Recursive Monte Carlo and Variational Inference with Auxiliary Variables (Supplemental Material)

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

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

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:*

- $\texttt{IMPORTANCE}(\mathcal{S}, \tilde{\pi})$ *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 $\mathbb{E}[\hat{Z}(\tilde{\pi}, \mathcal{S})] = Z$.*
- $\mathbb{E}[\check{Z}(\tilde{\pi}, \mathcal{S})^{-1}] = \mathbb{E}_{x\sim\pi}[\texttt{HME}(\mathcal{S}, x, \tilde{\pi})] = Z^{-1}$.

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

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

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

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, `IMPORTANCE` generates $x \sim \mathcal{S}.q$ and $r \sim \mathcal{S}.q(r \mid x)$. In the call to `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 `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 `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, `HME` calls `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

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

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* $\log p(y)$*:*

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

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:

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

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:

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

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** $u < min(1, \frac{\hat{Z}_{x'}}{\hat{Z}_x} w_{x'} w_x)$ **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 `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 `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 `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 `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.[1]

[1] This 'combinator' viewpoint evokes earlier work by [13] and [14]. For example, [14] 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**
  $x_i, w_i \sim \texttt{IMPORTANCE}(\tilde{\pi}, \mathcal{S})$ 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

$j \sim \text{Uniform}(1, N)$
$x_j \leftarrow x$
$\_ \sim \texttt{HME}(\tilde{\pi}, x_j, \mathcal{S})$ 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 [3]. However, [6] 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 [6]'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 [4, 8] 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 [11] 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. [11] 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
1 $x_1, \_ \sim$ `IMPORTANCE`$(\tilde{\pi}_1, \mathcal{S})$ w. aux. vars $v_{\mathcal{S}}$
2 **for** $t \in 2, \ldots, T$ **do**
3 $x_t \sim K_t(x_{t-1} \to \cdot)$
4 **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
1 $x_T \leftarrow x$
2 **for** $t \in T, \ldots, 2$ **do**
3 $x_{t-1} \sim \tilde{K}_t(x_t \to \cdot)$ `//` $\tilde{K}_t$ `is time reversal of` K_t
4 $\_ \sim$ `HME`$(\tilde{\pi}_1, x_1, \mathcal{S})$ w. aux. vars $v_{\mathcal{S}}$
5 **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. [10] 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
1 `// the targets` $\tilde{\pi}_t$ `depend on` M, p, q_1, `and` q
2 $(\theta, x_{1:T}, v^T_{\texttt{smc}}), \_ \sim \texttt{IMPORTANCE}(\tilde{\pi}_T, \texttt{smc}(\tilde{\pi}_{1:T}, K^2_{2:T}, L^2_{2:T}, N))$ w. aux. vars v
3 **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
1 $\_ \sim \texttt{HME}(p^\theta_T, x_{1:T}, \texttt{smc}(p^\theta_{1:T}, q_1, K_{2:T}, L_{2:T}, M))$ w. aux. vars $v^T_{\texttt{smc}}$
2 $\_ \sim \texttt{HME}(\tilde{\pi}_T, (\theta, x_{1:T}, v^T_{\texttt{smc}}), \texttt{smc}(\tilde{\pi}_{1:T}, K^2_{2:T}, L^2_{2:T}, N))$ w. aux. vars v
3 **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 [5]. We define extended targets

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

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 j^{th} trajectory. The corresponding backward kernel L^2_t deletes the t^{th} step of the particle deterministically, then reproposes j based on the step $t-1$ weights, setting $x_{1:t-1}$ to match the j^{th} 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. [5], 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. [9] 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
1 $x_0 \sim q_0$
2 $v \sim q_v$
3 $(x, v') \leftarrow \text{LF}(x_0, v)$
4 **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'_-
1 $v'_- \sim r_v(\cdot; x)$
2 $(x_0, v_-) \leftarrow \text{LF}(x, v'_-)$
3 **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
1 $(\_, v') \leftarrow \text{LF}(x_0, v)$
2 **return** $-v'$

Hamiltonian Variational Inference [12] 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
1 $x_0 \sim q$
2 $w_0 \leftarrow \tilde{\pi}(x_0)/q(x_0)$
3 $w_1 \leftarrow \tilde{\pi}(T(x_0))/q(x_0)$
4 $b \sim$ `Bernoulli`$(\frac{w_1}{w_0+w_1})$
5 **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
1 $b \sim$ Bernoulli(0.5)
2 $x_0 \leftarrow bT(x) + (1-b)x$
3 **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 [7] 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**
  $(\epsilon_r, \epsilon_x) \sim \mathcal{S}.q.g$
  $(x, r) \leftarrow (\mathcal{S}.q.f_x(\epsilon_x, \theta), \mathcal{S}.q.f_r(\epsilon_r, \theta))$
  $\hat{U} \leftarrow \texttt{EUBO}(\mathcal{S}.q(\cdot, x;\theta), r, \mathcal{S}.\mathcal{M}, (x, \theta))$
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**
  $\hat{L} \leftarrow \texttt{ELBO}(\mathcal{S}.q(\cdot, x;\theta), \mathcal{S}.\mathcal{M}, (x, \theta))$
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.

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

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

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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