Continuous adaptive path sampling for efficient multimodal sampling and marginalization

Yuling Yao* Collin Cademartori † Aki Vehtari‡ Andrew Gelman§

18 Aug 2025

Abstract

The normalizing constant of an unnormalized posterior distribution is central to Bayesian computation, and the estimation of these constants has been extensively studied. An important generalization of this problem is to estimate a family of normalizing constants indexed by a continuous parameter. This problem arises in marginal posterior density estimation and continuous simulated tempering. We propose an algorithm that extends the classical path sampling estimator to efficiently estimate these continuously-parametrized normalizing constants while integrating easily with existing, widely-used sampling algorithms. Since the path sampling estimator directly produces estimates for the whole family of constants, it is particularly natural in this setting. Our continuous adaptive path sampling algorithm iteratively adapts to the structure of the continuous normalizing constant, remedying bottlenecks that can frustrate convergence for non-adaptive estimators. We prove a near consistency result for our estimator and demonstrate its effectiveness in a collection of simulated tempering problems.

1. Introduction

While modern Markov chain Monte Carlo algorithms enable computationally efficient Bayesian inference across a large class of models, the most popular and generally applicable algorithms (e.g., Hamiltonian Monte Carlo and Gibbs samplers) share some fundamental limitations:

1. Sampling from multimodal posterior distributions can degrade convergence rates by orders of magnitude as samplers struggle to move between modes, making reliable inference for such models practically impossible.

2. While posterior samples can theoretically be used to estimate the expectation of any function of model parameters, estimates of marginal densities $p(\theta_1) = \mathrm{E}\, p(\theta_1|\boldsymbol{\theta}_2)$ can suffer large Monte Carlo error when $\dim(\boldsymbol{\theta}_2)$ is large. These errors threaten the estimation of downstream quantities that depend on marginal densities, such as Bayes factors and highest posterior density intervals.

Building on the work of Gelman and Meng (1998), we introduce a method for efficient marginal density estimation which, when combined with a simulated tempering algorithm, also enables tractable sampling of multimodal posteriors. Our algorithm—continuous adaptive path sampling (CAPS)—easily integrates with existing samplers (like Stan's HMC implementation, which we use throughout), making it simple for practitioners to implement within existing workflows. The central idea of our proposed algorithm is to exploit a duality between marginal *estimation* and marginal *control*.

When estimating a marginal density $p(\theta_1) = \int p(\theta_1, \boldsymbol{\theta}_2) d\boldsymbol{\theta}_2$ from joint samples $(\theta_1, \boldsymbol{\theta}_2) \sim p$, accuracy can degrade in the tails of $p(\theta_1)$ due to the scarcity of samples in this region. As we show

*Department of Statistics and Data Sciences, University of Texas, Austin.
†Department of Statistical Sciences, Wake Forest University, North Carolina. Joint first author.
‡Department of Computer Science, Aalto University, Finland.
§Department of Statistics and Political Science, Columbia University, New York.

in detail later, this can be overcome if we can sample from a distribution $\widetilde{p}(\theta_1, \boldsymbol{\theta}_2)$ which differs from p only in that the marginal $\widetilde{p}(\theta_1)$ places greater density over the tails of $p(\theta_1)$. Any such joint distribution $\widetilde{p}$ has the form

$$\widetilde{p}(\theta_1, \boldsymbol{\theta}_2) = p(\theta_1, \boldsymbol{\theta}_2) \frac{\widetilde{p}(\theta_1)}{p(\theta_1)}. \tag{1}$$

But, given knowledge of the original density $p(\theta_1, \boldsymbol{\theta}_2)$ and target marginal $\widetilde{p}(\theta_1)$, constructing the density (1)—and thus controlling the marginal distribution of θ_1—reduces to estimating the marginal $p(\theta_1)$.

This relationship between estimation and control also explains the connection between the marginal estimation and multimodal sampling problems. A common approach to the latter is to introduce an auxiliary parameter λ, forming a joint density $\overline{p}(\boldsymbol{\theta}, \lambda)$, so that $\overline{p}(\boldsymbol{\theta} \mid \lambda = 1) = p(\boldsymbol{\theta})$ and $\overline{p}(\boldsymbol{\theta} \mid \lambda = 0)$ is unimodal. In the best case, the disparate modes in $\boldsymbol{\theta}$-space become connected by regions of nonvanishing probability in $(\boldsymbol{\theta}, \lambda)$-space, allowing MCMC samplers to converge. However, the success of this strategy requires that the marginal $\overline{p}(\lambda)$ places density near $\lambda = 0$ and does not itself exhibit intractable features like multimodality. In other words, this approach requires control over the marginal of λ, which in turn requires its accurate estimation.

The CAPS algorithm exploits this duality by alternating between estimation and control steps. As we show in Theorem 1, adapting in this way can yield large convergence speedups in the pre-asymptotic regime. This theoretical result is validated in our experiments, which show improvements compared to existing algorithms for traditional (non-adaptive) tempering (e.g., Geyer, 1991) and bridge sampling (Meng and Wong, 1996). We also demonstrate that CAPS combined with the HMC implementation in Stan can reliably sample two classes of distributions that are intractable using HMC or traditional tempering methods: high-dimensional Gaussian mixtures with separated modes, and a highly non-log-concave distribution (the flower distribution; Sejdinovic et al., 2014).

1.1. Previous work

Because marginal densities can be cast as normalizing constants of conditional distributions, CAPS can be viewed as an estimate of a normalizing constant or, more generally, a normalizing function if it depends on parameters of a model. Other widely-used approaches to normalizing constant estimation include importance sampling, bridge sampling (Meng and Wong, 1996), Rao-Blackwellized estimation (Carlson et al., 2016), and non-equilibrium methods (Jarzynski, 1997). We compare CAPS to these methods in more detail in Section 2. We refer to Lelievre et al. (2010) for a comprehensive review on normalizing functions.

A wide array of algorithms have also been proposed for sampling from multimodal distributions. Classic approaches which modify commonly used sampling algorithms include simulated tempering (Marinari and Parisi, 1992) and mode jumping (Tjelmeland and Hegstad, 2001). Other approaches employ less common or entirely new sampling algorithms, including sequential Monte Carlo (Schweizer, 2012; Paulin et al., 2018) and adiabatic Monte Carlo (Betancourt, 2015). For a review of Monte Carlo algorithms for multimodal distributions, we refer the reader to Latuszyński et al. (2025).

While we suspect that specialized algorithms may outperform our proposed CAPS algorithm in many multimodal problems, such algorithms usually do not have widely-available, mature implementations in software, making them particularly difficult for practitioners to employ. We thus restrict our attention in this article to methods which can be deployed within existing and popular MCMC implementations such as Stan. Specifically, for multimodal problems, we compare our CAPS-based tempering procedure to simulated tempering and a recently-proposed variant with a continuous temperature variable (Graham and Storkey, 2017).

1.2. Outline

The remainder of this paper is organized as follows. In Section 2, we relate marginal estimation to normalizing constant estimation and review existing methods for this problem. Section 3 introduces our one-shot path sampling algorithm in general as well as its extension to tempering problems. In Section 4, we analyze the one-shot estimator theoretically, deriving a bound on the error probability which both characterizes the asymptotic behavior of the estimator and demonstrates the importance of the adaptation step for improving pre-asymptotic performance. Section 5 presents simulation experiments which compare the CAPS algorithm to alternatives in a series of tempering problems, demonstrating the practical speedups enabled by the adaptive iteration. Finally, we discuss limitations and directions for future work in Section 6.

2. Estimates of the normalizing function

Marginal density estimation can be viewed as a special case of the following generalized normalizing constant problem. Given a scalar-indexed family of unnormalized densities $q(\boldsymbol{\theta};\lambda)_{\lambda\in\Lambda}$ for $\Lambda\subset\mathbb{R}$, we wish to compute the normalizing constant function

$$z(\lambda)=\int q(\boldsymbol{\theta};\lambda)d\boldsymbol{\theta} \text{ for all } \lambda\in\Lambda. \tag{2}$$

This function normalizes the family in that $\int q(\boldsymbol{\theta};\lambda)/z(\lambda)d\boldsymbol{\theta}=1$ for all λ. Taking $q(\boldsymbol{\theta};\lambda)=q(\theta_1,\ldots,\theta_i=\lambda,\ldots,\theta_d,\mathbf{y})$ recovers the problem of estimating the (unnormalized) marginal $q(\theta_i\mid\mathbf{y})$. As in this case, we will usually only need to estimate $z(\lambda)$ up to a constant, in which case it suffices to estimate $z(\lambda)/z(\lambda_0)$ for any fixed $\lambda_0\in\Lambda$. We next review some popular methods for estimating this ratio.

Importance sampling. The ratio $z(\lambda,\lambda_0)\overset{\text{def}}{=}z(\lambda)/z(\lambda_0)$ can be expressed as $\mathrm{E}_{\lambda_0}[q(\boldsymbol{\theta};\lambda)/q(\boldsymbol{\theta};\lambda_0)]$, which is naturally estimated by the importance sampling estimate:

$$\hat{z}(\lambda,\lambda_0)=\frac{1}{S}\sum_{s=1}^{S}\frac{q(\boldsymbol{\theta}_s;\lambda)}{q(\boldsymbol{\theta}_s;\lambda_0)},\quad \{\boldsymbol{\theta}_s\}_{s=1}^{S}\overset{iid}{\sim}q(\boldsymbol{\theta};\lambda_0). \tag{3}$$

The efficiency of (3) may be improved using a multistage strategy. For a sequence $\lambda_0<\lambda_1<\ldots<\lambda_K=\lambda$, form the intermediate estimates

$$\hat{z}(\lambda_{j+1},\lambda_j)=\frac{1}{S}\sum_{s=1}^{S_j}\frac{q(\boldsymbol{\theta}_{js};\lambda_{j+1})}{q(\boldsymbol{\theta}_{js};\lambda_j)},\quad \{\boldsymbol{\theta}_{js}\}_{s=1}^{S_j}\overset{iid}{\sim}q(\boldsymbol{\theta};\lambda_j),$$

where the sample sizes S_j are allowed to differ. The multistage estimate $\hat{z}_{\mathrm{MS}}(\lambda_K,\lambda_0)$ of $z(\lambda_K,\lambda_0)$ is then

$$\hat{z}_{\mathrm{MS}}(\lambda_K,\lambda_0)=\prod_{j=0}^{K-1}\hat{z}(\lambda_{j+1},\lambda_j). \tag{4}$$

Bridge sampling. Even in its multistage form, importance sampling remains extremely sensitive to mismatch between the tails of $q(\cdot;\lambda_j)$ and $q(\cdot;\lambda_{j+1})$ due to the presence of the ratio $q(\cdot;\lambda_{j+1})/q(\cdot;\lambda_j)$ in (3). Bridge sampling (Meng and Wong, 1996) eliminates this ratio using a clever identity: for any integrable functions α_j,

$$\frac{z(\lambda_{j+1})}{z(\lambda_j)} = \frac{\mathrm{E}_{\lambda_j}\left(\alpha_j(\boldsymbol{\theta}) q(\boldsymbol{\theta}; \lambda_{j+1})\right)}{\mathrm{E}_{\lambda_{j+1}}\left(\alpha_j(\boldsymbol{\theta}) q(\boldsymbol{\theta}; \lambda_j)\right)}. \tag{5}$$

Taking the usual Monte Carlo estimates of the numerator and denominator yields the bridge sampling estimator, which may be substituted into the multistage estimator (4). Under appropriate assumptions, the variance of the bridge sampling estimate is minimized by choosing $\alpha_j^{\mathrm{opt}}(\cdot) = \frac{S_j z_j^{-1}}{\sum_k S_k z_k^{-1} q_k(\cdot)}$ (Meng and Wong, 1996; Shirts and Chodera, 2008).

For relatively low-dimensional or approximately Gaussian distributions, bridge sampling can be effective with a simple two stage scheme ($K = 1$). In many applications, we are interested in a particular value λ_1 such that $q(\boldsymbol{\theta}; \lambda_1)$ is an unnormalized density of interest (e.g. an unnormalized posterior distribution). Then $p(\boldsymbol{\theta}; \lambda_0)$ is chosen to be a multivariate normal with mean and covariance estimated based on an MCMC sample from $p(\boldsymbol{\theta}; \lambda_1)$ (see, e.g., Gronau et al., 2017), and (5) then allows us to estimate the normalizing constant of $q(\boldsymbol{\theta}; \lambda_1)$.

Rao-Blackwellization. For an increasing sequence $\{\lambda_j\}_{j=0}^K$, Carlson et al. (2016) note that $q(\boldsymbol{\theta}; \lambda_j)$ is proportional to some joint density function for $\boldsymbol{\theta}$ and λ. In this framework, we have that $z(\lambda_j) \propto \mathbb{P}(\lambda = \lambda_j)$, which leads to the Rao-Blackwellized estimate:

$$\Pr(\lambda = \lambda_j) \approx \frac{1}{S} \sum_{s=1}^{S} \frac{q(\theta_s; \lambda_j)}{\sum_{j'} q(\theta_s; \lambda_{j'})}. \tag{6}$$

We show in the appendix that this estimate is equivalent to multistage bridge sampling (5) when we take α_j to be an empirical approximation of the theoretical optimum α_j^{opt} defined above.

Non-equilibrium methods. The importance sampling and bridge sampling estimates requires $S_j > 1$ simulation draws from each $q(\boldsymbol{\theta}; \lambda_j)$. In non-equilibrium methods, we start with a simulation draw $\boldsymbol{\theta}_0$ from an "equilibrium" distribution $q(\boldsymbol{\theta}; \lambda_0)$, evolve it through a sequence of transitions that keeps $q(\boldsymbol{\theta} \mid \lambda_j), j = 1, \ldots, K$ invariant at each step, and collect one non-equilibrium trajectory $(\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{K-1})$. We do not draw from $q(\boldsymbol{\theta} \mid \lambda_j)$ directly, and $\boldsymbol{\theta}_j$ is in general not distributed as $q(\boldsymbol{\theta} \mid \lambda_j)$. We still obtain an unbiased estimate (Jarzynski, 1997; Neal, 2001):

$$\frac{z(\lambda_K)}{z(\lambda_0)} \approx \mathrm{E}\left(\exp \mathcal{W}(\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{K-1})\right), \quad \mathcal{W}(\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_{K-1}) = \log \prod_{j=0}^{k-1} \frac{q(\boldsymbol{\theta}_j \mid \lambda_{j+1})}{q(\boldsymbol{\theta}_j \mid \lambda_j)}. \tag{7}$$

where the expectation is over all initial draws and trajectories.

Path sampling. Gelman and Meng (1998) introduce a path sampling estimator for the normalizing constant problem derived from the following identity:

$$\frac{\partial}{\partial \lambda} \log z(\lambda) = \mathrm{E}\left(\frac{\partial}{\partial \lambda} \log q(\boldsymbol{\theta}; \lambda) \middle| \lambda \right). \tag{8}$$

Suppose that $\lambda_0 \in \Lambda$ and Λ is connected. Then, integrating this identity, we get that

$$\log \frac{z(\lambda)}{z(\lambda_0)} = \int_{\lambda_0}^{\lambda} \mathrm{E}\left(\frac{\partial}{\partial \lambda} \log q(\boldsymbol{\theta}; \lambda') \middle| \lambda' \right) d\lambda'. \tag{9}$$

Now let $\zeta(\lambda) = \log \frac{z(\lambda)}{z(\lambda_0)}$ and $U(\lambda) = \mathrm{E}\left(\frac{\partial}{\partial\lambda} \log q(\boldsymbol{\theta}; \lambda) \mid \lambda\right)$. The path sampling estimator $\hat{\zeta}(\lambda)$ is derived from this identity by taking a Riemann sum approximation to the integral. In particular, with $\{\lambda_{(i)}\}_{i=1}^{S}$ any increasing sequence in $[\lambda_0, \lambda] \subset \Lambda$, define

$$\hat{\zeta}(\lambda) = \sum_{i=1}^{S-1} \left(\lambda_{(i+1)} - \lambda_{(i)}\right) \left(\frac{\widehat{U}\left(\lambda_{(i+1)}\right) + \widehat{U}\left(\lambda_{(i)}\right)}{2} \right), \tag{10}$$

where $\widehat{U}\left(\lambda_{(i)}\right)$ is an estimate of $U(\lambda_{(i)})$. For $J \geq 1$, if we can sample

$$\{\boldsymbol{\theta}_{(ij)}\}_{j=1}^{J} \overset{iid}{\sim} p(\boldsymbol{\theta}; \lambda_{(i)}) \tag{11}$$

for each $i = 1, \ldots, S$, then it is natural to take as our estimate

$$\widehat{U}(\lambda_{(i)}) = \frac{1}{J} \sum_{j=1}^{J} \frac{\partial}{\partial\lambda} \log q\left(\boldsymbol{\theta}_{(ij)}; \lambda_{(i)}\right). \tag{12}$$

Compared to the previous approaches, path sampling enjoys a natural advantage from working on the log scale, allowing it to avoid potentially unstable ratios. Furthermore, all of the methods discussed above require the λ to be discrete or discretized. Path sampling, on the other hand, naturally applies to continuous λ and thus avoids the problem of discretization.

3. Continuous adaptive path sampling

We now develop an algorithm for the efficient application of the path sampling estimator (10) in cases where λ is a continuous variable and $(\boldsymbol{\theta}, \lambda)$ is sampled jointly from some distribution $p(\boldsymbol{\theta}, \lambda)$, as occurs in the continuous marginalization and tempering problems described in Section 1. In this context, we will be interested in the normalizing constant problem for the unnormalized family

$$q(\boldsymbol{\theta}; \lambda) \propto p(\boldsymbol{\theta} \mid \lambda). \tag{13}$$

In general, we will refer to joint distributions $p(\cdot, \cdot)$ and unnormalized families $\{q(\cdot; \lambda)\}_{\lambda \in \Lambda}$ satisfying (13) as being *matched*. When sampling jointly from such a matched distribution p, we will almost surely have only a single $\boldsymbol{\theta}$ such that $\boldsymbol{\theta} \sim p(\cdot \mid \lambda)$ for each value of λ in our sample. In the notation of the last section, we will almost certainly have $J = 1$.

3.1. Challenges of sampling from the joint distribution of θ and λ

Relative to conditional sampling (11), attempting to construct the path sampling estimator (10) from joint samples $\{(\boldsymbol{\theta}_{(i)}, \lambda_{(i)})\}_{i=1}^{S} \sim p(\boldsymbol{\theta}, \lambda)$ directly poses two challenges:

1. The quantities $\hat{U}$ are estimates of expectations conditional on λ. Since the sampled λ will almost certainly all be unique in continuous problems, it is not immediately clear how we can construct reliable estimates of these conditional expectations.

2. When sampling λ, rather than choosing the values of λ deterministically, subsets of Λ with low density will be poorly explored by the sampler, yielding unreliable estimates in these regions.

We circumvent the first challenge by constructing a one-shot path sampling estimator:

$$\hat{\zeta}(\lambda) = \frac{1}{2} \sum_{i=1}^{S-1} \left(\lambda_{(i+1)} - \lambda_{(i)}\right) \left(\frac{\partial}{\partial \lambda} \log q\left(\boldsymbol{\theta}_{(i+1)}; \lambda_{(i+1)}\right) + \frac{\partial}{\partial \lambda} \log q\left(\boldsymbol{\theta}_{(i)}; \lambda_{(i)}\right) \right). \tag{14}$$

When we sample $\boldsymbol{\theta}$ conditional on λ, we can reduce the error of (10) by decreasing the mesh size of $\{\lambda_{(i)}\}$ and increasing the conditional sample size J. By contrast, using joint samples of $(\boldsymbol{\theta}, \lambda)$ in the one-shot estimator, we can only increase the overall joint sample size S. This decreases the mesh size of the sampled $\lambda_{(i)}$, but will still almost surely leave us with a single $\boldsymbol{\theta}$ value corresponding to each λ, even for large S. We demonstrate in Section 4 that this still yields a nearly consistent estimator of the normalizing constant.

While one-shot estimation can leverage joint samples of $(\boldsymbol{\theta}, \lambda)$, the efficiency of the estimator may be threatened by poorly behaved marginal distributions $p(\lambda)$. While we can increase the number of marginal λ samples by increasing the joint sample size S, this may be inefficient if:

1. We are interested in low-probability regions of Λ, as often occurs when estimating highest density regions; or

2. The marginal distribution has properties that frustrate common sampling algorithms, e.g., the marginal density is multimodal.

More generally, when evaluating the one-shot path sampling estimator using samples from a matched joint distribution $p(\boldsymbol{\theta}, \lambda)$, we do not directly control for which values of λ we get estimates $\widehat{\zeta}(\lambda)$. Thus, if $\Lambda_0 \subset \Lambda$ is some region of interest, we may find the proportion of samples from Λ_0 to be too low for practical estimation.

By definition, if $p_1(\boldsymbol{\theta}, \lambda)$ is a matched distribution, and $p_2(\boldsymbol{\theta}, \lambda)$ is obtained from p_1 by changing only the marginal distribution of λ, then $p_2(\boldsymbol{\theta}, \lambda)$ is again matched. Thus, we can solve this problem if we can modify the marginal distribution of λ to place more probability over Λ_0. Specifically, let $p_1(\lambda)$ be the marginal distribution of $p_1(\boldsymbol{\theta}, \lambda)$ and $p^{\text{target}}(\lambda)$ be a preferable marginal distribution (e.g., one which is unimodal or which places more density on a subset Λ_0 of interest). Then

$$p_2(\boldsymbol{\theta}, \lambda) = p_1(\boldsymbol{\theta}, \lambda) \frac{p^{\text{target}}(\lambda)}{p_1(\lambda)} \tag{15}$$

has marginal distribution $p^{\text{target}}(\lambda)$.

In practice, however, it is usually not obvious how to implement the modification (15) in a sampling algorithm unless we already know the marginal density $p_1(\lambda)$. The fundamental idea of adaptation is to use the path sampling estimator (14) with $q(\boldsymbol{\theta}; \lambda) \propto p(\boldsymbol{\theta}, \lambda)$ to obtain a noisy estimate $\widehat{q}_1(\lambda)$ of $q_1(\lambda) \propto p_1(\lambda)$. Using $\widehat{q}_1$ in place of $p_1(\boldsymbol{\theta})$ in (15) will not yield a distribution with marginal exactly equal to p^{target}, but we can hope to approach such a distribution by iterating this update

$$q_{k+1}(\boldsymbol{\theta}, \lambda) = q_k(\boldsymbol{\theta}, \lambda) \frac{p^{\text{target}}(\lambda)}{\widehat{q}_k(\lambda)}, \tag{16}$$

where the initial $q_1(\boldsymbol{\theta}, \lambda)$ is just the unnormalized joint density $q(\boldsymbol{\theta}, \lambda)$, and $\widehat{q}_k(\lambda)$ is just the one-shot path sampling estimate of $q_k(\lambda)$ constructed from joint samples of $q_k(\boldsymbol{\theta}, \lambda)$.

In order to give the full details of this iteration, we introduce some additional notation. For any family of unnormalized densities $\{q(\boldsymbol{\theta}; \lambda)\}_{\lambda \in \Lambda}$, let

$$\widehat{U}_q(\boldsymbol{\theta}, \lambda) = \frac{\partial}{\partial \lambda} \log q(\boldsymbol{\theta}; \lambda).$$

Furthermore, let the function Sort_λ sort a sequence of tuples $\{(\boldsymbol{\theta}_i, \lambda_i)\}_{i=1}^S$ by the λ coordinate in increasing order, and let the function Smooth output a smooth function of λ which interpolates a sequence $\{(\lambda_i, \phi(\lambda_i))\}_{i=1}^S$. With these definitions, we can state the complete iteration in Algorithm 1.

Algorithm 1: Continuous Adaptive Path Sampling

Input: Unnormalized family $\{q(\boldsymbol{\theta}; \lambda)\}_{\lambda \in \Lambda}$, unnormalized density of matched distribution $q^{\text{match}}(\boldsymbol{\theta}, \lambda)$, target marginal p^{target}, joint sample size $S \geq 1$, number of iterations $K \geq 1$.

Output: Estimates $\left(\widehat{\zeta}\left(\lambda_{(i)}\right)\right)_{i=1}^S$.

Set $q_1^{\text{match}}(\boldsymbol{\theta}, \lambda) \leftarrow q^{\text{match}}(\boldsymbol{\theta}, \lambda)$;

Set cumulative samples $\mathcal{C} = \{\}$;

for $k \leftarrow 1, \ldots, K$ **do**

  Sample $\left\{\left(\boldsymbol{\theta}_{(i)}, \lambda_{(i)}\right)\right\}_{i=1}^S \overset{iid}{\sim} q_k^{\text{match}}(\boldsymbol{\theta}, \lambda)$;

  Update $\mathcal{C} \leftarrow \mathcal{C} \cup \left\{\left(\boldsymbol{\theta}_{(i)}, \lambda_{(i)}\right)\right\}_{i=1}^S$;

  Set $\left\{\left(\boldsymbol{\theta}_{(j)}, \lambda_{(j)}\right)\right\}_{j=1}^{kS} \leftarrow \mathsf{Sort}_\lambda\left(\mathcal{C}\right)$;

  for $j = 2, \ldots, kS$ **do**

    Estimate $\log \widehat{q}_k^{\text{match}}(\lambda_{(j)}) \leftarrow \sum_{i=1}^{j-1} \left(\frac{\lambda_{(i+1)} - \lambda_{(i)}}{2}\right) \left[\widehat{U}_{q_k^{\text{match}}}\left(\boldsymbol{\theta}_{(i+1)}, \lambda_{(i+1)}\right) + \widehat{U}_{q_k^{\text{match}}}\left(\boldsymbol{\theta}_{(i)}, \lambda_{(i)}\right)\right]$;

    Estimate $\widehat{\zeta}_k(\lambda_{(j)}) \leftarrow \sum_{i=1}^{j-1} \left(\frac{\lambda_{(i+1)} - \lambda_{(i)}}{2}\right) \left[\widehat{U}_q\left(\boldsymbol{\theta}_{(i+1)}, \lambda_{(i+1)}\right) + \widehat{U}_q\left(\boldsymbol{\theta}_{(i)}, \lambda_{(i)}\right)\right]$

  Set $\log \widehat{q}_k^{\text{match}}(\lambda) \leftarrow \mathsf{Smooth}\left(\{\log \widehat{q}_k^{\text{match}}(\lambda_{(j)})\}_{j=1}^S\right)$;

  Set $\log q_{k+1}^{\text{match}}(\boldsymbol{\theta}, \lambda) = \log q_k^{\text{match}}(\boldsymbol{\theta}, \lambda) - \log \widehat{q}_k^{\text{match}}(\lambda) + \log p^{\text{target}}(\lambda)$;

The last step of the outer loop of Algorithm 1 performs the update (16) on the log scale. Performing these computations on the log scale has two advantages: (i) computations involving sums are more numerically stable than those involving products and (ii) the path sampling estimator is already naturally defined on the log scale.

We accumulate our sampled points over each iteration, using the cumulative set to compute our path sampling estimates at each step. We use these sampled points to estimate conditional expectations over the unnormalized distributions $q_k^{\text{match}}(\boldsymbol{\theta} \mid \lambda)$. Since we only modify the marginal over λ at each iteration, we have

$$q_k^{\text{match}}(\boldsymbol{\theta} \mid \lambda) = q^{\text{match}}(\boldsymbol{\theta} \mid \lambda)$$

for all $k \geq 1$, so using the cumulative samples still yields valid estimates.

Most popular MCMC algorithms for sampling require a smooth (or at least continuous) unnormalized density as an input. The path sampling estimator, however, only produces a sequence of point estimates $\widehat{q}_k^{\text{match}}(\lambda_{(j)})$ of the marginal density $q_k(\lambda)$ at the sampled points $\lambda_{(j)}$. In order to use these point estimates to obtain a smooth update to the joint density for the next iteration, we first smoothly interpolate the points $(\lambda_{(j)}, \widehat{q}_k^{\text{match}}\left(\lambda_{(j)}\right))$. We achieve this by simply regressing $\widehat{q}_k^{\text{match}}\left(\lambda_{(j)}\right)$ on a sequence of smooth basis functions evaluated at the $\lambda_{(j)}$ (e.g., Gaussian or logistic kernels). In this case, the smoothing problem reduces to calculating and keeping track of a sequence of regression coefficients, which can be efficiently computed and stored.

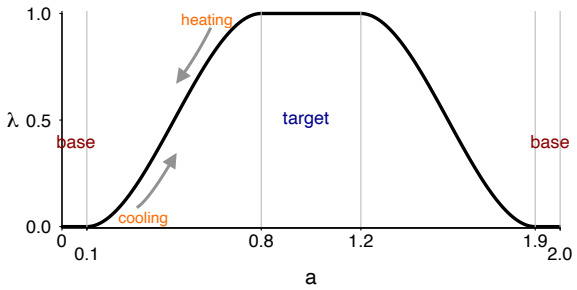

Figure 1: *The link function $\lambda = f(a)$. Because the inverse temperature λ is constant from 0.8 to 1.2, the continuous sampler generates exact draws from the target distribution when $0.8 < a < 1.2$.*

3.2. Adaptive path sampling for efficient tempering

A common approach to sampling from a multimodal posterior $\propto q(\boldsymbol{\theta} \mid \mathbf{y})$ starts with a broad and unimodal base distribution $p^{\text{base}}(\boldsymbol{\theta})$. Then an extended distribution is constructed with a geometric bridge as

$$q(\boldsymbol{\theta}, \lambda) = q(\boldsymbol{\theta} \mid \mathbf{y})^{\lambda} p^{\text{base}}(\boldsymbol{\theta})^{1-\lambda} \mathbb{1}_{[0,1]}(\lambda). \tag{17}$$

As λ traverses the interval $[0, 1]$, $q(\boldsymbol{\theta} \mid \lambda)$ varies smoothly between $p^{\text{base}}(\boldsymbol{\theta})$ and $q(\boldsymbol{\theta} \mid \mathbf{y})$, theoretically allowing the modes of the posterior to be connected through the unimodal base distribution. However, the marginal $p(\lambda)$ of this extended distribution usually cannot be calculated analytically, which prevents us from controlling it directly. Moreover, this marginal can exhibit near-vanishing probability on subsets of $[0, 1]$, essentially preventing any sampler from fully traversing the paths connecting the posterior modes.

Expressing this as a normalizing constant problem with $q(\boldsymbol{\theta}; \lambda) = q(\boldsymbol{\theta}, \lambda)$ yields

$$z(\lambda) = \int q(\boldsymbol{\theta}, \lambda) d\boldsymbol{\theta} \propto \int p(\boldsymbol{\theta}, \lambda) d\boldsymbol{\theta} = p(\lambda).$$

Thus, we can use our adaptive path sampling algorithm (Algorithm 1) with $q(\boldsymbol{\theta}, \lambda)$ as our initial matched distribution to iteratively estimate the marginal $p(\lambda)$ up to a constant. By taking a uniform distribution as our target marginal, we can ensure that after successful adaptation, the final matched distribution $q^{\text{match}}(\boldsymbol{\theta}, \lambda)$ satisfies both $q^{\text{match}}(\boldsymbol{\theta} \mid \lambda = 1) \propto p(\boldsymbol{\theta} \mid \mathbf{y})$ and $q^{\text{match}}(\lambda) = 1$, thus eliminating bottlenecks in the paths between posterior modes.

A final wrinkle in applying this procedure is that the posterior, the distribution from which we ultimately want to sample, is only obtained at the extreme point $\lambda = 1$ for which we may obtain no exact samples when sampling from $q(\boldsymbol{\theta}, \lambda)$ jointly. We overcome this problem with a simple smooth reparameterization of the temperature variable which allows us to recover the exact posterior over a subset of $[0, 1]$ with nonzero marginal probability. In particular, we take $\lambda = f(a)$, where $f(a)$ is a smooth function over $[0, 2]$ with image $[0, 1]$, and which is identically 0 and 1 on sets $[0, \delta] \cup [2 - \delta, 2]$ and $[1 - \delta, 1 + \delta]$ respectively, for some $\delta > 0$. Such a function can be easily constructed from piecewise polynomials, an example of which is given in Figure 1.

The adaptive path sampling tempering algorithm uses the initial matched distribution

$$q(\boldsymbol{\theta}, a) = q(\boldsymbol{\theta} \mid \mathbf{y})^{1-f(a)} p^{\text{base}}(\boldsymbol{\theta})^{f(a)}. \tag{18}$$

Algorithm 2 gives the steps of the path sampling Algorithm 1 adapted to this tempering problem. Because the marginal distribution is the normalizing constant of interest in this case, we drop the second path sampling step, and we ultimately output the samples which were drawn from the posterior rather than the path sampling estimate itself.

Algorithm 2: Path Sampling Adaptive Tempering

Input: Unnormalized posterior distribution $q(\boldsymbol{\theta} \mid \mathbf{y})$, base distribution $p^{\text{base}}(\boldsymbol{\theta})$, joint sample size $S \geq 1$, number of iterations $K \geq 1$.

Output: Samples $\{\theta_{(i)}\}$ from $p(\boldsymbol{\theta} \mid \mathbf{y})$.

Set $q_1^{\text{match}}(\boldsymbol{\theta}, a) \leftarrow q(\boldsymbol{\theta} \mid \mathbf{y})^{1-f(a)} p^{\text{base}}(\boldsymbol{\theta})^{f(a)}$;

Set cumulative samples $\mathcal{C} = \{\}$;

for $k \leftarrow 1, \ldots, K$ **do**

  Sample $\left\{\left(\theta_{(i)}, a_{(i)}\right)\right\}_{i=1}^{S} \overset{iid}{\sim} q_k^{\text{match}}(\boldsymbol{\theta}, a)$;

  Update $\mathcal{C} \leftarrow \mathcal{C} \cup \left\{\left(\theta_{(i)}, a_{(i)}\right)\right\}_{i=1}^{S}$;

  Set $\left\{\left(\theta_{(j)}, a_{(j)}\right)\right\}_{i=1}^{kS} \leftarrow \mathsf{Sort}_a\left(\mathcal{C}\right)$;

  for $j = 2, \ldots, kS$ **do**

    Estimate $\log \widehat{q}_k^{\text{match}}(a_{(j)}) \leftarrow \sum_{i=1}^{j-1} \left(\frac{a_{(i+1)} - a_{(i)}}{2}\right) \left[\widehat{U}_{q_k^{\text{match}}}\left(\theta_{(i+1)}, a_{(i+1)}\right) + \widehat{U}_{q_k^{\text{match}}}\left(\theta_{(i)}, a_{(i)}\right)\right]$;

  Set $\log \widetilde{q}_k^{\text{match}}(a) \leftarrow \mathsf{Smooth}\left(\{\log \widehat{q}_k^{\text{match}}(a_{(j)})\}_{j=1}^{S}\right)$;

  Set $\log q_{k+1}^{\text{match}}(\boldsymbol{\theta}, a) = \log q_k^{\text{match}}(\boldsymbol{\theta}, a) - \log \widetilde{q}_k^{\text{match}}(a)$;

return $\{\theta_{(j)} \mid a_{(j)} \in [1-\delta, 1+\delta]\}$;

4. Error bounds and convergence

In this section, we present an upper bound on the probability of error for the one-shot path sampling estimator when Λ is a compact interval. As a special case, we obtain a near-consistency result for the estimator.

Theorem 1. *Let $q(\boldsymbol{\theta}; \lambda)_{\lambda \in \Lambda}$ be a family of unnormalized densities, and let $q^{\text{match}}(\boldsymbol{\theta}, \lambda)$ be a possibly unnormalized matched joint density with normalized marginal $p(\lambda)$. In terms of these quantities, let $\zeta(\lambda) = \log z(\lambda)/z(\lambda_0)$ be the log normalizing constant for $\lambda_0 \in \Lambda$ a fixed reference point, and let $\widehat{\zeta}(\lambda)$ be the smoothed one-shot path sampling estimator computed by running Algorithm 1 for a single iteration. We make the following assumptions:*

1. *Λ is compact.*
2. *$\{(\boldsymbol{\theta}_{(i)}, \lambda_{(i)})\}_{i=1}^{S} \overset{iid}{\sim} p(\boldsymbol{\theta}, \lambda)$.*
3. *$\mathbb{E}\left(\frac{\partial}{\partial \lambda} \log q\left(\boldsymbol{\theta}; \lambda\right) \mid \lambda\right)$ is continuously differentiable in λ.*
4. *$\text{Var}\left(\frac{\partial}{\partial \lambda} \log q\left(\boldsymbol{\theta}; \lambda\right) \mid \lambda\right)$ is continuous in λ.*
5. *The smoothing operation in Algorithm 1 is performed by regressing on a finite set of continuously differentiable kernel functions, and the regression coefficients are bounded (which can be achieved, e.g., by clipping).*

Then, for each sample size $S \geq 1$ and parameters $\delta, \gamma \in (0, 1)$, we have for all $\epsilon > \gamma$,

$$\Pr\left(\sup_{\lambda \in \Lambda} \left|\zeta(\lambda) - \hat{\zeta}(\lambda)\right| \geq \epsilon\right) \leq C \left(\frac{1}{\sqrt{\epsilon'}} S^2 \left[1 - cp(\lambda_{\min}) \sqrt{\frac{\epsilon'}{S^{1+\delta}}}\right]^S + \frac{\epsilon'}{S^\delta}\right), \tag{19}$$

where $c, C > 0$ are constants independent of S and ϵ, $\lambda_{\min} = \arg\min_{\lambda \in \Lambda} p(\lambda)$, and $\epsilon' = \epsilon - \gamma$.

Theorem 1 immediately implies near-consistency for the one-shot path sampling estimator.

Corollary 1. *With the notation and assumptions of Theorem 1, we have for $\delta \in (0,1)$ and $\epsilon > \gamma$ that*

$$\Pr\left(\sup_{\lambda\in\Lambda}\left|\zeta(\lambda) - \hat{\zeta}(\lambda)\right| \geq \epsilon\right) = O\left(S^{-\delta}\right). \tag{20}$$

Proof. Taking logarithms and applying L'Hopital's rule directly shows that

$$S^2\left(1 - cp(\lambda_{\min})\sqrt{\frac{\epsilon'}{S^{1+\delta}}}\right)^S = o\left(S^{-\delta}\right),$$

which establishes the claim. □

Remarks

1. We only attain near-consistency since we must require $\epsilon > \gamma$ for some $\gamma > 0$. This γ parameter represents the error we incur from using a smooth parametric approximation to the normalizing constant when constructing the adaptive path sampling estimator. Using such an approximation allows us to extend the estimate over the entire set Λ and to perform the adaptation step in Algorithm 1. But because the true $\zeta(\lambda)$ function is unlikely to lie in the set of linear combinations of our smooth basis functions, we expect this approximation error to be nonzero. We can however shrink this approximation error by choosing more expressive collections of basis functions. In practice, we find this approximation error to be small enough to ignore.

2. Corollary 1 shows that, despite the one-shot estimation scheme we employ in our estimator, we come arbitrarily close to the $O(n^{-1})$ Chebyshev rate, which is the best rate possible given our moment assumptions.

This result is given in terms of the log normalizing constants $\zeta(\lambda)$. But if we define $\hat{z}(\lambda) = \exp\left(\widehat{\zeta}(\lambda)\right)$ and use the approximation $\exp(\epsilon) \approx 1 + \epsilon$ for ϵ small enough, we have

$$\begin{aligned}
&\Pr\left(\sup_{\lambda\in\Lambda}|\hat{z}(\lambda) - z(\lambda)| \geq \epsilon \sup_{\lambda\in\Lambda} z(\lambda)\right) \\
&\quad\leq \Pr\left(\sup_{\lambda\in\Lambda}\left|\frac{\hat{z}(\lambda)}{z(\lambda)} - 1\right| \geq \epsilon\right) \\
&\quad\leq \Pr\left(\sup_{\lambda\in\Lambda}\frac{\hat{z}(\lambda)}{z(\lambda)} \geq 1+\epsilon\right) + \Pr\left(\inf_{\lambda\in\Lambda}\frac{\hat{z}(\lambda)}{z(\lambda)} \leq 1-\epsilon\right) \\
&\quad\approx \Pr\left(\sup_{\lambda\in\Lambda}\frac{\hat{z}(\lambda)}{z(\lambda)} \geq \exp(\epsilon)\right) + \Pr\left(\inf_{\lambda\in\Lambda}\frac{\hat{z}(\lambda)}{z(\lambda)} \leq \exp(-\epsilon)\right) \\
&\quad= \Pr\left(\sup_{\lambda\in\Lambda}\left[\hat{\zeta}(\lambda) - \zeta(\lambda)\right] \geq \epsilon\right) + \Pr\left(\inf_{\lambda\in\Lambda}\left[\hat{\zeta}(\lambda) - \zeta(\lambda)\right] \leq -\epsilon\right) \\
&\quad\leq 2\Pr\left(\sup_{\lambda\in\Lambda}\left|\zeta(\lambda) - \hat{\zeta}(\lambda)\right| \geq \epsilon\right).
\end{aligned}$$

Thus, Theorem 1 also applies to the additive error of the normalizing constant at the expense of incurring a multiplicative factor of $\sup_{\lambda\in\Lambda} z(\lambda)$.

While Corollary 1 shows that the $S^{-\delta}$ term in Theorem 1 dominates asymptotically, the first term can easily dominate in the pre-asymptotic regime. In particular, the rate at which this first term tends to 0 is determined by the constant $c\sqrt{\epsilon'}p(\lambda_{\min})$, with smaller values leading to slower convergence. Since c is essentially controlled by

$$\left(\sup_{\lambda\in\Lambda}\frac{\partial}{\partial\lambda}\zeta(\lambda)\right)^{-1}$$

and $\zeta(\lambda)$ is on the log scale, it is not hard to construct examples in which $p(\lambda_{\min})$ dominates in order of magnitude (e.g., in multimodal marginals $p(\lambda)$ with well-separated modes). In such cases, tiny values of $p(\lambda_{\min})$ can bottleneck our convergence toward the asymptotic regime in which the $S^{-\delta}$ term dominates. Furthermore, $p(\lambda)$ can be modified without altering the values of $\zeta(\lambda)$ which we wish to estimate. These observations justify the adaptation scheme in Algorithm 1. When Λ is compact, we can take $p^{\text{target}}(\lambda)$ to be uniform, and successful adaptation will then eliminate troughs in the marginal distribution $p(\lambda)$, increasing $p(\lambda_{\min})$. We can thus describe the adaptation step as accelerating convergence towards the asymptotic regime, where our estimator obtains near-optimal efficiency. In the next section, we present a series of simulation experiments demonstrating that adaptation can often converge in just a few iterations with only a moderate sample size per iteration.

5. Simulation experiments

We now turn to a series of experiments which demonstrate the advantages of our adaptive one-shot path sampling scheme for continuous normalizing constant estimation problems. In Section 5.1, we use a conjugate model with known normalizing constant to compare the accuracy of different normalizing constant estimators. In Section 5.2, we show that the path sampling tempering algorithm described in Section 3.2 is scalable to high dimensional multimodal posteriors. In Section 5.3 and 5.4, we test the computational efficiency of the proposed method for sampling difficult distributions, demonstrating higher effective sample sizes and faster mixing times compared to alternative methods.

5.1. Beta-binomial example: Comparing accuracy of estimates of the normalizing constant

We start with an example adapted from Betancourt (2015), in which the true normalizing constant can be evaluated analytically. Consider a model with a binomial likelihood, $y \sim \text{binomial}(\theta, n)$, and a beta prior, $\theta \sim \text{beta}(\alpha, \beta)$. We then construct a geometric bridge between the unnormalized posterior and the prior, yielding an unnormalized density of the form

$$q(\theta, \lambda) = \text{binomial}(y|\theta, n)^{\lambda}\text{beta}(\theta|\alpha, \beta), \quad \theta \in [0, 1],\ \lambda \in [0, 1].$$

By conjugacy, the normalized conditional distributions have closed forms

$$p(\theta|\lambda) = \text{beta}(\lambda y + \alpha, \lambda(n - y) + \beta),$$

and the normalizing constants $z(\lambda) \propto p(\lambda)$ are given by

$$z(\lambda) = \int_0^1 q(\theta, \lambda)d\theta = \left(\frac{\Gamma(n + 1)}{\Gamma(y + 1)\Gamma(n - y + 1)}\right)^{\lambda}\frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)}\frac{\Gamma(\lambda y + \alpha)\Gamma(\lambda(n - y) + \beta)}{\Gamma(\lambda n + \alpha + \beta)}.$$

Using this closed form ground truth, we compare four sample-based estimates of the log normalizing constant / marginal distribution. Each method has the form of the iterative update (16) but using different estimators in the construction of $\widehat{q}_k(\lambda)$. These estimators are:

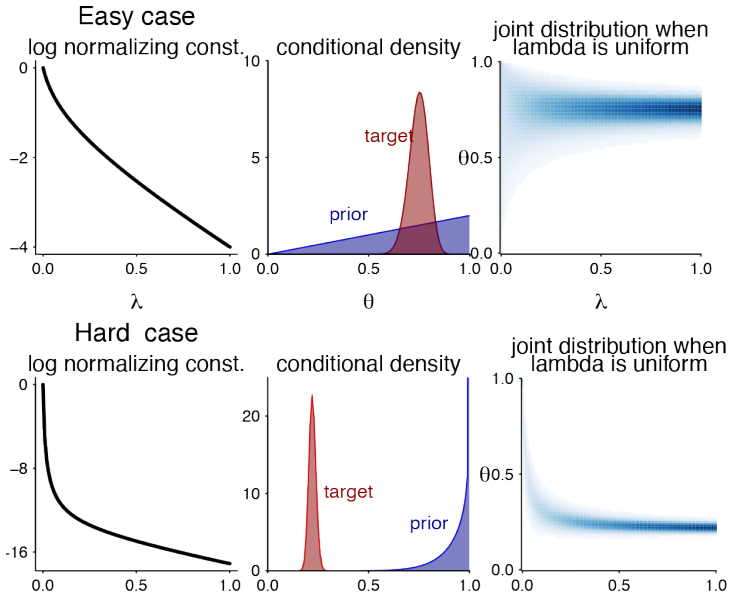

Figure 2: *For the easy (top) and hard (bottom) cases: (a) the analytic log normalizing constant. (b) the base and the target distributions of θ, (c) the joint distribution of (θ, λ) when the marginal of λ is uniform. In the hard case the target and base distributions are separated and the log normalizing constant changes rapidly.*

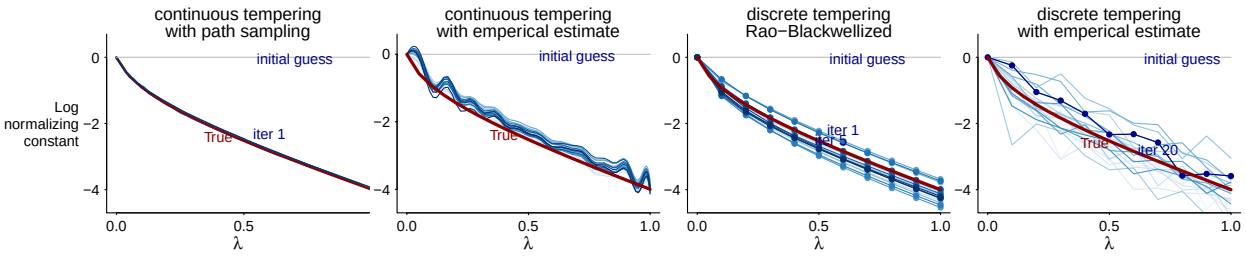

Figure 3: *Estimates of the log normalizing constant in the easy beta-binomial example across first 20 adaptation iterations. All methods eventually approximate the true function, while continuous tempering with path sampling achieves the fastest convergence rate.*

1. The one-shot path sampling estimator, yielding the proposed CAPS method.

2. A continuous approximation constructed using a kernel density estimate $\widehat{q}_k(\lambda)$ rather than a path sampling estimate. We refer to this method as continuous tempering with an empirical estimate.

3. A discretized estimate, for which we restrict λ to a finite grid $0 = \lambda_0 < \lambda_1 < \cdots < \lambda_D = 1$. In this case, at each iteration k, we generate S samples $\{(\theta_{(s)}, \lambda_{(s)})\}_{s=1}^S$ from the discretized joint distribution proportional to $q(\theta, \lambda)$ and construct empirical estimates of the marginal probabilities:

$$\widehat{q}_k(\lambda_d) = \frac{1}{S} \sum_{s=1}^{S} \mathbf{1}\left\{\lambda_{(s)} = \lambda_d\right\}, \text{ for } 0 \leq d \leq D. \tag{21}$$

   We refer to this method as discretized tempering with an empirical estimate.

4. A discretized estimate using the Rao-Blackwellized estimator (6) of the marginal probabilities rather than the empirical estimator (21). As noted in Section 2, this is equivalent to using the bridge sampling estimator (5) with an estimate of the optimal bridge function $\alpha^{\text{opt}}(\cdot)$. We refer to this method as discretized tempering with a Rao-Blackwellized estimate.

Easy case: approximately linear $\log p(\lambda)$

We begin by setting $\alpha = 2$, $\beta = 1$, $y = 60$, and $n = 80$ (left half of Figure 2). Figure 3 presents the log normalizing constant estimates in the first 20 adaptation iterations. All methods start with a flat initial guess $\log z(\lambda) = 0$. For the first two methods that use a continuous temperature variable, we sample 3000 joint (λ, θ) draws in each adaptation iteration and discard the first half as warmup. We also use a length $I = 100$ grid for our parametric basis function approximation. For the methods using a discrete λ, we choose an evenly spaced grid $\lambda_d = (d-1)/10, d = 1, \ldots, 11$ of temperatures in $[0, 1]$. We then draw 150 λ values from this set per adaptation iteration, each followed 100 HMC updates in θ (discarding warmup samples as before). These numbers ensure the continuous tempering has a smaller computation cost (3000 joint draws) than in discrete implementations (100 HMC updates on θ $\times$ 150 updates on λ_d) per adaptation iteration.

We find that all four methods approach the true marginal / normalizing constant (the red curve in the first row of Figure 3) after sufficient adaptation iterations. The proposed CAPS algorithm converges most rapidly, nearly recovering the ground truth after a single adaptation iteration.

Hard case: highly nonlinear $\log p(\lambda)$

Moving to a harder case, we next take hyperparameters, $\alpha = 9$, $\beta = 0.75$, $k = 115$, and $n = 550$ (right half of Figure 2). Now, the base ($\lambda = 0$) and target ($\lambda = 1$) are two isolated spikes, and the log normalizing constant changes rapidly, especially for small λ. Figure 4 compares the same four log normalizing constant estimation methods through 20 adaptation iterations. Continuous tempering with path sampling achieves nearly exact marginal recovery in 8 adaptation iterations. The Rao-Blackwellized estimator also achieves nearly exact recovery at all points on its discrete grid after 13 adaptation iterations. However, the estimate remains unstable in subsequent iterations due to discretization error (particularly visible for small λ where the evenly-spaced grid struggles to resolve the rapidly varying $\log p(\lambda)$). Neither of the methods using empirical estimates are able to achieve near-recovery prior to termination.

Figure 5 summarizes the relative performance of these methods, displaying the L_2 error of the $\log p(\lambda)$ estimates. Only continuous tempering with path sampling achieves a monotonically

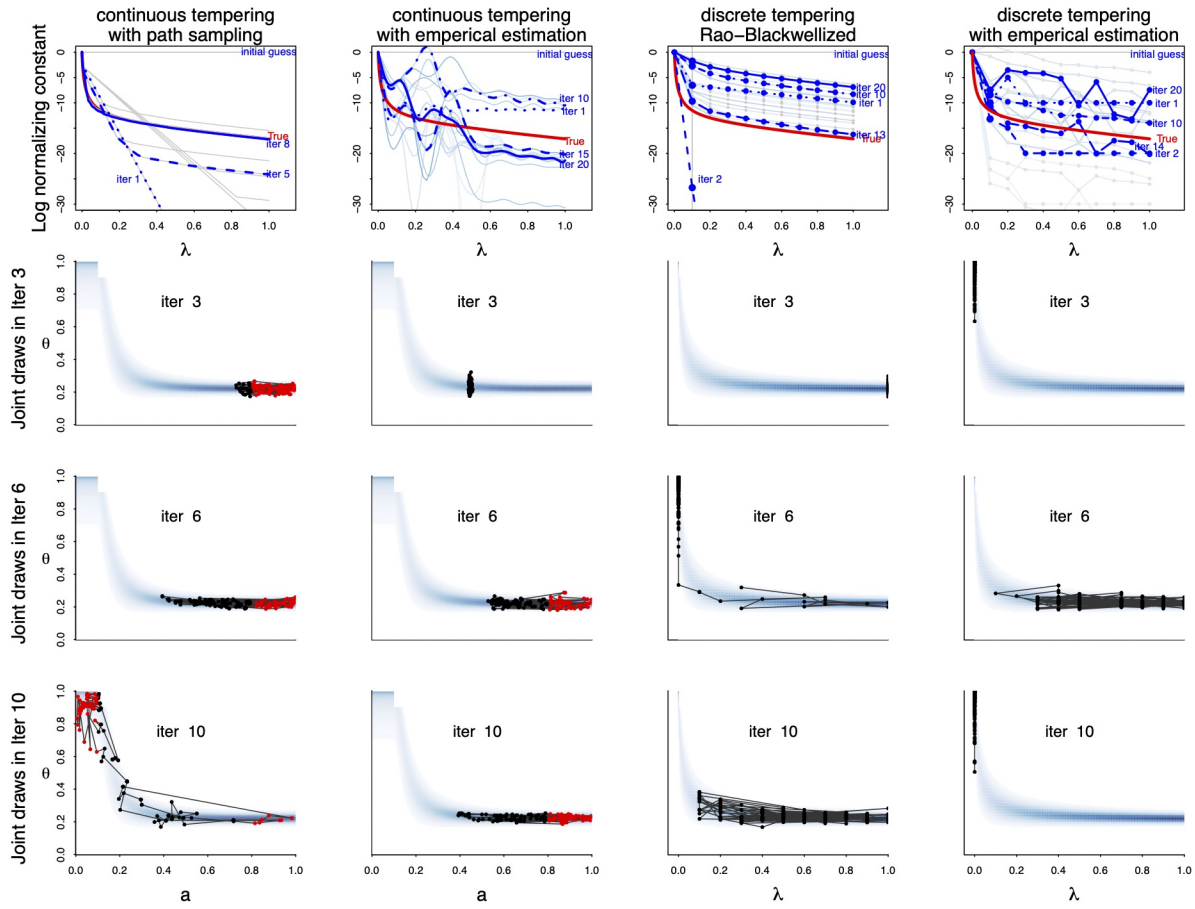

Figure 4: *Comparison of four tempering methods in the hard beta-binomial example. Row 1: Starting from a flat guess, only the proposed method converges to the the true log normalizing constant (red curve) after 8 adaptations. Rows 2–4 compare the first 150 draws of the joint distribution of parameter θ and temperature λ in adaptations 3, 6, and 10. The proposed method fully explores the the joint space efficiently, while the two discrete schemes exhibit random walk behavior in temperature updates, and cannot adapt to the rapid changing regions near $\lambda = 0$.*

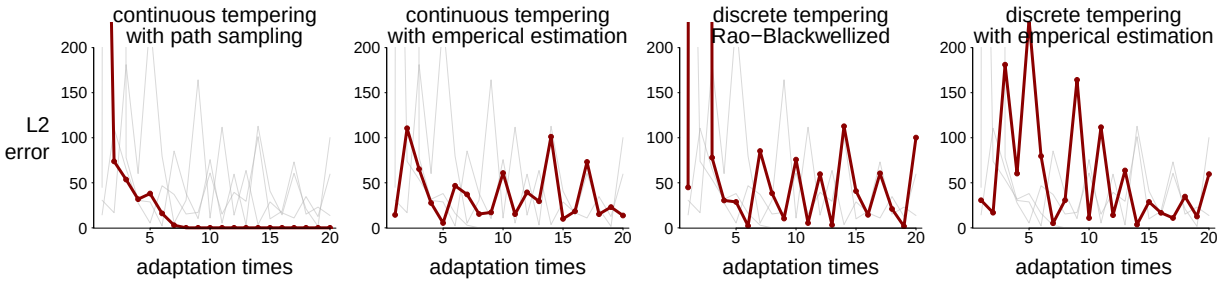

Figure 5: *L_2 errors of estimates of the log normalizing constant $\log z(\lambda)$ from the four methods considered during the first 20 adaptation iterations. Only continuous tempering with path sampling gives a monotonically decreasing error which shrinks to zero in practical amount of time.*

decreasing error with number of adaptations. Rows 2–4 in Figure 4 show the first 150 joint draws in adaptation iterations 3, 6, and 10 from all four methods. The proposed path sampling method fully explores the joint distribution in adaptation 10. By comparison, we observe that while the empirical continuous tempering method may eventually explore the whole joint distribution, it adapts at a much slower rate. On the other hand, both discrete tempering procedures struggle to move between $\lambda = 0$ and $\lambda = 0.1$ since the conditional distributions $p(\boldsymbol{\theta} \mid \lambda)$ at $\lambda = 0$ and $\lambda = 0.1$ differ significantly.

5.2. Sampling from Gaussian mixtures: Comparing accuracy of multimodal sampling

Next we consider the problem of sampling from Gaussian mixtures. For visual illustration we begin with the simple case of a mixture of two one-dimensional Gaussians. We use equal weighting for the two modes, taking their standard deviations small enough that HMC and other MCMC algorithms cannot efficiently move between them. We then apply continuous tempering with path sampling using a geometric bridge and a normal base distribution centered between the two modes and with standard deviation large enough to encompass both modes of the mixture distribution.

Figure 6 shows five out of ten total adaptation iterations for which we ran our tempering algorithm. At initialization, the joint sampler can only sample points around the base distribution. After each adaptation, we generate more joint samples near, and ultimately within, the target temperature region $0.8 < a < 1.2$, for which the corresponding θ samples are draws from the target mixture distribution (first row). This is also reflected in a more accurate estimate of the log normalizing constant (second row) and in flatter marginal distributions on the temperature after each iteration (third row). After four adaptations, the sampler has fully explored $a \in [0, 2]$. Selecting those θ for which $f(a) = 1$ at this stage recovers samples from the target distribution (fourth row).

Next we consider harder 10-component mixtures of Gaussians. The mixture components are given diagonal covariance matrices with constant diagonal elements all equal to one tenth the minimum Euclidean distance between any two of the component centers. This ensures that all modes of the mixture distribution are separated by regions of vanishingly small probability density. We construct these multimodal mixtures in a range of dimensions from 10 to 100, with the number of components fixed at 10, and run continuous tempering with path sampling (again using a geometric bridge). We compare our proposed tempering algorithm with two benchmark algorithms:

1. Discrete tempering with the Rao-Blackwellized bridge sampling estimate, as described in the previous beta-binomial experiment.

2. The continuous tempering algorithm proposed in Graham and Storkey (2017), which constructs the normalizing constant / marginal estimate $\widehat{q}(\lambda)$ as a log-linear approximation to the true $q(\lambda)$. The only free parameter in this log-linear approximation is an estimate ζ of the true normalizing constant of the target distribution (in this case, the Gaussian mixture). In order to achieve a fair comparison, we initialize this algorithm by setting ζ to the true target normalizing constant. Consequently, unlike the other two algorithms, this method is *not* run within an adaptive iteration.

We do not include the discrete and continuous tempering algorithms using empirical normalizing constant estimates, since these were to seen have substantially lower sample efficiency in the previous section. For all methods, we use the same over-dispersed (unimodal) normal base distribution.

In order to assess whether a given tempering procedure has succeeded, we assign to each draw the label of the component with the closest center. Because the mixture components are assigned

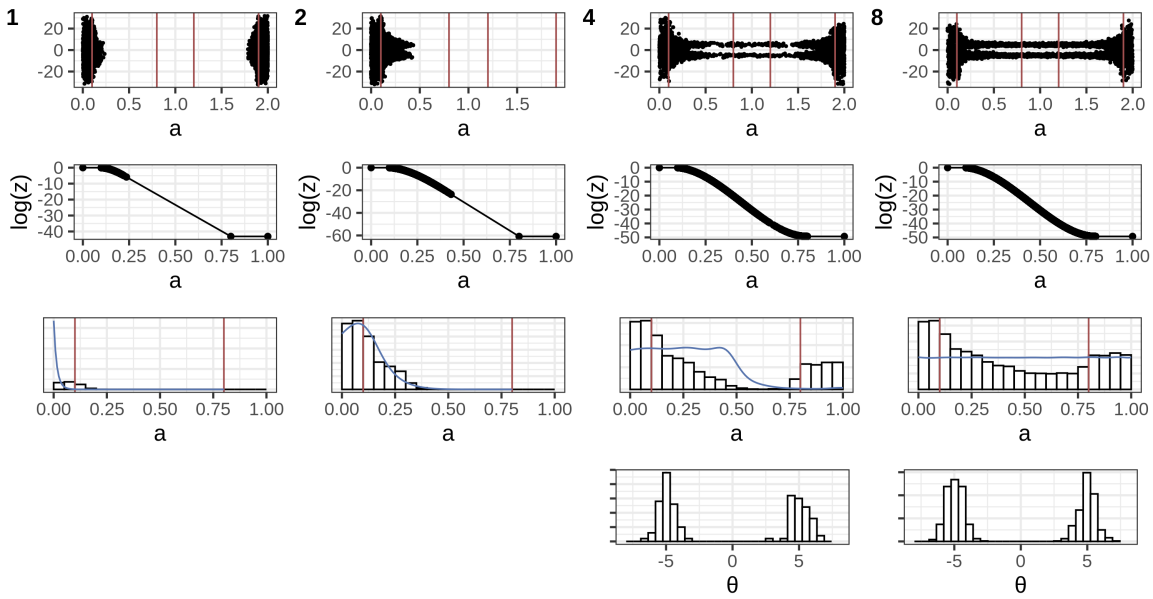

Figure 6: *Tempering using path sampling for a Gaussian mixture target distribution, plotting adaptations 1, 2, 4, 6, and 10. The first row shows the joint simulation draws; the second row displays the estimate of the log normalizing constant; the third row displays the marginal density of and cumulative draws of the temperature variable; the fourth row shows the draws from the target density in the later adaption iterations.*

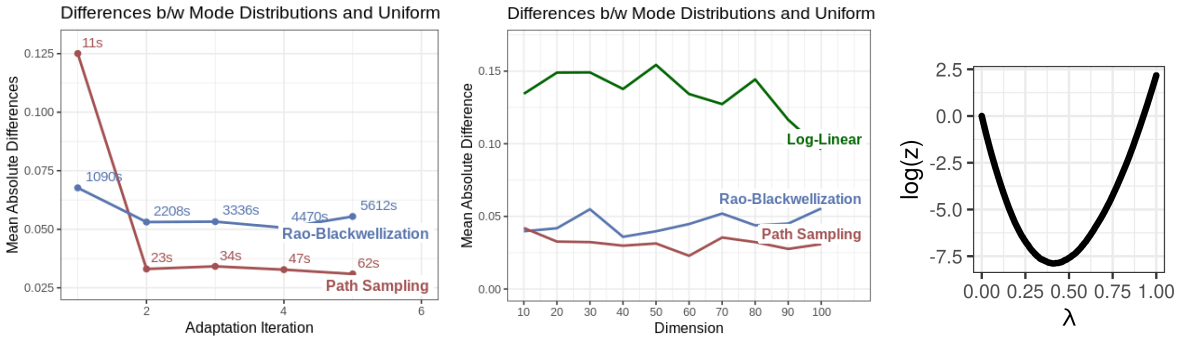

Figure 7: *Mean absolute error between empirical probabilities of component labels and desired uniform distribution for the Gaussian mixture model example. Errors are shown for CAPS-based t empering, and two benchmarks: continuous tempering with log-linear normalizing constant discrete tempering with Rao-Blackwellized bridge sampling estimate. Results are averages over 5 repeated runs. Left: Sampling error and time at each adaptation for a Gaussian mixture target with 100 dimensions and 10 components. Numerical labels indicate the number of seconds of CPU time before each adaptation iteration completed. Middle: Comparison of sampling errors as dimensions range from* $d = 10$ *to* 100*. The log-linear tempering procedure is supplied with true normalization constant while the other two are with uniform initialization. Right: The estimated log normalizing constant* $\log z(\lambda)$ *from path sampling.*

equal weight and are well-separated by regions of low probability, we expect the draws to be nearly uniformly distributed between across the ten component labels if the sampler has appropriately converged. The results are averaged over five independent runs for each of the three tempering methods.

Figure 7 displays the mean absolute difference (MAD) between the observed component label proportions and the ideal uniform probabilities (in this case 1/10) for each algorithm. The left panel displays the evolution of the MAD over adaptation iterations for the two adaptive algorithms in the 100-dimensional case, with each point labeled with the cumulative CPU time. The middle panel compares the MAD across the tested dimensions for all three algorithms (using only the samples from the fifth and final adaptation iteration for the two adaptive algorithms). We find that, in all but the smallest dimension tested, the component label distribution is closest to uniform for our proposed CAPS method on average. While the discrete Rao-Blackwellized method achieves only slightly worse MAD performance, it requires substantially longer computation time to perform even a single adaptation iteration. This is largely due to its use of an HMC-in-Gibbs sampling scheme, which is required to handle the discrete temperature parameters. The worse performance of the log-linear algorithm can be explained by the substantially non-linear and non-monotone form of the true log normalizing constant, which is shown in the right panel of Figure 7.

In Figure 7, we do not see the MAD degrade with increasing dimension for path sampling. This can be explained by the fact that the log normalizing constant itself scales mildly with the dimension in Gaussian mixtures. It general, the number of dimensions is usually not the limiting factor for path sampling, though large dimensions could exacerbate problems induced by severe base-target mismatch. We further discuss dimension scalability in Appendix D.

5.3. Flower target: A comparison of computation efficiency

Next we consider sampling from a flower-shaped distribution, which has been previously used as a test case in Sejdinovic et al. (2014) and Nemeth et al. (2019) due to its challenging, non-convex geometry. This is a two-dimensional distribution with probability density spread throughout multiple "petals." Its probability density function is given by

$$p(x, y \mid \sigma, r, A, \omega) \propto \exp\left(-\frac{1}{2\sigma^2}\left(\sqrt{x^2+y^2} - r - A\cos(\omega \arctan(y, x))\right)\right).$$

Between the petals, the probability density is pinched into narrow regions, slowing exploration of the target for most MCMC samplers, including HMC.

Figure 8 plots samples from the flower distribution with 6 petals generated by standard HMC (left) and by our proposed CAPS tempering algorithm (right). As before, we use a normal base distribution, and both algorithms generated roughly the same number of samples from the target distribution. While standard HMC fails to adequately explore all of the petals of the distribution, the CAPS tempering scheme succeeds in generating draws from all petals in roughly equal proportions, as desired.

We also compare these algorithms on the basis of the mixing time of the corresponding Markov chains. Figure 9 plots the computed $\hat{R}$ (Vehtari et al., 2021) for the x and y coordinates of a flower distribution with 40 petals against computational time. Compared to standard HMC, the CAPS-based tempering algorithm crosses the conventional $\hat{R} = 1.05$ threshold for adequate mixing in about a quarter of the time (including the time for adaptation iterations). Thus, despite the much lower time cost per sample for standard HMC (due to the lack of adaptation), path sampling still manages to be more efficient in terms of total mixing time. We expect this disparity to increase with the complexity and non-convexity of the target distribution's geometry.

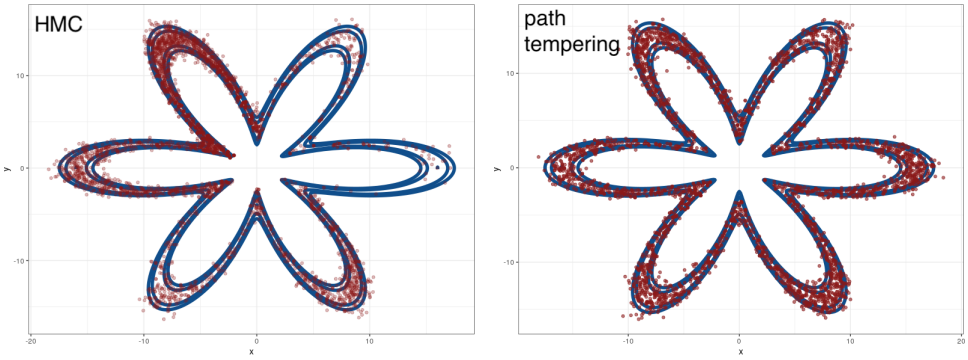

Figure 8: *Draws (red dots) from the flower target generated generated by standard HMC (left) and CAPS-based tempering (right). The blue curves show contours of the target density.*

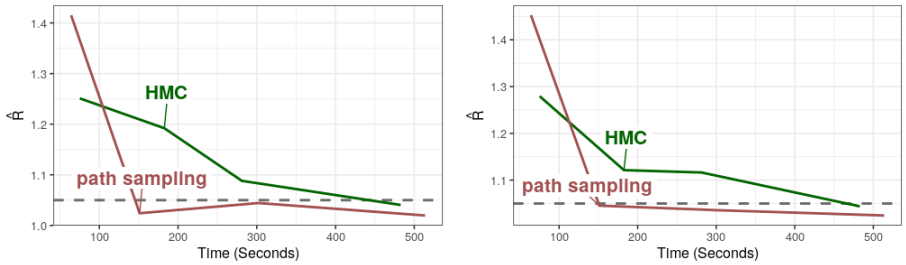

Figure 9: *Comparison of computational cost of CAPS-based tempering versus standard HMC for a flower distribution with 40 petals. The x-axis displays the time in seconds, and the y-axis shows the achieved $\widehat{R}$-value for the x coordinate (left) and y coordinate (right) of the flower distribution. The dashed gray line displays the 1.05 cutoff below which we consider our chains to have mixed sufficiently. All results are averaged over 15 replications.*

5.4. Regularized horseshoe regression: Expanding models continuously and efficiently

We now turn to an example of path sampling tempering in which we use a geometric bridge to connect two different posterior distributions. In this case, the posteriors in question do not exhibit multimodality or other challenging geometry. Rather, the motivation for employing the geometric bridge is to allow us to sample both models simultaneously with the hope of improving computational efficiency compared to sampling the two models separately. The mixing time of HMC scales polynomially with dimension, and thus it is reasonable to expect that fitting one slightly larger model, with a single additional parameter for temperature, may be computationally cheaper than fitting two models separately.

Specifically, we consider a regression model with a regularized horseshoe prior (Piironen and Vehtari, 2017a,b), an effective tool for sparsifying Bayesian regression inferences. Let $\mathbf{y} \in \mathbb{R}^n$ be a binary outcome vector, $\mathbf{X} \in \mathbb{R}^{n \times D}$ be a matrix of D predictors, and $\boldsymbol{\beta} \in \mathbb{R}^D$ be a coefficient vector. The regularized horseshoe prior on $\boldsymbol{\beta}$ is given as follows:

$$\begin{aligned} \beta_d \mid \tau, \zeta, \gamma &\sim \text{normal}\left(0,\, \gamma\zeta_d(\gamma^2 + \tau^2\zeta_d^2)^{-1/2}\right) \\ \tau &\sim \text{Cauchy}^+\left(0,\, 2/((D-1)\sqrt{n})\right) \\ \zeta_d &\sim \text{Cauchy}^+(0,1),\ d = 1, \ldots, D. \end{aligned}$$

We then consider binary regression models with the logit link,

$$\Pr(y_i = 1 \mid \boldsymbol{\beta}, \text{logit}) = \text{logit}^{-1}\left(\sum_{d=1}^{D} \beta_d x_{id}\right)$$

and the probit link,

$$\Pr(y_i = 1 \mid \boldsymbol{\beta}, \text{probit}) = \Phi\left(\sum_{d=1}^{D} \beta_d x_{id}\right).$$

Our final model is given by constructing a geometric bridge between the logit and probit models, resulting in the joint model,

$$p(a, \beta, \tau, \zeta, \gamma) \propto \prod_{i=1}^{n} \left(\Pr(y = y_i|\boldsymbol{\beta}, \text{logit})^{1-f(a)} \Pr(y = y_i|\boldsymbol{\beta}, \text{probit})^{f(a)}\right) p^{\text{prior}}(\boldsymbol{\beta}, \tau, \zeta, \gamma), \tag{22}$$

where $p^{\text{prior}}(\beta, \tau, \zeta, \gamma)$ is the regularized horseshoe prior given above.

In the first experiment, we generate $n = 40$ data points and $D = 100$ covariates with a maximum pairwise correlation 0.5. We take the regression coefficients for the first three covariates to be nonzero and set $\beta_d = 0$ for all $4 \leq d \leq 100$. Furthermore, the sampling distribution is chosen to have a logit link in the true data generating process. We run our CAPS-based tempering procedure for two adaptation iterations with this simulated data plugged into the unnormalized density (22). We take $S = 500$ draws in the first adaptation iteration and $S = 3000$ in the second.

Using the link function $f(a)$, we can recover exact draws from the two posteriors corresponding to the logistic and probit link by extracting those samples for which a lies in certain subsets of $[0, 2]$ (as discussed in Section 3.2). We investigate the sensitivity of our inference to specification of the link $f(a)$ by varying the proportion of the interval $[0, 2]$ which yields exact samples from the logit/probit models. If this proportion is too small, our efficiency may suffer due to most samples landing "between" the two target posteriors, and hence being discarded in the final output. If this proportion is too high, however, our efficiency may also suffer due to a relatively rapid transition between

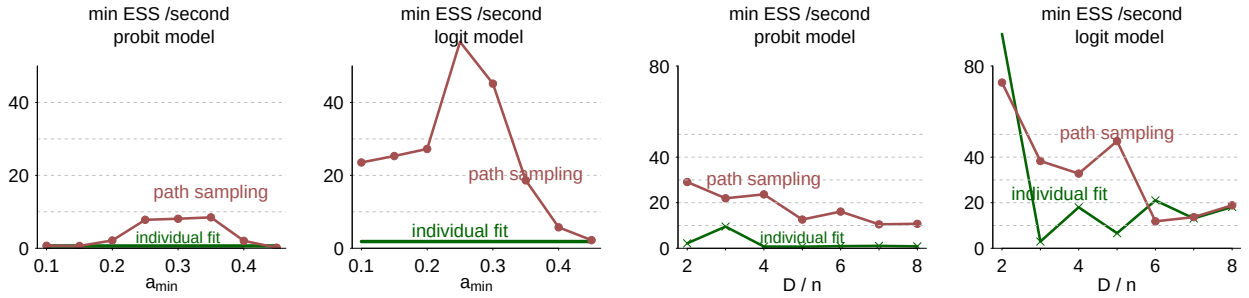

Figure 10: *The smallest computed tail effective sample size per second (ESS/sec) among all regression coefficients in the horseshoe regression example with both Bernoulli logit and probit likelihood. The green line shows the minimum ESS/sec for the two models fit separately, while the red line shows the minimum ESS/sec for the geometrically bridged logit-probit model sampled using CAPS (where the time calculation includes time for adaptation). The left two panels show varying* $a_{\min}$*, i.e. varying the proportion of joint samples from the bridged model which are exact samples from the logit or probit models. The right two panels show varying the input dimension/sample size, fixing* $a_{\min} = 0.35, a_{\max} = 0.65$*. In most cases, the CAPS-derived samples achieve higher tail ESS/sec for both probit and logit models.*

the two target posteriors across $a \in [0, 2]$ and correspondingly difficult joint temperature-posterior geometry.

For the separate logit and probit target samples, we measure this sampling efficiency by dividing the tail effective sample size (tail-ESS; Vehtari, Gelman, Simpson, Carpenter, and Bürkner, 2021) by the total sampling time. We denote this effective sample size per second as ESS/sec for the two posteriors of interest. For comparison, we also fit these two models separately and compute the same ESS/sec metric. To ensure reliable inference for all parameters, we report the smallest ESS/sec among all regression coefficients β_d. As seen in the left two panels of Figure 10, the CAPS-based tempering algorithm yields a substantially larger ESS/sec than fitting the two models separately as long as we construct the link $f(a)$ appropriately.

In the second experiment, we fix the link function $f(a)$, and vary the dimension $D = n\rho, 2 \leq \rho \leq 8$. We average the resulting simulations over 10 repeated runs. Again we see in the bottom two panels in Figure 10 that the CAPS-based tempering approach usually yields better ESS/sec than separate fits across a range of dimensions. However, a joint distribution interpolating two arbitrary models will not be more efficient than separate fits in all cases or by all metrics. For instance, *averaging* the ESS/sec over all coefficients β_d (rather than taking the minimum as in Figure 10), we find that the mean ESS/sec in path sampling is smaller than individual fits in this example, which can be viewed as a parameter-wise efficiency-robustness tradeoff.

6. Discussion

6.1. Selecting the base distribution

In Section 3.2, we merely require the base distribution p^{base} to have the same support as the target q. When q is an unnormalized posterior for a given model, a natural candidate for p^{base} is the model prior. Ideally, the base distribution should achieve a balance between being easy to sample from and being sufficiently close to the target distribution. This is not a unique challenge in our method, as all (discrete) tempering and, more generally, importance sampling methods are sensitive to the choice of base or proposal distribution. Our approach treats the base measurement as a

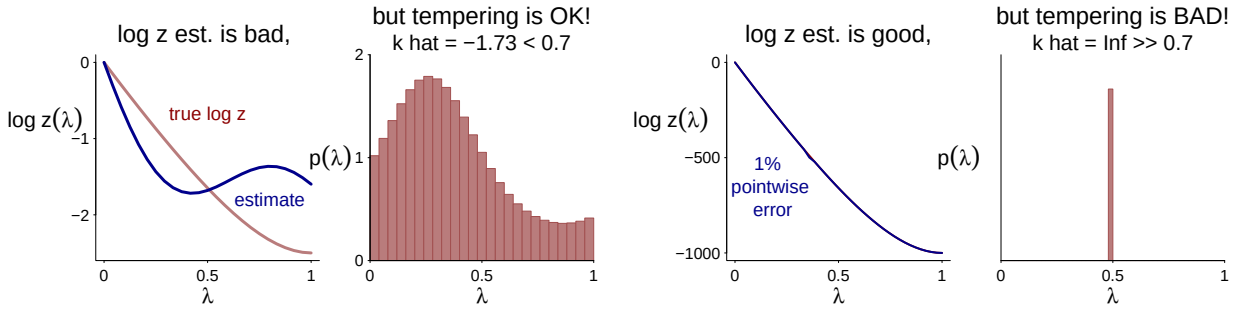

Figure 11: *Two perspectives on normalizing constant estimation and simulated tempering. In the left two panels, the blue curve poorly fits the true log normalizing constant, but there is enough mass everywhere in the resulting marginal density to ensure complete exploration of the temperature space simulated tempering. The right two panels: with a much larger scale, when log z estimation is off by 1% at a single point, the resulting path becomes a point mass in* λ.

user-specified input. It may be possible to automate this task while achieving better performance by borrowing ideas from the literature on constructing and optimizing the proposal distribution in adaptive importance sampling (Geweke, 1989; Owen and Zhou, 2000; Bugallo et al., 2017; Paananen et al., 2020).

That said, for any given target distribution, it is well known that the challenge of finding a base distribution with small KL divergence to the target distribution grows with dimension. Indeed, if this were not the case, then we would rarely need to move beyond basic importance sampling methods, for which the performance is controlled by this KL divergence. Thus, while a better base distribution can speed up CAPS-based tempering, the adaptive path sampling step itself succeeds largely because it is less sensitive to discrepancies between base and target compared to other methods. As we have seen in experiments, our path sampling-based method still yields accurate log normalizing constant estimates and adequate joint samples in cases when importance sampling and bridge sampling fail, even starting from a crudely chosen base distribution.

6.2. Dimension limitations

As our simulation results demonstrate, the performance of CAPS in a variety of tasks scales more favorably with dimension than many counterparts. For the tempering problem in particular, the success of simulated tempering does not require a precise estimate of $z(\lambda)$ (left half of Figure 11; see also Geyer and Thompson, 1995). As long as the posterior marginal density $p(a)$ or $p(\lambda)$ is not too far from uniform, the MCMC sampler will usually be able to draw samples across the full support.

However, CAPS-based tempering can still fail in high dimensional target distributions. In Appendix D, we provide a failure mode example in a latent Dirichlet allocation model, where the log normalizing constant estimate has a scale of $\sim 10^4$, and path sampling estimation manages to estimate it with pointwise errors $\sim 10^2$. For fitting a curve, a 1% error is accurate enough. But a pointwise 1% error in the log normalizing constant amounts to inflating the marginal density $p(a)$ in Algorithm 2 by a factor $\exp(10^2)$ at that point. This causes the marginal temperature distribution to collapse to a near point mass, which prevents the sampler from escaping this narrow region of temperature space. We note however that this problem is not unique to CAPS-based tempering and can equally affect discrete tempering procedures as well.

Explosive log normalizing constants can occur more generally in cases of poor model fit. For instance, when using the prior as the base distribution in CAPS-based tempering, the log normalizing constant $\zeta(\lambda)$ at $\lambda = 1$ reduces to the log marginal likelihood. Thus, $\zeta(\lambda)$ grows linearly in both

the sample size and the log likelihood. When the model exhibits poor predictive fit and small log-likelihood (as in the latent Dirichlet allocation example), the log normalizing constant can span many orders of magnitude, with severe consequences for the accuracy of its estimate. This behavior is analogous to the "folk theorem of statistical computing": When you have computational problems, often there's a problem with your model (Gelman, 2008).

6.3. Geometric path function

For CAPS-based tempering, we consider only the geometric link function $f(a)$. While this is a common choice in the tempering literature, it is unclear if it is optimal in any sense for bridging between disparate modes. In particular, this geometric link can produce abrupt phase transitions whereby the conditional distributions $q(\cdot \mid a)$ vary rapidly over a small range of temperatures a. This translates to difficult geometry in temperature space and potentially prohibitively slow exploration of the joint space by MCMC algorithms (Bhatnagar and Randall, 2004). Since our framework in Section 2 permits arbitrary functions $f(a)$, it is therefore possible that the performance of CAPS-based tempering could be improved with a more careful choice of $f(a)$, but we leave this for future work.

6.4. General recommendations

We have developed a method that integrates adaptive path sampling and tempering and have applied it to several examples. Critically, the procedure is easily automated and integrates with existing MCMC samplers. We have implemented such an integration with Stan's HMC sampler in the R package pathtemp. This package may be used to add CAPS and CAPS-based tempering with existing Stan models, returning both the desired posterior sample and the estimated log normalizing constant at convergence. See Appendix C for software details.

Finally, although we have argued its relative advantage over existing methods, CAPS will not always achieve convergence even with iterative adaptation. This can occur, for instance, when a poorly chosen base distribution or explosive normalizing constant function create difficult joint distributions which are intractable for existing samplers. In such cases, cross validation and multi-chain stacking (Yao et al., 2022) may be effective alternatives for dealing with non-mixing sampler chains, depending on the ultimate goals of the analysis.

Acknowledgements

We thank the U.S. National Science Foundation, Institute of Education Sciences, Office of Naval Research, Sloan Foundation, Schmidt Futures, and the Research Council of Finland Flagship programme: Finnish Center for Artificial Intelligence, and Research Council of Finland project (340721) for partial support. Replication `R` and `Stan` code for experiments is available at `https://github.com/yao-yl/path-tempering`.

References

Betancourt, M. (2015). "Adiabatic Monte Carlo." *arXiv:1405.3489*.

Bhatnagar, N. and Randall, D. (2004). "Torpid mixing of simulated tempering on the Potts model." In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, 478–487.

Bugallo, M. F., Elvira, V., Martino, L., Luengo, D., Miguez, J., and Djuric, P. M. (2017). "Adaptive importance sampling: The past, the present, and the future." *IEEE Signal Processing*, 34: 60–79.

Carlson, D., Stinson, P., Pakman, A., and Paninski, L. (2016). "Partition functions from Rao-Blackwellized tempered sampling." In *Proceedings of the 33rd International Conference on Machine Learning*.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P., and Ridell, A. (2017). "Stan: A probabilistic programming language." *Journal of Statistical Software*, 76(1): 1–32.

Gelman, A. (2008). "The folk theorem of statistical computing." `https://statmodeling.stat.columbia.edu/2008/05/13/the_folk_theore/`.

Gelman, A. and Meng, X.-L. (1998). "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling." *Statistical Science*, 13: 163–185.

Geweke, J. (1989). "Bayesian inference in econometric models using Monte Carlo integration." *Econometrica*, 57: 1317–1339.

Geyer, C. J. (1991). "Markov chain Monte Carlo maximum likelihood." In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, volume 156–163. New York.

Geyer, C. J. and Thompson, E. A. (1995). "Annealing Markov chain Monte Carlo with applications to ancestral inference." *Journal of the American Statistical Association*, 90: 909–920.

Graham, M. M. and Storkey, A. J. (2017). "Continuously tempered Hamiltonian Monte Carlo." In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Gronau, Q. F., Sarafoglou, A., Matzke, D., Ly, A., Boehm, U., Marsman, M., Leslie, D. S., Forster, J. J., Wagenmakers, E.-J., and Steingroever, H. (2017). "A tutorial on bridge sampling." *Journal of Mathematical Psychology*, 81: 80–97.

Jarzynski, C. (1997). "Nonequilibrium equality for free energy differences." *Physical Review Letters*, 78: 2690.

Latuszyński, K., Moores, M. T., and Stumpf-Fétizon, T. (2025). "MCMC for multi-modal distributions." *arXiv:2501.05908*.

Lelievre, T., Rousset, M., and Stoltz, G. (2010). *Free Energy Computation: A Mathematical Perspective*. London: Imperial College Press.

Marinari, E. and Parisi, G. (1992). "Simulated tempering: A new Monte Carlo scheme." *Europhysics Letters*, 19: 451.

Meng, X.-L. and Wong, W. H. (1996). "Simulating ratios of normalizing constants via a simple identity: A theoretical exploration." *Statistica Sinica*, 6: 831–860.

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

Nemeth, C., Lindsten, F., Filippone, M., and Hensman, J. (2019). "Pseudo-extended Markov chain Monte Carlo." In *Advances in Neural Information Processing Systems*, 4312–4322.

Owen, A. and Zhou, Y. (2000). "Safe and effective importance sampling." *Journal of the American Statistical Association*, 95: 135–143.

Paananen, T., Piironen, J., Bürkner, P.-C., and Vehtari, A. (2020). "Implicitly adaptive importance sampling." *arXiv:1906.08850*.

Paulin, D., Jasra, A., and Thiery, A. (2018). "Error bounds for sequential Monte Carlo samplers for multimodal distributions." *arXiv:1509.08775*.

Piironen, J. and Vehtari, A. (2017a). "On the hyperprior choice for the global shrinkage parameter in the horseshoe prior." *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*.

— (2017b). "Sparsity information and regularization in the horseshoe and other shrinkage priors." *Electronic Journal of Statistics*, 11: 5018–5051.

R Core Team (2020). "R: A language and environment for statistical computing." `https://www.R-project.org/`.

Schweizer, N. (2012). "Non-asymptotic error bounds for sequential MCMC methods in multimodal settings." *arXiv:1205.6733*.

Sejdinovic, D., Strathmann, H., Garcia, M. L., Andrieu, C., and Gretton, A. (2014). "Kernel adaptive Metropolis-Hastings." In *International Conference on Machine Learning*, 1665–1673.

Shirts, M. R. and Chodera, J. D. (2008). "Statistically optimal analysis of samples from multiple equilibrium states." *Journal of Chemical Physics*, 129: 124105.

Stan Development Team (2020). *Stan Modeling Language User's Guide*. `http://mc-stan.org`.

Tjelmeland, H. and Hegstad, B. K. (2001). "Mode jumping proposals in MCMC." *Scandinavian Journal of Statistics*, 28(1): 205–223.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., and Bürkner, P.-C. (2021). "Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC." *Bayesian Analysis*, 16: 667–718.

Yao, Y., Vehtari, A., and Gelman, A. (2022). "Stacking for non-mixing Bayesian computations: The curse and blessing of multimodal posteriors." *Journal of Machine Learning Research*, 23: 79.

Appendix A. Derivation of equations

A.1. Fundamental path sampling equation

Let $q : \mathbb{R}^{d+1} \to \mathbb{R}$ be an unnormalized density function. Let the normalizing constant (function) $z : \mathbb{R} \to \mathbb{R}$ be defined by $z(\lambda) = \int_{\mathbb{R}^d} q(\lambda, \boldsymbol{\theta}) d\boldsymbol{\theta}$, where $\boldsymbol{\theta} \in \mathbb{R}^d$. Calculus shows

$$\begin{aligned} \frac{d}{d\lambda} \log z(\lambda) &= \frac{\frac{d}{d\lambda} \int_{\mathbb{R}^d} q(\lambda, \boldsymbol{\theta}) d\boldsymbol{\theta}}{\int_{\mathbb{R}^d} q(\lambda, \boldsymbol{\theta}) d\boldsymbol{\theta}} \\ &= \frac{\int_{\mathbb{R}^d} \frac{\partial}{\partial \lambda} q(\lambda, \boldsymbol{\theta}) d\boldsymbol{\theta}}{z(\lambda)} \\ &= \frac{\int_{\mathbb{R}^d} \frac{\partial}{\partial \lambda} \log q(\lambda, \boldsymbol{\theta}) q(\lambda, \boldsymbol{\theta}) d\boldsymbol{\theta}}{z(\lambda)} \\ &= \int_{\mathbb{R}^d} \frac{\partial}{\partial \lambda} \log q(\lambda, \boldsymbol{\theta}) \frac{q(\lambda, \boldsymbol{\theta})}{z(\lambda)} d\boldsymbol{\theta}, \end{aligned} \tag{23}$$

That is, $\frac{d}{d\lambda} \log z(\lambda) = \mathrm{E}_{\boldsymbol{\theta}|\lambda}(\frac{\partial}{\partial \lambda} \log q(\lambda, \boldsymbol{\theta}))$, which leads to the fundamental equation of path sampling (8) by the chain rule.

At the second equation of (23), we are assuming the legitimacy of changing the order of derivative and integral. One sufficient condition is that there exists a sufficiently large constant M such that $\int_{\mathbb{R}^d} |\frac{\partial}{\partial \lambda} q(\lambda, \boldsymbol{\theta})| d\boldsymbol{\theta} < M$ for all λ, and the interchangeability follows by the dominated convergence theorem.

A.2. The link function in continuous tempering

In continuous tempering, we choose the following piecewise polynomial link function (see Figure 1 for a visualization):

$$f(a) = \begin{cases} 0, & \text{if } \ 0 \le a < a_{\min} \\ -2\left(\frac{a - a_{\min}}{a_{\max} - a_{\min}}\right)^3 + 3\left(\frac{a - a_{\min}}{a_{\max} - a_{\min}}\right)^2, & \text{if } \ a_{\min} \le a < a_{\max} \\ 1, & \text{if } \ a_{\max} \le a < 2 - a_{\max} \\ -2\left(\frac{2 - a_{\min} - a}{a_{\max} - a_{\min}}\right)^3 + 3\left(\frac{2 - a_{\min} - a}{a_{\max} - a_{\min}}\right)^2, & \text{if } \ 2 - a_{\max} \le a < 2 - a_{\min} \\ 0, & \text{if } \ 2 - a_{\min} \le a \le 2. \end{cases} \tag{24}$$

It has a continuous first order derivative. In experiments and default software implementation, we set $a_{\min} = 0.1$ and $a_{\max} = 0.8$.

A.3. Rao-Blackwellization as optimal multistate bridge sampling

In this subsection we follow the notation of bridge sampling discussed in Lelievre et al. (2010). Let $q_i = q(\boldsymbol{\theta}|\lambda_i)$ be the unnormalized density at a sequence of discrete temperatures λ_i, $i = 1, \ldots I$, and $Z = (z_2, \ldots, z_I)$ the (ratio of) normalizing constants (assuming $z_1 = 1$). With an arbitrary list of

$\boldsymbol{\theta} \to \mathbb{R}$ functions $\alpha_{i,j}$, we define A an $(I-1)\times(I-1)$ matrix, and B an $(I-1)$ vector:

$$A = \begin{bmatrix} a_2 & -b_{23} & \dots & -b_{2I} \\ -b_{32} & a_3 & \dots & -b_{3I} \\ \dots & \dots & \dots & \dots \\ -b_{I2} & -b_{I3} & \dots & a_I \end{bmatrix}, \quad B = \begin{bmatrix} b_{21} \\ b_{31} \\ \vdots \\ b_{I1} \end{bmatrix},$$

with each entry a, b shorthand for

$$b_{ij} = \mathrm{E}_{\pi_j}(\alpha_{ij} q_i), \quad a_i = \sum_{j=1, j\neq i}^{I} \mathrm{E}_{\pi_i}(\alpha_{ij} q_j).$$

In discrete tempering, each time we sample from $q(\boldsymbol{\theta}, \lambda) = \frac{1}{c(\lambda)} q(\boldsymbol{\theta}, \lambda)$. Since λ is discrete here, we denote $c_m = c(\lambda_m)$ and $q_m = q(\boldsymbol{\theta}, \lambda = \lambda_m)$, both of which are given in each adaptation. Carlson et al. (2016) considered the Rao-Blackwellized estimate of the normalizing constant:

$$\hat{z}_k^{RB} = \sum_{l=1}^{n} \frac{q_k(\theta_l)}{\sum_{j=1}^{I} c_j q_j(\theta_l)}, \quad k = 1, \dots, I.$$

Proposition 1. *The Rao-Blackwellized estimate can be derived from multistate bridge sampling by choosing*

$$\alpha_{ij}(\boldsymbol{\theta}) = \frac{n_j \hat{z}_j^{-1}}{\sum_m c_m^{-1} q_m(\boldsymbol{\theta})},$$

which is further an empirical estimate of the optimal bridge sampling functions.

Proof. We rearranged the multistate bridge sampling estimates $z_i b_{ji} = z_j b_{ij},\ \forall i, j$ into a matrix form,

$$AZ = B.$$

Shirts and Chodera (2008) show that the optimal sequence of functions that minimizes the variance of the estimated Z is

$$\alpha_{ij}(\boldsymbol{\theta}) = \frac{n_j z_j^{-1}}{\sum_m n_m z_m^{-1} q_m(\boldsymbol{\theta})},$$

which in practice, starting from some initial guess $\hat{z}$, this can be approximated by $\alpha_{ij}(\boldsymbol{\theta}) = \frac{n_j \hat{z}_j^{-1}}{\sum_m n_m \hat{z}_m^{-1} q_m(\boldsymbol{\theta})}$.

Denote $S(\boldsymbol{\theta}) = \sum_m c_m q_m(\boldsymbol{\theta})$, then $\alpha_{ij}(\boldsymbol{\theta}) = \frac{n_j \hat{z}_j^{-1}}{S(\boldsymbol{\theta})}$. We estimate the matrices A and B by their empirical means.

$$\begin{aligned} \hat{a}_i \hat{z}_i &= n_i^{-1} \sum_{k=1}^{n_i} \hat{z}_i \sum_{j=1, j\neq i}^{I} \alpha_{ij}(\theta_{i,k}) q_j(\theta_{i,k}) \\ &= \hat{z}_i n_i^{-1} \sum_{k=1}^{n_i} \frac{\sum_{j=1, j\neq i}^{I} n_j \hat{z}_j^{-1} q_j(\boldsymbol{\theta})}{S(\theta_{i,k})} \\ &= \hat{z}_i - \sum_{k=1}^{n_i} \frac{q_i(\theta_{i,k})}{S(\theta_{i,k})}. \end{aligned}$$

$$\sum_{j=1,j\neq i}^{I} \hat{b}_{ij}\hat{z}_j = \sum_{j=1,j\neq i}^{I} \hat{z}_j n_i^{-j} \sum_{k=1}^{n_j} \frac{n_j \hat{z}_j^{-1} q_i(\theta_{j,k})}{S(\theta_{j,k})} = \sum_{j=1,j\neq i}^{I} \sum_{k=1}^{n_i} \frac{q_i(\theta_{j,k})}{S(\theta_{j,k})}.$$

Combining these two parts yields the final estimate,

$$\begin{aligned} z_i &= \sum_{j=1,j\neq i}^{I} \sum_{k=1}^{n_i} \frac{q_i(\theta_{j,k})}{S(\theta_{j,k})} + \sum_{k=1}^{n_i} \frac{q_i(\theta_{i,k})}{S(\theta_{i,k})} \\ &= \sum_{m=1}^{N} \frac{q_i(\theta_m)}{\sum_{j=1}^{I} c_j q_j(\theta_m)}, \end{aligned}$$

which is identical to the Rao-Blackwellized estimates. □

A.4. Path sampling as the limit of bridge sampling and annealed importance sampling

Proposition 2. *Path sampling can be viewed as the continuous limit of bridge sampling* (5) *and annealed importance sampling* (7) *when the intermediate states* $(0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{L+1} = 1)$ *is infinitely dense, such that* $\max_l \delta_l \to 0$*, where* $\delta_l = \lambda_{l+1} - \lambda_l$ *is the neighboring spacing.*

Proof. The proof is similar to the reasoning in Gelman and Meng (1998). Bridge sampling and annealed importance sampling work in the scale of the normalizing constant z, essentially computing $z(1)/z(0)$ by $\prod_{l=0}^{L}(z(l+1)/z(l))$ or, equivalently, $\log z(1)/z(0) = \sum_{l=0}^{L}(\log z(l) - \log z(l-1))$. Further, both bridge sampling and annealed importance sampling are based on importance sampling identity (with potential refinement of more intermediate states in bridge sampling):

$$\frac{z(l)}{z(l-1)} = \int_{\boldsymbol{\theta}} \frac{q(\boldsymbol{\theta}|\lambda_{l+1})}{q(\boldsymbol{\theta}|\lambda_l)} p(\boldsymbol{\theta}|\lambda_l) d\boldsymbol{\theta}.$$

In general, $\log \mathrm{E}\left(q(\boldsymbol{\theta}|\lambda_{l+1})/q(\boldsymbol{\theta}|\lambda_l)\right) \neq \mathrm{E}\left(\log q(\boldsymbol{\theta}|\lambda_{l+1}) - \log q(\boldsymbol{\theta}|\lambda_l)\right)$, where the expectation is taken over $\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\lambda_l)$. However, such difference will be be approaching zero when we have fine ladder.

For a fixed λ_l, let

$$G_l(\xi) = \log \int_{\boldsymbol{\theta}} \frac{q(\boldsymbol{\theta}|\lambda_l + \xi)}{q(\boldsymbol{\theta}|\lambda_l)} p(\boldsymbol{\theta}|\lambda_l) d\boldsymbol{\theta}.$$

It satisfies $G_l(0) = 0$, and its derivative is

$$\begin{aligned} G_l'(\xi) &= \frac{d}{d\xi}\left(\log \int_{\boldsymbol{\theta}} \frac{q(\boldsymbol{\theta}|\lambda_l + \xi)}{q(\boldsymbol{\theta}|\lambda_l)} p(\boldsymbol{\theta}|\lambda_l) d\boldsymbol{\theta}\right) \\ &= \frac{z(l)}{z(l+\xi)}\left(\int_{\boldsymbol{\theta}} \frac{\partial}{\partial \xi} \frac{q(\boldsymbol{\theta}|\lambda_l + \xi)}{q(\boldsymbol{\theta}|\lambda_l)} p(\boldsymbol{\theta}|\lambda_l) d\boldsymbol{\theta}\right). \end{aligned}$$

At $\xi = 0$, $G_l'(0)$ becomes identical to the path sampling gradient in (8), as

$$G_l'(0) = \int_{\boldsymbol{\theta}} \frac{\partial}{\partial \lambda} \log q(\boldsymbol{\theta}|\lambda_l) p(\boldsymbol{\theta}|\lambda_l) d\boldsymbol{\theta}.$$

By Taylor series expansion, $G_l(\xi) = G_l(0) + \xi \int_{\boldsymbol{\theta}} \frac{\partial}{\partial \lambda} \log q(\boldsymbol{\theta}|\lambda_l) p(\boldsymbol{\theta}|\lambda_l) d\boldsymbol{\theta} + o(\xi)$, Hence, in the limit

as $\max_l \delta_l \to 0$, the importance sampling based estimate can be rearranged into

$$\begin{aligned}\log z(1)/z(0) &= \sum_{l=0}^{L} G_l(\delta_l) \\ &= \sum_{l=0}^{L} \left(\delta_l \int_{\boldsymbol{\theta}} \frac{\partial}{\partial \lambda} \log q(\boldsymbol{\theta}|\lambda_l) p(\boldsymbol{\theta}|\lambda_l) d\boldsymbol{\theta} + o(\delta_l) \right) \\ &= \int_0^1 \int_{\boldsymbol{\theta}} \frac{\partial}{\partial \lambda} \log q(\boldsymbol{\theta}|\lambda_l) p(\boldsymbol{\theta}|\lambda_l) d\boldsymbol{\theta} d\lambda + o(1),\end{aligned}$$

where the dominant term equals the path sampling estimate, and the remainder approaches 0 in the dense limit $\delta_l \to 0, \forall l$ since $\sum_l \delta_l = 1$. □

A.5. On the choice of prior $p(a)$

In path sampling, the final marginal distribution of a relies on user specification. By default we use $z(\cdot) \to c(\cdot)$, which enforce a uniform marginal distribution. More generally, by updating $c(\cdot) \leftarrow z(\cdot)/p^{\text{prior}}(\cdot)$, the final marginal distribution of a will approach p^{prior}. In this section we discuss the choice of p^{prior} beyond uniformity.

The choice of p^{prior} is subject to a efficiency-robustness tradeoff. There are three separate goals to pursue via prior tuning:

1. **Robustness.** Because a uniform a ensures the Markov chain has explored the whole temperature space, it is a conservative choice and we use for default in adaptations.

$$p(a) \propto 1.$$

2. **Minimal variance of log z.** On then other end of the spectrum, we can ask for efficiency. Gelman and Meng (1998) proved that the generalized Jeffreys prior minimizes the variance of estimated log normalizing constant.

$$p^{\text{opt}}(\lambda) \propto \sqrt{\mathrm{E}_{\boldsymbol{\theta}|\lambda} U^2(\boldsymbol{\theta}, \lambda)}.$$

   where $U(\boldsymbol{\theta}, \lambda) = \frac{\partial}{\partial \lambda} \log q(\boldsymbol{\theta}, \lambda)$.

3. **Smooth transition in the joint sampling.** From the perspective of successful sampling, we could ask for

$$\mathrm{KL}\Big(\pi_a, \pi_{a+\delta a}\Big) \approx \text{constant},$$

   which is related to the constant acceptance rate in discrete Gibbs update (when the discrete temperature update is restricted to either a one-step jump $\lambda_k \to \lambda_{k\pm 1}$ or remain unchanged, This constant acceptance rate is often used as a tuning target in discrete tempering (Geyer and Thompson, 1995). The next proposition gives an closed form optimal prior that ensures this constant KL gap.

Proposition 3. *The desired prior to achieve a smooth KL gap is*

$$p^{\text{opt}}(a) \propto \frac{1}{f'(a)} \sqrt{\mathrm{Var}_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|a)} U(\boldsymbol{\theta}, a)}, \ \forall a_{\min} < a < a_{\max}.$$

This is similar to the minimal-variance prior above, with a slight twist from the second moment to variance.

Proof. It is easy to verify that

$$\mathrm{KL}\Big(\pi_a, \pi_{a+\delta a}\Big) = \int \log\left(\frac{\pi_a}{\pi_{a+\delta a}}\right) d\pi_a = \frac{1}{2}(\delta a)^2\Big(\frac{d^2}{da^2}\log z(a) - \frac{f''(a)}{f'(a)}\frac{d}{da}\log z(a)\Big) + o(\delta a)^2.$$

Assuming we have already sampled from the joint stationary distribution, the gap between two neighboring order statistics reflects how dense the local density is, i.e., $\delta a \propto 1/p(a)$.

The two derivative terms can be expressed by expectations,

$$\frac{d}{da}\log(z(a)) = f'(a)\mathrm{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|a)}\left(\log(\psi) - \log(\phi)\right).$$

$$\begin{aligned}\frac{d^2}{da^2}\log z(a) =& f''(a)\mathrm{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|a)}\left(\log\psi - \log q\right) + f'(a)^2\mathrm{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|a)}\left(\left(\log\psi - \log q\right)^2\right)\\ &- \Big(f'(a)\mathrm{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|a)}\left(\log\psi - \log q\right)\Big)^2,\end{aligned}$$

which further simplifies to

$$\begin{aligned}&\frac{d^2}{da^2}\log(z(a)) - \frac{f''(a)}{f'(a)}\frac{d}{da}\log(z(a))\\ =& f'(a)^2\mathrm{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|a)}\left((\log\psi - \log q)^2\right) - \Big(f'(a)\mathrm{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|a)}\left(\log(\psi) - \log(q)\right)\Big)^2.\end{aligned}$$

Put all together, the constant KL gap will be achieved by

$$\begin{aligned}p(a) \propto& \left(\frac{d^2}{da^2}\log z(a) - \frac{f''(a)}{f'(a)}\frac{d}{da}\log z(a)\right)^{\frac{1}{2}}\\ =& \frac{1}{f'(a)}\left(\mathrm{Var}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|a)}\big(\log(\psi) - \log(q)\big)\right)^{\frac{1}{2}}.\end{aligned}$$

It is also evident that under this prior, both $\mathrm{KL}\Big(\pi_a, \pi_{a+\delta a}\Big)$ and the reserve jump $\mathrm{KL}\Big(\pi_{a+\delta a}, \pi_a\Big)$ approximate (different) constants along the trajectory. □

Due to dependence on the unknown normalizing constant (and higher orders), these two efficiency-optimal priors require additional tuning and adaptations. In general, we still prefer to use the simple uniform margin for robustness. Nevertheless, our method enables any user specific-prior choice.

A.6. Choice of regression kernels

In regularized path sampling (Section 2), we regularize $\log z$ and $\log c$ via a parametric regression form:

$$\min_{\beta}\sum_{i=1}^{I}\left(\log z(f(a_i^*)) - \left(\beta_0 f(a_i^*) + \sum_{j=1}^{J}\beta_j\gamma_j(f(a_i^*))\right)\right)^2. \tag{25}$$

In other words, we approximate $\log z(f(a))$ by $\beta_0 f(a) + \sum_{j=1}^{J}\beta_j\gamma_j(f(a))$. This also enables a closed form expression for the gradient c' and z' that will be used in Algorithm 1.

The term $\log z(f(a))$ will be a constant function wherever $f(a)$ is constant. In our experiment, we have tried a sequence of Gaussian kernels,

$$\gamma_j(\lambda) = \exp\left(\frac{-(\lambda - \lambda_j^{\text{kernel}})^2}{2\sigma_{\text{kernel}}^2}\right),\ \lambda_j^{\text{kernel}} = \frac{j}{J+1},\ \sigma_{\text{kernel}} = 1/J,\ 1 \le j \le J,$$

logit kernels,

$$\gamma_j(\lambda) = \text{logit}^{-1}\left(\frac{\lambda - \lambda_j^{\text{kernel}}}{\sigma_{\text{kernel}}}\right),\ \lambda_j^{\text{kernel}} = \frac{j}{J+1},\ \sigma_{\text{kernel}} = 1/J,\ 1 \le j \le J,$$

and cubic splines.

We have not found the kernel choice to have a large impact on the final tempering procedure or log normalizing constant. For the experiments in Section 5, we used a combination of the Gaussian and logit kernels at $J = 10$ points within the path sampling adaptation steps for speed and simplicity. After running our final adaptation, we then smoothed the final estimate of the normalizing constant or marginal posterior using cubic splines since these introduce less pseudo-periodic behavior than the Gaussian kernels.

To be clear, although $z(\cdot)$ can be further used in density estimation, the kernels are used here for regularization and functional approximation, and is *not* relevant to *kernel* density estimation. The latter is noisy and contingent on various smoothness assumptions. In contrast, (25) is a *linear regression* problem as $\log z(f(a_i^*))$ is known from the path sampling estimate. Besides, the choice of inner kernel points should not be confused with the tempering ladder. Here we are sampling a and λ continuously and evaluate $z(\lambda)$ for all $\lambda \in [0, 1]$.

Appendix B. Proof of Theorem 1

Proof. The parametric estimate $\hat{\zeta}(\lambda)$ is obtained by approximating the $\widetilde{\zeta}(\lambda_{(s)})$ with a linear combination of kernel functions from a given family (e.g., Gaussians). For purposes of our theoretical analysis, we will fix a finite set of "anchor" points $\lambda_a \in \Lambda$ for $a \in \{1, \ldots, A\}$ which are independent of sample size n and iteration t, and which we will use to construct the estimate $\hat{\zeta}$.

The function ζ is Lipschitz continuous since we assume that $\zeta'(\lambda) = \mathbb{E}\left(\frac{\partial}{\partial\lambda}\log q\left(\boldsymbol{\theta};\lambda\right) \mid \lambda\right)$ is continuous and thus bounded over Λ. Furthermore, $\hat{\zeta}$ is also Lipschitz since we assume that basis functions used in the smoothing procedure are Lipschitz and the resulting regression coefficients are bounded. Let L be such that ζ and $\hat{\zeta}$ are both L-Lipschitz.

We can then decompose the error of our estimate in the following way. For any $\lambda \in \Lambda$, let $a^* = \text{argmin}_{1\le a\le A}|\lambda - \lambda_a|$. Then

$$\begin{aligned}
\left|\zeta(\lambda) - \hat{\zeta}(\lambda)\right| &\le |\zeta(\lambda) - \zeta(\lambda_{a^*})| + \left|\hat{\zeta}(\lambda) - \hat{\zeta}(\lambda_{a^*})\right| + \left|\hat{\zeta}(\lambda_{a^*}) - \zeta(\lambda_{a^*})\right| \\
&\le 2\epsilon_a + \left|\hat{\zeta}(\lambda_{a^*}) - \zeta(\lambda_{a^*})\right| \qquad (26) \\
&\le 2\epsilon_a + \epsilon_p + \left|\widetilde{\zeta}(\lambda_{a^*}) - \zeta(\lambda_{a^*})\right|. \qquad (27)
\end{aligned}$$

In the above, (26) follows from ζ and $\hat{\zeta}$ both being Lipschitz with constant L, allowing us to take $\epsilon_a = L\max_{1\le a\le A-1}|\lambda_{a+1} - \lambda_a|$. This in turn can be made arbitrary small by decreasing the mesh size of the $\{\lambda_a\}$. Likewise, (27) is obtained by choosing our parametric family of kernel functions dense enough so that the maximum pointwise approximation error between $\hat{\zeta}(\lambda_a)$ and $\widetilde{\zeta}(\lambda_a)$ is at most ϵ_p. By Stone-Weierstrass, this ϵ_p can also be shrunk by increasing the density of the approximating family.

Since this holds pointwise for any $\lambda \in \Lambda$, we can take suprema to obtain

$$\sup_{\lambda\in\Lambda}\left|\zeta(\lambda) - \hat{\zeta}(\lambda)\right| \le \sup_{1\le a\le A}\left|\widetilde{\zeta}(\lambda_a) - \zeta(\lambda_a)\right| + 2\epsilon_a + \epsilon_p. \qquad (28)$$

With this, we can control the probability of sufficiently large errors as follows. For any $\epsilon > 2\epsilon_a + \epsilon_p$, if we define $\epsilon' = \epsilon - 2\epsilon_a + \epsilon_p$, then we have

$$\Pr\left(\sup_{\lambda\in\Lambda}\left|\zeta(\lambda) - \hat{\zeta}(\lambda)\right| \geq \epsilon\right) \leq \Pr\left(\sup_{1\leq a\leq A}\left|\widetilde{\zeta}(\lambda_a) - \zeta(\lambda_a)\right| \geq \epsilon'\right) \tag{29}$$

$$\leq \sum_{a=1}^{A}\Pr\left(\left|\widetilde{\zeta}(\lambda_a) - \zeta(\lambda_a)\right| \geq \epsilon'\right), \tag{30}$$

where (29) follows directly from (28), and (30) is just Boole's inequality. Now if we let $\boldsymbol{\lambda} = (\lambda_{(1)}, \ldots, \lambda_{(S)})$, then for any $1 \leq a \leq A$,

$$\left|\widetilde{\zeta}(\lambda_a) - \zeta(\lambda_a)\right| \leq \left|\widetilde{\zeta}(\lambda_a) - \mathbb{E}\left(\widetilde{\zeta}(\lambda_a) \mid \boldsymbol{\lambda}\right)\right| + \left|\mathbb{E}\left(\widetilde{\zeta}(\lambda_a) \mid \boldsymbol{\lambda}\right) - \zeta(\lambda_a)\right|. \tag{31}$$

Thus it suffices to bound $\Pr\left(\left|\widetilde{\zeta}(\lambda_a) - \mathbb{E}\left(\widetilde{\zeta}(\lambda_a) \mid \boldsymbol{\lambda}\right)\right| \geq \frac{\epsilon'}{2}\right)$ and $\Pr\left(\left|\mathbb{E}\left(\widetilde{\zeta}(\lambda_a) \mid \boldsymbol{\lambda}\right) - \zeta(\lambda_a)\right| \geq \frac{\epsilon'}{2}\right)$ To bound the first of these, we proceed as follows:

$$\Pr\left(\left|\widetilde{\zeta}(\lambda_a) - \mathbb{E}\left(\widetilde{\zeta}(\lambda_a) \mid \boldsymbol{\lambda}\right)\right| \geq \frac{\epsilon'}{2}\right) \tag{32}$$

$$= \mathbb{E}_{\boldsymbol{\lambda}}\Pr\left(\left|\widetilde{\zeta}(\lambda_a) - \mathbb{E}\left[\widetilde{\zeta}(\lambda_a) \mid \boldsymbol{\lambda}\right]\right| \geq \frac{\epsilon'}{2}\middle|\boldsymbol{\lambda}\right) \tag{33}$$

$$\leq 2\,\mathbb{E}_{\boldsymbol{\lambda}}\left(\left(\frac{2}{\epsilon'}\right)^2\sum_{s=1}^{S}(\lambda_{(s)} - \lambda_{(s-1)})^2\mathrm{Var}\left(\frac{\partial}{\partial\lambda}\log q(\theta_{(s)}; \lambda_{(s)}) \mid \lambda_{(s)}\right)\mathbb{1}_{\{\lambda_{(s)}\leq\lambda_a\}}\right) \tag{34}$$

$$\leq \frac{8S}{[\epsilon']^2}\sup_{\lambda\in\Lambda}\left(\mathrm{Var}\left(\frac{\partial}{\partial\lambda}\log q(\theta_{(s)}; \lambda_{(s)}) \mid \lambda\right)\right)\mathbb{E}_{\lambda}\left(\max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2\right)$$

$$\leq \frac{8SV}{[\epsilon']^2}\mathbb{E}_{\boldsymbol{\lambda}}\left(\max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2\right), \tag{35}$$

where (32) follows from the law of total expectation, (34) follows from Chebyshev's inequality noting that since the $\theta_{(s)}$ are mutually independent given $\boldsymbol{\lambda}$, the terms in the sum are mutually independent, and (35) follows from the assumption that the conditional variances are bounded over Λ.

To bound the second term, let $e(\lambda) = \mathbb{E}\left(\frac{\partial}{\partial\lambda}\log q(\boldsymbol{\theta}; \lambda) \mid \lambda\right)$. Then $e(\lambda)$ is continuously differentiable by assumption, and by compactness of Λ, $e(\lambda)$ is L-Lipschitz for some L and bounded in absolute value by some M. For a fixed, define $S_a = \max(s \mid \lambda_{(s)} \leq \lambda_a)$, and let $\lambda_{(0)} = \lambda_0$. Using this, we have

$$\left|\zeta(\lambda_a) - \mathbb{E}\left(\widetilde{\zeta}(\lambda_a) \mid \boldsymbol{\lambda}\right)\right| = \left|\int_{\lambda_0}^{\lambda_a} e(\lambda)d\lambda - \sum_{s=1}^{S_a}(\lambda_{(s)} - \lambda_{(s-1)})e(\lambda_{(s)})\right|$$

$$\leq \sum_{s=1}^{S_a}\int_{\lambda_{(s-1)}}^{\lambda_{(s)}}\left|e(\lambda) - e(\lambda_{(s)})\right|d\lambda + \int_{\lambda_{(S_a)}}^{\lambda_a}|e(\lambda)|d\lambda \tag{36}$$

$$\leq L\sum_{s=1}^{S_a}\int_{\lambda_{(s-1)}}^{\lambda_{(s)}}(\lambda_{(s)} - \lambda)d\lambda + M\left[\lambda_a - \lambda_{(S_a)}\right] \tag{37}$$

$$\leq \max(L, M)\sum_{s=1}^{S_a+1}(\lambda_{(s)} - \lambda_{(s-1)})^2$$

$$\leq \max(L, M)S\max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2,$$

where (36) follows from $\int_{\lambda_{(s-1)}}^{\lambda_{(s)}} e(\lambda_{(s)})d\lambda = (\lambda_{(s)} - \lambda_{(s-1)})e(\lambda_{(s)})$, and (37) follows from the Lipschitz property of $e(\lambda)$. Letting $L' = \max(L, M)$, this in turn directly implies,

$$\Pr\left(\left|\mathbb{E}\left(\widetilde{\zeta}(\lambda_a) \mid \boldsymbol{\lambda}\right) - \zeta(\lambda_a)\right| \geq \frac{\epsilon'}{2}\right) \leq \Pr\left(S \max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2 \geq \frac{\epsilon'}{2L'}\right). \tag{38}$$

In order to bound (35) and (38), we first take a partition $\{I_j\}_{j=1}^J$ of Λ where each I_j is an interval of length at least $\ell/2$ and no more than ℓ, for $\ell = \frac{1}{2}\sqrt{\epsilon'/2L'S^{1+\delta}}$ and some $\delta \in [0, 1)$. Furthermore, let $\lambda_{\min} = \arg\min_{\lambda\in\Lambda} p(\lambda)$ where $p(\lambda)$ is the marginal distribution of λ, and let D be the length of the compact interval Λ. With this, it is easy to see that

$$\begin{aligned}
\Pr\left(\text{some } I_j \text{ empty}\right) &\leq \sum_{j=1}^{J} \Pr\left(I_j \text{ empty}\right) && (39)\\
&\leq \sum_{j=1}^{J}\prod_{s=1}^{S} \Pr\left(\lambda_{(s)} \notin I_j\right) && (40)\\
&= \sum_{j=1}^{J}\left(1 - \int_{I_j} p(\lambda)d\lambda\right)^S \\
&\leq J\left[1 - \frac{p(\lambda_{\min})}{4}\sqrt{\frac{\epsilon'}{2L'S^{1+\delta}}}\right]^S \\
&\leq 4D\sqrt{\frac{2L'S^{1+\delta}}{\epsilon'}}\left(1 - \frac{p(\lambda_{\min})}{4}\sqrt{\frac{\epsilon'}{2L'S^{1+\delta}}}\right)^S
\end{aligned}$$

where (39) follows from Boole's inequality and (40) follows from the $\lambda_{(s)}$ being sampled independently from $p(\lambda)$.

Now we can use this to bound (38) by observing that

$$\begin{aligned}
\Pr\left(S \max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2 \geq \frac{\epsilon'}{2L'}\right) &\leq \Pr\left(\max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2 \geq \frac{\epsilon'}{2L'S^{1+\delta}}\right) \\
&= \Pr\left(\max_{1\leq s\leq S}\left|\lambda_{(s)} - \lambda_{(s-1)}\right| \geq 2\ell\right) \\
&\leq \Pr(\text{some } I_j \text{ empty}) && (41)
\end{aligned}$$

where (41) follows from the fact that if each I_j contains at least one sample $\lambda_{(s)}$, then the maximum distance between (ordered) samples cannot be more than twice the maximum length of the I_j.

Now to bound (35), let $\mathcal{E} = \{S\max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2 \geq \frac{\epsilon'}{2L'S^\delta}\}$. Then

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\lambda}}\left(S \max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2\right) &= \mathbb{E}_{\boldsymbol{\lambda}}\left(S \max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2 \middle| \mathcal{E}\right)\Pr(\mathcal{E}) \\
&\quad + \mathbb{E}_{\boldsymbol{\lambda}}\left(S \max_{1\leq s\leq S}(\lambda_{(s)} - \lambda_{(s-1)})^2 \middle| \mathcal{E}^c\right)\Pr(\mathcal{E}^c) \\
&\leq D^2 S\Pr(\mathcal{E}) + \frac{\epsilon'}{2L'S^\delta} \\
&\leq 4D^3\sqrt{\frac{L'}{\epsilon'}}S^{(3+\delta)/2}\left(1 - \frac{p(\lambda_{\min})}{2}\sqrt{\frac{\epsilon'}{2L'S^{1+\delta}}}\right)^S + \frac{\epsilon'}{2L'S^\delta} && (42)
\end{aligned}$$

where (42) follows directly from (41) given the definition of $\mathcal{E}$. Thus, combining all of the above, and letting C and c be constants independent of S and ϵ (but depending possibly on L', D, V, and A), we get that for any $\epsilon > 2\epsilon_a + \epsilon_p$ and any $\delta \in (0, 1)$,

$$\Pr\left(\sup_{\lambda\in\Lambda}\left|\zeta(\lambda) - \hat{\zeta}(\lambda)\right| \geq \epsilon\right) \leq C\frac{1}{\sqrt{\epsilon'}}S^2\left(1 - cp(\lambda_{\min})\sqrt{\frac{\epsilon'}{S^{1+\delta}}}\right)^S + \frac{\epsilon'}{S^\delta} \tag{43}$$

□

Appendix C. Software implementation in `Stan`

To provide an `R` (R Core Team, 2020) interface of path sampling and continuous tempering, we create a package `pathtemp`, with the underlying execution inside the general-purpose Bayesian inference engine `Stan` (Carpenter et al. (2017); Stan Development Team (2020)). The source code is available at `https://github.com/yao-yl/path-tempering`. The procedure is automated and requires minimal tuning.

To install the package, call

```
devtools::install_github(
"yao-yl/path-tempering/package/pathtemp",upgrade="never")
```

We demonstrate the practical implementation of continuous tempering on a Cauchy mixture example. Consider the following Stan model:

```
data {
  real y;
}
parameters {
  real theta;
}
model{
  y ~ cauchy(theta, 0.2);
 -y ~ cauchy(theta, 0.2);
}
```

Yao et al. (2022) have analyzed the posterior behavior of this mixture model. If y is large, the posterior distribution of θ will be asymptotically concentrated at two points close to $\pm y$. As a result, Stan cannot fully sample from this two-point-spike even with a large number of iterations.

To run continuous tempering, a user can specify any base model, say $\boldsymbol{\theta} \sim \text{normal}(0, 5)$, and list it in an `alternative model` block as if it were a regular model.

```
...
model {     // keep the original model
  y ~ cauchy(theta, 0.2);
 -y ~ cauchy(theta, 0.2);
}
alternative model { // add a new block of base measure (e.g., the prior)
 theta ~ normal(0, 5);
}
```

After saving this code to the file `cauchy.stan`, we run the function `code_temperature_augment()`, which automatically constructs a tempered path between the orginal model and the alternative model, and generates a working model named `cauchy_augmented.stan`:

```
library(pathtemp)
update_model <- stan_model("solve_tempering.stan")
file_new <- code_temperature_augment("cauchy.stan")
> output:
> A new stan file has been created: cauchy_augmented.stan.
```

We have automated path sampling and its adaptation into a function `path_sample()`. The following two lines realize adaptive path sampling.

```
sampling_model <- stan_model(file_new) # translated to C++ code
path_sample_fit <- path_sample(sampling_model=sampling_model,
  data=list(gap=10)) # data list in original model
```

The returned value `path_sample_fit` provides access to the posterior draws $\boldsymbol{\theta}$ from the target density and base density, the join path in the $(\boldsymbol{\theta}, a)$ space in the final adaptation, and the estimated log normalizing constant $\log z(\lambda)$.

```
sim_cauchy <- extract(path_sample_fit$fit_main)
in_target <- sim_cauchy$lambda==1
in_prior  <- sim_cauchy$lambda==0
# sample from the target
hist(sim_cauchy$theta[in_target])
# sample from the base
hist(sim_cauchy$theta[in_prior])
# the joint "path"
plot(sim_cauchy$a, sim_cauchy$theta)
# the normalizing constant
plot(g_lambda(path_sample_fit$path_post_a), path_sample_fit$path_post_z)
```

The output is presented in Figure 12.

Second, this automated procedure enables to fit two models together.

The following Stan code fits a regression with both the probit and logit link. A path between them effectively expands the model continuously such both individual model are special cases of the augmented model. The computational efficiency is enhanced as we are fitting one slightly larger model rather than fitting two models. In addition, the log normalizing constant tells us which models fits the data better, which is related to but distinct from the log Bayes factor in model comparisons. The output in one run is presented in Figure 13. The data favor the logit link accordingly.

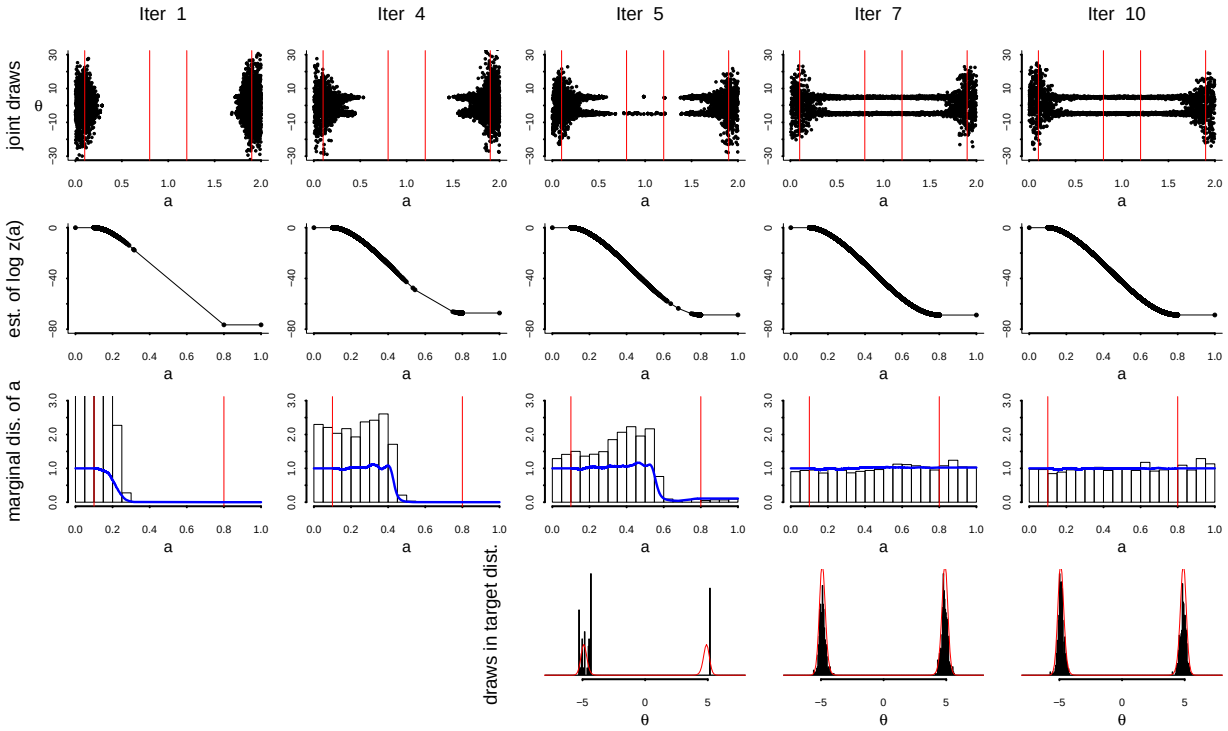

Figure 12: *Output from the Cauchy example: Posterior draws θ from the target density and base density, the join path in the (θ, a) space in the final adaptation iteration, and the estimated log normalizing constant* $\log z(\lambda)$. *The visualization is based on 6 adaptations and S=1500 posterior draws.*

```
data {
int n;
int y[n];
real x[n];
}
parameters {
real beta;
}
model {
    beta ~ normal (0,2);
    y ~ bernoulli_logit(beta * x);  // logistic regression
}
alternative model {
    beta ~ normal (0,1);  // can be a different prior
    y ~  bernoulli(Phi(beta * x)); // probit regression
}
```

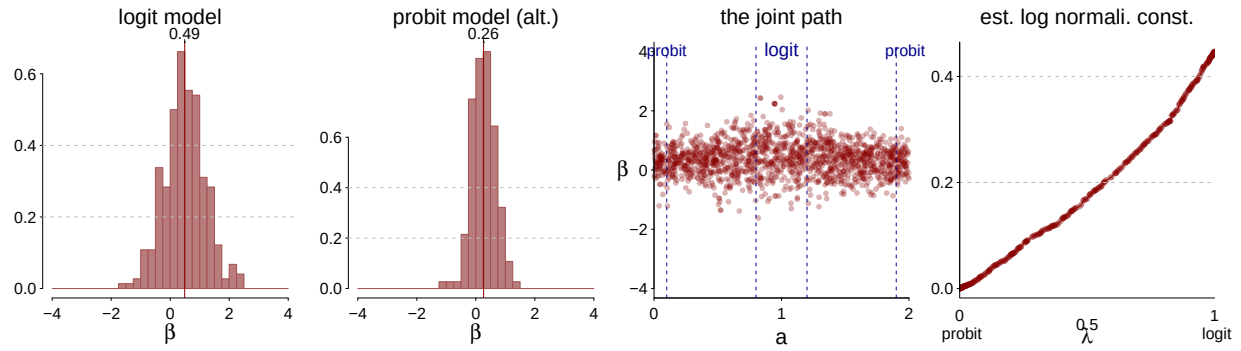

Figure 13: *Output from the logit-probit code example: the posterior draws β from the probit and logit link, the join path in the (β, a) space in the final adaptation, and the estimated log normalizing constant* $\log z(\lambda)$*. The visualization is based on 2 adaptations and S=1500 posterior draws.*

Appendix D. A failure mode: Simulated tempering on latent Dirichlet allocation

Here we present a failure mode of simulated tempering on a high dimensional latent Dirichlet allocation (LDA) model, which is widely used in natural language processing, computer vision, and population genetics. In the model, the j-th document $(1 \le j \le J)$ is drawn from the l-th topic $(1 \le l \le L)$ with probability θ_{jl}, where the topic is defined by a vector of probabilities ϕ_l over the vocabulary, such that each word in the document from topic l is independently drawn from a multinomial distribution with probability ϕ_l. We apply the LDA model to texts in the novel *Pride and Prejudice*. After removing frequent and rare words, the book contains 2025 paragraphs and 32877 words, with a total unique vocabulary size of 1495. We randomly split the words in the data into a training and test set. The dimension of the parameters $\boldsymbol{\theta}$ and ϕ grows as a function of the number of topics L by $2025 \times L$ and $L \times 1495$ respectively. We place independent Dirichlet(0.1) priors on $\boldsymbol{\theta}$ and ϕ. In our experiment, we use $L = 5$ topics.

LDA is prone to a multimodal posterior distribution. The variational based inference often does not replicate itself from multiple runs or data shuffle, which can create the appearance of random results for the user and reduces the predictive power. Yao et al. (2022) has reported posterior multimodality in this dataset and model setting.

We use continuous tempering to sample from the joint distribution proportional to the joint density: $c^{-1}(\lambda)\text{prior}^{1-\lambda}\text{likelihood}^{\lambda}$. For initialization, we start by simulating draws $(\boldsymbol{\theta})_i^{\text{prior}}$ in the prior, where $\boldsymbol{\theta}$ now denotes all parameters in the model, and assign the importance sampling estimate $\log\left(\frac{1}{S}\sum_{i=1}^{S} p(y|(\boldsymbol{\theta})_i^{\text{prior}})\right)$ to the initial slope coefficient b_0, which is close to -7×10^4 in our first run.

After 10 adaptations and 3000 joint HMC iterations per adaptation, the sampler still explores a thin range of the transformed temperature a, both in the last sample, and all 10 samples mixed, shown in the two histograms in Figure 14.

In adaptive path sampling, the log normalizing constant is computed by the integral of the pointwise gradient $u_i = \log p(y|a_i, \theta_i)\frac{d}{da}\lambda\big|_{a=a_i}$. We examined these three terms: log likelihood $\log p(y|a_i, \theta_i)$, pointwise gradient u_i, and the λ derivative $\frac{d}{da}\lambda\big|_{a=a_i}$ along all sampled a_i in Figure 15. The variation in log likelihood $\log p(y|a_i, \theta_i)$ could be a concern as we approximate its pointwise expectation by one Monte Carlo draw. However, the variation in pointwise log likelihood (scale of 10^3) is small compared with its absolute scale ($\sim 10^5$). Hence, the gradient seems to have been computed with small Monte Carlo error (panel 2). Expect if we zoom in, the potinwise variation

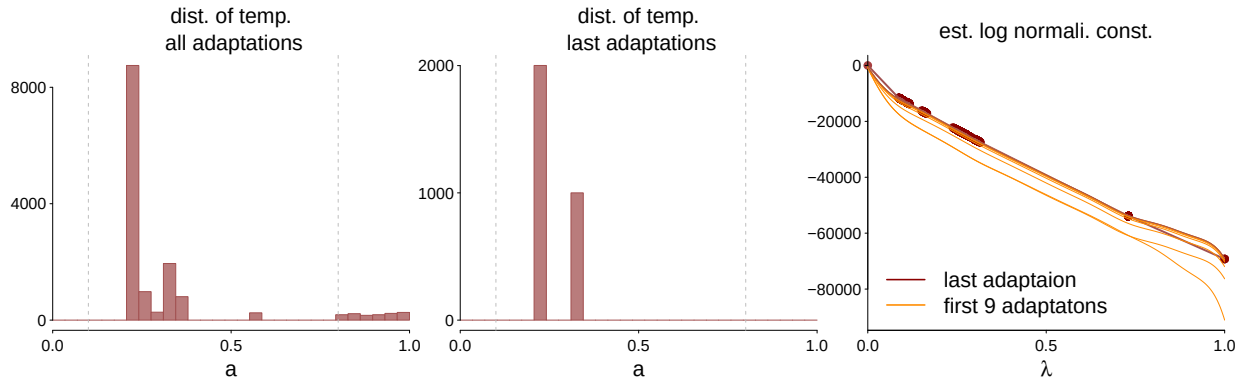

Figure 14: *(a, b): Histograms of transformed temperature among all adaptations and the final adaptation; (c): estimated log normalizing constant as a function of temperature* λ.

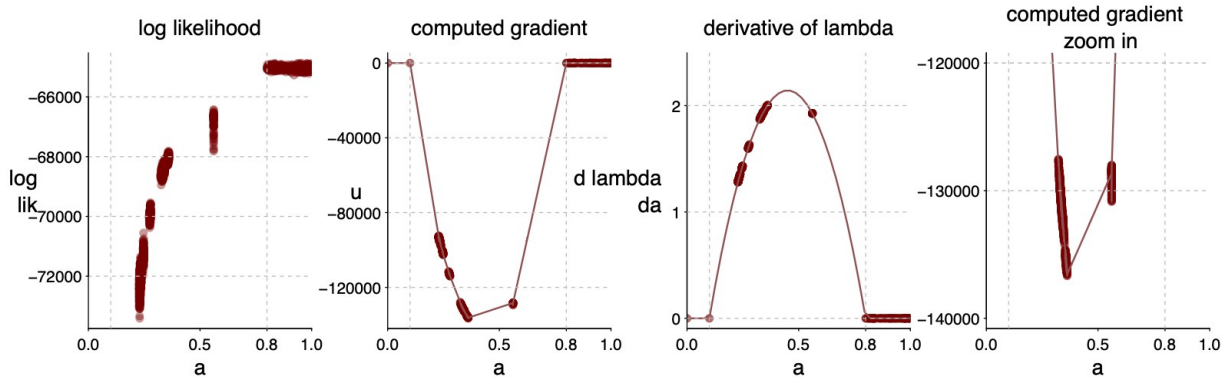

Figure 15: *(1) the pointwise log likelihood* $p(y|\theta, a)$ *among all simulations draws as a function of* a. *(2) the computed pointwise gradient* u. *(3) the derivative* $d\lambda/da$. *(4) zoom-in of panel 2 showing only smallest values of* u.

can still be found around $a \approx 0.6$ (panel 4).

Why does this matter? Recall what we typically require for *curve fitting* in data analysis. We often do not care about the absolute scale of the curve. Indeed we often transform all input to the unit scale in regressions, implicitly assuming that approximating a process normal$(1:10, 1)$ with pointwise errors normal$(0, 0.1)$ is operationally equivalent to approximating the multiplicatively inflated process normal$(100:1000, 100)$ with pointwise errors normal$(0, 10)$.

Measured from its relative error, the path sampling estimate of log normalizing constant has been stable after 10 adaptations (last panel in Figure 14). However, because the log normalizing constant enters the joint density in an additive way, the absolute scale of the approximation error does matter for the purpose of tempering. Having an approximation error $\sim 10^3$ in the scale of $\log z$ is tiny compared to the range of the whole curve, and inevitable if we use any parametric regularization to fit the curve. But this 10^3 error is comparable with the scale of log likelihood. That is, if we start with the exactly known log normalizing constant $\log z(a) = \mathcal{O}(10^4)$, we will obtain the exact uniform margin of a in the posterior. But if at one single temperature point a_1 we add an 1% noise $\log \hat{z}(a_0) += 0.01 \log z(a_0)$, we will create an $\exp(100) = 10^{43}$ bump in the marginal density $p(a)$: essentially a point mass at a_0.

This pitfall does not mean our method is particularly flawed. Discrete tempering methods are

even worse, as (a) in the best case they estimate a coarse discrete ladder, while here every point matters, and (b) they work in the scale of normalizing constant z directly, and it is hopeless to estimate a quantity in the scale of $\exp(-80{,}000)$ with absolute accuracy.

We know of no attempts to run discrete tempering on LDA models. In the usual Metropolis-within-Gibbs scheme, a Gibbs swap requires draws from the conditional distribution given temperature, a step taking several hours in this large model, and discrete tempering often requires several thousand such steps! In contrast, our method is feasible to run on this dataset because it only requires one joint HMC sample per adaptation.

What is the general lesson we can learn from this counterexample? First, estimating the log normalizing constant is not equivalent to successful tempering, where we care more about the absolute approximation error in the second case.

Second, tempering imposes dimensional limitations. The log likelihood scales linearly with the data input, and the magnitude depends on the how good the model fits the data. A generic prior-posterior geometric path will essentially fail when we add more and more data.

Third, in this example LDA is not even designed for predictions and its negative log likelihood (i.e., pointwise training error) is large even for one single input point. In general, a weak prediction model will amplify the log likelihood explosion in the prior-posterior path.

One remedy here is to start with a better constructed base measurement, such that the log normalizing constant will be smaller. In this example, the discrepancy between the prior and posterior is too large, as KL(prior, posterior) is of the order $\exp(70{,}000)$, and even path sampling fails to fill the gap.

For this particular model, if the goal is the log normalizing constant or, equivalently, the log marginal likelihood, we believe the path sampling estimate is still useful, and arguably more reliable than importance sampling and bridge sampling for reasons we have explained in our paper. But we will not use path sampling and simulated tempering for this LDA model when the goal is posterior sampling, for which we instead recommend multi-chain stacking (Yao et al., 2022).