best results are highlighted using blue shading.

Each method trains one model per system for 60k steps on the same energy, grid and conditioning distribution with 3 seeds, and draws 5000 samples at each of the grid points. Phase diagrams are computed using $\log Z(T, \Delta\mu)$ estimates paired with the common-tangent construction of Min et al. (2026). For further details see Sections E.3.1 to E.3.3.

DIG-IN leads on every quantity measured, on both systems (Figure 3). Its free energy is one to two orders of magnitude closer to the reference and its ESS is significantly higher. Moreover, DIG-IN correctly recovers the phase boundaries without adding spurious ones on both systems. In contrast, FrOGS fails at finding all phase boundaries and adds several spurious ones.

6 Conclusion

We introduced DIG-IN, a fixed-point neural sampler for discrete, unnormalized distributions. DIG-IN uses Gibbs labels which turn local energy differences into a soft target for the masked denoiser that has the same optimum as the one-hot objective but lower variance. Since target samples are unavailable, DIG-IN trains on self-generated configurations and corrects the resulting bias with importance weights tempered under ESS control. Assuming exact sampling and exact population fitting, this iteration provably converges to the target: for a fixed reveal order at any tempering exponent, and for arbitrary reveal orders once the tempering exponent is sufficiently close to one. The same Gibbs supervision carries over from masked diffusion to general discrete interpolants between two distributions. Empirically, DIG-IN avoids mode collapse in the ordered phase of the Ising and Potts models while also giving accurate $\ln Z$ estimates, successfully scales to lattices with up to $D = 2500$ sites when amortizing over thermodynamic conditions, and recovers the phase diagrams of Ag–Pd and Cu–Au alloys.

Limitations and future work. Despite lowering or matching the computational costs compared to most baselines, Gibbs labels requires computing single-site energy differences. For local energies such as lattice models and cluster expansions, this table costs about one energy evaluation. For a black-box energy, however, it requires $D(N-1)+1$ evaluations per configuration. This motivates cheaper label estimators, for instance by subsampling sites. While we establish convergence guarantee for the certain setting of our algorithm, it does not cover the practical setting in our experiments all. Moreover, the benefit and extension of general interpolants are worth further studying. The broader applications of discrete samplers in biology and chemistry are left as future work.

AI USE STATEMENT

LLMs were used to draft parts of the paper from the authors' notes and polish the manuscript for grammar and consistency; help develop preliminary proofs; implement parts of the experiment code and the figure and table scripts; help with literature search; and review the draft for errors before submission. The research question, method, and experimental design originate from the authors.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *International Conference on Learning Representations*, 2025.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, 2021.

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

Piotr Białas, Piotr Korcyl, and Tomasz Stebel. Hierarchical autoregressive neural networks for statistical systems. *Computer Physics Communications*, 281:108502, 2022.

Piotr Białas, Piotr Korcyl, and Tomasz Stebel. Analysis of autocorrelation times in neural Markov chain Monte Carlo simulations. *Physical Review E*, 107(1):015303, 2023.

Piotr Białas, Piotr Korcyl, Tomasz Stebel, Adam Stefański, and Dawid Zapolski. Sampling two-dimensional spin systems with transformers. *arXiv preprint arXiv:2604.27738*, 2026.

Indaco Biazzo. The autoregressive neural network architecture of the Boltzmann distribution of pairwise interacting spins systems. *Communications Physics*, 6:296, 2023.

Raymond T. Birge. The calculation of errors by the method of least squares. *Physical Review*, 40: 207–227, 1932.

Denis Blessing, Julius Berner, Lorenz Richter, Carles Domingo i Enrich, Yuanqi Du, Arash Vahdat, and Gerhard Neumann. Trust region constrained measure transport in path space for stochastic optimal control and inference. *Advances in Neural Information Processing Systems*, 38:165462–165510, 2026a.

Denis Blessing, Mouyang Cheng, Maximilian Schebek, Jutta Rogal, Mingda Li, Carles Domingo-Enrich, and Yuanqi Du. JANUS: A multi-modal foundation neural sampler for disordered materials. *arXiv preprint arXiv:2608.19116*, 2026b.

Denis Blessing, Lorenz Richter, Julius Berner, Egor Malitskiy, and Gerhard Neumann. Bridge matching sampler: Scalable sampling via generalized fixed-point diffusion matching. *arXiv preprint arXiv:2603.00530*, 2026c. URL https://arxiv.org/abs/2603.00530.

Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P. Breckon, and Chris G. Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In *European Conference on Computer Vision*, 2022.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, 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. In *International Conference on Machine Learning*, 2024.

Arran Carter, Sanghyeok Choi, Kirill Tamogashev, Víctor Elvira, and Esmeralda S Whitammer. Discrete diffusion samplers and bridges: Off-policy algorithms and applications in latent spaces. *arXiv preprint arXiv:2602.05961*, 2026.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Wuhao Chen, Zijing Ou, and Yingzhen Li. Neural flow samplers with shortcut models. *arXiv preprint arXiv:2502.07337*, 2025. URL https://arxiv.org/abs/2502.07337.

Mouyang Cheng, Denis Blessing, Botao Yu, Gerhard Neumann, Mingda Li, Carles Domingo-Enrich, and Yuanqi Du. Atlas: A foundation neural sampler for amorphous materials. *arXiv preprint arXiv:2607.19198*, 2026.

James Damewood, Daniel Schwalbe-Koda, and Rafael Gómez-Bombarelli. Sampling lattices in semi-grand canonical ensemble with autoregressive machine learning. *npj Computational Materials*, 8 (1):61, 2022.

Luca Maria Del Bono, Federico Ricci-Tersenghi, and Francesco Zamponi. Nearest-neighbours neural network architecture for efficient sampling of statistical physics models. *Machine Learning: Science and Technology*, 6(2):025029, 2025.

Xiaochen Du, Juno Nam, Jaemoo Choi, Wei Guo, Sathya Edamadaka, Junyi Sha, Elton Pan, Yongxin Chen, Molei Tao, and Rafael Gómez-Bombarelli. MetaDNS: Enhancing exploration in discrete neural samplers via well-tempered metadynamics. *arXiv preprint arXiv:2605.21722*, 2026a.

Xiaochen Du, Juno Nam, Sulin Liu, and Rafael Gómez-Bombarelli. Scaling autoregressive models for lattice thermodynamics. *arXiv preprint arXiv:2603.14695*, 2026b.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems*, 2024.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-Predict: Parallel decoding of conditional masked language models. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. In *International Conference on Learning Representations*, 2025.

Wei Guo, Jaemoo Choi, Yuchen Zhu, Molei Tao, and Yongxin Chen. Proximal diffusion neural sampler. In *International Conference on Learning Representations*, 2026a.

Wei Guo, Yuchen Zhu, Xiaochen Du, Juno Nam, Yongxin Chen, Rafael Gómez-Bombarelli, Guan-Horng Liu, Molei Tao, and Jaemoo Choi. Discrete adjoint Schrödinger bridge sampler. *arXiv preprint arXiv:2602.08243*, 2026b. URL https://arxiv.org/abs/2602.08243.

Aaron Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Brandon Wood, Daniel Levine, Bin Hu, Brandon Amos, Brian Karrer, et al. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. *arXiv preprint arXiv:2504.11713*, 2025.

Aaron J Havens, Brian Karrer, and Neta Shaul. Flow sampling : Learning to sample from unnormalized densities via denoising conditional processes. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YlcyOMTPNl.

Jiajun He, Yuanqi Du, Francisco Vargas, Yuanqing Wang, Carla P Gomes, José Miguel Hernández-Lobato, and Eric Vanden-Eijnden. Feat: Free energy estimators with adaptive transport. *arXiv preprint arXiv:2504.11516*, 2025.

Jiajun He, Zijing Ou, Francisco Vargas, Yingzhen Li, José Miguel Hernández-Lobato, Carles Domingo-Enrich, and Yuanqi Du. Free energy estimation on any state space. *arXiv preprint arXiv:2605.31063*, 2026. doi: 10.48550/arXiv.2605.31063. URL https://arxiv.org/abs/2605.31063.

Mohamed Hibat-Allah, Estelle M. Inack, Roeland Wiersema, Roger G. Melko, and Juan Carrasquilla. Variational neural annealing. *Nature Machine Intelligence*, 3(11):952–961, 2021.

Peter Holderrieth, Michael S. Albergo, and Tommi Jaakkola. LEAPS: A discrete neural sampler via locally equivariant networks. *arXiv preprint arXiv:2502.10843*, 2025a. URL https://arxiv.org/abs/2502.10843.

Peter Holderrieth, Marton Havasi, Jason Yim, Neta Shaul, Itai Gat, Tommi Jaakkola, Brian Karrer, Ricky T. Q. Chen, and Yaron Lipman. Generator matching: Generative modeling with arbitrary Markov processes. In *International Conference on Learning Representations*, 2025b.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, 2021.

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with GFlowNets. In *International Conference on Machine Learning*, 2022.

Bruria Kaufman. Crystal statistics. II. Partition function evaluated by spinor analysis. *Physical Review*, 76(8):1232–1243, 1949.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. In *International Conference on Machine Learning*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Augustine Kong, Jun S. Liu, and Wing Hung Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association*, 89(425):278–288, 1994.

Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. *Advances in neural information processing systems*, 32, 2019.

Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin Kurt Miller, and Ricky T. Q. Chen. Adjoint Schrödinger bridge sampler. *arXiv preprint arXiv:2506.22565*, 2025a. URL https://arxiv.org/abs/2506.22565.

Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. In *International Conference on Learning Representations*, 2025b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2024.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in GFlowNets. In *Advances in Neural Information Processing Systems*, 2022.

B. McNaughton, M. V. Milošević, A. Perali, and S. Pilati. Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks. *Physical Review E*, 101(5):053312, 2020.

Chenlin Meng, Kristy Choi, Jiaming Song, and Stefano Ermon. Concrete score matching: Generalized score matching for discrete data. In *Advances in Neural Information Processing Systems*, 2022.

Kyucheol Min, Elyssa Hofgard, and Tess Smidt. FrOGS: Discrete neural sampler for independent alloy configurations across chemical conditions. *arXiv preprint arXiv:2609.02948*, 2026.

Kim A. Nicoli, Shinichi Nakajima, Nils Strodthoff, Wojciech Samek, Klaus-Robert Müller, and Pan Kessel. Asymptotically unbiased estimation of physical observables with neural samplers. *Physical Review E*, 101(2):023304, 2020.

Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text. In *International Conference on Learning Representations*, 2025a.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems*, 2025b.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *International Conference on Learning Representations*, 2025.

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.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. In *International Conference on Learning Representations*, 2025a.

Zijing Ou, Ruixiang Zhang, and Yingzhen Li. Discrete neural flow samplers with locally equivariant transformer. In *Advances in Neural Information Processing Systems*, 2025b.

Feng Pan, Pengfei Zhou, Hai-Jun Zhou, and Pan Zhang. Solving statistical mechanics on sparse graphs with feedback-set variational autoregressive networks. *Physical Review E*, 103(1):012103, 2021.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Avishek Joey Bose, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3942–3951, 2018.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems*, 2024.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. In *International Conference on Machine Learning*, 2025.

J. M. Sanchez, F. Ducastelle, and D. Gratias. Generalized cluster description of multicomponent systems. *Physica A: Statistical Mechanics and its Applications*, 128(1–2):334–350, 1984.

Sebastian Sanokowski, Wilhelm Berghammer, Sepp Hochreiter, and Sebastian Lehner. Variational annealing on graphs for combinatorial optimization. In *Advances in Neural Information Processing Systems*, 2023.

Sebastian Sanokowski, Sepp Hochreiter, and Sebastian Lehner. A diffusion model framework for unsupervised neural combinatorial optimization. In *International Conference on Machine Learning*, 2024.

Sebastian Sanokowski, Wilhelm Berghammer, Martin Ennemoser, Haoyu Peter Wang, Sepp Hochreiter, and Sebastian Lehner. Scalable discrete diffusion samplers: Combinatorial optimization and statistical physics. In *International Conference on Learning Representations*, 2025.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-Torre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. In *International Conference on Learning Representations*, 2025.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky T. Q. Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.03487.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, 2024.

Andy Shih, Dorsa Sadigh, and Stefano Ermon. Training and inference on any-order autoregressive models the right way. In *Advances in Neural Information Processing Systems*, 2022.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*, 2022.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. In *International Conference on Learning Representations*, 2023.

Robert H. Swendsen and Jian-Sheng Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88, 1987.

Benigno Uria, Iain Murray, and Hugo Larochelle. A deep and tractable density estimator. In *International Conference on Machine Learning*, 2014.

Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 1747–1756, 2016.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems*, 2025.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. In *International Conference on Machine Learning*, 2024.

Ulli Wolff. Collective Monte Carlo updating for spin systems. *Physical Review Letters*, 62(4): 361–364, 1989.

Dongyeop Woo, Minsu Kim, Minkyu Kim, Kiyoung Seong, and Sungsoo Ahn. Energy-based generator matching: A neural sampler for general state space. *arXiv preprint arXiv:2505.19646*, 2025. URL https://arxiv.org/abs/2505.19646.

Dian Wu, Lei Wang, and Pan Zhang. Solving statistical mechanics using variational autoregressive networks. *Physical Review Letters*, 122(8):080602, 2019.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. In *International Conference on Learning Representations*, 2025.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7B: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Dinghuai Zhang, Nikolay Malkin, Zhen Liu, Alexandra Volokhova, Aaron Courville, and Yoshua Bengio. Generative flow networks for discrete probabilistic modeling. In *International Conference on Machine Learning*, 2022.

Dinghuai Zhang, Hanjun Dai, Nikolay Malkin, Aaron C Courville, Yoshua Bengio, and Ling Pan. Let the flows tell: Solving graph combinatorial problems with gflownets. *Advances in neural information processing systems*, 36:11952–11969, 2023.

Ruixiang Zhang, Shuangfei Zhai, Yizhe Zhang, James Thornton, Zijing Ou, Joshua Susskind, and Navdeep Jaitly. Target concrete score matching: A holistic framework for discrete diffusion. In *International Conference on Machine Learning*, 2025.

Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. *arXiv preprint arXiv:2407.21243*, 2024.

Yixiu Zhao, Jiaxin Shi, Feng Chen, Shaul Druckmann, Lester Mackey, and Scott Linderman. Informed correctors for discrete diffusion models. *Advances in Neural Information Processing Systems*, 38: 125510–125538, 2026.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *International Conference on Learning Representations*, 2025.

Yuchen Zhu, Wei Guo, Jaemoo Choi, Guan-Horng Liu, Yongxin Chen, and Molei Tao. MDNS: Masked diffusion neural sampler via stochastic optimal control. *arXiv preprint arXiv:2508.10684*, 2025. URL https://arxiv.org/abs/2508.10684.

APPENDIX CONTENTS

A Additional method details **17**

A.1 Damping the fixed-point iteration . . . 17

A.2 Importance weighting and Free Energy Estimation . . . 17

B Convergence of DIG-IN **18**

B.1 Iterative predictor updates and convergence statement . . . 18

B.2 Gibbs-label fitting does not amplify log-ratio error . . . 19

B.3 Fixed-point exactness and global contraction for arbitrary reveal orders: proof of Theorem B.1*(i)–(ii)* . . . 20

B.4 Global convergence for a fixed reveal order: proof of Theorem B.1*(iii)* . . . 22

B.5 Global convergence of label-only fixed-order fitting: proof of Theorem B.1*(iv)* . . . 24

B.6 ESS adaptation preserves fixed-order convergence . . . 25

B.7 Tempering scales the local Jacobian . . . 27

B.8 Robustness to approximate population fitting . . . 28

C General discrete interpolants **28**

C.1 Minimum kinetic energy local flux . . . 29

C.2 Learning the discrete interpolants . . . 29

D Further related work **33**

E Experimental setup **34**

E.1 Lattice models . . . 34

E.2 Amortized training for lattice models . . . 36

E.3 Alloy phase diagrams . . . 37

F Further experiments **42**

F.1 Bridging two Boltzmann distributions . . . 43

F.2 Additional results . . . 43

A Additional method details

A.1 Damping the fixed-point iteration

Each step of the fixed-point iteration $\theta_{i+1} = \arg\min_\theta \mathcal{L}_{\mathrm{FP}}(\theta; \theta_i, \nu_i)$ fits the predictor to Gibbs labels computed on samples of the current sampler q_{θ_i}. To further stabilize training, we follow Blessing et al. (2026c) and use a damped fixed-point iteration, which replaces the Gibbs label by a mixture of the Gibbs label and the current model's own prediction, i.e.,

$$h^d_\eta(c \mid X) = \eta\, h^d(c \mid X^{-d}) + (1-\eta)\, \mu^d_{\theta_i}(c \mid X_R), \qquad \eta \in (0, 1], \tag{29}$$

and use it in place of h^d in the tempered loss:

$$\mathcal{L}^\eta_{\mathrm{FP}}(\theta; \theta_i, \nu_i) \propto -\mathbb{E}_{X\sim q_{\theta_i}}\Big[\bar{w}(X)\, \mathbb{E}_R\Big[\sum_{d\notin R}\sum_{c\in\mathcal{A}} h^d_\eta(c \mid X) \log \mu^d_\theta(c \mid X_R)\Big]\Big], \tag{30}$$

which gives the damped iteration $\theta_{i+1} = \arg\min_\theta \mathcal{L}^\eta_{\mathrm{FP}}(\theta; \theta_i, \nu_i)$. Here $\bar{w}$ are the tempered importance weights introduced in Section 3. Setting $\eta = 1$ recovers the undamped iteration. In practice μ_{θ_i} is the EMA model, which is also used to collect samples. If the minimization of (30) is exact, the damped update is therefore the average of the undamped update $\mu^{\eta=1}_{\theta_{i+1}}$ and the current predictor,

$$\mu_{\theta_{i+1}} = \eta\, \mu^{\eta=1}_{\theta_{i+1}} + (1-\eta)\, \mu_{\theta_i}. \tag{31}$$

If the sampler matches the target exactly, the undamped update leaves it unchanged, and by (31) the damped update does too. Conversely, a fixed point of the damped iteration is also fixed point of the undamped one.

A.2 Importance weighting and Free Energy Estimation

In this section, we provide details for calculating the importance weights w for both training and free energy estimation.

Masked diffusion. For a generated configuration x, the log importance weight is

$$\log w = \log \tilde{p}(x) - \sum_{\text{reveals } (R,d)} \log \mu^d_\theta(x^d \mid x_R), \tag{32}$$

where the sum runs over the reveals, with R the revealed set immediately before generating site d. This follows from the ratio of the forward masking and backward generation path densities with masking and unmasking order probability canceled (a more detailed derivation can be found in Appendix G in He et al. (2026)).

Any-order generation. For a configuration x generated in a sampled reveal order, the importance weight is the same as in Equation (32). The product follows the sampled order, using the probability assigned to each generated token given the previously revealed tokens. When the order is sampled independently of token values, its probability cancels between the augmented target and proposal.

General forward-backward processes. For endpoints $p_i = \tilde{p}_i/Z_i$, $i \in \{0, 1\}$, with evaluable unnormalized densities, a trajectory generated from p_0 using Q_t has log importance weight

$$\log w = \log \frac{\tilde{p}_1(X_1)}{\tilde{p}_0(X_0)} + \sum_{\substack{0<t<1\\ X_{t^-}\neq X_t}} \log \frac{\bar{Q}_t(X_t, X_{t^-})}{Q_t(X_{t^-}, X_t)} + \int_0^1 \big[\bar{Q}_t(X_t, X_t) - Q_t(X_t, X_t)\big]\, \mathrm{d}t. \tag{33}$$

For a time grid $0 = t_0 < \cdots < t_K = 1$, the discrete-time log importance weight is

$$\log w = \log \frac{\tilde{p}_1(X_1)}{\tilde{p}_0(X_0)} + \sum_{k=0}^{K-1} \log \frac{\bar{P}_{t_k|t_{k+1}}(X_{t_k}|X_{t_{k+1}})}{P_{t_{k+1}|t_k}(X_{t_{k+1}}|X_{t_k})}. \tag{34}$$

Here $P_{t_{k+1}|t_k}$ and $\bar{P}_{t_k|t_{k+1}}$ are the forward and backward transition densities used by the discrete-time sampler. The sum includes steps with $X_{t_{k+1}} = X_{t_k}$, so the holding-time contribution is already included.

Free-energy estimation. For M independent masked-diffusion or any-order samples, with weights w_i calculated using (32), we have

$$\mathbb{E}[w] = Z, \qquad \widehat{Z} = \frac{1}{M}\sum_{i=1}^{M} w_i, \qquad \widehat{F} = -\log \widehat{Z}, \tag{35}$$

where $F = -\log Z$ is the dimensionless free energy. For M independent forward trajectories initialized from p_0, the weights in (33) or (34) instead satisfy

$$\mathbb{E}[w] = \frac{Z_1}{Z_0}, \qquad \widehat{\Delta F} = -\log\left(\frac{1}{M}\sum_{i=1}^{M} w_i\right), \tag{36}$$

which estimates $\Delta F = -\log(Z_1/Z_0)$. This is an instance of the escorted Jarzynski and Crooks in discrete space (He et al., 2025; 2026).

B Convergence of DIG-IN

This section studies an idealized population version of the masked training loop, designed to isolate the dynamics of the fixed-point iteration from finite-sample and optimization errors. We replace the neural network by an unrestricted table of positive categorical heads g, one for every training-supported context. The state space is finite, the target distribution p is strictly positive, and both the sampling distribution and the tempered training objective are evaluated exactly. Each outer iteration returns the exact population minimizer rather than the result of finite neural-network optimization.

The reveal-order law ρ is fixed and independent of the generated values. Training masks are drawn independently of both the endpoint and the generating order, and their support covers every predictor head that can be queried during generation. Under these assumptions, one outer iteration is the deterministic map $g \mapsto \Phi_\nu(g)$ given contextwise by Equation (39). The analysis below establishes when repeated application of this map converges to the target heads.

B.1 Iterative predictor updates and convergence statement

With g denoting the frozen categorical predictors, for a reveal order $\sigma = (\sigma_1, \ldots, \sigma_D)$, let $q_g(x \mid \sigma)$ be the probability that the categorical predictors g generate the complete configuration x by revealing coordinates in the order σ:

$$q_g(x \mid \sigma) = \prod_{k=1}^{D} g^{\sigma_k}\left(x^{\sigma_k} \,\middle|\, x_{\sigma_{<k}}\right), \qquad \sigma_{<k} = (\sigma_1, \ldots, \sigma_{k-1}).$$

Marginalizing over the order σ while tempering the augmented weights induces the endpoint distribution

$$r_{\nu,g}(x) \propto p(x)^\nu \sum_\sigma \rho(\sigma) q_g(x \mid \sigma)^{1-\nu}. \tag{37}$$

Note that the fixed-order case is recovered by setting $\rho = \delta_{\sigma_0}$.

Let $\Phi_\nu(g)$ be the exact population minimizer of the Gibbs-label cross-entropy loss $\mathcal{L}_{\nu,g}$ when the completed configurations are distributed according to $r_{\nu,g}$, which admits the following expression:

$$\mathcal{L}_{\nu,g}(\widetilde{g}) = -\mathbb{E}_{X\sim r_{\nu,g},\, R}\left[\sum_{d\notin R}\sum_{c\in\mathcal{A}} h^d(c \mid X^{-d}) \log \widetilde{g}^d(c \mid X_R)\right]. \tag{38}$$

For each context (d, R, x_R),

$$[\Phi_\nu(g)]^d(c \mid x_R) = \mathbb{E}_{X\sim r_{\nu,g}}[h^d(c \mid X^{-d}) \mid X_R = x_R]. \tag{39}$$

The right-hand side is a conditional expectation under the current softened buffer, rather than an assumed target expectation. The following theorem, which is a longer version of Theorem 3.2, studies the behavior resulting from iteratively optimizing $\mathcal{L}_{\nu,g}$ and resetting g to $\Phi_\nu(g)$.

Theorem B.1 (Convergence of the population iteration)**.** *Assume a finite product space, $p(x) > 0$, positive categorical heads, exact sampling, and exact population fitting for the problem (38). Assume that reveal orders σ are sampled from ρ. Given an arbitrary categorical predictor g_0, define the sequence $(g_n)_{n\geq 0}$ such that $g_{n+1} = \Phi_\nu(g_n)$ for all $n \geq 0$. For any fixed order law ρ, define*

$$\Delta(g) = \max_{R,d,x_R} \operatorname{osc}_c \log[g^d(c \mid x_R)/g^d_*(c \mid x_R)], \tag{40}$$

where the maximum is over trained contexts and the oscillation $\operatorname{osc}_c f(c)$ is defined as $\max_c f(c) - \min_c f(c)$ for an arbitrary function f over a finite set, and $g^d_*(c \mid x_R) = p(X^d = c \mid X_R = x_R)$ is the target head.

(i) The target heads $g^d_*(c \mid x_R) = p(X^d = c \mid X_R = x_R)$ satisfy $\Phi_\nu(g_*) = g_*$ for every $\nu \in [0, 1]$. Full weighting gives $\Phi_1(g) = g_*$ for every positive initialization.

(ii) Then,

$$\Delta(\Phi_\nu(g)) \leq 2D(1-\nu)\Delta(g). \tag{41}$$

Thus a fixed $\nu > 1 - 1/(2D)$ guarantees global convergence of this sequence to g_* (since g is defined over a finite sets, all norms are equivalent).

(iii) With one fixed reveal order (i.e. setting $\rho = \delta_{\sigma_0}$ in (37)) and training on its prefixes, every fixed $\nu \in (0, 1)$ gives global convergence of the sequence $(g_n)_{n\geq 0}$ to g_* with error $\Delta(g_n) = O(k^{D-1}(1-\nu)^k)$ after k outer iterations.

(iv) With the same fixed-order training scheme, the label-only ($\nu = 0$) iteration $g_{k+1} = \Phi_0(g_k)$ also converges globally to g_* from every positive initialization.

The any-order threshold in *(ii)* is a conservative sufficient condition: in general we are interested in taking values of ν substantially smaller than 1 to avoid high importance weight variance. The fixed-order result in *(iii)* holds for an arbitrary ν, and covers an ordered masked bridge.

The following subsections prove the four statements: statements *(i)* and *(ii)* in Section B.3, statement *(iii)* in Section B.4, and statement *(iv)* in Section B.5. and in Section B.6 we show that the ESS rule (13) supplies a uniformly positive exponent in the fixed-order population setting. In Section B.7, we also show that tempering multiplies the Jacobian of the population update at the target by $1 - \nu$, yielding a complementary local stability condition. All of these results establish a target-directed population iteration: they do not assert that finite neural optimization is exact. In Section B.8 we explore how robust our arguments are to approximate fitting.

B.2 Gibbs-label fitting does not amplify log-ratio error

We begin by introducing a lemma that constitutes the backbone of our argument.

Lemma B.2. *Let $a, b : \mathcal{X} \to (0, +\infty)$ be arbitrary positive scalar-valued functions, with $\mathcal{X}$ a finite set, and K be a nonnegative linear map with positive output, defined as*

$$(Ka)(y) = \sum_x K(y, x)a(x). \tag{42}$$

Then,

$$\min_x \frac{a(x)}{b(x)} \leq \frac{(Ka)(y)}{(Kb)(y)} \leq \max_x \frac{a(x)}{b(x)}. \tag{43}$$

Proof. These inequalities hold because $(Ka)(y) = \sum_x K(y, x)b(x)\frac{a(x)}{b(x)}$, which means that $(Ka)(y)/(Kb)(y)$ can be regarded as the weighted average of the terms $\frac{a(x)}{b(x)}$ with normalized weights proportional to $K(y, x)b(x)$. □

Based on this result, the following proposition, which we use in Section B.3, shows that a certain operator associated to Gibbs-label fitting is non-expansive.

Proposition B.3 (Non-expansiveness of Gibbs-label fitting)**.** *Let p and r be strictly positive distributions on the finite product space $\mathcal{X}$. Define the exact Gibbs-label fitting operator by*

$$[\mathcal{T}(r)]^d(c \mid x_R) = \mathbb{E}_r\left[h^d(c \mid X^{-d}) \,\middle|\, X_R = x_R\right].$$

Then

$$\max_{(R,d,x_R)\in\mathcal{C}_{\text{train}}} \operatorname{osc}_c \log \frac{[\mathcal{T}(r)]^d(c \mid x_R)}{g_*^d(c \mid x_R)} \le \operatorname{osc}_x \log \frac{r(x)}{p(x)}. \tag{44}$$

Proof. Since by definition, $h^d(c \mid X^{-d}) = \mathbb{E}_p[\mathbf{1}_{X^d=c} \mid X^{-d}]$, the target distribution p is a fixed point of $\mathcal{T}$ as a consequence of the tower property of conditional expectation:

$$\begin{aligned}[\mathcal{T}(p)]^d(c \mid x_R) &= \mathbb{E}_p[h^d(c \mid X^{-d}) \mid X_R = x_R] = \mathbb{E}_p[\mathbf{1}_{X^d=c} \mid X_R = x_R] \\ &= p^d(X^d = c \mid x_R) = g_*^d(c \mid x_R).\end{aligned} \tag{45}$$

Fix a context (R, d, u), where $d \notin R$ and $u \in \mathcal{A}^R$, and define the nonnegative linear map

$$(K_{R,d,u}a)(c) = \sum_{x:\, x_R=u} a(x)h^d(c \mid x^{-d}).$$

Applying this map to an arbitrary r gives

$$(K_{R,d,u}r)(c) = r_R(u)[\mathcal{T}(r)]^d(c \mid u).$$

Applying it to p and using equation (45) yields

$$(K_{R,d,u}p)(c) = p_R(u)g_*^d(c \mid u).$$

Hence,

$$\frac{(K_{R,d,u}r)(c)}{(K_{R,d,u}p)(c)} = \frac{r_R(u)}{p_R(u)} \frac{[\mathcal{T}(r)]^d(c \mid u)}{g_*^d(c \mid u)}. \tag{46}$$

The first factor on the right-hand side is constant in c and therefore disappears under $\operatorname{osc}_c \log$. By Lemma B.2,

$$\min_x \frac{r(x)}{p(x)} \le \frac{(K_{R,d,u}r)(c)}{(K_{R,d,u}p)(c)} \le \max_x \frac{r(x)}{p(x)}.$$

Taking logarithms and then the oscillation over c, and plugging equation (46) (where the factor $r_R(u)/p_R(u)$ can be simplified because it does not depend on c), yields

$$\operatorname{osc}_c \log \frac{[\mathcal{T}(r)]^d(c \mid u)}{g_*^d(c \mid u)} \le \operatorname{osc}_x \log \frac{r(x)}{p(x)}.$$

Since this holds for every context (R, d, u),

$$\max_{R,d,x_R} \operatorname{osc}_c \log \frac{[\mathcal{T}(r)]^d(c \mid x_R)}{g_*^d(c \mid x_R)} \le \operatorname{osc}_x \log \frac{r(x)}{p(x)}. \tag{47}$$

This inequality, which we will leverage in the next subsection, shows that for any context, the oscillation of the log-ratio to the optimal categorical predictor is bounded by the log-ratio. □

B.3 Fixed-point exactness and global contraction for arbitrary reveal orders: proof of Theorem B.1*(i)–(ii)*

Fixed point and full importance weighting. For every reveal order $\sigma = (\sigma_1, \ldots, \sigma_D)$, the target heads satisfy the chain-rule identity

$$q_{g_*}(x \mid \sigma) = \prod_{k=1}^{D} g_*^{\sigma_k}\left(x^{\sigma_k} \mid x_{\sigma_{<k}}\right) = \prod_{k=1}^{D} p\left(X^{\sigma_k} = x^{\sigma_k} \mid X_{\sigma_{<k}} = x_{\sigma_{<k}}\right) = p(x). \tag{48}$$

Consequently, substituting $g = g_*$ into equation (37) gives $r_{\nu,g_*}(x, \sigma) = \rho(\sigma)p(x)$ for every $\nu \in [0, 1]$. Its endpoint marginal is therefore p, and the Gibbs-label identity gives $\Phi_\nu(g_*) = \mathcal{T}(p) = g_*$. Thus g_* is a fixed point for every tempering exponent.

At $\nu = 1$, the proposal contribution in equation (37) disappears: $P_{1,g}(x, \sigma) = \rho(\sigma)p(x)$ for every positive g. Hence

$$\Phi_1(g) = \mathcal{T}(p) = g_*,$$

which proves one-step recovery under full importance weighting.

Accumulation of head-wise error. Now fix an arbitrary positive predictor g and write $\Delta = \Delta(g)$, where $\Delta(\cdot)$ is defined in (37). For every training-supported context (R, d, x_R),

$$\sum_c g_*^d(c \mid x_R) \frac{g^d(c \mid x_R)}{g_*^d(c \mid x_R)} = 1. \tag{49}$$

Therefore,

$$\min_c \frac{g^d(c \mid x_R)}{g_*^d(c \mid x_R)} \le 1 \le \max_c \frac{g^d(c \mid x_R)}{g_*^d(c \mid x_R)}. \tag{50}$$

If we write $\ell(c) = \log\big(g^d(c \mid x_R)/g_*^d(c \mid x_R)\big)$, the definition of $\Delta(\cdot)$ and equation (50) imply, respectively, that

$$\max_c \ell(c) - \min_c \ell(c) \le \Delta, \qquad \min_c \ell(c) \le 0 \le \max_c \ell(c). \tag{51}$$

Combining these inequalities, we obtain that for all c,

$$\left|\log \frac{g^d(c \mid x_R)}{g_*^d(c \mid x_R)}\right| \le \Delta. \tag{52}$$

For a reveal order σ, define the joint proposal error

$$a_\sigma(x) = \log \frac{q_g(x \mid \sigma)}{p(x)} = \sum_{k=1}^{D} \log \frac{g^{\sigma_k}\left(x^{\sigma_k} \mid x_{\sigma_{<k}}\right)}{g_*^{\sigma_k}\left(x^{\sigma_k} \mid x_{\sigma_{<k}}\right)}. \tag{53}$$

Every factor queried by the reveal policy is training-supported, so each summand has absolute value at most Δ. Hence

$$-D\Delta \le a_\sigma(x) \le D\Delta \qquad \text{for every } (x, \sigma). \tag{54}$$

Tempering and endpoint error. Using the definition of a_σ, the tempered endpoint ratio can be written as

$$\frac{r_{\nu,g}(x)}{p(x)} = C_{\nu,g}^{-1} \sum_\sigma \rho(\sigma) \exp\{(1-\nu) a_\sigma(x)\}. \tag{55}$$

By Equation (54),

$$e^{-D(1-\nu)\Delta} \le \sum_\sigma \rho(\sigma) e^{(1-\nu)a_\sigma(x)} \le e^{D(1-\nu)\Delta}. \tag{56}$$

Thus, plugging (55) into (56) and taking logarithms and the oscillation yields

$$\operatorname{osc}_x \log \frac{r_{\nu,g}(x)}{p(x)} \le 2D(1-\nu)\Delta(g). \tag{57}$$

Note that in this equation, the normalizing constant $C_{\nu,g}^{-1}$ is independent of x and therefore disappears under log-ratio oscillation.

Since $\Phi_\nu(g) = \mathcal{T}(r_{\nu,g})$ by the definitions of Φ_ν and $\mathcal{T}$, Theorem B.3 and Equation (57) give

$$\Delta(\Phi_\nu(g)) \le \operatorname{osc}_x \log \frac{r_{\nu,g}(x)}{p(x)} \le 2D(1-\nu)\Delta(g), \tag{58}$$

which proves Equation (41).

Iteration and convergence. Let $c_\nu = 2D(1-\nu)$. If $c_\nu < 1$, equivalently $\nu > 1 - \frac{1}{2D}$, then repeated application of the contraction gives

$$\Delta(g_k) \le c_\nu^k \Delta(g_0) \longrightarrow 0.$$

Because the heads are normalized categorical distributions, vanishing log-ratio oscillation implies

$$g_k^d(\cdot \mid x_R) \longrightarrow g_*^d(\cdot \mid x_R)$$

at every training-supported context. Consequently, $q_{g_k}(\cdot \mid \sigma) \to p$ for every order in the support of ρ, and the corresponding marginal proposal also converges to p. The contraction also establishes

uniqueness in this regime. Indeed, if $g = \Phi_\nu(g)$, then $\Delta(g) \le c_\nu \Delta(g)$, and $c_\nu < 1$ implies $\Delta(g) = 0$, hence $g = g_*$ on all training-supported contexts.

For varying exponents ν_k, define $c_k = 2D(1 - \nu_k)$. The same argument gives $\Delta(g_k) \le \big(\prod_{j=0}^{k-1} c_j\big)\Delta(g_0)$. It is therefore sufficient that $\prod_{j=0}^{k-1} c_j \longrightarrow 0$. In particular, convergence holds if $\sup_k c_k < 1$.

The factor $2D$ is a worst-case bound obtained by adding the maximum possible error across all D reveals. It is a sufficient theoretical bound, not a necessary condition, and can be conservative in high dimensions. It should therefore not be interpreted as prescribing the empirically optimal tempering exponent.

B.4 Global convergence for a fixed reveal order: proof of Theorem B.1*(iii)*

Relabel the coordinates so that the fixed reveal order is $1, \dots, D$. At outer iteration k, write the corresponding order-conditional proposal as

$$q_k(x) = \prod_{d=1}^{D} q_k^d(x^d \mid x^{<d}),$$

where q_k^d denotes the current categorical head evaluated at the prefix $x^{<d} = (x^1, \dots, x^{d-1})$. For a tempering exponent ν_k, the effective endpoint distribution is

$$r_k(x) = \frac{q_k(x)^{1-\nu_k} p(x)^{\nu_k}}{\sum_{x'} q_k(x')^{1-\nu_k} p(x')^{\nu_k}}.$$

Population update at a prefix. Fix a coordinate d and a prefix $u \in \mathcal{A}^{d-1}$. Write $y = X^{>d}$ for the suffix following site d. Exact Gibbs-label fitting gives

$$q_{k+1}^d(c \mid u) = \sum_y r_k(X^{>d} = y \mid X^{<d} = u) h^d(c \mid u, y). \tag{59}$$

The value of X^d is absent from the averaging distribution because the Gibbs label $h^d(c \mid X^{-d})$ depends only on $X^{-d} = (u, y)$.

Under the target distribution, the same Gibbs-label identity gives

$$p(X^d = c \mid X^{<d} = u) = \sum_y p(X^{>d} = y \mid X^{<d} = u) h^d(c \mid u, y). \tag{60}$$

Thus the updated-head error is controlled by the discrepancy between the suffix laws of r_k and p.

Head-wise error. Define

$$e_k^d = \max_u \operatorname{osc}_c \log \frac{q_k^d(c \mid u)}{p(X^d = c \mid X^{<d} = u)}.$$

We claim that

$$e_{k+1}^D = 0, \qquad e_{k+1}^d \le (1 - \nu_k) \left(e_k^d + 2 \sum_{j>d} e_k^j \right), \quad d < D. \tag{61}$$

For $d = D$, the suffix is empty. Hence equation (59) reduces directly to

$$q_{k+1}^D(c \mid u) = h^D(c \mid u) = p(X^D = c \mid X^{<D} = u),$$

which proves $e_{k+1}^D = 0$.

Now fix $d < D$ and a prefix u. For compactness, let

$$\mu_k(c) = q_k^d(c \mid u), \qquad \mu_*(c) = p(X^d = c \mid X^{<d} = u),$$

and define

$$v_k(c, y) = \frac{\mu_k(c) q_k(y \mid u, c)}{\mu_*(c) p(y \mid u, c)}.$$

Conditioning the softened distribution r_k on the prefix u cancels all proposal and target factors preceding site d. Therefore,

$$\frac{r_k(c, y \mid u)}{p(c, y \mid u)} \propto v_k(c, y)^{1-\nu_k}. \tag{62}$$

Marginalizing c is a positive linear operation. Applying Lemma B.2 to Equation (62) gives

$$\min_c \frac{r_k(c, y \mid u)}{p(c, y \mid u)} \le \frac{r_k(y \mid u)}{p(y \mid u)} \le \max_c \frac{r_k(c, y \mid u)}{p(c, y \mid u)}, \tag{63}$$

$$\implies \operatorname{osc}_y \log \frac{r_k(y \mid u)}{p(y \mid u)} \le (1 - \nu_k) \operatorname{osc}_{c,y} \log v_k(c, y). \tag{64}$$

Likewise, equations (59) and (60) apply the same positive Gibbs-label kernel to the two suffix distributions. Another application of Lemma B.2 therefore gives

$$\min_y \frac{r_k(y \mid u)}{p(y \mid u)} \le \frac{q^d_{k+1}(c \mid u)}{p(X^d = c \mid X^{<d} = u)} \le \max_y \frac{r_k(y \mid u)}{p(y \mid u)}, \tag{65}$$

$$\implies \operatorname{osc}_c \log \frac{q^d_{k+1}(c \mid u)}{p(X^d = c \mid X^{<d} = u)} \le \operatorname{osc}_y \log \frac{r_k(y \mid u)}{p(y \mid u)}. \tag{66}$$

It remains to control the right-hand side. The autoregressive factorizations of q_k and p give

$$\log v_k(c, y) = \log \frac{\mu_k(c)}{\mu_*(c)} + \sum_{j>d} \log \frac{q^j_k\left(y^j \mid u, c, y^{d+1:j-1}\right)}{p\left(X^j = y^j \mid X^{<j} = (u, c, y^{d+1:j-1})\right)}. \tag{67}$$

The first term in (67) has oscillation at most e^d_k by definition. For every later head $j > d$ and every fixed prefix, let

$$\ell_{y^{d+1:j-1}}(y^j) = \log \frac{q^j_k\left(y^j \mid u, c, y^{d+1:j-1}\right)}{p\left(X^j = y^j \mid X^{<j} = (u, c, y^{d+1:j-1})\right)} \tag{68}$$

We can mirror the normalization argument of equations (49)-(52) to conclude that $|\ell_{y^{d+1:j-1}}(c)| \le \operatorname{osc}_c \ell_{y^{d+1:j-1}}(c) \le e^j_k$, since e^j_k is a uniform bound of such oscillations over all prefixes. Hence, we can further bound the oscillation of $\ell_{y^{d+1:j-1}}(c)$ with respect to both c and $y^{d+1:j-1}$:

$$\operatorname{osc}_{c, y^{d+1:j-1}} \ell_{y^{d+1:j-1}}(c) \le 2|\ell_{y^{d+1:j-1}}(c)| \le 2e^j_k. \tag{69}$$

By subadditivity of oscillation, we conclude that

$$\operatorname{osc}_{c,y} \log v_k(c, y) \le e^d_k + 2 \sum_{j>d} e^j_k.$$

Combining this bound with (64), (66) and maximizing over u proves Equation (61).

Triangular iteration. Let $e_k = (e^1_k, \ldots, e^D_k)^\top$ and define the upper-triangular matrix $H \in \mathbb{R}^{D \times D}$ by

$$H_{dj} = \begin{cases} 1, & j = d, \\ 2, & j > d, \\ 0, & j < d. \end{cases}$$

Then equation (61) implies, componentwise, $e_{k+1} \le (1 - \nu_k) H e_k$. Iterating gives

$$e_k \le \left\{ \prod_{i=0}^{k-1} (1 - \nu_i) \right\} H^k e_0. \tag{70}$$

Write $H = I + N$, where N is strictly upper triangular. Since $N^D = 0$, $H^k = (I+N)^k = \sum_{m=0}^{D-1} \binom{k}{m} N^m$. Thus $\|H^k\| = O(k^{D-1})$ in any matrix norm.

For a fixed exponent $\nu \in (0,1)$, Equation (70) therefore gives

$$\max_d e_k^d = O\big(k^{D-1}(1-\nu)^k\big)\,.$$

The exponential factor dominates the polynomial factor, so $e_k^d \to 0$ for every site d.

Because both $q_k^d(\cdot \mid u)$ and $p(X^d = \cdot \mid X^{<d} = u)$ are normalized categorical distributions, vanishing log-ratio oscillation implies $q_k^d(\cdot \mid u) \longrightarrow p(X^d = \cdot \mid X^{<d} = u)$ at every prefix. Consequently,

$$q_k(x) = \prod_{d=1}^{D} q_k^d(x^d \mid x^{<d}) \longrightarrow p(x),$$

which proves global convergence for every positive initialization.

More generally, if the exponents vary but satisfy $\nu_k \geq \nu_{\min} > 0$, then $\prod_{i=0}^{k-1}(1-\nu_i) \leq (1-\nu_{\min})^k$, and the same argument gives $\max_d e_k^d = O\big(k^{D-1}(1-\nu_{\min})^k\big)$.

B.5 Global convergence of label-only fixed-order fitting: proof of Theorem B.1*(iv)*

The fixed-order proof in Section B.4 uses the exponential factor $(1-\nu)^k$ and therefore does not cover the endpoint $\nu = 0$. Nevertheless, exact label-only fitting also converges globally under the same strict-positivity assumptions. The argument uses contraction of the Gibbs transition kernels rather than tempered importance correction.

Proposition B.4 (Fixed-order convergence without importance weighting)**.** *Consider one fixed reveal order and exact population fitting with $\nu = 0$. If p is strictly positive on the finite product space, then the fixed-order heads converge globally to the target heads.*

Proof. Relabel the fixed order as $1, \ldots, D$. Strict positivity of p on the finite state space implies that the Gibbs labels are uniformly positive:

$$\epsilon = \min_{d, x^{-d}, c} h^d(c \mid x^{-d}) > 0.$$

Fix a site d, a prefix $u = x^{<d}$, and a possible value $a \in \mathcal{A}$ at site d. Define

$$K_{d,k}(a, c \mid u) = \sum_y q_k(X^{>d} = y \mid X^{<d} = u, X^d = a) h^d(c \mid u, y). \tag{71}$$

This kernel first samples the suffix from the current proposal, conditional on the prefix and current symbol a, and then resamples site d from its target Gibbs conditional.

When $\nu = 0$, the buffer distribution is q_k. Marginalizing the current value a in the population update gives

$$q_{k+1}^d(c \mid u) = \sum_{a \in \mathcal{A}} q_k^d(a \mid u) K_{d,k}(a, c \mid u). \tag{72}$$

Thus the current head is updated by applying the Markov kernel $K_{d,k}$.

Every entry of $K_{d,k}$ is at least ϵ, because it is an average of Gibbs labels: $K_{d,k}(a, c \mid u) \geq \epsilon$. Its total-variation contraction coefficient is therefore uniformly bounded by $\kappa = 1 - N\epsilon < 1$, where $N = |\mathcal{A}|$. In particular, for any categorical distributions μ and $\widetilde{\mu}$,

$$\|\mu K_{d,k} - \widetilde{\mu} K_{d,k}\|_{\mathrm{TV}} \leq \kappa \, \|\mu - \widetilde{\mu}\|_{\mathrm{TV}}\,. \tag{73}$$

We now proceed backward through the reveal order. For $d = D$, the suffix is empty, so

$$q_{k+1}^D(c \mid u) = h^D(c \mid u) = p(X^D = c \mid X^{<D} = u).$$

Hence the last head becomes exact after one iteration.

Suppose inductively that all later heads $j > d$ converge to their target conditionals. Their product then satisfies

$$q_k(X^{>d} = y \mid u, a) \longrightarrow p(X^{>d} = y \mid u, a)$$

uniformly over the finitely many (u, a, y). Consequently, $K_{d,k}$ converges to the target two-block Gibbs kernel

$$K_d^*(a, c \mid u) = \sum_y p(X^{>d} = y \mid u, a) h^d(c \mid u, y).$$

The target head $\pi_d(c \mid u) := p(X^d = c \mid X^{<d} = u)$ is stationary for this limiting kernel. Indeed,

$$\sum_a \pi_d(a \mid u) K_d^*(a, c \mid u) = \sum_{a,y} p(a, y \mid u) h^d(c \mid u, y) = \sum_y p(y \mid u) p(c \mid u, y) = \pi_d(c \mid u). \tag{74}$$

Define the maximum head error

$$A_k^d = \max_u \left\| q_k^d(\cdot \mid u) - \pi_d(\cdot \mid u) \right\|_{\mathrm{TV}}.$$

Using Equations (72) and (73) and adding and subtracting $\pi_d(\cdot \mid u) K_{d,k}$ gives

$$A_{k+1}^d \le \kappa A_k^d + B_k^d, \tag{75}$$

where

$$B_k^d = \max_u \left\| \pi_d(\cdot \mid u) K_{d,k}(\cdot, \cdot \mid u) - \pi_d(\cdot \mid u) K_d^*(\cdot, \cdot \mid u) \right\|_{\mathrm{TV}}.$$

The induction hypothesis implies $B_k^d \to 0$. Since $\kappa < 1$, unrolling Equation (75) gives $A_k^d \to 0$. Backward induction over $d = D, D-1, \ldots, 1$ proves convergence of every head and hence of the fixed-order joint proposal to p. □

This result shows that tempered importance weighting is not the only source of contraction. Even at $\nu = 0$, repeated Gibbs-label fitting propagates the correct target conditionals backward through a fixed reveal order. Positive tempering strengthens this mechanism by adding the explicit factor $1 - \nu$ in Equation (61).

B.6 ESS ADAPTATION PRESERVES FIXED-ORDER CONVERGENCE

The fixed-order result remains valid when the tempering exponent is selected adaptively by the empirical ESS rule, provided that each selected exponent is followed by the corresponding exact population fit. The key point is that the selected exponents remain uniformly bounded away from zero.

Proposition B.5 (Uniform positivity of ESS-selected exponents)**.** *Consider the fixed-order population iteration, and suppose that at iteration k the exponent ν_k is selected from a finite batch by Equation (13). Then, after the first population update, there exists a constant $\nu_{\min} > 0$, independent of k and of the sampled batch, such that*

$$\nu_k \ge \nu_{\min}.$$

Consequently, if each update is the exact population update $g_{k+1} = \Phi_{\nu_k}(g_k)$, then g_k converges to g_*.

Proof. Because p is strictly positive on a finite product space, every single-site Gibbs label is uniformly positive. Define

$$\epsilon = \min_{d, x^{-d}, c} h^d(c \mid x^{-d}) > 0.$$

Every fitted categorical head is a conditional average of Gibbs labels. Hence, after the first population update, $g_k^d(c \mid x^{<d}) \ge \epsilon$ for every site, prefix, and symbol. The corresponding fixed-order proposal therefore satisfies

$$q_k(x) = \prod_{d=1}^D g_k^d(x^d \mid x^{<d}) \ge \epsilon^D, \qquad k \ge 1. \tag{76}$$

The unnormalized importance weight is

$$W_k(x) = \frac{e^{-U(x)}}{q_k(x)},$$

so its log is

$$\lambda_k(x) = -U(x) - \log q_k(x).$$

Since $q_k(x) \le 1$ and Equation (76) gives $q_k(x) \ge \epsilon^D$,

$$0 \le -\log q_k(x) \le D \log(1/\epsilon).$$

It follows that the full log-weight range is uniformly bounded:

$$L_k := \operatorname{osc}_x \lambda_k(x) \le U_{\max} - U_{\min} + D \log(1/\epsilon) =: L_{\max} < \infty. \tag{77}$$

For any finite batch drawn at iteration k, its empirical log-weight range is at most L_k, and therefore at most $L_{\max}$. By Lemma B.6, the exponent selected by the ESS rule satisfies

$$\nu_k \ge \min\left\{1, \frac{\log(1/\xi)}{2L_{\max}}\right\} =: \nu_{\min} > 0,$$

with the convention $\nu_{\min} = 1$ if $L_{\max} = 0$.

Thus the adaptively selected exponents satisfy $\inf_k \nu_k \ge \nu_{\min} > 0$. The fixed-order matrix bound Equation (70) consequently gives

$$\max_d e_k^d = O\big(k^{D-1}(1-\nu_{\min})^k\big) \longrightarrow 0.$$

Hence all fixed-order heads, and therefore the joint proposal q_k, converge to the target. □

The preceding result only requires the selected exponents to remain positive. In fact, the ESS rule eventually selects full weighting in the exact population setting. Since $q_k \to p$ on the finite state space,

$$\operatorname{osc}_x \log \frac{e^{-U(x)}}{q_k(x)} = \operatorname{osc}_x \log \frac{p(x)}{q_k(x)} \longrightarrow 0.$$

By Equation (78), once this range is small enough that

$$B \exp(-2L_k) \ge \xi B,$$

every possible batch satisfies the ESS threshold at $\nu = 1$. The rule then selects $\nu_k = 1$, and the next exact population update recovers g_* in one step.

This conclusion concerns an idealized hybrid procedure: a finite batch is used only to select ν_k, while $\Phi_{\nu_k}(g_k)$ is still evaluated as an exact population fit. If the same finite batch is also used to approximate the fitting expectation, additional sampling and optimization errors arise and are not covered by this proposition.

Lemma B.6 (Lower bound for ESS-selected tempering)**.** *Let $W_i = e^{\lambda_i} > 0$, $i = 1, \dots, B$, be a finite collection of importance weights, and define their log-weight range by $L = \max_i \lambda_i - \min_i \lambda_i$. For $\nu \in [0,1]$, let $\mathrm{ESS}(\nu) = \big(\sum_{i=1}^B e^{\nu\lambda_i}\big)^2 / \sum_{i=1}^B e^{2\nu\lambda_i}$. Then*

$$\mathrm{ESS}(\nu) \ge B e^{-2\nu L}. \tag{78}$$

Consequently, for any threshold $\xi \in (0,1)$, the largest feasible tempering exponent $\nu^\star = \max\{\nu \in [0,1] : \mathrm{ESS}(\nu) \ge \xi B\}$ satisfies

$$\nu^\star \ge \min\left\{1, \frac{\log(1/\xi)}{2L}\right\} \tag{79}$$

when $L > 0$. If $L = 0$, then $\mathrm{ESS}(\nu) = B$ for every $\nu \in [0,1]$ and hence $\nu^\star = 1$.

Proof. Let $\lambda_{\min} = \min_i \lambda_i$, $\lambda_{\max} = \max_i \lambda_i$. The numerator of $\mathrm{ESS}(\nu)$ satisfies $\left(\sum_{i=1}^B e^{\nu\lambda_i}\right)^2 \geq B^2 e^{2\nu\lambda_{\min}}$, while its denominator satisfies $\sum_{i=1}^B e^{2\nu\lambda_i} \leq B e^{2\nu\lambda_{\max}}$. Therefore,

$$\mathrm{ESS}(\nu) \geq \frac{B^2 e^{2\nu\lambda_{\min}}}{B e^{2\nu\lambda_{\max}}} = B e^{-2\nu(\lambda_{\max}-\lambda_{\min})} = B e^{-2\nu L},$$

which proves Equation (78).

Suppose $L > 0$ and define $\bar{\nu} = \min\left\{1, \frac{\log(1/\xi)}{2L}\right\}$. By construction, $e^{-2\bar{\nu}L} \geq \xi$, and hence Equation (78) gives $\mathrm{ESS}(\bar{\nu}) \geq \xi B$. Thus $\bar{\nu}$ is feasible. Since $\nu^\star$ is the largest feasible exponent, $\nu^\star \geq \bar{\nu}$, which proves Equation (79).

If $L = 0$, all log-weights, and therefore all weights, are equal. It follows directly that $\mathrm{ESS}(\nu) = B$ for every $\nu \in [0, 1]$, so the largest feasible exponent is $\nu^\star = 1$. □

B.7 Tempering scales the local Jacobian

The global contraction condition in Equation (41) is sufficient but can be conservative. We now study the linearized population update near the target. This local result applies to any fixed, value-independent reveal-order law ρ and is independent of the fixed-order triangular argument.

Proposition B.7 (Local Jacobian scaling)**.** *At the target fixed point,*

$$D\Phi_\nu(g_*) = (1-\nu)D\Phi_0(g_*), \qquad \nu \in [0, 1]. \tag{80}$$

Consequently, every eigenvalue of $D\Phi_\nu(g_*)$ is equal to $1-\nu$ times the corresponding eigenvalue of $D\Phi_0(g_*)$.

Proof. Let $\mathcal{R}_\nu(g) = r_{\nu,g}$ denote the map from the current categorical heads to the tempered endpoint distribution. The population update decomposes as $\Phi_\nu = \mathcal{T} \circ \mathcal{R}_\nu$, where $\mathcal{T}$ is the exact Gibbs-label fitting operator from Theorem B.3.

Choose independent coordinates for the categorical simplices and let v be a tangent perturbation of g_*. Set $g_\epsilon = g_* + \epsilon v$ for sufficiently small ϵ. For each reveal order σ, define $\dot{q}_\sigma(x) = \frac{\mathrm{d}}{\mathrm{d}\epsilon} q_{g_\epsilon}(x \mid \sigma)\big|_{\epsilon=0}$. Because $q_{g_\epsilon}(\cdot \mid \sigma)$ is normalized for every ϵ,

$$\sum_x \dot{q}_\sigma(x) = 0. \tag{81}$$

Consider the unnormalized endpoint law

$$\widetilde{r}_{\nu,g}(x) = p(x)^\nu \sum_\sigma \rho(\sigma) q_g(x \mid \sigma)^{1-\nu}.$$

At $g = g_*$, the target chain rule gives $q_{g_*}(x \mid \sigma) = p(x)$ for every σ, and hence $\widetilde{r}_{\nu,g_*}(x) = p(x)$.

Differentiating in the direction v gives

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\widetilde{r}_{\nu,g_\epsilon}(x)\bigg|_{\epsilon=0} = p(x)^\nu \sum_\sigma \rho(\sigma)(1-\nu)p(x)^{-\nu}\dot{q}_\sigma(x) = (1-\nu)\sum_\sigma \rho(\sigma)\dot{q}_\sigma(x). \tag{82}$$

The derivative of the normalizing constant vanishes, since

$$\sum_x \frac{\mathrm{d}}{\mathrm{d}\epsilon}\widetilde{r}_{\nu,g_\epsilon}(x)\bigg|_{\epsilon=0} = (1-\nu)\sum_\sigma \rho(\sigma)\sum_x \dot{q}_\sigma(x) = 0 \tag{83}$$

by equation (81). Therefore normalization does not contribute an additional first-order term at g_*, and

$$D\mathcal{R}_\nu(g_*)[v] = (1-\nu)\sum_\sigma \rho(\sigma)\dot{q}_\sigma.$$

For $\nu = 0$, the same calculation gives $D\mathcal{R}_0(g_*)[v] = \sum_\sigma \rho(\sigma)\dot{q}_\sigma$. Thus

$$D\mathcal{R}_\nu(g_*) = (1-\nu)D\mathcal{R}_0(g_*). \tag{84}$$

On the interior of the finite probability simplex, the fitting map $\mathcal{T}$ is differentiable. Moreover, $\mathcal{R}_\nu(g_*) = p$ for every ν. Applying the chain rule and equation (84) yields

$$D\Phi_\nu(g_*) = D\mathcal{T}(p)\, D\mathcal{R}_\nu(g_*) = (1-\nu)D\mathcal{T}(p)\, D\mathcal{R}_0(g_*) = (1-\nu)D\Phi_0(g_*), \tag{85}$$

which proves equation (80). □

Let $s_0 = \mathrm{spr}(D\Phi_0(g_*))$ be the spectral radius of the unweighted population update. By Proposition B.7, $\mathrm{spr}(D\Phi_\nu(g_*)) = (1-\nu)s_0$. Therefore,

$$(1-\nu)s_0 < 1 \tag{86}$$

is sufficient for local asymptotic stability of g_*. In other words, under this condition, every sufficiently small perturbation of the target heads converges back to g_* under repeated population updates.

This local condition can be substantially weaker than the global sufficient condition $2D(1-\nu) < 1$ from equation (41). The global result controls every positive initialization using a worst-case accumulation of error across all D coordinates, whereas equation (86) describes only the dynamics of small perturbations around g_*.

At $\nu = 1$, the Jacobian vanishes. This is consistent with the stronger global fact that full importance weighting maps every positive initialization to g_* in a single exact population update. Thus the fixed-point identity establishes that the target is correct, while Equation (80) quantifies how tempering improves its local stability.

B.8 Robustness to approximate population fitting

The preceding convergence results assume unrestricted categorical heads and exact population minimization. In the arbitrary-order contraction regime, bounded fitting errors can be incorporated through a standard perturbation argument.

Let $\widehat{\Phi}_k(g)$ denote the model produced by the implemented fit at iteration k, and suppose its error relative to the exact population update satisfies

$$\delta_k = \max_{(R,d,x_R)\in\mathcal{C}_{\mathrm{train}}} \mathrm{osc}_c \log \frac{[\widehat{\Phi}_k(g_k)]^d(c \mid x_R)}{[\Phi_\nu(g_k)]^d(c \mid x_R)}. \tag{87}$$

If $g_{k+1} = \widehat{\Phi}_k(g_k)$, then subadditivity of log-ratio oscillation and Equation (41) give

$$\Delta(g_{k+1}) \le c_\nu \Delta(g_k) + \delta_k, \qquad c_\nu = 2D(1-\nu). \tag{88}$$

Assume $c_\nu < 1$. Iterating Equation (88) yields

$$\Delta(g_k) \le c_\nu^k \Delta(g_0) + \sum_{j=0}^{k-1} c_\nu^{k-1-j}\delta_j. \tag{89}$$

Consequently:

- If $\delta_k \to 0$ for every k, then $\Delta(g_k) \to 0$.
- If $\delta_k \le \overline{\delta}$ for every k, then

$$\Delta(g_k) \le c_\nu^k \Delta(g_0) + \frac{1-c_\nu^k}{1-c_\nu}\,\overline{\delta},$$

and hence

$$\limsup_{k\to\infty} \Delta(g_k) \le \frac{\overline{\delta}}{1-c_\nu}.$$

Thus vanishing fitting errors preserve convergence, while persistent bounded errors yield convergence only to a controlled neighborhood of the target. This is a conditional robustness statement: it does not establish that any particular neural architecture, finite replay buffer, or optimizer achieves a prescribed sequence δ_k.

Both this argument and Theorem B.4 rely on strict positivity. Hard constraints can produce zero Gibbs labels or inaccessible contexts, in which case the uniform kernel contraction and log-ratio metrics used above may fail and require a separate analysis.

C General discrete interpolants

We recap and prove the construction of general forward and backward CTMC that we introduced in Section 3.3. We will prove the construction with $p_t^d(\cdot \mid x^{-d})$ and $\phi_t^d(\cdot \mid x^{-d})$ in Equations (22) and (25) minimizes the kinetic energy in Section C.1, and derive the training objective Section C.2.

C.1 MINIMUM KINETIC ENERGY LOCAL FLUX

We aim to find a flux to minimize the following kinetic energy

$$\mathcal{E}_t(j) = \frac{1}{2}\sum_d \sum_{x^{-d}} p_t(x^{-d}) \sum_a \sum_{c\neq a} \frac{j_t^d(a, c \mid x^{-d})^2}{p_t(x^d = a|x^{-d})p_t(x^d = c|x^{-d})} \tag{90}$$

subject to the Kolmogorov constraint:

$$\partial_t^{(d)} p_t(x^d = a \mid x^{-d}) = \sum_{c\neq a} j_t^d(c, a|x^{-d}), \quad \forall a, d \tag{91}$$

Proof. We solve the constrained minimization problem by introducing Lagrange multipliers:

$$\begin{aligned}\mathcal{L}(j, \lambda) = \sum_d \sum_{x^{-d}} p_t(x^{-d}) \sum_a \Bigg[&\frac{1}{2} \sum_{c\neq a} \frac{j_t^d(a, c \mid x^{-d})^2}{p_t(x^d = a \mid x^{-d})\, p_t(x^d = c \mid x^{-d})} \\ &+ \lambda_t^d(a \mid x^{-d}) \left(\partial_t^{(d)} p_t(x^d = a \mid x^{-d}) - \sum_{c\neq a} j_t^d(c, a \mid x^{-d}) \right) \Bigg].\end{aligned} \tag{92}$$

Therefore, taking the variation, we have

$$\begin{aligned}&\frac{\delta \mathcal{L}}{\delta j_t^d(a, c \mid x^{-d})} \\ =&p_t(x^{-d}) \left[\frac{j_t^d(a, c \mid x^{-d})}{p_t(x^d = a \mid x^{-d})\, p_t(x^d = c \mid x^{-d})} + \lambda_t^d(a \mid x^{-d}) - \lambda_t^d(c \mid x^{-d}) \right] = 0.\end{aligned} \tag{93}$$

Hence

$$j_t^d(a, c \mid x^{-d}) = p_t(x^d = a \mid x^{-d})\, p_t(x^d = c \mid x^{-d}) \left[\lambda_t^d(c \mid x^{-d}) - \lambda_t^d(a \mid x^{-d}) \right]. \tag{94}$$

Now, plugging this into the Kolmogorov constraint, we obtain

$$\begin{aligned}\partial_t^{(d)} p_t(x^d = a \mid x^{-d}) &= p_t(x^d = a \mid x^{-d}) \sum_{c\neq a} p_t(x^d = c \mid x^{-d}) \left[\lambda_t^d(a \mid x^{-d}) - \lambda_t^d(c \mid x^{-d}) \right] \\ &= p_t(x^d = a \mid x^{-d}) \left[\lambda_t^d(a \mid x^{-d}) - \sum_c p_t(x^d = c \mid x^{-d}) \lambda_t^d(c \mid x^{-d}) \right].\end{aligned} \tag{95}$$

Dividing both sides by $p_t(x^d = a \mid x^{-d})$ gives

$$\phi_t^d(a \mid x^{-d}) = \lambda_t^d(a \mid x^{-d}) - \sum_c p_t(x^d = c \mid x^{-d}) \lambda_t^d(c \mid x^{-d}). \tag{96}$$

Since the second term is a constant w.r.t a, taking the difference between states a and c yields

$$\lambda_t^d(a \mid x^{-d}) - \lambda_t^d(c \mid x^{-d}) = \phi_t^d(a \mid x^{-d}) - \phi_t^d(c \mid x^{-d}). \tag{97}$$

Therefore, the minimizing flux is

$$\begin{aligned}J_t^d(c, a \mid x^{-d}) &= p_t(x^d = a \mid x^{-d}) p_t(x^d = c \mid x^{-d}) \left[\phi_t^d(a \mid x^{-d}) - \phi_t^d(c \mid x^{-d}) \right] \\ &= p_t(x^d = c \mid x^{-d}) \partial_t^{(d)} p_t(x^d = a \mid x^{-d}) - p_t(x^d = a \mid x^{-d}) \partial_t^{(d)} p_t(x^d = c \mid x^{-d}).\end{aligned} \tag{98}$$

□

C.2 LEARNING THE DISCRETE INTERPOLANTS

We now derive the training objective. We also detail the form after using tempered importance reweighting.

C.2.1 LEARNING $p_t^d(\cdot \mid x^{-d})$

The entire derivation follows in three steps:

1. we want to use Gibbs label as efficient energy supervision, and hence we write down $p_t^d(\cdot \mid x^{-d})$ with both ends explicitly $p_t^d(\cdot \mid x^{-d}) = \mathbb{E}_{x_0,x_1|x^{-d}}[\cdots] = \mathbb{E}_{x_0,x_1^{-d},x_1^d|x^{-d}}[\cdots]$, the second equality allows us to bake in the endpoints' Gibbs label into the training;
2. we apply CE between the optimal $p_t^d(\cdot \mid x^{-d})$ and our network, writing down a CE objective;
3. we take expectation over end point distributions, and swapping the conditions: $\mathbb{E}_{x^{-d}}\mathbb{E}_{x_0,x_1|x^{-d}}[\cdots] = \mathbb{E}_{x_0,x_1}\mathbb{E}_{x^{-d}|x_0,x_1}[\cdots]$, allowing us to directly train the model without simulation or sampling from the endpoint posteriors.

Step 1 Under the independent coupling and the coordinate-wise interpolant in Equation (16), we have

$$p_t(x^d = c|x^{-d}) = \frac{p_t(x^d = c, x^{-d})}{p_t(x^{-d})} \tag{99}$$

$$= \frac{1}{p_t(x^{-d})} \sum_{x_0,x_1} p_0(x_0)p_1(x_1)\kappa_t^d(x^d = c \mid x_0^d, x_1^d) \prod_{j\neq d} \kappa_t^j(x^j \mid x_0^j, x_1^j) \tag{100}$$

We can also factorize both endpoints into Gibbs label:

$$p_0(x_0) = p_0^d(x_0^d|x_0^{-d})p_0(x_0^{-d}); \quad p_1(x_1) = p_1^d(x_1^d|x_1^{-d})p_1(x_1^{-d}) \tag{101}$$

We can achieve three different variations from these factorization: (1) factorize both endpoints; (2) factorize p_0; and (3) factorize p_1. In our experiments, we consider having samples from one endpoints, and hence we only factorize p_1. However, we can also derive other variations.

The factorization over p_1 gives us:

$$p_t(x^d = c|x^{-d}) = \frac{1}{p_t(x^{-d})} \sum_{x_0,x_1} p_0(x_0)p_1^d(x_1^d|x_1^{-d})p_1(x_1^{-d})\kappa_t^d(x^d = c \mid x_0^d, x_1^d) \prod_{j\neq d} \kappa_t^j(x^j \mid x_0^j, x_1^j) \tag{102}$$

Note that

$$\prod_{j\neq d} \kappa_t^j(x^j \mid x_0^j, x_1^j) = p_t(x^{-d}|x_0, x_1^{-d}) \tag{103}$$

and hence

$$p_0(x_0)p_1(x_1^{-d}) \prod_{j\neq d} \kappa_t^j(x^j \mid x_0^j, x_1^j)/p_t(x^{-d}) = p_t(x_0, x_1^{-d}|x^{-d}) \tag{104}$$

Therefore,

$$p_t(x^d = c|x^{-d}) = \sum_{x_0,x_1^{-d}} p_t(x_0, x_1^{-d}|x^{-d}) \sum_{x_1^d} p_1^d(x_1^d|x_1^{-d})\kappa_t^d(x^d = c \mid x_0^d, x_1^d) \tag{105}$$

Step 2 To learn this objective, we can simply use cross entropy:

$$\mathcal{L}_q(\theta) \tag{106}$$

$$= -\sum_{d,c} \mathbb{E}_{x^{-d},t} p_t(x^d = c \mid x^{-d}) \log p^d_{\theta,t}(c \mid x^{-d}) \tag{107}$$

$$= -\sum_{d,c} \mathbb{E}_t \sum_{x^{-d}} p(x^{-d}) \sum_{x_0, x_1^{-d}} p_t(x_0, x_1^{-d} | x^{-d}) \sum_{x_1^d} p_1^d(x_1^d | x_1^{-d}) \kappa_t^d(x^d = c \mid x_0^d, x_1^d) \log p^d_{\theta,t}(c \mid x^{-d}) \tag{108}$$

$$= -\sum_{d,c} \mathbb{E}_t \sum_{x^{-d}} \sum_{x_0, x_1^{-d}} p_t(x_0, x_1^{-d}, x^{-d}) \sum_{x_1^d} p_1^d(x_1^d | x_1^{-d}) \kappa_t^d(x^d = c \mid x_0^d, x_1^d) \log p^d_{\theta,t}(c \mid x^{-d}) \tag{109}$$

$$\begin{aligned} = & -\sum_{d,c} \mathbb{E}_t \sum_{x_0, x_1^{-d}, x^{-d}} p_0(x_0)\, p_1(x_1^{-d}) \prod_{j \neq d} \kappa_t^j(x^j \mid x_0^j, x_1^j) \\ & \qquad \times \left[\sum_a p_1^d(a \mid x_1^{-d}) \kappa_t^d(c \mid x_0^d, a) \right] \log p^d_{\theta,t}(c \mid x^{-d}) \end{aligned} \tag{110}$$

$$\begin{aligned} = & -\sum_{d,c} \mathbb{E}_t \sum_{x_0, x_1, x} p_0(x_0)\, p_1(x_1^{-d}) p_1^d(x_1^d \mid x_1^{-d}) \\ & \qquad \times \left[\prod_{j \neq d} \kappa_t^j(x^j \mid x_0^j, x_1^j) \right] \kappa_t^d(x^d \mid x_0^d, x_1^d) \\ & \qquad \times \left[\sum_a p_1^d(a \mid x_1^{-d}) \kappa_t^d(c \mid x_0^d, a) \right] \log p^d_{\theta,t}(c \mid x^{-d}) \end{aligned} \tag{111}$$

The last line hold as the remaining part does not depend on the newly added x^d variable.

$$\begin{aligned} \mathcal{L}_q(\theta) = & -\sum_{d,c} \mathbb{E}_t \sum_{x_0, x_1, x} p_0(x_0) p_1(x_1) K_t(x \mid x_0, x_1) \\ & \qquad \times \left[\sum_a p_1^d(a \mid x_1^{-d}) \kappa_t^d(c \mid x_0^d, a) \right] \log p^d_{\theta,t}(c \mid x^{-d}) \end{aligned} \tag{112}$$

Step 3 Finally,

$$-\mathbb{E}_{t, X_0 \sim p_0, X_1 \sim p_1, X_t \sim K_t(\cdot | X_0, X_1)} \sum_{d,c} \left[\sum_a p_1^d(a \mid X_1^{-d}) \kappa_t^d(c \mid X_0^d, a) \right] \log p^d_{\theta,t}(c \mid X_t^{-d}). \tag{113}$$

C.2.2 Learning the temporal potential $\phi_t^d(\cdot \mid x^{-d})$

Now, we consider how to learn $\phi_t^d(\cdot \mid x^{-d})$. We follow the same three steps as above.

Step 1 Recall that

$$\phi_t^d(c \mid x^{-d}) = \partial_t^{(d)} \log p_t(x^d = c \mid x^{-d}) = \frac{\partial_t^{(d)} p_t(x^d = c \mid x^{-d})}{p_t(x^d = c \mid x^{-d})}. \tag{114}$$

Since $\partial_t^{(d)}$ only differentiates the d-th interpolant kernel, we have $\partial_t^{(d)} p_t(x^{-d}) = 0$, and therefore

$$\partial_t^{(d)} p_t(x^d = c \mid x^{-d}) = \frac{1}{p_t(x^{-d})} \partial_t^{(d)} p_t(x^d = c, x^{-d}) \tag{115}$$

$$= \frac{1}{p_t(x^{-d})} \sum_{x_0, x_1} p_0(x_0) p_1(x_1) \partial_t \kappa_t^d(c \mid x_0^d, x_1^d) \prod_{j \neq d} \kappa_t^j(x^j \mid x_0^j, x_1^j). \tag{116}$$

Factorizing p_1 as before gives

$$\partial_t^{(d)} p_t(x^d = c \mid x^{-d}) = \sum_{x_0, x_1^{-d}} p_t(x_0, x_1^{-d} \mid x^{-d}) \sum_{x_1^d} p_1^d(x_1^d \mid x_1^{-d}) \partial_t \kappa_t^d(c \mid x_0^d, x_1^d). \tag{117}$$

Step 2 To learn ϕ_t^d, we consider the quadratic objective

$$\mathcal{L}_\phi(\psi) = \sum_{d,c} \mathbb{E}_{x^{-d},t}\Bigg[\frac{1}{2} p_t(x^d = c \mid x^{-d}) \phi^d_{\psi,t}(c \mid x^{-d})^2 \\ - \partial_t^{(d)} p_t(x^d = c \mid x^{-d}) \phi^d_{\psi,t}(c \mid x^{-d})\Bigg]. \tag{118}$$

Its first-order condition is

$$p_t(x^d = c \mid x^{-d}) \phi^d_{\psi,t}(c \mid x^{-d}) - \partial_t^{(d)} p_t(x^d = c \mid x^{-d}) = 0, \tag{119}$$

whose solution is exactly $\phi^d_{\psi,t} = \phi^d_t$.

Substituting the expressions derived above for $p_t(x^d = c \mid x^{-d})$ and $\partial_t^{(d)} p_t(x^d = c \mid x^{-d})$ gives

$$\mathcal{L}_\phi(\psi) = \sum_{d,c} \mathbb{E}_t \sum_{x^{-d}, x_0, x_1^{-d}} p_t(x_0, x_1^{-d}, x^{-d}) \Bigg[\frac{1}{2}\left(\sum_a p_1^d(a \mid x_1^{-d}) \kappa_t^d(c \mid x_0^d, a)\right) \phi^d_{\psi,t}(c \mid x^{-d})^2 \\ - \left(\sum_a p_1^d(a \mid x_1^{-d}) \partial_t \kappa_t^d(c \mid x_0^d, a)\right) \phi^d_{\psi,t}(c \mid x^{-d})\Bigg]. \tag{120}$$

Using

$$p_t(x_0, x_1^{-d}, x^{-d}) = p_0(x_0) p_1(x_1^{-d}) \prod_{j \neq d} \kappa_t^j(x^j \mid x_0^j, x_1^j), \tag{121}$$

we obtain

$$\mathcal{L}_\phi(\psi) = \sum_{d,c} \mathbb{E}_t \sum_{x_0, x_1^{-d}, x^{-d}} p_0(x_0) p_1(x_1^{-d}) \prod_{j \neq d} \kappa_t^j(x^j \mid x_0^j, x_1^j) \\ \times \Bigg[\frac{1}{2}\left(\sum_a p_1^d(a \mid x_1^{-d}) \kappa_t^d(c \mid x_0^d, a)\right) \phi^d_{\psi,t}(c \mid x^{-d})^2 \\ - \left(\sum_a p_1^d(a \mid x_1^{-d}) \partial_t \kappa_t^d(c \mid x_0^d, a)\right) \phi^d_{\psi,t}(c \mid x^{-d})\Bigg]. \tag{122}$$

As above, the remaining integrand does not depend on the newly introduced x_1^d or x^d. We can therefore multiply by the normalized distributions $p_1^d(x_1^d \mid x_1^{-d})$ and $\kappa_t^d(x^d \mid x_0^d, x_1^d)$ and sum over these variables. This gives

$$\mathcal{L}_\phi(\psi) = \sum_{d,c} \mathbb{E}_t \sum_{x_0, x_1, x} p_0(x_0) p_1(x_1) K_t(x \mid x_0, x_1) \\ \times \Bigg[\frac{1}{2}\left(\sum_a p_1^d(a \mid x_1^{-d}) \kappa_t^d(c \mid x_0^d, a)\right) \phi^d_{\psi,t}(c \mid x^{-d})^2 \\ - \left(\sum_a p_1^d(a \mid x_1^{-d}) \partial_t \kappa_t^d(c \mid x_0^d, a)\right) \phi^d_{\psi,t}(c \mid x^{-d})\Bigg]. \tag{123}$$

Step 3 Recognizing the preceding joint distribution as the usual endpoint and interpolant sampling procedure, we finally obtain

$$\mathcal{L}_\phi(\psi) = \mathbb{E}_{t,\, X_0 \sim p_0,\, X_1 \sim p_1,\, X_t \sim K_t(\cdot \mid X_0, X_1)} \sum_{d,c} \Bigg[\frac{1}{2}\left(\sum_a p_1^d(a \mid X_1^{-d}) \kappa_t^d(c \mid X_0^d, a)\right) \phi^d_{\psi,t}(c \mid X_t^{-d})^2 \\ - \left(\sum_a p_1^d(a \mid X_1^{-d}) \partial_t \kappa_t^d(c \mid X_0^d, a)\right) \phi^d_{\psi,t}(c \mid X_t^{-d})\Bigg]. \tag{124}$$

C.2.3 Final Objectives

The objectives above assume that endpoint samples are drawn from the correct distributions. In practice, as in the masked-diffusion case, we only have direct samples from one endpoint, while samples from the other endpoint are generated by the current sampler. We therefore use the same fixed-point iteration with tempered importance weighting and ESS control.

Let $\bar{w}(X_1)$ denote the tempered importance weight. The final training objectives are

$$\mathcal{L}_q(\theta) = -\mathbb{E}_{\substack{t,\, X_0\sim p_0,\, X_1\sim q_{\theta^-},\bar{w}(X_1)\\ X_t\sim K_t(\cdot|X_0,X_1)}} \sum_{d,c}\left[\sum_a p_1^d(a \mid X_1^{-d})\kappa_t^d(c \mid X_0^d, a)\right]\log p_{\theta,t}^d(c \mid X_t^{-d}), \tag{125}$$

and

$$\mathcal{L}_\phi(\psi) = \mathbb{E}_{\substack{t,\, X_0\sim p_0,\, X_1\sim q_{\theta^-},\bar{w}(X_1)\\ X_t\sim K_t(\cdot|X_0,X_1)}} \sum_{d,c}\left[\frac{1}{2}\left(\sum_a p_1^d(a \mid X_1^{-d})\kappa_t^d(c \mid X_0^d, a)\right)\phi_{\psi,t}^d(c \mid X_t^{-d})^2 - \left(\sum_a p_1^d(a \mid X_1^{-d})\partial_t\kappa_t^d(c \mid X_0^d, a)\right)\phi_{\psi,t}^d(c \mid X_t^{-d})\right]. \tag{126}$$

D Further related work

Neural samplers on discrete spaces. Early work on discrete neural samplers uses autoregressive models (Wu et al., 2019; Pan et al., 2021; Hibat-Allah et al., 2021; Nicoli et al., 2020; McNaughton et al., 2020; Białas et al., 2022; 2023; Sanokowski et al., 2023), with subsequent work developing architectural refinements (Biazzo, 2023; Del Bono et al., 2025; Białas et al., 2026). These models provide explicit generation probabilities and can be trained through variational objectives (Wu et al., 2019; Nicoli et al., 2020). GFlowNets instead learn generation policies from unnormalized rewards (Bengio et al., 2021; Malkin et al., 2022; Zhang et al., 2022; Jain et al., 2022; Zhang et al., 2023), while variational annealing and diffusion samplers address combinatorial optimization and statistical physics (Sanokowski et al., 2024; 2025). Recent CTMC-based approaches include locally equivariant rate models (Holderrieth et al., 2025a; Ou et al., 2025b), importance-weighted energy-based generator matching with bootstrapping (Woo et al., 2025), optimal-control approaches to masked diffusion (Zhu et al., 2025; Guo et al., 2026a), and discrete adjoint matching or off-policy bridge learning (Guo et al., 2026b; Carter et al., 2026). Applications include sampling disordered materials and alloys (Damewood et al., 2022; Du et al., 2026b; Blessing et al., 2026b; Min et al., 2026), with related work also investigating enhanced sampling (Du et al., 2026a).

Among masked energy-based samplers, MDNS trains with importance-weighted denoising cross entropy (Zhu et al., 2025). Our method retains the cross-entropy objective but replaces one-hot reconstruction targets with Gibbs soft labels, combining this supervision with tempered importance weighting. This use of local energy information is motivated by continuous neural samplers, including adjoint Schrödinger bridge sampling, neural flow samplers, and bridge matching samplers, which incorporate energy-gradient information into their learning objectives (Liu et al., 2025a; Chen et al., 2025; Blessing et al., 2026c). Single-site energy differences play a related informational role in discrete spaces, without requiring a differentiable extension of the energy, and are also exploited by discrete ASBS (Guo et al., 2026b). Our construction converts these local comparisons into categorical supervision. This local conditional-matching objective is also related to distribution-based target concrete score matching (Zhang et al., 2025).

Masked diffusion models. Discrete diffusion with uniform or masking noise (Hoogeboom et al., 2021; Austin et al., 2021) has been formulated in continuous time (Campbell et al., 2022), through ratio estimation (Meng et al., 2022; Sun et al., 2023; Lou et al., 2024; Zhang et al., 2025), and through flow or generator matching along general probability paths (Campbell et al., 2024; Gat et al., 2024; Shaul et al., 2025; Holderrieth et al., 2025b). For masking processes, training reduces to a weighted cross-entropy objective for conditional categorical predictions (Sahoo et al., 2024; Shi et al., 2024). Under value-independent masking, the exact clean-symbol conditional does not depend on time once

the revealed context is given (Ou et al., 2025a; Zheng et al., 2025). Masked models also admit an any-order autoregressive interpretation (Uria et al., 2014; Hoogeboom et al., 2022; Shih et al., 2022), while masked parallel decoding predates these diffusion formulations (Ghazvininejad et al., 2019; Chang et al., 2022; Bond-Taylor et al., 2022).

Trained from data, masked diffusion has been developed for language and protein modeling (Nie et al., 2025a; Gong et al., 2025; Nie et al., 2025b; Ye et al., 2025; Arriola et al., 2025; Wang et al., 2024). Related advances include guidance (Nisonoff et al., 2025; Schiff et al., 2025), correctors and remasking (Zhao et al., 2024; Wang et al., 2025), reveal-order planning (Liu et al., 2025b; Peng et al., 2025; Kim et al., 2025), and energy-based or uniform-state variants (Xu et al., 2025; Sahoo et al., 2025). Our setting differs from these data-trained constructions: the target is specified through an unnormalized energy, without access to target-distributed training samples. We nevertheless exploit the same conditional-generation structure. For a sampled single-site reveal order, the conditional generation probability is a tractable product of categorical prediction probabilities (Hoogeboom et al., 2022). This permits exact importance weights on an augmented space that includes the generation path and its reveal order. It does not imply that the marginal probability of a random-order model, obtained by summing over reveal orders, is tractable.

Discrete interpolant and flow models. Discrete flow matching and generator matching learn transition dynamics along general interpolant paths (Gat et al., 2024; Holderrieth et al., 2025b). However, they do not consider constructing time-reversal forward and backward between two arbitrary distributions. He et al. (2026) develop a general framework for learning such dynamics from endpoint interpolants. However, all these models only consider data-based training, while our interpolant framework makes use of the Gibbs label for efficient energy-based training.

E Experimental setup

This section collects the setup of every experiment of Section 5, one subsection per task family. Further results are in Section F.

E.1 Lattice models

E.1.1 Tasks

Both models are defined on a periodic two-dimensional lattice of D sites carrying q-state spins $\sigma_i \in \{1, \dots, q\}$, with

$$E(\sigma) = -\sum_{\langle ij \rangle} \delta_{\sigma_i \sigma_j} \ \text{(Potts)}, \qquad E(\sigma) = -\sum_{\langle ij \rangle} s_i s_j, \quad s_i = 2\sigma_i - 3 \in \{\pm 1\} \ \text{(Ising)}, \tag{127}$$

each bond counted once, and $p_\beta(\sigma) = e^{-\beta E(\sigma)}/Z(\beta)$ at inverse temperature β. The Ising energy is the $q = 2$ Potts energy in its usual convention using $\delta_{\sigma_i \sigma_j} = (1 + s_i s_j)/2$. The three phases of the Ising model are $\beta = 0.30$ (disordered), $\beta = \beta_c = \frac{1}{2}\ln(1 + \sqrt{2}) = 0.4407$ (critical) and $\beta = 0.60$ (ordered) and $\beta = 0.70$, $\beta \approx \beta_c = \ln(1 + \sqrt{3}) = 1.0050$, and $\beta = 1.20$ for the Potts model. The ordered phase of the Potts model has $q = 3$ symmetry-related modes rather than two. The reference structures used to compute reference values come from a long Monte Carlo chain with cluster updates (Wolff, 1989; Swendsen & Wang, 1987). Example configurations are shown in Figure 4.

E.1.2 Evaluation metrics

Each method draws $M = 8192$ independent configurations. Writing $a^{(m)} = \ln \hat{\nu}(\sigma^{(m)}) - \ln q(\sigma^{(m)})$ for the log importance weight of draw m and $\bar{w}^{(m)} = e^{a^{(m)}} / \sum_{m'} e^{a^{(m')}}$ for its self-normalized form. The evaluation metrics are computed as follows:

$$\mathrm{ESS} = \frac{1}{M \sum_m (\bar{w}^{(m)})^2} \in \left[\tfrac{1}{M}, 1\right], \tag{128}$$

$$|\Delta \ln Z| = \left| \ln \tfrac{1}{M} \textstyle\sum_m e^{a^{(m)}} - \ln Z \right|, \tag{129}$$

$$\mathrm{TVD}_{\mathrm{mode}} = \tfrac{1}{2} \textstyle\sum_{k=1}^{q} \left| \hat{\pi}_k - \pi_k^{\mathrm{ref}} \right|, \qquad \hat{\pi}_k = \tfrac{1}{M} \textstyle\sum_m \rho_k(\sigma^{(m)}), \tag{130}$$

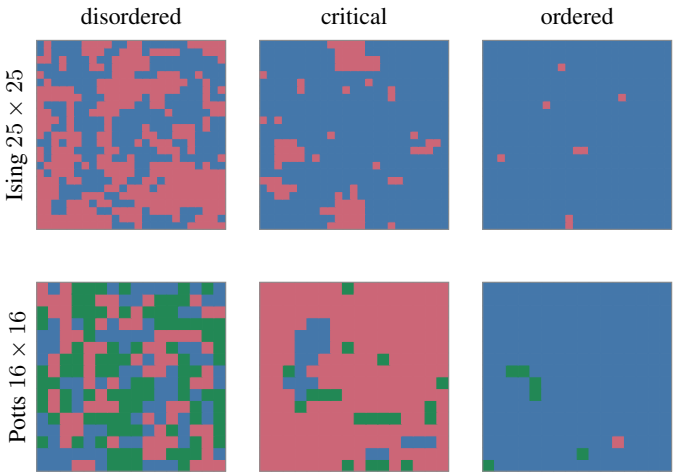


Figure 4: Example configurations of the Ising and Potts model at different phases.

where $\rho_k(\sigma)$ assigns each configuration to one of the q ordered modes: for Ising, ρ_k is the indicator of the sign of the magnetization $m(\sigma) = \frac{1}{D}\sum_i s_i$ (the two modes of $m > 0$ and $m < 0$); for Potts, of the majority color $\arg\max_k \sum_i \delta_{\sigma_i k}$, with a tie shared equally between the tied colors. π^{ref} is the reference distribution over the modes, exactly uniform at zero field (Section E.1.1). In the amortized runs, where a field breaks the symmetry, it is estimated from reference samples.

E.1.3 Hyperparameters

All methods train from the energy alone, with no reference data, and are given the same step budget, the same batch size and the same network capacity where the architecture admits it. No method uses MCMC at evaluation.

Table 2: Settings shared by every method and state point.

Systems	Ising 25×25 ($q = 2$, $D = 625$); Potts 16×16 ($q = 3$, $D = 256$)
Phases per system	3 (disordered, critical, ordered); see Section E.1.1
Seeds	3 per (system, phase, method)
Training steps	150k
Evaluation draws	8192 per seed, independent, no MCMC correction
Backbone	periodic residual CNN, 2-D circular convolutions, 96 channels, 7 residual blocks, kernel 3, dilations covering the torus (all methods but LEAPS)
Optimizer	AdamW (Loshchilov & Hutter, 2019) without weight decay (Adam for LEAPS)

DIG-IN. A masked interpolant. The sampler reveals one site per network evaluation in a uniformly random order, D evaluations per draw, so the proposal density of every draw is exact. Training runs the fixed-point loop: every 100 gradient steps the EMA model draws 512 configurations into a replay buffer of 2048, each stored with its exact table of single-site energy differences. The denoiser is trained by soft cross-entropy onto the Gibbs labels of the buffered endpoints at freshly masked copies of them, with the labels damped toward the EMA model's prediction ($\eta = 0.1$) and the per-endpoint losses weighted by importance weights tempered to an ESS of 0.8 within each training batch. Batch 128, learning rate 3×10^{-4}, EMA decay 0.995, gradient clipping at 1.0.

MDNS. The masked diffusion neural sampler (Zhu et al., 2025), run in the same codebase with the same interpolant, backbone, sampler, buffer and optimizer as DIG-IN. It is trained by cross-entropy onto the endpoint's own token rather than a heat-bath label, weighted by the full self-normalized importance weights, without tempering and without damping.

PDNS. The proximal diffusion neural sampler (Guo et al., 2026a): MDNS with its importance weights tempered to an ESS of 0.8 within each training batch, the same rule as DIG-IN. Labels are

hard and there is no damping, so PDNS differs from DIG-IN in exactly the label and the damping, and from MDNS in exactly the tempering.

DASBS. The discrete adjoint Schrödinger bridge sampler (Guo et al., 2026b): a time-conditioned controller and a time-free corrector, both on the shared backbone, over a uniform reference CTMC with a log-linear noise schedule ($\gamma = 1$, $\alpha = 0.5$). The two are trained in alternating stages of Bregman matching, 400 controller steps and then 200 corrector steps; the controller regresses onto the target's single-site score ratios at the endpoint, which needs the energy-difference table. Each batch of 128 endpoints is replicated 8 times, the buffer of 2048 is refreshed with 512 rollouts every 20 steps, and a draw takes 100 steps. Learning rate 10^{-3}, EMA decay 0.9999. With memory, the path weights give $\ln Z$ only at convergence and no importance weights, so its $|\Delta \ln Z|$ and ESS are not reported; the memoryless variant drops the corrector and has unbiased path weights.

LEAPS. LEAPS (Holderrieth et al., 2025a) learns a CTMC that transports the uniform distribution to the target along the annealed path $\rho_t \propto e^{-t\beta E}$, with a locally equivariant rate network (20 channels, kernels 3, 5, 7 and 9) and a free-energy MLP (hidden width 200), trained on the squared residual of the Kolmogorov forward equation at buffered trajectory states. We use training rollouts of 128 walkers every 20 steps, fill a buffer of 50 rollouts, and each loss batch holds 128 states. Rates are clamped at 10. Learning rate 3×10^{-4}, decayed by 0.995 every 1000 steps. Importance weights follow from the discrete Jarzynski equality.

E.1.4 Computational costs

Each run trains on one 80 GB H100 for 150k steps. Table 3 reports the training time and the number of energy calls during training, counted per configuration. A method calls the target for two things: the energy $E(\sigma)$ of a configuration, for its importance weight or the annealed path, and the table of single-site energy differences $\{E(\sigma^{d\leftarrow n}) - E(\sigma)\}_{d\le D,\, n\le q}$, which DIG-IN needs for its heat-bath labels and DASBS and LEAPS for their training targets. For the lattice models the energy is local, and the whole table follows from one neighbour-count pass at about the cost of a single energy evaluation, so we count it as one call. For a target that is only available as a black box, the table needs the energy of every configuration that differs from σ in one site, that is $D(q-1)+1$ energy evaluations instead of 1; the black-box column counts the same runs at this rate, 626 calls per table for the Ising model and 513 for the Potts model.

Table 3: **Computational cost.** Training time in hours (mean±standard deviation over 3 seeds and the three phases) and the total number of energy calls during training, with the single-site difference table counted as one call (local) or as $D(q-1)+1$ calls (black box), the cost it would have if the energy were not local.

	Ising 25×25			Potts 16×16		
		energy calls			energy calls	
Method	train (h)	local	black box	train (h)	local	black box
DIG-IN (ours)	$6.33_{\pm 0.11}$	$1.5{\cdot}10^{6}$	$4.8{\cdot}10^{8}$	$1.89_{\pm 0.07}$	$1.5{\cdot}10^{6}$	$3.9{\cdot}10^{8}$
MDNS	$6.05_{\pm 0.09}$	$7.7{\cdot}10^{5}$	$7.7{\cdot}10^{5}$	$1.79_{\pm 0.10}$	$7.7{\cdot}10^{5}$	$7.7{\cdot}10^{5}$
PDNS	$6.42_{\pm 0.35}$	$7.7{\cdot}10^{5}$	$7.7{\cdot}10^{5}$	$2.22_{\pm 0.30}$	$7.7{\cdot}10^{5}$	$7.7{\cdot}10^{5}$
DASBS	$11.57_{\pm 0.09}$	$1.1{\cdot}10^{8}$	$6.4{\cdot}10^{10}$	$5.60_{\pm 0.28}$	$1.1{\cdot}10^{8}$	$5.3{\cdot}10^{10}$
DASBS, memoryless	$10.72_{\pm 0.13}$	$1.6{\cdot}10^{8}$	$9.6{\cdot}10^{10}$	$5.19_{\pm 0.19}$	$1.6{\cdot}10^{8}$	$7.9{\cdot}10^{10}$
LEAPS	$5.85_{\pm 0.12}$	$5.2{\cdot}10^{8}$	$2.5{\cdot}10^{11}$	$2.86_{\pm 0.17}$	$4.2{\cdot}10^{8}$	$1.6{\cdot}10^{11}$

E.2 Amortized training for lattice models

E.2.1 Setup

The distribution over the thermodynamic conditions (β, h) is set as follows: Half of the samples are drawn uniformly over $[\beta_{\min}, \beta_{\max}] \times [h_{\min}, h_{\max}]$. The other half is drawn from an acquisition map on an 8×12 grid of cells C_k of the box. Each cell is scored by the trace of the Fisher information of the target family, estimated from the model's own draws,

$$a_k = \operatorname{Tr} \operatorname{Cov}_{\sigma \sim q(\cdot|\theta),\, \theta \in C_k} \left[\nabla_{\tilde{\theta}} \ln \tilde{p}(\sigma \mid \theta) \right], \tag{131}$$

with $\theta = (\beta, h)$ rescaled to the unit box ($\tilde{\theta}$). For Ising the two components are $\partial_\beta \ln \tilde{p} = \sum_{\langle ij \rangle} s_i s_j + h \sum_i s_i$ and $\partial_h \ln \tilde{p} = \beta \sum_i s_i$, so a_k combines the specific heat and the magnetic susceptibility, which peak at the phase transition. Each component is divided by its median over the cells, and cells are drawn with probability proportional to $a_k^{1/2}$. The acquisition only decides which conditions receive more training, not the labels or the loss, so the fixed point is unchanged. Importance weights are self-normalized per acquisition cell as $\ln Z(\beta, h)$ depends on (β, h). Table 4 lists the settings.

Table 4: Settings of the amortized runs.

	Ising	Potts, $q = 3$
Lattice	50×50 torus, $D = 2500$	24×24 torus, $D = 576$
Training box	$\beta \in [0.39, 0.65]$, $\lvert h \rvert \le 0.1$	$\beta \in [0.8, 1.3]$, $\lvert h \rvert \le 0.01$
Zero-field β_c	$\frac{1}{2}\ln(1+\sqrt{2}) = 0.4407$	$\ln(1+\sqrt{3}) = 1.00505$
Network	residual CNN on the torus: 7 blocks, 96 channels, 3×3 circular convolutions, conditioned by FiLM (Perez et al., 2018) on Fourier features of (t, β, h); dilations $(1, 1, 2, 4, 8, 1, 1)$ for Ising so that 7 blocks span the 50×50 torus, none for Potts	
Optimizer	AdamW (Loshchilov & Hutter, 2019), learning rate 3×10^{-4}, no weight decay, gradient clipping at 1.0, EMA 0.995	
Batch / buffer	1536 / 49 152	1536 / 196 608
Damping, ESS target	$\eta = 0.5$; weights tempered to ESS 0.8 per acquisition cell	
Training	4120–4200 rounds of 100 steps	18713–19063 rounds of 100 steps
Sampler	random-order unmasking with an exact proposal density, 512 (Ising) and 288 (Potts) network evaluations, the first 64 revealing a single site	
Seeds	3	2
Evaluation	9×9 lattice over the box, 5000 draws per cell, EMA weights of the last checkpoint, no MCMC, no resampling	

E.3 Alloy phase diagrams

E.3.1 Tasks

Semi-grand canonical ensemble. A binary alloy on a fixed lattice of D sites is described by occupations $\sigma = (\sigma_1, \ldots, \sigma_D) \in \{-1, +1\}^D$, where $\sigma_i = +1$ marks the first species (Ag in Ag–Pd, Au in Cu–Au). Its configurational energy is a cluster expansion (Sanchez et al., 1984), a multilinear polynomial in the occupations,

$$E(\sigma) = J_\emptyset + \sum_\omega J_\omega \prod_{i \in \omega} \sigma_i, \tag{132}$$

where the sum runs over the clusters ω (points, pairs and higher-order clusters) of the orbits that the expansion retains. We use the DFT-fitted effective cluster interactions J_ω of Damewood et al. (2022), which are also the energies used by Min et al. (2026) and Du et al. (2026b), without subtracting a pure-element reference. At condition $\theta = (T, \Delta\mu)$, with $\Delta\mu$ the difference between the chemical potentials of the first and second species and $\beta = 1/k_\mathrm{B}T$, the target is

$$\nu_\theta(\sigma) = \frac{\tilde{\nu}_\theta(\sigma)}{Z(\theta)}, \qquad \tilde{\nu}_\theta(\sigma) = e^{-\beta H_\theta(\sigma)}, \qquad H_\theta(\sigma) = E(\sigma) - \frac{\Delta\mu}{2} \sum_{i=1}^{D} \sigma_i. \tag{133}$$

Let $n(\sigma) = \sum_i (1 + \sigma_i)/2$ be the number of atoms of the first species, so that $c = n/D$ is the composition. Since $\sum_i \sigma_i = 2n - D$, Equation (133) equals $E - \Delta\mu\, n$ up to a constant that does not depend on the configuration. That form is the convention of Min et al. (2026), and both forms define the same distribution.

Systems. Table 5 lists the two cells. Sites are indexed by their integer coordinates in the fcc primitive basis. This maps each cell onto a dense periodic grid on which 3-D convolutions act, while the energy is always evaluated on the physical fcc geometry through the cluster tables. The training box is the same for both systems. The evaluation sweep is the grid on which the reference data are tabulated.

Table 5: The two alloy tasks.

	Ag–Pd	Cu–Au
Lattice, cell	fcc, $5 \times 5 \times 5$ primitive	fcc, $8 \times 4 \times 4$ primitive
Sites D	125	128
$\sigma_i = +1$	Ag	Au
Phases	one miscibility gap	Cu_3Au ($L1_2$), CuAu ($L1_0$), $CuAu_3$ ($L1_2$)
Training box	$T \in [200, 900]$ K, $\Delta\mu \in [-0.4, 0.4]$ eV	$T \in [200, 900]$ K, $\Delta\mu \in [-0.4, 0.4]$ eV
Evaluation T	200–900 K, step 25 K (29)	200–900 K, step 25 K (29)
Evaluation $\Delta\mu$	-0.40 to 0.40 eV, step 0.01 eV (81)	-0.20 to 0.15 eV, step 0.01 eV (36)
State points	2349	1044
Reference tie lines	15, highest at 550 K	74, highest at 750 K

Training conditions. All three methods draw training conditions with $1/T$ uniform on $[1/900, 1/200]\,\mathrm{K}^{-1}$ and $\Delta\mu$ uniform on $[-0.4, 0.4]$ eV. Because the log-density is linear in β, sampling uniformly in β spreads the magnitude of the energy term evenly across the box. DIG-IN replaces half of each batch of conditions with draws from an adaptive acquisition map (Section E.3.4).

Exact reference. The reference data tabulate the fixed-composition partition functions

$$Z_n(T) = \sum_{\sigma:\, n(\sigma)=n} e^{-\beta E(\sigma)}, \qquad n = 0, \dots, D, \tag{134}$$

at every temperature of the sweep. Since $\Delta\mu$ couples only to n, the partition function and the composition distribution follow in closed form on the whole plane,

$$\ln Z(\theta) = \mathrm{logsumexp}_n \Big[\ln Z_n(T) + \beta \tfrac{\Delta\mu}{2}(2n - D) \Big], \qquad p_\theta(n) = \frac{Z_n(T)\, e^{\beta\Delta\mu(2n-D)/2}}{Z(\theta)}, \tag{135}$$

and so does the mean composition $\langle c \rangle_\theta = \sum_n (n/D)\, p_\theta(n)$. All errors are measured against these exact values.

E.3.2 Constructing the phase diagram

We follow the constructor of Min et al. (2026, App. A.4) and apply it unchanged to every method. However, for completeness, we outline the procedures below.

Fixed-composition free energy from weighted draws. At condition θ, a sampler returns draws $\sigma^{(m)} \sim q(\cdot \mid \theta)$, $m = 1, \dots, M$, with log-weights $a^{(m)} = \ln \tilde{\nu}_\theta(\sigma^{(m)}) - \ln q(\sigma^{(m)} \mid \theta)$ (for FrOGS, the path weight A_1 of Holderrieth et al., 2025a). Since $\mathbb{E}_q[e^a\, \mathbf{1}\{n(\sigma) = n\}] = Z_n(T)\, e^{\beta\Delta\mu(2n-D)/2}$,

$$\hat{Z}_n(T) = e^{-\beta\Delta\mu(2n-D)/2}\, \frac{1}{M} \sum_{m:\, n(\sigma^{(m)})=n} e^{a^{(m)}}, \qquad \hat{f}(c, T) = -\frac{k_\mathrm{B} T}{D} \ln \hat{Z}_n(T), \quad c = n/D, \tag{136}$$

is an unbiased estimate of $Z_n(T)$ at every composition that some draw visits. It is a histogram without a bin-width parameter, because composition is discrete. Because $\hat{Z}$ itself is estimated from the same draws, the estimate is absolute.

Pooling over chemical potentials. $Z_n(T)$ does not depend on $\Delta\mu$, so every condition $\theta_i = (T, \Delta\mu_i)$ at a given temperature gives an independent estimate $\hat{y}_i(n) = \ln \hat{Z}_n$ of the same curve. Its standard error is $s_i(n) = N_{\mathrm{eff},i}(n)^{-1/2}$, with $N_{\mathrm{eff},i}(n) = \big(\sum_{m \in n} e^{a^{(m)}}\big)^2 / \sum_{m \in n} e^{2a^{(m)}}$ the effective sample count of the draws at θ_i that land at composition n. The estimates are merged by precision weighting,

$$\hat{y}(n) = \frac{\sum_i w_i(n)\, \hat{y}_i(n)}{\sum_i w_i(n)}, \qquad \mathrm{Var}\, \hat{y}(n) = \frac{1}{\sum_i w_i(n)}, \qquad w_i = s_i^{-2}. \tag{137}$$

Error bars at temperature T are then widened by the Birge ratio (Birge, 1932), $\lambda = \max\{1, \mathrm{median}_{n:\, \sum_i w_i(n) \geq 5}\, \chi^2_\nu(n)\}^{1/2}$ with $\chi^2_\nu(n) = \frac{1}{k_n - 1} \sum_i w_i (\hat{y}_i - \hat{y})^2$ over the k_n contributing conditions. This correction only acts when the conditions disagree by more than their errors predict.

Common tangent. Two phases coexist at temperature T when a single line is tangent to $f(\cdot, T)$ at two compositions $c_a < c_b$. These are the edges of the lower convex hull of $\{(c, \hat{f}(c, T))\}$. A hull edge is accepted as a tie line only if (i) at least four compositions lie strictly between its ends, counted on the composition axis $n = 0, \ldots, D$ rather than on the list of resolved compositions; (ii) $\hat{f}$ lies at least three standard errors above the chord at some interior composition, or no draw at any $\Delta\mu$ at that temperature visited the interior; and (iii) an accepted gap overlapping it exists at the next lower temperature, chained from the lowest temperature. Criterion (i) matters at low temperature, where ordered phases are sharp and most compositions are never visited. Criterion (iii) must chain on *accepted* edges, since otherwise a rejected isotherm could license the one above it. A two-phase region ends at the highest temperature at which its edge is still accepted. The reference curves in Figure 3 are the same common-tangent construction on the exact $\ln Z_n(T)$.

Figure 5 shows the whole construction on Cu–Au, at the coldest isotherm and at the last one before the two-phase region closes. At both, the curve estimated from DIG-IN's draws follows the exact one and yields the same tie lines, 4 of 4 at 200 K and 2 of 2 at 700 K.

E.3.3 Evaluation metrics

At every state point θ of the sweep each method draws $M = 5000$ independent configurations, with no MCMC moves and no resampling. With self-normalized weights $\bar{w}^{(m)} = e^{a^{(m)}} / \sum_{m'} e^{a^{(m')}}$,

$$\mathrm{ESS}(\theta) = \frac{1}{M \sum_m (\bar{w}^{(m)})^2} \in \left[\tfrac{1}{M}, 1\right], \tag{138}$$

$$|\Delta \ln Z|(\theta)/D = \frac{1}{D}\left| \ln \frac{1}{M} \sum_m e^{a^{(m)}} - \ln Z(\theta) \right|, \tag{139}$$

$$|\Delta c|(\theta) = \left| \sum_m \bar{w}^{(m)}\, c(\sigma^{(m)}) - \langle c \rangle_\theta \right|, \tag{140}$$

with $\ln Z(\theta)$ and $\langle c \rangle_\theta$ exact from Equation (135). The ESS is that of Kong et al. (1994): it equals 1 when all weights are equal and $1/M$ when one draw carries all the weight. Figure 3 reports each quantity averaged over all state points of the sweep, then the mean and standard deviation over seeds.

E.3.4 Hyperparameters

Tables 6 and 7 list the settings of DIG-IN and FrOGS, which are identical for Ag–Pd and Cu–Au. Both methods train for 60k steps on the same energy, conditioning box and conditioning distribution, and are evaluated with 5000 draws per state point without MCMC moves. Code for FrOGS was not public at the time of our experiments, so we reimplemented it following Sections 4 and A.2–A.9 of Min et al. (2026).

Differences from the published FrOGS configuration. To keep the step budget and evaluation equal across methods, our runs differ from Min et al. (2026, App. A.11, A.14): 60k training steps rather than 200k, 50 CTMC steps rather than 125, no MCMC moves during training rather than 10 per time step, a flat locally equivariant CNN as rate network rather than a three-encoder, two-decoder pyramid with 16 channels, a prior with 64 channels rather than 32 and a prior learning rate of 10^{-3} rather than 10^{-4}, and a Cu–Au box that stops at 900 K rather than 1200 K.

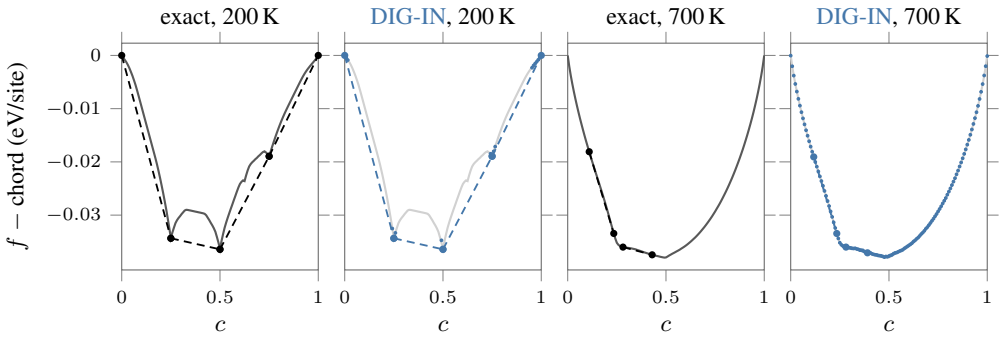


Figure 5: **Common-tangent construction on Cu–Au.** Mixing free energy against composition at 200 K (left pair) and 700 K (right pair), measured from the chord through the pure compositions. In each pair: exact (left) and from the draws of one DIG-IN run (right, exact curve in grey). Dashed chords are the accepted tie lines, whose ends are the phase boundaries at that temperature.

Table 6: DIG-IN hyperparameters, identical for Ag–Pd and Cu–Au.

Interpolant and sampler	
Noising process	masked (absorbing state)
Reveal schedule	one site per network evaluation (D evaluations per draw)
Training labels	heat-bath conditionals from the exact single-site energy-difference table, cached with the buffered endpoint
Loss	soft cross-entropy on masked sites
Network	
Backbone	periodic residual CNN, 3-D circular convolutions over the cell
Width, depth, kernel	96 channels, 7 pre-norm residual blocks (GroupNorm), kernel 3
Dilations	chosen so that the receptive field covers the periodic cell
Conditioning	Fourier features of normalized $(T, \Delta\mu)$ (6 frequencies), 2-layer MLP, FiLM in every block
Output head	zero-initialized (uniform predictor at step 0)
Optimization	
Optimizer	AdamW (Loshchilov & Hutter, 2019), learning rate 3×10^{-4}, weight decay 0
Batch size, steps	1024, 60 000
Gradient clipping	global norm 1.0
EMA of weights	decay 0.995 (the EMA model is the sampler)
Fixed-point loop	
Replay buffer	25 000 endpoints, pre-filled with 12×2048 uniform configurations
Exploration	2048 fresh draws from the EMA model every 200 optimizer steps
Proximal damping η	0.5 (labels blended with the EMA model's prediction; $\eta = 1$ is undamped)
Importance weighting	
Scheme	tempered weights, exponent and normalization per conditioning cell
Weight grid	4×4 cells over $(T, \Delta\mu)$
Target ESS	0.9 of the cell's buffered endpoints
Minimum occupancy	24 buffered endpoints; thinner cells use the pooled exponent
Conditioning acquisition	
Uniform fraction	0.5
Grid, EMA, floor, power	16×16 cells, moment EMA 0.9, uniform floor 0.05, score exponent 0.25
Evaluation	5000 draws per state point

Table 7: FrOGS hyperparameters, identical for Ag–Pd and Cu–Au.

Transport	
Rate network	locally equivariant CNN: centre-masked convolution (kernel 3), then a state-dependent convolution (kernel 5); 24 channels; FiLM conditioning
Rates	Metropolis-reparametrized, clamped at 10
Free-energy head	MLP, hidden width 200, input $(t, T, \Delta\mu)$
Path discretization	$\Delta t = 0.02$ (50 CTMC steps)
Horizon curriculum	from 1 step to 50, quadratic over the first 2000 optimizer steps
Prior	
Network	PixelCNN-style causal 3-D convolutions (van den Oord et al., 2016), 64 channels, 4 blocks, kernel 3, FiLM conditioning (Perez et al., 2018)
Training	
Rollouts	64 walkers every 20 optimizer steps; the buffer keeps the last 20 rollouts
Loss batch	64 (time, walker) states drawn from one buffered rollout
MCMC moves during training	none
Optimizer	Adam (Kingma & Ba, 2015); learning rate 3×10^{-4} (transport and free-energy head), 10^{-3} (prior)
Schedule	step decay by 0.995 every 1000 steps
Gradient clipping, steps	global norm 1.0, 60 000
Evaluation	5000 draws per state point, 50 CTMC steps, no MCMC moves

F FURTHER EXPERIMENTS

COMPONENT ABLATION

DIG-IN adds three ingredients to the masked interpolant: heat-bath (Gibbs) labels, importance weights tempered to a target ESS, and proximal damping. We remove one at a time on Ising 25×25 at the settings of Table 1: 'no damping' sets $\eta = 1$, 'no Gibbs score' trains on the endpoint's own token, once with tempered weights and once at full strength ('no smoothing'), and 'no importance weights' weights every endpoint equally. Away from the ordered phase all variants converge (Figure 6 and Table 8). In the ordered phase only the full method is both mode-complete and accurate: without damping 2 of 3 seeds collapse to one mode, without the Gibbs score all seeds collapse at full weights and the run never orders at tempered ones, and without importance weights both modes survive but $|\Delta \ln Z|$ grows from $0.7_{\pm 0.4}$ to $15.8_{\pm 9.9}$ $\times 10^{-3}$ nats.

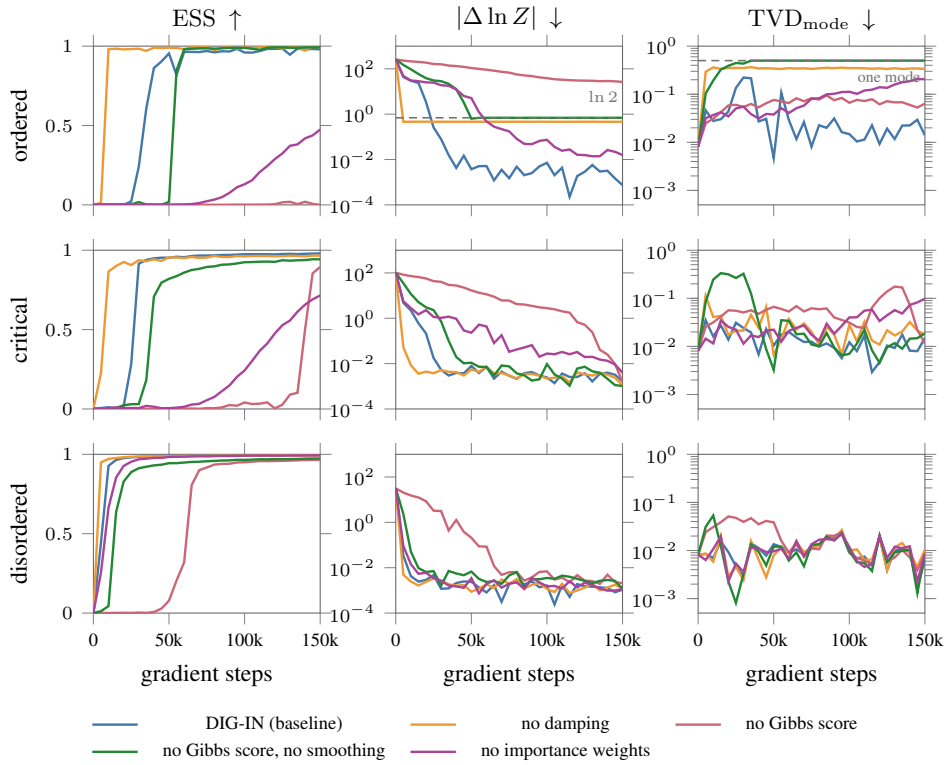


Figure 6: **Component ablation on Ising** 25×25**.** ESS, $|\Delta \ln Z|$ and $\mathrm{TVD}_{\mathrm{mode}}$ against training steps for the ordered (top), critical (middle) and disordered phase (bottom), mean over 3 seeds. Dashed: $\ln 2$ and $\mathrm{TVD}_{\mathrm{mode}} = 0.5$, the values of a sampler that holds only one of the two ordered modes.

Table 8: **Component ablation, final evaluation.** Mean±standard deviation over 3 seeds, $|\Delta \ln Z|$ in 10^{-3} nats. PDNS and MDNS are repeated from Table 1.

	disordered, $\beta = 0.30$			critical, $\beta = \beta_c = 0.4407$			ordered, $\beta = 0.60$		
variant	$\lvert\Delta \ln Z\rvert$ $\times 10^3$ ↓	ESS ↑	$\mathrm{TVD}_{\mathrm{mode}}$ ↓	$\lvert\Delta \ln Z\rvert$ $\times 10^3$ ↓	ESS ↑	$\mathrm{TVD}_{\mathrm{mode}}$ ↓	$\lvert\Delta \ln Z\rvert$ $\times 10^3$ ↓	ESS ↑	$\mathrm{TVD}_{\mathrm{mode}}$ ↓
DIG-IN (baseline)	$1.1_{\pm 0.4}$	$.995_{\pm .000}$	$.006_{\pm .004}$	$1.6_{\pm 0.3}$	$.980_{\pm .001}$	$.015_{\pm .004}$	$0.7_{\pm 0.4}$	$.979_{\pm .011}$	$.014_{\pm .007}$
no damping	$1.2_{\pm 0.5}$	$.992_{\pm .001}$	$.010_{\pm .007}$	$1.1_{\pm 0.4}$	$.966_{\pm .007}$	$.017_{\pm .012}$	$462.0_{\pm 326.4}$	$.994_{\pm .005}$	$.336_{\pm .232}$
no Gibbs score	$2.1_{\pm 1.9}$	$.965_{\pm .001}$	$.008_{\pm .004}$	$1.5_{\pm 0.9}$	$.895_{\pm .009}$	$.011_{\pm .008}$	$26479.6_{\pm 456.9}$	$.001_{\pm .000}$	$.064_{\pm .022}$
no Gibbs, no smoothing	$1.2_{\pm 0.8}$	$.974_{\pm .002}$	$.009_{\pm .005}$	$1.0_{\pm 0.2}$	$.943_{\pm .003}$	$.019_{\pm .004}$	$693.1_{\pm 0.8}$	$.993_{\pm .000}$	$.500_{\pm .000}$
no importance weights	$1.1_{\pm 0.8}$	$.993_{\pm .000}$	$.009_{\pm .004}$	$4.0_{\pm 1.7}$	$.715_{\pm .012}$	$.098_{\pm .050}$	$15.8_{\pm 9.9}$	$.475_{\pm .219}$	$.206_{\pm .127}$
reference, from Table 1									
PDNS	$2.9_{\pm 1.9}$	$.956_{\pm .001}$	$.014_{\pm .006}$	$1.5_{\pm 0.6}$	$.917_{\pm .008}$	$.016_{\pm .008}$	$693.1_{\pm 0.4}$	$.988_{\pm .003}$	$.500_{\pm .000}$
MDNS	$1.4_{\pm 0.8}$	$.961_{\pm .002}$	$.012_{\pm .004}$	$1.9_{\pm 1.0}$	$.925_{\pm .006}$	$.016_{\pm .006}$	$692.7_{\pm 0.4}$	$.983_{\pm .007}$	$.500_{\pm .000}$

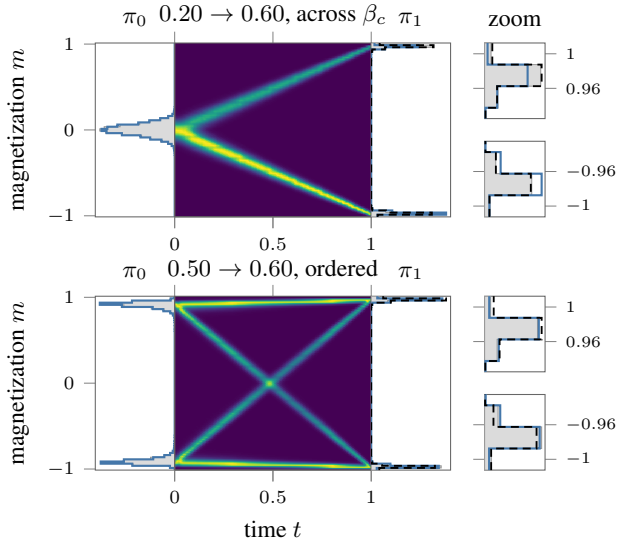
F.1 Bridging two Boltzmann distributions

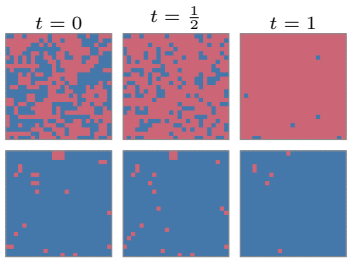
The interpolant does not need a tractable reference: here both ends are Ising distributions $\pi_\beta(s) \propto e^{\beta \sum_{\langle ij \rangle} s_i s_j}$ on a 25×25 torus at different β, and neither can be sampled exactly. The sampler is given draws from π_{β_0}, produced by MCMC, and the energy of π_{β_1}, and learns a CTMC that carries one into the other; path weights give an unbiased estimate of $Z(\beta_1)/Z(\beta_0)$, which is known exactly (Kaufman, 1949). We consider two transports: from $\beta_0 = 0.20$ to $\beta_1 = 0.60$, across the phase transition, which splits one mode into two, and from $\beta_0 = 0.50$ to $\beta_1 = 0.60$, within the ordered phase, which moves two modes apart. Both recover the free-energy difference to within $0.5_{\pm 0.6}$ and $10.0_{\pm 6.3} \times 10^{-4}$ of its value, and after reweighting both split their draws between the two modes close to evenly (Figure 7). The ordered-to-ordered transport has the lower ESS: the interpolant pairs opposite modes of the two ends half of the time, and those paths lose their order midway.

F.2 Additional results

Amortized lattice models. The ESS of the amortized samplers stays high across the (β, h) grid and is lowest at and just below the zero-field critical point, where the fluctuations are largest: at worst 0.20 for Ising and 0.56 for Potts (Figure 8).

Alloys. DIG-IN keeps a high ESS over nearly the whole $(T, \Delta\mu)$ sweep of both alloys and loses it only at a small fraction of state points, mostly on the coldest isotherms, whereas the ESS of FrOGS is low almost everywhere (Figure 9).





	0.20 → 0.60	0.50 → 0.60
exact $\Delta \ln Z$	297.9	115.2
$\lvert\Delta \ln Z\rvert \downarrow$	$0.014_{\pm 0.019}$	$0.115_{\pm 0.073}$
ESS ↑	$.105_{\pm .023}$	$.036_{\pm .019}$
$\mathrm{TVD}_{\mathrm{mode}} \downarrow$	$.007_{\pm .003}$	$.030_{\pm .024}$

Figure 7: **Bridging two Ising temperatures.** *Left:* one row per transport, time running left to right. The map shows $P(m)$ along the path; beside it, the magnetization at β_0 and β_1 (grey: Swendsen–Wang reference, blue: the model's draws, dashed: reweighted), and the two target modes magnified. *Right, top:* one draw carried along each transport, at $t = 0$, $\frac{1}{2}$ and 1 (upper row: across β_c, lower row: ordered). *Right, bottom:* $\ln Z(\beta_1) - \ln Z(\beta_0)$ error, ESS and reweighted $\mathrm{TVD}_{\mathrm{mode}}$, mean ± standard deviation over 3 seeds.

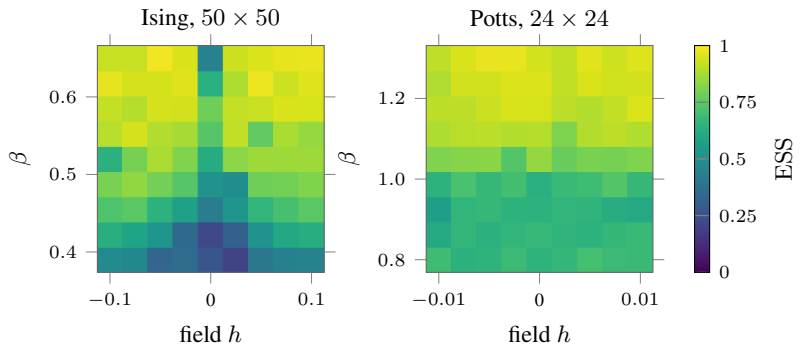


Figure 8: **Effective sample size of the amortized samplers across** (β, h)**.** Fraction of the 5000 draws per cell, mean over seeds.

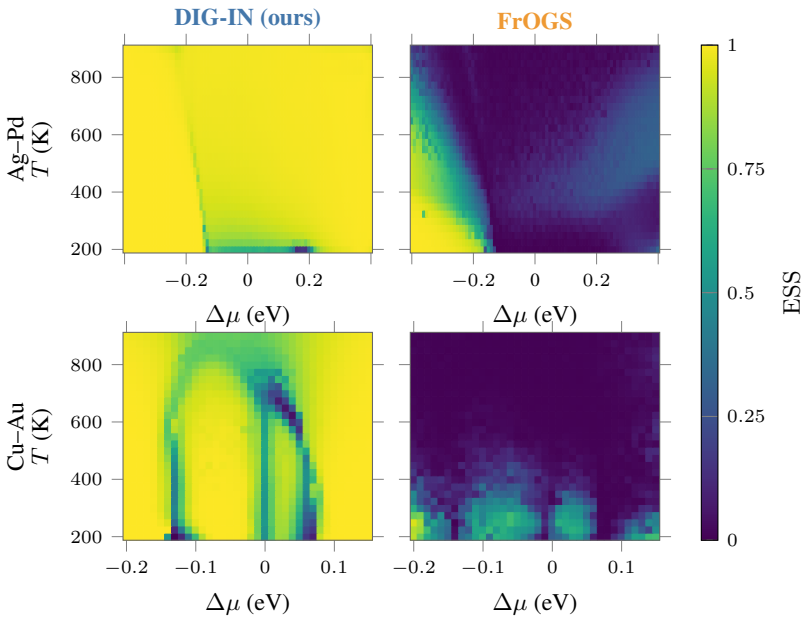


Figure 9: **Effective sample size across the alloy sweep.** Fraction of the 5000 draws per state point, mean over 3 seeds, for Ag–Pd (top) and Cu–Au (bottom).