How to Approximate Inference with Subtractive Mixture Models

Lena Zellinger
University of Edinburgh

Nicola Branchini
University of Warwick

Lennert De Smet
KU Leuven

Víctor Elvira
University of Edinburgh

Nikolay Malkin
University of Edinburgh

Antonio Vergari
University of Edinburgh

Abstract

Classical mixture models (MMs) are widely used tractable proposals for approximate inference settings such as variational inference (VI) and importance sampling (IS). Recently, mixture models with negative coefficients, called subtractive mixture models (SMMs), have been proposed as a potentially more expressive alternative. However, how to effectively use SMMs for VI and IS is still an open question as they do not provide latent variable semantics and therefore cannot use sampling schemes for classical MMs. In this work, we study how to circumvent this issue by designing several expectation estimators for IS and learning schemes for VI with SMMs, and we empirically evaluate them for distribution approximation. Finally, we discuss the additional challenges in estimation stability and learning efficiency that they carry and propose ways to overcome them. Code is available at https://github.com/april-tools/delta-vi.

1 INTRODUCTION

Mixture models (MMs) are a staple in probabilistic modeling since they can represent complex multimodal distributions via a *convex combination of simple probability density functions* (PDFs) (McLachlan et al., 2019). A classical MM is defined as

$$q_{\text{MM}}(\boldsymbol{x}) = \sum_{k=1}^{K} \alpha_k q_k(\boldsymbol{x}), \quad \alpha_k \geq 0, \quad \sum_{k=1}^{K} \alpha_k = 1 \quad (1)$$

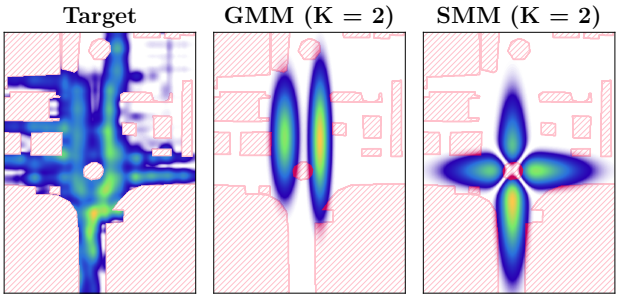


Figure 1: **SMMs are effective variational families for targets with disconnected support** such as trajectories over the walkable area from the *Stanford drone dataset* (Robicquet et al., 2016). With just $K = 2$ components, the SMM *learns the absence of density* at the central roundabout while a GMM requires $K > 2$. We lay the foundation to use SMMs for VI.

for every input $\boldsymbol{x} \in \mathbb{R}^D$, where q_k are the mixture components and α_k are the mixture weights. The coefficients α_k can be interpreted as the parameters of a categorical prior in a discrete latent variable model (Bishop and Nasrabadi, 2006; Peharz et al., 2016), which enables efficient ancestral sampling from q_{MM}. As a result, MMs have been extensively used in approximate inference settings, where the goal is to estimate intractable quantities w.r.t. a target density by sampling from a *tractable surrogate*. In particular, MMs are common surrogates in variational inference (VI) (Jaakkola and Jordan, 1998; Guo et al., 2016; Morningstar et al., 2021; Kviman et al., 2022) and well-studied proposals for (multiple) importance sampling (IS) (Veach and Guibas, 1995; Owen and Zhou, 2000; Elvira et al., 2019; Sbert et al., 2018).

Recently, *subtractive mixture models* (SMMs) (Marteau-Ferey et al., 2020; Rudi and Ciliberto, 2021; Loconte et al., 2024; Cai et al., 2024a) have been introduced in ML as a generalization of classical additive MMs that relaxes the convexity constraint on

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

the mixture coefficients in Eq. (1) by allowing them to take negative values. The resulting *subtraction of density* can be particularly beneficial when modeling targets that have deep valleys or disconnected support (see Fig. 1). *Squaring SMMs* (Loconte et al., 2024) has enabled unconstrained gradient-based learning via maximum likelihood, and squared SMMs have shown good results in data-driven settings (Loconte et al., 2023, 2025b; Wang and Van den Broeck, 2025). Using SMMs also for approximate inference – without access to target samples – can open up a wider class of tractable surrogates that can be more *expressive efficient* (Choi et al., 2020): they allow to model complex distributions with potentially exponentially fewer parameters than classical MMs. However, this additional flexibility comes at the cost of *losing the latent variable interpretation* of the mixture weights. Negative coefficients cannot be interpreted as probabilities, and hence SMMs do not allow for ancestral sampling, raising the question:

> *How can we make SMMs viable tractable surrogates for IS and VI despite the lack of a latent variable semantics?*

A first step in that direction comes from Cai et al. (2024a) who devise a specialized closed-form VI scheme for squared SMMs with orthogonal polynomials as components, optimizing the Fisher divergence (Hyvärinen and Dayan, 2005). In this work, we take a broader perspective and study *general expectation estimation over SMMs*, which is central to (1) learning SMMs via (black-box) VI and (2) using the resulting models as proposals for IS to estimate quantities of interest. We detail our contributions below.

Contributions. In §3 we first provide a **(C1)** comprehensive discussion of sampling schemes for expectation estimation over SMMs. We cover the standard techniques auto-regressive inverse transform sampling (ARITS, Loconte et al., 2024) and rejection sampling (Bignami and De Matteis, 1971), but also discuss **(C2)** ΔIS – a new estimator recently hinted at by Robert and Stoehr (2025), which relies on a decomposition of the SMM into two additive MMs, recovering ancestral sampling. We further develop a *safe variant* of ΔIS, inspired by the safe adaptive IS literature (Delyon and Portier, 2021; Korba and Portier, 2022), to stabilize it in practice. In §4, we then **(C3)** show how each of these techniques can be used for black-box VI. Notably, we discuss how ΔIS opens up the use of the reparameterization trick (Morningstar et al., 2021). Finally, in §6, we **(C4)** extensively evaluate our pipelines for VI and IS on targets of varying dimensionality, providing a first empirical comparison to additive MMs and highlighting open challenges and future opportunities for approximate inference with SMMs.

2 SUBTRACTIVE MIXTURES

Subtractive mixture models (SMMs) generalize classical MMs (Eq. (1)) by allowing for negative mixture coefficients, leading to a potential subtraction of density. A SMM over $\boldsymbol{x} \in \mathbb{R}^D$ is defined as

$$q_{\text{SMM}}(\boldsymbol{x}) = Z^{-1} \cdot \sum_{k=1}^{K} \alpha_k q_k(\boldsymbol{x}), \text{ where } \alpha_k \in \mathbb{R}, \quad \text{(SMM)}$$

where each q_k is a (possibly unnormalized) PDF and $Z = \sum_{k=1}^{K} \alpha_k \int q_k(\boldsymbol{x}) d\boldsymbol{x}$ is the normalizing constant of the SMM. A key challenge when constructing and *learning* SMMs is ensuring $q_{\text{SMM}}(\boldsymbol{x}) \geq 0$ for all inputs $\boldsymbol{x}$ in order to retain a valid PDF. While it is possible to derive closed-form constraints for simple parametric distributions, such as Gaussian, Gamma, and Weibull (Jiang et al., 1999; Zhang and Zhang, 2005; Rabusseau and Denis, 2014), this is non-trivial in general. To this end, Loconte et al. (2024) learned *squared SMMs*, ensuring non-negativity by squaring Eq. (SMM):

$$\begin{aligned} q_{\text{SMM}^2}(\boldsymbol{x}) &= Z^{-1} \cdot \left(\sum_{k=1}^{K} \alpha_k q_k(\boldsymbol{x}) \right)^2 \qquad \text{(SMM}^2\text{)} \\ &= Z^{-1} \cdot \sum_{k=1}^{K} \sum_{k'=1}^{K'} \alpha_k \alpha_{k'} q_k(\boldsymbol{x}) q_{k'}(\boldsymbol{x}), \end{aligned}$$

where $Z = \sum_{k=1}^{K} \sum_{k'=1}^{K'} \alpha_k \alpha_{k'} \int q_k(\boldsymbol{x}) q_{k'}(\boldsymbol{x}) d\boldsymbol{x}$. Fig. 7 shows the computational graph of a squared SMM. Note that Eq. (SMM^2) is still a SMM since negative coefficients $\alpha_k \alpha_{k'} < 0$ are possible.While a squared SMM has $\binom{K+1}{2}$ components after squaring, the number of learnable parameters is still $\mathcal{O}(K)$. To exactly compute Z, and retain tractability, we need an analytical form for $\int q_k(\boldsymbol{x}) q_{k'}(\boldsymbol{x}) d\boldsymbol{x}$, which is the case for exponential families and other functions, such as polynomials on bounded intervals (Loconte et al., 2024). If the components q_k form an orthonormal basis, and $\sum_i \alpha_i^2 = 1$, the SMM will be normalized by design (Cai et al., 2024a; Loconte et al., 2026). In terms of *expressive efficiency* (Choi et al., 2020) squared SMMs and additive MMs are *incomparable*: A single squared SMM can require exponentially fewer parameters than an additive MM to represent certain distributions (Loconte et al., 2024), but the opposite can also be true (Loconte et al., 2025b; Wang and Van den Broeck, 2025). However, a *sum of squared (SOS) SMMs* can be more expressive efficient than both classical MMs and a squared SMM (Loconte et al., 2025b). A simple SOS model can be built by using *complex mixture weights* $\alpha_k := a_k + b_k i$ and multiplying it with its complex conjugate (Loconte et al., 2025b) resulting in

$$\begin{aligned} Z^{-1} \cdot \Bigg(&\sum_{k=1}^{K} \sum_{k'=1}^{K'} a_k a_{k'} q_k(\boldsymbol{x}) q_{k'}(\boldsymbol{x}) \qquad \text{(SOS)} \\ &+ \sum_{k=1}^{K} \sum_{k'=1}^{K'} b_k b_{k'} q_k(\boldsymbol{x}) q_{k'}(\boldsymbol{x}) \Bigg). \end{aligned}$$

Algorithm 1: ARITS(q, S)

Input: a SMM q (Eq. (SMM)) and sample budget S;
Output: S i.i.d. samples from q;
$\mathcal{X} \leftarrow \{\}$;
for $s \in \{1, ..., S\}$ **do**
  $\boldsymbol{x}^{(s)} \leftarrow \{\}$;
  for $d \in \{1, ..., D\}$ **do**
    $u \overset{\text{i.i.d.}}{\sim} \mathsf{Uniform}(0, 1)$;
    $x_d^{(s)} \leftarrow \mathsf{CDF}^{-1}(u \mid \boldsymbol{x}_{<d}^{(s)})$;
  $\mathcal{X} \leftarrow \mathcal{X} \cup \{\boldsymbol{x}^{(s)}\}$;
return $\mathcal{X}$

Algorithm 2: rejectionSampling(q, S)

Input: a decomposed SMM q (Eq. (ΔSMM)) and sample budget S;
Output: $K \leq S$ i.i.d. samples from q;
$M \leftarrow Z/Z_+$, $\mathcal{X} \leftarrow \{\}$;
for $s \in \{1, ..., S\}$ **do**
  $\boldsymbol{x}^{(s)} \overset{\text{i.i.d.}}{\sim} q_+$;
  $u \overset{\text{i.i.d.}}{\sim} \mathsf{Uniform}(0, 1)$;
  if $u \leq q(\boldsymbol{x}^{(s)})/(M \cdot q_+(\boldsymbol{x}^{(s)}))$: $\mathcal{X} \leftarrow \mathcal{X} \cup \{\boldsymbol{x}^{(s)}\}$;
return $\mathcal{X}$

We use complex mixture weights in our experiments (§6), as they have been shown to facilitate learning (Loconte et al., 2025b). However, our methodological results apply to general SMMs as expressed in Eq. (SMM). In the following section, we study how to estimate expectations over SMMs which is central to their application to both IS and VI.

3 EXPECTATION ESTIMATION WITH SMMS

We focus on the task of estimating an intractable integral I of the form

$$I = \mathbb{E}_{p(\boldsymbol{x})}[f(\boldsymbol{x})] = \int f(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} \tag{2}$$

where we have access to an unnormalized density $\widetilde{p}(\boldsymbol{x})$ and $f(\boldsymbol{x})$ is a quantity of interest for the distribution $p(\boldsymbol{x}) = \widetilde{p}(\boldsymbol{x})/Z_p$. We next study how to use a given SMM to estimate I via importance sampling using different sampling techniques. In §4, we then discuss how we can *learn* a proposal via black-box VI (§4), which relies on sampling as a subroutine.

3.1 How to sample SMMs?

To sample an additive MM (Eq. (1)), one can exploit its latent variable semantics and first sample a component q_k with probability proportional to α_k, then a full instantiation from the corresponding component, i.e., $\boldsymbol{x} \sim q_k$. Alg. 3 details this process, called *ancestral sampling* (Bishop and Nasrabadi, 2006), whose complexity is linear in the number of components K and dimensions D if we assume each component factorizes into independent marginals (see App. A.5). To further reduce the variance of the resulting estimator for Eq. (2), one can use *stratified sampling* (Owen, 2013, Alg. 4), which still leverages the latent variable interpretation. As the mixture coefficients in SMMs can no longer be interpreted as probabilities, neither Alg. 3 nor Alg. 4 can be directly used, and hence sampling from SMMs requires specialized techniques.

Auto-regressive inverse transform sampling (ARITS). For (squared) SMMs, whose components allow for tractable marginalization, one can use ARITS (Loconte et al., 2024; Cai et al., 2024a): Given a variable ordering, we can decompose the joint as $q_{\text{SMM}}(\boldsymbol{x}) = \prod_{i=1}^{D} q_{\text{SMM}}(x_i \mid \boldsymbol{x}_{<i})$ and sample each variable *sequentially* by inverting the corresponding conditional CDF (Alg. 1). This can be done up to a certain numerical precision ϵ, e.g., via binary search (Alg. 5). Due to its sequential nature, ARITS incurs an additional cost that is at least $\mathcal{O}(D)$ times greater than ancestral sampling for classical MMs (App. A.5). Furthermore, a non-negligible cost can come from inverting the CDF numerically. In our experiments (§6), we find ARITS to be well-behaved for $\epsilon = 10^{-6}$, but we can hardly scale it beyond $D = 32$. This prompts us to look for more scalable ways to sample q_{SMM} or *avoid sampling it directly.*

We start by noting that the general form of a SMM (Eq. (SMM)) can be rewritten as a ***difference of two additive MMs*** (Bignami and De Matteis, 1971; Robert and Stoehr, 2025), that is,

$$q_{\text{SMM}}(\boldsymbol{x}) = Z^{-1}\Big(Z_+ \cdot q_+(\boldsymbol{x}) - Z_- \cdot q_-(\boldsymbol{x})\Big), \quad (\Delta\text{SMM})$$

where $q_+(\boldsymbol{x}) = \widetilde{q}_+(\boldsymbol{x})/Z_+$, $q_-(\boldsymbol{x}) = \widetilde{q}_-(\boldsymbol{x})/Z_-$ are *additive* mixture PDFs composed of the positively and negatively weighted components of q_{SMM} respectively. Fig. 7 shows this decomposition for a squared SMM. $Z_+ = \int \widetilde{q}_+(\boldsymbol{x}) d\boldsymbol{x}$ and $Z_- = \int \widetilde{q}_-(\boldsymbol{x}) d\boldsymbol{x}$ are their normalizing constants, and $Z = Z_+ - Z_-$ is the normalizing constant of q_{SMM}, as before. Since q_+ and q_- are additive MMs, they are amenable to ancestral sampling (Alg. 3). This enables the design of more scalable approximate inference routines for SMMs.

Rejection sampling with SMMs. Eq. (ΔSMM) yields the bound $q_{\text{SMM}}(\boldsymbol{x}) \leq (Z_+/Z)\, q_+(\boldsymbol{x})$, enabling rejection sampling (RS) from q_+ (Bignami and De Matteis, 1971) (Alg. 2). A proposed sample $\boldsymbol{x} \sim q_+$ is accepted with probability $q_{\text{SMM}}(\boldsymbol{x})/(M\, q_+(\boldsymbol{x}))$, where $M = Z_+/Z$. The expected acceptance probability is $a = Z/Z_+$. This avoids auto-regressive sampling, but can suffer when a is small. While cleverer acceptance schemes can be devised (Robert and Stoehr, 2025), we found that the vanilla rejection sampling

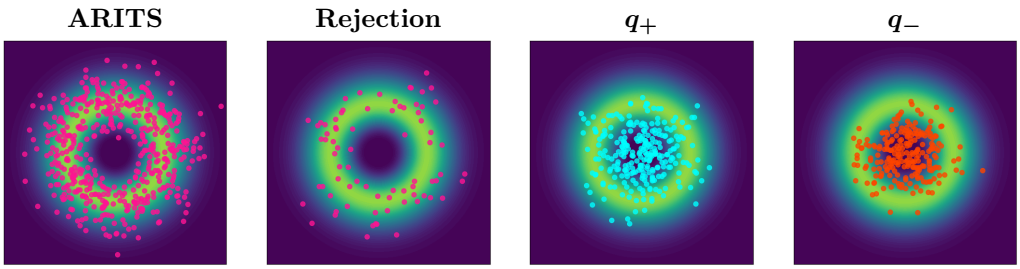


Figure 2: **Visual comparison of sampling strategies on a 2D SMM.** ARITS directly simulates samples from the ring. Rejection sampling discards many samples since the average acceptance probability in this example is only around 0.137. ΔIS uses samples from both positively and negatively weighted components (depicted in blue and red respectively) to estimate a difference of expectations. All methods are depicted with $S = 500$.

scales well and delivers good estimations in our experiments (§6). For our theoretical analysis, we consider a fixed-budget variant of RS with S proposed samples and denote the (random) number of acceptances by K. We analyze the variance when estimating $I = \mathbb{E}_{q_{\text{SMM}}}[f(\boldsymbol{x})]$ as $\widehat{I}_{\text{RS}} = \frac{1}{K}\sum_{k=1}^{K} h(\boldsymbol{x}^{(k)})$ where $\boldsymbol{x}^{(k)}$ are the accepted samples resulting from Alg. 2.

Proposition 1 (Variance of rejection sampling) *Assume* $\int h(\boldsymbol{x})^2 q_{\text{SMM}}(\boldsymbol{x})\, d\boldsymbol{x} < \infty$ *and* K *is a zero-truncated Binomial RV to avoid zero acceptances, i.e.,* $K \sim \text{TrBin}(S; a)$. *Then,*

$$\mathbb{V}_{\substack{\boldsymbol{x}\sim q_{SMM} \\ K\sim\text{TrBin}(S;a)}}\left[\widehat{I}_{RS}\right] = \mathbb{V}_{\boldsymbol{x}\sim q_{\text{SMM}}}[h(\boldsymbol{x})]\cdot\gamma(S,a),$$

where $\gamma(S,a) = \sum_{k=1}^{S}\frac{1}{k}\binom{S}{k}a^k(1-a)^{S-k}/(1-(1-a)^S)$.

The proof is in App. A.4. Note that RS is unbiased and consistent (Robert and Casella, 1999) and that direct i.i.d. sampling (no rejections) from q_{SMM} would have a scaling of $1/S$ deterministically and $\gamma(a,S) \geq 1/S$. As $a = Z/Z_+$ goes from 0 to 1, $\gamma(a,S)$ goes from 1 to $1/S$, impacting the MC convergence rate. Fig. 8 (Appendix) illustrates this. We next study an estimation scheme, that *avoids the rejection step.*

3.2 Importance Sampling with SMMs

It follows from Eq. (ΔSMM) that we can rewrite any expectation $\mathbb{E}_{q_{\text{SMM}}}[h(\boldsymbol{x})]$, for an absolutely integrable $h : \mathbb{R}^d \to \mathbb{R}$, into a *difference of expectations* as

$$Z_+/Z \cdot \mathbb{E}_{q_+}\left[h(\boldsymbol{x})\right] - Z_-/Z \cdot \mathbb{E}_{q_-}\left[h(\boldsymbol{x})\right]. \qquad (\Delta\text{EX})$$

When approximating $\mathbb{E}_{q_{\text{SMM}}}[h(\boldsymbol{x})]$, we can hence estimate a *weighted difference of expectations* w.r.t. to the additive MMs q_+ and q_- via (stratified) ancestral sampling, instead of sampling q_{SMM} directly. The ΔEX representation has been recently noted in Robert and Stoehr (2025) but, to the best of our knowledge, has not been studied for constructing scalable approximate inference schemes with SMMs, which we discuss next for *unnormalized IS* (UIS) (Owen, 2013): Given i.i.d. samples from a *proposal of choice* q, the UIS estimator approximates Eq. (2) as

$$\widehat{I}_{\text{UIS}} = \sum_{s=1}^{S} f(\boldsymbol{x}^{(s)})p(\boldsymbol{x}^{(s)})/q(\boldsymbol{x}^{(s)}), \quad \boldsymbol{x}^{(s)} \overset{\text{i.i.d.}}{\sim} q. \quad (\text{UIS})$$

The variance of the UIS estimator (Eq. (UIS)) is minimized when q closely matches the integrand, $|f|p/\int |f|p$. In our setting q is chosen to be a (squared) SMM. Both ARITS and rejection sampling can be used to realize Eq. (UIS) with an SMM proposal.

The ΔIS estimator. Alternatively, we can use Eq. (ΔEX) to derive a scalable IS estimator based on samples from q_+ and q_-, *without the need for a rejection criterion.* This leads to our ΔIS estimator:

$$\widehat{I}_{\Delta\text{IS}} = \frac{Z_+}{Z}\frac{1}{S_+}\sum_{s=1}^{S_+} f(\boldsymbol{x}_+^{(s)})\frac{p(\boldsymbol{x}_+^{(s)})}{q_{\text{SMM}}(\boldsymbol{x}_+^{(s)})} \qquad (\Delta\text{IS})$$
$$- \frac{Z_-}{Z}\frac{1}{S_-}\sum_{s=1}^{S_-} f(\boldsymbol{x}_-^{(s)})\frac{p(\boldsymbol{x}_-^{(s)})}{q_{\text{SMM}}(\boldsymbol{x}_-^{(s)})},$$
$$\text{with} \quad \boldsymbol{x}_+^{(s)} \overset{\text{i.i.d.}}{\sim} q_+(\boldsymbol{x}_+), \quad \boldsymbol{x}_-^{(s)} \overset{\text{i.i.d.}}{\sim} q_-(\boldsymbol{x}_-),$$

where $S = S_+ + S_-$ denotes the overall sampling budget and samples are drawn i.i.d. from q_+ and q_-. With ΔIS, we can sample both q_+ and q_- via ancestral sampling, without rejection, while still obtaining an unbiased and consistent estimator, as we show next.

Theorem 1 (Properties of ΔIS) *Under mild assumptions, the ΔIS estimator has the following properties. See App. B.1 for proofs.*

Unbiasedness and strong consistency: *ΔIS is unbiased, i.e.,* $\mathbb{E}_{\substack{\boldsymbol{x}_+\sim q_+ \\ \boldsymbol{x}_-\sim q_-}}[\widehat{I}_{\Delta IS}] = I$, *and it is strongly consistent,* $\mathbb{P}_{\substack{\boldsymbol{x}_+\sim q_+ \\ \boldsymbol{x}_-\sim q_-}}(\lim_{\substack{S_+\to+\infty \\ S_-\to+\infty}} \widehat{I}_{\Delta IS} = I) = 1$.

Variance. *The variance of* $\widehat{I}_{\Delta IS}$ *is given by*

$$\mathbb{V}[\widehat{I}_{\Delta IS}] = \frac{Z_+^2}{Z^2}\frac{1}{S_+}\mathbb{V}_{q_+}[f(\boldsymbol{x})w(\boldsymbol{x})] + \frac{Z_-^2}{Z^2}\frac{1}{S_-}\mathbb{V}_{q_-}[f(\boldsymbol{x})w(\boldsymbol{x})],$$

where $w(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{q_{SMM}(\boldsymbol{x})}$.

Optimal proposal. *The SMM proposal minimizing the variance is equivalent to the optimal UIS proposal,*

$$q^{\star}(\boldsymbol{x}) = \arg\min_{q} \mathbb{V}_{\substack{\boldsymbol{x}_+ \sim q_+ \\ \boldsymbol{x}_- \sim q_-}}[\widehat{I}_{\Delta IS}] = \frac{|f(\boldsymbol{x})|p(\boldsymbol{x})}{\int |f(\boldsymbol{x})|p(\boldsymbol{x})d\boldsymbol{x}}.$$

Further, $\mathbb{V}_{\substack{\boldsymbol{x}_+ \sim q_+ \\ \boldsymbol{x}_- \sim q_-}}\left[\widehat{I}_{\Delta IS}\right] = 0$ *if and only if* $q = q^{\star}$ *and* $f(\boldsymbol{x}) \geq 0$ *almost everywhere (or* $f(\boldsymbol{x}) \leq 0$*).*

A safer ΔIS. Although the optimal proposal of ΔIS matches UIS, their variances differ in general. This can lead to noticeable differences in estimation quality given the same proposal. To see why, consider that ΔIS samples from regions where q_+ and q_- are large individually even when $q_{\text{SMM}} \approx 0$, which can induce very large importance weights and result in high variance (see Fig. 2). To stabilize estimates, we propose to add a *"safe component"* q_{safe} to the proposal to guarantee non-negligible mass where p has support:

$$q(\boldsymbol{x}) = (1-\beta)\, q_{\text{SMM}}(\boldsymbol{x}) + \beta\, q_{\text{safe}}(\boldsymbol{x}), \quad \beta \in [0,1). \tag{3}$$

A "flat" q_{safe} effectively fills low-density valleys of the SMM. This integrates naturally with ΔIS by treating q_{safe} as part of q_+. In spirit, this mirrors the safe adaptive IS (SAIS) literature that mixes a heavy-tailed density into IS proposals (Owen and Zhou, 2000; Delyon and Portier, 2021; Korba and Portier, 2022).

A stratified ΔIS. To further reduce the variance of ΔIS, we make use of stratified sampling (Alg. 4) (Owen, 2013; Elvira et al., 2019), as opposed to standard ancestral sampling (Alg. 3). Moreover, for a sampling budget S, we heuristically fix the sample size for q_+ and q_- as $S_+ = \left\lfloor \frac{Z_+}{Z_+ + Z_-} S \right\rfloor$ and $S_- = \left\lfloor \frac{Z_-}{Z_+ + Z_-} S \right\rfloor$ to sample from the mixtures in proportion to their relative contribution to the estimator. One could *correlate* samples from q_+ and q_- for additional variance reduction; we leave this technique for future work.

4 BLACK-BOX VI WITH SMMS

We now investigate how we can use the IS estimators we designed in §3.2—ARITS, rejection, and ΔIS—as subroutines for black-box VI (BBVI). From now on, we will simply refer to the SMM proposal as q_θ as a shorthand, unless confusing. The parameter θ encompasses all learnable parameters of the SMM, i.e., the mixture weights as well as the parameters of the components. The main objective of BBVI is to find the optimal surrogate q_{θ^*} that minimizes a given divergence $L(q_\theta, p)$, which often can be written as an expectation $\mathbb{E}_{q_\theta}[\ell(\boldsymbol{x};\theta)]$, for a given loss $\ell(\boldsymbol{x};\theta)$ During learning, *gradient estimators* are used to approximate $\nabla_\theta \mathbb{E}_{q_\theta}[\ell(\boldsymbol{x};\theta)]$ via MC (Mohamed et al., 2020). We will focus on the commonly used reverse KL divergence (RKL) for which $\ell(\boldsymbol{x};\theta) = \log(q_\theta(\boldsymbol{x})/p(\boldsymbol{x}))$, but note that our results can be extended to other objectives of the form $\mathbb{E}_{q_\theta}[\ell(\boldsymbol{x};\theta)]$. Crucially, *squaring* the SMM allows us to retain a valid PDF during optimization without introducing constraints on the model parameters. We discuss next how well-studied gradient estimators for BBVI can be combined with the expectation estimators for SMMs defined in §3.

The **REINFORCE** (or *score function*) estimator (Glynn, 1986; Williams, 1992) relies on the log-derivative trick, i.e., $\nabla_\theta q_\theta(\boldsymbol{x}) = q_\theta(\boldsymbol{x}) \nabla_\theta \log q_\theta(\boldsymbol{x})$, to construct an unbiased estimator for $\nabla_\theta \mathbb{E}_{q_\theta}[\ell(\boldsymbol{x};\theta)]$. We use REINFORCE with a *leave-one-out control variate* (RLOO) for variance reduction (Salimans and Knowles, 2014; Kool et al., 2019). For the RKL and S samples $\boldsymbol{x}^{(s)} \overset{\text{i.i.d.}}{\sim} q_\theta$, this estimator is given as

$$\widehat{\nabla}_\theta^{\text{RLOO}} \text{KL}(q_\theta||p) = \frac{1}{S}\sum_{s=1}^{S}\Big[\log\Big(\frac{q_\theta(\boldsymbol{x}^{(s)})}{\widetilde{p}(\boldsymbol{x}^{(s)})}\Big) - \frac{1}{S-1}\sum_{l\neq s}\log\Big(\frac{q_\theta(\boldsymbol{x}^{(l)})}{\widetilde{p}(\boldsymbol{x}^{(l)})}\Big)\Big]\nabla_\theta \log q_\theta(\boldsymbol{x}^{(s)}), \tag{RLOO}$$

Notably, his estimator does not require the unknown expectation to have a differentiable integrand.

ΔVI. Another option is to obtain a *pathwise gradient estimator* using the reparameterization trick, which might come with lower variance than naive REINFORCE (Kingma and Welling, 2014; Rezende et al., 2014). We show that, following the approach of Morningstar et al. (2021) for classical MMs, this is also possible for SMMs, once represented as ΔSMM. In particular, since both q_+ and q_- are additive mixtures, we can apply *stratification* within each mixture, which results in a fully reparameterizable sampling scheme for q_+ and q_- respectively (Morningstar et al., 2021). For the RKL, the objective is

$$\text{KL}(q_\theta||p) = \sum_{k=1}^{K}\frac{\alpha_k Z_k}{Z}\mathbb{E}_{q_k}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right], \tag{ΔVI}$$

where q_k denote the individual components of the SMM[1] (cf. Eq. (SMM)) and $Z_k = \int \widetilde{q}_k(\boldsymbol{x})d\boldsymbol{x}$. See App. B.2 for the full derivation. We refer to the above objective as ΔVI to highlight the possibility of negative coefficients combining the individual expectations, which differentiates it from the stratified ELBO (SELBO) for additive mixture models described by Morningstar et al. (2021). While we apply this treatment to the RKL, we point out that similar estimators for SMMs are possible for other losses expressed as $\mathbb{E}_{q_\theta}[\ell(\boldsymbol{x};\theta)]$, such as the Fisher divergence (Yang et al., 2019; Cai et al., 2024b). We provide an empirical comparison of the discussed VI strategies in §6.

[1] For a squared SMM, q_k denotes a product of two components (Eq. (SMM^2)).

5 RELATED WORK

Classical MMs have been used extensively in (black-box) VI. Additionally, there has been work on learning MMs with a variety of divergences beyond KL (Ryu, 2016; El-Laham et al., 2020; Lambert et al., 2022; Daudel et al., 2023). IS methods often employ mixture proposals, or equivalently a collection of proposals that can be interpreted as a mixture; this approach is known as multiple importance sampling (MIS; Veach and Guibas, 1995; Owen and Zhou, 2000; Elvira et al., 2019; Sbert et al., 2018). MIS methods have been widely applied in graphics (Sbert et al., 2018; Kondapaneni et al., 2019; Müller et al., 2019). Learning sampling mixtures is the basis of many adaptive importance sampling (AIS) algorithms, e.g., by resampling (Cappé et al., 2004; Elvira et al., 2017), via expectation-maximization (Cappé et al., 2008), with MCMC to adapt the proposals (Martino et al., 2017), or exploiting geometry of the target (Fasiolo et al., 2018; Elvira and Chouzenoux, 2022) (see Bugallo et al. (2017) for a review). Our work thus contributes to many recent works connecting VI and AIS (Yao et al., 2018; Domke and Sheldon, 2018; Finke and Thiery, 2019; Jerfel et al., 2021; Guilmeau et al., 2024). Further, ΔIS can be seen as a linear combination of two MC estimators that allows for negative coefficients. Similar constructions have been studied to combine a set of given estimators in the context of multiple IS (Kondapaneni et al., 2019) and multilevel MC (Giles, 2015), but not used as subroutines for learning a SMM. Concurrently to us, Martino (2025) studies sampling from SMMs, and derives different estimators, but not in a VI context.

MMs can further be extended to *deep hierarchical mixtures*, also referred to as *probabilistic circuits* (PCs; Choi et al., 2020). Monotonic PCs, i.e., deep additive MMs, have been investigated for VI in specialized settings, such as inference in discrete graphical models (Shih and Ermon, 2020), quantized continuous distributions (Sladek et al., 2025) and hierarchical mixtures of VAEs (Tan and Peharz, 2019).

Closer to our work, Cai et al. (2024a) recently proposed *EigenVI*, a VI method that learns squared SMMs with orthogonal basis functions as components. EigenVI optimizes the Fisher divergence (Hyvärinen and Dayan, 2005) between the surrogate and target by solving an eigenvalue problem, which sidesteps stochastic gradient-based optimization. Learning with EigenVI is fast, but the components are not learnable and the approach currently does not allow for alternative objectives. Moreover, inverting the CDF of the orthogonal basis is harder than doing so for Gaussian components, which we use in our experiments.

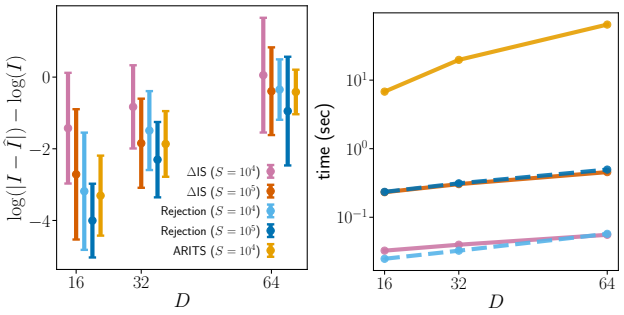


Figure 3: **Rejection sampling and ΔIS can achieve comparable estimation quality to ARITS when given sufficient sampling budget, but can be orders of magnitude faster** in high dimensions as shown for MC estimation. We depict (mean ± stddev) over 30 instances. Details in App. C.2.

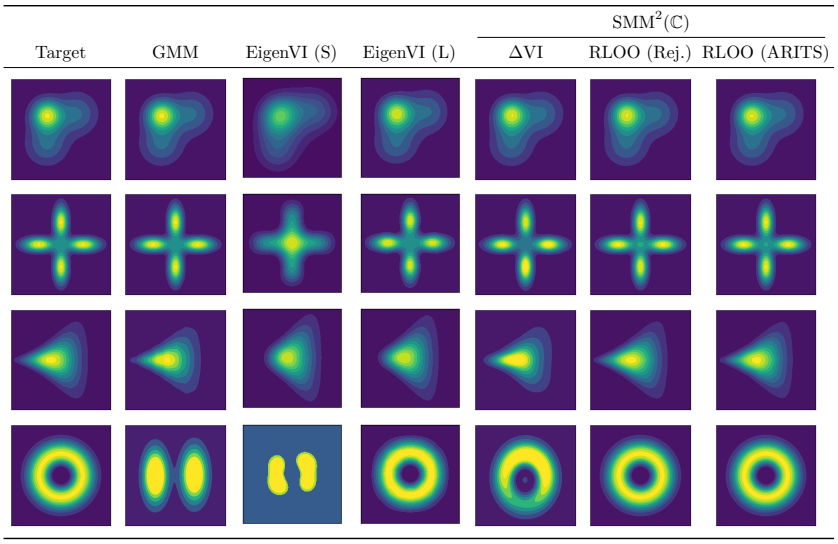
6 EXPERIMENTS

We now empirically assess how the estimators discussed in this paper perform in three settings: *vanilla MC*, *BBVI*, and *IS with learned proposals*. We explore the following research questions: (**RQ 1**) How does the approximation quality and runtime of different expectation estimation strategies compare? (**RQ 2**) How do our VI strategies for SMMs compare and what are prominent challenges in comparison to VI with classical MMs? (**RQ 3**) How can we perform effective IS with SMMs? And lastly, (**RQ 4**) Can we use SMMs to model challenging neuro-symbolic targets?

Throughout the experiments, we choose q_{SMM} as a squared SMM with complex weights (§2) and zero-covariance Gaussian components. We use ARITS with tolerance $\epsilon = 10^{-6}$ (Alg. 1). The target densities are defined in App. C.1.

RQ1) Scaling sampling with SMMs. First, we illustrate how ARITS, rejection sampling, and ΔIS trade off estimation quality with execution time. We estimate $I = \mathbb{E}_{q_{\text{SMM}}}[f(\boldsymbol{x})]$ via *standard MC* where we choose $f(\boldsymbol{x})$ as the density of an unnormalized GMM, which allows us to compute the ground-truth expectation in closed form. We generate 30 combinations of q_{SMM} and f, and average the results. We measure the estimation error as $\log(|\widehat{I} - I|) - \log(I)$ and report the runtime in seconds. App. C.2 provides further details. Fig. 3 summarizes the results for SMMs with 6 components (before squaring). App. D.1 discusses experiments with other values of K. Our main insight is the following: Both ΔIS and rejection sampling can achieve similar performance to ARITS given sufficient sampling budget while being much faster in terms of runtime. Having scalable alternatives to ARITS is crucial for learning in higher dimensional settings.

Table 1: Our BBVI methods ΔVI and RLOO with rejection and ARITS can recover differently shaped 2D targets while being more parameter-efficient than EigenVI (Cai et al., 2024a). Table 9 in App. D.2.1 reports the number of learnable parameters for each model and the corresponding FKL values (which can also be found in Table 2 for our VI variants). We note that even when fitted densities look similar between SMMs and GMMs, the learned components can greatly differ, see Fig. 5 for an example.



RQ2) Comparing VI strategies for SMMs. We start by comparing the the quality of variational approximations achieved with the three VI strategies discussed in §3 against EigenVI (**RQ2.1**), adopting their two-dimensional densities (Cai et al., 2024a) to which we add a ring-shaped density, on which we also investigate the effect of sampling budget size for learning (**RQ2.2**). Then, we use further synthetic SMM targets to test whether our proposed VI strategies manage to learn negative parameters in high-dimensional settings, given that the target has prominent holes (**RQ2.3**). Lastly, we test SMMs on standard Bayesian logistic regression targets, as used in recent benchmarks (Blessing et al., 2024) (**RQ2.4**). We include a standard GMM baseline with zero-covariance components (matching the SMMs), learned with the SELBO (Morningstar et al., 2021) in all experiments. Further details on the targets and setup can be found in App. C.1 and App. C.3 respectively.

RQ2.1) EigenVI. Tab. 1 visualizes our results and Table 9 in App. D.2.1 reports the the forward KL (FKL) between the model and the target. For EigenVI we use two model sizes: one comparable to the number of learnable parameters of our models (S) and the largest model used in Cai et al. (2024a) (L). Overall, for the same number of learnable parameters, our estimators for SMMs perform similarly or better than EigenVI. On *GMM4*, as well as the *Funnel*, squared SMMs obtain comparable metrics to additive GMMs. The difference in fit is clearly visible for the *Ring* density as an additive GMM with two components is not expressive enough (Tab. 1). But also for the *GMM4* target, which can be captured by both models, learned components can differ (see Fig. 5). Using ARITS with the RLOO gradient estimator is computationally feasible and gives consistently good results, and RLOO with rejection sampling similarly works well on all targets, while ΔVI seems to lag behind.

RQ2.2) Influence of sampling budget. We further analyze the behavior of our VI methods by varying the sampling budget S during training. Fig. 4 shows how this impacts the RKL and FKL on the *Ring* target. We find (1) the RLOO variants perform well even for a substantially smaller sampling budget than the 10^5 samples used for Tab. 1 and (2) ΔVI needs a higher sampling budget to perform similarly to RLOO. Interestingly, at 10^6 samples per update step ΔVI manages to capture a ring-like shape more consistently than the RLOO variants. See App. D.2.2 for a visualization of the models used for Fig. 4.

RQ2.3) Higher-dimensional SMM targets. Next, we test our VI schemes on higher-dimensional targets that require learning negative mixture weights. To this end, we generate *Hollow* spheres, defined as squared SMMs of varying dimensionality with a substantial influence of negative components (see App. C.1 for their definitions). Rejection sampling on the resulting targets yields a rejection rate between 80% and 90%. The aim of this setup is to understand whether we can effectively learn to subtract density with our BBVI methods. Our main insights match the two-dimensional targets: Both RLOO variants manage to recover the shape well while the ΔVI models are not up to par. Interestingly, we noticed that optimization was very sensitive to the initialization across all VI variants. See App. D.2.3 for examples of varying initializations and the resulting models. How to design robust initializations for BBVI with SMMs is an important open question.

RQ2.4) Standard high-dimensional targets. We consider a 10-dimensional *Funnel* (Neal, 2003) and various Bayesian logistic regression (BLR) posteriors,

Table 2: On densities with more prominent holes such as *Ring* and *Hollow*, our ΔVI and RLOO variants with SMMs deliver better performance than classical GMMs, while being comparable on other densities that do not necessarily require to subtract probability mass. See Table 1 for visual examples of the two-dimensional fits.

	GMM		SMM + ΔVI		SMM + RLOO (Rej.)		SMM + RLOO (ARITS)	
Target (D)	RKL (↓)	FKL (↓)	RKL (↓)	FKL (↓)	RKL (↓)	FKL (↓)	RKL (↓)	FKL (↓)
GMM3 (2)	$1.9 \cdot 10^{-6} \pm 1.3 \cdot 10^{-5}$	$6.0 \cdot 10^{-6} \pm 1.6 \cdot 10^{-5}$	$2.4 \cdot 10^{-4} \pm 6.8 \cdot 10^{-5}$	$2.0 \cdot 10^{-4} \pm 8.4 \cdot 10^{-5}$	$1.8 \cdot 10^{-4} \pm 6.7 \cdot 10^{-5}$	$2.3 \cdot 10^{-4} \pm 5.4 \cdot 10^{-5}$	$2.7 \cdot 10^{-4} \pm 6.8 \cdot 10^{-5}$	$2.6 \cdot 10^{-4} \pm 9.1 \cdot 10^{-5}$
GMM4 (2)	$1.4 \cdot 10^{-5} \pm 1.3 \cdot 10^{-5}$	$1.9 \cdot 10^{-5} \pm 9.8 \cdot 10^{-6}$	$5.2 \cdot 10^{-3} \pm 4.0 \cdot 10^{-4}$	$5.5 \cdot 10^{-3} \pm 2.9 \cdot 10^{-4}$	$1.1 \cdot 10^{-4} \pm 4.1 \cdot 10^{-5}$	$1.1 \cdot 10^{-4} \pm 4.8 \cdot 10^{-5}$	$5.8 \cdot 10^{-5} \pm 4.1 \cdot 10^{-5}$	$7.4 \cdot 10^{-5} \pm 3.9 \cdot 10^{-5}$
Funnel (2)	$3.5 \cdot 10^{-3} \pm 3.0 \cdot 10^{-4}$	$3.7 \cdot 10^{-3} \pm 3.5 \cdot 10^{-4}$	$2.7 \cdot 10^{-2} \pm 1.7 \cdot 10^{-3}$	$4.1 \cdot 10^{-2} \pm 8.3 \cdot 10^{-4}$	$8.4 \cdot 10^{-4} \pm 1.8 \cdot 10^{-4}$	$8.2 \cdot 10^{-4} \pm 1.9 \cdot 10^{-4}$	$1.1 \cdot 10^{-3} \pm 1.8 \cdot 10^{-4}$	$1.3 \cdot 10^{-3} \pm 1.7 \cdot 10^{-4}$
Ring (2)	$2.9 \cdot 10^{-1} \pm 2.3 \cdot 10^{-3}$	$3.2 \cdot 10^{-1} \pm 2.2 \cdot 10^{-3}$	$8.2 \cdot 10^{-2} \pm 1.3 \cdot 10^{-3}$	$8.4 \cdot 10^{-2} \pm 1.2 \cdot 10^{-3}$	$9.3 \cdot 10^{-6} \pm 1.6 \cdot 10^{-5}$	$1.2 \cdot 10^{-5} \pm 1.7 \cdot 10^{-5}$	$5.5 \cdot 10^{-6} \pm 1.6 \cdot 10^{-5}$	$2.0 \cdot 10^{-6} \pm 1.0 \cdot 10^{-5}$
Hollow (16)	$2.8 \cdot 10^{-1} \pm 3.3 \cdot 10^{-3}$	$1.8 \cdot 10^{-1} \pm 1.9 \cdot 10^{-3}$	$2.3 \cdot 10^{-1} \pm 2.8 \cdot 10^{-3}$	$2.0 \cdot 10^{-1} \pm 2.2 \cdot 10^{-3}$	$3.7 \cdot 10^{-5} \pm 2.6 \cdot 10^{-5}$	$2.6 \cdot 10^{-5} \pm 1.8 \cdot 10^{-5}$	$2.2 \cdot 10^{-5} \pm 2.0 \cdot 10^{-5}$	$3.1 \cdot 10^{-5} \pm 3.8 \cdot 10^{-5}$
Hollow (32)	$1.9 \cdot 10^{-1} \pm 2.2 \cdot 10^{-3}$	$1.4 \cdot 10^{-1} \pm 1.8 \cdot 10^{-3}$	$1.8 \cdot 10^{-1} \pm 1.9 \cdot 10^{-3}$	$2.0 \cdot 10^{-1} \pm 1.7 \cdot 10^{-3}$	$1.8 \cdot 10^{-5} \pm 2.3 \cdot 10^{-5}$	$2.8 \cdot 10^{-5} \pm 2.4 \cdot 10^{-5}$	$2.5 \cdot 10^{-5} \pm 1.7 \cdot 10^{-5}$	$1.8 \cdot 10^{-5} \pm 1.8 \cdot 10^{-5}$
Hollow (64)	$2.3 \cdot 10^{-1} \pm 3.3 \cdot 10^{-3}$	$1.4 \cdot 10^{-1} \pm 1.7 \cdot 10^{-3}$	$2.3 \cdot 10^{-1} \pm 2.7 \cdot 10^{-3}$	$2.1 \cdot 10^{-1} \pm 1.1 \cdot 10^{-3}$	$7.6 \cdot 10^{-5} \pm 2.7 \cdot 10^{-5}$	$6.8 \cdot 10^{-5} \pm 2.9 \cdot 10^{-5}$	/	/
Funnel (10)	$7.9 \cdot 10^{-2} \pm 1.3 \cdot 10^{-3}$	$1.0 \cdot 10^{0} \pm 3.5 \cdot 10^{-1}$	$3.0 \cdot 10^{-1} \pm 1.5 \cdot 10^{-3}$	$2.6 \cdot 10^{1} \pm 5.3 \cdot 10^{0}$	$2.7 \cdot 10^{-1} \pm 1.4 \cdot 10^{-3}$	$1.9 \cdot 10^{1} \pm 2.9 \cdot 10^{0}$	/	/

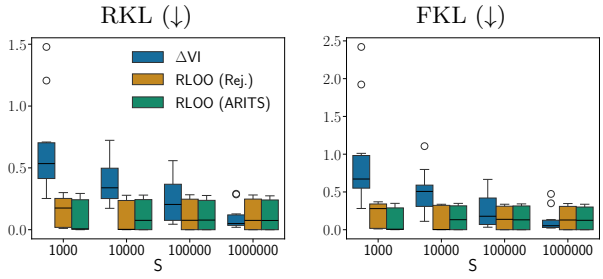


Figure 4: ΔVI requires a higher number of samples than RLOO variants to achieve comparable RKL and FKL. The RKL and FKL values were collected from 10 models learned with a budget of S samples per step.

Table 3: On BLR posteriors, GMMs and SMMs perform similarly in terms of ELBO.

Density	GMM	SMM + ΔVI	SMM + RLOO (Rej.)
credit	−1.72±0.0231	−1.74±0.0102	−1.71±0.01
ionosphere	−124±0.0476	−124±0.0382	−124±0.0295
breastcancer	−67.3±0.0358	−67.6±0.0311	−67.5±0.0335
sonar	−137±0.0323	−138±0.0345	−138±0.0328

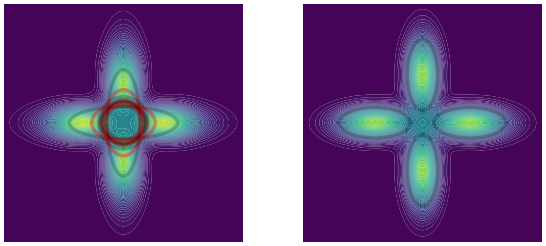


Figure 5: When fitting the same density (*GMM4* target, Table 1), SMMs (left) and classical GMMs (right) behave differently as shown by how they place additive (gray) and subtractive (red) Gaussian components.

which are very common benchmarks in statistics and VI. In particular, for BLR we use the datasets *GermanCredit*, *BreastCancer*, *Ionosphere* and *Sonar* from Blessing et al. (2024). In Tab. 3, we report estimated ELBOs for RLOO (with rejection), ΔVI, and the GMM baseline. We do not report RLOO (ARITS) due to it exceeding reasonable runtimes. On these targets, all methods perform similarly. Plots of bivariate conditionals on a grid of values suggest that these posteriors may be similar to Gaussians, which is also a fact that has been noted in the Bayesian statistics literature (Chopin and Ridgway, 2017). For these posteriors, we noticed that it was difficult, and very dependent on initialization, to obtain non-negligible negative contributions $Z_-/(Z_+ - Z_-)$. See App. C for details about hyperparameters and learned models. For the 10-dimensional *Funnel*, we obtain better approximations with GMMs than with SMMs. Empirically, we found the SMM components to closely cluster together, resulting in fits that did not cover the spread-out funnel shape well.

RQ3) IS with SMMs. We test the expectation estimation strategies discussed in §3 for *normalizing constant estimation:* Given the unnormalized target density $\widetilde{p}(\boldsymbol{x})$, we aim to estimate $I = \int \widetilde{p}(\boldsymbol{x}) d\boldsymbol{x}$ via an IS estimator based on a proposal q_{SMM}. In this setting, we first show with synthetic proposals how ΔIS can result in high variance compared to the standard UIS estimator and how a safe component can mitigate this issue in practice (**RQ3.1**). We then use (**RQ3.2**) *learned* SMM proposals from from the previous RQ, and compare the resulting estimation quality with ARITS, rejection, ΔIS, and IS estimators using a standard GMM proposal.

RQ3.1) Safe ΔIS. As we show in App. D.3.1, ΔIS can result in high variance when used on targets with high negative contribution, such as the *Ring* and *Hollow* targets. We mitigate this by mixing the proposal with a safe component (Eq. (3)). We choose q_{safe} as a flat Gaussian $\mathcal{N}(0, \sigma^2_{\text{safe}} I^{D\times D})$, where $I^{D\times D}$ is the D-dimensional identity and σ_{safe} is a hyperparameter that we set heuristically. Even with a small mixing coefficient β, a safe component can substantially reduce the variance of ΔIS. How to automatically construct a safe component for a given problem is an interesting open question.

RQ3.2) IS with learned proposals. Lastly, we complete a full SMM-based approximate inference

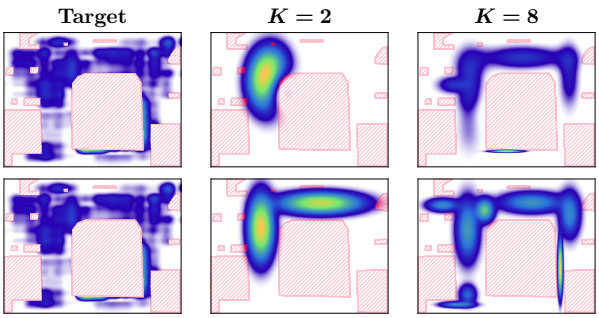


Figure 6: A second scenario from the SDD dataset. **SMMs (top)** and **GMMs (bottom)** result in very different fits to the target for the same component budget. We once again observe that for $K = 2$, the SMM effectively learns to use subtraction in order to model the absence of density induced by a constraint.

pipeline and use learned SMM proposals for normalizing constant estimation. We use proposals learned via rejection from Tab. 2 and compare against IS with learned GMMs. For ΔIS, we again choose the safe component as $\mathcal{N}(0, \sigma^2_{\text{safe}} I^{D\times D})$. We select σ_{safe} and β via grid search based on the empirical variance (see App. D.3.2 for details). Tab. 11 summarizes the results and Fig. 12 provides boxplots based on different sample sizes. The results mirror the previous RQs: when negative components are useful to represent a target density, ARITS performs the best and rejection sampling is a scalable alternative. When targets do not require subtraction, GMMs give better estimates but SMMs do not lag too much behind.

RQ4) SMMs for learning under constraints. Lastly, we evaluate SMMs on a real-world scenario where they are intuitive and effective VI proposals. In particular, we consider a *neuro-symbolic* setting, where inputs violating domain constraints results in 0-density. It is natural to model such densities with SMMs, as they can learn to "cut" the density of invalid areas. Consider the targets in Figs. 1 and 6 fit with the *probabilistic algebraic layer* (PAL) (Kurscheidt et al., 2025) to the *Stanford Drone Dataset* (SDD) (Robicquet et al., 2016), which captures the walkable area of a map. The densities are naturally constrained due to obstacles, such as a roundabout and buildings. These constraints were recently manually annotated by Kurscheidt et al. (2025), resulting in challenging benchmark densities. We find that SMMs trained with RLOO + rejection can learn to use subtraction to approximately adhere to some of the domain constraints, even at a low component budget. However, we also find that general difficulties in fitting SMMs observed in the previous subsections persist: Akin to the 10-dimensional funnel, we find it difficult to learn SMMs that cover the full target, even at high component budgets. See App. D.5 for experimental details, results for additional values of K, and a quantitative comparison to GMMs in terms of ELBO. Using SMMs to model domain constraints is a promising future direction. It will be interesting to explore in future work how SMMs can be used in rare-event probability estimation in other safety-critical systems, such as aircraft collision avoidance (Corso et al., 2021) and power grids (Owen et al., 2019), since these problems all involve domain constraints.

Main insights. We conclude §6 with a summary of the main insights obtained throughout the experiments. **(I1)** On the tested targets, rejection sampling is a surprisingly effective alternative to ARITS and performs well for both VI and IS. **(I2)** ΔIS on the other hand does not reach the same performance as rejection and ARITS and the same holds for ΔVI. While the safe component can help to mitigate the variance of ΔIS, effectively using ΔIS in practice will require future work on further reducing its variance. **(I3)** When a density has prominent "holes" (such as the *Ring* and *Hollow* targets), we observe a clear benefit of using SMMs, both in terms of fit to the target and normalizing constant estimation. **(I4)** Lastly, while SMMs show promising initial results, there are several open learning challenges, including finding robust initializations for VI with SMMs, and ways to effectively fit squared SMMs with a high number of components, despite parameter sharing across components.

7 DISCUSSION

In this work, we laid the foundations to effectively use SMMs for IS and VI, overcoming the lack the latent variable interpretation of classical additive MMs. To this end, we introduced and compared several estimators based on three key sampling routines—ARITS, rejection sampling and ΔIS—highlighting in theory and in practice the induced trade-offs in terms of accuracy of estimation, scalability, and statistical stability. Gaining a deeper theoretical understanding of the optimization challenges that negative components and the squaring operation bring will be crucial for further stabilizing approximate inference with SMMs.

Future work. We leave for future work the exploration of self-normalized importance sampling (Owen, 2013) and particle filtering (the latter can be viewed as IS with mixture proposals, (Li et al., 2016; Branchini and Elvira, 2025)) with SMMs. Moreover, we aim to connect our findings to the literature of tensor factorizations and networks, where negative parameters are commonly used (Loconte et al., 2025a). We further plan to investigate how to adapt the techniques we proposed for shallow SMMs to the general deep PCs (Vergari et al., 2019; Choi et al., 2020).

Acknowledgments

NB acknowledges support from the ProbAI Hub. LZ and AV were supported by the "UNREAL: Unified Reasoning Layer for Trustworthy ML" project (EP/Y023838/1) selected by the ERC and funded by UKRI EPSRC. LDS was supported by the Internal Funds KU Leuven (projects iBOF/21/075 and PDMT2/25/057). He also acknowledges support from the Flemish Government (AI Research Program) and Una Europa. NM acknowledges support from the CIFAR Learning in Machines and Brains Programme. We are grateful to the april lab for valuable feedback, in particular Lorenzo Loconte for useful discussion on the implementation of ARITS and the `cirkit` library, Adrián Javaloy for early helpful discussions around SMMs, and Leander Kurscheidt for providing the neuro-symbolic targets.

Contributions. NB suggested using the difference representation of SMMs for IS and, together with LZ, developed the theory of ΔIS, including its formalization, proofs, and derivations. LZ led experiments with help from NB and LDS, who respectively suggested benchmarks and provided GPU-accelerated implementations of ΔVI and other baselines. LZ also identified the instability of ΔIS, proposed a safe component to address it, and suggested the stratified variant. NM and VE provided useful discussion and feedback on the manuscript. AV supervised all the stages of the project.

References

Anna Bignami and A. De Matteis. A note on sampling from combinations of distributions. *IMA Journal of Applied Mathematics*, 8(1):80–81, 1971. ISSN 0272-4960. doi: 10.1093/imamat/8.1.80. URL https://doi.org/10.1093/imamat/8.1.80.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Denis Blessing, Xiaogang Jia, Johannes Esslinger, Francisco Vargas, and Gerhard Neumann. Beyond elbos: A large-scale evaluation of variational methods for sampling. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=fVg9YrSllr.

Nicola Branchini and Víctor Elvira. An adaptive mixture view of particle filters. *Foundations of Data Science*, 7(4), 2025. doi: 10.3934/fods.2024017.

Monica F Bugallo, Victor Elvira, Luca Martino, David Luengo, Joaquin Miguez, and Petar M Djuric. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017.

Alberto Cabezas, Adrien Corenflos, Junpeng Lao, Rémi Louf, Antoine Carnec, Kaustubh Chaudhari, Reuben Cohn-Gordon, Jeremie Coullon, Wei Deng, Sam Duffield, et al. Blackjax: composable bayesian inference in jax. *arXiv preprint arXiv:2402.10797*, 2024.

Diana Cai, Chirag Modi, Charles Margossian, Robert M. Gower, David M. Blei, and Lawrence K. Saul. Eigenvi: score-based variational inference with orthogonal function expansions. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a.

Diana Cai, Chirag Modi, Loucas Pillaud-Vivien, Charles Margossian, Robert M. Gower, David M. Blei, and Lawrence K. Saul. Batch and match: black-box variational inference with a score-based divergence. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=bplNmU2ROC.

Olivier Cappé, Arnaud Guillin, Jean-Michel Marin, and Christian P Robert. Population monte carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

Olivier Cappé, Randal Douc, Arnaud Guillin, Jean-Michel Marin, and Christian P Robert. Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18:447–459, 2008.

George Casella and Christian P Robert. Post-processing accept-reject samples: recycling and rescaling. *Journal of Computational and Graphical Statistics*, 7(2):139–157, 1998.

Y Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA. URL: http://starai. cs. ucla. edu/papers/ProbCirc20. pdf*, page 6, 2020.

Nicolas Chopin and James Ridgway. Leave Pima Indians Alone: Binary Regression as a Benchmark for Bayesian Computation. *Statistical Science*, 32 (1):64 – 87, 2017. doi: 10.1214/16-STS581. URL https://doi.org/10.1214/16-STS581.

Anthony Corso, Robert Moss, Mark Koren, Ritchie Lee, and Mykel Kochenderfer. A survey of al-

gorithms for black-box safety validation of cyber-physical systems. *Journal of Artificial Intelligence Research*, 72:377–428, 2021.

Kamélia Daudel, Randal Douc, and Franccois Roueff. Monotonic alpha-divergence minimisation for variational inference. *Journal of Machine Learning Research*, 24(62):1–76, 2023.

Bernard Delyon and François Portier. Safe adaptive importance sampling: A mixture approach. *The Annals of Statistics*, 49(2):885–917, 2021.

Justin Domke and Daniel R. Sheldon. Importance weighting and variational inference. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4475–4484, 2018.

Yousef El-Laham, Petar M. Djuric, and Mónica F. Bugallo. Enhanced mixture population monte carlo via stochastic optimization and markov chain monte carlo sampling. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 5475–5479. IEEE, 2020. doi: 10.1109/ICASSP40776.2020.9053410. URL https://doi.org/10.1109/ICASSP40776.2020.9053410.

Víctor Elvira and Emilie Chouzenoux. Optimized population monte carlo. *IEEE Transactions on Signal Processing*, 70:2489–2501, 2022.

Víctor Elvira, Luca Martino, David Luengo, and Mónica F Bugallo. Improving population monte carlo: Alternative weighting and resampling schemes. *Signal Processing*, 131:77–91, 2017.

Víctor Elvira, Luca Martino, David Luengo, and Mónica F. Bugallo. Generalized Multiple Importance Sampling. *Statistical Science*, 34(1):129 – 155, 2019. doi: 10.1214/18-STS668.

Matteo Fasiolo, Flávio Eler de Melo, and Simon Maskell. Langevin incremental mixture importance sampling. *Statistics and Computing*, 28(3):549–561, 2018.

Axel Finke and Alexandre H Thiery. On importance-weighted autoencoders. 2019. URL https://arxiv.org/abs/1907.10477.

Michael B Giles. Multilevel monte carlo methods. *Acta numerica*, 24:259–328, 2015.

Peter W Glynn. Stochastic approximation for monte carlo optimization. In *Proceedings of the 18th conference on Winter simulation*, pages 356–365, 1986.

Thomas Guilmeau, Nicola Branchini, Emilie Chouzenoux, and Victor Elvira. Adaptive importance sampling for heavy-tailed distributions via α-divergence minimization. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, pages 3871–3879. PMLR, 2024. URL https://proceedings.mlr.press/v238/guilmeau24a.html.

Fangjian Guo, Xiangyu Wang, Kai Fan, Tamara Broderick, and David B Dunson. Boosting variational inference. In *Advances in Neural Information Processing Systems*, 2016.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Tommi S Jaakkola and Michael I Jordan. Improving the mean field approximation via the use of mixture distributions. In *Learning in graphical models*, pages 163–173. Springer, 1998.

Ghassen Jerfel, Serena Wang, Clara Wong-Fannjiang, Katherine A. Heller, Yian Ma, and Michael I. Jordan. Variational refinement for importance sampling using the forward kullback-leibler divergence. In Cassio P. de Campos, Marloes H. Maathuis, and Erik Quaeghebeur, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pages 1819–1829. AUAI Press, 2021. URL https://proceedings.mlr.press/v161/jerfel21a.html.

Renyan Jiang, Ming J. Zuo, and Han-Xiong Li. Weibull and inverse weibull mixture models allowing negative weights. *Reliability Engineering & System Safety*, 66(3):227–234, 1999.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on*

Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014. URL http://arxiv.org/abs/1312.6114.

Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomávs Skvrivan, Philipp Slusallek, and Jaroslav Křivánek. Optimal multiple importance sampling. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! *ICLR Deep RL Meets Structured Prediction Workshop*, 2019.

Anna Korba and François Portier. Adaptive importance sampling meets mirror descent : a bias-variance tradeoff. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 11503–11527. PMLR, 2022. URL https://proceedings.mlr.press/v151/korba22a.html.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David M Blei. Automatic differentiation variational inference. *Journal of machine learning research*, 18(14):1–45, 2017.

Leander Kurscheidt, Paolo Morettin, Roberto Sebastiani, Andrea Passerini, and Antonio Vergari. A probabilistic neuro-symbolic layer for algebraic constraint satisfaction. *ArXiv preprint*, abs/2503.19466, 2025. URL https://arxiv.org/abs/2503.19466.

Oskar Kviman, Harald Melin, Hazal Koptagel, Victor Elvira, and Jens Lagergren. Multiple importance sampling ELBO and deep ensembles of variational approximations. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 10687–10702. PMLR, 2022. URL https://proceedings.mlr.press/v151/kviman22a.html.

Marc Lambert, Sinho Chewi, Francis R. Bach, Silvère Bonnabel, and Philippe Rigollet. Variational inference via wasserstein gradient flows. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Wentao Li, Rong Chen, and Zhiqiang Tan. Efficient sequential monte carlo with multiple proposals and control variates. *Journal of the American Statistical Association*, 111(513):298–313, 2016.

Lorenzo Loconte, Nicola Di Mauro, Robert Peharz, and Antonio Vergari. How to turn your knowledge graph embeddings into generative models via probabilistic circuits. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*. Curran Associates, Inc., 2023.

Lorenzo Loconte, Aleksanteri M. Sladek, Stefan Mengel, Martin Trapp, Arno Solin, Nicolas Gillis, and Antonio Vergari. Subtractive mixture models via squaring: Representation and learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=xIHi5nxu9P.

Lorenzo Loconte, Antonio Mari, Gennaro Gala, Robert Peharz, Cassio de Campos, Erik Quaeghebeur, Gennaro Vessio, and Antonio Vergari. What is the relationship between tensor factorizations and circuits (and how can we exploit it)? *Transactions of Machine Learning Research*, 2025a. URL https://openreview.net/forum?id=Y7dRmpGiHj.

Lorenzo Loconte, Stefan Mengel, and Antonio Vergari. Sum of squares circuits. In *The 39th Annual AAAI Conference on Artificial Intelligence*, 2025b.

Lorenzo Loconte, Adrián Javaloy, and Antonio Vergari. How to square tensor networks and circuits without squaring them. *ICLR*, 2026.

Ulysse Marteau-Ferey, Francis R. Bach, and Alessandro Rudi. Non-parametric models for non-negative functions. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/968b15768f3d19770471e9436d97913c-Abstract.html.

Luca Martino. Sampling from mixtures with negative weights: application to density approximation by gaussian processes. Research Square preprint, 2025.

Luca Martino, Victor Elvira, David Luengo, and Jukka Corander. Layered adaptive importance sampling. *Statistics and Computing*, 27:599–623, 2017.

Geoffrey J McLachlan, Sharon X Lee, and Suren I

Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6(1):355–378, 2019.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21:132:1–132:62, 2020. URL http://jmlr.org/papers/v21/19-346.html.

Warren R. Morningstar, Sharad M. Vikram, Cusuh Ham, Andrew G. Gallagher, and Joshua V. Dillon. Automatic differentiation variational inference with mixtures. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 3250–3258. PMLR, 2021. URL http://proceedings.mlr.press/v130/morningstar21b.html.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (ToG)*, 38(5):1–19, 2019.

Radford M Neal. Slice sampling. *The annals of statistics*, 31(3):705–767, 2003.

Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

Art B. Owen. *Monte Carlo theory, methods and examples.* https://artowen.su.domains/mc/, 2013.

Art B Owen, Yury Maximov, and Michael Chertkov. Importance sampling the union of rare events with an application to power systems analysis. 2019.

Robert Peharz, Robert Gens, Franz Pernkopf, and Pedro Domingos. On the latent variable interpretation in sum-product networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(10): 2030–2044, 2016.

Guillaume Rabusseau and François Denis. Learning negative mixture models by tensor decompositions. *arXiv preprint arXiv:1403.4224*, 2014.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/rezende14.html.

Christian P Robert and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Christian P Robert and Julien Stoehr. Simulating signed mixtures. *Statistics and Computing*, 35(1): 1–21, 2025.

Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

Alessandro Rudi and Carlo Ciliberto. PSD representations for effective probability models. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 19411–19422, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a1b63b36ba67b15d2f47da55cdb8018d-Abstract.html.

Ernest K Ryu. *Convex optimization for Monte Carlo: Stochastic optimization for importance sampling.* Stanford University, 2016.

Tim Salimans and David A Knowles. On using control variates with stochastic approximation for variational bayes and its connection to stochastic linear regression. *arXiv preprint arXiv:1401.1022*, 2014.

Mateu Sbert, Vlastimil Havran, and Laszlo Szirmay-Kalos. Multiple importance sampling revisited: breaking the bounds. *EURASIP Journal on Advances in Signal Processing*, 2018(1):1–15, 2018.

Andy Shih and Stefano Ermon. Probabilistic circuits for variational inference in discrete graphical models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/31784d9fc1fa0d25d04eae50ac9bf787-Abstract.html.

Aleksanteri Sladek, Martin Trapp, and Arno Solin. Approximate bayesian inference via bitstring representations. In *Eighth Workshop on Tractable Probabilistic Modeling*, 2025.

Ping Liang Tan and Robert Peharz. Hierarchical decompositional mixtures of variational autoencoders.

In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6115–6124. PMLR, 2019. URL http://proceedings.mlr.press/v97/tan19b.html.

The april lab. cirkit, 2024. URL https://github.com/april-tools/cirkit.

Eric Veach and Leonidas J Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428, 1995.

Antonio Vergari, Nicola Di Mauro, and Guy Van den Broeck. Tractable probabilistic models: Representations, algorithms, learning, and applications. *Tutorial at UAI*, 2019.

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13189–13201, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/6e01383fd96a17ae51cc3e15447e7533-Abstract.html.

Benjie Wang and Guy Van den Broeck. On the relationship between monotone and squared probabilistic circuits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 21026–21034, 2025.

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

Yue Yang, Ryan Martin, and Howard Bondell. Variational approximations using fisher divergence. *ArXiv preprint*, abs/1905.05284, 2019. URL https://arxiv.org/abs/1905.05284.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5577–5586. PMLR, 2018. URL http://proceedings.mlr.press/v80/yao18a.html.

Baibo Zhang and Changshui Zhang. Finite mixture models with negative components. In *4th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM)*, pages 31–41. Springer, 2005.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes, provided throughout the paper, primarily §2 and §3, App. A, App. B.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes, see App. A.5.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes, see https://github.com/april-tools/delta-vi.
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes, see Theorem 1, Prop. 1, and App. B.
   (b) Complete proofs of all theoretical results. Yes, see App. B.
   (c) Clear explanations of any assumptions. Yes, see Theorem 1, Prop. 1, and App. B.
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes, see https://github.com/april-tools/delta-vi.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes, see App. C, Table 5, Table 7 and App. D.5.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes, see §6.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes, see App. C.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes, see App. C.
   (b) The license information of the assets, if applicable. Yes, see App. C.
   (c) New assets either in the supplemental material or as a URL, if applicable. Yes, see https://github.com/april-tools/delta-vi.
   (d) Information about consent from data providers/curators. Not Applicable.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable.
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable.

Supplementary material for: How to Approximate Inference with Subtractive Mixture Models

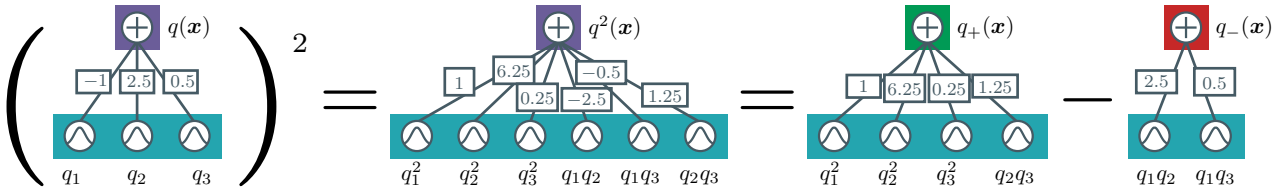


Figure 7: **A squared mixture can be split into its positive and negative parts** as illustrated via its representation as a computational graph, also called circuit (Choi et al., 2020; Loconte et al., 2025a).

A SAMPLING ALGORITHMS

In this section, we provide further details on the sampling algorithms discussed throughout the paper. Alg. 3 provides the algorithm for ancestral mixture sampling. Alg. 4 covers stratified sampling for additive mixture models. Alg. 5 gives detailed pseudocode for our ARITS implementation. We derive the variance of a MC estimator based on rejection in App. A.4. We discuss the complexity of the sampling strategies in App. A.5.

A.1 Ancestral sampling

Algorithm 3: ancestralSampling(q, S)

Input: an additive MM q Eq. (1), and sample budget S;
Output: S i.i.d. samples from q;
$\mathcal{X} \leftarrow \{\}$;
for $s \in \{1, ..., S\}$ **do**
  $k \sim \mathsf{Categorical}(\boldsymbol{\alpha})$;
  $\boldsymbol{x}^{(s)} \sim q_k$;
  $\mathcal{X} \leftarrow \mathcal{X} \cup \{\boldsymbol{x}^{(s)}\}$;
return $\mathcal{X}$

A.2 Stratified sampling

Algorithm 4: stratifiedSampling(q, S)

Input: an additive MM q Eq. (1), and sample budget S;
Output: up to S stratified samples from q;
$\mathcal{X} \leftarrow \{\}$;
$S_1, \ldots, S_K \leftarrow \lfloor \alpha_1 S \rfloor, \ldots, \lfloor \alpha_K S \rfloor$;
for $k \in \{1, ..., K\}$ **do**
  $\mathcal{X}^{(k)} \leftarrow \{\}$;
  for $s \in \{1, ..., S_k\}$ **do**
    $\boldsymbol{x}^{(k,s)} \sim q_k$;
    $\mathcal{X}^{(k)} \leftarrow \mathcal{X}^{(k)} \cup \{\boldsymbol{x}^{(k,s)}\}$;
  $\mathcal{X} \leftarrow \mathcal{X} \cup \mathcal{X}^{(k)}$;
return $\mathcal{X}$

A.3 Auto-regressive Inverse Transform Sampling (ARITS)

For performing ARITS in practice, we use a binary search for numerically inverting the conditional CDF. Alg. 5 provides the detailed algorithm. In our experiments, the start and end points of the binary search are set to

$L = -100$ and $B = 100$ respectively for RQ 1) and to $L = -50$ and $B = 50$ for learning[2]. The search is stopped $|L - B| > \epsilon$, for $\epsilon = 10^{-6}$. Note that computing the conditional CDF in the algorithm is tractable for SMMs as they are smooth and decomposable circuits (Vergari et al., 2021).

Algorithm 5: binarySearchArits(q, S, L, B, ϵ)

Input : a SMM q (Eq. (SMM)),
sample budget S,
initial upper bound B,
initial lower bound L,
tolerance ϵ
Output: S i.i.d. samples from q
for $s \in \{1, \ldots, S\}$ **do**
  $x^{(s)} \leftarrow \{\}$;
  for $d \in \{1, ..., D\}$ **do**
    $u \sim \text{Unif}(0, 1)$;
    `/* Pre-compute the evidence of previously sampled dimensions */`
    if $d > 1$ **then**
      $e \leftarrow q(x_1^{(s)}, ..., x_{d-1}^{(s)})$;
    else
      $e \leftarrow 1$
    `/* Perform binary search to numerically invert conditional CDF */`
    while $|L - B| > \epsilon$ **do**
      $M \leftarrow L + (B - L)/2$;
      `/* Compute conditional CDF at midpoint */`
      $c \leftarrow q(x_d^{(s)} \leq M,\ x_1^{(s)}, ..., x_{d-1}^{(s)})/e$;
      if $c > u$ **then**
        $B \leftarrow M$
      else
        $L \leftarrow M$
    `/* Recompute midpoint and set` $x_d^{(s)}$ `*/`
    $x_d^{(s)} \leftarrow L + (B - L)/2$
  $\boldsymbol{x}^{(s)} \leftarrow (x_1^{(s)}, \ldots, x_D^{(s)})$;
  $\mathcal{X} \leftarrow \mathcal{X} \cup \{\boldsymbol{x}^{(s)}\}$;
return $\mathcal{X}$

A.4 Rejection Sampling for SMMs

Derivation of rejection sampling variance.

Definition 1 *We define a zero-truncated binomial distribution* $\mathsf{TrBin}(S; a)$ *on* $\{1, \ldots, S\}$ *with probability mass function (PMF)*

$$\mathbb{P}[K = k] = \frac{\binom{S}{k} a^k (1-a)^{S-k}}{1 - (1-a)^S},\ k\ \in \{1, \ldots, S\},$$

and parameters $a \in [0, 1]$ *and* $S \in \mathbb{N}$.

Given the above definition, it is straightforward to derive the variance of the rejection estimator by applying the law of total variance

$$\begin{aligned} \mathbb{V}_{\substack{\boldsymbol{x} \sim q_{\text{SMM}} \\ K \sim \mathsf{TrBin}(S;a)}} \left[\widehat{I}_{\text{RS}}\right] &= \mathbb{E}_{K \sim \mathsf{TrBin}} \left[\mathbb{V}_{\boldsymbol{x} \sim q_{\text{SMM}}} [\widehat{I}_{\text{RS}} | K]\right] \\ &\quad + \mathbb{V}_{K \sim \mathsf{TrBin}} \left[\mathbb{E}_{\boldsymbol{x} \sim q_{\text{SMM}}} [\widehat{I}_{\text{RS}} | K]\right] \end{aligned} \tag{4}$$

[2] To ensure the validity of the algorithm, we always check whether these bounds result in a (conditional) CDF of 0 and 1 respectively.

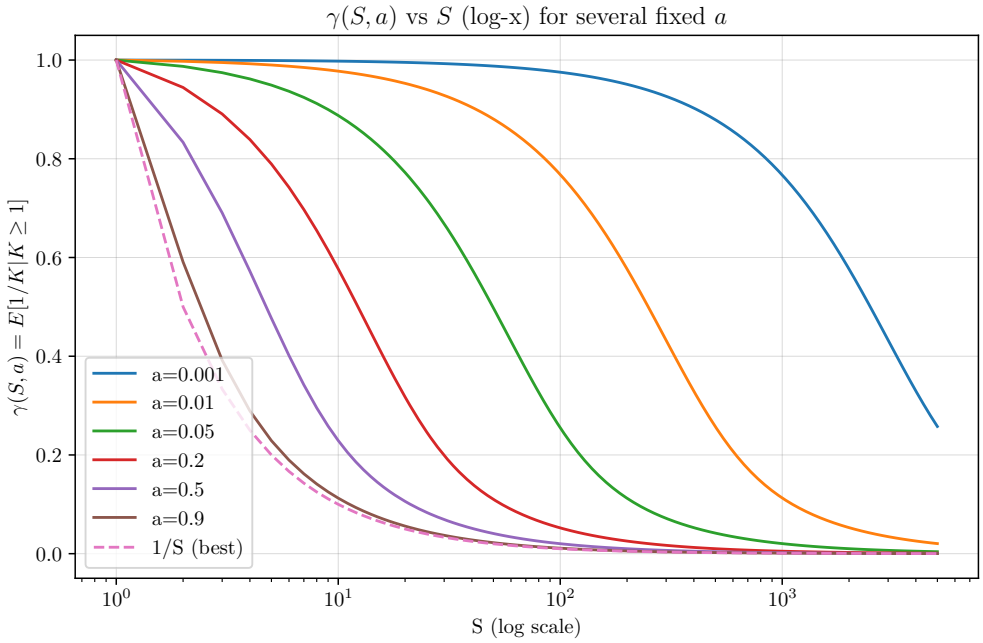


Figure 8: $\gamma(a, S)$ versus S (log-scale x-axis) for several values of a, the acceptance probability of rejection. The best convergence rate is $1/S$, which is almost achieved when a is close to 1.

Since $\mathbb{E}_{\boldsymbol{x}\sim q_{\text{SMM}}}[\widehat{I}_{\text{RS}}|K] = I$[3] for any K the second term is zero, so expanding the first term (using that accepted samples are i.i.d.),

$$\mathbb{V}[\widehat{I}_{\text{RS}}] = \mathbb{V}_{\boldsymbol{x}\sim q_{\text{SMM}}}[h(\boldsymbol{x})] \cdot \mathbb{E}\left[\frac{1}{K}\right]$$

In the following, we set $\gamma(S, a) := \mathbb{E}[1/K]$ under the defined truncated binomial

$$\gamma(S, a) := \mathbb{E}[1/K] = \frac{\sum_{k=1}^{S} \frac{1}{k}\binom{S}{k} a^k (1-a)^{S-k}}{1-(1-a)^S}. \tag{5}$$

The dependence on a is intuitive: as a goes from 0 to 1, 5 goes from 1 to $1/S$, i.e., impacting the MC convergence rate. See Fig. 8 for an illustration. Note that advanced schemes recycling rejected samples would be possible (Casella and Robert, 1998).

A.5 COMPLEXITY ANALYSIS

To derive the complexity of our sampling algorithm, we will consider classical additive MMs and SMMs to be represented as ***probabilistic circuits*** (PCs) (Choi et al., 2020; Vergari et al., 2021), computational graphs involving three types of computational units: sum, product and input units. Sum units compute linear combinations of their inputs and are used to represent the sum operation in MMs (Eq. (1)) and SMMs (Eq. (SMM)). Input distributions are custom neurons that encode parametric PDFs or mass functions. Product units encode local factorizations and appear in our models when we consider input distributions that factorize into independent marginals. Fig. 7 illustrates one squared SMM as a sum unit over many multivariate (and possibly unnormalized) Gaussian PDF units. Alternatively, we could have replaced each input unit in Fig. 7 with a product unit over D input units each encoding a (possibly unnormalized) univariate Gaussian PDF. Under this construction, we discuss complexity next.

Ancestral sampling (Alg. 3). For an additive MM, we first sample one component out of K from a Categorical, which can be done in $\mathcal{O}(K)$, assuming unitary cost to sample from uniform distributions. Then, we can sample from the relative isotropic Gaussian component, i.e., sampling from a univariate Gaussian D times. Assuming

[3]Conditioning on K, the number of accepted samples is equivalent to conditioning on the acceptance pattern, i.e., binary r.v.s. which denote exactly which sample was accepted.

the cost of sampling from a univariate Gaussian is unitary, one has to perform $\mathcal{O}(S(K+D))$ operations. If we adopt *stratified sampling* (Alg. 4) for an additive MM instead, we can first precompute how many samples each mixture component will yield by multiplying its corresponding mixture coefficient with S. We then visit the circuit in a feedforward way, sampling from all KD input units at once and propagate the partial samples by concatenating them along columns (dimensions D) when encountering a product unit and along rows (samples S) when encountering a sum unit. Again assuming the cost of sampling to be unitary, we have to compute $\mathcal{O}(S(K+KD)) = \mathcal{O}(SKD)$ operations, which corresponds to evaluating the full circuit S times.

ARITS (Alg. 1). To generate a single sample, we have to compute the inverse of the marginal CDF D times. To do that, we first have to marginalize out the circuit, which can be done exactly in time linear in the circuit size, i.e. $\mathcal{O}(KD)$. To then invert the CDF, we have to evaluate the marginal circuit in a binary search process, retrieving a sample up to precision ϵ, which has a cost of $\mathsf{b}(\epsilon)$. As such, the overall procedure should take $\mathcal{O}(SKD^2)\mathsf{b}(\epsilon)$.

Rejection sampling (Alg. 2). The worst-case complexity is the same as ancestral sampling, as we have to sample S times from the positive part of an SMM. While for the analysis we assumed that we have at least one acceptance, in practice it is not guaranteed to have at least one acceptance and therefore one should consider the runtime of rejection sampling more generally as a random variable (with not necessarily finite variance).

B PROOFS AND DERIVATIONS

B.1 Properties of ΔIS

We now prove the properties of ΔIS stated in Theorem 1.

B.1.1 Consistency of ΔIS

This property amounts to showing consistency separately for the two expectations. The conditions are slightly different than for UIS because the IS weight is different (and we have a combination of two estimators). Let the following assumptions be satisfied (beyond the usual $q \gg p$, $\int |f|p < \infty$)

- **(A1)** $q > 0$ q_+-a.e. and $q > 0$ q_--a.e. (that is, $q \gg q_+$ and $q \gg q_-$)
- **(A2)** $\mathbb{E}_{x\sim q_+}\left[|f(\boldsymbol{x})|\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right] < \infty$ and $\mathbb{E}_{x\sim q_-}\left[|f(\boldsymbol{x})|\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right] < \infty$.

The above are sufficient for an individual strong law of large numbers (SLLN) holding for the two estimators within ΔIS as both $S_+ \to \infty$ and $S_- \to \infty$. Finally, since the mapping $(x, y) \to (Z_+/Z)x - (Z_-/Z)y$ is continuous, by the continuous mapping theorem we have the desired SLLN for ΔIS,

$$\mathbb{P}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}\left[\lim_{S_+,S_-\to\infty} \widehat{I}_{\Delta\text{IS}} = I\right] = 1.$$

B.1.2 Unbiasedness of ΔIS for UIS

We show next show that ΔIS is unbiased.

Recall that our estimator for $I = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$ is given as

$$\widehat{I}_{\Delta\text{IS}} = \frac{Z_+}{Z}\frac{1}{S_+}\sum\nolimits_{s=1}^{S_+} f(\boldsymbol{x}_+^{(s)})w(\boldsymbol{x}_+^{(s)}) - \frac{Z_-}{Z}\frac{1}{S_-}\sum\nolimits_{s=1}^{S_-} f(\boldsymbol{x}_-^{(s)})w(\boldsymbol{x}_-^{(s)}), \text{ where } \begin{matrix}\boldsymbol{x}_+^{(s)} \sim q_+(\boldsymbol{x}_+)\\ \boldsymbol{x}_-^{(s)} \sim q_-(\boldsymbol{x}_-)\end{matrix}.$$

Note that $q(\boldsymbol{x}) = \frac{1}{Z}(Z_+q_+(\boldsymbol{x}) - Z_-q_-(\boldsymbol{x}))$ and $\{\boldsymbol{x}_+^{(s)}\}_{s=1}^{S_+} \overset{\text{i.i.d.}}{\sim} q_+$ and $\{\boldsymbol{x}_-^{(s)}\}_{s=1}^{S_-} \overset{\text{i.i.d.}}{\sim} q_-$. Assuming that

$\int |f(\boldsymbol{x})|p(\boldsymbol{x})dx < \infty$ and $q(\boldsymbol{x}) \neq 0$ almost-everywhere in the support of q_+ and q_-, we have

$$\mathbb{E}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}[\widehat{I}_{\Delta\text{IS}}] = \mathbb{E}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}\left[\frac{Z_+}{Z}\frac{1}{S_+}\sum_{s=1}^{S_+} f(\boldsymbol{x}_+^{(s)})\frac{p(\boldsymbol{x}_+^{(s)})}{q(\boldsymbol{x}_+^{(s)})} - \frac{Z_-}{Z}\frac{1}{S_-}\sum_{s=1}^{S_-} f(\boldsymbol{x}_-^{(s)})\frac{p(\boldsymbol{x}_-^{(s)})}{q(\boldsymbol{x}_-^{(s)})}\right] \tag{6}$$

$$= \frac{Z_+}{Z}\frac{1}{S_+}\sum_{s=1}^{S_+}\mathbb{E}_{q_+}\left[f(\boldsymbol{x}_+^{(s)})\frac{p(\boldsymbol{x}_+^{(s)})}{q(\boldsymbol{x}_+^{(s)})}\right] - \frac{Z_-}{Z}\frac{1}{S_-}\sum_{s=1}^{S_-}\mathbb{E}_{q_-}\left[f(\boldsymbol{x}_-^{(s)})\frac{p(\boldsymbol{x}_-^{(s)})}{q(\boldsymbol{x}_-^{(s)})}\right] \tag{7}$$

$$= \frac{Z_+}{Z}\mathbb{E}_{q_+}\left[f(\boldsymbol{x}_+)\frac{p(\boldsymbol{x}_+)}{q(\boldsymbol{x}_+)}\right] - \frac{Z_-}{Z}\mathbb{E}_{q_-}\left[f(\boldsymbol{x}_-)\frac{p(\boldsymbol{x}_-)}{q(\boldsymbol{x}_-)}\right] \tag{8}$$

$$= \frac{Z_+}{Z}\int f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q_+(\boldsymbol{x})d\boldsymbol{x} - \frac{Z_-}{Z}\int f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q_-(\boldsymbol{x})d\boldsymbol{x} \tag{9}$$

$$= \int f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\frac{1}{Z}(Z_+q_+(\boldsymbol{x}) - Z_-q_-(\boldsymbol{x}))d\boldsymbol{x} \tag{10}$$

$$= \int f(\boldsymbol{x})\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})d\boldsymbol{x} = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x} = I. \tag{11}$$

B.1.3 Variance of ΔIS

Using the independence of $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$ as well as the fact that $\{\boldsymbol{x}_+^{(s)}\}_{s=1}^{S_+} \overset{\text{i.i.d.}}{\sim} q_+$ and $\{\boldsymbol{x}_-^{(s)}\}_{s=1}^{S_-} \overset{\text{i.i.d}}{\sim} q_-$, we arrive at the following variance expression, where $w(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$,

$$\begin{aligned}\mathbb{V}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}[\widehat{I}_{\Delta\text{IS}}] &= \mathbb{V}_{q_+}\left[\frac{Z_+}{Z}\frac{1}{S_+}\sum\nolimits_{s=1}^{S_+} f(\boldsymbol{x}_+^{(s)})w(\boldsymbol{x}_+^{(s)})\right] + \mathbb{V}_{q_-}\left[\frac{Z_-}{Z}\frac{1}{S_-}\sum\nolimits_{s=1}^{S_-} f(\boldsymbol{x}_-^{(s)})w(\boldsymbol{x}_-^{(s)})\right] \\ &= \frac{Z_+^2}{Z^2}\frac{1}{S_+}\left(\mathbb{E}_{q_+}\left[(f(\boldsymbol{x}_+)w(\boldsymbol{x}_+))^2\right] - \left(\mathbb{E}_{q_+}\left[f(\boldsymbol{x}_+)w(\boldsymbol{x}_+)\right]\right)^2\right) \\ &+ \frac{Z_-^2}{Z^2}\frac{1}{S_-}\left(\mathbb{E}_{q_-}\left[(f(\boldsymbol{x}_-)w(\boldsymbol{x}_-))^2\right] - \left(\mathbb{E}_{q_-}\left[f(\boldsymbol{x}_-)w(\boldsymbol{x}_-)\right]\right)^2\right).\end{aligned}$$

B.1.4 Optimal UIS proposal for ΔIS

$f \geq 0$ or $f \leq 0$. It is easy to see that when $f(\boldsymbol{x}) > 0$ a.e. (or $f(\boldsymbol{x}) \leq 0$) a.e., the optimal proposal for ΔIS is $q^\star(\boldsymbol{x}) = \frac{f(\boldsymbol{x})p(\boldsymbol{x})}{I}$, where $I = \int f(\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}$ is the integral of interest. Plugging $q^\star$ into Eq. (ΔIS) gives

$$\frac{Z_+}{Z}\frac{1}{S_+}\sum_{s=1}^{S_+}\frac{f(\boldsymbol{x}^{(s)})p(\boldsymbol{x}^{(s)})}{f(\boldsymbol{x}^{(s)})p(\boldsymbol{x}^{(s)})/I} - \frac{Z_-}{Z}\frac{1}{S_-}\sum_{s=1}^{S_-}\frac{f(\boldsymbol{x}^{(s)})p(\boldsymbol{x}^{(s)})}{f(\boldsymbol{x}^{(s)})p(\boldsymbol{x}^{(s)})/I} = \left(\frac{Z_+}{Z} - \frac{Z_-}{Z}\right)I = I.$$

Since I is a constant, we have $\mathbb{V}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}[\widehat{I}_{\Delta\text{IS}}] = 0$ for $q = q^\star$.

General f. We now prove the optimal proposal for general f, which might take both positive and negative values. This proposal is given as $q^\star(\boldsymbol{x}) = \frac{|f(\boldsymbol{x})|p(\boldsymbol{x})}{\int |f(\boldsymbol{x})|p(\boldsymbol{x})d\boldsymbol{x}}$. The proof has two main steps (1) explicitly relating the variance of ΔIS to the standard UIS variance and (2) expressing the optimal proposal $q^\star$ as q_+.

Note first that

$$q = Z^{-1}(Z_+q_+ - Z_-q_-) \Rightarrow q_+ = Z/Z_+q + Z_-/Z_+q_-. \tag{12}$$

We now rewrite the variance of ΔIS in terms of the variance of UIS with the same proposal q. Let $w(\boldsymbol{x}) := \frac{f(\boldsymbol{x})p(\boldsymbol{x})}{q(\boldsymbol{x})}$. We know from App. B.1.4 that

$$\mathbb{V}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}[\widehat{I}_{\Delta\text{IS}}] = \frac{1}{S_+}\frac{Z_+^2}{Z^2}\mathbb{V}_{q_+}\Big[w(\boldsymbol{x})\Big] + \frac{1}{S_-}\frac{Z_-^2}{Z^2}\mathbb{V}_{q_-}\Big[w(\boldsymbol{x})\Big]. \tag{13}$$

Now, by rewriting the first term $\mathbb{V}_{q_+}\Big[w(\boldsymbol{x})\Big]$ as a function of q and q_- we will introduce the term $\mathbb{V}_q[w(\boldsymbol{x})]$

(variance of UIS) and establish an inequality between $\mathbb{V}_q[w(\boldsymbol{x})]$ and $\mathbb{V}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}[\widehat{I}_{\Delta\text{IS}}]$. Using

$$\mathbb{V}_{q_+}[w(\boldsymbol{x})] = \mathbb{E}_{q_+}[w(\boldsymbol{x})^2] - \mathbb{E}_{q_+}[w(\boldsymbol{x})]^2 \tag{14}$$

and Eq. (12), letting $a := Z/Z_+$, $b := Z_-/Z_+$, we have (abbreviating $w(\boldsymbol{x})$ as w)

$$\mathbb{V}_{q_+}[w] = \overbrace{a\mathbb{E}_q[w^2] + b\mathbb{E}_{q_-}[w^2]}^{=\mathbb{E}_{q_+}[w(\boldsymbol{x})^2]} - (a\mathbb{E}_q[w] + b\mathbb{E}_{q_-}[w])^2 \tag{15}$$

We rewrite Eq. (15) by expressing the second moments in terms of the corresponding variances, resulting in

$$\mathbb{V}_{q_+}[w] = a\mathbb{V}_q[w] + b\mathbb{V}_{q_-}[w] + \left[a\mathbb{E}_q^2[w] + b\mathbb{E}_{q_-}^2[w] - (a\mathbb{E}_q[w] + b\mathbb{E}_{q_-}[w])^2\right]. \tag{16}$$

The bracket in the above can be simplified. Expanding the square in the bracket:

$$\begin{aligned}
&a\mathbb{E}_q^2[w] + b\mathbb{E}_{q_-}^2[w] - \left(a\mathbb{E}_q[w] + b\mathbb{E}_{q_-}[w]\right)^2 \\
&= a\mathbb{E}_q^2[w] + b\mathbb{E}_{q_-}^2[w] - a^2\mathbb{E}_q^2[w] - b^2\mathbb{E}_{q_-}^2[w] - 2ab\mathbb{E}_q[w]\mathbb{E}_{q_-}[w] \\
&= a(1-a)\mathbb{E}_q^2[w] + b(1-b)\mathbb{E}_{q_-}^2[w] - 2ab\mathbb{E}_q[w]\mathbb{E}_{q_-}[w].
\end{aligned}$$

Since $a + b = 1$, we have $1 - a = b$ and $1 - b = a$, and therefore

$$\begin{aligned}
a(1-a)\mathbb{E}_q^2[w] + b(1-b)\mathbb{E}_{q_-}^2[w] - 2ab\mathbb{E}_q[w]\mathbb{E}_{q_-}[w] &= ab\mathbb{E}_q^2[w] + ab\mathbb{E}_{q_-}^2[w] - 2ab\mathbb{E}_q[w]\mathbb{E}_{q_-}[w] \\
&= ab\left(\mathbb{E}_q[w] - \mathbb{E}_{q_-}[w]\right)^2.
\end{aligned}$$

We can finally rewrite Eq. (15) as

$$\mathbb{V}_{q_+}[w] = a\mathbb{V}_q[w] + b\mathbb{V}_{q_-}[w] + ab(\mathbb{E}_q[w] - \mathbb{E}_{q_-}[w])^2. \tag{17}$$

Noting that $b\mathbb{V}_{q_-}[w] \geq 0$ and $ab(\mathbb{E}_q[w] - \mathbb{E}_{q_-}[w])^2 \geq 0$, it follows that

$$\mathbb{V}_{q_+}[w] \geq a\mathbb{V}_q[w] = \frac{Z}{Z_+}\mathbb{V}_q[w]. \tag{18}$$

Using this identity and going back to Eq. (13), we obtain

$$\mathbb{V}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}[\widehat{I}_{\Delta\text{IS}}] = \frac{1}{S_+}\frac{Z_+^2}{Z^2}\mathbb{V}_{q_+}\Big[w(\boldsymbol{x})\Big] + \frac{1}{S_-}\frac{Z_-^2}{Z^2}\mathbb{V}_{q_-}\Big[w(\boldsymbol{x})\Big] \tag{19}$$

$$\geq \frac{1}{S_+}\frac{Z_+^2}{Z^2}\mathbb{V}_q\Big[w(\boldsymbol{x})\Big] \tag{20}$$

$$\geq \frac{1}{S_+}\cdot\frac{Z}{Z_+}\cdot\frac{Z_+^2}{Z^2}\mathbb{V}_q\Big[w(\boldsymbol{x})\Big] = \frac{1}{S_+}\cdot\frac{Z}{Z_+}\mathbb{V}_q\Big[w(\boldsymbol{x})\Big], \tag{21}$$

which relates the variance of ΔIS and $\mathbb{V}_q\Big[w(\boldsymbol{x})\Big]$. To complete the first part of the proof, we show

$$\frac{Z_+}{Z}\frac{1}{S_+} \geq \frac{1}{S}.$$

which implies $\frac{1}{S_+}\cdot\frac{Z}{Z_+}\mathbb{V}_q\Big[w(\boldsymbol{x})\Big] \geq 1/S\mathbb{V}_q\Big[w(\boldsymbol{x})\Big] = \mathbb{V}_{x\sim q}[\widehat{I}_{\text{UIS}}]$. Using that

$$\frac{Z_+}{ZS_+} - \frac{1}{S} = \frac{Z_+S - ZS_+}{ZS_+S},$$

and further rewriting the numerator using $S = S_+ + S_-$and $Z = Z_+ - Z_-$, we obtain

$$\begin{aligned}
Z_+S - ZS_+ &= Z_+\left(S_+ + S_-\right) - \left(Z_+ - Z_-\right)S_+ \\
&= Z_+S_- + Z_-S_+ \geq 0.
\end{aligned}$$

We have hence shown $\mathbb{V}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}[\widehat{I}_{\Delta\text{IS}}] \geq \mathbb{V}_{x\sim q}[\widehat{I}_{\text{UIS}}]$.

To prove that the optimal proposal $q^\star \propto |f|p$ is the same for $\widehat{I}_{\Delta\text{IS}}$ and $\widehat{I}_{\text{UIS}}$, we will show $\mathbb{V}_{\substack{\boldsymbol{x}_+\sim q_+\\ \boldsymbol{x}_-\sim q_-}}[\widehat{I}_{\Delta\text{IS}}]$ using $q = q^\star$ is no greater than using any other proposal q. In particular, there is a choice of $q(\boldsymbol{x}) = (Z_+/Z)\ q_+(\boldsymbol{x}) - (Z_-/Z)\ q_-(\boldsymbol{x})$ that obtains minimum variance: Set $q = q_+$ and $S = S_+$, which implies $Z_- = 0$ (no negative component). Then, the ΔIS estimator is equivalent to UIS. We can now simply set $q = q_+ = q^\star$ in ΔIS. Then, for any q the below holds

$$\underbrace{\mathbb{V}_{q^\star}[\widehat{I}_{\Delta\text{IS}}]}_{\text{choosing } S=S_+} = \mathbb{V}_{q^\star}[\widehat{I}_{\text{UIS}}] \leq \mathbb{V}_{q}[\widehat{I}_{\text{UIS}}] \leq \mathbb{V}_{q}[\widehat{I}_{\Delta\text{IS}}] \quad \blacksquare. \tag{22}$$

B.2 Derivation of ΔVI

Let q_θ denote an SMM proposal with K components and corresponding learnable parameters $\theta = (\theta_1, \ldots, \theta_K)$. The learnable parameters include both the mixture weights and the component parameters. Applying Eq. (ΔEX), we decompose the RKL objective:

$$\mathbb{E}_{q_\theta}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] = \frac{Z_+}{Z}\mathbb{E}_{q_+}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] - \frac{Z_-}{Z}\mathbb{E}_{q_-}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] \tag{23}$$

$$= \frac{Z_+}{Z}\int\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right]q_+(\boldsymbol{x})d\boldsymbol{x} - \frac{Z_-}{Z}\int\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right]q_-(\boldsymbol{x})d\boldsymbol{x}. \tag{24}$$

We then apply stratification with respect to the underlying components, as described in Morningstar et al. (2021), within q_+ and q_- respectively. For the first term, we plug in $q_+(\boldsymbol{x}) = \frac{1}{Z_+}\sum_{k\in\mathcal{K}_+}\alpha_k\ \widetilde{q}_k(\boldsymbol{x})$, where $\mathcal{K}_+$ denotes the set of indices corresponding to positively weighted components. We denote the normalizing constant of $\widetilde{q}_k$ as by Z_k, i.e., $Z_k = \int \widetilde{q}_k(\boldsymbol{x})d\boldsymbol{x}$.

We obtain the following for the first part

$$\frac{Z_+}{Z}\int\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right]q_+(\boldsymbol{x})d\boldsymbol{x} = \frac{Z_+}{Z}\int\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right]\frac{1}{Z_+}\sum_{k=1}\alpha_k\ \widetilde{q}_k(\boldsymbol{x})d\boldsymbol{x} \tag{25}$$

$$= \frac{1}{Z}\int\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right]\sum_{k\in\mathcal{K}_+}\alpha_k Z_k\ \frac{\widetilde{q}_k(\boldsymbol{x})}{Z_k}d\boldsymbol{x} \tag{26}$$

$$= \frac{1}{Z}\int\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right]\sum_{k\in\mathcal{K}_+}\alpha_k Z_k\ q_k(\boldsymbol{x})d\boldsymbol{x} \tag{27}$$

$$= \sum_{k\in\mathcal{K}_+}\frac{\alpha_k Z_k}{Z}\int\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right]q_k(\boldsymbol{x})d\boldsymbol{x} \tag{28}$$

$$= \sum_{k\in\mathcal{K}_+}\frac{\alpha_k Z_k}{Z}\mathbb{E}_{q_k}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right]. \tag{29}$$

Analogously, we can rewrite the second expectation w.r.t. $q_-(\boldsymbol{x}) = \frac{1}{Z_-}\sum_{j\in\mathcal{K}_-}|\alpha_j|\ \widetilde{q}_j(\boldsymbol{x})$, resulting in

$$\frac{Z_-}{Z}\int\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right]q_-(\boldsymbol{x})d\boldsymbol{x} = \sum_{j\in\mathcal{K}_-}\frac{|\alpha_j|Z_j}{Z}\mathbb{E}_{q_j}\left[\log\left(\frac{q_{\text{SMM}}(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right].$$

Putting the two terms back together, we arrive at

$$\mathbb{E}_{q_\theta}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] = \sum_{k\in\mathcal{K}_+}\frac{\alpha_k Z_k}{Z}\mathbb{E}_{q_k}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] - \sum_{j\in\mathcal{K}_-}\frac{|\alpha_j|Z_j}{Z}\mathbb{E}_{q_j}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] \tag{30}$$

$$= \sum_{k\in\mathcal{K}_+}\frac{\alpha_k Z_k}{Z}\mathbb{E}_{q_k}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] + \sum_{j\in\mathcal{K}_-}\frac{\alpha_j Z_j}{Z}\mathbb{E}_{q_j}\left[\log(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})})\right] \tag{31}$$

$$= \sum_{k=1}^{K}\frac{\alpha_k Z_k}{Z}\mathbb{E}_{q_k}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right], \tag{32}$$

which is our ΔVI objective. The last few steps use that for $j \in \mathcal{K}_-$, the associated coefficients α_j were negative

before taking the absolute value by construction, and $\mathcal{K}_+ \cup \mathcal{K}_-$ results in the full set of components.

Estimation of the ΔIS gradient. We have shown that we can safely rewrite the initial objective in terms of a decomposition into individual components, despite negative coefficients, namely

$$\mathbb{E}_{q_\theta}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] = \sum_{k=1}^{K} \frac{\alpha_k Z_k}{Z} \mathbb{E}_{q_k}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right].$$

We assume that the components q_k are *reparameterizable* with respect to some parameter-independent reference distribution q_0, i.e., $\boldsymbol{x} = h_{\theta_k}(\boldsymbol{z}),\ \boldsymbol{z} \sim q_0$, for some mapping $h_{\theta_k} : \mathbb{R}^D \to \mathbb{R}^D$. Rewriting the above in terms of $\boldsymbol{z}$, we obtain

$$\mathbb{E}_{q_\theta}\left[\log\left(\frac{q_\theta(\boldsymbol{x})}{\widetilde{p}(\boldsymbol{x})}\right)\right] = \sum_{k=1}^{K} \frac{\alpha_k Z_k}{Z} \mathbb{E}_{\boldsymbol{z}\sim q_0}\left[\log\left(\frac{q_\theta(h_{\theta_k}(\boldsymbol{z}))}{\widetilde{p}(h_{\theta_k}(\boldsymbol{z}))}\right)\right].$$

In practice, given a sampling budget of S samples during training, we distribute the budget evenly across components, i.e., $S_k = \lfloor \frac{S}{K} \rfloor$. The (unbiased) estimator of the objective is then given as

$$\sum_{k=1}^{K} \frac{\alpha_k Z_k}{Z} \frac{1}{S_k} \sum_{s=1}^{S_k} \log\left(\frac{q_\theta(h_{\theta_k}(\boldsymbol{z}^{(s)}))}{\widetilde{p}(h_{\theta_k}(\boldsymbol{z}^{(s)}))}\right), \boldsymbol{z}^{(s)} \overset{\text{i.i.d.}}{\sim} q_0,$$

which is a fully differentiable objective in terms of mixture weights, allowing us to compute the corresponding gradients via automatic differentiation (Kucukelbir et al., 2017; Morningstar et al., 2021).

C EXPERIMENTAL SETUP

In the following, we describe the experimental setup used for the research questions discussed in §6. We implement all of our experiments in Python using the `cirkit` library[4] (The april lab, 2024), which was released under the GPL-3.0 license. Moreover, we adapt code from the repositories of EigenVI[5] (Cai et al., 2024a), released under the Apache-2.0 license, and BeyondELBOs[6] (Blessing et al., 2024), released under the MIT license. Experiments reporting runtime are run on a single NVIDIA RTX A6000 (48GiB VRAM) each and runtimes are measured using `time.perf_counter`. Experiments on synthetic and neuro-symbolic targets were also run on NVIDIA RTX A6000 (48GiB VRAM). Bayesian logistic regression experiments were run on NVIDIA A100 and RTX 3080 Ti GPUs. We use Weights and Biases[7] and PyTorch Lightning[8] for training.

C.1 Target Specifications

In the following, we define the targets used throughout the paper.

C.1.1 Two-dimensional Targets

The first three targets are taken from Cai et al. (2024a).

GMM3 ($d = 2$). *GMM3* is a 3-component GMM defined as follows:

$$p_{\text{GMM3}}(\boldsymbol{x}) = 0.4 \cdot \mathcal{N}(\boldsymbol{x}|[-1,1]^T, \Sigma) + 0.3 \cdot \mathcal{N}(\boldsymbol{x}|[1.1,1.1]^T, \Sigma_2) + 0.3 \cdot \mathcal{N}(\boldsymbol{x}|[-1,-1]^T, \Sigma_2),$$

where $\Sigma = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$ and $\Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

GMM4 ($d = 2$). *GMM4* is a 4-component GMM defined as follows:

$$p_{\text{GMM4}}(\boldsymbol{x}) = \frac{1}{4} \cdot \mathcal{N}(\boldsymbol{x}|[0,2]^T, \Sigma_1) + \frac{1}{4} \cdot \mathcal{N}(\boldsymbol{x}|[-2,0]^T, \Sigma_2) + \frac{1}{4} \cdot \mathcal{N}(\boldsymbol{x}|[2,0]^T, \Sigma_2) + \frac{1}{4} \cdot \mathcal{N}(\boldsymbol{x}|[0,-2]^T, \Sigma_1)$$

where $\Sigma_1 = \begin{pmatrix} 0.15^{0.9} & 0 \\ 0 & 1 \end{pmatrix}$ and $\Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 0.15^{0.9} \end{pmatrix}$.

[4] https://github.com/april-tools/cirkit
[5] https://github.com/dicai/eigenVI
[6] https://github.com/DenisBless/variational_sampling_methods
[7] https://github.com/wandb/wandb
[8] https://github.com/Lightning-AI/pytorch-lightning

Funnel ($d = 2$). The two-dimensional funnel for $\boldsymbol{x} = (x_1, x_2)$ is given by

$$p_{\text{Funnel}}(\boldsymbol{x}) = \mathcal{N}(x_1|0, \sigma^2)\mathcal{N}(x_2|0, \exp(x_1/2)),$$

where $\sigma = 1.2$. Note that we add a small constant (10^{-6}) to the log-density of the funnel to match the implementation by Cai et al. (2024a).

Ring ($d = 2$). We define the *Ring* target as a squared SMM with real mixture weights. Note that the SMM need to be renormalized after squaring, hence a normalizing constant appears in the following definition.

$$p_{\text{Ring}}(\boldsymbol{x}) = \frac{1}{Z_{\text{Ring}}}\left(\mathcal{N}(\boldsymbol{x}|[0,0]^T, \Sigma_1) - 0.46 \cdot \mathcal{N}(\boldsymbol{x}|[0,0]^T, \Sigma_2)\right)^2$$

where $\Sigma_1 = \begin{pmatrix} 3^2 & 0 \\ 0 & 3^2 \end{pmatrix}$ and $\Sigma_2 = \begin{pmatrix} 2^2 & 0 \\ 0 & 2^2 \end{pmatrix}$.

DeepRing ($d = 2$). For our analysis of the effect of a safe component on ΔIS (see D.3.1), we additionally define the following *DeepRing*, which has a mode in the middle.

$$p_{\text{DeepRing}}(\boldsymbol{x}) = \frac{1}{Z_{\text{DeepRing}}}\left(0.16 \cdot \mathcal{N}(\boldsymbol{x}|[0,0]^T, \Sigma_1) - 0.36 \cdot \mathcal{N}(\boldsymbol{x}|[0,0]^T, \Sigma_2)\right)^2$$

where $\Sigma_1 = \begin{pmatrix} 0.6^2 & 0 \\ 0 & 0.6^2 \end{pmatrix}$ and $\Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

C.1.2 Higher-dimensional Targets

All of the *Hollow* targets are squared SMMs with two components and with real mixture weights.

Hollow ($d = 16$).

$$p_{\text{Hollow}(d=16)}(\boldsymbol{x}) = \frac{1}{Z_{\text{Hollow(d=16)}}}\left(\mathcal{N}(\boldsymbol{x}|[0,0]^T, \sigma_1^2 \cdot I^{16\times16}) - 0.3 \cdot \mathcal{N}(\boldsymbol{x}|[0,0]^T, \sigma_2^2 \cdot I^{16\times16})\right)^2$$

where $\sigma_1 = 7$ and $\sigma_2 = 6$. $I^{16\times16}$ denotes the 16-dimensional identity matrix.

Hollow ($d = 32$).

$$p_{\text{Hollow}(d=32)}(\boldsymbol{x}) = \frac{1}{Z_{\text{Hollow(d=32)}}}\left(\mathcal{N}(\boldsymbol{x}|[0,0]^T, \sigma_1^2 \cdot I^{32\times32}) - 0.11 \cdot \mathcal{N}(\boldsymbol{x}|[0,0]^T, \sigma_2^2 \cdot I^{32\times32})\right)^2$$

where $\sigma_1 = 7$ and $\sigma_2 = 6$. $I^{32\times32}$ denotes the 32-dimensional identity matrix.

Hollow ($d = 64$).

$$p_{\text{Hollow}(d=64)}(\boldsymbol{x}) = \frac{1}{Z_{\text{Hollow(d=64)}}}\left(\mathcal{N}(\boldsymbol{x}|[0,0]^T, \sigma_1^2 \cdot I^{64\times64}) - 0.074 \cdot \mathcal{N}(\boldsymbol{x}|[0,0]^T, \sigma_2^2 \cdot I^{64\times64})\right)^2$$

where $\sigma_1 = 7$ and $\sigma_2 = 6.5$. $I^{64\times64}$ denotes the 64-dimensional identity matrix.

Funnel ($d = 10$). The 10-dimensional *Funnel* is taken from Blessing et al. (2024):

$$p_{\text{Funnel}(D\,=\,10)}(\boldsymbol{x}) = \mathcal{N}(x_1|0, \sigma^2)\mathcal{N}(x_2|0, \exp(x_1)) \ldots \mathcal{N}(x_{10}|0, \exp(x_1)),$$

where $\sigma = 3$.

Bayesian Logistic Regression. These targets are obtained as posterior distributions of a real valued parameter $\boldsymbol{x} \in \mathbb{R}^D$, given a dataset $\{y_n, \boldsymbol{z}_n\}_{n=1}^N$ where $y \in \{0, 1\}$ are labels and $\boldsymbol{z}_n \in \mathbb{R}^{d_z}$ are covariates. The targets are then given as

$$p(\boldsymbol{x}) := p(\boldsymbol{x}|\{y_n, \boldsymbol{z}_n\}_{n=1}^N) \propto \prod_n p(y_n|\boldsymbol{x}, \boldsymbol{z}_n) \cdot p_0(\boldsymbol{x}),$$

where $p(y_n|\boldsymbol{x}, \boldsymbol{z}_n) = \text{Bernoulli}(y_n; \text{sigmoid}(\boldsymbol{x}^\top \boldsymbol{z}_n))$ and p_0 is a standard Gaussian prior, $p_0(\boldsymbol{x}) = \mathcal{N}(0; I^{D\times D})$. We obtain different targets by using various datasets from Blessing et al. (2024).

C.1.3 Stanford Drone Dataset

The *Stanford Drone Dataset (SDD)* (Robicquet et al., 2016) consists of top-down street views of the Stanford campus including the trajectories of pedestrians and vehicles. We target densities that were fit to the given

Table 4: Average acceptance probability of the random instances generated for RQ1, given as $\frac{Z}{Z_+}$. Each cell reports the mean and standard deviation over the 30 generated instances for the corresponding combination of features (D) and components (K).

	K		
	2	4	6
D	$\frac{Z}{Z_+}$	$\frac{Z}{Z_+}$	$\frac{Z}{Z_+}$
16	0.489 ± 0.240	0.373 ± 0.217	0.349 ± 0.247
32	0.591 ± 0.193	0.423 ± 0.171	0.517 ± 0.200
64	0.719 ± 0.142	0.673 ± 0.191	0.611 ± 0.180

trajectories with the *probabilistic algebraic layer* (PAL) by (Kurscheidt et al., 2025) using 10 mixture components and 14 knots per spline. As in Kurscheidt et al. (2025), we refer to the first target (Fig. 1 and top half of App. D.5) as *scenario 1* and the second target (Fig. 6 and bottom half of App. D.5) as *scenario 2*. Note that PAL models *hard constraints* and hence sets the density of invalid areas (such as buildings, obstacles, and roundabouts) to 0. This is impractical for VI with the RKL since the target density appears in the denominator of the objective. Therefore, we pre-process the PAL densities, such that invalid areas are assigned a log-density of -20 for scenario 1 and -25 for scenario 2. We found that these values lead to stable training dynamics for both SMMs and GMMs.

C.2 RQ1: Scaling sampling with SMMs

In our first set of experiments, addressing RQ1), our sampling distribution is a squared SMM with Gaussian inputs. The means are initialized with a $\text{Unif}([-0.5, 0.5])$ distribution and standard deviations are drawn from a $\text{Unif}([2, 3])$ distribution. The mixture weights are initialized with a $\text{Unif}([-1, 1])$ distribution. We repeat the random initialization until the generated model has at least one negatively weighted component after squaring. Table 4 gives an overview of the average accepted probability when performing rejection sampling (Alg. 2) on the resulting SMMs. The target function f is initialized with 100 Gaussian components for all settings. The means for are initialized using a standard normal distribution and the standard deviations are sampled from a $\text{Unif}([1, 2])$ distribution. The weights of the sum layer are sampled from a $\text{Unif}([10000, 100000])$ distribution to encourage a non-zero target expectation in high dimensions. All methods were sampled with a maximum batch size of 5000.

C.3 RQ2: Comparing VI strategies for SMMs

In the following, we describe the experimental setup that was used to obtain the results in Tab. 1 and Tab. 2. For our *squared, complex SMMs*, we do the following: For each target density, we train models from 5 different initializations and perform hyperparameter search according to Tab. 5. Within each run, we save the 5 best checkpoints according to the training loss. The resulting set of models is what we select the models in Tab. 1 and Tab. 2 from. In particular, *we choose the model with the best average training loss* according to the following scheme: For ΔVI and RLOO (Rej.), recomputing the training loss is inexpensive, and hence we repeat the estimation 30 times for targets with $D < 16$, and 10 times otherwise. To further stabilize the selection for models on synthetic targets, we only consider models for which the average train loss exceeds -0.1, as we expect it to be positive for these targets. For RLOO with ARITS, we repeat the loss estimation 10 times for targets with less than 16 dimensions, and compute it only once otherwise. All models models in Tab. 1 and Tab. 2 were trained with 10^5 samples per update step, except for the 10-dimensional *Funnel* (Blessing et al., 2024), which we trained with 10^4 samples per update for all models. We use the Adam optimizer (Kingma and Ba, 2015) for all models.

For *EigenVI* (Cai et al., 2024a), we use a $\text{Unif}([-9, 9])$ proposal for all targets and fit the models based on 10^5 samples from this proposal. We use normalized Hermite polynomials as the basis functions of EigenVI, as was done by Cai et al. (2024a) in their experiments. We train a single EigenVI model with this setup for each target as EigenVI does not rely on stochastic gradient descent.

For the logistic regression targets (Tab. 3), we did not run ARITS due computational constraints. The hyperparameter details are in Table 7. The model selection procedure used here is the same as of the other targets.

The weights for SMMs are initialised so that the real part has a probability of 0.5 of being negative. This is to encourage initial negative contributions. Then the positive weights absolute values are initialised $\mathsf{Unif}(0.5, 2.0)$, while absolute values for the negative weights in $\mathsf{Unif}(0.1, 0.5)$.

Evaluation. We report the *evidence lower bound (ELBO)* for all settings and the forward Kullback-Leibler divergence (FKL) for synthetic targets. We estimate the forward KL for the synthetic densities as

$$\mathrm{FKL}(p, q) = \frac{1}{S}\sum_{s=1}^{S} \log\Big(\frac{p(\boldsymbol{x}^{(s)})}{q(\boldsymbol{x}^{(s)})}\Big), \quad \boldsymbol{x}^{(s)} \overset{\text{i.i.d.}}{\sim} p.$$

We do not compute the FKL for the Bayesian logistic regressions in Tab. 3 as we do not have access to ground-truth target samples in this setting. For synthetics targets, for which we know the normalizing constant, we additionally report the reverse Kullback-Leibler divergence (RKL), which is given as

$$\mathrm{RKL}(q, p) = \frac{1}{S}\sum_{s=1}^{S} \log\Big(\frac{q(\boldsymbol{x}^{(s)})}{p(\boldsymbol{x}^{(s)})}\Big), \quad \boldsymbol{x}^{(s)} \overset{\text{i.i.d.}}{\sim} q.$$

Lastly, for Bayesian logistic regression targets, for which we neither have access to ground-truth samples nor the normalizing constants, we report the ELBO given as

$$\mathrm{ELBO}(q, p) = -\frac{1}{S}\sum_{s=1}^{S} \log\Big(\frac{q(\boldsymbol{x}^{(s)})}{\widetilde{p}(\boldsymbol{x}^{(s)})}\Big), \quad \boldsymbol{x}^{(s)} \overset{\text{i.i.d.}}{\sim} q.$$

Table 5: Initializations and hyperparameter search spaces for Table 1 and Table 2. Weight decay is applied to the mixture weights only for all models.

2D Synthetics: GMM3, GMM4, Funnel, Ring		
Model	**Parameter**	**Initialization**
GMM	mean of Gaussian	$\text{Unif}(-1, 1)$
	stddev of Gaussian	$\text{Unif}(1, 3)$
	mixture weights	$\frac{1}{K}$
SMM ($\mathbb{C}$)	mean of Gaussian	$\text{Unif}(-1, 1)$
	stddev of Gaussian	$\text{Unif}(1, 3)$
	mixture weights (real)	$\text{Unif}(0, 1)$
	mixture weights (imaginary)	$\mathcal{N}(0, 1)$
Loss	**Hyperparameter/Config**	**Range**
ΔVI and GMM	lr	{0.01, 0.001, 0.0001}
	patience	3000
	weight decay	None
	max steps	15000
RLOO + ARITS	lr	{0.01}
	patience	500
	weight decay	None
	max steps	15000
RLOO + Rejection	lr	{0.01}
	patience	500
	weight decay	None
	max steps	15000
Hollow ($d = 16, 32, 64$) *... Parameters equal across d unless denoted otherwise*		
Model	**Parameter**	**Initialization**
GMM	mean of Gaussian	$\text{Unif}(-1, 1)$
	stddev of Gaussian	$d \in \{16, 64\} : \text{Unif}(5, 7),\ D = 32 : \text{Unif}(6, 8)$
	mixture weights	$\frac{1}{K}$
SMM ($\mathbb{C}$)	mean of Gaussian	$\text{Unif}(-1, 1)$
	stddev of Gaussian	$d \in \{16, 64\} : \text{Unif}(5, 7),\ D = 32 : \text{Unif}(6, 8)$
	mixture weights (real)	$\text{Unif}(0, 1)$
	mixture weights (imaginary)	$\mathcal{N}(0, 1)$
Loss	**Hyperparameter/Config**	**Range**
ΔVI and GMM	lr	{0.01, 0.001}
	patience	None
	weight decay	{0.0, 0.001}
	max steps	30000
RLOO + ARITS	lr	$d \in \{16, 32\} : 0.01,\ D = 64 : 0.001$
	patience	100
	weight decay	{0.0, 0.001}
	max steps	30000
RLOO + Rejection	lr	$d \in \{16, 32\} : 0.01,\ D = 64 : 0.001$
	patience	1000
	weight decay	{0.0, 0.001}
	max steps	30000
Funnel ($d = 10$)		
Model	**Parameter**	**Initialization**
GMM	mean of Gaussian	$\text{Unif}(-4, 4)$
	stddev of Gaussian	$\text{Unif}(2, 4)$
	mixture weights	$\frac{1}{K}$
SMM ($\mathbb{C}$)	mean of Gaussian	$\text{Unif}(-4, 4)$
	stddev of Gaussian	$\text{Unif}(2, 4)$
	mixture weights (real)	$\text{Unif}(0, 1)$
	mixture weights (imaginary)	$\mathcal{N}(0, 1)$
Loss	**Hyperparameter/Config**	**Range**
ΔVI and GMM	lr	{0.01, 0.001}
	patience	None
	weight decay	{0, 0.001}
	max steps	60000
RLOO + Rejection	lr	{0.01, 0.001}
	patience	None
	weight decay	{0, 0.001}
	max steps	60000

Table 6: Selected hyperparameters for Tab. 1 and Tab. 2. For the full search space see Tab. 5. The number of components, K, was fixed for these experiments. Weight dacay was only used for higher-dimensional targets and was only applied to the mixture weights.

Target	Hyperparameter	GMM	ΔVI	RLOOKL (Rej.)	RLOOKL (ARITS)
GMM3 ($D = 2$)	lr	0.001	0.001	0.01	0.01
	K	3	3	3	3
GMM4 ($D = 2$)	lr	0.001	0.01	0.01	0.01
	K	4	4	4	4
Funnel ($D = 2$)	lr	0.01	0.001	0.01	0.01
	K	16	16	16	16
Ring ($D = 2$)	lr	0.01	0.01	0.01	0.01
	K	2	2	2	2
Hollow ($D = 16$)	lr	0.01	0.01	0.01	0.01
	weight decay	0.001	0	0	0.001
	K	2	2	2	2
Hollow ($D = 32$)	lr	0.01	0.001	0.01	0.01
	weight decay	0	0	0	0.001
	K	2	2	2	2
Hollow ($D = 64$)	lr	0.01	0.001	0.001	/
	weight decay	0	0.001	0.001	/
	K	2	2	2	/
Funnel ($D = 10$)	lr	0.001	0.01	0.01	/
	weight decay	0.001	0	0	/
	K	16	16	16	/

D Additional Results and Experiments

D.1 Complete Results for RQ1

Tab. 8 reports the concrete runtime and estimation error for the results visualized in Fig. 3. Additionally, it reports the results when choosing $K = 2$ and $K = 4$, while Fig. 3 only shows $K = 6$. The general setup is the same across all experiments (see App. C.2). All methods were sampled with a batch size of 5000. We note that for rejection sampling, the very first instance that was run in our setup took considerably longer than the remaining, explaining the comparatively high standard deviation of the cell ($D = 16, K = 2, S = 10000$). The remaining runtimes were stable.

D.2 Further Evaluation of RQ2

In the following, we provide additional results and evaluations for our VI experiments. We group these results according to their overarching research question in the main text.

D.2.1 RQ2.1: Quantitative Comparison to EigenVI

In this section, we provide a quantitative comparison of the models depicted in Tab. 1. All models, except for EigenVI, were fit with 10^5 samples per update step with the Adam optimizer (Kingma and Ba, 2015) and hyperparameter search according to Tab. 5. The selected hyperparameters for each density can be found in Tab. 6. The number of components is fixed to 3 for *GMM3*, 4 for *GMM4*, 16 for *Funnel* and 2 for *Ring*. All EigenVI models were fit using 10^5 samples from a Unif($[-9, 9]$) distribution. We provide the results for EigenVI with two different number of parameters: *EigenVI (S):* a model that roughly matches ours in terms of parameter count, *EigenVI (L):* the largest model reported by (Cai et al., 2024a). Since the *Ring* target was not used by Cai et al. (2024a), to determine a suitable *EigenVI (L)* model, we increased the parameter count until we observed a good fit. Following (Cai et al., 2024a), we report the forward KL (FKL) between the target and the variational approximation. The reported forward FKL is estimated from 10^5 target samples and averaged over 10 repeated estimations. We re-estimated the FKL for EigenVI models ourselves in order to have a consistent setup for all methods. We found the resulting FKL estimates to be very close to the ones reported by Cai et al. (2024a) for GMM3 and GMM4. For the Funnel, we observe worse results than reported in Cai et al. (2024a) by

Table 7: Initializations and hyperparameter search spaces for BLR posteriors.

Logistic regressions

Model	**Parameter**	**Initialization**
GMM	mean of Gaussian	$\text{Unif}(-2, 2)$
	stddev of Gaussian	$\text{Unif}(6, 8)$
	mixture weights	$1/K$
SMM ($\mathbb{C}$)	mean of Gaussian	$\text{Unif}(-2, 2)$
	stddev of Gaussian	$\text{Unif}(6, 8)$
	mixture weights	positive: $\mathsf{Unif}(0.5, 2.0)$; negative: $\mathsf{Unif}(0.1, 0.5)$; Prob. of negative: 0.5

Loss	**Hyperparameter/Config**	**Range**
ΔVI and GMM	lr	{0.001, 0.01}
	patience	1500
	max steps	10000
	weight decay	0
	K (components)	{4, 8, 16}
	samples/update	5000
	optimizer	Adam
RLOO + Rejection	lr	{0.001, 0.01}
	patience	1500
	max steps	10000
	weight decay	0
	K (components)	{4, 8, 16}
	samples/update	5000

Table 8: **Rejection sampling and ΔIS are consistently faster than ARITS for expectation estimation while achieving comparable estimation quality when given a sufficient sampling budget.** Results for MC comparing our method (ΔIS) with ARITS for a varying number of features (d) and components (K) as well as different sampling budgets (S) for ΔIS. The error is given as $\log(|\widehat{I} - I|) - \log(I)$, hence lower is better, and time is in seconds. Results are averaged over 30 initializations of q_{SMM} and f (mean ± stddev).

			Number of components (K)					
			2		4		6	
Method	D	S	$\log(\lvert\widehat{I} - I\rvert) - \log(I)$ (↓)	Time (s)	$\log(\lvert\widehat{I} - I\rvert) - \log(I)$ (↓)	Time (s)	$\log(\lvert\widehat{I} - I\rvert) - \log(I)$ (↓)	Time (s)
ΔIS	16	10000	-2.221 ± 1.453	0.028 ± 0.002	-1.667 ± 1.671	0.029 ± 0.000	-1.425 ± 1.544	0.033 ± 0.000
ΔIS	16	100000	-2.989 ± 1.218	0.201 ± 0.003	-2.436 ± 1.349	0.211 ± 0.002	-2.714 ± 1.818	0.232 ± 0.003
ΔIS	16	1000000	-4.138 ± 1.379	1.901 ± 0.031	-3.617 ± 1.391	2.002 ± 0.025	-3.854 ± 1.664	2.194 ± 0.029
Rej.	16	10000	-2.727 ± 1.082	0.028 ± 0.017	-3.323 ± 1.880	0.024 ± 0.001	-3.185 ± 1.635	0.025 ± 0.001
Rej.	16	100000	-4.221 ± 1.284	0.232 ± 0.011	-3.874 ± 0.967	0.232 ± 0.011	-4.004 ± 1.029	0.235 ± 0.005
Rej.	16	1000000	-5.520 ± 1.365	2.283 ± 0.058	-5.384 ± 1.823	2.284 ± 0.046	-5.189 ± 0.999	2.338 ± 0.052
ARITS	16	10000	-3.415 ± 1.084	4.905 ± 0.073	-3.823 ± 1.185	5.517 ± 0.066	-3.307 ± 1.116	6.794 ± 0.018
ΔIS	32	10000	-0.806 ± 1.539	0.035 ± 0.001	-0.009 ± 1.090	0.037 ± 0.001	-0.830 ± 1.161	0.040 ± 0.000
ΔIS	32	100000	-1.860 ± 1.173	0.274 ± 0.013	-1.197 ± 1.370	0.284 ± 0.002	-1.845 ± 1.243	0.304 ± 0.003
ΔIS	32	1000000	-2.948 ± 1.412	2.612 ± 0.030	-2.340 ± 1.323	2.719 ± 0.018	-2.819 ± 1.213	2.919 ± 0.027
Rej.	32	10000	-1.795 ± 0.973	0.031 ± 0.001	-1.561 ± 0.822	0.030 ± 0.001	-1.492 ± 1.101	0.033 ± 0.001
Rej.	32	100000	-2.343 ± 0.964	0.291 ± 0.012	-2.475 ± 1.305	0.283 ± 0.011	-2.304 ± 1.051	0.313 ± 0.019
Rej.	32	1000000	-3.828 ± 0.951	2.904 ± 0.121	-3.543 ± 1.161	2.825 ± 0.106	-3.903 ± 1.117	3.112 ± 0.127
ARITS	32	10000	-1.713 ± 0.816	10.316 ± 0.107	-1.654 ± 0.743	13.042 ± 0.050	-1.865 ± 0.916	19.740 ± 0.048
ΔIS	64	10000	-0.253 ± 0.858	0.051 ± 0.002	0.144 ± 1.320	0.054 ± 0.003	0.054 ± 1.601	0.055 ± 0.002
ΔIS	64	100000	-0.035 ± 1.313	0.428 ± 0.010	-0.588 ± 1.047	0.444 ± 0.016	-0.393 ± 1.224	0.455 ± 0.009
ΔIS	64	1000000	-0.830 ± 1.612	4.136 ± 0.073	-1.075 ± 1.379	4.256 ± 0.098	-0.878 ± 1.249	4.408 ± 0.084
Rej.	64	10000	-0.405 ± 0.626	0.050 ± 0.002	-0.356 ± 0.614	0.051 ± 0.003	-0.348 ± 0.843	0.057 ± 0.002
Rej.	64	100000	-0.967 ± 0.927	0.445 ± 0.029	-0.553 ± 0.498	0.436 ± 0.024	-0.948 ± 1.517	0.496 ± 0.027
Rej.	64	1000000	-1.471 ± 1.008	4.832 ± 0.185	-1.316 ± 0.716	4.877 ± 0.306	-1.389 ± 1.286	5.462 ± 0.225
ARITS	64	10000	-0.566 ± 0.836	22.059 ± 0.235	-0.509 ± 0.781	37.049 ± 0.051	-0.415 ± 0.622	64.612 ± 0.194

roughly one order of magnitude. Tab. 9 summarizes the results. EigenVI can achieve a good fit for all targets in terms of FKL, but generally requires a high parameter count to do so. This is likely do to the fact that EigenVI uses non-learnable components. Our squared SMM models with complex mixture weights can achieve better or comparable fits to EigenVI while being more parameter-efficient. However, our VI strategies require hyperparameter tuning (see Tab. 5) and can be sensitive to the initialization (see Fig. 4 and Fig. 10).

Table 9: **Our proposed VI variants for squared SMMs can achieve comparable performance to EigenVI while being more parameter-efficient.** All models are fit and selected as described in App. C.3. See Tab. 1 for a visualization of the densities. The FKL is estimated from 10^5 target samples and estimation is repeated 10 times (reported is mean ± stddev). #P denotes the parameter count. All targets are two-dimensional.

	EigenVI (S)		EigenVI (L)		ΔVI		RLOO (Rej.)		RLOO (ARITS)	
	# P	FKL (↓)	# P	FKL (↓)	# P	FKL (↓)	# P	FKL (↓)	# P	FKL (↓)
GMM3	16	$1.78 \cdot 10^{-2} \pm 5.90 \cdot 10^{-4}$	64	$5.55 \cdot 10^{-4} \pm 1.07 \cdot 10^{-4}$	18	$2.04 \cdot 10^{-4} \pm 8.36 \cdot 10^{-5}$	18	$2.34 \cdot 10^{-4} \pm 5.37 \cdot 10^{-5}$	18	$2.63 \cdot 10^{-4} \pm 9.14 \cdot 10^{-5}$
GMM4	25	$1.38 \cdot 10^{-1} \pm 1.40 \cdot 10^{-3}$	196	$4.93 \cdot 10^{-3} \pm 4.77 \cdot 10^{-4}$	24	$5.51 \cdot 10^{-3} \pm 2.90 \cdot 10^{-4}$	24	$1.07 \cdot 10^{-4} \pm 4.81 \cdot 10^{-5}$	24	$7.44 \cdot 10^{-5} \pm 3.88 \cdot 10^{-5}$
Funnel	100	$2.43 \cdot 10^{-1} \pm 6.40 \cdot 10^{-3}$	256	$2.65 \cdot 10^{-1} \pm 5.00 \cdot 10^{-3}$	96	$4.11 \cdot 10^{-2} \pm 8.31 \cdot 10^{-4}$	96	$8.17 \cdot 10^{-4} \pm 1.94 \cdot 10^{-4}$	96	$1.28 \cdot 10^{-3} \pm 1.68 \cdot 10^{-4}$
Ring	16	$2.32 \cdot 10^{0} \pm 8.02 \cdot 10^{-3}$	196	$1.62 \cdot 10^{-2} \pm 6.33 \cdot 10^{-4}$	12	$8.35 \cdot 10^{-2} \pm 1.21 \cdot 10^{-3}$	12	$1.24 \cdot 10^{-5} \pm 1.74 \cdot 10^{-5}$	12	$1.98 \cdot 10^{-6} \pm 1.01 \cdot 10^{-5}$

D.2.2 RQ2.2: Model Visualizations for Ring Target

The boxplots in Fig. 4 are created based on 10 random initializations, taking the best recorded checkpoint for each and estimating the RKL and FKL between the model and the target from 10^4 samples. All models were trained as described in App. C.3 with a learning rate of 10^{-2} and no weight decay. See Fig. 9 for visualizations of the learned models at sample size 10^6. At a large sampling budget of $S = 10^6$, ΔVI manages to capture a ring-like structure for more initializations than the RLOO models, which tend to get stuck in local optima. However, the best RLOO models are visually a better fit to the target than the best ΔVI models. Potentially, better initializations or a different choice of hyperparameters could mitigate the issues ΔVI and the RLOO variants face on this target.

D.2.3 RQ2.3: Initializations and Resulting Models for Hollow Target

As mentioned in the main text, our SMM models can be quite sensitive to the initialization. We show this behavior on the example of the *Hollow* target with $d = 16$ in Fig. 10. The figure shows the 5 different initializations generated in our experiments and the learned models for ΔVI and the RLOO variants. All models were trained

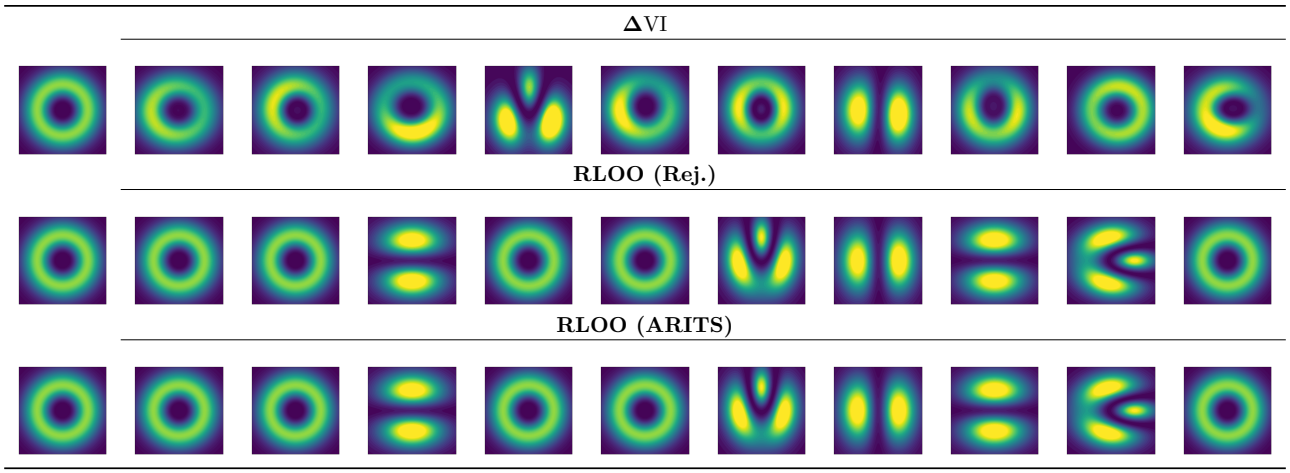


Figure 9: Visual comparisons of models obtained with $\boldsymbol{S = 10^6}$ samples per update step for 10 initializations. The first column shows the target. The depicted models were used to create Fig. 4.

with a sampling budget of $S = 10^5$ and a learning rate of 0.01 with no weight decay. We show the best checkpoint based on training divergence. We create two-dimensional visualizations of the density by setting all variables except the first two to a fixed value of 5 and evaluating the density as a function of the first two variables.

D.3 RQ3: IS with SMMs

In this section, we provide results for IS with SMM proposals. We group the section into two research questions: (**RQ3.1**) tackles the effect of a safe component on ΔIS using synthetic proposals whereas (**RQ3.2**) uses learned proposals from RQ2 and compares various estimation strategies.

D.3.1 RQ3.1: Safe ΔIS

We study the effect of a safe component on the *DeepRing* and *Hollow* targets. See App. C.1 for their specification. Note that these targets are squared SMMs and we have access to their ground-truth parameters. To create proposals that are close to the optimal UIS proposal, which in this case is simply p (Robert and Casella, 1999; Owen, 2013), we gently noise the standard deviations of the Gaussian components of p while keeping the remaining parameters fixed. Let $\boldsymbol{\sigma}_p$ denote the vector consisting of the standard deviations of p *before squaring* and let $|\boldsymbol{\sigma}_p|$ denote its size. We obtain a corresponding vector $\boldsymbol{\sigma}_q$ as follows

$$\boldsymbol{Z} \sim \mathcal{N}(0, I^{(|\boldsymbol{\sigma}_p|)})$$
$$\boldsymbol{\sigma}_q := \boldsymbol{\sigma}_p \cdot \exp(0.01 \cdot \boldsymbol{Z}).$$

The resulting vector $\boldsymbol{\sigma}_q$ is then used to realize the synthetic proposal q_{SMM}. We then mix the proposal with a safe component and assess whether this improves estimation quality. The full proposal is therefore given as

$$q(\boldsymbol{x}) = (1-\beta) \cdot q_{\text{SMM}}(\boldsymbol{x}) + \beta \cdot q_{\text{safe}}(\boldsymbol{x}),$$

where $\beta \in [0, 1)$ is a hyperparameter.

We realize the corresponding estimator for $\mathbb{E}_p[f]$ as follows

$$\widehat{I}^{(\text{safe})}_{\Delta\text{IS}} = (1-\beta)\left[\frac{Z_+}{Z}\frac{1}{S_+}\sum\nolimits_{s=1}^{S_+} f(\boldsymbol{x}_+^{(s)})\frac{p(\boldsymbol{x}_+^{(s)})}{q(\boldsymbol{x}_+^{(s)})} - \frac{Z_-}{Z}\frac{1}{S_-}\sum\nolimits_{s=1}^{S_-} f(\boldsymbol{x}_-^{(s)})\frac{p(\boldsymbol{x}_-^{(s)})}{q(\boldsymbol{x}_-^{(s)})}\right]$$
$$+ \beta\left[\frac{1}{S_{\text{safe}}}\sum\nolimits_{s=1}^{S_{\text{safe}}} f(\boldsymbol{x}_-^{(s)})\frac{p(\boldsymbol{x}_{\text{safe}}^{(s)})}{q(\boldsymbol{x}_{\text{safe}}^{(s)})}\right],$$

where $\boldsymbol{x}_+ \sim q_+$, $\boldsymbol{x}_- \sim q_-$, $\boldsymbol{x}_{\text{safe}} \sim q_{\text{safe}}$, and each set of samples is i.i.d. Note that the first part of the estimator corresponds to a standard ΔIS estimator in structure, but with the crucial difference of having the combined proposal q in the denominator of the IS weight as opposed to just q_{SMM}. Similarly, the second term is a weighted IS estimator using samples from the safe component but evaluating the full q in the IS weight. In practice, we split the total sampling budget such that $S := S_{\Delta\text{IS}} + S_{\text{safe}}$ as $S_{\Delta\text{IS}} = \lfloor(1-\beta)S\rfloor$ and $S_{\text{safe}} = \lfloor\beta S\rfloor$ and then

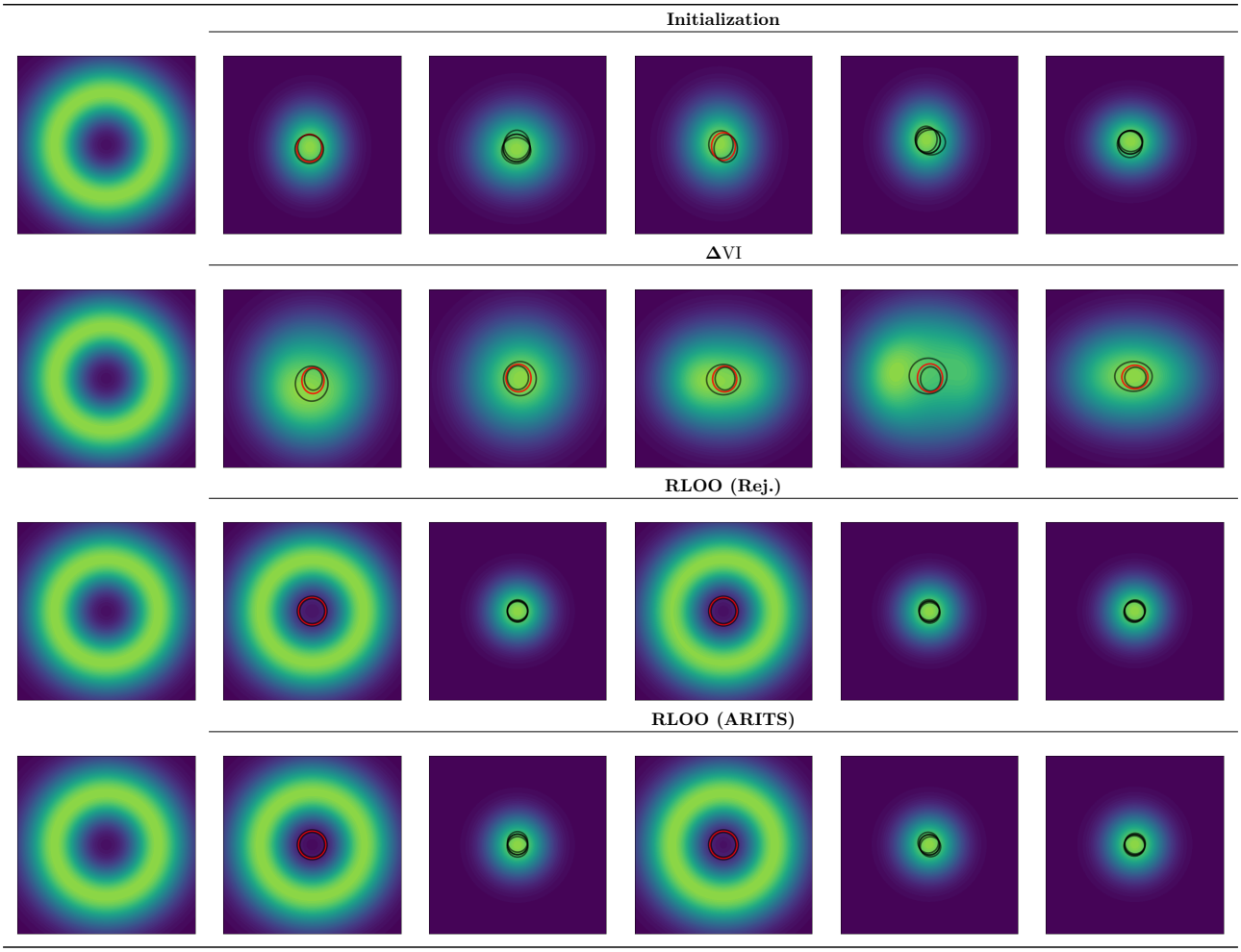


Figure 10: **The quality of the variational approximation can be sensitive to the initialization.** The figure shows learned models for 5 different initializations for the *Hollow* target in 16 dimensions. The components are sketched as ellipses with width and height corresponding to their standard deviations. Positively weighted components are illustrated in black and negative components are shown in red. The first column shows the target. Notably, the initializations look visually similar. Nevertheless, only two of the five RLOO models (with either sampling scheme) converge to a ring-like unnormalized conditional density.

distribute $S_{\Delta\text{IS}}$ across q_+ and q_- as described in §3.

In our experiments, we heuristically choose the safe component as a zero-mean isotropic Gaussian with a comparatively high standard deviation for the target. In the experiments below, we use $\sigma = 3$ for the *DeepRing* target and $\sigma = 8$ for the *Hollow* targets. The estimation quality is measured by computing $\log(|\widehat{I} - I|) - \log(I)$ over 100 repetitions. Fig. 11 summarizes the results. For all targets, ΔIS initially results in high variance and atypical behavior for an MC estimator: The average estimation error *increases* as the sampling budget increases. A possible explanation could be that the noising process sketched above creates ill-defined proposals. However, we only observe this behavior for ΔIS: The standard UIS estimators based on rejection and ARITS achieve both (1) better average estimation error than ΔIS and (2) a steady decrease in estimation error with increased sampling budget, as is to be expected. Adding a safe component stabilizes the ΔIS estimator. Interestingly, the performance is not very sensitive to the value of β in the range $\{0.2, 0.4, 0.8\}$. Moreover, for the *DeepRing* target and the *Hollow* target in 64 dimensions, using the safe component in isolation gives better estimates than any of the ΔIS estimators. This is surprising, as the safe component in isolation should be a worse fit to the target than any of the mixed ΔIS proposals. As a result, it seems like the mixed ΔIS proposals are a suboptimal choice for the sampling distribution induced by ΔIS. Possibly, a more principled safe component that is fit for the specific SMM proposal would improve estimation with ΔIS.

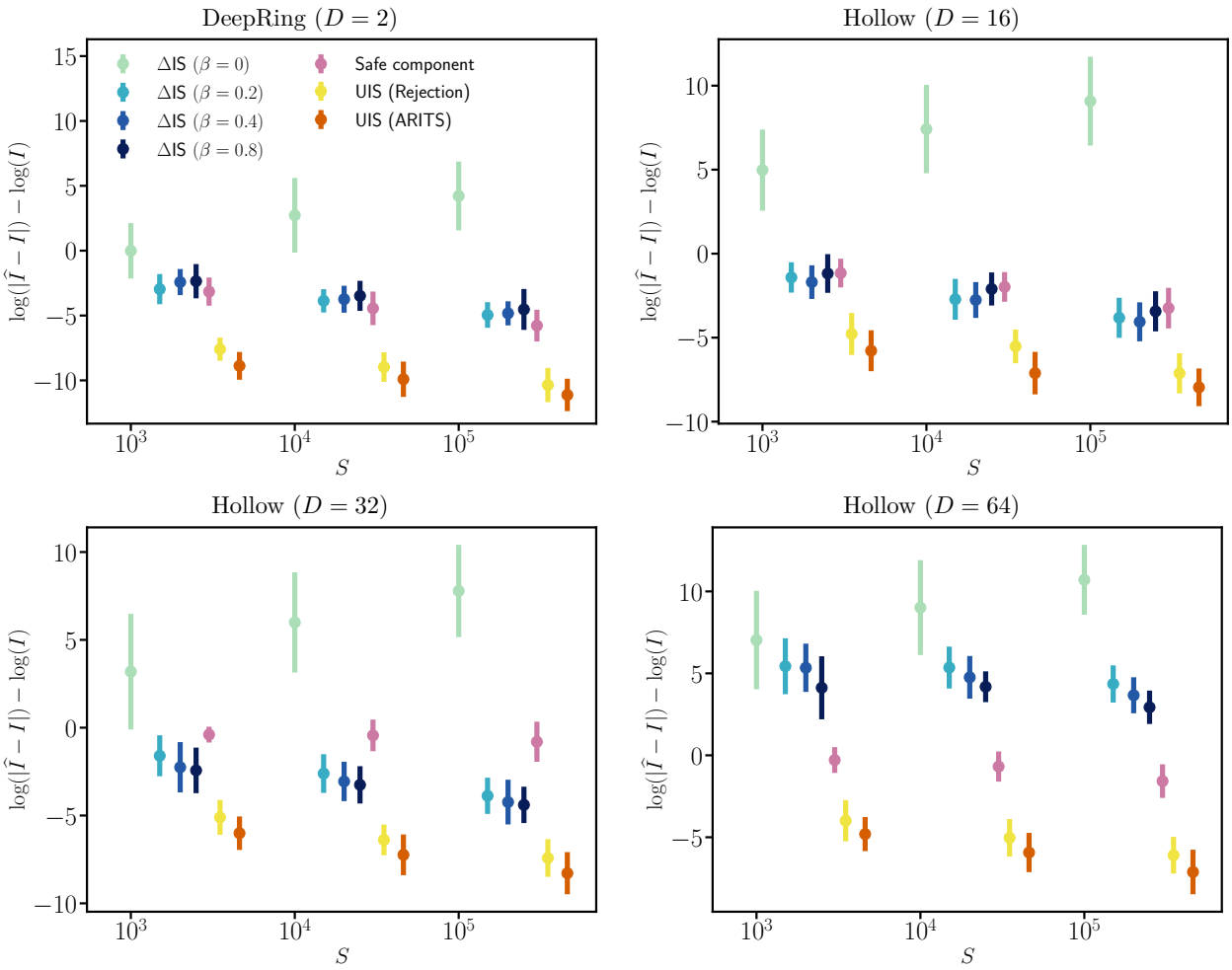


Figure 11: **A safe component can mitigate the potentially large variance of ΔIS.** Depicted is the estimation error, $\log(|\widehat{I} - I|) - \log(I)$, averaged over 100 repeated estimations (mean ± stddev) for various sampling budgets S. *Lower is better.* Without a safe component (i.e., $\beta = 0$), ΔIS can result in high variance and the average estimation error gets *worse* as the sampling budget increases. Standard UIS estimators do not share this behavior despite using the same proposal. Mixing the proposal with a safe component stabilizes the variance of ΔIS but does not match the performance of the UIS estimators in these examples.

D.3.2 RQ3.2: IS with learned proposals

In this section, we evaluate the the estimation quality achieved by the IS schemes discussed in §3 using learned proposals. The integral of interest is the normalizing constant of the target density, i.e., $I = \int \tilde{p}(\boldsymbol{x})d\boldsymbol{x}$. We use the models selected for Tab. 1 and Tab. 2 as proposals. For SMMs, we use the proposals trained via RLOO with rejection sampling since this learning scheme generally offered a good compromise between approximation quality and computational efficiency. For ΔIS, we perform a grid search of $\beta \in \{0, 0.1, 0.2, \ldots, 0.9\}$ and $\sigma_{safe} \in \{3, 5, 7, 9\}$ and select the best setting based on the empirical variance of the estimator with 10^4 samples based on 30 repetitions.We use a sampling budget of $S = 10^4$ for all estimators. We repeat the estimation 100 times and report the average estimation error, measured as $\log(|I - \widehat{I}|) - \log(I)$, and its standard deviation across the repetitions.

Tab. 11 summarizes the results. Our main takeaways match the observations made throughout the paper: When a GMM is sufficient to model the target well, as is the case for the *Funnels* and *GMM3* and *GMM4*, using a GMM proposal gives the best results on normalizing constant estimation. For targets that require an SMM for finding a good proposal, i.e., the *Ring* and the *Hollow* targets, using an SMM proposal with either rejection sampling or ARITS can result in better normalizing constant estimation than using a GMM. Interestingly, ΔIS

tends to perform worse than a simple GMM even on targets that benefit from negative components. For most targets, no safety was chosen for ΔIS by the gird search we perform to select β and σ_{safe} (see Tab. 10). How to construct and select a better safe component for a given proposal is an interesting open question.

Table 10: Selected parameters for the safe component of ΔIS for Tab. 11 and Fig. 12.

Target	β	σ_{safe}
GMM3 ($D = 2$)	0	/
GMM4 ($D = 2$)	0	/
Funnel ($D = 2$)	0	/
Ring ($D = 2$)	0	/
Hollow ($D = 16$)	0	/
Hollow ($D = 32$)	0	/
Hollow ($D = 64$)	0	/
Funnel ($D = 10$)	0.1	3

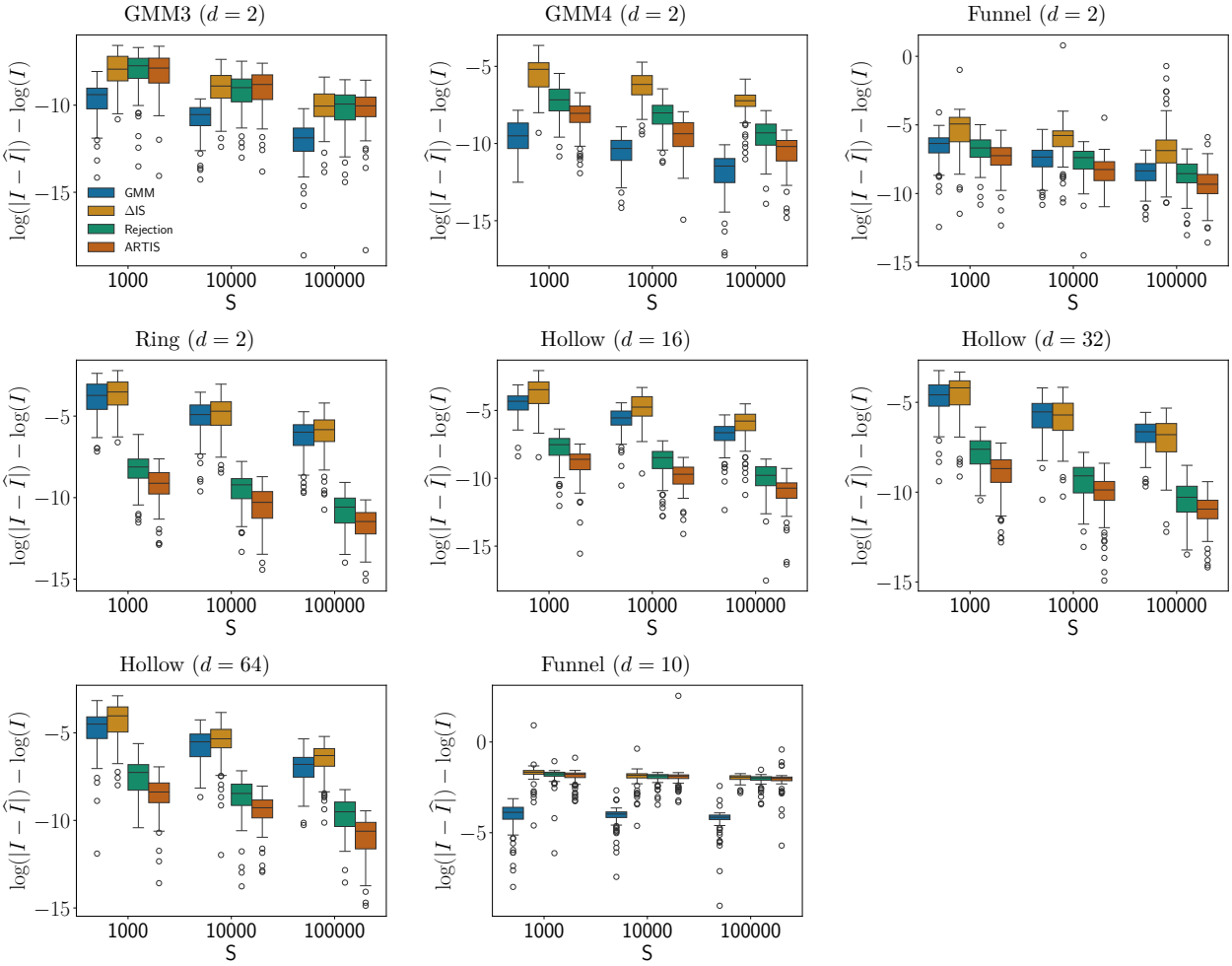


Figure 12: **The trends observed in Tab. 11 hold across various sample sizes.** The boxplots summarize the error for normalizing constant estimation over 100 repetitions when using GMM and SMM proposals for varying sampling budgets. *Lower is better.* For targets on which the SMM achieves a better fit, we see better estimation performance when using the SMM proposal with either rejection or ARITS. On the remaining targets, GMMs tend to be better. Note that the SMM proposal was always learned with RLOO + rejection.

Table 11: **SMMs can improve upon GMMs for normalizing constant estimation when an SMM proposal allows for a better fit to the target.** All estimates are computed with a sampling budget of $S = 10^4$. The estimation error is measured as $\log(|I - \widehat{I}|) - \log(I)$ reported over 100 repeated estimations (mean ± standard deviation). The SMM proposals were fit with RLOO using rejection sampling and the GMM proposals were learned via the SELBO (Morningstar et al., 2021).

	UIS (GMM)	ΔIS (safe)	UIS (Rej.)	UIS (ARITS)
GMM3 ($D = 2$)	$-1.08 \cdot 10^1 \pm 9.89 \cdot 10^{-1}$	$-9.04 \cdot 10^0 \pm 1.03 \cdot 10^0$	$-9.21 \cdot 10^0 \pm 1.04 \cdot 10^0$	$-9.10 \cdot 10^0 \pm 1.15 \cdot 10^0$
GMM4 ($D = 2$)	$-1.05 \cdot 10^1 \pm 1.03 \cdot 10^0$	$-6.34 \cdot 10^0 \pm 9.29 \cdot 10^{-1}$	$-8.22 \cdot 10^0 \pm 1.02 \cdot 10^0$	$-9.55 \cdot 10^0 \pm 1.16 \cdot 10^0$
Funnel ($D = 2$)	$-7.59 \cdot 10^0 \pm 1.00 \cdot 10^0$	$-6.09 \cdot 10^0 \pm 1.38 \cdot 10^0$	$-7.67 \cdot 10^0 \pm 1.13 \cdot 10^0$	$-8.41 \cdot 10^0 \pm 1.00 \cdot 10^0$
Ring ($D = 2$)	$-5.13 \cdot 10^0 \pm 1.16 \cdot 10^0$	$-4.94 \cdot 10^0 \pm 1.13 \cdot 10^0$	$-9.56 \cdot 10^0 \pm 1.03 \cdot 10^0$	$-1.05 \cdot 10^1 \pm 1.13 \cdot 10^0$
Hollow ($D = 16$)	$-5.80 \cdot 10^0 \pm 1.02 \cdot 10^0$	$-4.86 \cdot 10^0 \pm 1.11 \cdot 10^0$	$-8.85 \cdot 10^0 \pm 1.22 \cdot 10^0$	$-9.89 \cdot 10^0 \pm 9.84 \cdot 10^{-1}$
Hollow ($D = 32$)	$-5.84 \cdot 10^0 \pm 1.05 \cdot 10^0$	$-5.95 \cdot 10^0 \pm 1.18 \cdot 10^0$	$-9.36 \cdot 10^0 \pm 1.06 \cdot 10^0$	$-1.02 \cdot 10^1 \pm 1.19 \cdot 10^0$
Hollow ($D = 64$)	$-5.83 \cdot 10^0 \pm 1.02 \cdot 10^0$	$-5.55 \cdot 10^0 \pm 1.24 \cdot 10^0$	$-8.74 \cdot 10^0 \pm 1.16 \cdot 10^0$	$-9.46 \cdot 10^0 \pm 9.42 \cdot 10^{-1}$
Funnel ($D = 10$)	$-4.11 \cdot 10^0 \pm 5.73 \cdot 10^{-1}$	$-1.96 \cdot 10^0 \pm 4.70 \cdot 10^{-1}$	$-1.95 \cdot 10^0 \pm 2.91 \cdot 10^{-1}$	$-1.94 \cdot 10^0 \pm 5.39 \cdot 10^{-1}$

D.4 RQ2.4: Bayesian Logistic Regressions

In Fig. 13, we show plots of some (unnormalized) conditionals for the logistic regression targets. Proper marginalization would require sampling with MCMC and or symbolic integration. We observe that most plots strongly resemble a simple Gaussian. As a result, negative parameters might not be beneficial for modeling these targets, which could explain why GMMs and SMMs perform similarly in Table 3. The ELBO is calculated using $p(\boldsymbol{x}) = \widetilde{p}(\boldsymbol{x})/Z_\pi$ where we use a ground truth Z_π when available and otherwise use $\widetilde{p}(\boldsymbol{x})$. We computed this ground truth for credit and sonar only using the adaptive tempered SMC implementation from the BlackJAX library (Cabezas et al., 2024) as for other datasets this proved less reliable - changing SMC hyperparameters led to different results. We used $5 \cdot 10^5$ samples (per SMC iteration t) and found that using more did not lead to different results.

D.5 RQ4: SMMs for Neuro-symbolic Targets

We now provide a quantitative comparison between SMMs and GMMs for the neuro-symbolic targets discussed in RQ4). We initialize Gaussian components in a grid covering the target and tune their initial standard deviation in $\{1, 2\}$ for *scenario 1* (Fig. 1) and $\{2, 4\}$ for *scenario 2* (Fig. 6). For $K = 2$, we set the initial component means to $(3, 5)$ and $(6, 5)$ respectively. For *scenario 1* (Fig. 1), we further adjust the bounds for the grid initialization based on K as follows: for $K = 4$, we set the limits for x_1 as $[2, 4]$ and for x_2 as $[3, 7]$; for higher values of K, they are set to $[1, 5.5]$ and $[1, 9]$ respectively. For *scenario 2* (Fig. 6), we set the bounds to $[2, 8]$ for x_1 and $[1, 7]$ for x_2 for $K > 2$.

Mixture weights are initialized as $\frac{1}{K}$ (where is the number of components). For SMMs, we also initialize complex weights by sampling from a Unif($[0, 1]$) distribution. We train with 3 random seeds using Adam (lr=0.01, weight decay=0.001 on mixture weights), a maximum of 15000 update steps with samples, and a patience of 1000 on the training loss. We select a model as described in App. C.3.

Since the target does not have informative gradients w.r.t. the inputs at constrained areas, we use the RLOO gradient estimator for *both* SMMs and GMMs as opposed to reparameterization. We learn the SMMs via rejection sampling as it has shown to provide a good tradeoff between learning efficiency and approximation quality in our synthetic experiments (Table 2). The models were trained and selected based on 10^4 samples. The ELBO values in App. D.5 were estimated from 10^5 samples; estimation was repeated 10 times and we report the resulting mean and standard deviation.

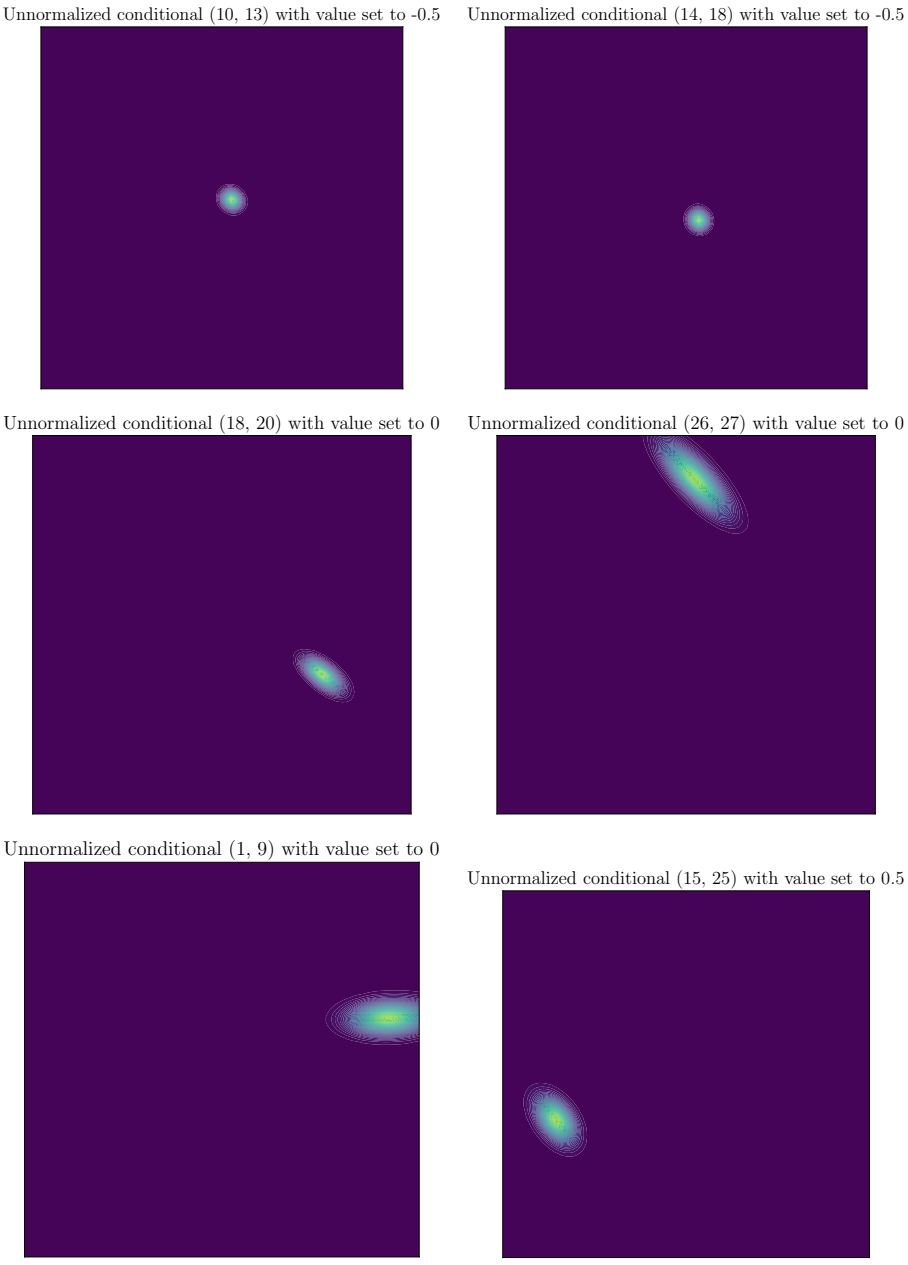


Figure 13: Some unnormalized bivariate conditionals for the GermanCredit (first row), BreastCancer (second row), and Ionosphere (last row) logistic regression targets. Each panel conditions all remaining features on evidence values mentioned in the caption.

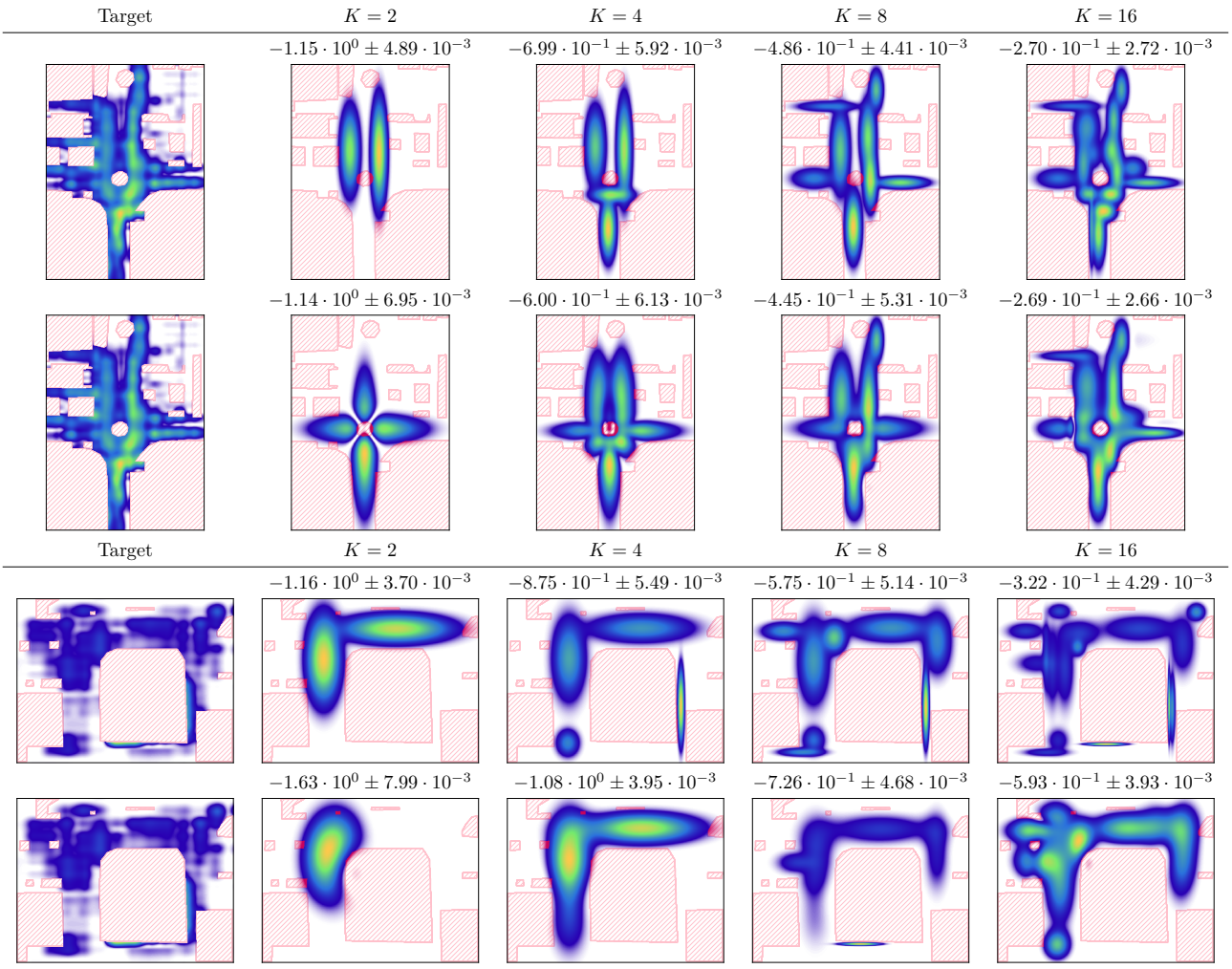


Figure 14: Additional results for the SDD targets. GMMs are shown in the top row, SMMs in the bottom row. The ELBO above each image was estimated from 10^5 samples. We report the mean and standard deviation over 10 repeated estimations. *Higher is better.*