LEARNING TO SAMPLE FROM POWER DISTRIBUTIONS FOR AMORTIZED MPE INFERENCE IN PROBABILISTIC GRAPHICAL MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a neural sampling approach for repeated most probable explanation (MPE) inference on a fixed probabilistic graphical model (PGM). Our approach learns to sample from the power family $P_\beta(\mathbf{q} \mid \mathbf{e}) \propto P(\mathbf{q} \mid \mathbf{e})^\beta$, where increasing β concentrates probability on high-probability configurations and progressively biases sampling toward MPE solutions. We instantiate this formulation with a factor-reading neural sampler that directly uses the PGM's potentials to generate assignments conditioned on the observed evidence. The sampler is amortized across evidence queries and trained with a reverse-KL objective and a curriculum that progressively increases β, requiring neither labeled MPE solutions nor computation of normalizing constants. At inference time, we draw samples from the learned proposal and return the highest-scoring assignment, with optional query-specific test-time refinement using the same objective to further improve solution quality. Across nine PGMs, a single sample per query achieves higher mean solution quality than the evaluated neural MPE baselines. With query-specific refinement, our method also outperforms the evaluated branch-and-bound solvers on six benchmark PGMs under matched wall-clock budgets. Beyond direct MPE inference, we analyze how the power parameter β affects distributional fidelity and MPE solution quality, and show that the learned proposals can improve existing inference procedures by warm-starting annealed sequential Monte Carlo and guiding branch-and-bound search.

1 INTRODUCTION

Many machine learning tasks, including medical diagnosis, structured prediction, planning, and decision making, can be formulated as inference in a probabilistic graphical model (PGM) (Koller & Friedman, 2009). In many applications, the PGM remains fixed while inference is repeated under changing evidence configurations. Each choice of observed variables and their values defines a different conditional inference problem over the remaining query variables, so a single PGM can induce a combinatorially large family of related inference queries. Classical algorithms such as belief propagation (Pearl, 1988), variable elimination, mini-bucket methods (Dechter, 1999), and branch-and-bound (Otten, 2012; Marinescu & Dechter, 2009; Givry, 2023) solve these queries independently, requiring substantial repeated computation as the evidence changes. This motivates *amortized inference* (Gershman & Goodman, 2014), which learns reusable computation across queries.

Neural methods have amortized computation for approximate marginal inference, MAP/MPE inference, learned search heuristics, and related tasks (Arya et al., 2024a; 2025; Malhotra et al., 2025; Agarwal et al., 2022; Arya et al., 2024c). Building on this idea, we study a complementary approach based on amortized sampling for most probable explanation (MPE) inference, which seeks the highest-probability assignment to the unobserved variables given evidence and is NP-hard in general (Shimony, 1994). A general-purpose sampler must accurately represent the full conditional distribution, but MPE does not require this level of fidelity. Because our sampler is used specifically for mode discovery, it can instead place increasing emphasis on high-probability configurations most relevant to the MPE solution. This specialization makes the learning problem less demanding and motivates controlling how strongly the learned distribution concentrates around such configurations. The *power family* (Kirkpatrick et al., 1983; Neal, 2001; Friel & Pettitt, 2008) provides a natural

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.

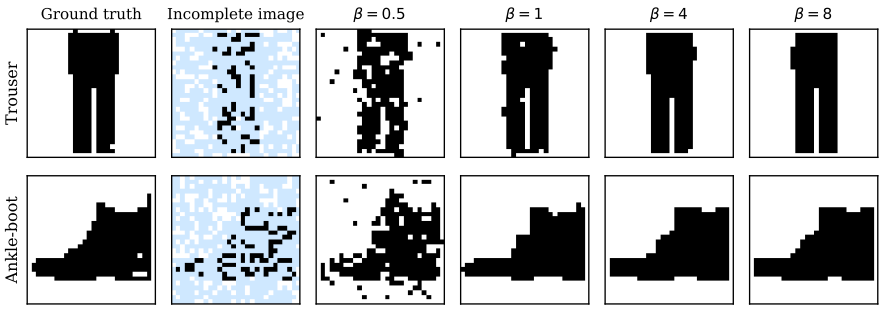


Figure 1: A single amortized sampler completes a partially observed Fashion-MNIST image (Xiao et al., 2017) as an MPE query. With 70% of pixels hidden, increasing β progressively concentrates the completions from broad, noisy samples toward a clean, high-probability completion.

way to control this concentration: $P_\beta(\mathbf{q} \mid \mathbf{e}) = P(\mathbf{q} \mid \mathbf{e})^\beta / Z_\beta(\mathbf{e})$ for $\beta > 0$ where $Z_\beta(\mathbf{e})$ is the normalizing constant. At $\beta = 1$, P_β recovers the posterior; increasing β preserves the ordering, and therefore the global maximizers, of $P(\mathbf{q} \mid \mathbf{e})$ while concentrating mass on high-probability configurations. As $\beta \to \infty$, it concentrates on the MPE assignments. The power family thus provides a continuous path from posterior representation to mode discovery.

We ask how a neural sampler can exploit this path to amortize mode discovery across evidence queries of a fixed PGM. Larger β increases concentration around high-probability modes but can reduce coverage of distinct modes in multimodal PGMs. We therefore study this concentration-coverage tradeoff and its effect on MPE performance. Figure 1 illustrates this behavior on a PGM fitted to Fashion-MNIST (Xiao et al., 2017): increasing β yields progressively more concentrated completions of a partially observed image. Additional details are provided in Appendix F.

We instantiate this approach with a factor-reading neural sampler that directly consumes the numerical potentials defining the PGM. Propagating these potentials through the factor graph allows the sampler to account for both the model structure and the observed evidence when scoring candidate assignments. The sampler generates assignments through a masked-reveal process (Hoogeboom et al., 2022) and is trained across evidence configurations of the same PGM by minimizing the reverse-KL objective $\mathrm{KL}(q_\theta(\mathbf{x} \mid \mathbf{e}) \,\|\, P_\beta(\mathbf{x} \mid \mathbf{e}))$ using a REINFORCE estimator with a batch-mean baseline (Williams, 1992). This mode-seeking objective allows a finite-capacity sampler to focus on high-probability regions rather than requiring faithful coverage of the full probability mass.

Our contributions are:

1. **A power-tempering formulation for amortized MPE.** We learn neural proposals for the power family P_β using on-policy reverse KL and only the PGM's unnormalized scores, without labeled MPE solutions. A power curriculum increases β and checkpoints each power, producing a ladder of proposals from a single training run.
2. **A factor-reading amortized sampler for fixed PGMs.** We convert the PGM's factor potentials, including soft costs and hard constraints, into a masked-reveal sampler through iterated message passing. The sampler is trained once per PGM, reused across unseen evidence configurations, and generates diverse candidates whose quality improves with the sample budget.
3. **An analysis of concentration versus coverage.** We study how increasing β shifts samples from broad coverage toward high-probability assignments and how this tradeoff affects MPE performance.
4. **Amortized MPE inference and composition with existing solvers.** We use the sampler for best-of-K inference, improve solutions through query-specific test-time training, and accelerate existing inference procedures by warm-starting annealed SMC and guiding branch-and-bound.

2 Preliminaries

Let $\mathbf{X} = \{X_1, \ldots, X_n\}$ denote the random variables of a discrete probabilistic graphical model (PGM). We use uppercase bold letters for sets of random variables and lowercase bold letters for their assignments; thus, $\mathbf{X} = \mathbf{x}$ denotes an assignment $\mathbf{x}$ to $\mathbf{X}$. For an inference query, we partition

$\mathbf{X}$ into disjoint sets of evidence variables $\mathbf{E}$ and free (query) variables $\mathbf{Q}$, such that $\mathbf{X} = \mathbf{E} \cup \mathbf{Q}$ and $\mathbf{E} \cap \mathbf{Q} = \emptyset$. An **evidence configuration** is an assignment $\mathbf{E} = \mathbf{e}$.

We represent the PGM by its unnormalized density $\widetilde{P}(\mathbf{x}) = \exp(\sum_c \phi_c(\mathbf{x}_c))$, with normalized distribution $P(\mathbf{x}) = \widetilde{P}(\mathbf{x})/Z$, where $Z = \sum_{\mathbf{x}} \widetilde{P}(\mathbf{x})$. Given evidence $\mathbf{E} = \mathbf{e}$, the posterior over the query variables is $P(\mathbf{q} \mid \mathbf{e})$. We define the corresponding **power distribution** with power $\beta > 0$ as $P_\beta(\mathbf{q} \mid \mathbf{e}) = P(\mathbf{q}, \mathbf{e})^\beta / Z_\beta(\mathbf{e})$, where $Z_\beta(\mathbf{e}) = \sum_{\mathbf{q}} P(\mathbf{q}, \mathbf{e})^\beta$. Because $P(\mathbf{e})^\beta$ is constant with respect to $\mathbf{q}$, $P_\beta(\mathbf{q} \mid \mathbf{e})$ is equivalently the normalized power posterior $P(\mathbf{q} \mid \mathbf{e})^\beta$. At β=1, it recovers the ordinary posterior. Increasing β preserves the ordering of assignments while concentrating probability mass on higher-scoring configurations. As $\beta \to \infty$, the distribution concentrates on $\arg\max_{\mathbf{q}} P(\mathbf{q}, \mathbf{e})$, the **most probable explanation (MPE)** solution set. We therefore treat $\beta \geq 0$ as a concentration parameter: $0 < \beta < 1$ flattens the posterior, while the regime $\beta \geq 1$ spans the posterior at β=1 and approaches MPE in the high-β limit.

Challenges in Power Sampling. The power family connects posterior inference to MPE, but increasing β makes exploration harder in multimodal distributions. As mass concentrates around high-probability configurations, transitions between modes become unlikely, leading to poor mixing in MCMC and importance-weight degeneracy in annealing and SMC (Neal, 2001; Del Moral et al., 2006; Kirkpatrick et al., 1983; Geyer, 1991). At the same time, the target is generally available only through its unnormalized score, since computing $Z_\beta(\mathbf{e})$ is intractable for the high-treewidth PGMs of interest (Roth, 1996) and ground-truth MPE assignments require solving an NP-hard problem (Shimony, 1994). Thus, effective high-β inference must discover relevant modes while learning directly from the unnormalized PGM score.

Amortization across evidence. We seek to reuse the same learned inference mechanism across evidence configurations of a fixed PGM (Gershman & Goodman, 2014). Since changing $\mathbf{e}$ changes the target $P_\beta(\mathbf{q} \mid \mathbf{e})$, a useful amortized sampler must generalize to unseen queries rather than memorize individual solutions. It must instead learn how to combine the shared factorization and potentials of the PGM with each new evidence configuration to generate high-quality solutions.

The central question is therefore: *for a fixed PGM, can we amortize exploration of high-probability configurations across the power family while retaining sufficient multimodal coverage for MPE inference on unseen evidence?*

3 METHOD

This section presents our method for learning amortized proposals for the power distributions of a PGM under varying evidence. We first derive the variational objective, then describe the masked-reveal proposal and factor-reading architecture used to parameterize it, its on-policy optimization, and the power curriculum used to obtain proposal checkpoints at multiple target powers.

3.1 VARIATIONAL FORMULATION

Direct sampling from $P_\beta(\mathbf{q} \mid \mathbf{e})$ is generally intractable at the scale of our target PGMs. We therefore learn a proposal $q_\theta(\mathbf{q} \mid \mathbf{e})$ that can be sampled efficiently. Specifically, we minimize the *reverse KL* (exclusive KL), $\mathrm{KL}(q_\theta(\mathbf{q} \mid \mathbf{e}) \,\|\, P_\beta(\mathbf{q} \mid \mathbf{e}))$ (Blei et al., 2017). Unlike forward KL, which penalizes insufficient proposal mass in regions occupied by the target and is associated with mass-covering behavior, reverse KL strongly penalizes proposal mass in low-probability target regions and is therefore typically mode-seeking (Minka, 2005; Blei et al., 2017). This behavior is well aligned with MPE, where discovering high-probability modes is more important than reproducing the probability mass of every mode. Reverse KL also avoids evaluating the target normalizer. Expanding the objective gives

$$\begin{aligned}\mathrm{KL}(q_\theta(\mathbf{q} \mid \mathbf{e}) \,\|\, P_\beta(\mathbf{q} \mid \mathbf{e})) &= \mathbb{E}_{q_\theta}\left[\log q_\theta(\mathbf{q} \mid \mathbf{e}) - \log P_\beta(\mathbf{q} \mid \mathbf{e})\right] \\ &= \mathbb{E}_{q_\theta}\left[\log q_\theta(\mathbf{q} \mid \mathbf{e}) - \beta \log P(\mathbf{q}, \mathbf{e})\right] + \log Z_\beta(\mathbf{e}),\end{aligned}$$

where $Z_\beta(\mathbf{e})$ is independent of θ. Training therefore requires neither labeled MPE solutions nor samples from the conditional power target $P_\beta(\mathbf{q} \mid \mathbf{e})$. Instead, it uses only proposal

samples and pointwise evaluations of the unnormalized PGM score $P(\mathbf{q}, \mathbf{e})$, without evaluating $Z_\beta(\mathbf{e})$. Dropping the constant $\log Z_\beta(\mathbf{e})$ yields the mode-seeking objective $\mathcal{L}_\beta(\theta; \mathbf{e}) = \mathbb{E}_{\mathbf{q}\sim q_\theta}[\log q_\theta(\mathbf{q} \mid \mathbf{e}) - \beta \log P(\mathbf{q}, \mathbf{e})]$ which we minimize for MPE.

3.2 Training the Amortized Sampler

Masked-reveal proposal. We parameterize $q_\theta(\mathbf{q} \mid \mathbf{e})$ as a masked-reveal autoregressive model (Austin et al., 2021; Hoogeboom et al., 2022). With the evidence fixed and the query variables masked, it reveals the variables over S steps in an arbitrary, order-agnostic order, predicting at each step a categorical distribution for the newly revealed variables from the current partial assignment and evidence. For a reveal trajectory $\tau = (C_1, \ldots, C_S)$, the proposal factorizes as $\log q_\theta(\mathbf{q} \mid \mathbf{e}, \tau) = \sum_s \sum_{i \in C_s} \log q_\theta(q_i \mid \mathbf{q}_{<s}, \mathbf{e}, C_s)$. We draw reveal trajectories from a fixed, θ-independent distribution $p(\tau \mid \mathbf{e})$, giving the joint proposal $q_\theta(\mathbf{q}, \tau \mid \mathbf{e}) = p(\tau \mid \mathbf{e}) q_\theta(\mathbf{q} \mid \mathbf{e}, \tau)$ and marginal $q_\theta(\mathbf{q} \mid \mathbf{e}) = \sum_\tau p(\tau \mid \mathbf{e}) q_\theta(\mathbf{q} \mid \mathbf{e}, \tau)$. Because this marginal sums over reveal orders and is intractable, we sample one trajectory per configuration and minimize the reverse KL of the joint proposal against the extended target $P_\beta(\mathbf{q} \mid \mathbf{e}) p(\tau \mid \mathbf{e})$. This trajectory-level reverse KL upper-bounds the marginal reverse KL $\mathrm{KL}(q_\theta(\mathbf{q} \mid \mathbf{e}) \,\|\, P_\beta(\mathbf{q} \mid \mathbf{e}))$ (Malkin et al., 2023). Each sampled trajectory also has an exactly computable log-probability and trajectory-level importance weight $w(\mathbf{q}, \tau) \propto P(\mathbf{q}, \mathbf{e})^\beta / q_\theta(\mathbf{q} \mid \mathbf{e}, \tau)$, where $p(\tau \mid \mathbf{e})$ cancels between the joint proposal and extended target.

Appendix C shows that the autoregressive parameterization is important in practice. A mean-field proposal that samples query variables independently collapses under the mode-seeking objective and obtains substantially lower $\log P$.

Factor-reading architecture. The masked-reveal policy is a message-passing network that operates on the PGM factor graph. The factor graph supplies the connectivity, while the numerical factor potentials expose the soft costs and hard constraints associated with assignments, which a connectivity-only network cannot observe. This factor representation allows the same learned parameters to be reused across evidence configurations of the fixed PGM. The architecture follows SINE's factor encoding (Arya et al., 2025), but uses iterated message passing and a per-variable reveal head to parameterize a sampler rather than a one-shot predictor. Appendix B details the architecture, including arbitrary-arity factors, and Appendix C provides ablations.

Gradient estimator. We optimize the trajectory-level objective $\widetilde{\mathcal{L}}_\beta(\theta; \mathbf{e})$ on-policy using the REINFORCE score-function estimator (Williams, 1992). For a sampled reveal trajectory τ, define the learning signal $A(\mathbf{q}, \tau) = \beta \log P(\mathbf{q}, \mathbf{e}) - \log q_\theta(\mathbf{q} \mid \mathbf{e}, \tau)$. Using $\mathbb{E}_{q_\theta(\mathbf{q}, \tau \mid \mathbf{e})}[\nabla_\theta \log q_\theta(\mathbf{q} \mid \mathbf{e}, \tau)] = 0$, the corresponding trajectory-level gradient is

$$\nabla_\theta \widetilde{\mathcal{L}}_\beta = -\mathbb{E}_{(\mathbf{q}, \tau) \sim q_\theta(\mathbf{q}, \tau \mid \mathbf{e})} \left[A(\mathbf{q}, \tau) \nabla_\theta \log q_\theta(\mathbf{q} \mid \mathbf{e}, \tau) \right].$$

For a minibatch of B sampled trajectories, we use a batch-mean baseline and the baseline-subtracted estimator

$$\widehat{\nabla_\theta \widetilde{\mathcal{L}}_\beta} = -\frac{1}{B} \sum_{b=1}^{B} \mathrm{sg}\Big(A(\mathbf{q}^{(b)}, \tau_b) - \bar{A} \Big) \nabla_\theta \log q_\theta(\mathbf{q}^{(b)} \mid \mathbf{e}, \tau_b), \text{ where } \bar{A} = \frac{1}{B} \sum_{b=1}^{B} A(\mathbf{q}^{(b)}, \tau_b)$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The $-\log q_\theta$ term in A distinguishes reverse-KL optimization from direct reward maximization. Removing it instead optimizes expected PGM score, directly favoring increasingly concentrated mass on high-scoring configurations.

Power curriculum. Rather than conditioning a single network on β, we anneal the training objective toward successive target powers. For each target, we ramp β from the previous power to the target and then hold it there to convergence before retaining the resulting model (Kirkpatrick et al., 1983; Rose, 1998). We then warm-start from this model checkpoint and anneal to the next target, so a single sequential training run produces one proposal per target power (Huang et al., 2017). Algorithm 1 summarizes the full training procedure, and the specific target powers, schedule, and convergence criterion are given with the experimental recipe in Sec. 4.

Algorithm 1: Amortized power-sampler training with masked-reveal sampling and baseline-subtracted REINFORCE.

Input: Fixed PGM P over variables V ($|V| = n$); batch size B; reveal steps S; target powers $\beta_1 < \cdots < \beta_R$; base power β_0; ramp/hold lengths $T_{\mathrm{r}}, T_{\mathrm{h}}$; evidence-ratio bounds $\mathrm{ER}_{\min}, \mathrm{ER}_{\max}$; learning rate η

Output: One checkpoint q_{θ_r} per target power β_r

for $r \leftarrow 1$ **to** R **do**
  for $k \leftarrow 1$ **to** $T_{\mathrm{r}} + T_{\mathrm{h}}$ **do**
    $\beta \leftarrow \beta_{r-1} + \min(k, T_{\mathrm{r}})\,(\beta_r - \beta_{r-1})/T_{\mathrm{r}}$ `// ramp over` T_{r} `steps, then hold at` β_r
    sample a complete assignment $\mathbf{x} \sim P$ by Gibbs sampling
    draw $\alpha \sim \mathcal{U}(\mathrm{ER}_{\min}, \mathrm{ER}_{\max})$; pick a random $\mathbf{E} \subseteq V$ with $|\mathbf{E}| = \lceil \alpha n \rceil$; set $\mathbf{e} \leftarrow \mathbf{x}_{\mathbf{E}}$, $\mathbf{Q} \leftarrow V \setminus \mathbf{E}$ `// sample evidence and project onto` $\mathbf{x}$
    initialize B assignments with $\mathbf{E}$ clamped to $\mathbf{e}$ and $\mathbf{Q}$ masked
    for $b \leftarrow 1$ **to** B **do**
      sample a reveal trajectory $\tau_b = (C_1, \ldots, C_S) \sim p(\tau \mid \mathbf{e})$, then reveal each block by sampling variable values from logits computed on the current masked assignment and PGM factors while accumulating $\log q_\theta(\mathbf{q}^{(b)} \mid \mathbf{e}, \tau_b)$
      $A_b \leftarrow \beta \log P(\mathbf{q}^{(b)}, \mathbf{e}) - \log q_\theta(\mathbf{q}^{(b)} \mid \mathbf{e}, \tau_b)$
    $\bar{A} \leftarrow B^{-1} \sum_b A_b$
    $\theta \leftarrow \theta + \frac{\eta}{B} \sum_{b=1}^{B} \mathrm{sg}(A_b - \bar{A})\, \nabla_\theta \log q_\theta(\mathbf{q}^{(b)} \mid \mathbf{e}, \tau_b)$
  $q_{\theta_r} \leftarrow \theta$ `// checkpoint after the hold at` β_r
return $\{q_{\theta_r}\}_{r=1}^{R}$

4 Empirical Evaluation of the Amortized Power Sampler

We evaluate the amortized power sampler by analyzing the effect of β on proposal quality, assessing one-shot and refined MPE inference, and testing its use for SMC warm-starting and variable and value ordering in branch-and-bound search. All evaluations use unseen evidence configurations of fixed PGMs, with matched wall-clock budgets for anytime comparisons.

Benchmark and amortized inference setting. Our main MPE evaluation uses nine fixed PGMs from three families: grids (GRID-20, GRID-40, GRID-80), graph-coloring WCSPs (LE450-2, LE450-3, LE450-4), both from UAI inference competition benchmarks (Elidan & Globerson, 2010; Dechter et al., 2022), and synthetic spin glasses on 6-regular graphs (GLASS-400, GLASS-800, GLASS-1200). The ordering experiment in Sec. 4.5 uses three additional instances, for twelve trained samplers total; instance sizes are listed in Appendix A.

For each PGM, we train one neural sampler with the REINFORCE procedure of Sec. 3 and amortize it across evidence queries of that fixed PGM. The factor-reading message-passing network has six layers and hidden width 64. We use Adam with learning rate 10^{-3}, 128 on-policy rollouts per step, and S masked-reveal steps per rollout on one NVIDIA A40 GPU. We set S to the largest value fitting memory: $S = 20$ for GRID-20, GRID-40, and GLASS-400, and $S = 10$ otherwise. The curriculum targets $\beta \in \{1, 2, 4, 6, 8, 10\}$, starting from $\beta_0 = 0.2$ and using 400 linear-ramp and 1,500 hold steps per rung, for 11,400 steps total.

For each PGM, we generate disjoint training, validation, and test query sets. Training queries are constructed from 16,000 Gibbs samples (Koller & Friedman, 2009), with a random 2%–10% of variables fixed to their sampled values. The same procedure, using independent Gibbs samples and evidence selections, produces 100 validation and 100 test queries per PGM. Validation queries are used only to analyze β and select checkpoints in Sec. 4.1, while the disjoint test queries are used for single-shot inference (Sec. 4.2), test-time training (Sec. 4.3), and SMC warm-starting (Sec. 4.4). The ordering experiment uses separate instances and the protocol in Sec. 4.5. Thus, the main evaluation measures generalization to unseen evidence configurations using one model per PGM.

4.1 Analysis of the power-ladder checkpoints

The power curriculum of Sec. 3 produces a family of checkpoints, one per power β, whose proposals become increasingly concentrated as β grows. Table 1 reports *diversity*, the mean free-variable

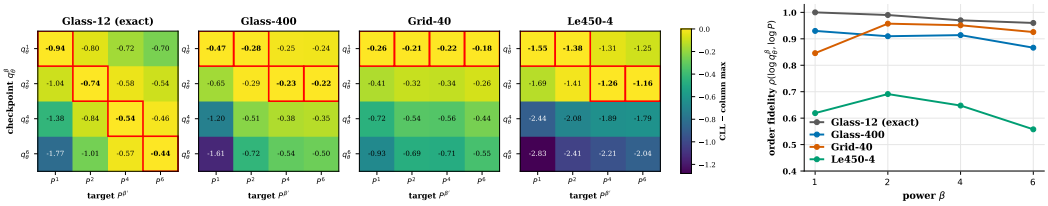
Table 1: Checkpoint validation: mean best-of-128 $\log P$ (higher is better) and diversity. Bold marks the selected β (row maximum in $\log P$).

		β=1		β=2		β=4		β=6		β=8		β=10	
Family	Instance	$\log P$	div	$\log P$	div	$\log P$	div	$\log P$	div	$\log P$	div	$\log P$	div
Grids	GRID-20	601.5	0.200	622.0	0.083	622.2	0.030	618.4	0.016	**622.6**	0.010	619.3	0.009
	GRID-40	2561.8	0.160	2649.6	0.065	2662.2	0.028	2665.2	0.017	2661.3	0.012	**2665.4**	0.007
	GRID-80	10162.1	0.156	10646.5	0.042	10706.9	0.021	10714.9	0.015	10720.5	0.010	**10731.1**	0.006
Spin glass	GLASS-400	682.2	0.443	736.0	0.293	**745.8**	0.179	742.6	0.133	739.2	0.078	740.0	0.090
	GLASS-800	1333.4	0.410	1469.6	0.235	**1490.8**	0.151	1489.4	0.103	1487.2	0.085	1474.8	0.047
	GLASS-1200	1993.2	0.426	2211.3	0.246	2234.7	0.140	**2236.9**	0.110	2218.4	0.079	2234.0	0.062
Coloring	LE450-2	−18304.9	0.007	**−18229.9**	0.012	−18313.7	0.004	−18372.6	0.014	−18240.9	0.006	−18581.3	0.007
	LE450-3	−8746.6	0.061	−8650.9	0.017	−8618.6	0.014	**−8559.2**	0.010	−8611.3	0.008	−8628.8	0.008
	LE450-4	−4802.3	0.137	**−4238.2**	0.067	−4391.1	0.041	−4327.0	0.031	−4697.8	0.047	−4304.9	0.022

Gini index $1 - \sum_v p_v^2$, where p_v is the sampled frequency of value v, and best-of-128 validation $\log P(\mathbf{q}, \mathbf{e})$. Throughout, for each query we take the best (highest) joint log-score $\log P(\mathbf{q}, \mathbf{e})$ that a method achieves, and every table reports its mean over the query set. For the amortized sampler this best is taken over K drawn samples (best-of-K), while for the anytime methods it is the best value reached within the wall-clock budget. Diversity decreases with β across all families, while best-of-128 $\log P$ generally improves before turning over on more multimodal instances, exposing a concentration–coverage trade-off.

We evaluate *distributional fidelity* along two complementary axes: probability-mass coverage and rank order. Coverage is measured by conditional log-likelihood (CLL), the mean per-free-variable log-probability assigned to reference samples from $P^\beta(\cdot \mid \mathbf{e})$ (Fig. 2a). On the 12-variable spin-glass instance GLASS-12, where P^β can be computed exactly, each checkpoint is most faithful to its corresponding target, yielding a column-wise diagonal-dominant cross-CLL matrix. The same qualitative pattern holds on larger instances, whose target distributions are approximated using parallel-tempering references (Geyer, 1991) (App. D), although coverage becomes increasingly selective at higher powers rather than reproducing the full target mass.

To test whether reduced mass coverage also harms MPE-relevant ordering, we measure Spearman correlation between $\log q_\theta^\beta$ and $\log P$ on samples from each checkpoint's target (Fig. 2b). Positive powers preserve configuration order, so the reference ranking is invariant to β. Correlation remains high on grids and spin glasses ($\rho = 0.85$–0.97) and weaker but substantial on coloring instace ($\rho \approx 0.6$–0.7). Thus, proposals may under-cover target mass while preserving much of the ranking relevant to best-of-K MPE. For each instance, we deploy the checkpoint maximizing best-of-128 $\log P$ on validation queries, balancing concentration and coverage.



(a) (b)

Figure 2: Checkpoint fidelity across four instances. (a) Cross-CLL matrix with checkpoints q_θ^β as rows and targets $P^{\beta'}$ as columns. (b) Spearman rank correlation between proposal and target scores. GLASS-12 uses exact references; the others use parallel tempering.

Although MPE mode discovery is our primary objective, we also evaluate the $\beta = 1$ checkpoint for posterior approximation (Fig. 3) on a separate 40×40 grid-structured Bayesian network, GRID-BN, with column-cutset evidence. This conditioning makes posterior inference tractable using iterative join-graph propagation (IJGP) (Mateescu et al., 2010), providing exact posterior marginals for evaluation (App. E). We compare with likelihood weighting (Koller & Friedman, 2009). The learned proposal attains low mean marginal ℓ_1 error with substantially fewer samples, but has substantially lower importance-weight effective sample size (ESS), $(\sum_k w_k)^2 / \sum_k w_k^2$ consistent with greater concentration. Because its unweighted, samples are not asymptotically posterior-distributed, this experiment measures finite-budget amortized posterior approximation rather than asymptotically consistent sampling.

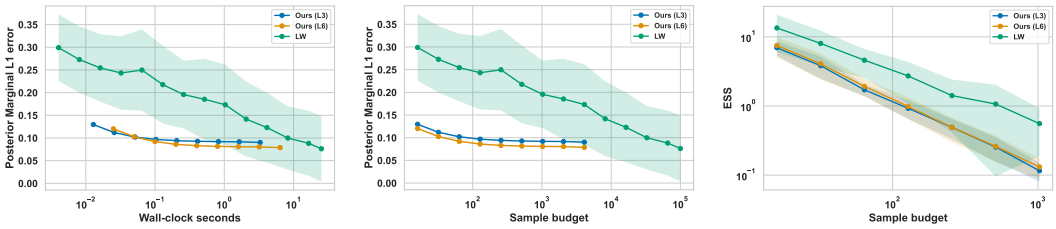


Figure 3: Posterior sampling at β=1 on GRID-BN column-cutset queries, scored against exact marginals. *Left/center:* mean marginal ℓ_1 vs. wall-clock time and sample budget K (log axes), for our sampler with 3- and 6-layer backbones (L3, L6) and likelihood weighting (LW). *Right:* importance-weight ESS vs. K; the sampler is used unweighted, so ESS is only a proposal-concentration diagnostic.

Table 2: Single-shot MPE performance: mean best $\log P$ over 100 test queries and per-query wall-clock time. Bold marks the best $\log P$.

		SSMP		SINE		GUIDE		Ours K=1		Ours K=8		Ours K=32	
Family	Instance	$\log P$	ms	$\log P$	ms	$\log P$	ms	$\log P$	ms	$\log P$	ms	$\log P$	ms
Grids	GRID-20	536.1	2	522.5	4	546.3	2	620.5	79	621.6	81	**621.9**	156
	GRID-40	2406	2	2210	4	2339	2	2661.8	82	2663.8	163	**2664.6**	579
	GRID-80	9790	2	8600	4	9371	2	10722.7	56	10726.2	299	**10727.9**	1117
Spin glass	GLASS-400	535.1	2	549.4	4	588.7	2	733.6	80	739.1	82	**742.9**	214
	GLASS-800	1164	2	1102	4	1208	2	1471.3	41	1482.3	61	**1486.1**	206
	GLASS-1200	1783	2	1653	4	1821	2	2221.1	41	2231.3	85	**2237.2**	303
Coloring	LE450-2	−20636	2	−21305	4	−20808	2	−18267	41	−18256	106	**−18251**	383
	LE450-3	−11458	2	−11484	4	−11490	2	−8617	41	−8605	109	**−8597**	399
	LE450-4	−6804	2	−6925	4	−5886	2	−4319	41	−4257	102	**−4225**	366

4.2 SINGLE-SHOT MPE INFERENCE

We evaluate whether the learned sampler can directly produce high-quality MPE solutions. Increasing β concentrates the proposal on high-probability regions, but finite-power concentration does not guarantee recovery of the global mode. In multimodal landscapes, the sampler may instead concentrate around a suboptimal basin.

Table 2 compares our best-of-K solutions with three neural MPE baselines: SSMP and GUIDE (Arya et al., 2024a), and SINE (Arya et al., 2025). These methods provide fast amortized inference but can converge to relatively poor local maxima on difficult multimodal instances. In contrast, our sampler produces higher-scoring configurations across all nine instances. Notably, a single sample ($K = 1$) already exceeds every baseline, while increasing the budget to $K = 8$ or $K = 32$ further improves $\log P$, with gains ranging from a few nats on grids to larger improvements on spin-glass and coloring instances. This shows that our learned proposal already provides a strong direct MPE solver; next, we show that its solutions can be further improved through query-specific refinement and that its intermediate checkpoints can also strengthen conventional inference procedures.

4.3 TEST-TIME TRAINING

Table 3: Test-time training for anytime MPE. We report the mean best $\log P$ over the 100 test queries within matched 20/60 s budgets. Bold marks the best directly comparable full-test result; "—" indicates no feasible solution, * a proven optimum, and † a `daoopt` mean over only 54/72 solved GRID-40 queries at 20/60 s

			B&B		Neural Solvers		
Family	Instance	Ours+TTT	`toulbar2`	`daoopt`	SSMP+ITSELF	SINE+ITSELF	GUIDE+ITSELF
Grids	GRID-20	624.6/625.4	624.3/625.1	**627.4/627.4***	559.7/559.7	522.5/522.5	564.8/564.8
	GRID-40	**2670.9/2673.1**	2637/2637	2677/2679†	2428.8/2428.8	2209.5/2209.5	2411.4/2411.4
	GRID-80	**10731.3/10731.6**	—	—	9869.9/9869.9	8600.3/8600.3	9715.5/9715.5
Spin glass	GLASS-400	**751.8/755.8**	738.4/741.3	698.7/700.4	669.0/669.0	550.1/550.1	630.5/630.5
	GLASS-800	**1499.9/1506.7**	1466.3/1470.3	—	1223.4/1223.4	1101.5/1101.5	1287.3/1287.3
	GLASS-1200	**2257.3/2268.1**	2204.9/2208.6	—	1867.5/1867.5	1653.1/1653.1	1947.7/1947.7
Coloring	LE450-2	**−18222/ − 18214**	−20534/ − 20422	—	−20331/ − 20331	−21263/ − 21263	−20419/ − 20419
	LE450-3	**−8494/ − 8469**	−10100/ − 10041	—	−11143/ − 11143	−11481/ − 11481	−11009/ − 11009
	LE450-4	**−4033/ − 3967**	−5198/ − 5145	—	−6484/ − 6484	−6924/ − 6924	−5523/ − 5523

A finite-power proposal can still concentrate around a suboptimal basin, motivating *test-time training* (TTT) of the already strong one-shot solutions. This is the same inference-time optimization idea that ITSELF applies to the neural MPE baselines with a self-supervised loss (Arya et al., 2024a), except that we fine-tune the sampler with our own on-policy objective. For each evidence query, we reset the model to the deployed checkpoint and fine-tune it *on that query* using the same on-policy reverse-KL objective and REINFORCE estimator as in training, at $\beta = 8$ or the deployment power if larger. At each step, we draw 64 fresh samples and retain the best configuration found, yielding an anytime solution under a fixed wall-clock budget.

Table 3 compares *Ours+TTT* with two conventional solvers, `toulbar2` (Givry, 2023) and `daoopt` (Otten, 2012), as well as neural test-time-training baselines obtained by applying ITSELF (Arya et al., 2024a) to SSMP (Arya et al., 2024b), SINE (Arya et al., 2025), and GUIDE (Arya et al., 2024a), under matched 20- and 60-second budgets.

Starting from a strong one-shot solution, test-time training yields further gains. Ours+TTT achieves the best full-test-set mean on all three spin-glass and all three coloring instances. On GRID-20, `daoopt` reaches the optimum, while Ours+TTT remains close at both budgets. On GRID-40, `daoopt` returns incumbents for only $54/72$ of the 100 queries at $20/60$ s, so its mean reflects only queries solved within budget and is not directly comparable to full-test-set results. Ours+TTT solves all queries and achieves the best full-test-set mean. On GRID-80, neither branch-and-bound method returns a feasible solution within budget, while Ours+TTT continues improving its sampled initialization. Thus, the learned proposal provides a strong evidence-conditioned starting point for further query-specific refinement.

4.4 WARM-STARTING ANNEALED SMC

Table 4: Annealed-SMC anytime MPE quality on 100 test queries per spin-glass instance. We report mean best $\log P$ within matched $5/60$ s budgets, including warm-start overhead. All runs terminate at $\beta_{\text{end}} = 8$. Bold marks the best result per budget.

	cold (β=0)		warm β=1		warm β=2		warm β=4	
Instance	5 s	60 s	5 s	60 s	5 s	60 s	5 s	60 s
GLASS-400	257.5	**765.3**	693.4	764.5	741.2	762.0	**748.8**	758.3
GLASS-800	198.3	1149.7	1372.4	**1533.1**	1480.9	1531.3	**1493.3**	1519.4
GLASS-1200	241.8	1084.5	2022.4	2296.0	2216.7	**2298.8**	**2238.1**	2282.1

Annealed SMC (Del Moral et al., 2006) transports a particle population along a temperature ladder toward the target power, with resampling and rejuvenation throughout the schedule. A cold run starts at $\beta = 0$ and must traverse the entire ladder, including the costly low-power region. Because our curriculum already provides converged proposals at intermediate powers, we warm-start SMC from checkpoints at $\beta \in \{1, 2, 4\}$ and continue annealing from the corresponding power to $\beta_{\text{end}} = 8$. Higher starting powers shorten the remaining schedule and reach the high-power regime faster, while the cold baseline starts at $\beta = 0$.

We compare methods by the best $\log P$ reached within matched wall-clock budgets. Table 4 reports quality at 5 and 60 seconds, with full trajectories in Appendix I. Warm-starting substantially outperforms cold SMC at 5 seconds on all three instances. At 60 seconds, cold SMC slightly exceeds the warm starts on GLASS-400, while warm starts remain substantially better on GLASS-800 and GLASS-1200. Higher starting powers generally reach strong solutions faster, while lower powers can achieve better final quality, reflecting the diversity–concentration trade-off. Intermediate checkpoints therefore complement classical inference by accelerating access to the high-power regime and, on larger instances, producing stronger solutions within the same budget.

4.5 BRANCH-AND-BOUND VARIABLE ORDERING

Beyond direct MPE inference, the learned sampler can also guide exact search. For each query, we draw a batch of configurations from the sampler and estimate the per-variable conditional marginals $P(q_i \mid \mathbf{e})$ from these samples. We use these marginals as a variable-value ordering heuristic (Siddiqi & Huang, 2008) for branch-and-bound MPE in `SCIP` (Achterberg, 2009), directing branching toward the most confident variables and node selection toward the marginal argmax. We compare

Table 5: Neural variable-value ordering for branch-and-bound MPE (SCIP). Sampler-derived marginals guide branching and node selection, compared with SCIP's default ordering and L2C. We report mean best $\log P$ within $10/30/60$ s over 100 queries per instance. Bold marks the best result per instance and budget.

	10 s			30 s			60 s		
Instance	SCIP	L2C	Ours	SCIP	L2C	Ours	SCIP	L2C	Ours
GRID-20B	1444.6	1446.6	**1479.8**	1465.3	1478.8	**1481.9**	1472.9	1480.2	**1482.4**
BN-30	−4112.4	−4112.4	**−900.6**	−1174.3	−802.7	**−655.6**	−793.8	−793.5	**−637.3**
BN-45	−291.1	−292.1	**−272.5**	−287.5	−286.8	**−268.3**	−279.9	−268.7	**−267.5**

against SCIP's default ordering and L2C (Malhotra et al., 2025), a solver-specific neural branching heuristic. Following the L2C protocol (Malhotra et al., 2025), we evaluate 100 Gibbs-generated queries per instance with 25% of variables fixed as evidence. Table 5 shows that sampler-guided SCIP achieves the best mean anytime objective on all three instances at every budget. The gains are especially pronounced on BN-30, while the learned ordering also remains consistently ahead on GRID-20B and BN-45. Thus, the sampler provides useful search guidance in addition to serving as a standalone MPE solver.

5 Related Work

MPE and amortized neural inference. MPE inference is NP-hard in general (Shimony, 1994). Exact methods include variable elimination (Dechter, 1999), AND/OR branch-and-bound (Otten, 2012; Marinescu & Dechter, 2009), and soft arc consistency methods such as `toulbar2` (de Givry et al., 2006; Givry, 2023), while stochastic and guided local search provide incomplete alternatives (Park, 2002; Hutter et al., 2005). Recent neural approaches amortize MPE inference across evidence queries, including SSMP and GUIDE (Arya et al., 2024a), SINE (Arya et al., 2025), learned conditioning heuristics (Malhotra et al., 2025) and and, more broadly, neural approximations of exact inference (Agarwal et al., 2022). In constrast, our approach learns an amortized sampler for the high-β power distributions, using sampling itself as the mechanism for generating MPE candidates across evidence queries.

Neural sampling and tempering. Neural samplers for unnormalized distributions include GFlowNets (Bengio et al., 2021; 2023), stochastic-control and diffusion approaches (Zhang & Chen, 2022; Vargas et al., 2023), and Boltzmann generators (Noé et al., 2019). Masked, order-agnostic autoregressive models provide another route to discrete sampling (Austin et al., 2021; Hoogeboom et al., 2022). Tempering and annealing are classical tools for moving between broad distributions and concentrated modes, underlying simulated annealing, annealed importance sampling, and SMC (Kirkpatrick et al., 1983; Neal, 2001; Del Moral et al., 2006). Our setting differs in that the target is a family of power distributions of a known PGM, and the learned sampler is amortized across evidence queries. A single curriculum produces checkpoints across powers, with MPE as the high-β endpoint, and the learned samples can subsequently be composed with SMC and branch-and-bound inference.

6 Conclusion

We proposed amortized power sampling for repeated MPE inference on fixed PGMs. A factor-reading neural sampler is trained with reverse KL along a curriculum over β, producing high-probability configurations across unseen evidence queries without labeled MPE solutions or normalizing constants. A single sample already outperforms the evaluated neural MPE baselines, while test-time refinement further improves solution quality and enables competitive performance against branch-and-bound. The learned proposals also complement existing inference procedures by warm-starting annealed SMC and guiding branch-and-bound search.

Limitations and future work. The sampler does not faithfully cover the full posterior or power distribution, because its reverse-KL objective is mode-seeking. It also amortizes across evidence for a single fixed PGM, not across different PGM structures. Future work could improve multimodal coverage while retaining the efficiency of the current approach, and extend amortization across PGMs and additional downstream inference procedures.

AI USE STATEMENT

We used generative AI tools to edit the manuscript for clarity and readability and to assist with scientific writing and organization. We reviewed all AI-assisted material and take full responsibility for the final manuscript and its contents.

REPRODUCIBILITY STATEMENT

We provide the implementation and experimental code accompanying a README with instructions for reproducing the experiments. The appendix contains additional experimental results, ablations, implementation details, benchmark specifications, and training and evaluation protocols.

REFERENCES

Tobias Achterberg. SCIP: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

Sakshi Agarwal, Kalev Kask, Alexander Ihler, and Rina Dechter. Neurobe: Escalating neural network approximations of bucket elimination. In James Cussens and Kun Zhang (eds.), *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands*, volume 180 of *Proceedings of Machine Learning Research*, pp. 11–21. PMLR, 2022. URL https://proceedings.mlr.press/v180/agarwal22a.html.

Shivvrat Arya, Tahrima Rahman, and Vibhav Gogate. A neural network approach for efficiently answering most probable explanation queries in probabilistic models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 33538–33601. Curran Associates, Inc., 2024a. URL https://openreview.net/forum?id=ufPPf9ghzP.

Shivvrat Arya, Tahrima Rahman, and Vibhav Gogate. Neural Network Approximators for Marginal MAP in Probabilistic Circuits. volume 38, pp. 10918–10926, March 2024b. doi: 10.1609/aaai.v38i10.28966.

Shivvrat Arya, Tahrima Rahman, and Vibhav Gogate. Learning to Solve the Constrained Most Probable Explanation Task in Probabilistic Graphical Models. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238 of *Proceedings of Machine Learning Research*, pp. 2791–2799. PMLR, April 2024c. URL https://proceedings.mlr.press/v238/arya24b.html.

Shivvrat Arya, Tahrima Rahman, and Vibhav Giridhar Gogate. Sine: Scalable mpe inference for probabilistic graphical models using advanced neural embeddings. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4465–4473. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/arya25a.html.

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 (NeurIPS)*, 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 (NeurIPS)*, 2021.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. GFlowNet foundations. *Journal of Machine Learning Research*, 24, 2023.

Julian Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 24(3):179–195, 1975.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773.

Simon de Givry, Thomas Schiex, and Gérard Verfaillie. Exploiting tree decomposition and soft local consistency in weighted CSP. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI)*, pp. 22–27. AAAI Press, 2006. URL http://www.aaai.org/Library/AAAI/2006/aaai06-004.php.

Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artif. Intell.*, 113(1-2): 41–85, 1999. doi: 10.1016/S0004-3702(99)00059-4. URL https://doi.org/10.1016/S0004-3702(99)00059-4.

Rina Dechter, Alexander Ihler, Vibhav Gogate, Junkyu Lee, Bobak Pezeshki, Annie Raichev, and Nick Cohen. UAI 2022 competition, 2022. URL https://uaicompetition.github.io/uci-2022/.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B*, 68(3):411–436, 2006.

Samuel Frederick Edwards and Phil W Anderson. Theory of spin glasses. *Journal of Physics F: Metal Physics*, 5(5):965–974, 1975.

G. Elidan and A. Globerson. The 2010 UAI Approximate Inference Challenge, 2010.

Nial Friel and Anthony N. Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607, 2008.

Samuel J. Gershman and Noah D. Goodman. Amortized inference in probabilistic reasoning. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2014.

Charles J. Geyer. Markov chain Monte Carlo maximum likelihood. *Computing Science and Statistics: Proc. 23rd Symposium on the Interface*, pp. 156–163, 1991.

Simon De Givry. toulbar2, an exact cost function network solver. In *24ème édition du congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision ROADEF 2023*, Rennes, France, February 2023. URL https://hal.science/hal-04021879.

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 (ICLR)*, 2022.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E. Hopcroft, and Kilian Q. Weinberger. Snapshot ensembles: Train 1, get M for free. In *International Conference on Learning Representations (ICLR)*, 2017.

Frank Hutter, Holger H. Hoos, and Thomas Stützle. Efficient stochastic local search for MPE solving. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 169–174, 2005.

Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193.

Brij Malhotra, Shivvrat Arya, Tahrima Rahman, and Vibhav Giridhar Gogate. Learning to condition: A neural heuristic for scalable MPE inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=otIdC4tsYf.

Nikolay Malkin, Salem Lahlou, Tristan Deleu, Xu Ji, Edward Hu, Katie Everett, Dinghuai Zhang, and Yoshua Bengio. GFlowNets and variational inference. In *International Conference on Learning Representations (ICLR)*, 2023.

Radu Marinescu and Rina Dechter. AND/OR branch-and-bound search for combinatorial optimization in graphical models. *Artif. Intell.*, 173(16-17):1457–1491, 2009. doi: 10.1016/J.ARTINT.2009.07.003. URL https://doi.org/10.1016/j.artint.2009.07.003.

Robert Mateescu, Kalev Kask, Vibhav Gogate, and Rina Dechter. Join-graph propagation algorithms. *Journal of Artificial Intelligence Research*, 37:279–328, 2010. doi: 10.1613/jair.2842. URL https://doi.org/10.1613/jair.2842.

Marc Mézard and Andrea Montanari. *Information, Physics, and Computation*. Oxford University Press, 2009.

Thomas P. Minka. Divergence measures and message passing. Technical Report MSR-TR-2005-173, Microsoft Research, 2005.

Radford M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

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

Lars Otten. DAOOPT: Sequential and distributed AND/OR branch and bound for MPE problems. GitHub, https://github.com/lotten/daoopt, 2012.

James D. Park. Using weighted MAX-SAT engines to solve MPE. In Rina Dechter, Michael J. Kearns, and Richard S. Sutton (eds.), *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada*, pp. 682–687. AAAI Press / The MIT Press, 2002. URL http://www.aaai.org/Library/AAAI/2002/aaai02-102.php.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, 1998.

Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1–2):273–302, 1996.

Solomon Eyal Shimony. Finding maps for belief networks is np-hard. *Artificial Intelligence*, 68(2): 399–410, 1994.

Sajjad Siddiqi and Jinbo Huang. Variable and Value Ordering for MPE search. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI)*, Arlington, Virginia, USA, 2008. AUAI Press.

Francisco Vargas, Will Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *International Conference on Learning Representations (ICLR)*, 2023.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

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

A DESCRIPTION OF THE PGM INSTANCES USED IN THE BENCHMARK

Table 6 summarizes all PGM instances used in our experiments, including the aliases used in the paper, their source names, and their sizes in terms of variables and factors. The grid and graph-coloring (WCSP) instances come from the UAI inference competition benchmarks (Elidan & Globerson, 2010; Dechter et al., 2022). The spin-glass instances are synthetically generated on 6-regular graphs with q=3 states per variable and no external field. The first nine instances constitute the main benchmark in Sec. 4, the next three (GRID-20B, BN-30, and BN-45) are used in the branch-and-bound variable-value ordering experiment (Table 5), and the final two are auxiliary instances. GLASS-12 is a small, exactly enumerable spin glass used to validate the fidelity metrics of Sec. 4.1, and GRID-BN is a grid-structured Bayesian network used for the posterior-sampling experiment (App. E).

Table 6: Benchmark PGM instances: alias, source name, family, and size.

Alias	Actual name	Family	#Variables	#Factors
GRID-20	`grid20x20.f2`	Grid	400	1,160
GRID-40	`grid40x40.f2.wrap`	Grid	1,600	4,800
GRID-80	`grid80x80.f2.wrap`	Grid	6,400	19,200
GLASS-400	`glass_reg6_n400_q3_f0`	Spin glass (6-regular)	400	1,600
GLASS-800	`glass_reg6_n800_q3_f0`	Spin glass (6-regular)	800	3,200
GLASS-1200	`glass_reg6_n1200_q3_f0`	Spin glass (6-regular)	1,200	4,800
LE450-2	`le450_5a_2`	Coloring (WCSP)	450	5,714
LE450-3	`le450_5a_3`	Coloring (WCSP)	450	5,714
LE450-4	`le450_5a_4`	Coloring (WCSP)	450	5,714
GRID-20B	`grid20x20.f5.wrap`	Grid	400	1,200
BN-30	`BN_30`	Bayesian network	1,156	1,156
BN-45	`BN_45`	Bayesian network	880	880
GLASS-12	`glass_small_n12_q3`	Spin glass (6-regular)	12	36
GRID-BN	`bn_grid40`	Bayesian network (grid)	1,600	1,600

Generating the synthetic spin-glass instances. The spin-glass instances are generated on random d-regular graphs. Let $\mathbf{X} = \{X_1, \ldots, X_n\}$ denote the n random variables and $G = (V, E)$ their pairwise interaction graph. We use d=6, giving $|E| = dn/2$ pairwise factors, namely 1,200, 2,400, and 3,600 for GLASS-400, GLASS-800, and GLASS-1200, respectively. Each variable X_i takes q=3 states. For every edge $(u, v) \in E$, the log-potential is

$$\phi_{uv}(x_u, x_v) = \begin{cases} J_{uv}, & x_u = x_v, \\ 0, & x_u \neq x_v, \end{cases}$$

where $J_{uv} = \varepsilon_{uv}|g_{uv}|$, with $\varepsilon_{uv} \in \{-1, +1\}$ drawn uniformly and $g_{uv} \sim \mathcal{N}(0, 1)$. Positive couplings therefore favor agreement, while negative couplings favor disagreement, producing a frustrated spin-glass landscape (Edwards & Anderson, 1975; Mézard & Montanari, 2009). We use no external field (the "`f0`" suffix), so the n unary factors are uniform and all nontrivial structure comes from the pairwise couplings. The resulting PGM is therefore defined by

$$\widetilde{P}(\mathbf{x}) = \exp\left(\sum_{(u,v)\in E} \phi_{uv}(x_u, x_v)\right),$$

with strictly positive factor values.

B NEURAL SAMPLER ARCHITECTURE

The masked-reveal proposal $q_\theta(\mathbf{q} \mid \mathbf{e})$ is a factor-reading message-passing network that maps the current partial assignment to per-variable categorical logits at each reveal step (Sec. 3). We use two closely related backbones that share a common skeleton and differ only in how a factor's potential

table reaches its variables, a general *any-arity factor-graph* backbone that reads factors of arbitrary scope, and a lighter *pairwise* backbone specialized to models with only unary and pairwise factors. The any-arity backbone subsumes the pairwise one, since on a scope of size two they compute messages of the same form, but the two are not bit-identical, as detailed below.

Notation and shared skeleton. Let V be the n variables, K_v the domain size of variable v, and $K = \max_v K_v$. At a given reveal step, variable v carries a token $x_v \in \{1, \dots, K_v\} \cup \{\textsc{mask}\}$ (its revealed value or the mask symbol) with one-hot encoding $o(x_v) \in \{0,1\}^{K+1}$, and an evidence indicator $e_v \in \{0,1\}$. Both backbones maintain hidden variable states $h_v^{(l)} \in \mathbb{R}^H$ over L layers, condition on the reveal progress $t \in [0,1]$ through a small MLP $c = \mathrm{MLP}_{\mathrm{cond}}(\rho)$, and share the same per-layer update and output head. Given the aggregated message $a_v^{(l)}$ produced by a layer (defined per backbone below), the update is a FiLM-modulated residual block

$$u_v^{(l)} = \mathrm{MLP}_{\mathrm{upd}}^{(l)}\big([\, h_v^{(l)} \,;\, a_v^{(l)} \,]\big), \qquad (\gamma^{(l)}, \delta^{(l)}) = W_{\mathrm{FiLM}}^{(l)}\, c,$$

$$h_v^{(l+1)} = \mathrm{LayerNorm}\Big(h_v^{(l)} + (1+\gamma^{(l)}) \odot u_v^{(l)} + \delta^{(l)}\Big),$$

and a shared head maps the final state to logits, masking values outside a variable's domain,

$$z_v = \mathrm{MLP}_{\mathrm{head}}\big(h_v^{(L)}\big) \in \mathbb{R}^K, \qquad z_{v,k} \leftarrow -\infty \ \text{ for } k > K_v.$$

The categorical distribution used at a reveal step is the softmax of z_v over the variables revealed at that step. All weights are independent of n, so a single trained network applies to every evidence query on the fixed PGM. Unless noted, L=6 and H=64.

Any-arity factor-graph backbone (default). This backbone operates on the bipartite factor graph, adopting the hypergraph representation of the PGM introduced by SINE (Arya et al., 2025), in which each factor is a hyperedge whose feature is its numerical potential table (with the same zero-padding and per-variable domain masking used to support factors of mixed arity and variables of mixed cardinality). Let $\{f\}$ index the factors, $\mathrm{sc}(f) \subseteq V$ the scope (variables) of factor f, and $\theta_f \in \mathbb{R}^T$ its (zero-padded) raw potential table, where T is the largest table across factors. Variable states are initialized from the token and evidence bit only,

$$h_v^{(0)} = W_{\mathrm{enc}}\big[\, o(x_v) \,;\, e_v \,\big],$$

and each factor is embedded once from a *learned* linear encoding of its potential table, $g_f = W_{\mathrm{tab}}\,\theta_f + b_{\mathrm{tab}} \in \mathbb{R}^H$. Unary potentials are represented as arity-one factors and therefore enter through the same factor pathway. Each layer sends a *leave-one-out* message from every factor to each variable in its scope. Writing $s_f^{(l)} = \sum_{u \in \mathrm{sc}(f)} h_u^{(l)}$ for the scope sum, the message and its aggregation are

$$m_{f\to v}^{(l)} = \mathrm{MLP}_{\mathrm{msg}}^{(l)}\Big(\big[\, h_v^{(l)} \,;\, \underbrace{s_f^{(l)} - h_v^{(l)}}_{\sum_{u \in \mathrm{sc}(f)\setminus\{v\}} h_u^{(l)}} \,;\, g_f \,\big]\Big), \qquad a_v^{(l)} = \frac{1}{\deg(v)} \sum_{f:\, v \in \mathrm{sc}(f)} m_{f\to v}^{(l)},$$

where $\deg(v)$ is the number of factors incident to v. The message to v thus depends on the factor's potential embedding and the pooled states of the *other* variables in its scope, letting a single layer read a factor of any arity. The conditioning input is the reveal progress only, $\rho = [\, t \,]$.

Pairwise backbone. For models whose factors are all unary or pairwise, we use a lighter variable-only graph in which each pairwise factor $\{u, v\}$ contributes directed edges $u \to v$ and $v \to u$. Here the unary potential $\phi_v \in \mathbb{R}^K$ is concatenated directly into the node input (rescaled by the global potential scale σ),

$$h_v^{(0)} = W_{\mathrm{enc}}\big[\, o(x_v) \,;\, \phi_v/\sigma \,;\, e_v \,\big],$$

and the pairwise potential table $\Psi_{uv} \in \mathbb{R}^{K\times K}$ is passed *directly* (flattened and rescaled) into the edge message rather than being pre-encoded. The layer message and aggregation over the incoming edges of v (degree d_v) are

$$m_{u\to v}^{(l)} = \mathrm{MLP}_{\mathrm{msg}}^{(l)}\Big(\big[\, h_v^{(l)} \,;\, h_u^{(l)} \,;\, \mathrm{vec}(\Psi_{uv})/\sigma \,\big]\Big), \qquad a_v^{(l)} = \frac{1}{d_v} \sum_{u:\,(u\to v)} m_{u\to v}^{(l)}.$$

The conditioning input is the reveal progress only, $\rho = [\, t \,]$, as in the any-arity backbone.

Relationship between the two backbones. On a pairwise factor $\{u, v\}$ the leave-one-out context of the any-arity backbone reduces to the single neighbor, $s_f^{(l)} - h_v^{(l)} = h_u^{(l)}$, so its message $\mathrm{MLP}_{\mathrm{msg}}([h_v^{(l)}; h_u^{(l)}; g_f])$ has exactly the functional form of the pairwise edge message. The two implementations nonetheless differ in two concrete ways. First, the any-arity backbone uses a *learned* table embedding g_f, whereas the pairwise backbone feeds the raw table $\mathrm{vec}(\Psi_{uv})$. Second, unary potentials are routed as arity-one factors in the any-arity backbone but concatenated into the node feature in the pairwise backbone. Both have parameter counts independent of n, and the per-step message tensors scale with $\sum_f |\mathrm{sc}(f)|$ for the any-arity backbone and with the number of directed edges for the pairwise backbone.

Relation to SINE. Our any-arity backbone shares its *input representation* with SINE (Arya et al., 2025), the same hypergraph in which factors are hyperedges carrying their potential tables as features, but differs in both the encoder and the decoder. SINE encodes with a single layer of hypergraph *attention*, in which each variable attends over its incident factor-hyperedges, and decodes with a *global readout* that concatenates all variable embeddings and predicts every variable's value jointly in one shot. Our backbone instead uses several rounds of *leave-one-out message passing*, so that each factor propagates the states of the other variables in its scope, and decodes with a *shared per-variable head* applied within the iterative masked-reveal process. These differences follow from the different goals. SINE is a discriminative one-shot MPE predictor, whereas our network parameterizes a masked-reveal *sampler* trained data-free with the reverse-KL objective of Sec. 3.

C Ablations

We ablate the four design choices behind the sampler, the factor-reading architecture, the autoregressive masked-reveal proposal, the annealing curriculum, and the saving of a checkpoint ladder from a single run. We run the first two on both the frustrated GLASS-400 spin glass and, as a control, the GRID-20 lattice. The curriculum ablation uses an enumerable $n{=}12$ companion glass where the exact P^β is available. Unless noted, we report best-of-K $\log P$ ($K{\in}\{1, 8, 32, 64\}$) at the deployed power $\beta{=}4$ over the 100 test queries (higher is better), together with the cross-sample diversity (mean per-site Gini $1 - \sum_v p_v^2$ over the drawn samples), which exposes whether a proposal has collapsed.

Reading and propagating the factor potentials is essential. We compare three backbones trained with an identical curriculum and evidence protocol, differing only in how much of the PGM they read, a structure-blind MLP over raw variable tokens, a node-wise *1-hop factor reader* that uses each variable's state together with the numerical potentials of its incident factors but performs no message passing, and our multi-hop factor-graph GNN that propagates potential information across the graph. Table 7 shows a clear ordering. The structure-blind MLP performs worst despite its greater capacity. The 1-hop reader produces more diverse samples, but its predictions are effectively local, since each variable sees only its incident factors and cannot incorporate information propagated from more distant parts of the graph. As a result, its independently chosen values are poorly coordinated across coupled variables and its samples remain low-scoring. Propagating factor information across the graph allows our model to coordinate these local decisions, producing samples that are both concentrated on high-probability configurations and retain useful diversity.

Table 7: Factor-reading ablation: best-of-K $\log P$ at $\beta{=}4$ over the 100 test queries, with cross-sample diversity ("div") over the 128 draws. Same curriculum and evidence protocol.

Instance	Backbone	$K{=}1$	$K{=}8$	$K{=}32$	$K{=}64$	div
GLASS-400	MLP (structure-blind, 9.6M)	394.0	394.0	394.0	394.0	0.000
	1-hop factor reader (0.09M)	553.0	575.9	586.1	590.4	0.528
	Multi-hop GNN (ours, 0.24M)	732.8	740.1	743.2	**744.4**	0.192
GRID-20	MLP (structure-blind, 9.6M)	584.0	585.8	586.5	586.8	0.013
	1-hop factor reader (0.09M)	488.9	506.2	513.8	516.9	0.165
	Multi-hop GNN (ours, 0.24M)	618.3	620.4	621.1	**621.4**	0.030

The autoregressive reveal beats a mean-field proposal. We compare our masked-reveal autoregressive sampler with a *mean-field* proposal using the identical backbone and evidence protocol, but with a single reveal step, so that all query variables are sampled independently given the evidence. Table 8 shows that the mean-field proposal collapses under the mode-seeking objective. Without autoregressive conditioning on previously revealed variables, each site's prediction is optimized largely independently, preventing the proposal from representing the dependencies and coordinated assignments induced by the PGM factors. The resulting samples have very limited diversity, and additional draws provide little benefit. In contrast, the masked-reveal sampler conditions each successive prediction on the partially revealed configuration, allowing it to model dependencies among variables, maintain useful diversity, and reach substantially higher $\log P$.

Table 8: Masked-reveal vs. mean-field proposal: best-of-K $\log P$ at β=4, same backbone with S=20 vs. S=1.

Instance	Proposal	K=1	K=8	K=32	K=64	div
GLASS-400	Mean-field (S=1, factorized)	652.8	660.0	663.0	664.3	0.041
	Masked-reveal (S=20, ours)	732.8	740.1	743.2	**744.4**	0.192
GRID-20	Mean-field (S=1, factorized)	588.9	591.1	591.8	592.1	0.017
	Masked-reveal (S=20, ours)	618.3	620.4	621.1	**621.4**	0.030

A power curriculum improves high-β training. On the enumerable n=12 companion glass, where the exact power distribution P^β is available, we compare three training regimes, our annealed curriculum, direct training at a fixed high power, and training with a randomly sampled power at each step. We evaluate each regime using the exact $\mathrm{KL}(q_\theta \| P^\beta)$, with multiple random seeds (Table 9, Fig. 4). The annealed curriculum consistently provides the best fit in the power range used for MPE, while direct training at high power is less effective and random-β training degrades as the power increases. These results show that progressively sharpening the target allows the model to reach and fit the high-β regime more effectively than training directly on the peaked target or distributing capacity across powers. Because the target is exactly enumerable here, this experiment provides a direct validation of the curriculum before we use approximate parallel-tempering references on larger instances.

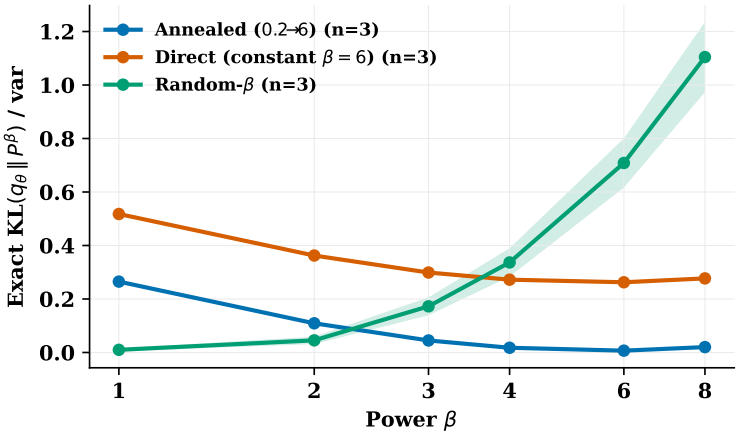


Figure 4: Curriculum ablation on the enumerable n=12 glass: exact $\mathrm{KL}(q_\theta \,\|\, P^\beta)$ versus β for the three training regimes (line = mean over 3 seeds, band = $\pm$1 std).

Table 9: Exact $\mathrm{KL}(q_\theta \,\|\, P^\beta)$ (lower is better) on the enumerable n=12 glass for the three training regimes (mean over 3 seeds, std $\leq$ 0.13). Bold marks the best regime at each power.

Regime	β=1	β=2	β=3	β=4	β=6	β=8
Annealed (0.2→6)	0.27	0.11	**0.04**	**0.02**	**0.01**	**0.02**
Direct (cold β=6)	0.52	0.36	0.30	0.27	0.26	0.28
Random-β	**0.01**	**0.05**	0.17	0.34	0.71	1.10

A single curriculum yields the full power ladder. We verify that intermediate checkpoints from the annealed curriculum retain the quality of models trained directly to the corresponding power.

At the deployed power $\beta{=}4$, we compare the staircase checkpoint with a budget-matched model trained directly to $\beta{=}4$ on GLASS-400 and GRID-20. Table 10 shows comparable solution quality and diversity across sampling budgets, with no consistent advantage for either training regime. Thus, a single annealed run can provide checkpoints across the power ladder without a measurable quality penalty relative to training separate models for individual powers.

Table 10: Ladder-checkpoint vs. dedicated-run ablation on GLASS-400 and GRID-20 at the deployed power $\beta{=}4$ (best-of-K $\log P$ over the 100 test queries, with cross-sample diversity "div"). The dedicated run is trained directly to $\beta{=}4$ and budget-matched to the staircase.

Instance	$\beta{=}4$ model	$K{=}1$	$K{=}8$	$K{=}32$	$K{=}64$	div
GLASS-400	Ladder checkpoint (staircase)	732.8	740.1	743.2	**744.4**	0.192
	Dedicated (direct to $\beta{=}4$)	729.4	737.8	741.3	742.8	0.216
GRID-20	Ladder checkpoint (staircase)	618.3	620.4	621.1	621.4	0.030
	Dedicated (direct to $\beta{=}4$)	618.4	621.0	621.7	**622.0**	0.034

D PARALLEL-TEMPERING REFERENCE

The cross-CLL and order-fidelity analyses (Fig. 2) compare each checkpoint against reference samples from the conditioned power distribution $P^{\beta}(\cdot \mid \mathbf{e})$. On GLASS-12 these references are exact (full enumeration). On the larger instances we approximate them by evidence-conditioned *parallel tempering* (replica-exchange MCMC) (Geyer, 1991), which runs a ladder of Gibbs chains at increasing powers and periodically proposes Metropolis swaps between adjacent rungs, letting chains cross energy barriers that a single high-β chain would not. Per query we hold the evidence fixed and run 24 replicas spanning $\beta \in [0.5, 10]$ with 64 independent systems, for 120 Gibbs sweeps with a replica-exchange swap attempted every 8 sweeps, and then read off the population at each target power $\beta \in \{1, 2, 4, 6\}$.

E POSTERIOR SAMPLING EXPERIMENT

The posterior-sampling evaluation of Sec. 4.1 uses GRID-BN, a 40×40 grid-structured Bayesian network with 1,600 binary variables (Table 6). We use a directed model here because our importance-sampling baseline, likelihood weighting (Koller & Friedman, 2009), samples from a normalized generative model and reweights by the evidence likelihood, which the undirected Markov-random-field and WCSP instances of the main benchmark do not provide directly.

Scoring marginal accuracy requires exact posterior marginals, which are intractable on the full grid because its treewidth grows with the grid width. We therefore sample evidence along a *column cutset*, fixing for each query every few columns of the grid to values drawn from the model, chosen so that conditioning on them decomposes the grid into independent narrow strips of bounded treewidth. Exact marginals of the remaining free variables are then obtained with iterative join-graph propagation (Mateescu et al., 2010). We generate 100 such queries and, for each sample budget K, report the mean ℓ_1 distance between the estimated and exact free-variable marginals, the wall-clock time, and the importance-weight ESS, comparing the learned $\beta{=}1$ checkpoint (used unweighted) against likelihood weighting (Fig. 3).

F ILLUSTRATIVE IMAGE-COMPLETION INSTANCE

Figure 1 uses a visually interpretable PGM that we fit to the Fashion-MNIST dataset (Xiao et al., 2017). We stress that this instance is included only as a qualitative illustration. The attractive potentials that make a clean image also make this instance easy for exact solvers. It nonetheless offers an intuitive picture of what amortized power sampling does, because here an MPE assignment is literally a completed image.

Fitting the PGM. For a single Fashion-MNIST class (we use *Trouser* and *Ankle-boot*), we binarize each 28×28 image at mid-intensity to spins $x \in \{-1, +1\}^{784}$ and fit a pairwise Ising grid MRF by maximum pseudo-likelihood (Besag, 1975), learning a per-pixel field h_i and nearest-neighbour

couplings J_{ij} (horizontal and vertical) with Adam. Because the fields encode the class silhouette and the couplings encode local smoothness, real garments become high-probability configurations of the learned model, so image completion is a *self-consistent* MPE task rather than denoising under a hand-specified prior. The fitted model consists of 784 binary variables and 2296 factors (784 unary, 1512 pairwise).

Training the sampler. We train a single sampler on this fixed PGM exactly as in the main experiments. Each evidence query hides a random fraction of the pixels of a training-pool image and reveals the rest, so the sampler amortizes image completion over both the image and the mask, and test queries use held-out images. The completions in the method comparison (Fig. 6) are best-of-K samples (K=64) from the trained model, while the power sweep (Fig. 5) shows a single typical sample per β.

Results. Figure 5 shows completions across the power range. At low β the samples are broad and noisy, and increasing β concentrates them onto a clean, high-probability completion. Quality peaks near β=4, the same behaviour as the grid benchmarks. Figure 6 compares our sampler against the neural MPE baselines on the same held-out images and masks. All baselines are trained on the same 20–45%-observed evidence distribution as our sampler. Visually, our sampler recovers the cleanest, most coherent garments of all methods, matching its lead on the harder benchmark PGMs.

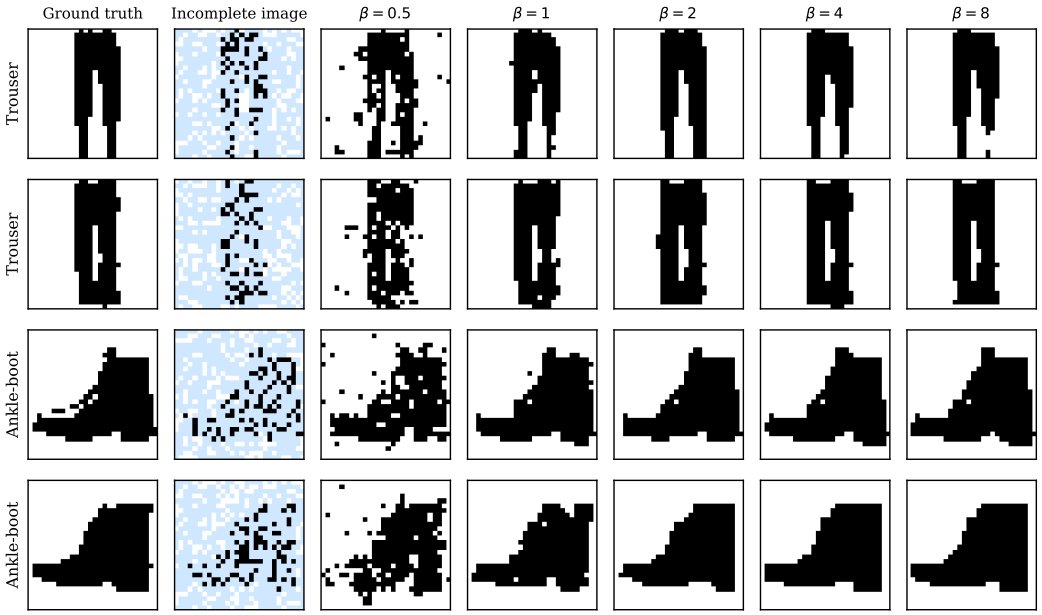


Figure 5: Completions across the power range on the learned Fashion-MNIST MRFs (30% of pixels observed, one sample per β).

G Training curves and amortization cost

Figure 7 shows the training trajectory of the deployed staircase run on each of the nine benchmark instances, namely the best $\log P$ of the on-policy training batch (train) and of a fixed held-out evidence set (val) at each step, with the power β curriculum on the right axis. Two things are visible. First, training converges quickly, reaching most of the final quality within the first β rungs, and then improves gradually as the curriculum raises β. Second, the train and held-out curves track each other almost exactly, because the objective is data-free and amortized over evidence, there is no ground-truth label to overfit, and the sampler generalizes to unseen evidence configurations of the same PGM. The coloring instances are visibly noisier, reflecting the higher variance of the REINFORCE gradient on those energies, but still converge.

Figure 8 shows the same for the three instances used in the branch-and-bound variable-value ordering experiment (Table 5). The two Bayesian networks are trained with the *any-arity* hypergraph

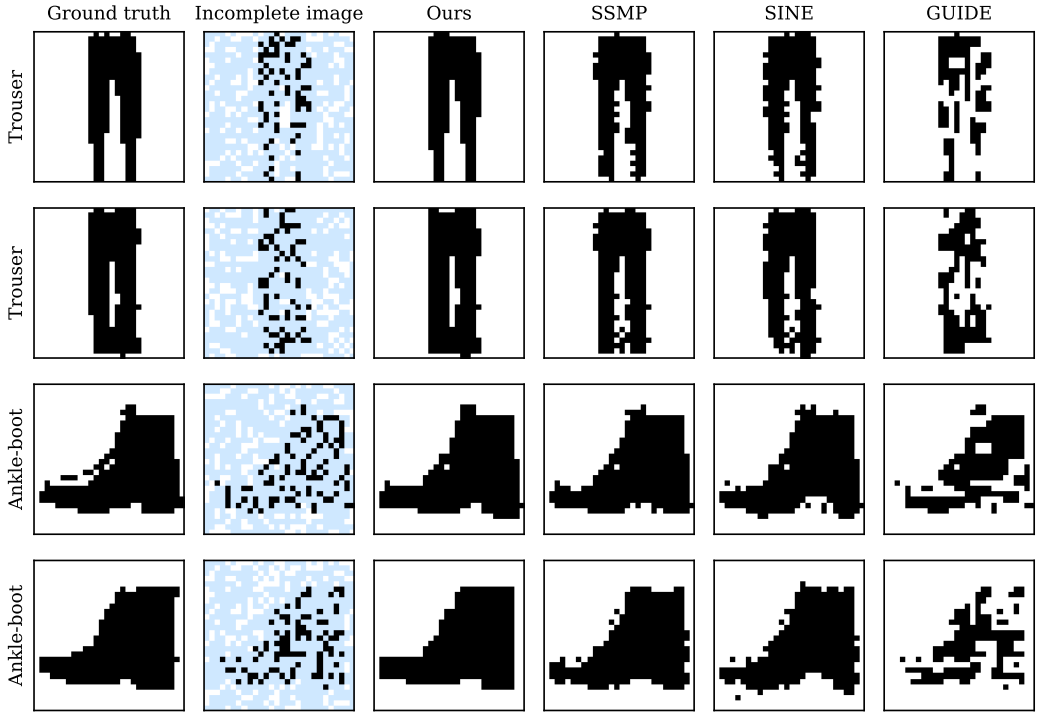


Figure 6: Our amortized sampler versus the neural MPE baselines on the same held-out images and masks (30% observed), all trained on the same 20–45%-observed evidence distribution.

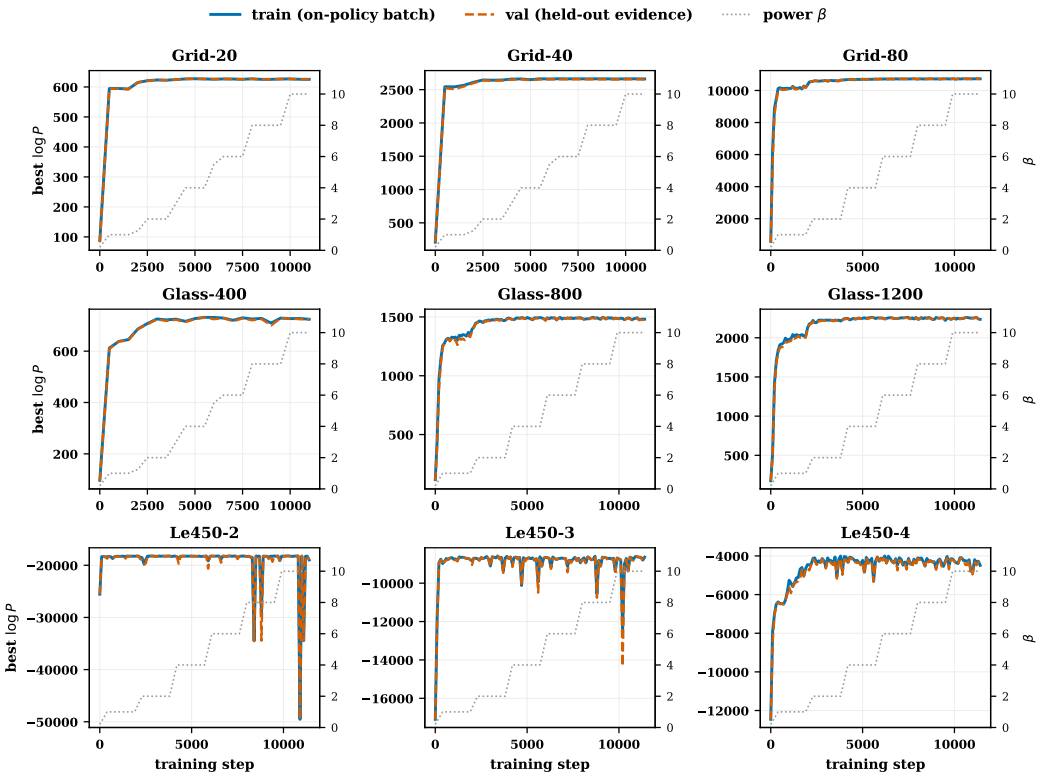


Figure 7: Training curves on the nine benchmark instances (rows: grids, spin glasses, coloring). Best $\log P$ versus training step for the training batch (train) and a held-out evidence set (val), with the β staircase on the right axis.

backbone (Appendix B) rather than the pairwise one, since their factors have arity greater than two, and their curves confirm that the general factor-reading architecture also trains stably and converges under the same curriculum.

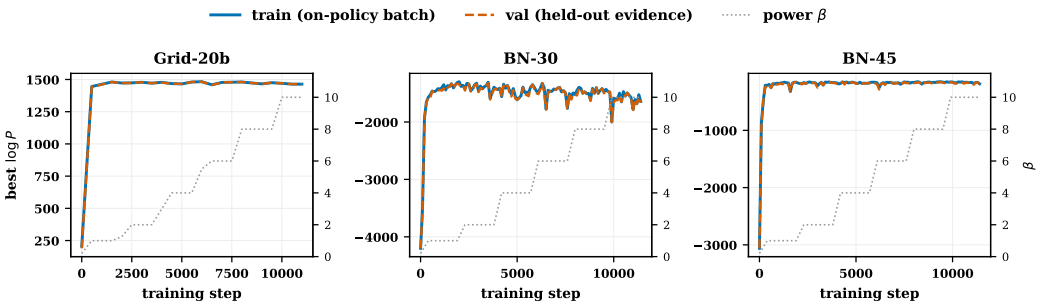


Figure 8: Training curves for the variable-value ordering instances (GRID-20B pairwise backbone; BN-30 and BN-45 any-arity backbone). Axes as in Fig. 7.

Amortization pays off over a deployment lifetime. The amortization benefit becomes clear when comparing the one-time training cost with the per-query cost of annealed SMC. The sampler requires a one-time training cost of 9.5–19 hours across the three spin-glass instances, after which a single-shot (best-of-1) query takes only 0.04–0.08 seconds. To reach that same per-query solution from scratch, cold SMC needs 24 seconds on GLASS-400 and 98 and 183 seconds on GLASS-800 and GLASS-1200 (Table 11), hundreds to thousands of times the sampler's per-query cost. The resulting break-even query count, beyond which the amortized sampler is cheaper in total compute, is approximately 1400 on GLASS-400 and 580 and 375 on GLASS-800 and GLASS-1200. This break-even query count is negligible compared to combinatorially large query space typical of applications such as medical diagnosis for large number of patients or automated planning in a fixed environment. Thus, once a PGM receives a few hundred to about 1400 queries, the one-time training cost is amortized against repeated per-query SMC computation. Moreover, the trained sampler can be reused beyond one-shot MPE, providing warm starts for annealed SMC and variable-value ordering guidance for branch-and-bound search.

Table 11: Amortization cost against annealed SMC on the three spin-glass instances. We report the sampler's single-shot (best-of-1) solution, the wall-clock time cold SMC needs to reach that same per-query solution, the one-time training cost, and the break-even query count $Q^\star = T_{\text{train}}/(t_{\text{SMC}} - t_{\text{neural}})$.

Instance	Sampler best-of-1 $\log P$ (time)	Cold-SMC time to reach it	Train (once)	$Q^\star$
GLASS-400	733.6 (0.08 s)	24.3 s	9.5 h	≈ 1400
GLASS-800	1471.3 (0.04 s)	98.1 s	15.8 h	≈ 580
GLASS-1200	2221.1 (0.04 s)	183.1 s	19.0 h	≈ 375

H PER-QUERY HEAD-TO-HEAD

The main results report the mean best $\log P$ over the 100 test queries (Sec. 4). To complement these averages, Figure 9 reports paired per-query wins, where each query is awarded to the solver with the highest $\log P$. The left figure compares a single sample (K=1) from our deployed checkpoint against the one-shot neural predictors in Table 2, while the right compares our test-time-trained sampler against the five anytime baselines in Table 3 at the 60-second budget. A solver that returns no feasible solution receives no win.

In the single-shot comparison, a single sample from our sampler wins all 100 test queries on all nine instances. With test-time training, our sampler wins every query on all three spin-glass instances, all three coloring instances, and the largest grid, while branch-and-bound dominates on the smaller grids where it can perform more complete search and `daoopt` can prove optimality. The neural baselines refined with ITSELF win no query on any instance. This shows the same instance-dependent behavior observed in Sec. 4.3 at the individual-query level.

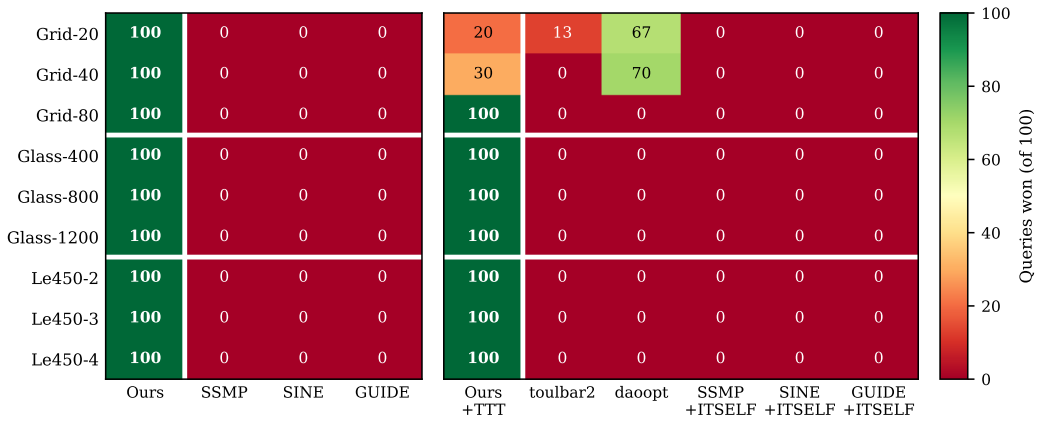


Figure 9: Per-query head-to-head: each of the 100 test queries is won by the solver reaching the highest $\log P$, and each cell counts the queries won by that solver, so every row sums to 100. Color runs from red (0) to green (100). *Left:* a single sample (K=1) from our sampler against the one-shot neural predictors of Table 2. *Right:* Ours+TTT against the five anytime baselines of Table 3 at the 60-second budget.

I ANYTIME SMC CURVES

Figure 10 shows the full anytime trajectories underlying the SMC comparison in Sec. 4 (Table 4). Each curve reports the mean best $\log P$ reached so far over the 100 test queries as a function of end-to-end wall-clock time. Warm-started runs initialize the 512-particle population from converged checkpoints at $\beta \in \{1, 2, 4\}$. The trajectories show the speed advantage of warm-starting, with strong solutions reached within seconds, while cold SMC spends substantially longer traversing the low-power portion of the schedule. Differences in final quality among the warm starts are smaller and are more clearly reflected in Table 4.

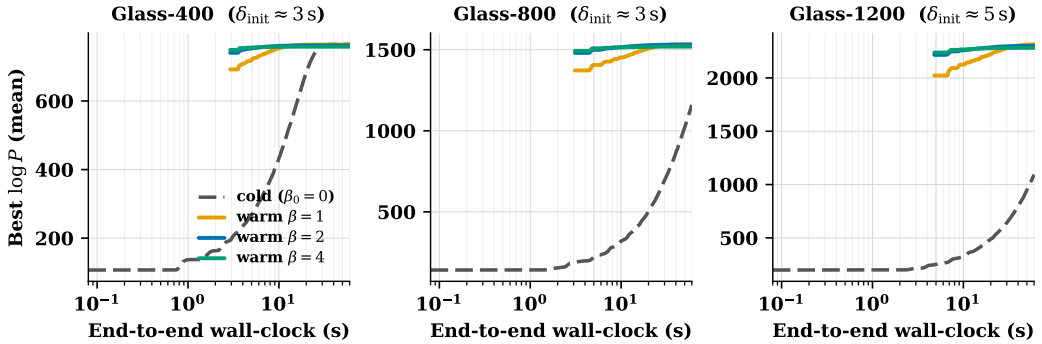


Figure 10: Anytime annealed-SMC MPE quality on the three glass instances (companion to Table 4): mean best $\log P$ versus end-to-end wall-clock time (log scale) over the 100 test queries. Cold SMC anneals from β_0=0. Warm starts begin from the converged $\beta \in \{1, 2, 4\}$ checkpoints, each after its warm-init cost δ_{init} (annotated per panel). The grey line marks the 5 s budget.

J ANYTIME CURVES FOR TEST-TIME TRAINING

Figure 11 provides the anytime view corresponding to Table 3. For each benchmark instance, it plots the mean best $\log P$ over the 100 test queries against end-to-end wall-clock time at 9 checkpoints, $\{1, 2, 5, 10, 15, 20, 30, 45, 60\}$ s. We show full trajectories for Ours+TTT and the neural baselines SSMP+ITSELF, SINE+ITSELF, and GUIDE+ITSELF, and discrete markers for `toulbar2` and `daoopt` at 20 and 60 s. A marker is omitted when no feasible solution is found within the budget. Ours+TTT consistently remains above the neural baselines and continues improving throughout the budget, while the ITSELF-refined baselines plateau early. Branch-and-bound is competitive on the smaller grid instances but fails to return solutions within the budget on the larger and more difficult

instances. Most of the advantage of Ours+TTT appears within the first few seconds, with further gains from continued test-time training.

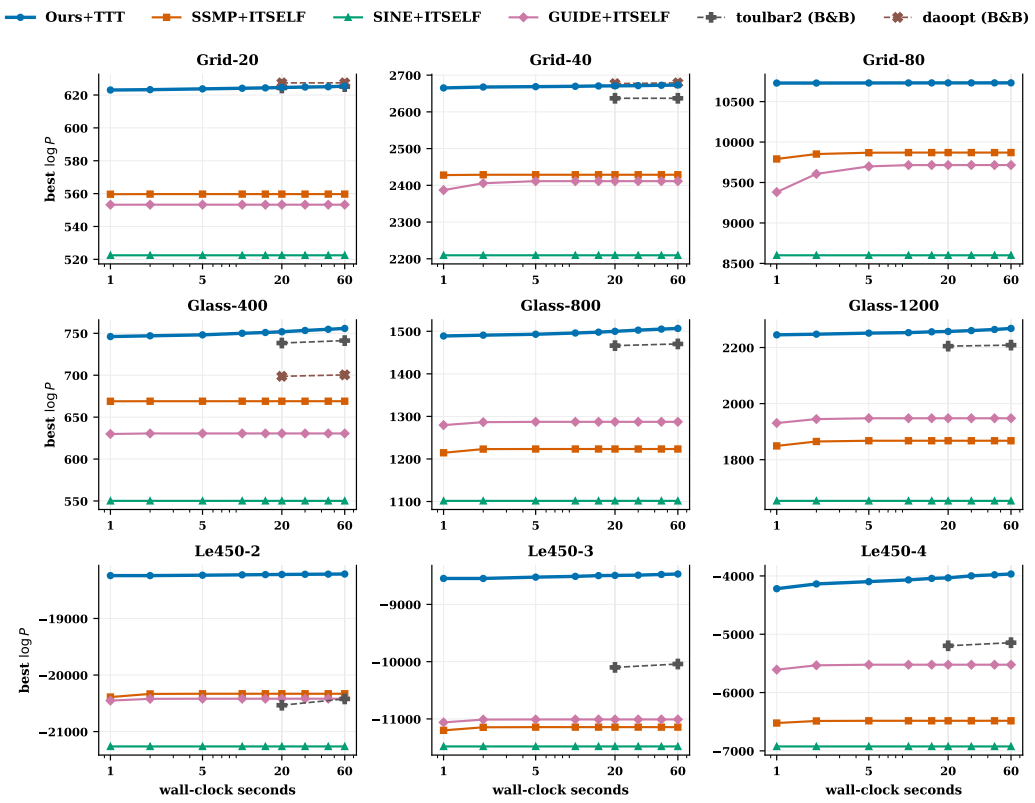


Figure 11: Anytime test-time-training curves on the nine benchmark instances (rows: grids, spin glasses, coloring): mean best $\log P$ over the 100 test queries versus wall-clock time (log scale). Neural methods (Ours+TTT, SSMP/SINE/GUIDE+ITSELF) are full curves. `toulbar2` and `daoopt` are B&B markers at 20 and 60 s, absent where no solution is returned in budget.