Recursive Monte Carlo and Variational Inference with Auxiliary Variables

Alexander K. Lew[1] **Marco Cusumano-Towner**[1] **Vikash K. Mansinghka**[1]

[1]Massachusetts Institute of Technology, Cambridge, Massachusetts, USA

Abstract

A key design constraint when implementing Monte Carlo and variational inference algorithms is that it must be possible to cheaply and exactly evaluate the marginal densities of proposal distributions and variational families. This takes many interesting proposals off the table, such as those based on involved simulations or stochastic optimization. This paper broadens the design space, by presenting a framework for applying Monte Carlo and variational inference algorithms when proposal densities cannot be exactly evaluated. Our framework, *recursive auxiliary-variable inference* (RAVI), instead approximates the necessary densities using *meta-inference*: an additional layer of Monte Carlo or variational inference, that targets the proposal, rather than the model. RAVI generalizes and unifies several existing methods for inference with expressive approximating families, which we show correspond to specific choices of meta-inference algorithm, and provides new theory for analyzing their bias and variance. We illustrate RAVI's design framework and theorems by using them to analyze and improve upon Salimans et al. [37]'s Markov Chain Variational Inference, and to design a novel sampler for Dirichlet process mixtures, achieving state-of-the-art results on a standard benchmark dataset from astronomy and on a challenging data-cleaning task with Medicare hospital data.

1 INTRODUCTION

Monte Carlo and variational inference algorithms are the workhorses of modern probabilistic inference, a fundamental problem with applications in many disciplines [27]. A key challenge in applying these algorithms is the design of *proposal distributions* (in VI, variational families), which can greatly affect their performance [6]. A good proposal should incorporate any knowledge the practitioner might have about the shape of the posterior; however, this goal is often in tension with the requirement that a proposal's marginal density be analytically tractable, in order to compute importance weights, MCMC acceptance probabilities, or gradient updates for VI. The challenge is that proposal distributions that are simple enough to admit exact density evaluators may not be flexible enough to solve real-world posterior inference problems.

In this paper, we present a new framework, called *Recursive Auxiliary-Variable Inference* (RAVI), for incorporating more complex proposals, without exact marginal density evaluators, into standard Monte Carlo and VI algorithms. The key idea is to approximate the proposal densities using *meta-inference* [10]: an additional layer of Monte Carlo or variational inference targeting the proposal, rather than the model. RAVI generalizes and unifies several existing methods for inference with expressive proposals [37, 35, 40], which we show correspond to specific choices of meta-inference algorithm (see Appendix B for 10 examples).

Contributions. Our key contributions are:

- the RAVI framework, including new recursive algorithms for IS, VI, SMC, and MH using proposals without exact marginal density evaluators (Sections 2 & 3);
- theorems characterizing the impact of RAVI's estimated densities on inference quality (sampler variance, or tightness of variational bounds) (Section 4); and
- two extended examples of RAVI's application to algorithm design and analysis: (1) a novel variant of Salimans et al. [37]'s Markov Chain Variational Inference (MCVI) algorithm that, unlike vanilla MCVI, scales to handle proposals incorporating long MCMC chains; and (2) a novel sampler for Dirichlet process mixtures that uses a randomized agglomerative clustering algorithm as a proposal, outperforming strong baselines on a standard benchmark from astronomy [16] and a challenging Medicare data cleaning problem [25, 21].

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

Monte Carlo or variational inference algorithm	Distributions that no longer need fast exact density evaluators	Example applications
Importance Sampling [19] (Alg. 1, Appendix B.1)	proposal $q(x; y)$	Nested IS [29] (Appendix B.6), Agglomerative Monte Carlo (Section 5, RAVI strategy 2), Annealed IS [31] (Appendix B.5)
Particle Filtering [13] (Appendix B.3)	initial proposal $q_0(x_0; y_0)$, step proposals $q_t(x_i \mid x_{t-1}, y_t)$	Nested SMC [29] (Appendix B.6), SMC^2 [8] (Appendix B.7)
Del-Moral SMC [12] (Appendix B.3)	initial proposal $q_0(x_0)$, forward kernels $K_t(x_t \mid x_{t-1})$, reverse kernels $L_t(x_{t-1} \mid x_t)$, targets $\tilde{\pi}_t(x)$	
Black-Box Variational Inference [34] (Alg. 3)	variational family $q_\theta(x; y)$	IWAE [5] (Appendix B.2), MCVI [37] (Section 2, Appendix B.9), Variational SMC [30] (Appendix B.4)
Amortized Variational Inference [20] (Alg. 4)	variational family $q_\theta(x; y)$	Amortized Rejection Sampling [28] (Appendix B.8)
Metropolis-Hastings (Alg. 5)	transition proposal $q(x'; x)$	pseudo-marginal ratio MH [2]
Hierarchical Variational Inference [35]	variational family $q_\theta(z, x; y)$, reverse proposal $r_\theta(z; x, y)$	Importance-Weighted HVI [40], RAVI-MCVI (Sections 2 and 5, RAVI strategy 1)

Table 1: RAVI generalizes many algorithms for Monte Carlo and variational inference, by allowing practitioners to choose proposals, variational families, and intermediate targets for which exact density evaluators are not available. In the "example applications" column, we list both novel examples of algorithms that exploit this degree of freedom (e.g., the Agglomerative Monte Carlo algorithm we develop in Section 5), and algorithms from the literature that — as we show in Appendix B — can be viewed as instances of simpler algorithms, but with certain sophisticated proposals whose density RAVI estimates.

2 RECURSIVE AUXILIARY-VARIABLE INFERENCE

In this section, we introduce the RAVI framework in the context of a running example: we incorporate a chain of MCMC steps into a proposal, so that it can more accurately approximate a posterior distribution. Our approach generalizes Salimans et al. [37]'s Markov Chain Variational Inference (MCVI) algorithm, and fixes a flaw that prevents it from scaling to longer MCMC chains.

An expressive proposal based on MCMC. Let $p(x, y)$ be a latent-variable model and y an observation. Suppose we wish to approximate $p(x \mid y)$ using an expressive proposal $q(x)$, that generates an initial location x_0 from a simple parametric distribution q_0, then iterates M steps of an MCMC kernel T:[1]

$$q(x) = \int q_0(x_0) \left(\prod_{i=1}^{M} T(x_{i-1} \to x_i) \right) \delta_{x_M}(x) \mathrm{d}x_{0:M}.$$

Even when q_0 is a poor approximation to $p(x \mid y)$, $q(x)$ may be close to the posterior, if M is sufficiently high. However, because the density $q(x)$ cannot be efficiently evaluated, we cannot use $q(x)$ as a proposal within importance sampling (we have no way to evaluate the importance weight $\frac{p(x,y)}{q(x)}$), nor as a variational family in VI (we cannot estimate the ELBO $\mathcal{L} = \mathbb{E}_{x \sim q}[\log \frac{p(x,y)}{q(x)}]$ or its gradient, making it impossible to learn p's or q's parameters).

Approximating proposal densities with meta-inference. RAVI's goal is to enable inference even when we cannot compute the marginal densities of our proposals and variational families exactly. To apply RAVI, we must specify not just the proposal itself but also a *meta-inference* algorithm, bundled with the proposal into an *inference strategy*:

Definition. An *inference strategy* $\mathcal{S}$ *targeting* π specifies:

- a posterior approximation $\mathcal{S}.q(x) \approx \pi(x)$[2] that either has an efficient density evaluator, or is the marginal distribution of a joint distribution with a tractable density, i.e. $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x) \mathrm{d}r$, and,
- if $\mathcal{S}.q$'s marginal density cannot be efficiently evaluated, a *meta-inference strategy* $\mathcal{S}.\mathcal{M}$, assigning to each value of x an inference strategy $\mathcal{S}.\mathcal{M}(x)$ targeting $\mathcal{S}.q(r \mid x)$.

[1] Why incorporate M MCMC steps into a proposal q, rather than simply running MCMC? Several reasons: (1) if we use q as an importance sampling proposal, the importance weights are unbiased estimates of the marginal likelihood $p(y)$, which we can use to evaluate our model; (2) if we use q as a variational family, we can optimize the ELBO to learn parameters of the initial proposal or the MCMC transition kernel; and (3) if we generate many importance sampling particles using q, their importance weights can in theory correct for the bias of finite-sample MCMC.

[2] To simplify the exposition, we assume that if an inference strategy $\mathcal{S}$ targets π, then the approximation $\mathcal{S}.q$ is *mutually absolutely continuous* with π, i.e. the measure-zero events under π are exactly the same as those under $\mathcal{S}.q$. This requirement can be relaxed somewhat; see Appendix C.

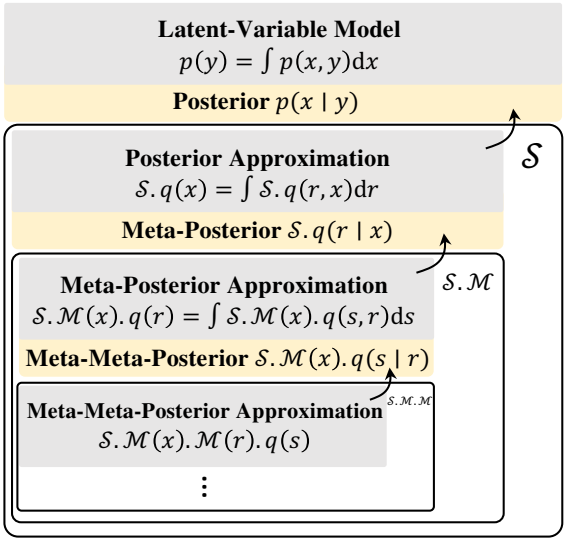


Figure 1: Structure of a RAVI inference strategy $\mathcal{S}$ targeting the posterior $p(x \mid y)$ of a latent-variable model. The proposal $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x)\mathrm{d}r$ has an intractable marginal density, so the strategy also specifies a *meta-inference* strategy $\mathcal{S}.\mathcal{M}$ that targets $\mathcal{S}.q(r \mid x)$. Nesting continues until the q approximation at some layer has a tractable density, at which point no further meta-inference is needed.

Figure 1 illustrates the recursive structure of an inference strategy. The key novelty is the inclusion of *meta-inference*, in the form of *meta-posterior approximations*: additional proposals that the user specifies for inferring auxiliary variables introduced by existing proposal distributions. In our running example, we take $\mathcal{S}.q(x)$ to be our MCMC-based posterior approximation: it lacks a tractable density, but is the marginal of a tractable joint density $\mathcal{S}.q(x_{0:M}, x)$ over entire MCMC traces. A *meta-posterior approximation*, then, is a probability distribution $\mathcal{S}.\mathcal{M}(x).q(x_{0:M})$ that approximates the *meta-posterior* $\mathcal{S}.q(x_{0:M} \mid x)$: the distribution over traces of the MCMC chain, given the final location x.

The meta-posterior approximations enable RAVI to estimate the intractable marginal density of the top-level posterior approximation, to compute weights and gradients:

In Monte Carlo: If $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x)\mathrm{d}r$ is intended for use as a Monte Carlo proposal, RAVI uses meta-inference to obtain an unbiased estimate of $\frac{1}{\mathcal{S}.q(x)}$ (Algorithm 2), which is then multiplied by $p(x, y)$ to estimate the importance weight $\frac{p(x,y)}{\mathcal{S}.q(x)}$. This process relies on the *harmonic mean identity* [33], that for any meta-posterior approximation h,

$$\mathbb{E}_{\mathcal{S}.q(r|x)}\left[\frac{h(r)}{\mathcal{S}.q(r, x)}\right] = \frac{1}{\mathcal{S}.q(x)}\mathbb{E}\left[\frac{h(r)}{\mathcal{S}.q(r \mid x)}\right] = \frac{1}{\mathcal{S}.q(x)}.$$

(Harmonic mean estimators are infamous for having potentially infinite variance, but only when h is set to a broad prior; we give a general analysis of the variance of RAVI's importance weights in Section 4.)

In Variational Inference: If $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x)\mathrm{d}r$ is intended as a variational family, then RAVI uses the meta-posterior approximation to formulate an *upper bound* on $\log \mathcal{S}.q(x)$: for any meta-posterior approximation $h(r)$,

$$\log \mathcal{S}.q(x) \leq \mathcal{U}(x) := \mathbb{E}_{\mathcal{S}.q(r|x)}[\log \mathcal{S}.q(r, x) - \log h(r)].$$

This follows from Jensen's inequality, and the harmonic mean identity from above. With this upper bound in hand, we formulate a surrogate ELBO $\mathcal{L}_{\mathcal{S}} = \mathbb{E}_{\mathcal{S}.q(x)}[\log p(x, y) - \mathcal{U}(x)] \leq \mathcal{L}$, which we can tractably estimate and optimize via stochastic gradient descent (Algorithm 3).

In Section 3, we show how similar estimators can be built up recursively when the meta-posterior approximations themselves have intractable marginal densities.

A meta-inference strategy that recovers the MCVI objective [37]. In our running example, where the auxiliary randomness r is a trace $x_{0:M}$ of locations visited by MCMC, one option for meta-inference is to learn neurally parameterized reverse Markov kernels $R_i(x_{i+1} \to x_i)$, and apply them in sequence to infer a plausible trace of MCMC steps leading to the final location x:

$$\mathcal{S}.\mathcal{M}(x).q(x_{0:M}) = \delta_x(x_M) \prod_{i=0}^{M-1} R_i(x_{i+1} \to x_i).$$

This approximation to $\mathcal{S}.q(x_{0:M} \mid x)$ has a tractable density, and so completely specifies the meta-inference strategy $\mathcal{S}.\mathcal{M}$; there is no need to specify a *meta*-meta-inference strategy. Given $\mathcal{S}$, RAVI optimizes the surrogate objective $\mathcal{L}_{\mathcal{S}} = \mathbb{E}_{x \sim \mathcal{S}.q}[\log p(x, y) - \mathcal{U}_{\mathcal{S}.\mathcal{M}(x)}]$, where

$$\mathcal{U}_{\mathcal{S}.\mathcal{M}(x)} = \mathbb{E}_{x_{0:M} \sim \mathcal{S}.q(x_{0:M}|x)}\left[\log \frac{\mathcal{S}.q(x_{0:M}, x)}{\mathcal{S}.\mathcal{M}(x).q(x_{0:M})}\right].$$

For the above choice of $\mathcal{S}.\mathcal{M}$, the RAVI objective $\mathcal{L}_{\mathcal{S}}$ exactly coincides with the Markov Chain Variational Inference (MCVI) objective of Salimans et al. [37]. In fact, RAVI unifies and generalizes many existing methods; 10 examples are collected in Appendix B.

Analyzing MCVI within the RAVI framework. Framing MCVI as a RAVI algorithm lets us analyze it using general theory about RAVI objectives. For example, the relative tightness of the bound $\mathcal{L}_{\mathcal{S}}$ is controlled by the quality of meta-inference:

$$\mathcal{L} - \mathcal{L}_{\mathcal{S}} = \mathbb{E}_{\mathcal{S}.q(x)}[KL(\mathcal{S}.q(x_{0:M} \mid x) || \mathcal{S}.\mathcal{M}(x).q(x_{0:M}))].$$

We can use this characterization to analyze the MCVI objective's behavior as M grows, i.e., as MCMC steps are added. Informally, as the MCMC chain begins to mix, the marginal distribution $\mathcal{S}.q(x)$ over the final location of the chain should grow closer to the posterior $p(x \mid y)$, tightening the (intractable) ELBO $\mathcal{L}$. Unfortunately, the meta-inference gap $\mathcal{L} - \mathcal{L}_{\mathcal{S}}$ *grows* with M, unless each kernel R_i exactly

RAVI Inference Strategy 1: RAVI-MCVI

Posterior Approx. `rmcvi(`M, K`).q()`

Target of inference : latent variable x

Auxiliary variables : MCMC trace $x_{0:M}$

$x_0 \sim q_0$

for $i \in 1, \ldots, M$ **do**

  $x_i \sim T(x_{i-1} \to \cdot)$

return x_M

Meta-Posterior Approx. `rmcvi(`M, K`).M(`x`).q()`

Target of inference : MCMC trace $x_{0:M}$

Auxiliary variables : SMC particles $x_{0:M}^{1:K}$, ancestor indices $a_0, a_{1:M}^{1:K}$

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

  $(x_M^k, w_M^k, t_k) \leftarrow (x, q_m(x), [x])$

for $i \in M-1, \ldots, 0$ **do**

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

    $a_{i+1}^k \sim \text{Discrete}(w_{i+1}^{1:K})$

    $x_i^k \sim R_i(x_{i+1}^{a_{i+1}^k} \to \cdot)$ `// MCVI backward kernel`

    $w_i^k \leftarrow \frac{q_i(x_i^k) T(x_i \to x_{i+1}^{a_{i+1}^k})}{q_{i+1}(x_{i+1}^{a_{i+1}^k}) R_i(x_{i+1}^{a_{i+1}^k} \to x_i^k)}$

    $t_k \leftarrow [x_i^k, t_k^{a_{i+1}^k} \ldots]$

$a_0 \sim \text{Discrete}(w_0^{1:K})$

return t_{a_0}

Meta-Meta-Posterior Approx. `rmcvi(`M, K`).M(`x`).M(`$x_{0:M}$`).q()`

Target of inference : SMC particles $x_{0:M}^{1:K}$, ancestor indices $a_0, a_{1:M}^{1:K}$

Auxiliary variables : None

for $i \in 0, \ldots, M$ **do**

  $b_i \sim \text{Uniform}(1, K)$

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

  $(x_M^k, w_M^k) \leftarrow (x, q_m(x)])$

for $i \in M-1, \ldots, 0$ **do**

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

    if $k = b_i$ **then**

      $(a_{i+1}^k, x_i^k) \leftarrow (b_{i+1}, x_i)$

    else

      $a_{i+1}^k \sim \text{Discrete}(w_{i+1}^{1:K})$

      $x_i^k \sim R_i(x_{i+1}^{a_{i+1}^k} \to \cdot)$

    $w_i^k \leftarrow \frac{q_i(x_i^k) T(x_i \to x_{i+1}^{a_{i+1}^k})}{q_{i+1}(x_{i+1}^{a_{i+1}^k}) R_i(x_{i+1}^{a_{i+1}^k} \to x_i^k)}$

$a_0 \leftarrow b_0$

return $(a_0, a_{1:M}^{1:K}, x_{0:M}^{1:K})$

captures the local posterior $\mathcal{S}.q(x_i \mid x_{i+1})$. (This can be seen as an instance of the well-known *degeneracy problem* of sequential importance sampling [15, Proposition 1].) As MCMC converges, the rate of improvement in $\mathcal{L}$ slows, and the meta-inference penalty for increasing the chain's length eventually outweighs the benefit of improving the posterior approximation $\mathcal{S}.q$. The red curves in Figure 3 show this phenomenon playing out on two toy targets: we see that $\mathcal{L}_{\mathcal{S}}$ does become tighter as more MCMC steps are added, but only to a point, before the bound begins to *loosen*.

Resolving the issue with improved meta-inference. RAVI clarifies that the variational bound loosens with increasing M due to poor meta-inference: as the MCMC chain grows longer, error in the learned backward kernels accumulates. This analysis also points to a solution: use a meta-inference strategy $\mathcal{S}.\mathcal{M}$ that *can* scale to longer MCMC histories.

A standard approach to resolving the degeneracy problem when inferring sequences of latent variables is *sequential Monte Carlo* (SMC) [12]. SMC tracks K hypotheses about a latent sequence, periodically weighting the hypotheses and *resampling*, to clone promising particles and cull poor ones. Using RAVI, we can use SMC for *meta-inference*: we choose $\mathcal{S}.\mathcal{M}(x).q(x_{0:M})$ to generate a collection of K possible backward MCMC trajectories, using SMC, before selecting one to return. This meta-posterior approximation is shown in RAVI Inference Strategy 1.

This algorithm does not itself have a tractable marginal density: computing $\mathcal{S}.\mathcal{M}(x).q(x_{0:M})$ would require large sums over the resampling variables and intractable integrals over the particle collection. But this is where RAVI's recursive structure comes into play: a meta-inference strategy may use an intractable meta-posterior approximation, so long as we attach a *meta*-meta-inference strategy $\mathcal{S}.\mathcal{M}(x).\mathcal{M}(x_{0:M}).q(a_0, a_{1:M}^{1:K}, x_{0:M}^{1:K})$. In this case meta-meta-inference must infer the auxiliary variables of SMC (ancestor variables and unchosen trajectories), given the final chosen trajectory $x_{0:M}$. For this we can use the *conditional SMC* algorithm [1], which runs SMC, with the same auxiliary variables, but constrained to ensure that one of the K particles traces the observed trajectory $x_{0:M}$. Because cSMC introduces no new auxiliary variables, it has a tractable density, and there is no need to specify a fourth layer of meta-inference. The full tower of posterior approximations is given in RAVI Inference Strategy 1.

In Section 5, we compare MCVI to `rmcvi`, for different K and M. Figure 3 shows that meta-inference error is greatly reduced by using SMC, so that the variational bound $\mathcal{L}_{\mathcal{S}}$ continues to tighten as the MCMC chain grows longer.

Using the inference strategy within a Monte Carlo algorithm, to estimate marginal likelihoods from MCMC results. Our inference strategy $\mathcal{S}$ can also be used as proposal within Monte Carlo algorithms, such as importance sampling. In the context of our example, where $\mathcal{S}.q$ incorporates M steps of a Markov chain, this allows us to assign an *importance weight* to each run of the Markov chain. The weight is an unbiased estimate of the marginal likelihood $p(y)$ of the model; thus, we can view the algorithm as a way to derive marginal likelihood estimates from MCMC runs, a task of long-standing interest in the Monte Carlo community [31]. In Section 5, we show that in some settings MCVI compares favorably a standard algorithm for the task, annealed importance sampling (AIS) [31].

3 ALGORITHMS

In this section, we present algorithms for using RAVI inference strategies within Monte Carlo and variational inference algorithms, as proposals and variational families.

RAVI for Importance Sampling and SMC. In importance sampling and SMC algorithms, proposals q are used to (1) generate proposed values $x \sim q$, and (2) compute importance weights $\frac{p(x)}{q(x)}$. But in both IS and SMC, it suffices to produce *unbiased estimates* of $\frac{p(x)}{q(x)}$ [7]. RAVI exploits this

Recursive Monte Carlo Estimation

Algorithm 1: RAVI Importance Sampling (`IMPORTANCE`)
Input: unnormalized target $\tilde{\pi}(x) = Z\pi(x)$
Input: inference strategy $\mathcal{S}$
Output: $(x, \hat{Z})$ properly weighted for $\pi(x)$, s.t. $\mathbb{E}[\hat{Z}] = Z$

if $\mathcal{S}.q$ *has a tractable marginal density* **then**
  $x \sim \mathcal{S}.q$
  $w \leftarrow \frac{1}{\mathcal{S}.q(x)}$
else if $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x)dr$ **then**
  $(r, x) \sim \mathcal{S}.q$
  $w \leftarrow \texttt{HME}(\mathcal{S}.q(\cdot \mid x), r, \mathcal{S}.\mathcal{M}(x))$
return $(x, w\tilde{\pi}(x))$

Algorithm 2: RAVI Harmonic Mean Estimation (`HME`)
Input: unnormalized target $\tilde{\pi}(x) = Z\pi(x)$
Input: exact sample $x \sim \pi$
Input: inference strategy $\mathcal{S}$
Output: unbiased estimate $\check{Z}^{-1}$ of Z^{-1}

if $\mathcal{S}.q$ *has a tractable marginal density* **then**
  $w \leftarrow \mathcal{S}.q(x)$
else if $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x)dr$ **then**
  $(r, w) \leftarrow \texttt{IMPORTANCE}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x))$
return $w/\tilde{\pi}(x)$

Recursive Variational Objectives and Gradient Estimation

Algorithm 3: RAVI ELBO and gradient estimator (`ELBO`∇)
Input: model $p(x, y)$
Input: data y
Input: inference strategy $\mathcal{S}$
Output: unbiased estimates of $\mathcal{L}(p, y, \mathcal{S})$ and of $\nabla_\theta \mathcal{L}(p, y, \mathcal{S})$

if $\mathcal{S}.q$ *has a tractable marginal density* **then**
  $x \sim \mathcal{S}.q$
  $(\hat{U}, \widehat{\nabla_\theta}) \leftarrow (\log \mathcal{S}.q(x), \nabla_\theta \log \mathcal{S}.q(x) \cdot (1 + \log \mathcal{S}.q(x)))$
  $\mathbf{g} \leftarrow \nabla_\theta \log \mathcal{S}.q(x)$
else if $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x)dr$ **then**
  $(r, x) \sim \mathcal{S}.q$
  $(\hat{U}, \widehat{\nabla_\theta}, \mathbf{g}) \leftarrow \texttt{EUBO}\nabla(\mathcal{S}.q, x, r, \mathcal{S}.\mathcal{M}(x))$
$\hat{L} \leftarrow \log p(x, y) - \hat{U}$
$\widehat{\nabla_\theta}' \leftarrow \nabla_\theta \log p(x, y) + \mathbf{g} \log p(x, y) - \widehat{\nabla_\theta}$.
return $(\hat{L}, \widehat{\nabla_\theta}')$

Algorithm 4: RAVI EUBO and gradient estimator (`EUBO`∇)
Input: model $p(x, y)$
Input: data y
Input: exact sample $x \sim p(x \mid y)$
Input: inference strategy $\mathcal{S}$
Output: unbiased estimates of $\mathcal{U}(p, y, \mathcal{S})$ and $\nabla_\theta \mathcal{U}(p, y, \mathcal{S})$
Output: quantity $\mathbf{g}$ (see Thm. 2)

if $\mathcal{S}.q$ *has a tractable marginal density* **then**
  $(\hat{L}, \widehat{\nabla_\theta}) \leftarrow (\log \mathcal{S}.q(x), \nabla_\theta \log \mathcal{S}.q(x))$
else if $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x)dr$ **then**
  $(\hat{L}, \widehat{\nabla_\theta}) \leftarrow \texttt{ELBO}\nabla(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))$
$\hat{U} \leftarrow \log p(x, y) - \hat{L}$
$\mathbf{g} \leftarrow \nabla_\theta \log p(x, y)$
$\widehat{\nabla_\theta}' \leftarrow \nabla_\theta \log p(x, y) + \mathbf{g} \cdot \hat{U} - \widehat{\nabla_\theta}$
return $(\hat{U}, \widehat{\nabla_\theta}', \mathbf{g})$

MCMC

Algorithm 5: RAVI Metropolis-Hastings
Input: model $\tilde{\pi}(x) = Z \int \pi(r, x)\mathrm{d}r$
Input: proposal $q(x'; x) = \int q(s, x'; x)\mathrm{d}s$
Input: family $\mathcal{S}(x)$ of inference strategies targeting $\pi(r \mid x)$
Input: family $\mathcal{M}(x, x')$ of inference strategies targeting $q(s \mid x'; x)$
Input: initial position x and estimate $\hat{Z}_x$ of $\tilde{\pi}(x)$
Output: next position x' and estimate $\hat{Z}_{x'}$ of $\tilde{\pi}(x')$

$(s, x') \sim q(s, x'; x)$
$w_{x'} \leftarrow \texttt{HME}(q(\cdot, x'; x), s, \mathcal{M}(x, x'))$
$(_, w_x) \leftarrow \texttt{IMPORTANCE}(q(\cdot \mid x; x'), \mathcal{M}(x', x))$
$(_, \hat{Z}_{x'}) \leftarrow \texttt{IMPORTANCE}(\pi(\cdot \mid x'), \mathcal{S}(x'))$
$u \sim \text{Uniform}(0, 1)$
if $u < min(1, \frac{\hat{Z}_{x'}}{\hat{Z}_x} w_{x'} w_x)$ **then**
  return $(x', \hat{Z}_{x'})$
else
  return $(x, \hat{Z}_x)$

degree of freedom to generate proper importance weights even when $q(x)$ is intractable. Suppose $\tilde{\pi} = Z\pi$ is an unnormalized target density, and $\mathcal{S}$ is a RAVI inference strategy targeting π. Algorithm 1 simulates $x \sim \mathcal{S}.q$ and computes an unbiased estimate $\hat{Z}$ of $\frac{\tilde{\pi}(x)}{\mathcal{S}.q(x)}$:

Theorem 1. *Let $\tilde{\pi}(x) = Z\pi(x)$ be an unnormalized target density, and $\mathcal{S}$ an inference strategy targeting $\pi(x)$. Then:*

- *`IMPORTANCE`$(\mathcal{S}, \tilde{\pi})$ generates $(x, \hat{Z})$ with $x \sim \mathcal{S}.q$ and $\mathbb{E}[\hat{Z} \mid x] = Z\frac{\pi(x)}{\mathcal{S}.q(x)}$. Furthermore, the unconditional expectation $\mathbb{E}[\hat{Z}(\tilde{\pi}, \mathcal{S})] = Z$.*
- *When $x \sim \pi$, `HME`$(\mathcal{S}, x, \tilde{\pi})$ generates $\check{Z}$ with $\mathbb{E}[\check{Z}^{-1}] = Z^{-1}$.*

When $\mathcal{S}.q$ has a tractable marginal density, Algorithm 1 computes an exact importance weight. Otherwise, it calls Algorithm 2, which uses the meta-inference strategy $\mathcal{S}.\mathcal{M}(x)$ to estimate $\frac{1}{\mathcal{S}.q(x)}$. The proof of Theorem 1 is by induction on the level of nesting in the strategy (see Appendix A).

RAVI for MCMC. When models or proposals (or both) in a Metropolis-Hastings sampler do not have tractable closed-form densities, RAVI inference strategies enable computation of MH acceptance probabilities (Algorithm 5). Intuitively, to compute the usual Metropolis-Hastings acceptance probability $\alpha = \frac{\tilde{\pi}(x')q(x;x')}{\tilde{\pi}(x)q(x';x)}$, Algorithm 5 estimates the necessary proposal densities, using `HME` for the forward proposal density that appears in the denominator, and `IMPORTANCE` for the backward proposal density that appears in the numerator. If necessary, it also uses `IMPORTANCE` to estimate the new model density $\tilde{\pi}(x')$.

We show the algorithm implements a stationary kernel for π in Appendix A.5.

RAVI for Variational Inference. Let $p_\theta(x, y)$ be a latent-variable generative model with parameters θ, and $\mathcal{S}_\theta(y)$ is a family of strategies targeting $p_\theta(x \mid y)$. Given a dataset y, variational inference can be applied to maximize (a lower bound on) $\log p_\theta(y)$, and also to optimize parameters of the posterior approximations in $\mathcal{S}_\theta$, to bring them closer (in KL divergence) to their targets. Let

$$\mathcal{L}(p, y, \mathcal{S}) := \mathbb{E}[\log \hat{Z}(p(\cdot, y), \mathcal{S})] \leq \log p(y)$$
$$\text{and } \mathcal{U}(p, y, \mathcal{S}) := \mathbb{E}[\log \check{Z}(p(\cdot, y), \mathcal{S})] \geq \log p(y),$$

where $\hat{Z}(\tilde{\pi}, \mathcal{S})$ is the estimate returned by IMPORTANCE (Alg. 1) on $\mathcal{S}$ and unnormalized target $\tilde{\pi}$, and $\check{Z}(\tilde{\pi}, \mathcal{S})$ is the inverse of the weight returned from HME (Alg. 2) when run with unnormalized target $\tilde{\pi}$, inference strategy $\mathcal{S}$, and an exact sample $x \sim \pi$. Because $\hat{Z}$ is an unbiased estimate of $p_\theta(y)$, and $\check{Z}^{-1}$ is an unbiased estimate of $p_\theta(y)^{-1}$, we have by Jensen's inequality that $\mathcal{L}(p, y, \mathcal{S})$ and $\mathcal{U}(p, y, \mathcal{S})$ are lower and upper bounds (respectively) on $\log p_\theta(y)$. As such, we can fit the model parameters θ to data y by minimizing $\mathcal{U}(p, y, \mathcal{S})$ or maximizing $\mathcal{L}(p, y, \mathcal{S})$.

Recursive stochastic gradient estimation. ELBO∇ (Alg. 3) is a procedure for estimating $\mathcal{L}(p, y, \mathcal{S})$ and its gradient $\nabla_\theta \mathcal{L}(p, y, \mathcal{S})$ with respect to the parameters θ of the model and the strategy. When $(x, y) \sim p(x, y)$, EUBO∇ (Alg. 4) estimates $\mathcal{U}(p, y, \mathcal{S})$ and the gradient $\nabla_\theta \mathbb{E}_{y \sim p}[\mathcal{U}(p, y, \mathcal{S})]$. These procedures employ score function estimation of gradients, but it is straightforward to incorporate baselines within each procedure to reduce variance. Depending on $\mathcal{S}$, the reparametrization trick may also be applicable (Appendix E).

Theorem 2. *Given a model $p_\theta(x, y)$ and an inference strategy $\mathcal{S}_\theta$ targeting $p_\theta(x \mid y)$, Alg. 3 yields unbiased estimates of $\mathcal{L}(p, y, \mathcal{S})$ and of $\nabla_\theta \mathcal{L}(p, y, \mathcal{S})$. Furthermore, when $(x, y) \sim p_\theta$, Alg. 4 yields (i) $\hat{U}$ such that $\mathbb{E}[\hat{U} \mid y] = \mathcal{U}(p, y, \mathcal{S})$, (ii) $\widehat{\nabla_\theta}$ such that $\mathbb{E}[\widehat{\nabla_\theta}] = \nabla_\theta \mathbb{E}_{y \sim p}[\mathcal{U}(p, y, \mathcal{S})]$, and (iii) a value $\mathbf{g}$ such that for any function R that does not depend on θ, $\mathbb{E}[\mathbf{g} \cdot R(y)] = \nabla_\theta \mathbb{E}_{y \sim p_\theta}[R(y)]$ if $\nabla_\theta \mathbb{E}_{y \sim p_\theta}[R(y)]$ is defined.*

In Section 4, we show the tightness of the variational bounds $\mathcal{L}$ and $\mathcal{U}$ is given by sums of KL divergences between posterior approximations in $\mathcal{S}_\theta$ and their targets. Thus, optimizing these bounds improves the posterior approximations, either encouraging mass-capturing or mode-seeking behavior.

4 THEORETICAL ANALYSIS

We now present theorems characterizing the quality of RAVI inference: Thm. 3 concerns the variance of weights in a Monte Carlo sampler, and Thm. 4 the tightness of variational bounds. In both cases, error is related to each approximation in the RAVI strategy's divergence to its target posterior.

Sampler variance in Monte Carlo. Let $\tilde{\pi} = Z\pi$ be an unnormalized target density, and $\mathcal{S}$ an inference strategy targeting π. As in Section 3, we write $\hat{Z}(\tilde{\pi}, \mathcal{S})$ for the weight returned by IMPORTANCE, and $\mathrm{Var}_{\hat{Z}}(\pi, \mathcal{S})$ for the *relative variance* of the estimator, $\mathrm{Var}(\frac{\hat{Z}(\tilde{\pi}, \mathcal{S})}{Z})$, which does not depend on Z (and therefore is a function of π, not $\tilde{\pi}$). Similarly, we write $\check{Z}(\tilde{\pi}, \mathcal{S})$ for the reciprocal of the weight returned by HME, run with an input $x \sim \pi$. $\mathrm{Var}_{\check{Z}}(\pi, \mathcal{S})$ is its relative variance, $\mathrm{Var}(\frac{Z}{\check{Z}(\tilde{\pi}, \mathcal{S})})$.

Theorem 3. *Consider an unnormalized target distribution $\tilde{\pi}(x) = Z\pi(x)$ and an inference strategy $\mathcal{S}$ targeting $\pi(x)$. Then the relative variances of the estimators $\hat{Z}(\tilde{\pi}, \mathcal{S})$ and $\check{Z}(\tilde{\pi}, \mathcal{S})$ are given by the following recursive equations:*

$$\mathrm{Var}_{\hat{Z}}(\pi, \mathcal{S}) = \chi^2(\pi || \mathcal{S}.q) + \mathbb{E}_{x \sim \mathcal{S}.q}\left[\left(\frac{\pi(x)^2}{\mathcal{S}.q(x)^2}\right) \cdot \mathrm{Var}_{\check{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))\right]$$
$$\mathrm{Var}_{\check{Z}}(\pi, \mathcal{S}) = \chi^2(\mathcal{S}.q || \pi) + \mathbb{E}_{x \sim \pi}\left[\left(\frac{\mathcal{S}.q(x)^2}{\pi(x)^2}\right) \cdot \mathrm{Var}_{\hat{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))\right]$$

When $\mathcal{S}.q$ is tractable, the second term of each sum is 0.

Tightness of variational bounds. In VI, the tightness of the variational bounds $\mathcal{L}$ and $\mathcal{U}$ can be characterized as a sum of a KL divergence and a term measuring meta-inference error. The random variables $\hat{\mathcal{L}}$ and $\hat{\mathcal{U}}$ returned by ELBO∇_θ and EUBO∇_θ, respectively, are unbiased estimators of $\mathcal{L}(p, y, \mathcal{S})$ and $\mathcal{U}(p, y, \mathcal{S})$, and so can also be viewed as *biased* estimators of $\log p(y)$. Writing their bias as $\mathrm{Bias}_{\mathcal{L}}(p, y, \mathcal{S})$ (and similarly for $\mathcal{U}$), we have:

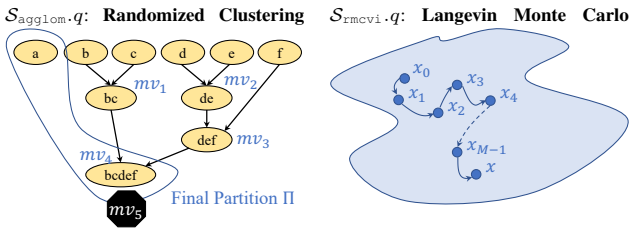
Theorem 4. *Consider a joint distribution $p(x, y)$ and an inference strategy $\mathcal{S}$ targeting $p(x \mid y)$. Then the following equations give the bias of $\hat{\mathcal{L}}$ and $\hat{\mathcal{U}}$ as estimators of $\log p(y)$:*

$$\mathrm{Bias}_{\mathcal{L}}(p, y, \mathcal{S}) = -\mathrm{KL}(\mathcal{S}.q || p(\cdot \mid y)) - \mathbb{E}_{x \sim \mathcal{S}.q}[\mathrm{Bias}_{\mathcal{U}}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))]$$
$$\mathrm{Bias}_{\mathcal{U}}(p, y, \mathcal{S}) = \mathrm{KL}(p(\cdot \mid y) || \mathcal{S}.q) - \mathbb{E}_{x \sim p(\cdot \mid y)}[\mathrm{Bias}_{\mathcal{L}}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))]$$

where the second term in each equation is 0 when $\mathcal{S}.q$ has a tractable marginal density.

Maximizing $\mathcal{L}$, or minimizing $\mathcal{U}$, also minimizes these KL divergences. In particular, maximizing $\mathcal{L}(p, y, \mathcal{S})$ minimizes a 'mode-seeking' KL from $\mathcal{S}.q$ to the posterior, whereas minimizing $\mathbb{E}_{y \sim p}[\mathcal{U}(p, y, \mathcal{S})]$, e.g. by following the gradients of Alg. 4, implements amortized variational inference, and encourages $\mathcal{S}.q$ to cover the mass of the posterior.

Inference and Meta-Inference. In both Theorems 3 and 4, the first term of the sum is a divergence between $\mathcal{S}.q(x)$, the intractable posterior approximation, and the actual target posterior $p(x \mid y)$. The other term measures the *expected*



$$\mathcal{S}_{\texttt{agglom}}.q(\Pi) = \sum_{mv_1} \cdots \sum_{mv_{N-|\Pi|}} \prod_{i=1}^{N-|\Pi|} q(mv_i \mid mv_{<i})$$

$$\mathcal{S}_{\texttt{rmcvi}}.q(x) = \int_{\mathbb{R}^M} q_0(x_0) \left(\prod_{i=1}^{M-1} q(x_i \mid x_{i-1}) \right) q(x \mid x_{M-1}) \mathrm{d}x_{0:M-1}$$

Figure 2: Illustrations of the proposals $\mathcal{S}.q$ used in each experiment. In each case, $\mathcal{S}.q$ makes a sequence of auxiliary choices before returning a final proposal (the clustering Π, or the location x). Sequential Monte Carlo meta-inference is used to marginalize the sequence of auxiliary variables introduced by the inference process (the merges mv_i in `agglom`, and the locations x_i in `rmcvi`).

	Inference	**Meta-inference**	**Meta-meta-inference**
`agglom`	*Discrete*: 3.0×10^{1928}	*Discrete*: $\prod_{n=\lvert\Pi\rvert}^{1000} \frac{n(n-1)}{2}$	*Discrete*: $(K-1) \cdot (\prod_{n=\lvert\Pi\rvert}^{1000} \frac{n(n-1)}{2}) \cdot (1000 - \lvert\Pi\rvert)(K-1)!$
`rmcvi`	*Continuous*: 1	*Continuous*: M	*Continuous*: $(K-1) \cdot M$, *Discrete*: $M(K-1)!$

Table 2: Dimensionality of the continuous latent space, and cardinality of the discrete latent space, over which each layer's inference problem is defined. K is the number of SMC particles used for meta-inference (maximum 50 for `rmcvi`, 5 for `agglom`). In `rmcvi`, M is the number of MCMC steps (maximum 100 in our experiments).

quality of meta-inference. Thus the overall error of a RAVI algorithm can be understood as decomposing cleanly into (1) the mismatch between the posterior and the intractable proposal, and (2) the error introduced by meta-inference.

5 EXPERIMENTS

5.1 IMPROVING MCVI

In Section 2, we developed a variant of Salimans et al. [37]'s MCVI algorithm that used SMC for meta-inference. In Figure 3, we compare vanilla MCVI to the RAVI variant, with varying K (number of particles used for meta-inference) and M (number of MCMC steps in the variational family).

Experimental details.[3] For the MCMC kernel T, we use Langevin ascent with step size 0.015. For the meta-inference proposals $R_i(x_{i+1} \rightarrow x_i)$, we use $\mathcal{N}(x_i; f_\mu(x_{i+1}, i), e^{f_{\log\sigma}(x_{i+1}, i)})$, where f is a 4-layer MLP, the step number i is encoded as a one-hot vector, and f outputs the mean μ and log standard deviation $\log\sigma$ for a conditionally Gaussian proposal. The same f is used for each experiment, and is trained on forward rollouts of MCMC (equivalent to using Alg. 3 on `rmcvi` with $K = 1$). The unimodal model is Gaussian with $\sigma = 0.2$, and the multimodal model is a mixture of 3 Gaussians with standard deviations 0.2, 0.3, and 2.0. The distributions q_i used for importance weighting during sequential Monte Carlo meta-inference are Gaussians with learned μ and σ.

Results. Figure 3 plots the gap $\log p(y) - \mathcal{L}$ for each algorithm's variational bound $\mathcal{L}$. By Theorem 4 this gap is the sum of two terms: $KL(\mathcal{S}.q || p(x \mid y))$ and the expected meta-inference divergence $\mathbb{E}_{x \sim \mathcal{S}.q}[KL(\mathcal{S}.q(x_{0:M} \mid x) || \mathcal{S}.\mathcal{M}(x).q(x_{0:M}))]$. The first term is constant across the algorithms, since they all use the same MCMC-based posterior approximation, so the plots primarily illustrate differences in the quality of meta-inference. MCVI's meta-inference steadily worsens as the chain's length grows, and after 15-25 steps, the meta-inference cost of adding new steps outweighs the benefits to $\mathcal{S}_{\texttt{mcvi}}.q$, causing the bound $\mathcal{L}$ to loosen. Our variant, with SMC meta-inference, does not suffer the same penalty, and continues to improve as more steps are added. As discussed in Section 2, the same inference strategy (`rmcvi`) can be used within an importance sampler to derive unbiased marginal likelihood estimates from MCMC runs. The right-hand plot in Figure 3 shows that this technique can yield accurate estimates with less computation than AIS [31], at least on simple targets. (To fairly account for the computational cost of meta-inference, in the RAVI algorithm we multiply M by K when plotting the total number of MCMC steps.) Because the variance of AIS is bounded below by sums of divergences between subsequent pairs of intermediate target distributions, the MCMC chain must be long enough to support a very fine annealing schedule, without large jumps. By contrast, RAVI-MCVI requires only that the marginal distribution of the chain be a good approximation to the posterior, and that SMC meta-inference is sufficiently accurate. For some problems, this may be less expensive than the long chain required by AIS.

5.2 AGGLOMERATIVE CLUSTERING FOR DIRICHLET PROCESS MIXTURES

A promising application of RAVI is to transform heuristic randomized algorithms into unbiased and consistent Monte Carlo estimators, by using them as proposal distributions. In this section, we design a RAVI inference strategy for clustering in Dirichlet process mixtures, based on a randomized agglomerative clustering algorithm (Inference Strategy 2).

Datasets and Models. We test our algorithm on three clus-

[3] Code is available: https://github.com/probcomp/ravi-uai-2022

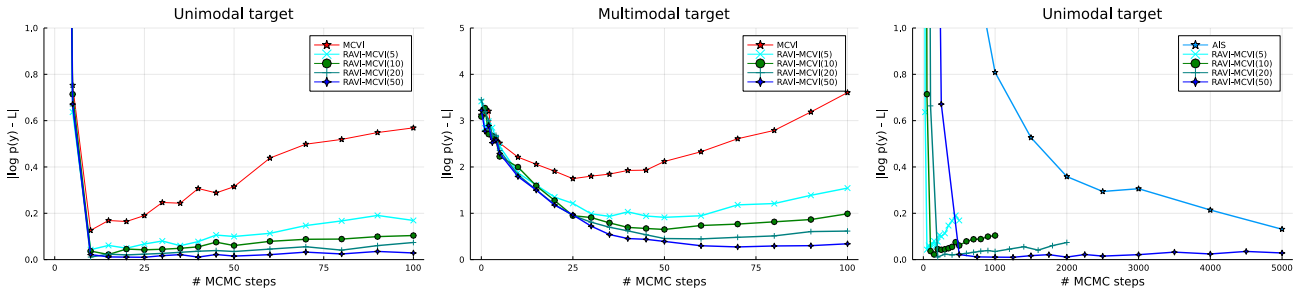


Figure 3: Improving Markov Chain Variational Inference with RAVI. **Left and Middle:** On unimodal and multimodal targets, MCVI begins to degrade after 15-25 steps of MCMC. RAVI-MCVI with sufficiently many particles continues to improve as more MCMC steps are added. **Right:** When MCMC converges quickly to a reasonable approximation of the posterior, RAVI-MCVI can give more accurate estimates of marginal likelihoods than standard techniques such as AIS. The x axis of this plot counts total MCMC steps simulated, whether as part of inference or meta-inference; for RAVI-MCVI(K), this is KM, where M is the length of the forward Markov chain and K is the number of SMC particles used for meta-inference.

RAVI Inference Strategy 2: Agglomerative Clustering

Posterior Approx. `agglom(X, K).q()`

Target of inference : partition Π of dataset X

Auxiliary variables : merge sequence $mv_{1:|X|-|\Pi|}$

1 $\Pi \leftarrow \{\{x\} \mid x \in X\}$ `// Initial partition`

2 **for** $l \in 1, \ldots, |X|$ **do**

3 **for** *unordered pair* $\{i, j\}$ *of clusters in* Π **do**

4 $w_{\{i,j\}} \leftarrow \pi((\Pi \setminus \{i, j\}) \cup \{i \cup j\})$

5 $w_{\text{stop}} = \pi(\Pi)$

6 $All \leftarrow \{\text{stop}\} \cup \{\{i, j\} \mid i, j \in \Pi\}$

7 $mv_l \sim \text{Discrete}(\{m \Rightarrow w_m \mid m \in All\})$

8 **if** $mv_l = stop$ **then break**

9 $\Pi \leftarrow (\Pi \setminus mv_l) \cup (\cup mv_l)$ `// Perform the merge`

10 **return** Π

Meta-Posterior Approx. `agglom(X, K).M(Π).q()`

Target of inference : merge sequence $mv_{1:|X|-|\Pi|}$

Auxiliary variables : particles $mv^{1:K}_{1:|X|-|\Pi|}$, ancestors $a^{1:K}_{1:|X|-|\Pi|}$

1 **for** $k \in 1, \ldots, K$ **do** $\Pi^k_0, tr_k \leftarrow \{\{x\} \mid x \in X\}, []$

2 **for** $l \in 1, \ldots, |X| - |\Pi|$ **do**

3 **for** $k \in 1, \ldots, K$ **do**

4 **for** *unordered pair* $\{i, j\}$ *in* Π^k_{l-1} **do**

5 $w_{\{i,j\}} \leftarrow \pi((\Pi^k_{l-1} \setminus \{i, j\}) \cup \{i \cup j\})$

6 $w_{\text{stop}} = \pi(\Pi^k_{l-1})$

7 $All \leftarrow \{\text{stop}\} \cup \{\{i, j\} \mid i, j \in \Pi^k_{l-1}\}$

8 $Ok \leftarrow \{\{i, j\} \in All \mid \exists c \in \Pi. i \cup j \subseteq c\}$

9 $mv_l \sim \text{Discrete}(\{m \Rightarrow w_m \mid m \in Ok\})$

10 $\Pi^k_{l-1} \leftarrow (\Pi^k_{l-1} \setminus mv_l) \cup (\cup mv_l)$

11 $tr_k \leftarrow [tr_k \ldots, mv_l]$

12 $W^k_l \leftarrow \frac{\sum_{m \in Ok} w_m}{\sum_{m \in All} w_m}$

13 **for** $k \in 1, \ldots, K$ **do**

14 $a^k_l \sim \text{Discrete}(W^{1:K}_l)$ `// resampling step`

15 $\Pi^k_l, tr_k \leftarrow \Pi^{a^k_l}_{l-1}, tr_{a^k_l}$

16 **return** $[tr_1 \ldots, stop]$

Meta-Meta-Posterior `agglom(X, K).M(Π).M(`$mv_{1:|X|-|\Pi|}$`).q()`

`// Conditional SMC (omitted for space, but similar to that of rmcvi)`

tering problems. The first is a synthetic 1D dataset sampled from a Dirichlet process (DP) mixture prior. The second is a standard benchmark dataset of galaxy velocities [16, 10], which we model using a DP mixture with Gaussian likelihoods and $\alpha = 1$. The last is a data-cleaning task, correcting typos in 1k strings from Medicare records [25]. We adapt the generative model of Lew et al. [21]. Using an English character-level bigram model $H(s) = h(s_1) \prod_{i=2}^{|s|} h(s_i \mid s_{i-1})$, we model the data $\{y_i\}$ with a DP prior:

$$G \sim DP(H, \alpha = 1.0), \quad x_i \mid G \sim G, \quad y_i \mid x_i \sim f(\cdot \mid x_i)$$

Here, the likelihood $f(y_i \mid x_i)$ models typos. We set f to be

$$f(y_i \mid x_i) \propto \begin{cases} \mathbf{1}[x_i = y_i] & (x_i, y_i) \notin \mathcal{L} \times \mathcal{L} \\ \frac{\text{NegBin}(\tau(x_i, y_i); \lceil \frac{|s|}{5} \rceil, 0.9)}{(5.09|s|)^{\tau(x_i, y_i)}} & (x_i, y_i) \in \mathcal{L} \times \mathcal{L} \end{cases},$$

where $\tau(x_i, y_i)$ is the Damerau-Levenshtein edit distance between x_i and y_i, and $\mathcal{L}$ is the set of all observed strings $\{y \mid \exists i. y = y_i\}$.[4] We perform inference in a collapsed version of the model, with the x_i marginalized out:

$$\Pi \sim CRP(n = N, \alpha = 1.0)$$
$$y_I \mid \Pi \sim F(y_I).$$

Here Π is a partition, I ranges over the components of Π (each of which is a subset of data indices), and $F(y_I) = \sum_{x \in \mathcal{L}} h(x) \prod_{i \in I} f(y_i \mid x)$ is the marginal likelihood of y_I as a sequence of noisy observations of a latent string.

Baseline. We compare to an SMC baseline, inspired by PClean's inference [21], that targets a sequence of posteriors, where the tth posterior incorporates the first t datapoints. The SMC proposal is locally optimal, assigning the newest datapoint to an existing component I with probability proportional to $\frac{|I|}{t+\alpha-1} \cdot F(y_I \cup \{y_t\})$, or to a new component with probability proportional to $\frac{\alpha}{t+\alpha-1} \cdot F(\{y_t\})$. We do not compare to a Gibbs sampling baseline, as Gibbs sampling does not yield marginal likelihood estimates, but do perform a Gibbs rejuvenation sweep every 20 iterations of SMC.

RAVI algorithm. We apply Algorithm 1 to the inference strategy `agglom` (Inference Strategy 2). The strategy is based on a randomized agglomerative clustering algorithm: each datapoint begins in its own cluster (L1), and we repeatedly choose to either merge two clusters (L9) or stop and propose the current partition (L8). The sequence of merge decisions $mv_1, \ldots, mv_{|X|-|\Pi|}$ are the auxiliary variables of

[4] We assume that the data $\mathcal{L}$ includes at least one example of every clean string. When $x_i \in \mathcal{L}$, we model a negative-binomially distributed number of typos, where the number of trials depends on the length of the string.

	$\hat{\mathcal{L}}$
Gaussian likelihood [10], synthetic data	
SMC + adapted proposals	-125.09 ± 0.38
RAVI agglomerative clustering	-125.97 ± 1.62
Gaussian likelihood [10], Galaxy data [16]	
SMC + adapted proposals	-426.20 ± 1.26
RAVI agglomerative clustering	$\mathbf{-423.03 \pm 0.94}$
PClean typos likelihood [21], Hospital data [25]	
SMC + adpated proposals	$-40,239 \pm 1,532$
RAVI agglomerative clustering	$\mathbf{-13,851.0 \pm 0.01}$

Table 3: RAVI agglomerative clustering vs. SMC baseline.

our proposal distribution; the final output is the proposed clustering Π. Our meta-inference $\texttt{agglom}(X, K).\mathcal{M}(\Pi).q$ infers the sequence of merges from the observed clustering Π, using K-particle SMC with proposals that mimic the forward process but choose only from a restricted set *Ok* of possible merges (L8), to avoid making any choices that disagree with Π. SMC introduces additional auxiliary variables, so we also include a conditional SMC meta-meta-posterior approximation (not shown, but nearly identical to `rmcvi`'s).

Results. Table 3 shows average log marginal likelihood estimates; higher is better. On synthetic Gaussian data, the algorithms perform comparably. On the galaxy data, RAVI agglomerative clustering finds modes that SMC misses, leading to a 3-nat improvement in the average log marginal likelihood. In the Medicare data example, SMC misses the ground-truth clustering and hypothesizes many unlikely typos to explain the data. The RAVI agglomerative clustering is less greedy, considering $O(N^2)$ possible merges at each step, rather than $O(N)$. As such, it is able to find the ground truth clustering, correctly identifying all typos (unlike PClean [21], which achieves only 90% accuracy on this dataset) and reporting a log marginal likelihood thousands of nats higher than the SMC algorithm.

6 RELATED WORK AND DISCUSSION

Related work. RAVI builds on and generalizes recent work from both the Monte Carlo and variational inference literatures. For example, Salimans et al. [37] and Ranganath et al. [35] showed how auxiliary variables could be used to construct and optimize variational bounds for specific families of expressive variational approximations. Sobolev and Vetrov [40] presented tighter bounds in a more general setting. RAVI is a further generalization, in two directions: first, we show that these bounds arise from particular choices of meta-inference strategy, and can be tightened by improving meta-inference; and second, we extend the results to the Monte Carlo setting, enabling learned variational families to be used as IS, SMC, or MH proposals. We also provide general theorems about the variance of RAVI samplers and the bias of RAVI variational bounds, which can be applied to analyze both new and existing algorithms.

RAVI is also related to other compositional or unifying frameworks for thinking about broad classes of inference algorithms [23, 46, 39, 38, 24, 11, 41, 32, 45, 3, 42, 17, 18], some of which involve recursive constructions [29, 12, 14]. However, to our knowledge, RAVI's inference strategies are novel. For example, although RAVI and Nested IS (NIS) [29] are both approaches to inference with 'intractable proposals,' NIS *approximately samples* a proposal distribution with a *tractable* (unnormalized) density, whereas RAVI *approximates the density* of a proposal that *can* be simulated tractably, but whose marginal density (even unnormalized) is intractable. As another example, Domke and Sheldon [14]'s framework of *estimator-coupling pairs* constructs variational bounds and marginal likelihood estimators recursively, but unlike in RAVI, the posterior approximations cannot be used to formulate objectives for *amortized* VI or as components of Metropolis-Hastings proposals.

Finally, researchers have used meta-inference to construct bounds on KL divergences [10] and other information-theoretic quantities [36]. In Appendix D, we show how to apply such bounds in the general RAVI setting.

Outlook and Limitations. RAVI expands the design space for Monte Carlo and variational inference. It gives unifying correctness proofs for over a dozen methods from the literature, and novel theorems that characterize their behavior. Experiments show that RAVI helps to design algorithms that significantly improve accuracy over previously introduced Monte Carlo and variational inference methods. This may seem surprising, because as Table 2 shows, complicated proposals can induce high-dimensional meta-inference problems for RAVI to solve; why should these be any easier than the original inference problem? One explanation is that many proposals feature conditional independence structure not present in the model. For instance, whether or not the original model is an HMM, a proposal that runs an MCMC chain has Markov structure that makes it amenable to sequential Monte Carlo meta-inference.

We are excited about the potential of RAVI to help practitioners develop more effective inference algorithms, but some difficulties remain. For example, the gradient estimators we present (Algs. 3 and 4) have high variance for some strategies $\mathcal{S}$; in Appendix E, we give estimators that exploit the reparameterization trick, but they only help when the proposals in $\mathcal{S}$ can be reparameterized, which is not the case, e.g., for SMC. In these cases, RAVI can still be used to derive objectives for optimization, but practitioners will need other ways of reducing the variance of gradient estimates; many results from the literature [26, 43] should apply.

Another difficulty is that RAVI algorithms can be complex to implement. We are exploring an automated implementation based on probabilistic programming languages [11, 44]: if the posterior and meta-posterior approximations in a RAVI strategy $\mathcal{S}$ are given as probabilistic programs, we can provide Algs. 1-5 as higher-order functions, which automate the necessary densities, gradients, and MCMC acceptance probabilities. This could be viewed as a generalization of existing PPL support for *programmable inference* [23, 24, 11, 22].

Acknowledgements

The authors are grateful to Feras Saad, Tan Zhi-Xuan, Ben Sherman, Cameron Freer, George Matheos, Sam Witty, McCoy Becker, Jan-Willem van de Meent, Sam Stites, Eli Sennesh, Cathy Wong, and Nishad Gothoskar for useful conversations and feedback. We also would like to thank our anonymous referees for helpful feedback on earlier drafts of the paper. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1745302.

References

[1] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[2] Christophe Andrieu, Arnaud Doucet, Sinan Yıldırım, and Nicolas Chopin. On the utility of Metropolis-Hastings with asymmetric acceptance ratio. *arXiv preprint arXiv:1803.09527*, 2018.

[3] Christophe Andrieu, Anthony Lee, and Sam Livingstone. A general perspective on the Metropolis-Hastings kernel. *arXiv preprint arXiv:2012.14881*, 2020.

[4] Philip Bachman and Doina Precup. Training deep generative models: Variations on a theme. In *NIPS Approximate Inference Workshop*, 2015.

[5] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[6] Sourav Chatterjee, Persi Diaconis, et al. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018.

[7] Nicolas Chopin and Omiros Papaspiliopoulos. *An introduction to sequential Monte Carlo*. Springer, 2020.

[8] Nicolas Chopin, Pierre E Jacob, and Omiros Papaspiliopoulos. SMC^2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426, 2013.

[9] Chris Cremer, Quaid Morris, and David Duvenaud. Reinterpreting importance-weighted autoencoders. *arXiv preprint arXiv:1704.02916*, 2017.

[10] Marco Cusumano-Towner and Vikash K Mansinghka. AIDE: An algorithm for measuring the accuracy of probabilistic inference algorithms. *Advances in Neural Information Processing Systems*, 30, 2017.

[11] Marco F Cusumano-Towner, Feras A Saad, Alexander K Lew, and Vikash K Mansinghka. Gen: a general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 221–236, 2019.

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

[13] Petar M Djuric, Jayesh H Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Mónica F Bugallo, and Joaquin Miguez. Particle filtering. *IEEE signal processing magazine*, 20(5):19–38, 2003.

[14] Justin Domke and Daniel R Sheldon. Divide and couple: Using Monte Carlo variational objectives for posterior approximation. *Advances in Neural Information Processing Systems*, 32, 2019.

[15] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3): 197–208, 2000.

[16] Michael J Drinkwater, Quentin A Parker, Dominique Proust, Eric Slezak, and Hernán Quintana. The large scale distribution of galaxies in the shapley supercluster. *Publications of the Astronomical Society of Australia*, 21(1):89–96, 2004.

[17] Axel Finke. *On extended state-space constructions for Monte Carlo methods*. PhD thesis, University of Warwick, 2015.

[18] Axel Finke and Alexandre H Thiery. On importance-weighted autoencoders. *arXiv preprint arXiv:1907.10477*, 2019.

[19] Peter W Glynn and Donald L Iglehart. Importance sampling for stochastic simulations. *Management science*, 35(11):1367–1392, 1989.

[20] Tuan Anh Le, Atilim Gunes Baydin, and Frank Wood. Inference compilation and universal probabilistic programming. In *Artificial Intelligence and Statistics*, pages 1338–1348. PMLR, 2017.

[21] Alexander Lew, Monica Agrawal, David Sontag, and Vikash Mansinghka. PClean: Bayesian data cleaning at scale with domain-specific probabilistic programming. In *International Conference on Artificial Intelligence and Statistics*, pages 1927–1935. PMLR, 2021.

[22] Alexander K Lew, Marco F Cusumano-Towner, Benjamin Sherman, Michael Carbin, and Vikash K Mansinghka. Trace types and denotational semantics for

sound programmable inference in probabilistic languages. *Proceedings of the ACM on Programming Languages*, 4(POPL):1–32, 2019.

[23] Vikash Mansinghka, Daniel Selsam, and Yura Perov. Venture: a higher-order probabilistic programming platform with programmable inference. *arXiv preprint arXiv:1404.0099*, 2014.

[24] Vikash K Mansinghka, Ulrich Schaechtle, Shivam Handa, Alexey Radul, Yutian Chen, and Martin Rinard. Probabilistic programming with programmable inference. In *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 603–616, 2018.

[25] Medicare. Hospital compare, 2012.

[26] Andriy Mnih and Danilo Rezende. Variational inference for Monte Carlo objectives. In *International Conference on Machine Learning*, pages 2188–2196. PMLR, 2016.

[27] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[28] Saeid Naderiparizi, Adam Ścibior, Andreas Munk, Mehrdad Ghadiri, Atılım Güneş Baydin, Bradley Gram-Hansen, Christian Schroeder de Witt, Robert Zinkov, Philip H. S. Torr, Tom Rainforth, Yee Whye Teh, and Frank Wood. Amortized rejection sampling in universal probabilistic programming, 2019.

[29] Christian Naesseth, Fredrik Lindsten, and Thomas Schon. Nested sequential Monte Carlo methods. In *International Conference on Machine Learning*, pages 1292–1301. PMLR, 2015.

[30] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 968–977. PMLR, 2018.

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

[32] Kirill Neklyudov, Max Welling, Evgenii Egorov, and Dmitry Vetrov. Involutive MCMC: a unifying framework. In *International Conference on Machine Learning*, pages 7273–7282. PMLR, 2020.

[33] Michael A Newton and Adrian E Raftery. Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(1):3–26, 1994.

[34] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[35] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International conference on machine learning*, pages 324–333. PMLR, 2016.

[36] Feras A. Saad, Marco Cusumano-Towner, and Vikash K. Mansinghka. Estimators of entropy and information via inference in probabilistic models. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 5604–5621. PMLR, 2022.

[37] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226. PMLR, 2015.

[38] Adam Ścibior, Ohad Kammar, and Zoubin Ghahramani. Functional programming for modular Bayesian inference. *Proceedings of the ACM on Programming Languages*, 2(ICFP):1–29, 2018.

[39] Adam Ścibior, Ohad Kammar, Matthijs Vákár, Sam Staton, Hongseok Yang, Yufei Cai, Klaus Ostermann, Sean K Moss, Chris Heunen, and Zoubin Ghahramani. Denotational validation of higher-order Bayesian inference. *Proceedings of the ACM on Programming Languages*, 2018.

[40] Artem Sobolev and Dmitry P Vetrov. Importance weighted hierarchical variational inference. *Advances in Neural Information Processing Systems*, 32, 2019.

[41] Sam Stites, Heiko Zimmermann, Hao Wu, Eli Sennesh, et al. Learning proposals for probabilistic programs with inference combinators. *arXiv preprint arXiv:2103.00668*, 2021.

[42] Geir Storvik. On the flexibility of Metropolis–Hastings acceptance probabilities in auxiliary variable proposal generation. *Scandinavian Journal of Statistics*, 38(2): 342–358, 2011.

[43] George Tucker, Dieterich Lawson, Shixiang Gu, and Chris J Maddison. Doubly reparameterized gradient estimators for Monte Carlo objectives. *arXiv preprint arXiv:1810.04152*, 2018.

[44] Jan-Willem van de Meent, Brooks Paige, Hongseok Yang, and Frank Wood. An introduction to probabilistic programming. *arXiv preprint arXiv:1809.10756*, 2018.

[45] Heiko Zimmermann, Hao Wu, Babak Esmaeili, and Jan-Willem van de Meent. Nested variational inference. *Advances in Neural Information Processing Systems*, 34:20423–20435, 2021.

[46] Robert Zinkov and Chung-chieh Shan. Composing inference algorithms as program transformations. *arXiv preprint arXiv:1603.01882*, 2016.

SUPPLEMENTARY MATERIAL FOR "RECURSIVE MONTE CARLO AND VARIATIONAL INFERENCE WITH AUXILIARY VARIABLES"

This document and the accompanying code files contain supplementary material for the submission "Recursive Monte Carlo and Variational Inference with Auxiliary Variables." In particular, we provide:

1. In Section A, **proofs** of Theorems 1-4.
2. In Section B, **RAVI inference strategies for many existing algorithms**.
3. In Section C, a further discussion of the **absolute continuity requirements** for RAVI and how they can be relaxed.
4. In Section D, **other applications** of RAVI inference strategies, to parameterize rejection sampling and KL divergence estimation algorithms.

A OMITTED PROOFS.

Throughout this section, we use the notation introduced in Section 4: the random variable $\hat{Z}(\tilde{\pi}, \mathcal{S})$ is the weight returned by $\texttt{IMPORTANCE}(\tilde{\pi}, \mathcal{S})$, and $\check{Z}(\tilde{\pi}, \mathcal{S})$ is the reciprocal of the weight returned by $\texttt{HME}(\tilde{\pi}, x, \mathcal{S})$, for $x \sim \pi$.

A.1 PROOF OF THEOREM 1.

Theorem 1. *Let $\tilde{\pi}(x) = Z\pi(x)$ be an unnormalized target density, and $\mathcal{S}$ an inference strategy targeting $\pi(x)$. Then:*

- $\mathit{IMPORTANCE}(\mathcal{S}, \tilde{\pi})$ *generates* $(x, \hat{Z})$ *with* $x \sim \mathcal{S}.q$ *and* $\mathbb{E}[\hat{Z} \mid x] = Z\frac{\pi(x)}{\mathcal{S}.q(x)}$. *Furthermore, the unconditional expectation* $\mathbb{E}[\hat{Z}(\tilde{\pi}, \mathcal{S})] = Z$.
- $\mathbb{E}[\check{Z}(\tilde{\pi}, \mathcal{S})^{-1}] = \mathbb{E}_{x\sim\pi}[\mathit{HME}(\mathcal{S}, x, \tilde{\pi})] = Z^{-1}$.

Proof. The proof is by induction on the level of nesting present in the inference strategy.

First consider the case where $\mathcal{S}.q$ has a tractable marginal density. Then:

- $\texttt{IMPORTANCE}$ samples $x \sim \mathcal{S}.q$ on line 2, and computes $\hat{Z} = \frac{\tilde{\pi}(x)}{\mathcal{S}.q(x)} = Z\frac{\pi(x)}{\mathcal{S}.q(x)}$ exactly (lines 3 and 7). By the standard importance sampling argument, the unconditional expectation $\mathbb{E}[\hat{Z}(\tilde{\pi}, \mathcal{S})] = \mathbb{E}_{x\sim\mathcal{S}.q}[Z\frac{\pi(x)}{\mathcal{S}.q(x)}] = Z\mathbb{E}_{x\sim\pi}[1] = Z$. (This argument relies on the fact that, because $\mathcal{S}$ targets π, π is absolutely continuous with respect to $\mathcal{S}.q$.)
- $\texttt{HME}(\mathcal{S}, x, \tilde{\pi})$ returns exactly $\frac{\mathcal{S}.q(x)}{\tilde{\pi}(x)}$ (lines 2 and 5), and

$$\mathbb{E}_{x\sim\pi}\left[\frac{\mathcal{S}.q(x)}{\tilde{\pi}(x)}\right] = \int \pi(x)\frac{\mathcal{S}.q(x)}{Z\pi(x)}\mathrm{d}x = \frac{1}{Z}\int \mathcal{S}.q(x)\mathrm{d}x = \frac{1}{Z},$$

  where the last step follows because $\mathcal{S}.q$ is a normalized probability density, and $\mathcal{S}.q$ is absolutely continuous with respect to π.

Now consider the inductive step. Assume $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x)\mathrm{d}r$ and that for all x, the theorem holds for the inference strategy $\mathcal{S}.\mathcal{M}(x)$ and the unnormalized target distribution $\mathcal{S}.q(\cdot, x)$. In this case:

- On line 5, $\texttt{IMPORTANCE}$ generates $x \sim \mathcal{S}.q$ and $r \sim \mathcal{S}.q(r \mid x)$. In the call to $\texttt{HME}$, the unnormalized target distribution is $\mathcal{S}.q(\cdot, x)$, and so the normalizing constant is $\mathcal{S}.q(x)$ and the normalized target is $\mathcal{S}.q(r \mid x)$. By the inductive hypothesis, the call to $\texttt{HME}$ on line 6 returns an unbiased estimate of the normalizing constant's reciprocal, i.e. $\mathbb{E}[w \mid x] = \frac{1}{\mathcal{S}.q(x)}$. Since $\texttt{IMPORTANCE}$ returns $\hat{Z} = w\tilde{\pi}(x)$ on line 7, this implies that $\mathbb{E}[\hat{Z} \mid x] = \frac{\tilde{\pi}(x)}{\mathcal{S}.q(x)} = Z\frac{\pi(x)}{\mathcal{S}.q(x)}$. From this, the same standard importance sampling argument as above shows that the unconditional expectation $\mathbb{E}[\hat{Z}(\tilde{\pi}, \mathcal{S})] = Z$.
- On line 4, $\texttt{HME}$ calls $\texttt{IMPORTANCE}$ on the unnormalized target $\mathcal{S}.q(\cdot, x)$, and so by the inductive hypothesis, $\mathbb{E}[w] = \mathcal{S}.q(x)$ (the normalizing constant). On line 5, the returned weight has expectation $\mathbb{E}_{x\sim\pi}\left[\frac{w}{\tilde{\pi}(x)}\right] = \frac{1}{Z}\int \pi(x)\cdot\frac{\mathcal{S}.q(x)}{\pi(x)}\mathrm{d}x = \frac{1}{Z}$, where the last equality again follows because $\mathcal{S}.q$ is a normalized density, and $\mathcal{S}.q$ is absolutely continuous with respect to π.

A.2 PROOF OF THEOREM 2

Lemma A.1. *For an inference strategy $\mathcal{S}$ targeting $p(x \mid y)$, if $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x)dr$ has an intractable marginal density, then:*

$$\mathcal{L}(p, y, \mathcal{S}) = \mathbb{E}_{x \sim \mathcal{S}.q}[\log p(x, y) - \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x)]$$

and

$$\mathcal{U}(p, y, \mathcal{S}) = \mathbb{E}_{x \sim p(\cdot \mid y)}[\log p(x, y) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))]$$

Proof. For the first conclusion,

$$\begin{aligned}
\mathcal{L}(p, y, \mathcal{S}) &= \mathbb{E}[\log \hat{Z}(p(\cdot \mid y), \mathcal{S})] && (1)\\
&= \mathbb{E}\left[\log \frac{p(x, y)}{\check{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x))}\right] && (2)\\
&= \mathbb{E}_{x \sim \mathcal{S}.q}[\mathbb{E}[\log p(x, y) - \log \check{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x)) \mid x]] && (3)\\
&= \mathbb{E}_{x \sim \mathcal{S}.q}[\log p(x, y) - \mathbb{E}[\log \check{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x)) \mid x]] && (4)\\
&= \mathbb{E}_{x \sim \mathcal{S}.q}[\log p(x, y) - \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] && (5)
\end{aligned}$$

The same approach, but with $\mathbb{E}[\log \check{Z}]$, can be used to prove the other conclusion.

Theorem 2. *Given a model $p_\theta(x, y)$ and an inference strategy $\mathcal{S}_\theta$ targeting $p_\theta(x \mid y)$, Alg. 3 yields unbiased estimates of $\mathcal{L}(p, y, \mathcal{S})$ and of $\nabla_\theta \mathcal{L}(p, y, \mathcal{S})$. Furthermore, when $(x, y) \sim p_\theta$, Alg. 4 yields (i) $\hat{U}$ such that $\mathbb{E}[\hat{U} \mid y] = \mathcal{U}(p, y, \mathcal{S})$, (ii) $\widehat{\nabla_\theta}$ such that $\mathbb{E}[\widehat{\nabla_\theta}] = \nabla_\theta \mathbb{E}_{y \sim p_\theta}[\mathcal{U}(p, y, \mathcal{S})]$, and (iii) a value $\mathbf{g}$ such that for any function R that does not depend on θ, $\mathbb{E}[\mathbf{g} \cdot R(y)] = \nabla_\theta \mathbb{E}_{y \sim p_\theta}[R(y)]$ if $\nabla_\theta \mathbb{E}_{y \sim p_\theta}[R(y)]$ is defined.*

Proof. The proof is by induction on the level of nesting present in the inference strategy.

First consider inference strategies $\mathcal{S}$ with tractable proposals $\mathcal{S}.q(x)$. In this case ELBO∇ generates $x \sim \mathcal{S}.q$ and returns $\hat{L} = \log p(x, y) - \log \mathcal{S}.q(x)$ and $\widehat{\nabla_\theta} = \nabla_\theta(\log p(x, y) - \log \mathcal{S}.q(x)) + (\nabla_\theta \log \mathcal{S}.q(x))(\log p(x, y) - \log \mathcal{S}.q(x))$. Clearly, $\mathbb{E}_{x \sim \mathcal{S}.q}[\hat{L}] = \mathbb{E}[\log \hat{Z}(p(\cdot, y), \mathcal{S})] = \mathcal{L}(p, y, \mathcal{S})$. And by the log-derivative trick, $\mathbb{E}_{x \sim \mathcal{S}.q}[\widehat{\nabla_\theta}] = \mathbb{E}[\nabla_\theta(\log p(x, y) - \log \mathcal{S}.q(x))] = \mathbb{E}[\mathcal{L}(p, y, \mathcal{S})]$. When we apply EUBO$\nabla$ to $\mathcal{S}$ with $(x, y) \sim p$, it returns (1) $\hat{U} = \log p(x, y) - \log \mathcal{S}.q(x)$ (for which $\mathbb{E}[\hat{U} \mid y] = \mathcal{U}(p, y, \mathcal{S})$), (2) $\widehat{\nabla_\theta} = \nabla_\theta(\log p(x, y) - \log \mathcal{S}.q(x)) + \nabla_\theta \log p(x, y)(\log p(x, y) - \log \mathcal{S}.q(x))$ (for which, by the log-derivative trick, $\mathbb{E}[\widehat{\nabla_\theta}] = \nabla_\theta \mathbb{E}_{y \sim p}[\mathcal{U}(p, y, \mathcal{S})]$), and (3) $\mathbf{g} = \nabla_\theta \log p(x, y)$. This last return value satisfies the spec for $\mathbf{g}$ because if R does not depend on θ, then $\mathbb{E}_{(x,y) \sim p}[R(y) \cdot \nabla_\theta \log p(x, y)] = \int \int p(x, y) \cdot \frac{\nabla_\theta p(x,y)}{p(x,y)} \cdot R(y) \mathrm{d}x \mathrm{d}y = \nabla_\theta \int \int p(x, y) R(y) \mathrm{d}x \mathrm{d}y = \nabla_\theta \mathbb{E}[R(y)]$, as required.

Now consider the inductive step. Assume the theorem holds for the inference strategy $\mathcal{S}.\mathcal{M}(x)$ and joint distribution $\mathcal{S}.q(r, x)$.

We first consider ELBO∇. It generates $(r, x) \sim \mathcal{S}.q$ before calling EUBO∇, which by induction returns $(\hat{U}, \widehat{\nabla_\theta}, \mathbf{g})$ such that:

1. $\mathbb{E}[\hat{U} \mid x] = \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))$
2. $\mathbb{E}[\widehat{\nabla_\theta}] = \nabla_\theta \mathbb{E}_{x \sim \mathcal{S}.q}[\mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))]$
3. $\mathbb{E}[g \cdot R(x)] = \nabla_\theta \mathbb{E}_{x \sim \mathcal{S}.q}[R(x)]$ for all valid R.

ELBO∇ computes its first return value, $\hat{L}$, as $\log p(x, y) - \hat{U}$, so

$$\begin{aligned}
\mathbb{E}[\hat{L}] &= \mathbb{E}[\log p(x, y) - \hat{U}]\\
&= \mathbb{E}_{x \sim \mathcal{S}.q}[\mathbb{E}[\log p(x, y) - \hat{U} \mid x]]\\
&= \mathbb{E}_{x \sim \mathcal{S}.q}[\log p(x, y) - \mathbb{E}[\hat{U} \mid x]]\\
&= \mathbb{E}_{x \sim \mathcal{S}.q}[\log p(x, y) - \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))]\\
&= \mathcal{L}(p, y, \mathcal{S}),
\end{aligned}$$

where the fourth equality holds by the inductive hypothesis and the final one by Lemma 1. Its second return value is computed as $\widehat{\nabla_\theta}' = \nabla_\theta \log p(x,y) + \mathbf{g} \log p(x,y) - \widehat{\nabla_\theta}$, and so

$$\begin{aligned}
\mathbb{E}[\widehat{\nabla_\theta}'] &= \mathbb{E}\left[\nabla_\theta \log p(x,y) + \mathbf{g} \cdot \log p(x,y) - \widehat{\nabla_\theta}\right] \\
&= \mathbb{E}\left[\nabla_\theta \log p(x,y)\right] + \nabla_\theta \mathbb{E}_{x\sim\mathcal{S}.q}[\log \boxed{p}(x,y)] - \nabla_\theta \mathbb{E}_{x\sim\mathcal{S}.q}[\mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\
&= \nabla_\theta \mathbb{E}\left[\log p(x,y)\right] - \nabla_\theta \mathbb{E}_{x\sim\mathcal{S}.q}[\mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\
&= \nabla_\theta \mathbb{E}_{x\sim\mathcal{S}.q}[\log p(x,y) - \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\
&= \nabla_\theta \mathcal{L}(p, y, \mathcal{S}),
\end{aligned}$$

where $\boxed{p}(x,y)$ denotes the distribution $p(x,y)$ but without a dependence on θ, for the purposes of differentiation with respect to θ. The second equality holds by the inductive hypothesis about $\mathbf{g}$ (with $R(x) = \log \boxed{p}(x,y)$) and about $\widehat{\nabla_\theta}$, and the third uses the log-derivative trick. The final equation is due to Lemma 1.

We now turn to EUBO∇. By induction, the call to ELBO∇ satisfies the theorem, and so:

1. $\mathbb{E}[\hat{L}] = \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))$
2. $\mathbb{E}[\widehat{\nabla_\theta}] = \nabla_\theta \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))$

We treat each of the return values, $(\hat{U}, \widehat{\nabla_\theta}, \mathbf{g})$, in sequence. We view them as random variables, accounting for stochasticity in the algorithm as well as the inputs (x,y), which are assumed in the theorem's statement to be jointly distributed according to p.

First, $\hat{U}$ is computed as $\log p(x,y) - \hat{L}$, and so

$$\begin{aligned}
\mathbb{E}[\hat{U}|y] &= \mathbb{E}_{x\sim p(\cdot|y)}[\mathbb{E}[\log p(x,y) - \hat{L}|x,y]] \\
&= \mathbb{E}_{x\sim p(\cdot|y)}[\log p(x,y) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\
&= \mathcal{U}(p, y, \mathcal{S}).
\end{aligned}$$

Next, $\mathbb{E}[\widehat{\nabla_\theta}']$:

$$\begin{aligned}
\mathbb{E}[\widehat{\nabla_\theta}'] &= \mathbb{E}_{x,y\sim p}\left[\mathbb{E}\left[\nabla_\theta \log p(x,y) + (\nabla_\theta \log p(x,y)) \cdot \hat{U} - \widehat{\nabla_\theta}|x,y\right]\right] \\
&= \mathbb{E}_{x,y\sim p}\left[\nabla_\theta \log p(x,y) + (\nabla_\theta \log p(x,y)) \cdot \mathbb{E}\left[\hat{U}|x,y\right] - \mathbb{E}\left[\widehat{\nabla_\theta}|x,y\right]\right] \\
&= \mathbb{E}_{x,y\sim p}\left[\nabla_\theta \log p(x,y) + (\nabla_\theta \log p(x,y)) \cdot \mathbb{E}\left[\log p(x,y) - \hat{L} \mid x,y\right] - \nabla_\theta \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))\right] \\
&= \mathbb{E}_{x,y\sim p}\left[\nabla_\theta \log p(x,y) - \nabla_\theta \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x)) + (\nabla_\theta \log p(x,y)) \cdot (\log p(x,y) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))\right] \\
&= \nabla_\theta \mathbb{E}_{x,y\sim p}\left[\log p(x,y) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))\right] \\
&= \nabla_\theta \mathbb{E}_{y\sim p}[\mathbb{E}_{x\sim p(\cdot|y)}[\log p(x,y) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))]] \\
&= \nabla_\theta \mathbb{E}_{y\sim p}[\mathcal{U}(p, y, \mathcal{S})].
\end{aligned}$$

Finally, we consider $\mathbb{E}_{y\sim p}[\mathbb{E}[\mathbf{g} \cdot R(y)|y]]$ (and recall that $R(y)$ is not to be treated as a function of θ):

$$\begin{aligned}
\mathbb{E}_{y\sim p}[\mathbb{E}[\mathbf{g} \cdot R(y)|y]] &= \mathbb{E}_{y\sim p}\left[\mathbb{E}[(\nabla_\theta \log p(x,y)) \cdot R(y)|y]\right] \\
&= \mathbb{E}_{x,y\sim p}\left[(\nabla_\theta \log p(x,y)) \cdot R(y)\right] \\
&= \nabla_\theta \mathbb{E}_{x,y\sim p}\left[R(y)\right] \\
&= \nabla_\theta \mathbb{E}_{y\sim p}[R(y)].
\end{aligned}$$

A.3 PROOF OF THEOREM 3

Theorem 3. *Consider an unnormalized target distribution $\tilde{\pi}(x) = Z\pi(x)$ and an inference strategy $\mathcal{S}$ targeting $\pi(x)$. Then the relative variances of the estimators $\hat{Z}(\tilde{\pi}, \mathcal{S})$ and $\check{Z}(\tilde{\pi}, \mathcal{S})$ are given by the following recursive equations:*

$$\begin{aligned}
&\mathrm{Var}_{\hat{Z}}(\pi, \mathcal{S}) = \chi^2(\pi || \mathcal{S}.q) + \\
&\quad \mathbb{E}_{x \sim \mathcal{S}.q}\left[\left(\frac{\pi(x)^2}{\mathcal{S}.q(x)^2}\right) \cdot \mathrm{Var}_{\check{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))\right] \\
&\mathrm{Var}_{\check{Z}}(\pi, \mathcal{S}) = \chi^2(\mathcal{S}.q || \pi) + \\
&\quad \mathbb{E}_{x \sim \pi}\left[\left(\frac{\mathcal{S}.q(x)^2}{\pi(x)^2}\right) \cdot \mathrm{Var}_{\hat{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))\right]
\end{aligned}$$

When $\mathcal{S}.q$ is tractable, the second term of each sum is 0.

Proof. The proof is by induction on the level of nesting present in the inference strategy $\mathcal{S}$.

First suppose $\mathcal{S}.q$ has a tractable marginal density. Then:

- $\hat{Z}(\pi, \mathcal{S})$ is the normalized importance weight $\frac{\pi(x)}{\mathcal{S}.q(x)}$, with $x \sim \mathcal{S}.q$. So the relative variance is:

$$\mathrm{Var}_{\hat{Z}}(\pi, \mathcal{S}) = \mathrm{Var}\left(\hat{Z}(\pi, \mathcal{S})\right) = \mathbb{E}_{x \sim \mathcal{S}.q}\left[\frac{\pi(x)^2}{\mathcal{S}.q(x)^2}\right] - \mathbb{E}_{x \sim \mathcal{S}.q}\left[\frac{\pi(x)}{\mathcal{S}.q(x)}\right]^2 = \mathbb{E}_{x \sim \mathcal{S}.q}\left[\frac{\pi(x)^2}{\mathcal{S}.q(x)^2} - 1\right] = \chi^2(\pi || \mathcal{S}.q),$$

  where the third equality holds because π is a normalized density and π is absolutely continuous with respect to $\mathcal{S}.q$.

- $\check{Z}(\pi, \mathcal{S})$ is the weight $\frac{\pi(x)}{\mathcal{S}.q(x)}$, with $x \sim \pi$. Then the relative variance

$$\mathrm{Var}_{\check{Z}}(\pi, \mathcal{S}) = \mathrm{Var}\left(\check{Z}(\pi, \mathcal{S})^{-1}\right) = \mathbb{E}_{x \sim \pi}\left[\frac{\mathcal{S}.q(x)^2}{\pi(x)^2}\right] - \mathbb{E}_{x \sim \pi}\left[\frac{\mathcal{S}.q(x)}{\pi(x)}\right]^2 = \mathbb{E}_{x \sim \pi}\left[\frac{\mathcal{S}.q(x)^2}{\pi(x)^2} - 1\right] = \chi^2(\mathcal{S}.q || \pi),$$

  where the third equality holds because $\mathcal{S}.q$ is a normalized density and is absolutely continuous with respect to π.

Now consider the inductive step. Assume that for all x, the theorem holds of the strategy $\mathcal{S}.\mathcal{M}(x)$ targeting $\mathcal{S}.q(\cdot \mid x)$. $\mathrm{Var}_{\hat{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))$, for all x. Then:

- The `IMPORTANCE`$(\pi, \mathcal{S})$ algorithm generates $x \sim \mathcal{S}.q$. It then calls `HME` (with $r \sim \mathcal{S}.q(\cdot \mid x)$) to obtain $w =$

$\check{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x))^{-1}$, and returns $\hat{Z} = w\pi(x)$. The variance of $\hat{Z}$ is then:

$$
\begin{aligned}
\mathrm{Var}_{\hat{Z}}(\pi, \mathcal{S}) &= \mathrm{Var}\left(\frac{\pi(x)}{\check{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x))}\right) \\
&= \mathbb{E}\left[\left(\frac{\pi(x)}{\check{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x))}\right)^2 - 1\right] && (\mathbb{E}[\hat{Z}(\pi, \mathcal{S})]^2 = Z^2 = 1) \\
&= \mathbb{E}\left[\left(\frac{\pi(x)}{\mathcal{S}.q(x)} \cdot \frac{\mathcal{S}.q(x)}{\check{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x))}\right)^2 - 1\right] && \text{(divide and multiply by } \mathcal{S}.q(x)) \\
&= \mathbb{E}\left[\left(\frac{\pi(x)}{\mathcal{S}.q(x)} \cdot \frac{1}{\check{Z}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))}\right)^2 - 1\right] && (\mathcal{S}.q(x) \text{ is the normalizing constant of } \mathcal{S}.q(\cdot, x) \\
&= \mathbb{E}\left[\left(\frac{\pi(x)}{\mathcal{S}.q(x)}\right)^2 \left(\mathbb{E}\left[\check{Z}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))^{-2} \middle| x\right]\right) - 1\right] \\
&= \mathbb{E}\left[\left(\frac{\pi(x)}{\mathcal{S}.q(x)}\right)^2 (\mathrm{Var}_{\check{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x)) + 1) - 1\right] && \text{(definition of } \mathrm{Var}_{\check{Z}}(\cdot, \cdot)) \\
&= \mathbb{E}\left[\left(\frac{\pi(x)}{\mathcal{S}.q(x)}\right)^2 (\mathrm{Var}_{\check{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))) + \left(\frac{\pi(x)}{\mathcal{S}.q(x)}\right)^2 - 1\right] && \text{(distributing product over sum)} \\
&= \mathbb{E}\left[\left(\frac{\pi(x)}{\mathcal{S}.q(x)}\right)^2 (\mathrm{Var}_{\check{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x)))\right] + \chi^2(\pi || \mathcal{S}.q).
\end{aligned}
$$

- The argument for $\check{Z}$ is largely the same:

$$
\begin{aligned}
\mathrm{Var}_{\check{Z}}(\pi, \mathcal{S}) &= \mathrm{Var}\left(\frac{\hat{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x))}{\pi(x)}\right) \\
&= \mathbb{E}\left[\left(\frac{\check{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x))}{\pi(x)}\right)^2 - 1\right] && (\mathbb{E}[\check{Z}(\pi, \mathcal{S})^{-1}]^2 = Z^{-2} = 1) \\
&= \mathbb{E}\left[\left(\frac{\mathcal{S}.q(x)}{\pi(x)} \cdot \frac{\hat{Z}(\mathcal{S}.q(\cdot, x), \mathcal{S}.\mathcal{M}(x))}{\mathcal{S}.q(x)}\right)^2 - 1\right] && \text{(divide and multiply by } \mathcal{S}.q(x)) \\
&= \mathbb{E}\left[\left(\frac{\mathcal{S}.q(x)}{\pi(x)} \cdot \hat{Z}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))\right)^2 - 1\right] && (\mathcal{S}.q(x) \text{ is the normalizing constant of } \mathcal{S}.q(\cdot, x \\
&= \mathbb{E}\left[\left(\frac{\mathcal{S}.q(x)}{\pi(x)}\right)^2 \left(\mathbb{E}\left[\hat{Z}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))^2 \middle| x\right]\right) - 1\right] \\
&= \mathbb{E}\left[\left(\frac{\mathcal{S}.q(x)}{\pi(x)}\right)^2 \left(\mathrm{Var}_{\hat{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x)) + 1\right) - 1\right] && \text{(definition of } \mathrm{Var}_{\hat{Z}}(\cdot, \cdot)) \\
&= \mathbb{E}\left[\left(\frac{\mathcal{S}.q(x)}{\pi(x)}\right)^2 \left(\mathrm{Var}_{\hat{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))\right) + \left(\frac{\mathcal{S}.q(x)}{\pi(x)}\right)^2 - 1\right] && \text{(distributing product over sum)} \\
&= \mathbb{E}\left[\left(\frac{\mathcal{S}.q(x)}{\pi(x)}\right)^2 \left(\mathrm{Var}_{\hat{Z}}(\mathcal{S}.q(\cdot \mid x), \mathcal{S}.\mathcal{M}(x))\right)\right] + \chi^2(\mathcal{S}.q || \pi).
\end{aligned}
$$

A.4 PROOF OF THEOREM 4.

Theorem 4. *Consider a joint distribution $p(x, y)$ and an inference strategy $\mathcal{S}$ targeting $p(x \mid y)$. Then the following equations give the bias of $\hat{\mathcal{L}}$ and $\hat{\mathcal{U}}$ as estimators of $\log p(y)$:*

$$\begin{aligned}\text{Bias}_{\mathcal{L}}(p, y, \mathcal{S}) = &- \text{KL}(\mathcal{S}.q || p(\cdot \mid y)) \\ &- \mathbb{E}_{x \sim \mathcal{S}.q}[\text{Bias}_{\mathcal{U}}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\ \text{Bias}_{\mathcal{U}}(p, y, \mathcal{S}) = &\text{KL}(p(\cdot \mid y) || \mathcal{S}.q) \\ &- \mathbb{E}_{x \sim p(\cdot \mid y)}[\text{Bias}_{\mathcal{L}}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))]\end{aligned}$$

where the second term in each equation is 0 when $\mathcal{S}.q$ has a tractable marginal density.

Proof.

In the base case, where $\mathcal{S}.q$ has a tractable marginal density, the theorem states that $\log p(y) - \mathcal{L}(p, y, \mathcal{S}) = KL(\mathcal{S}.q || p(\cdot \mid y))$, the familiar relationship between the standard ELBO and the KL divergence. The $\mathcal{U}$ case is similar:

$$\begin{aligned}\text{Bias}_{\mathcal{U}}(p, y, \mathcal{S}) &= \mathbb{E}_{x \sim p(\cdot \mid y)}[\log p(x, y) - \log \mathcal{S}.q(x)] - \log p(y) \\ &= \log p(y) + \mathbb{E}_{x \sim p(\cdot \mid y)}[\log p(x \mid y) - \log \mathcal{S}.q(x)] - \log p(y) \\ &= KL(p(\cdot \mid y) || \mathcal{S}.q).\end{aligned}$$

Now consider the inductive step, in which $\mathcal{S}.q$ does not have a tractable marginal density. We assume the theorem holds for $\mathcal{S}.q$ and $\mathcal{S}.\mathcal{M}(x)$. Then:

$$\begin{aligned}\text{Bias}_{\mathcal{L}}(p, y, \mathcal{S}) &= \mathcal{L}(p, y, \mathcal{S}) - \log p(y) \\ &= \mathbb{E}_{x \sim \mathcal{S}.q}[\log p(x, y) - \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] - \log p(y) \\ &= \log p(y) + \mathbb{E}_{\sim \mathcal{S}.q}[\log p(x \mid y) - \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\ &= \mathbb{E}_{x \sim \mathcal{S}.q}[\log p(x \mid y) - \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\ &= \mathbb{E}_{x \sim \mathcal{S}.q}[\log p(x \mid y) - \log \mathcal{S}.q(x) + \log \mathcal{S}.q(x) - \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\ &= -KL(\mathcal{S}.q || p(\cdot \mid y)) + \mathbb{E}_{x \sim \mathcal{S}.q}[\log \mathcal{S}.q(x) - \mathcal{U}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\ &= -KL(\mathcal{S}.q || p(\cdot \mid y)) - \mathbb{E}_{x \sim \mathcal{S}.q}[\text{Bias}_{\mathcal{U}}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))].\end{aligned}$$

Nearly the same proof applies for $\mathcal{U}$, flipping the necessary signs:

$$\begin{aligned}\text{Bias}_{\mathcal{U}}(p, y, \mathcal{S}) &= \mathcal{U}(p, y, \mathcal{S}) - \log p(y) \\ &= \mathbb{E}_{x \sim p(\cdot \mid y)}[\log p(x, y) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] - \log p(y) \\ &= \log p(y) + \mathbb{E}_{\sim p(\cdot \mid y)}[\log p(x \mid y) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\ &= \mathbb{E}_{x \sim p(\cdot \mid y)}[\log p(x \mid y) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\ &= \mathbb{E}_{x \sim p(\cdot \mid y)}[\log p(x \mid y) - \log \mathcal{S}.q(x) + \log \mathcal{S}.q(x) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\ &= KL(p(\cdot \mid y) || \mathcal{S}.q) + \mathbb{E}_{x \sim p(x \mid y)}[\log \mathcal{S}.q(x) - \mathcal{L}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))] \\ &= KL(p(\cdot \mid y) || \mathcal{S}.q) - \mathbb{E}_{x \sim p(\cdot \mid y)}[\text{Bias}_{\mathcal{L}}(\mathcal{S}.q, x, \mathcal{S}.\mathcal{M}(x))].\end{aligned}$$

A.5 STATIONARITY OF MCMC ALGORITHM

In Section 3, we mention that RAVI can be used to run Metropolis-Hastings kernels with proposals that have intractable densities. Here, we present and justify the algorithm.

Let $\tilde{\pi}(x) = \int \tilde{\pi}(r, x) \mathrm{d}r = Z \int \pi(r, x) \mathrm{d}r$ be a possibly unnormalized target density, and let $q(x'; x) = \int q(s, x'; x) \mathrm{d}s$ be a proposal kernel mapping previous state x to new state x'. We note that (1) both $\tilde{\pi}$ and q have intractable marginal densities, and (2) the target marginal $\tilde{\pi}(x)$ itself may be unnormalized. As is typical in pseudomarginal MCMC, even this unnormalized target density cannot be evaluated pointwise, due to the additional nuisance variables r.

Now suppose we have a family of inference strategies $\mathcal{S}(x)$ targeting $\pi(r \mid x)$, and a family of inference strategies $\mathcal{M}(x, x')$ targeting $q(s \mid x'; x)$. Let x be a starting position for our Markov chain. We can run Algorithm 1 on $\mathcal{S}$, targeting $\pi(r \mid x)$, to obtain an initial estimate $\hat{Z}_x$ of the unnormalized marginal density $\tilde{\pi}(x)$. Then Algorithm 5 defines a stationary MCMC kernel for the target distribution $\pi(x)$, starting at input point x:

Algorithm 5: RAVI Metropolis-Hastings
Input: model $\tilde{\pi}(x) = Z \int \pi(r, x) \mathrm{d}r$
Input: proposal $q(x'; x) = \int q(s, x'; x) \mathrm{d}s$
Input: family $\mathcal{S}(x)$ of inference strategies targeting $\pi(r \mid x)$
Input: family $\mathcal{M}(x, x')$ of inference strategies targeting $q(s \mid x'; x)$
Input: initial position x and estimate $\hat{Z}_x$ of $\tilde{\pi}(x)$
Output: next position x' and estimate $\hat{Z}_{x'}$ of $\tilde{\pi}(x')$

1 $(s, x') \sim q(s, x'; x)$
2 $w_{x'} \leftarrow \texttt{HME}(q(\cdot, x'; x), s, \mathcal{M}(x, x'))$
3 $(_, w_x) \leftarrow \texttt{IMPORTANCE}(q(\cdot \mid x; x'), \mathcal{M}(x', x))$
4 $(_, \hat{Z}_{x'}) \leftarrow \texttt{IMPORTANCE}(\pi(\cdot \mid x'), \mathcal{S}(x'))$
5 $u \sim \text{Uniform}(0, 1)$
6 **if** $u < \mathit{min}(1, \frac{\hat{Z}_{x'}}{\hat{Z}_x} w_{x'} w_x)$ **then**
7 **return** $(x', \hat{Z}_{x'})$
8 **else**
9 **return** $(x, \hat{Z}_x)$

When q's marginal density is known exactly, the above algorithm recovers variants of Particle-Marginal MH [1], except instead of using SMC to marginalize r, any RAVI algorithm can be applied. When q's marginal density is unavailable, however, the algorithm instead becomes a pseudo-marginal *ratio* algorithm [2], because not just p but also q is estimated unbiasedly. In general, it is *not* valid to use arbitrary unbiased estimates of p *and* q, or even of $\alpha = \frac{p(x')q(x;x')}{p(x)q(x';x)}$, within an MH algorithm. However, the added structure of the RAVI strategy ensures that the above procedure is sound.

To see why our MCMC kernel is stationary, we consider an extended target distribution. First, some notation. For an inference strategy $\mathcal{S}$ targeting $\pi(x)$, write $v_{\mathcal{S}}$ for the complete set of auxiliary variables in the strategy: if $\mathcal{S}.q$ has a tractable marginal density, then $v_{\mathcal{S}} = \emptyset$, and otherwise, if $\mathcal{S}.q(x) = \int \mathcal{S}.q(r, x) \mathrm{d}r$, then $v_{\mathcal{S}}$ is defined recursively as $\{r\} \cup v_{\mathcal{S}.\mathcal{M}}$. Calling $\texttt{IMPORTANCE}$ on $\mathcal{S}$ yields a joint distribution over these auxiliary variables and x, which we denote as $p^{\mathcal{S}}_{\texttt{IMP}}(v_{\mathcal{S}}, x)$. Calling $\texttt{HME}$ on $\mathcal{S}$ and a particular sample x yields a distribution over just $v_{\mathcal{S}}$, which we denote $p^{\mathcal{S}}_{\texttt{HME}}(v_{\mathcal{S}}; x)$. When $x \sim \pi$ and $v_{\mathcal{S}} \sim p^{\mathcal{S}}_{\texttt{HME}}(v_{\mathcal{S}}; x)$, the ratio $\frac{p^{\mathcal{S}}_{\texttt{IMP}}(v_{\mathcal{S}}, x)}{\tilde{\pi}(x) p^{\mathcal{S}}_{\texttt{HME}}(v_{\mathcal{S}}; x)}$ is the weight $\check{Z}(\tilde{\pi}, \mathcal{S})^{-1}$ returned by $\texttt{HME}$, and similarly, when $(v_{\mathcal{S}}, x) \sim p^{\mathcal{S}}_{\texttt{IMP}}$, the ratio $\frac{\tilde{\pi}(x) p^{\mathcal{S}}_{\texttt{HME}}(v_{\mathcal{S}}; x)}{p^{\mathcal{S}}_{\texttt{IMP}}(v_{\mathcal{S}}, x)}$ is the weight $\hat{Z}(\tilde{\pi}, \mathcal{S})$ returned by $\texttt{IMPORTANCE}$.

Using this notation, we can extend the target distribution $\tilde{\pi}(x)$ to one over $(x, s, x', s', v_{\mathcal{S}(x)}, v_{\mathcal{M}(x,x')}, v_{\mathcal{M}(x',x)})$ that admits $\tilde{\pi}(x)$ as a marginal:

$$\tilde{\pi}(r, x, s, x', s', v_{\mathcal{S}(x)}, v_{\mathcal{M}(x,x')}, v_{\mathcal{M}(x',x)}) = \tilde{\pi}(r, x) \cdot p^{\mathcal{S}(x)}_{\texttt{HME}}(v_{\mathcal{S}(x)}; r) \cdot q(s, x'; x) \cdot p^{\mathcal{M}(x,x')}_{\texttt{HME}}(v_{\mathcal{M}(x,x')}; s) \cdot p^{\mathcal{M}(x',x)}_{\texttt{IMP}}(v_{\mathcal{M}(x',x)}, s')$$

Our algorithm can be understood as sequencing two stationary kernels for this extended target. The first (implemented by lines 1-3) is a blocked Gibbs update on the variables $(s, x', s', v_{\mathcal{M}(x,x')}, v_{\mathcal{M}(x',x)})$, conditioned on everything else. Lines 1-3 sample exactly from the conditional distribution of these variables. The second is a Metropolis-Hastings proposal that simultaneously: (i) swaps x with x' (the 'main' proposed update), (ii) swaps $(s, v_{\mathcal{M}(x,x')})$ with $(s', v_{\mathcal{M}(x',x)})$, and (iii) proposes an update to r and to $v_{\mathcal{S}(x)}$ from $p^{\mathcal{S}(x')}_{\texttt{IMP}}$. The usual Metropolis-Hastings acceptance probability for this kernel, computed on the extended state space, is precisely the formula in Line 6.

One consequence of this justification is that the *same* family $\mathcal{S}$ of inference strategies for π must be used at each iteration. The family $\mathcal{M}$ can be freely switched out (as can q), however, to develop a cycle of kernels that use different proposal distributions.

B FURTHER EXAMPLES

This appendix lists examples of popular Monte Carlo and variational inference algorithms, and explains how they can be viewed as inference strategies. In addition, some of these algorithms can be viewed as *inference strategy combinators*, because they feature user-chosen proposal distributions or variational families that can themselves be instantiated with inference strategies.[5]

[5] This 'combinator' viewpoint evokes earlier work by [39] and [41]. For example, [41] introduce combinators for creating properly weighted samplers compositionally, with parameters that can be optimized using standard or nested variational objectives. Some of their combinators have equivalents in this section, e.g. their `propose` combinator is similar to the construction we present for Nested Importance Sampling in Section B.6. However: (1) the fundamental compositional operation in RAVI, of combining a posterior approximation with a meta-posterior approximation, cannot be achieved using their combinators; (2) as such, some of the algorithms that RAVI covers cannot be constructed using their combinators; and (3) their combinators produce properly weighted samplers, which contain 'less information' than inference strategies: an inference strategy can be used, e.g., as a proposal distribution in Metropolis-Hastings, whereas properly weighted samplers cannot in general be used this way.

B.1 N-PARTICLE IMPORTANCE SAMPLING

RAVI Inference Strategy: N-particle Importance Sampling
Posterior Approx. `sir(`$\tilde{\pi}, q, N$`).q()`
Target of inference : latent variable x
Auxiliary variables : particles $x_{1:N}$, chosen particle index j
for $i \in 1, \ldots, N$ **do**
  $x_i \sim q$
  $w_i \leftarrow \frac{\tilde{\pi}(x_i)}{q(x_i)}$
$j \sim \text{Discrete}(w_{1:N})$
return x_j

Meta-Posterior Approx. `sir(`$\tilde{\pi}, q, N$`).M(`x`).q()`
Target of inference : particles $x_{1:N}$, chosen particle index j
Auxiliary variables : None
$j \sim \text{Uniform}(1, N)$
$x_j \leftarrow x$
for $i \in 1, \ldots, j-1, j+1, \ldots, N$ **do**
  $x_i \sim q$
return $(x_{1:N}, j)$

RAVI Inference Strategy: N-particle IS with RAVI strategy $\mathcal{S}$
Posterior Approx. `ravi-sir(`$\tilde{\pi}, \mathcal{S}, N$`).q()`
Target of inference : latent variable x
Auxiliary variables : particles $x_{1:N}$, aux. proposal variables $v_{\mathcal{S}}^{1:N}$, chosen particle index j
for $i \in 1, \ldots, N$ **do**
  $x_i, w_i \sim$ `IMPORTANCE`$(\tilde{\pi}, \mathcal{S})$ w. aux. vars $v_{\mathcal{S}}^{i}$
$j \sim \text{Discrete}(w_{1:N})$
return x_j

Meta-Posterior Approx. `ravi-sir(`$\tilde{\pi}, \mathcal{S}, N$`).M(`$x$`).q()`
Target of inference : particles $x_{1:N}$, aux. proposal variables $v_{\mathcal{S}}^{1:N}$, chosen particle index j
Auxiliary variables : None
$j \sim \text{Uniform}(1, N)$
$x_j \leftarrow x$
$_ \sim$ `HME`$(\tilde{\pi}, x_j, \mathcal{S})$ w. aux. vars $v_{\mathcal{S}}^{j}$
for $i \in 1, \ldots, j-1, j+1, \ldots, N$ **do**
  $_, x_i \sim q$ w. aux. vars $v_{\mathcal{S}}^{i}$
return $(v_{\mathcal{S}}^{1:N}, x_{1:N}, j)$

Consider the N-particle importance sampling estimator

$$\hat{Z} = \frac{1}{N} \sum_{i=1}^{N} \frac{\tilde{\pi}(x_i)}{q(x_i)}, \text{ for } x_i \sim q.$$

The same estimator can be recovered as a *one-particle* `IMPORTANCE` estimate, by applying Alg. 1 to the `sir` inference strategy.

The proposal $\mathcal{S}.q$ generates N particles $x_{1:N}$, and selects an index j from a discrete distribution on $1, \ldots, N$, with weights proportional to $w_i = \tilde{\pi}(x_i)/q(x_i)$. The meta-proposal is responsible for inferring j and the complete set of particles $x_{1:M}$, given the chosen particle x. It uses the conditional SIR algorithm [1] to do so, proposing j uniformly in $\{1, \ldots, N\}$, and generating values for the un-chosen particles x_{-j} from q.

This is a suboptimal choice of $\mathcal{S}.\mathcal{M}(x).q$; lower-variance estimates $\hat{Z}$ can be obtained by improving meta-inference, either by incorporating problem-specific domain knowledge or via learning. However, in many cases, improved meta-inference may not be worth the computation required; it remains to be seen whether techniques such as amortized learning can be applied to deliver accuracy gains at low computational cost.

Instantiating the proposal q as its own inference strategy. The above assumes that q has a tractable marginal density. When it doesn't, the inner importance sampling loop can use a RAVI inference strategy $\mathcal{S}$ instead of a tractable proposal q. This modification is presented in the higher-order inference strategy `ravi-sir`. One way to think about this construction is as a way to improve any existing inference strategy $\mathcal{S}$ by 'adding replicates.' The resulting estimator of Z is the mean of N independent $\hat{Z}$ estimates from the original inference strategy.

B.2 IMPORTANCE-WEIGHTED AUTOENCODERS

The importance-weighted auto-encoder arises by considering the same inference strategy as in Section B.1, but as a variational inference procedure (Alg. 3) rather than a Monte Carlo procedure.

Because `sir`$(\tilde{\pi}, q, N).q$ of this inference strategy corresponds to N-particle sampling importance-resampling (SIR), it has been argued that IWAE is in fact 'vanilla' variational inference, but with a variational family that uses SIR to more closely approximate the posterior [4]. However, [9] show that deriving the ELBO for that variational family gives rise to a different objective, and that IWAE gives a looser lower bound on $\log Z$ than this idealized (but generally intractable) objective.

In the RAVI framework, these two objectives arise from different inference strategies, which share the same $\mathcal{S}.q$ (SIR in both cases), but use different meta-inference $\mathcal{S}.\mathcal{M}$. IWAE uses the simple conditional SIR meta-inference introduced in Section B.1, whereas [9]'s idealized objective can be derived by using the optimal choice of $\mathcal{S}.\mathcal{M}(x).q(j, x_{1:N})$—the exact posterior of the SIR procedure. The looser bound obtained by IWAE can be seen as a result of its $\mathcal{S}.\mathcal{M}$ performing poorer meta-inference: inference about the auxiliary variables of the SIR inference algorithm used in $\mathcal{S}.q$.

B.3 N-PARTICLE SEQUENTIAL MONTE CARLO

RAVI Inference Strategy: N-particle SMC w. RAVI strategies

Posterior Approx. `smc` $(\tilde{\pi}_{1:T}, \mathcal{S}, K_{2:T}, L_{2:T}, N)$ `.q()`

Target of inference : latent variable x targeting $\tilde{\pi}_T$

Auxiliary variables : particles $x_{1:N}^{1:T}$, aux. proposal variables $v_{\mathcal{S}}^{1:N}$, aux. K vars $v_{K_{2:T}}^{1:N}$, aux. L vars $v_{L_{2:T}}^{1:N}$, ancestor variables $a_{1:N}^{1:T-1}$, final chosen particle index j

for $i \in 1, \ldots, N$ **do**
  $x_i^1, w_i^1 \sim$ `IMPORTANCE`$(\tilde{\pi}_1, \mathcal{S})$ w. aux. vars $v_{\mathcal{S}}^i$
for $t \in 2, \ldots, T$ **do**
  for $i \in 1, \ldots, N$ **do**
    $a_i^{t-1} \sim \text{Discrete}(w_{1:N}^{t-1})$
    $x_i^t, \hat{w} \sim$ `IMPORTANCE`$(\tilde{\pi}_t, K_t(x_{a_i^{t-1}}^{t-1}))$ w. aux. vars $v_{K_t}^i$
    $\check{w} \sim$ `HME`$(\tilde{\pi}_{t-1}, x_{a_i^{t-1}}^{t-1}, L_t(x_i^t))$ w. aux. vars $v_{L_t}^i$
    $w_i^t \leftarrow \hat{w} \cdot \check{w}$
$j \sim \text{Discrete}(w_{1:N}^T)$
return x_j^T

Meta-Posterior Approx. `smc` $(\tilde{\pi}_{1:T}, \mathcal{S}, K_{2:T}, L_{2:T}, N)$ `.M(`x`).q()`

Target of inference : particles $x_{1:N}^{1:T}$, aux. proposal variables $v_{\mathcal{S}}^{1:N}$, aux. K vars $v_{K_{2:T}}^{1:N}$, aux. L vars $v_{L_{2:T}}^{1:N}$, ancestor variables $a_{1:N}^{1:T-1}$, final chosen particle index j

Auxiliary variables : None

$j \sim \text{Uniform}(1, N)$
$x_j^T, b_T \leftarrow x, j$
for $t \in T, \ldots, 2$ **do**
  $a_{b_t}^{t-1} \sim \text{Uniform}(1, N)$
  $b_{t-1} \leftarrow a_{b_t}^{t-1}$
  $x_{b_{t-1}}^{t-1}, \check{w} \sim$ `IMPORTANCE`$(\tilde{\pi}_{t-1}, L_t(x_{b_t}^t))$ w. aux. vars $v_{L_t}^{b_t}$
  $\hat{w} \sim$ `HME`$(\tilde{\pi}_t, x_{b_t}^t, K_t(x_{b_{t-1}}^{t-1}))$ w. aux. vars $v_{K_t}^{b_t}$
  $w_{b_t}^t \leftarrow (\hat{w} \cdot \check{w})^{-1}$
$w_{b_1}^1 \sim$ `HME`$(\tilde{\pi}_1, x_{b_1}^1, \mathcal{S})$ w. aux. vars $v_{\mathcal{S}}^{b_1}$
for $i \in 1, \ldots, b_1 - 1, b_1 + 1, \ldots, N$ **do**
  $x_i^1, w_i^1 \sim$ `IMPORTANCE`$(\tilde{\pi}_1, \mathcal{S})$ w. aux. vars $v_{\mathcal{S}}^i$
for $t \in 2, \ldots, T$ **do**
  for $i \in 1, \ldots, b_t - 1, b_t + 1, \ldots, N$ **do**
    $a_i^{t-1} \sim \text{Discrete}(w_{1:N}^{t-1})$
    $x_i^t, \hat{w} \sim$ `IMPORTANCE`$(\tilde{\pi}_t, K_t(x_{a_i^{t-1}}^{t-1}))$ w. aux. vars $v_{K_t}^i$
    $\check{w} \sim$ `HME`$(\tilde{\pi}_{t-1}, x_{a_i^{t-1}}^{t-1}, L_t(x_i^t))$ w. aux. vars $v_{L_t}^i$
    $w_i^t \leftarrow \hat{w} \cdot \check{w}$
return $(x_{1:N}^{1:T}, v_{\mathcal{S}}^{1:N}, v_{K_{2:T}}^{1:N}, v_{L_{2:T}}^{1:N}, a_{1:N}^{1:T-1}, j)$

The sequential Monte Carlo family of algorithms [7, 12] evolve a population of *weighted particles* to approximate a sequence of target distributions. SMC can be viewed as standard importance sampling, with an inference strategy in which $\mathcal{S}.q$ is the sampling distribution for SMC, and $\mathcal{S}.\mathcal{M}(x)$ is the conditional SMC algorithm [1].

Standard SMC is parameterized by:

1. A sequence $\tilde{\pi}_{1:T}$ of intermediate target distributions, with $\tilde{\pi}_T = \tilde{\pi}$ the ultimate target;
2. An initial proposal $q(x_1)$;
3. A sequence $K_t(x_{t-1} \to x_t)$ of proposal kernels for $t = 2, \ldots, T$; and
4. A sequence $L_t(x_t \to x_{t-1})$ of backward kernels for $t = 2, \ldots, T$.

Here, we show a version of SMC (the inference strategy `smc`) that behaves as a 'higher-order inference strategy,' or 'inference strategy combinator': it allows for an initial proposal, proposal kernels, and backward kernels that do not have tractable marginal densities. Our version is parameterized by:

1. A sequence $\tilde{\pi}_{1:T}$ of intermediate target distributions, with $\tilde{\pi}_T = \tilde{\pi}$ the ultimate target;
2. An initial proposal $\mathcal{S}$ (a RAVI strategy);
3. A sequence of inference strategy families $K_t(x_{t-1})$ parameterized by x_{t-1}, for $t = 2, \ldots, T$, targeting $\tilde{\pi}_t$; and
4. A sequence of inference strategy families $L_t(x_t)$ of backward kernels, parameterized by x_t, for $t = 2, \ldots, T$.

The posterior approximation $\mathcal{S}.q$ runs a version of SMC that uses `HME` and `IMPORTANCE` to compute weights. The meta-posterior approximation $\mathcal{S}.\mathcal{M}(x).q$ runs a similarly modified version of conditional SMC [1]. When `IMPORTANCE` is run on the `smc` inference strategy, the final weight $\hat{Z}$ is the SMC marginal likelihood esitmate, the product of the averages of the weights from each time step.

It is possible to adapt this strategy to use adaptive resampling and rejuvenation. (Rejuvenation moves do not actually require modification: can be incorporated by including them as explicit (K, L) pairs, where L is the time-reversal of an MCMC kernel K.) However, we are not aware of a way to justify the adaptive choice of rejuvenation kernel.

B.4 VARIATIONAL SEQUENTIAL MONTE CARLO

The Variational Sequential Monte Carlo [30] objective corresponds to Alg. 3, with the same RAVI inference strategy as in Appendix B.3. However, the default gradient estimator from Alg. 3 will have high variance. Naesseth et al. [30] recommend using a biased estimator of the gradient, that uses reparameterization where possible and discards the score function terms arising from resampling steps.

B.5 ANNEALED IMPORTANCE SAMPLING

RAVI Inference Strategy: Annealed Importance Sampling
Posterior Approx. `ais(`$\tilde{\pi}_{1:T}, \mathcal{S}, K_{2:T}$`).q()`
  Target of inference : latent variable x targeting $\tilde{\pi}_T$
  Auxiliary variables : $x^{1:T}$, aux. vars $v_{\mathcal{S}}$ of initial proposal
  $x_1, _ \sim$ `IMPORTANCE`$(\tilde{\pi}_1, \mathcal{S})$ w. aux. vars $v_{\mathcal{S}}$
  for $t \in 2, \ldots, T$ **do**
    $x_t \sim K_t(x_{t-1} \to \cdot)$
  return x_T
Meta-Posterior Approx. `ais(`$\tilde{\pi}_{1:T}, \mathcal{S}, K_{2:T}$`).M(`$x$`).q()`
  Target of inference : $x^{1:T}$, aux. vars $v_{\mathcal{S}}$ of initial proposal
  Auxiliary variables : None
  $x_T \leftarrow x$
  for $t \in T, \ldots, 2$ **do**
    $x_{t-1} \sim \tilde{K}_t(x_t \to \cdot)$ `//` $\tilde{K}_t$ `is time reversal of` K_t
  $_ \sim$ `HME`$(\tilde{\pi}_1, x_1, \mathcal{S})$ w. aux. vars $v_{\mathcal{S}}$
  return $(x_{1:T}, v_{\mathcal{S}})$

In annealed importance sampling, the practitioner chooses a sequence of unnormalized target distributions $\tilde{\pi}_{1:T}$, where π_T is the posterior distribution of interest. Typically π_1 is chosen to be a distribution that is easy to approximate with a proposal q, and each π_i is slightly closer to the true target π_T than the last. The user also chooses a sequence of kernels $K_t(x_{t-1} \to x_t)$, where K_t is stationary for π_{t-1}. The algorithm begins by sampling an initial point $x_1 \sim q$, transforming it through the sequence of kernels to obtain $x_2, \ldots, x_T$, and returning x_T as the inferred value of x. The associated weight is

$$\hat{Z} = \frac{\tilde{\pi}_1(x_1) \cdot \cdots \cdot \tilde{\pi}_T(x_T)}{q(x_1) \cdot \tilde{\pi}_1(x_2) \cdot \cdots \cdot \tilde{\pi}_{T-1}(x_T)}.$$

This procedure corresponds to running Alg. 1 on the `ais` inference strategy. The inference process runs the kernels K_t forward, whereas the meta-inference process runs their time reversals backward: $\tilde{K}_t(x_t \to x_{t-1}) \propto \pi_t(x_{t-1}) \cdot K_t(x_{t-1} \to x_t)$.

Note that if K is a stationary kernel for π_i, so is K^m for any natural number m. With sufficient computation (increasing m), we can ensure that the AIS top-level proposal `ais`$(\ldots)$.q is arbitrarily close to the target posterior π_T. However, doing so will not necessarily lead to lower-variance weights: RAVI makes clear that it is also necessary to consider the quality of meta-inference.

Consider the job of $\tilde{K}_T$, which in the context of the meta-posterior approximation `ais`.$\mathcal{M}(x)$ is supposed to infer x_{T-1} from x_T. $\tilde{K}_T$ is the exact meta-posterior of x_{T-1} given x_T *assuming that, in the forward direction,* x_{T-1} *was distributed according to* π_{T-1}. However, in the forward direction, if each K_t is run sufficiently many times to ensure mixing at each step, x_{T-1} will in fact be distributed according to π_{T-2}. This gap—between the optimal meta-inference kernels and the actual $\tilde{K}$ kernels—is partly responsible for the variance of the AIS estimator, and can be mitigated by using a finer annealing schedule that brings successive target distributions closer together. It could also be mitigated by learning a better reverse annealing chain.

B.6 NESTED SEQUENTIAL MONTE CARLO

We first consider Nested Importance Sampling. As in RAVI, Nested Importance Sampling is concerned with importance sampling when the proposal distribution q cannot be tractably evaluated. But RAVI and NIS take different approaches:

1. RAVI assumes q can be simulated, but that the (normalized) density cannot be evaluated. RAVI generates proposals exactly distributed according to the user's desired proposal $\mathcal{S}.q$, and generates approximations to the ideal importance weights.
2. NIS does not assume q can be simulated, but does assume that its unnormalized density $\tilde{q}$ is available. As such, proposals are not simulated from q, but rather from a Sampling/Importance-Resampling (SIR) approximation to q.

The NIS procedure with an intractable proposal q corresponds exactly to a special case of the RAVI algorithm, with the RAVI proposal $\mathcal{S}.q$ set *not* to q but rather to an SIR sampling distribution targeting q using some tractable proposal h. Compare:

- Ordinary SIR targeting $\tilde{\pi}$ with proposal h: recovered by running `IMPORTANCE`$(\tilde{\pi}, \texttt{sir}(\tilde{\pi}, h, N))$ (see Section B.1 for `sir` inference strategy).
- Nested IS targeting $\tilde{\pi}$ with unnormalized proposal density $\tilde{q}$, approximated using SIR with h as a proposal: recovered by running `IMPORTANCE`$(\tilde{\pi}, \texttt{sir}(\tilde{q}, h, N))$.

That is, under the RAVI perspective, the only difference between ordinary SIR using h and nested IS is that the ideal proposal density $\tilde{q}$ (rather than the target density $\tilde{\pi}$) is used to make the resampling decision about the particles generated by h (the index j in the listing for `sir`).

More generally, Naesseth et al. [29] consider procedures other than SIR for approximating q, arguing that any properly weighted sampler for the intractable proposal q will do. If we let $\mathcal{H}$ be a RAVI inference strategy representing the properly weighted sampler for the intractable proposal q (with unnormalized density $\tilde{q}$), then the Nested IS procedure that uses this properly weighted proposal to perform inference in $\tilde{\pi}$ is $\texttt{IMPORTANCE}(\tilde{\pi}, \texttt{ravi-sir}(\tilde{q}, \mathcal{H}, 1))$ (see `ravi-sir` in Section B.1).

Nested SMC is similar, performing Nested IS at each iteration of SMC. To recover this algorithm using RAVI, we use the `smc` inference strategy, but for the proposals $K_t(x_{t-1})$ (which, as described in Section B.3, can be instantiated with inference strategies), we use `ravi-sir` targeting the desired but intractable proposal.

B.7 SMC^2

RAVI Inference Strategy: SMC^2
Posterior Approx. $\texttt{smc}^2(p, q_1, q, M, N)\texttt{.q()}$
Target of inference : parameters θ, sequence $x_{1:T}$
Auxiliary variables : inner SMC vars $v^T_{\texttt{smc}}$ of chosen SMC^2 particle, other SMC^2 vars v

```
// the targets π̃_t depend on M, p, q_1, and q
```

$(\theta, x_{1:T}, v^T_{\texttt{smc}}), _ \sim \texttt{IMPORTANCE}(\tilde{\pi}_T, \texttt{smc}(\tilde{\pi}_{1:T}, K^2_{2:T}, L^2_{2:T}, N))$ w. aux. vars v

return $\theta, x_{1:T}$

Meta-Posterior Approx. $\texttt{smc}^2(p, q_1, q, M, N)\texttt{.M}(\theta, x_{1:T})\texttt{.q()}$
Target of inference : inner SMC vars $v^T_{\texttt{smc}}$ of chosen SMC^2 particle, other SMC^2 vars v
Auxiliary variables : None

$_ \sim \texttt{HME}(p^\theta_T, x_{1:T}, \texttt{smc}(p^\theta_{1:T}, q_1, K_{2:T}, L_{2:T}, M))$ w. aux. vars $v^T_{\texttt{smc}}$

$_ \sim \texttt{HME}(\tilde{\pi}_T, (\theta, x_{1:T}, v^T_{\texttt{smc}}), \texttt{smc}(\tilde{\pi}_{1:T}, K^2_{2:T}, L^2_{2:T}, N))$ w. aux. vars v

return (v^T_{smc}, v)

Suppose we are working with a state-space model $p(\theta)\prod_{i=1}^T p(x_i \mid x_{1:i}, \theta)p(y_i \mid x_i, \theta)$. For a fixed θ, an SMC algorithm could be used to target the successive posteriors $p^\theta_t(x_{1:t}) = p(x_{1:t} \mid y_{1:t}, \theta)$, with proposal kernels $K_t(x^{t-1}_{1:t-1} \to x^t_{1:t}) = \delta_{x^{t-1}_{1:t-1}}(x^t_{1:t-1})q(x^t_t; x^t_{1:t-1}, y_{1:t}, \theta)$ (for some choice of q) and deterministic backward kernels $L_t(x^t_{1:t} \to x^{t-1}_{1:t-1}) = \delta_{x^t_{1:t-1}}(x^{t-1}_{1:t-1})$. The RAVI strategy implementing that SMC algorithm is $\texttt{smc}(p_{1:T}, q_1, K_{2:T}, L_{2:T}, N)$, where $q_1(x_1; \theta)$ is a proposal for an initial x_1 and N is the number of particles.

If we also wish to infer θ, we can instead use the SMC^2 algorithm [8]. We define extended targets

$$\pi_t(\theta, x_{1:t}, v^t_{\texttt{smc}}) = p(\theta \mid y_{1:t})p(x_{1:t} \mid y_{1:t}, \theta)p_{\texttt{HME}}^{\texttt{smc}(p^\theta_{1:t}, q_1, K_{2:t}, L_{2:t}, N)}(v^t_{\texttt{smc}}; x_{1:t}),$$

which are defined over not only θ and $x_{1:t}$ but also all the auxiliary variables $v^t_{\texttt{smc}}$ used during steps 1 through t of SMC. The variables $v_{\texttt{smc}}$ and the $p_{\texttt{HME}}$ distribution over them are as defined in Appendix A.5. We write $\tilde{\pi}_t$ for the unnormalized versions of these targets, with normalizing constant $p(y_{1:t})$.

The SMC^2 algorithm targets this sequence of extended posteriors. We write K^2_t for the forward kernels used by this outer SMC algorithm. The kernel K^2_t extends the SMC state variables $v^{t-1}_{\texttt{SMC}}$ to new state variables $v^t_{\texttt{SMC}}$ by running the particle filter forward one step, resampling the chosen trajectory index j based on the new weights for time step t, and updating $x_{1:t}$ to match the jth trajectory. The corresponding backward kernel L^2_t deletes the tth step of the particle deterministically, then reproposes j based on the step $t-1$ weights, setting $x_{1:t-1}$ to match the jth trajectory.

The SMC^2 algorithm corresponds to the RAVI strategy `smc`^{2}. Running the other SMC yields an approximate sample from $\tilde{\pi}_T$, which includes auxiliary variables $v^T_{\texttt{smc}}$. Meta-inference runs two rounds of conditional SMC: first, to recover the inner layer of SMC's variables $v^T_{\texttt{smc}}$ for the chosen outer-layer particle, and second, to recover the outer layer of SMC's auxiliary variables v. As discussed by Chopin et al. [8], particle MCMC rejuvenation moves can also be included; to justify using RAVI, we would insert these kernels as additional proposals within the sequence $K^2_{2:T}$.

B.8 AMORTIZED REJECTION SAMPLING

Consider a generative model $p(K, x_{1:K+1}, y)$ where the latent variables $x_{1:K+1}$ to be marginalized or inferred represent the trace of a rejection sampling loop, with sampling distribution $h(x)$ and predicate $\mathcal{A}(x)$ determining acceptance:

$$p(K, x_{1:K+1}, y) = \prod_{i=1}^{K} \left[h(x_i)(1 - \mathcal{A}(x_i))\right] h(x_{K+1})\mathcal{A}(x_{K+1})p(y \mid x_{K+1})$$

RAVI Inference Strategy: Amortized Rejection Sampling

Posterior Approx. `amrej(`$h, q, \mathcal{A}, N, M$`).q()`

Target of inference : number K of rejected samples, rejected samples $x_{1:K}$, accepted sample x_{K+1}

Auxiliary variables : rejection loops $(K', x'_{1:K'})$ and $(K''_i, x''^{i}_{1:K''_i})_{i=1:M}$, index j

$K' \leftarrow 0$
$x'_1 \sim q$
while $\mathcal{A}(x'_{K'+1}) \neq 1$ **do**
  $K' \leftarrow K' + 1$
  $x'_{K'+1} \sim q$
$x_{K+1} \leftarrow x'_{K'+1}$
for $i \in 1, \ldots, M$ **do**
  $K''_i \leftarrow 0$
  $x''^{i}_1 \sim h$
  while $\mathcal{A}(x''^{i}_{K''_i+1}) \neq 1$ **do**
    $K''_i \leftarrow K''_i + 1$
    $x''^{i}_{K''_i+1} \sim h$
$j \sim \text{Discrete}(K''_{1:M})$
$K \sim \text{Uniform}(0, K''_j)$
$x_{1:K} \leftarrow x''^{j}_{1:K}$
return $(K, x_{1:K}, x_{K+1})$

Meta-Posterior Approx. `amrej(`$h, q, \mathcal{A}, N, M$`).M(`$K, x_{1:K}$`).q()`

Target of inference : rejection loops $(K', x'_{1:K'})$ and $(K''_i, x''^{i}_{1:K''_i})_{i=1:M}$, index j

Auxiliary variables : superfluous accepted sample $z_{K'+1}$

$j \sim \text{Uniform}(1, M)$
for $i \in 1, \ldots, j-1, j+1, \ldots, M$ **do**
  $K''_i \leftarrow 0$
  $x''^{i}_1 \sim h$
  while $\mathcal{A}(x''^{i}_{K''_i+1}) \neq 1$ **do**
    $K''_i \leftarrow K''_i + 1$
    $x''^{i}_{K''_i+1} \sim h$
$K''_j \leftarrow K$
$x''^{j}_{K+1} \sim h$
while $\mathcal{A}(x''^{j}_{K+K''_j+1}) \neq 1$ **do**
  $K''_j \leftarrow K''_j + 1$
  $x''^{j}_{K+K''_j+1} \sim h$
$K' \leftarrow 0$
$z_1 \sim q$
while $\mathcal{A}(z_{K'+1}) \neq 1$ **do**
  $x'_{K'+1} \leftarrow z_{K'+1}$
  $K' \leftarrow K' + 1$
  $z_{K'+1} \sim q$
return $(K', x'_{1:K'}, (K''_i, x''^{i}_{1:K''_i})_{i=1:M}, j)$

Meta-Meta-Posterior Approx.
`amrej(`$h, q, \mathcal{A}, N, M$`).M(`$K, x_{1:K}$`).M(`$K', x'_{1:K'}, (K''_i, x''^{i}_{1:K''_i})_{i=1:M}, j$`).q()`

Target of inference : superfluous accepted sample $z_{K'+1}$

Auxiliary variables : index l, unchosen particles z_{-l}

for $i \in 1, \ldots, N$ **do**
  $z_i \sim q$
$l \sim \text{Uniform}(\{i \mid \mathcal{A}(z_i)\})$
return z_l

Meta-Meta-Meta-Posterior Approx.
`amrej(`$h, q, \mathcal{A}, N, M$`).M(`$K, x_{1:K}$`).M(`$K', x'_{1:K'}, (K''_i, x''^{i}_{1:K''_i})_{i=1:M}, j$`).M(`$z_{K'+1}$`).q()`

Target of inference : index l, unchosen particles z_{-l}

Auxiliary variables : None

$l \sim \text{Uniform}(1, N)$
for $i \in 1, \ldots, l-1, l+1, \ldots, N$ **do**
  $z_i \sim q$
return $(z_1, \ldots, z_{l-1}, z_{l+1}, \ldots, z_N)$

Here, the x_i are drawn independently from a distribution h, until some predicate $\mathcal{A}$ holds of the most recent particle, at which point the loop stops. The observation y depends on the final sample x_{K+1}, but not the earlier, rejected samples $x_{1:K}$ or the number of rejected samples K. Naderiparizi et al. [28] proposed a technique called *Amortized Rejection Sampling* for performing inference in this model. The technique corresponds to the rather involved RAVI strategy `amrej`, which has parameters N and M that can be used to trade accuracy for computational cost.

The idea behind the top-level, intractable posterior approximation `amrej`$(h, q, \mathcal{A}, N, M).q$ is to:

- use the observation y to intelligently guess the *accepted* particle x_{K+1}, using a learned proposal q. (For example, q may be parameterized by a neural network that accepts y as input.) To satisfy the constraint that x_{K+1} satisfies $\mathcal{A}$, however, it is necessary to run q within a rejection sampling loop, generating auxiliary variables $x'_{1:K'}$, where K' is the number of rejected q-samples. (We could try directly using $x'_{1:K'}$ as our proposal for $x_{1:K}$, the rejected samples from the model. But q's goal is to propose x_{K+1} in a *data-driven* way, influenced by the observation y, and the rejected samples $x_{1:K}$ from the model have no connection to the data—so, using samples from q as proposals for the rejected model samples would result in a poor approximation.)
- use rejection sampling from the prior h to infer the *rejected* samples $x_{1:K}$. We

run M independent rejection sampling loops, randomly choose one with probability proportional to its length, and then randomly choose a *prefix* of the chosen loop as our proposal for $x_{1:K}$.

The meta-posterior approximation must solve two new challenges: recovering the rejected q samples $x'_{1:K'}$ from the posterior approximation, and recovering the many unused rejection loops (and the suffix of the chosen rejection loop) from the second step of the posterior approximation (the x'' variables). The latter of these tasks is simple enough: we can generate $M-1$ rejection loops from scratch for the un-chosen loops, and a further rejection loop from scratch to use as the suffix of the chosen loop. The first task is more complex: we run a new rejection loop using q as a proposal, and discard the final accepted sample. Meta-meta-inference must infer this discarded accepted sample, for which it uses SIR with N particles. The final layer, the Meta-Meta-Meta-Posterior Approximation, uses conditional SIR.

The meta-meta-posterior is not absolutely continuous with respect to its approximation (it is possible that the approximation generates N z-values that all fail to satisfy the predicate, in which case z_l is not in the support of the meta-meta-posterior). As such, this is an example of a *wide* inference strategy (Appendix C).

B.9 HAMILTONIAN VARIATIONAL INFERENCE

RAVI Inference Strategy: Hamiltonian Variational Inference
Posterior Approx. `hamvi(`q_0, q_v, r_v`, LF).q()`
  Target of inference : latent variable x
  Auxiliary variables : initial position x_0, momentum v
  $x_0 \sim q_0$
  $v \sim q_v$
  $(x, v') \leftarrow \text{LF}(x_0, v)$
  return x
Meta-Posterior Approx. `hamvi(`q_0, q_v, r_v`, LF).M(`x`).q()`
  Target of inference : initial position x_0, momentum v
  Auxiliary variables : negated final momentum v'_-
  $v'_- \sim r_v(\cdot; x)$
  $(x_0, v_-) \leftarrow \text{LF}(x, v'_-)$
  return $(x_0, -v_-)$
Meta-Posterior Approx.
`hamvi(`q_0, q_v, r_v`, LF).M(`x`).M(`x_0, v`).q()`
  Target of inference : negated final momentum v'_-
  Auxiliary variables : None
  $(_, v') \leftarrow \text{LF}(x_0, v)$
  return $-v'$

Hamiltonian Variational Inference [37] is a hybrid of Hamiltonian Monte Carlo and variational inference. It is a special case of Markov Chain Variational Inference (see Section 2 and Section 5 for detailed discussion, and `mcvi` for the RAVI implementation). The algorithm specializes the Markov Chain Variational Inference procedure for use with a Hamiltonian Monte Carlo kernel.

We present the specialized strategy as `hamvi`. It accepts as input:

1. a distribution q_0 from which to propose an initial point;
2. a momentum distribution q_v from which momenta v are proposed at each iteration;
3. a proposal distribution $r_v(\cdot; x)$ over momenta; and
4. a leapfrog integrator `LF` that runs Hamiltonian dynamics on an initial position and momentum (we think of both the number of leapfrog steps L and the Hamiltonian H being targeted as part of the `LF` object provided to `hamvi`).

Given these inputs, the top-level posterior approximation runs an iteration of HMC from a randomly initialized location x_0. The meta-posterior approximation randomly proposes a (negated) *final* momentum from the proposal r_v, and runs the leapfrog integrator to find a plausible initial location x_0. Finally, the (deterministic) meta-meta-posterior finds the initial momentum that could have taken x_0 to x.

B.10 ANTITHETIC SAMPLING

Consider a target $\tilde{\pi}(x)$ and a proposal $q(x)$ that approximates π. Suppose q is invariant under some bijective transformation T:

$$\forall x, q(x) = q(T(x)).$$

For example, a univariate Gaussian proposal with mean μ is invariant under $T(x) = 2\mu - x$. Antithetic sampling generates a sample x from q, but instead of using the estimator $\hat{Z} = \tilde{\pi}(x)/q(x)$, it uses

$$\hat{Z} = \frac{\tilde{\pi}(x) + \tilde{\pi}(T(x))}{2q(x)}.$$

RAVI Inference Strategy: Antithetic Sampling

Posterior Approx. `antithetic(`$\tilde{\pi}, q, T$`).q()`

Target of inference : latent variable x

Auxiliary variables : sampled x_0, choice b

$x_0 \sim q$

$w_0 \leftarrow \tilde{\pi}(x_0)/q(x_0)$

$w_1 \leftarrow \tilde{\pi}(T(x_0))/q(x_0)$

$b \sim \texttt{Bernoulli}(\frac{w_1}{w_0+w_1})$

return $bT(x_0) + (1-b)x_0$

Meta-Posterior Approx. `antithetic(`$\tilde{\pi}, q, T$`).M(`x`).q()`

Target of inference : sampled x_0, choice b

Auxiliary variables : None

$b \sim \text{Bernoulli}(0.5)$

$x_0 \leftarrow bT(x) + (1-b)x$

return (x_0, b)

This can be justified as Algorithm 1 (`IMPORTANCE`) applied to the strategy `antithetic`. The posterior approximation generates an initial sample $x_0 \sim q$, evaluates both x_0 and $T(x_0)$ as possible proposals, and selects one. The meta-posterior approximation must recover whether x or its transformed version was the sampled one; it does so by flipping a fair coin, which is optimal when $T = T^{-1}$, i.e., when T is an involution. In the general case a lower-variance estimator could be derived by setting $\mathcal{M}(x).q$ to the exact posterior of the proposal process. Antithetic sampling can also be generalized to the case where a finite family of bijective transformations T_i are available.

Note that although the final expression for $\hat{Z}$ falls out of this inference strategy only when $q(x) = q(T(x))$ for all x, nothing in the inference strategy itself exploits this assumption, and the same inference strategy could be applied to T without this property, to derive other estimators that—intuitively—simultaneously consider a proposal x and a deterministic function of it $T(x)$ as possible locations.

C ABSOLUTE CONTINUITY

When we defined inference strategies $\mathcal{S}$ targeting π, we required that $\mathcal{S}.q$ and π be *mutually* absolutely continuous, a stronger requirement than in importance sampling. We now consider relaxing this requirement, by requiring only *one-sided* absolute continuity. We define two *kinds* of inference strategy, depending on which direction of absolute continuity holds:

1. An inference strategy $\mathcal{S}$ targeting π is *wide* if π is absolutely continuous with respect to $\mathcal{S}.q$, and either $\mathcal{S}.q$ has a tractable marginal density or $\mathcal{S}.\mathcal{M}(x)$ is a narrow inference strategy targeting $\mathcal{S}.q(\cdot \mid x)$ for all x.

2. An inference strategy $\mathcal{S}$ targeting π is *narrow* if $\mathcal{S}.q$ is absolutely continuous with respect to π, and either $\mathcal{S}.q$ has a tractable marginal density or $\mathcal{S}.\mathcal{M}(x)$ is a wide inference strategy targeting $\mathcal{S}.q(\cdot \mid x)$ for all x.

Then an inference strategy as defined in the main paper is one that is both wide and narrow.

Narrow inference strategies can serve as variational families within variational inference algorithms. Wide inference strategies can be used as importance sampling and SMC proposals, as well as variational families for *amortized* variational inference. Inference strategies used as MCMC proposals must be both wide and narrow.

D OTHER APPLICATIONS OF RAVI INFERENCE STRATEGIES

D.1 REJECTION SAMPLING WITH RAVI

As in any properly weighted sampler, if the weights produced by Alg. 1 can be bounded above by a constant M, a RAVI inference strategy can be used for exact inference via rejection sampling: a sample $(x, \hat{Z})$ is drawn using Alg. 1, and then accepted with probability $\frac{\hat{Z}}{M}$. The weight $\hat{Z}$ for an inference strategy can be viewed as a product of the normalizing constant Z with normalized importance weights $w_{\mathcal{S}} = \frac{\pi(x)}{\mathcal{S}.q(x)}$, $w_{\mathcal{S}.\mathcal{M}(x)} = \frac{\mathcal{S}.q(r|x)}{\mathcal{S}.\mathcal{M}(x).q(r)}$, and so on. As such, if upper bounds M_Z and $M_{\mathcal{S}}$, $M_{\mathcal{S}.\mathcal{M}(x)}$, etc. can be found for these quantities, the product of these bounds is a bound on $\hat{Z}$. Thus, as in properly weighted sampling and in variational inference with RAVI, it is possible to reason about the RAVI inference strategy compositionally, in terms of bounds at each layer of nesting.

D.2 ESTIMATING KL DIVERGENCES BETWEEN MODELS WITH RAVI INFERENCE STRATEGIES EQUIPPED

Suppose $p(y) = \int p(x, y)\mathrm{d}x$ and $q(y) = \int q(z, y)\mathrm{d}z$ are mutually absolutely continuous distributions over some space $\mathcal{Y}$. Suppose also that we have two families of inference strategies, $\mathcal{S}_p(y)$ and $\mathcal{S}_q(y)$, targeting $p(x \mid y)$ and $q(z \mid y)$ respectively. Then the AIDE algortihm [10] can be adapted to give a stochastic upper bound on the symmetric KL divergence between $p(y)$ and $q(y)$.

First, we generate $(x, y_p) \sim p$, $(z, y_q) \sim q$, and run `HME` on each pair to obtain weights w_p^p and w_q^q respectively. Then, we run `IMPORTANCE` on p with data y_q, and on q with data y_p, to obtain weights w_q^p and w_p^q respectively. Finally, we sum the logs of the foru weights, to give an estimate $\hat{D}$ whose expectation is:

$$\mathbb{E}[\hat{D}] = \mathbb{E}_{y\sim p}[\mathcal{U}(p, y, \mathcal{S}_p(y)) - \mathcal{L}(q, y, \mathcal{S}_q(y))] + \mathbb{E}_{y\sim q}[\mathcal{U}(q, y, \mathcal{S}_q(y)) - \mathcal{L}_p(p, y, \mathcal{S}_p(y))] \geq KL(p||q) + KL(q||p).$$

As the marginal likelihood bounds $\mathcal{U}$ and $\mathcal{L}$ become tighter, this expectation approaches the true symmetric KL between p and q, i.e., $D = KL(p||q) + KL(q||p)$. Theorem 4 allows us to characterize the tightness of these bounds, and thus of the stochastic upper bound $\hat{D}$ on the symmetric KL, in terms of KL divergences between successive layers of each inference strategy. Improving inference at any layer of the inference strategy tightens the bound $\hat{D}$, yielding less biased estimates of D.

E REPARAMETERIZATION TRICK GRADIENT ESTIMATORS

In this section, we present versions of Algorithms 3 and 4 that utilize reparameterization gradients, rather than score function gradients. Using these algorithms requires that an inference strategy be *reparameterizable*.

Definition: A *reparameterizable inference strategy $\mathcal{S}$ with arguments θ* specifies:

- A reparameterizable posterior approximation $\mathcal{S}.q$, which is one of:
  - a tractable proposal: a tuple $(\mathcal{S}.q(x;\theta), \mathcal{S}.q.g(\epsilon), \mathcal{S}.q.f(\epsilon, \theta)$, such that q is the pushforward of g by f; or
  - an intractable proposal: a tuple $(\mathcal{S}.q(r, x;\theta), \mathcal{S}.q.g(\epsilon_r, \epsilon_x), \mathcal{S}.q.f_r(\epsilon_r, \theta), \mathcal{S}.q.f_x(\epsilon_x, \theta))$, such that q is the pushforward of g by $\lambda(\epsilon_r, \epsilon_x).(f_r(\epsilon_r, \theta), f_x(\epsilon_x, \theta))$.
- If the latter, a reparameterizable meta-inference strategy $\mathcal{S}.\mathcal{M}$, with arguments (x, θ), that given argument (x, θ), targets $\mathcal{S}.q(r \mid x;\theta)$.

Now, reparameterized estimators can be derived by applying standard automatic differentiation to the following algorithm, which only samples from distributions that do not depend on parameters:

Algorithm 6: RAVI ELBO estimator (`ELBO`)
Input: unnormalized model $\tilde{p}(x)$
Input: inference strategy $\mathcal{S}$ with arguments
Input: arguments θ
Output: unbiased estimates of $\mathcal{L}$ (differentiable w.r.t. θ)

if $\mathcal{S}.q$ *has a tractable marginal density* **then**
  $\epsilon_x \sim \mathcal{S}.q.g$
  $x \leftarrow \mathcal{S}.q.f(\epsilon_x, \theta)$
  $\hat{U} \leftarrow \log \mathcal{S}.q(x;\theta)$
else if $\mathcal{S}.q(x;\theta) = \int \mathcal{S}.q(r, x;\theta)dr$ **then**
  $(\epsilon_r, \epsilon_x) \sim \mathcal{S}.q.g$
  $(x, r) \leftarrow (\mathcal{S}.q.f_x(\epsilon_x, \theta), \mathcal{S}.q.f_r(\epsilon_r, \theta))$
  $\hat{U} \leftarrow \mathtt{EUBO}(\mathcal{S}.q(\cdot, x;\theta), r, \mathcal{S}.\mathcal{M}, (x, \theta))$
return $\log \tilde{p}(x) - \hat{U}$

Algorithm 7: RAVI EUBO estimator (`EUBO`)
Input: unnormalized model $\tilde{p}(x)$
Input: exact sample $x \sim p(x)$
Input: inference strategy $\mathcal{S}$ with arguments
Input: arguments θ
Output: unbiased estimate of $\mathcal{U}$ (differentiable w.r.t. θ)

if $\mathcal{S}.q$ *has a tractable marginal density* **then**
  $\hat{L} \leftarrow \log \mathcal{S}.q(x;\theta)$
else if $\mathcal{S}.q(x;\theta) = \int \mathcal{S}.q(r, x;\theta)dr$ **then**
  $\hat{L} \leftarrow \mathtt{ELBO}(\mathcal{S}.q(\cdot, x;\theta), \mathcal{S}.\mathcal{M}, (x, \theta))$
return $\log \tilde{p}(x) - \hat{L}$

Note that in fact only every *other* posterior approximation in the unrolled strategy requires a reparameterized version: Algorithm 7 never samples from its $\mathcal{S}.q$, only evaluates the densities.

It would be interesting to develop variants of these algorithms that allow users to combine score-function and reparameterization estimation at different layers of nesting, or exploit other variance reduction tactics compositionally.