IMPROVING MULTIMODAL MCMC WITH LEARNED PROPOSAL MIXTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Local Markov chain Monte Carlo (MCMC) samplers can remain trapped in one mode. We train a shared normalizing flow to propose states from different regions across targets that share the state's conditional distribution given a known statistic. We study a reversible rule that controls how often the sampler switches proposal components, with a Metropolis–Hastings (MH) correction. More frequent switching can help cross modes, but can also increase rejection. For non-atomic targets, we identify when switching cannot increase covariance between successive values of any square-integrable observable. This holds exactly when the target-to-mixture density ratio has the same distribution under every component. Acceptance is then unchanged, and long-run estimation variance cannot increase. We also bound the additional variance of bounded observables caused by component and weight errors, without assuming a spectral gap. Across four asymmetric inverse designs at $d = 200$, proposal mixtures double the per-cycle efficiency of matched target-only flows for estimating mode probabilities; switching adds a further 13%. On nonzero-field lattices, mixtures pass all 40 mixing checks, compared with 8/40 for target-only flows. Corrected flips and parallel tempering remain more efficient. Code is available at https://github.com/Alphachain123/Improving_multimodal_MCMC_with_learned_proposal_mixtures.

1 INTRODUCTION

Local Markov chain Monte Carlo (MCMC) updates can explore one mode of a distribution while rarely reaching another. A learned proposal generates a candidate state that may lie in a distant mode; a Metropolis–Hastings (MH) acceptance step corrects its fitting errors. Combining these global moves with Langevin or Hamiltonian updates can improve exploration (Tran et al., 2016; Hackett et al., 2025). We study how to choose among several learned proposals when the modes have unequal probabilities and different shapes.

We consider target distributions π_y over $x \in \mathbb{R}^d$, where y is an observation or model parameter. Changing y changes the distribution of a known statistic $U(x)$ but leaves the conditional distribution of x given $U(x)$ unchanged. In the inverse problem $y = Ax + \xi$, for example, A is a known observation matrix and ξ is independent noise. Once Ax is fixed, the likelihood is the same for all compatible states, so their relative posterior probabilities depend only on the prior (Stuart, 2010). Samples from different observations can therefore help learn the same conditional structure. Exponential-family models provide another example (Wainwright & Jordan, 2008).

We divide state space into K regions, or *blocks*, with label $C(x) \in \{1, \dots, K\}$. A single normalizing flow $r_\theta(x \mid a, y)$ approximates the distribution within each block, using the block label a and target condition y as inputs. The flow is an invertible neural model that generates states and evaluates their densities. We train it on exact posterior draws in our controlled inverse problems and parallel-tempering samples in our lattice experiments. The same partition and trained network are then used at new values of y (Section 4).

For a fixed target, abbreviate $r_a = r_\theta(\cdot \mid a, y)$ and form $q = \sum_{a=1}^{K} v_a r_a$, where $v_a > 0$ and $\sum_a v_a = 1$. Independent mixing chooses component a with probability v_a. Our switching rule first assigns the current state to a component probabilistically, then favors proposing from a different component (Figure 1). This encourages movement between modes. Favoring low-probability modes

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.

can also increase rejection. Fitting the mixture weights addresses this imbalance; errors in the component densities still affect acceptance.

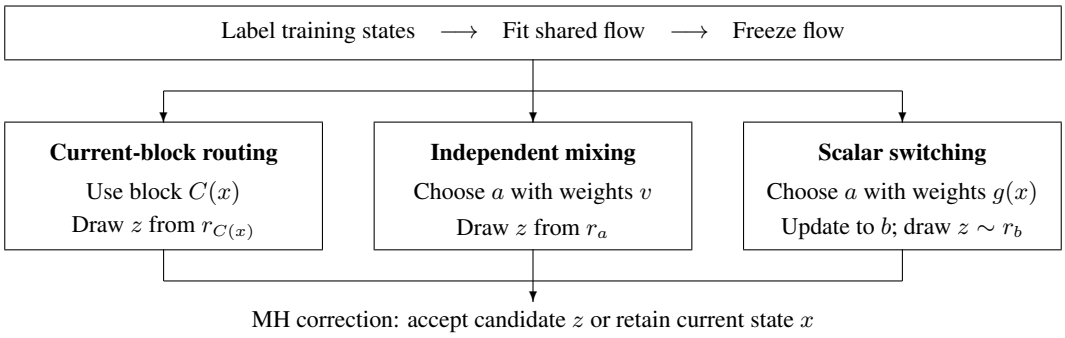


Figure 1: Three ways to use the same fitted proposal densities at a fixed target y. The current state is x, the candidate is z, and $C(x)$ is the current block. Component a has density $r_a = r_\theta(\cdot \mid a, y)$ and mixture weight v_a. Routing uses the current block; independent mixing chooses without using x. For switching, $g_a(x) = v_a r_a(x)/q(x)$ gives the component probabilities at the current state. The label update in Equation (1) favors a different component when $\tau < 0$. Each branch uses its own MH correction.

Our main result identifies when more switching cannot increase long-run estimation variance. We consider square-integrable observables, whose target second moments are finite. For a fixed target π, the ratio $W(x) = \pi(x)/q(x)$ measures how much the mixture over- or underrepresents each state. For our scalar switching rule and a non-atomic target, lag-one covariance cannot increase for any such observable if and only if W has the same distribution under every component. Under this condition, mean acceptance is unchanged and long-run estimation variance cannot increase, even when $q \neq \pi$. Without it, switching can improve estimates of mode probabilities while worsening estimates within a mode (Figure 2).

Contributions. We give an exact condition under which more frequent switching cannot increase covariance, and extend the comparison to switches between selected components (Theorem 1 and Appendix A.6). We show why increasingly accurate within-block proposals can trap current-block routing, while a correctly weighted independent mixture approaches independent target sampling. Our variance bounds quantify the effects of component and weight errors without requiring a positive spectral gap. Experiments on asymmetric inverse problems and nonzero-field lattices compare selection rules using the same fitted proposals, with matched training controls for the effect of block labels.

2 Related work

Flow-assisted MCMC. Normalizing flows (Dinh et al., 2017; Papamakarios et al., 2021) provide independence proposals, adaptive proposals and local transports (Gabrié et al., 2022; Brofos et al., 2022; Hoffman et al., 2019; Grenioux et al., 2023). Local–global methods include Ex^{2}MCMC and flowMC (Samsonov et al., 2022; Wong et al., 2023); independent flows also serve lattice sampling (Albergo et al., 2019). Hackett et al. (2025) use flow mixtures with full-mixture MH correction, including for weakly broken symmetries. We fix the component densities and ask when more frequent switching improves sampling after MH correction.

Mixtures of experts and conditional flows. Mixtures of experts (MoE) use input-dependent weights to select or combine models (Jacobs et al., 1991; Shazeer et al., 2017). Here each "expert" is a proposal density conditioned on a block label. Selection probabilities are computed from these densities at the current state, without a separate gating network. Conditional flows share parameters across targets (Radev et al., 2022; Singha et al., 2023); indexed and piecewise flows use latent components or learned regions (Dinh et al., 2019; Freitas Pires & Figueiredo, 2020; Cornish et al., 2020; Sidheekh et al., 2022; Bevins et al., 2023). Regional and mode-jumping methods correct proposals using source and destination information (Craiu et al., 2009; Bai et al., 2011; Tjelmeland & Hegstad,

2001; Tierney, 1994). JAMS combines adaptive regional proposals, auxiliary mode labels and local moves (Pompe et al., 2020).

Kernel comparison and block averaging. Peskun–Tierney ordering compares reversible kernels through accepted moves (Peskun, 1973; Tierney, 1998); group and projection methods study finite-chain averaging and partitions (Choi et al., 2025; Choi & Wang, 2026). Our analysis uses established mixture-reversible proposals (Tran et al., 2016), data-augmentation comparisons (Khare & Hobert, 2011), convexity of asymptotic variance (Mira & Leisen, 2009), and independence-MH spectral properties (Liu, 1996; Rudolf & Ullrich, 2013). For irreducible reversible kernels, Salmon & Rosenthal (2026) equate covariance and asymptotic-variance order, allowing infinite variances, and show preservation under common mixtures. Antithetic changes can improve discrete Gibbs updates (Neal, 2024; Neal & Rosenthal, 2025), but MH rejection can undo the gain when the proposal differs from the target. We characterize when the improvement survives for scalar and contrast switching.

3 Proposal selection under MH correction

We first fix one target π and its fitted proposals, suppressing y. An observable f is a quantity whose target mean we estimate from MCMC states $X_0, X_1, \ldots$. All densities share a dominating measure. For a stationary chain with transition kernel M, write $\operatorname{avar}(f, M) = \lim_{n\to\infty} n \operatorname{Var}(n^{-1}\sum_{i=1}^{n} f(X_i))$, allowing $+\infty$. The space $L_0^2(\pi)$ contains square-integrable observables with mean zero; $\langle f, h\rangle_\pi = \int fh\, d\pi$.

A reversible selector and its MH correction. Fix a non-atomic target π and $K \geq 2$ normalized proposal densities r_a. Let $q = \sum_a v_a r_a$, $v_a > 0$, and $\sum_a v_a = 1$. In the mixture model $A \sim v$, $X \sim r_A$, Bayes' rule gives $g_a(x) = \Pr(A = a \mid X = x) = v_a r_a(x)/q(x)$. This probability is distinct from the hard partition label $C(x)$. A *selector* is a transition matrix on component labels. *Scalar switching* draws a fresh label $A \sim g(x)$, then $B \sim T_\tau(A, \cdot)$, and finally $z \sim r_B$, where

$$T_\tau(a, b) = \tau \mathbf{1}\{a = b\} + (1 - \tau)v_b, \qquad -\min_a \frac{v_a}{1 - v_a} \leq \tau \leq 1. \tag{1}$$

The parameter τ controls how often the component is retained: $\tau = 0$ is independent mixing, $\tau < 0$ discourages reselecting A, and $\tau = 1$ keeps A while drawing a new candidate from that component. Assume π and q are equivalent, so $W = \pi/q$ is positive and finite q-almost surely. The proposal is q-reversible and its full MH ratio is $1 \wedge W(z)/W(x)$ (Tran et al., 2016). Write M_τ for the corrected kernel.

Let $\bar{\alpha}_\tau$ be acceptance averaged over the stationary chain. It varies linearly with τ, with a nonnegative slope D_W (Appendix Proposition A.3):

$$\bar{\alpha}_\tau = \bar{\alpha}_0 + \tau D_W, \qquad D_W \geq 0 \tag{2}$$

Reducing τ lowers acceptance unless $D_W = 0$. This is also the condition for covariance ordering to hold for every observable.

Theorem 1 (When switching cannot increase covariance)**.** *Let $K \geq 2$ be finite and let π be non-atomic and equivalent to q, so $0 < W < \infty$ q-almost surely. For any admissible $\tau_1 < \tau_2$,*

$$\langle f, (M_{\tau_2} - M_{\tau_1})f\rangle_\pi \geq 0 \text{ for all } f \in L_0^2(\pi) \quad \Longleftrightarrow \quad \mathbb{E}_q[g_a \mid W] = v_a \text{ for all } a. \tag{3}$$

All conditional identities hold q-almost surely. For centered f, $\langle f, Mf\rangle_\pi$ is the stationary covariance of successive values $f(X_0)$ and $f(X_1)$. The left side therefore says that more switching cannot increase this covariance. The right side means that $W = \pi/q$ has the same distribution whichever component generates the state. We call this condition *calibration*. It is also equivalent to $D_W = 0$: changing τ leaves the rejection probability unchanged at almost every current state.

To combine local and global moves, let P be a local kernel reversible with respect to π, and let λ be the probability of a global update. For $\tau_{\min} < \tau_1 < \tau_2 < 1$, where $\tau_{\min} = -\min_a v_a/(1 - v_a)$, calibration is also equivalent to asymptotic-variance ordering for every $f \in L_0^2(\pi)$. The equivalence holds for M_τ and for any common reversible mixture $(1-\lambda)P + \lambda M_\tau$, $0 < \lambda \leq 1$. It allows infinite variances and requires no local spectral gap (Corollary A.6).

If calibration fails on a non-atomic target, we can construct within-region observables for which the rejection cost dominates (Appendix A.5). Fitted components with a common U marginal are calibrated when W depends only on U (Appendix A.1).

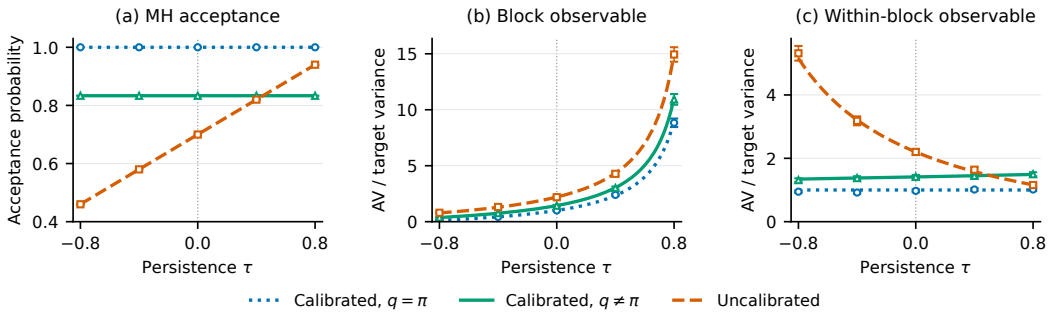


Figure 2: Switching can improve mode estimates while worsening estimates within a mode. Smaller τ means more switching; $\tau = 0$ is independent mixing. Lines show analytic or numerical reference values; markers show estimates from independent stationary chains with 95% Monte Carlo intervals. Estimation variance is normalized by the observable's target variance. More switching lowers the variance of the block-indicator estimate in all three examples, but increases within-block estimation variance in the uncalibrated example.

A continuous example. Fix two uniform proposals on $(0, 1)$ and $(1, 2)$ with equal mixture weights. Targets uniform within each block, with masses $(1/2, 1/2)$ and $(4/5, 1/5)$, give a calibrated ($\pi = q$) and an uncalibrated case. Figure 2 shows exact acceptance and variance for a block indicator and a centered observable that distinguishes the two halves of the first block.

For the uncalibrated target, decreasing τ from 0 to $-.8$ lowers the block indicator's normalized estimation variance from 2.20 to .78, while within-block estimation variance rises from 2.20 to 5.15 and acceptance falls from .70 to .46. For the within-block observable, the rejection cost outweighs the benefit of switching.

Calibration does not require exact proposals. With the same r_1, r_2, v, let $s = x$ on the first interval and $s = x-1$ on the second, and take $\pi(x) = q(x)(\frac{1}{2}+s)$. Both components give $W \sim \mathrm{Unif}(\frac{1}{2}, \frac{3}{2})$, so calibration holds although $q \neq \pi$. Mean acceptance is $5/6$ for every τ. This target's centered observables and numerical variance calculation are given in Appendix A.5.

Switching selected components. A nonzero *contrast* $c = (c_1, \ldots, c_K)$ with $\sum_a v_a c_a = 0$ favors switches between components with opposite signs of c_a. The selector in Appendix A.6 cannot increase lag-one covariance relative to independent mixing for any square-integrable observable exactly when $\mathbb{E}_q[\sum_a c_a g_a(X) \mid W] = 0$. For $K \geq 3$, this can hold even when full calibration fails.

Why better conditional fits can worsen hard routing. Now let the components specialize to blocks $B_a = \{x : C(x) = a\}$ with $0 < \omega_a = \pi(B_a) < 1$, and conditionals $\pi_a = \pi(\cdot \mid B_a)$. The blocks may have unequal masses and different shapes. With ideal components $r_a = \pi_a$, routing freezes the block label, whereas $\sum_a \omega_a r_a = \pi$ gives an independent target draw J_π. Write $\kappa_a = \mathrm{KL}(\pi_a \| r_a)$ for conditional-fit error, measured by Kullback–Leibler (KL) divergence, and $\ell_a = r_a(B_a^c)$ for off-block proposal mass.

Proposition 2 (Conditional fit and hard-routing exits)**.** *Fix a target, suppress y, and assume $0 < \omega_a := \pi(B_a) < 1$. Let r_a be normalized components with every $\kappa_a < \infty$, and write $\rho_a = 1 - \ell_a = r_a(B_a)$. Let $[t]_+ = \max(t, 0)$. Let M_{route} use $r_{C(x)}$ with the exact destination-indexed MH ratio, while M_{mix} is independence MH with proposal $\sum_a \omega_a r_a$. Writing* Gap *for the Poincaré gap and $\bar{\alpha}_{\mathrm{mix}}$ for stationary mean acceptance,*

$$\mathrm{Gap}(M_{\mathrm{route}}) \leq \min_a \frac{\ell_a}{1 - \omega_a} \leq \min_a \frac{1 - e^{-\kappa_a}}{1 - \omega_a}, \tag{4}$$

$$\bar{\alpha}_{\mathrm{mix}} \geq \left[1 - \sqrt{2 \sum_a \omega_a \kappa_a}\right]_+ . \tag{5}$$

Hence, along a fixed partition, $\max_a \kappa_a \to 0$ *drives the routed gap to zero and the mixture acceptance to one.*

Small KL can coexist with rare states from which escape is very slow. Using the stronger discrepancy $\chi^2(p\|q) = \int (p/q-1)^2\, dq$, we can compare estimation variance without requiring a positive spectral gap.

Corollary 3 (Variance separation without uniform coverage)**.** *Under Proposition 2, let* $D_{\mathrm{cond}} = \sum_a \omega_a \chi^2(\pi_a\|r_a) < \infty$ *and* $f_a = \mathbf{1}_{B_a} - \omega_a$*. Along a fixed partition,* $D_{\mathrm{cond}} \to 0$ *makes* $\mathrm{avar}(f_a, M_{\mathrm{route}})$ *diverge while* $\mathrm{avar}(f_a, M_{\mathrm{mix}})$ *approaches its iid value, even when every mixture kernel has zero spectral gap.*

In a combined sampler, sufficiently frequent local moves between blocks can prevent this divergence (Appendix Equation (A.7)). Weights that differ from the target block masses introduce a separate error. For $q_v = \sum_a v_a r_a$, $v_a > 0$, $\sum_a v_a = 1$, and $d_a = \chi^2(\pi_a\|r_a)$,

$$\chi^2(\pi\|q_v) \le B_v := \chi^2(\omega\|v) + \sum_a \frac{\omega_a^2}{v_a} d_a. \tag{6}$$

Let M_{q_v} denote independence MH with proposal q_v. For bounded f whose essential range has length L_f, the additional variance of $(1-\lambda)P + \lambda M_{q_v}$ relative to $(1-\lambda)P + \lambda J_\pi$ is at most $L_f^2(2B_v + \sqrt{B_v})/\lambda$, for any reversible P and $0 < \lambda \le 1$ when $B_v < \infty$ (Appendix A.3). The two terms in B_v separate weight error from conditional-fit error. Block indicators are one class of bounded observables covered by this result.

4 Block partitions and conditional proposal training

Let $\mathcal{Y}$ index the observations or model parameters of one target family. We assume a known reference distribution $\pi_\star$, a known low-dimensional statistic $U(x)$, and positive functions w_y such that

$$\pi_y(dx) = \frac{w_y(U(x))}{Z_y}\, \pi_\star(dx), \qquad \pi_y(dx \mid U = u) = \pi_\star(dx \mid U = u). \tag{7}$$

Here $Z_y = \mathbb{E}_{\pi_\star}[w_y(U)] \in (0, \infty)$ normalizes the target and $0 < w_y(U) < \infty$. The map U is non-injective: distinct states can have the same statistic. The conditional identity holds almost everywhere for each y; only the marginal of U changes. In inverse problems, $\pi_\star$ is the prior and $w_y(U(x)) = p(y \mid x)$ is the likelihood.

A partition $C(x) \in \{1, \dots, K\}$ assigns each complete state to a region. With $p_a(y) = \pi_y(C = a) > 0$, we have $\pi_y = \sum_a p_a(y)\pi_y(\cdot \mid C = a)$. A shared flow $r_\theta(\cdot \mid a, y)$ approximates each conditional $\pi_y(\cdot \mid C = a)$. We fit a separate network for each target family, sharing its parameters across blocks and training values of y. Within each block, Equation (7) gives $\pi_y(dx \mid U = u, C = a) = \pi_\star(dx \mid U = u, C = a)$. Pooling targets supplies data for this common conditional law; conditioning on y lets the flow fit each target's U marginal within the block.

Why preserve the distribution of U within each block. An additional design condition is independence of the block label and statistic, $C \perp U$, under $\pi_\star$. Reweighting by $w_y(U)$ then gives

$$\mathcal{L}_{\pi_y}(U \mid C = a) = \mathcal{L}_{\pi_y}(U), \qquad p_a(y) = p_a^\star := \pi_\star(C = a). \tag{8}$$

Here $\mathcal{L}_\pi$ denotes distribution under π. An ideal draw within any block then refreshes U from its full target marginal. This block condition is stronger than the shared conditional law in Equation (7); we test it in the inverse designs and also study extensions where it fails.

For example, suppose $x = (u, z)$, $U(x) = u$, and the reference density factors as $\pi_\star(u, z) = p_U(u)p_Z(z)$. Reweighting by $w_y(u)$ gives

$$\pi_y(u, z) = p_{U,y}(u)p_Z(z), \qquad p_{U,y}(u) = \frac{w_y(u)p_U(u)}{Z_y}.$$

Any partition depending only on z then preserves the U marginal within every block. The residual density p_Z may have unequal mode weights, different covariances or irregular shapes.

Step 1: fix the partition. For inverse problems with $U(x) = Ax$, we cluster the unobserved part of the state. We project unlabelled reference samples onto the null space of A, retain the first two principal components, and fit K-means with $K = 4$. The nearest cluster center determines $C(x)$ for every target condition. Lattice blocks use magnetization sign; a separate control learns a neural partition (Appendix D). We then freeze C.

Step 2: fit one flow across blocks and targets. Let $\mathcal{Y}_{\mathrm{tr}}$ be the finite set of training conditions. For each $y \in \mathcal{Y}_{\mathrm{tr}}$, collect N_y target samples $\mathcal{D}_y = \{x_{i,y}\}_{i=1}^{N_y}$. Our inverse problems permit independent exact posterior draws; lattice data come from separate parallel-tempering (PT) runs, which exchange states between chains at different temperatures. Apply the fixed partition to these samples, defining $I_{a,y} = \{i : C(x_{i,y}) = a\}$ and $n_{a,y} = |I_{a,y}| > 0$. We optimize the network parameters θ by negative log-likelihood (NLL):

$$\widehat{L}_{\mathrm{prop}} = -\frac{1}{|\mathcal{Y}_{\mathrm{tr}}|K} \sum_{y \in \mathcal{Y}_{\mathrm{tr}}} \sum_{a=1}^{K} \frac{1}{n_{a,y}} \sum_{i \in I_{a,y}} \log r_\theta(x_{i,y} \mid a, y). \tag{9}$$

Each target–block pair contributes equally to the loss, so rare blocks receive as much training weight as common ones. Each component is a density over the full state space; its samples can fall outside the block used for training. We select the network on separate validation conditions and freeze its parameters. At a new target y, we change the inputs a, y without retraining. A matched *target-only flow* uses $K = 1$ and omits the block label.

Step 3: set the weights and sample. Weights $v_a(y)$ specify each component's contribution to the mixture:

$$q_{v,y}(x') = \sum_{a=1}^{K} v_a(y) r_\theta(x' \mid a, y), \qquad v_a(y) > 0, \quad \sum_a v_a(y) = 1. \tag{10}$$

In the inverse experiments, we compare $v_a = 1/K$ with estimated block probabilities. From fresh reference draws $x_1^\star, \ldots, x_{N_\star}^\star \sim \pi_\star$, compute

$$\widehat{v}_a(y) = \frac{\sum_{j=1}^{N_\star} w_y(U(x_j^\star)) \mathbf{1}\{C(x_j^\star) = a\}}{\sum_{j=1}^{N_\star} w_y(U(x_j^\star))}.$$

This estimates each block's target probability by weighting reference samples with their likelihoods. For lattices, we fit weights by maximizing the mixture likelihood on separate PT samples at the test field. The flow remains fixed during weight fitting, and both are fixed during the subsequent sampling run.

Balanced training allocates equal fitting effort to blocks, while deployment weights reflect their probabilities. With exact within-block fits and $v_a(y) = p_a(y)$, Equation (10) gives $q_{v,y} = \pi_y$, even for unequal block masses. A rare block can therefore receive as much training effort as a common one without being oversampled in the proposal mixture.

Independent mixing draws x' from $q_{v,y}$ and applies MH using the full mixture density. Scalar switching uses these same components and weights in Equation (1), with $\tau = .8\tau_{\min}(v)$ in the main comparisons. Both use the ratio $W(x')/W(x)$ from Section 3. For comparison, *iterated sampling–importance resampling* (i-SIR) selects among several candidates, including the current state, with weights $\widetilde{\pi}_y/q_{v,y}$; $\widetilde{\pi}_y$ is the target density up to its unknown normalizing constant (Andrieu et al., 2018; Samsonov et al., 2022). Each global update is mixed with a local MCMC kernel preserving π_y.

Hard routing instead draws $x' \sim r_\theta(\cdot \mid C(x), y)$. With current label $a = C(x)$ and proposed label $b = C(x')$, it accepts with

$$\alpha_y(x, x') = 1 \wedge \frac{\widetilde{\pi}_y(x') r_\theta(x \mid b, y)}{\widetilde{\pi}_y(x) r_\theta(x' \mid a, y)}. \tag{11}$$

The reverse density uses the destination label; cross-block proposals can be accepted.

5 EXPERIMENTS

Efficiency measures. We compare how accurately each sampler estimates target expectations. Efficiency per cycle is inverse asymptotic variance (AV); a ratio of 2 means half the variance at the same number of cycles. We also account for elapsed time and target-density evaluations. Ratios above 1 favor the listed method over the stated baseline. We take geometric means over observables, conditions and model seeds. Five-seed comparisons use paired 95% intervals (Appendix E). Runs that fail mixing checks are excluded from efficiency comparisons. Timing includes burn-in and all executed particles/replicas, excluding training and setup.

5.1 ASYMMETRIC INVERSE PROBLEMS

For $d \in \{50, 200\}$, write $x = (u, z, t)$, with $u, z \in \mathbb{R}^2$ and $t \in \mathbb{R}^{d-4}$. The reference density $p_\star$ has four residual mixture groups, each with two observed subcomponents. Multimodality is confined to the four coordinates (u, z); t is an independent Gaussian tail. Observing $y = u + \xi$ with $\xi \sim \mathcal{N}(0, .7^2 I_2)$ gives

$$\pi_y(x) = \frac{p_\star(x)\varphi_2(y; u, .7^2 I_2)}{p(y)}, \qquad U(x) = u. \tag{12}$$

Here φ_2 is the bivariate Gaussian density and $p(y)$ is the normalizing constant. The likelihood depends on x only through $U(x) = u$, giving the shared conditional structure in Equation (7). These posterior mixtures have analytic moments and mode probabilities, so mixing errors can be checked independently of a sampler.

Two designs satisfy Equation (8): *weights* varies mode probabilities; *geometry* varies centers and covariances while keeping mode probabilities equal. In both, $U = u$ is independent of (z, t). The *moderate* and *strong coupling* designs introduce dependence between observed and residual coordinates, so mode masses depend on y. All four break mode-exchanging sign symmetries; Appendix B.1 gives their parameters.

Training and sampling. With five training seeds per design and dimension, we fit paired $K = 4$ block-conditioned and $K = 1$ target-only flows on the same data for 16,000 updates. Separate validation contexts select checkpoints and sampler tuning; evaluation uses six held-out contexts. Each cycle chooses, with equal probability, either two successive Metropolis-adjusted Langevin (MALA) updates (Roberts & Tweedie, 1996) or one global update. The MALA step size is chosen on validation targets. We compare routing, mixtures, scalar switching and i-SIR with the same block-conditioned proposals, alongside target-only MH, local MALA and parallel tempering (PT) (Earl & Deem, 2005). Mixture weights are uniform or estimated block masses; the latter use independent reference samples. Appendix B.1 gives the sampling budgets and tuning.

Efficiency and diagnostics. We estimate variance by batch means (Flegal & Jones, 2010), with alternative batch lengths and chain means as sensitivity checks (Table F.1). The mode summary uses posterior probabilities $\Pr(k \mid x, y)$ for each of the four generating mixture groups k. Their means are known analytically; these group indices are distinct from the learned block labels and selector probabilities g_a. All methods except hard routing and local MALA pass every mixing and mean check; these two baselines are excluded from efficiency rankings (Appendix F.3).

Mode efficiency. At $d = 200$, the proposal mixture roughly doubles mode efficiency over the target-only flow; switching adds another 13% (Table 1). i-SIR gives the largest gains per cycle and per second, but uses more target evaluations.

Block labels versus balanced training. With model size, data and training budget matched, uniform-mixture MH with labelled components achieves 1.15–5.74× the mode efficiency of an

Table 1: Estimating mode probabilities in asymmetric inverse problems. Efficiency is relative to the target-only flow (1.00 at each dimension; higher is better). Each row averages four designs, five seeds per design and six test conditions; brackets give 95% paired intervals. All methods include MALA; the mixture methods share their components and estimated-mass weights.

d	Method	Per cycle	Per second	Per target call
50	Target-only flow	1.00	1.00	1.00
50	Proposal mixture	1.26 [1.22, 1.32]	1.49 [1.43, 1.56]	1.26 [1.22, 1.32]
50	Scalar switching	1.45 [1.40, 1.51]	1.72 [1.63, 1.80]	1.45 [1.40, 1.51]
50	i-SIR	1.64 [1.57, 1.71]	2.57 [2.48, 2.67]	0.38 [0.35, 0.41]
200	Target-only flow	1.00	1.00	1.00
200	Proposal mixture	2.00 [1.88, 2.10]	2.21 [2.07, 2.33]	2.00 [1.88, 2.10]
200	Scalar switching	2.26 [2.12, 2.37]	2.50 [2.35, 2.64]	2.26 [2.12, 2.37]
200	i-SIR	3.64 [3.42, 3.83]	4.85 [4.56, 5.11]	0.90 [0.83, 0.97]

unlabelled flow trained with equal block sampling. This holds in the weights and geometry designs at both dimensions; every paired 95% interval lies above one (Table B.4).

Switching strength and weights. With fixed proposals in the strongly coupled $d = 200$ design, more switching improves mode estimation but reduces within-mode efficiency under uniform weights. Estimated block masses largely remove this loss (Table 3). Fitted switching directions give no clear gain over random directions (Appendix B.4).

5.2 Nonzero-field lattice models

On a periodic $L \times L$ lattice, $x_i \in \mathbb{R}$ is the field at site i, $d = L^2$, and e_1, e_2 are the two lattice unit directions. Define

$$E(x) = .2 \sum_i \left[(x_i - x_{i+e_1})^2 + (x_i - x_{i+e_2})^2\right] + \tfrac{1}{4} \sum_i (x_i^2 - 1)^2,$$
$$M(x) = \sum_i x_i, \qquad \pi_{\beta,h}(x) = Z_{\beta,h}^{-1} \exp\{-\beta E(x) + hM(x)\}. \tag{13}$$

Here β is inverse temperature, h is the external field, and $Z_{\beta,h}$ normalizes the density. The magnetization per site is $m = M/d$; $U(x) = (E(x), M(x))$ gives the shared statistic in Equation (7). The flow takes $y = (\beta, h)$ as input. We train at $L = 16, 32$ ($d = 256, 1024$) and $\beta = 3$, using the scaled field $\eta = L^2 h$ to retain both phases as the lattice grows. Training, validation and test fields are disjoint and nonzero (Appendix C), so global sign reversal is not a target symmetry. Independent PT chains provide training samples, weight-fitting samples and numerical estimates of the target expectations.

Models and sampling. At each size, we train five pairs of target-only and two-component flows with matched parameter counts and periodic CNN coupling layers. Two-component training uses balanced conditional NLL with the phase label $C(x) = \mathbf{1}\{M(x) > 0\}$ (labels $0, 1$ here). Since this label is determined by U, these blocks do not satisfy Equation (8). Both flows are frozen after 49,152 updates on the same training data (Appendix C.1).

We compare uniform-mixture MH, MH with NLL-fitted weights, and scalar switching with the fitted weights. All reuse the flow at unseen fields. Fitted weights require separate PT samples at each test field; uniform weights need no such samples. Each mixes its global move with validation-tuned Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 2011) with probability $1/2$. HMC with an MH-corrected flip and PT provide physical baselines; Appendix C gives their settings and diagnostics.

Phase mixing and efficiency. Mixtures pass all diagnostics; the target-only flow usually fails, and hard routing never changes phase (Appendix C.2). HMC with corrected flips and PT remain more efficient per cycle and per second (Table 2).

Table 2: Lattice sampling at $d = 256$ (16×16) and $d = 1024$ (32×32). Efficiency is relative to HMC with a corrected sign flip (1.00); higher is better. Ratios average $E/d, m, |m|, m^2, \mathbf{1}\{m > 0\}$ over four fields and five seeds, with 95% paired intervals. "Passed" counts runs meeting the diagnostic checks; – denotes unreliable efficiency estimates. All flow methods include HMC. PT costs include all replicas.

d	Method	Passed	Per cycle	Per second
256	Target-only flow	8/20	–	–
256	Uniform mixture	20/20	0.88 [0.85, 0.91]	0.18 [0.17, 0.19]
256	Fitted mixture	20/20	0.87 [0.85, 0.91]	0.18 [0.16, 0.19]
256	Scalar switching	20/20	0.93 [0.90, 0.96]	0.19 [0.17, 0.20]
256	HMC + flip	20/20	1.00	1.00
256	Parallel tempering	20/20	1.85 [1.78, 1.89]	4.92 [4.50, 5.39]
1024	Target-only flow	0/20	–	–
1024	Uniform mixture	20/20	0.51 [0.48, 0.54]	0.07 [0.06, 0.07]
1024	Fitted mixture	20/20	0.46 [0.42, 0.50]	0.06 [0.06, 0.07]
1024	Scalar switching	20/20	0.48 [0.44, 0.52]	0.06 [0.06, 0.07]
1024	HMC + flip	20/20	1.00	1.00
1024	Parallel tempering	20/20	1.27 [1.25, 1.29]	2.11 [2.02, 2.26]

At $d = 1024$, switching with uniform weights improves phase efficiency by 74% and the five-observable mean by 15%. Fitted weights yield smaller gains with almost no acceptance loss (Table 3).

Table 3: Switching versus independent mixing, with models, weights and local updates fixed. Per-cycle efficiency at $\tau = .8\tau_{\min}$ is relative to $\tau = 0$; higher is better. Acceptance shows the global MH rate before and after switching. Each panel uses two model seeds; full sweeps are in Appendices B.2 and C.3.

Strongly coupled inverse problem: $d = 200$, three test conditions			
Weights	Global acceptance	Mode efficiency	Within-mode efficiency
Uniform	$42.9\% \to 39.5\%$	1.28	0.91
Estimated masses	$55.3\% \to 55.4\%$	1.04	1.00
Lattice: $d = 1024$, two test fields			
Weights	Global acceptance	Phase efficiency	Five-observable mean
Uniform	$11.8\% \to 7.0\%$	1.74	1.15
Fitted	$17.8\% \to 17.8\%$	1.02	1.01

6 Limitations

The inverse tests use four residual groups; lattice phase labels are supplied. Learning a partition gives no clear advantage over target-only flows (Appendix D). We have not isolated the benefit of preserving U within blocks or verified the exact condition for improving every observable with the fitted flows.

Main comparisons use five seeds; switching sweeps use two. All 40 asymmetric fits select the largest MALA step and PT replica count tested. Broader tuning ranges, block counts and applications remain open. Timings cover sampling on shared devices, excluding training and data generation.

7 Conclusion

Component selection matters even when proposal densities are fixed. Our theory identifies when more switching can improve long-run precision without extra rejection. The experiments show how block labels and mixture weights affect the tradeoff between cross-mode movement and rejection.

Reproducibility statement

Figure 2 uses closed-form and numerically converged Poisson-equation curves, checked by direct MH simulations. The repository linked in the abstract provides the core implementation, inverse-problem training code, and scripts and saved records for this figure and Tables E.1 and 1. Appendix F describes the variance estimators and diagnostic checks. Complete observable traces were retained for the training and selection controls, switching-strength sweeps, the 360 lattice replication runs, and the reference and PT simulations. The earlier 160 mixture/flip runs retain only batch means.

AI use statement

Generative AI assisted programming, experimental analysis, mathematical checks and manuscript editing. The authors directed and reviewed this work and take responsibility for the submission.

References

Michael S. Albergo, Gurtej Kanwar, and Phiala E. Shanahan. Flow-based generative models for Markov chain Monte Carlo in lattice field theory. *Physical Review D*, 100(3):034515, 2019. doi: 10.1103/PhysRevD.100.034515.

Christophe Andrieu, Anthony Lee, and Matti Vihola. Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, 24(2):842–872, 2018. doi: 10.3150/15-BEJ785.

Yan Bai, Radu V. Craiu, and Antonio F. Di Narzo. Divide and conquer: A mixture-based approach to regional adaptation for MCMC. *Journal of Computational and Graphical Statistics*, 20(1): 63–79, 2011. doi: 10.1198/jcgs.2010.09035.

Harry Bevins, Will Handley, and Thomas Gessey-Jones. Piecewise normalizing flows. *arXiv preprint arXiv:2305.02930*, 2023. URL https://arxiv.org/abs/2305.02930.

James Brofos, Marylou Gabrié, Marcus A. Brubaker, and Roy R. Lederman. Adaptation of the independent Metropolis–Hastings sampler with normalizing flow proposals. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 5949–5986, 2022. URL https://proceedings.mlr.press/v151/brofos22a.html.

Michael C. H. Choi and Youjia Wang. Spectral partitioning for k-block averaging kernels of finite Markov chains, 2026. URL https://arxiv.org/abs/2608.21466.

Michael C. H. Choi, Max Hird, and Youjia Wang. Improving the convergence of Markov chains via permutations and projections. *Random Structures & Algorithms*, 66(4):e70016, 2025. doi: 10.1002/rsa.70016. URL https://arxiv.org/abs/2411.08295.

Rob Cornish, Anthony Caterini, George Deligiannidis, and Arnaud Doucet. Relaxing bijectivity constraints with continuously indexed normalising flows. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2133–2143, 2020. URL https://proceedings.mlr.press/v119/cornish20a.html.

Radu V. Craiu, Jeffrey Rosenthal, and Chao Yang. Learn from thy neighbor: Parallel-chain and regional adaptive MCMC. *Journal of the American Statistical Association*, 104(488):1454–1466, 2009. doi: 10.1198/jasa.2009.tm08393.

George Deligiannidis and Anthony Lee. Which ergodic averages have finite asymptotic variance? *The Annals of Applied Probability*, 28(4):2309–2334, 2018. doi: 10.1214/17-AAP1358. URL https://arxiv.org/abs/1606.08373.

Dinesh P. R. Locality-constrained autoregressive cum conditional normalizing flow for lattice field theory simulations, 2023. URL https://arxiv.org/abs/2304.01798.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Laurent Dinh, Jascha Sohl-Dickstein, Hugo Larochelle, and Razvan Pascanu. A RAD approach to deep mixture models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2019. URL https://arxiv.org/abs/1903.07714.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987. doi: 10.1016/0370-2693(87)91197-X.

David J. Earl and Michael W. Deem. Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7:3910–3916, 2005. doi: 10.1039/B509983H.

James M. Flegal and Galin L. Jones. Batch means and spectral variance estimators in Markov chain Monte Carlo. *The Annals of Statistics*, 38(2):1034–1070, 2010. doi: 10.1214/09-AOS735.

Guilherme G. P. Freitas Pires and Mário A. T. Figueiredo. Variational mixture of normalizing flows. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 205–210, 2020. URL https://arxiv.org/abs/2009.00585.

Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. Adaptive Monte Carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10):e2109420119, 2022. doi: 10.1073/pnas.2109420119.

Louis Grenioux, Alain Oliviero Durmus, Éric Moulines, and Marylou Gabrié. On sampling with approximate transport maps. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11698–11733, 2023. URL https://proceedings.mlr.press/v202/grenioux23a.html.

Daniel C. Hackett, Chung-Chun Hsieh, Sahil Pontula, Michael S. Albergo, Denis Boyda, Jiunn-Wei Chen, Kai-Feng Chen, Kyle Cranmer, Gurtej Kanwar, and Phiala E. Shanahan. Flow-based sampling for multimodal and extended-mode distributions in lattice field theory, 2025. URL https://arxiv.org/abs/2107.00734v2.

Matthew D. Hoffman, Pavel Sountsov, Joshua V. Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. NeuTra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport, 2019. URL https://arxiv.org/abs/1903.03704.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Kshitij Khare and James P. Hobert. A spectral analytic comparison of trace-class data augmentation algorithms and their sandwich variants. *The Annals of Statistics*, 39(5):2585–2606, 2011. doi: 10.1214/11-AOS916.

Jun S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119, 1996. doi: 10.1007/BF00162521.

Antonietta Mira. Ordering and improving the performance of Monte Carlo Markov Chains. *Statistical Science*, 16(4):340–350, 2001. doi: 10.1214/ss/1015346319.

Antonietta Mira and Fabrizio Leisen. Covariance ordering for discrete and continuous time Markov chains. *Statistica Sinica*, 19(2):651–666, 2009. URL https://www3.stat.sinica.edu.tw/sstest/oldpdf/A19n213.pdf.

Radford M. Neal. MCMC using Hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin L. Jones, and Xiao-Li Meng (eds.), *Handbook of Markov Chain Monte Carlo*, pp. 113–162. Chapman & Hall/CRC, 2011. URL https://arxiv.org/abs/1206.1901.

Radford M. Neal. Modifying Gibbs sampling to avoid self transitions, 2024. URL https://arxiv.org/abs/2403.18054.

Radford M. Neal and Jeffrey S. Rosenthal. Efficiency of reversible MCMC methods: Elementary derivations and applications to composite methods. *Journal of Applied Probability*, 62(1):188–208, 2025. doi: 10.1017/jpr.2024.48.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL https://jmlr.org/papers/v22/19-1028.html.

Peter H. Peskun. Optimum Monte Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612, 1973. doi: 10.1093/biomet/60.3.607.

Emilia Pompe, Chris Holmes, and Krzysztof Łatuszyński. A framework for adaptive MCMC targeting multimodal distributions. *The Annals of Statistics*, 48(5):2930–2952, 2020. doi: 10.1214/19-AOS1916.

Stefan T. Radev, Ulf K. Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1452–1466, 2022. doi: 10.1109/TNNLS.2020.3042395.

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996. doi: 10.2307/3318418.

Murray Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952. doi: 10.1214/aoms/1177729394.

Daniel Rudolf and Mario Ullrich. Positivity of hit-and-run and related algorithms. *Electronic Communications in Probability*, 18(49):1–8, 2013. doi: 10.1214/ECP.v18-2507. URL https://arxiv.org/abs/1212.4512.

Geoffrey T. Salmon and Jeffrey S. Rosenthal. Comparing the efficiency of general state space reversible MCMC algorithms. *Advances in Applied Probability*, 58(3):1063–1093, 2026. doi: 10.1017/apr.2025.10048.

Sergey Samsonov, Evgeny Lagutin, Marylou Gabrié, Alain Durmus, Alexey Naumov, and Eric Moulines. Local-global MCMC kernels: The best of both worlds. In *Advances in Neural Information Processing Systems*, volume 35, 2022. NeurIPS proceedings.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Sahil Sidheekh, Chris B. Dock, Tushar Jain, Radu Balan, and Maneesh K. Singh. VQ-Flows: Vector quantized local normalizing flows. In *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 1835–1845, 2022. URL https://proceedings.mlr.press/v180/sidheekh22a.html.

Ankur Singha, Dipankar Chakrabarti, and Vipul Arora. Conditional normalizing flow for Markov Chain Monte Carlo sampling in the critical region of lattice field theory. *Physical Review D*, 107:014512, 2023. doi: 10.1103/PhysRevD.107.014512.

Andrew M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010. doi: 10.1017/S0962492910000061.

Luke Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1762, 1994. doi: 10.1214/aos/1176325750.

Luke Tierney. A note on Metropolis–Hastings kernels for general state spaces. *The Annals of Applied Probability*, 8(1):1–9, 1998. doi: 10.1214/aoap/1027961031.

Håkon Tjelmeland and Bjørn Kåre Hegstad. Mode jumping proposals in MCMC. *Scandinavian Journal of Statistics*, 28(1):205–223, 2001. doi: 10.1111/1467-9469.00232.

Minh-Ngoc Tran, Michael K. Pitt, and Robert Kohn. Adaptive Metropolis–Hastings sampling using reversible dependent mixture proposals. *Statistics and Computing*, 26(1–2):361–381, 2016. doi: 10.1007/s11222-014-9509-6. URL https://arxiv.org/abs/1305.2634.

Aki Vehtari, Andrew Gelman, Daniel Simpson, Bob Carpenter, and Paul-Christian Bürkner. Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC (with discussion). *Bayesian Analysis*, 16(2):667–718, 2021. doi: 10.1214/20-BA1221.

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008. doi: 10.1561/2200000001.

Kaze W. K. Wong, Marylou Gabrié, and Daniel Foreman-Mackey. flowMC: Normalizing flow enhanced sampling package for probabilistic inference in JAX. *Journal of Open Source Software*, 8(83):5021, 2023. doi: 10.21105/joss.05021.