# Efficient Mixture Learning in Black-Box Variational Inference

**Alexandra Hotti** [*1 2 3] **Oskar Kviman** [*1 2] **Ricky Molén** [1 2] **Víctor Elvira** [4] **Jens Lagergren** [1 2]

## Abstract

Mixture variational distributions in black box variational inference (BBVI) have demonstrated impressive results in challenging density estimation tasks. However, currently scaling the number of mixture components can lead to a linear increase in the number of learnable parameters and a quadratic increase in inference time due to the evaluation of the evidence lower bound (ELBO). Our two key contributions address these limitations. First, we introduce the novel Multiple Importance Sampling Variational Autoencoder (MISVAE), which amortizes the mapping from input to mixture-parameter space using one-hot encodings. Fortunately, with MISVAE, each additional mixture component incurs a negligible increase in network parameters. Second, we construct two new estimators of the ELBO for mixtures in BBVI, enabling a tremendous reduction in inference time with marginal or even *improved* impact on performance. Collectively, our contributions enable scalability to hundreds of mixture components and provide superior estimation performance in shorter time, with fewer network parameters compared to previous Mixture VAEs. Experimenting with MISVAE, we achieve astonishing, SOTA results on MNIST. Furthermore, we empirically validate our estimators in other BBVI settings, including Bayesian phylogenetic inference, where we improve inference times for the SOTA mixture model on eight data sets.

## 1. Introduction

Recent advancements in variational inference (VI) have focused on enhancing performance through more sophisticated network architectures, formulation of flexible priors and variational posteriors, and the exploration of alternative formulations of the evidence lower bound (ELBO), the typical objective function in VI. Competitive developments include normalizing flows (NFs; Rezende & Mohamed (2015); Papamakarios et al. (2021)), hierarchical models (Burda et al., 2015; Sønderby et al., 2016; Vahdat & Kautz, 2020), autoregressive models (Van Oord et al., 2016), the VampPrior (Tomczak & Welling, 2018), and the importance weighted ELBO (IWELBO; Burda et al. (2015)).

Lately, using mixture models as variational distributions has garnered increased attention (Nalisnick et al., 2016; Kucukelbir et al., 2017; Morningstar et al., 2021; Kviman et al., 2022; 2023a). Specifically, Kviman et al. (2022) developed a formulation of the ELBO for uniformly weighted mixtures inspired by multiple importance sampling (MIS; see Elvira et al. (2019) for a review), termed MISELBO,[1]

$$\mathcal{L}_{\text{MIS}} = \frac{1}{A}\sum_{a=1}^{A} \mathbb{E}_{q_{\phi_a}(z_a|x)}\left[\log \frac{p_\theta(x, z_a)}{\frac{1}{A}\sum_{a'=1}^{A} q_{\phi_{a'}}(z_a|x)}\right], \tag{1}$$

where $z_a$ is a latent variable, $x$ is observed data, $\theta$ represents the parameters of the generative model $p_\theta(x, \cdot)$, $A$ is the number of mixture components, and $\phi_a$ denotes the variational parameters of the $a$-th mixture component $q_{\phi_a}(\cdot|x)$.

Mixtures are distinguished by their simple yet expressive nature, as well as their theoretical foundation. In the BBVI setting, they have achieved state-of-the-art (SOTA) results in applications like image processing and phylogenetics (Kviman et al., 2023a;b). However, computational complexities (parameter costs and inference times) hinder algorithm developers from utilizing a large $A$, ultimately leaving the full potential (e.g., their universal approximator property (Kostantinos, 2000)) untapped.

In BBVI-based mixture learning, the number of learnable parameters typically increases linearly with $A$ at an unpractical rate. For example, in Kviman et al. (2022; 2023a), a naive approach is used where each new component allocates a separate encoder network, while in Kviman et al. (2023b), there is one Bayesian network per component. Moreover,

[*]Equal contribution [1]KTH Royal Institute of Technology [2]Science for Life Laboratory [3]Klarna [4]University of Edinburgh. Correspondence to: Alexandra Hotti <hotti@kth.se>, Oskar Kviman <okviman@kth.se>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

[1]An alternative naming of this lower bound emerges in the work of Morningstar et al. (2021), where the sampling scheme is interpreted as stratified sampling. This perspective leads to the names SELBO or SIWELBO.

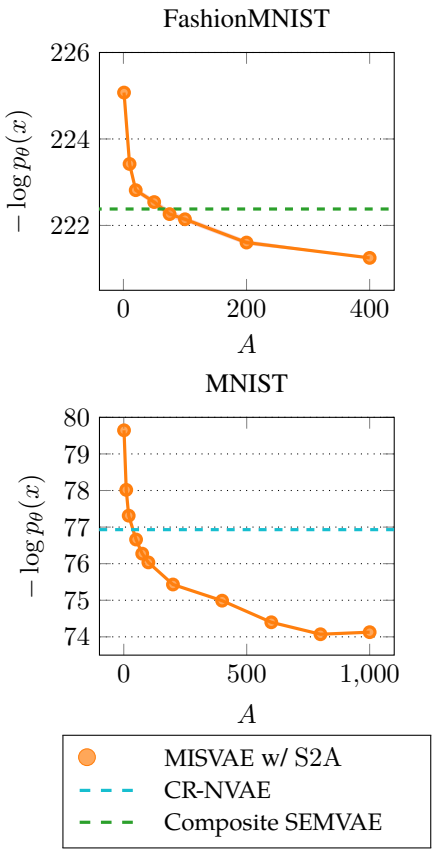



Figure 1: **SOTA Performance with Small and Efficient Networks:** NLL values for MISVAE trained with the S2A estimator with $S = 1$ and a gradually increasing $A$.

by inspecting Eq. (1), it is clear that the evaluation of the MISELBO objective, and thus the inference time, scales quadratically with $A$.

We make two contributions in this work, each addressing the aforementioned computational complexities. First, we introduce the Multiple Importance Sampling VAE (MISVAE), a novel VAE architecture that efficiently amortizes the mapping from data to mixture parameters (see Fig. 2). With our one-hot-encoding-based parameterization strategy, all network weights in a single encoder are shared among the $A$ mixture components. This is a novel construction, as, previously, either no (Kviman et al., 2022; 2023a) or only a subset (Nalisnick et al., 2016) of the encoder weights have been shared. MISVAE is described in detail in Sec. 5.

Our second contribution is inspired by the plethora of techniques for sampling from mixture models, from the MIS literature (Elvira et al., 2019). To make the evaluation of MISELBO objective more effective, we extend two established MIS schemes to develop two novel estimators of the MISELBO: the Some-to-All (S2A) and Some-to-Some (S2S) estimators. Both estimators sample a subset of $S < A$ unique components, from which the latent variables are subsequently simulated from. This results in estimations of a subset of the expectations in Eq. (1). The two estimators differ, however, in their formulation of the denominator in Eq. (1) and, thus, in their theoretical properties. In Sec. 4, we clearly explain how to implement the estimators, how they relate to popular MIS schemes, provide their respective time complexities, and give robust theoretical justifications of both estimators.

We have constrained our work to uniformly weighted mixtures. This is justified by existing analyses in the BBVI literature. Specifically, Morningstar et al. (2021) observed that inferring parameters for *weighted* mixture components often leads to mode collapse. To mitigate this issue, they suggested using the Importance Weighted ELBO (IWELBO; Burda et al. (2015)). However, using the IWELBO objective presents other challenges. Notably, it does not allow for training VAEs using the KL warm-up scheme, which is crucial for achieving SOTA NLL results (see e.g., Tomczak & Welling (2018)). Additionally, this approach can adversely affect the learning of encoder nets (Rainforth et al., 2018).

The S2A and S2S estimators are applicable to any BBVI mixture problem (i.e., not constrained to VAEs), and, as we demonstrate in various BBVI scenarios in Sec. 6, they can heavily decrease the parameter inference time. For VAEs, when paired with MISVAE, we can push the limits of $A$ in order to achieve astonishing marginal log-likelihood results on MNIST and FashionMNIST (see Fig. 1).

To summarize, our contributions are

- **MISVAE:** We introduce MISVAE. A novel Mixture VAE architecture that efficiently maps data to mixture parameters, significantly improving the scalability w.r.t. the number of mixture components, $A$.
- **Some-to-Some:** We propose S2S, a novel estimator of MISELBO (Eq. (1)). This estimator enables enhanced performance relative to MISVAE by allowing an increase in $A$, while preserving the same inference time per epoch.
- **Some-to-All:** We introduce S2A, which we prove to be an unbiased estimator of MISELBO for any $S < A$. This approach makes it possible to increase the total number of mixtures $A$ with only a small additional computational burden.

## 2. Related Work

Mixture VAEs can be traced back to Nalisnick et al. (2016), who employed a Gaussian mixture model with a Dirichlet prior on the mixture weights, and inferred these weights through a neural network mapping from the data, $x$, to the

simplex. Their encoder employed $A$ separate mappings from a hidden layer in the encoder to the different mixture parameters. As such, their architecture scales poorly to large $A$ in terms of number of network parameters. Furthermore, Roeder et al. (2017) utilized a weighted mixture ELBO, for training a VAE using stop gradients and different sampling strategies. Yet, their application was limited to a toy dataset.

Drawing inspiration from the MIS literature, Kviman et al. (2022) introduced the concept of the MISELBO, offering a straightforward method for evaluating the mixture ELBO. However, in their approach, individual mixture components were trained separately and aggregated only during the evaluation of the MISELBO objective. As a result, each component tended to gravitate towards the mode of the posterior, rather than all components collaboratively covering all regions of the posterior. Expanding on these concepts, Kviman et al. (2023a) devised methods for the joint training of all mixture components, which were subsequently applied by Kviman et al. (2023b). Collectively, achieving SOTA performance on datasets such as MNIST, FashionMNIST, and a range of phylogenetic datasets. Nevertheless, despite their impressive empirical performance, scaling these architectures to variational mixtures with more than ten components posed significant computational challenges. Here, we address this limitation, enabling the full potential of variational mixtures to be realized by significantly reducing the computational demands of scaling to a larger number of components. The idea of increasing the number of mixture components for variance reduction while limiting the computational complexity in MIS was introduced in (Elvira et al., 2015) and further developed in (Elvira et al., 2016a;b).

## 3. Background

**Estimating NLL** The estimate of the IWELBO,

$$\mathcal{L}^{L}_{\text{IWELBO}} = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{1}{L}\sum_{\ell=1}^{L}\frac{p_\theta(x,z_\ell)}{q_\phi(z_\ell|x)}\right], \qquad (2)$$

where $L$ is the number of importance samples, is often used to estimate the marginal log-likelihood, $\log p_\theta(x)$. The *negative log-likelihood* (NLL) refers to - $\log p_\theta(x)$. It is possible to estimate the NLL by using an importance-weighted version of MISELBO (Kviman et al., 2022).

**MIS** In the field of importance sampling (IS), MIS refers to techniques with more than one proposal/importance sampler (Elvira & Martino, 2021). MIS inherits strong theoretical guarantees and many methodological developments have been made (Veach & Guibas, 1995; Owen & Zhou, 2000; Sbert & Elvira, 2022). In this line of research, we rely on MIS schemes where the sampling is done either from a mixture or by deterministically choosing the proposal from a set of mixture components. In both cases, the weights are constructed in a way that reduces variances of the IS estimators (see Elvira et al. (2019) for more details).

Due to the logarithm in the ELBO, many theoretical insights gathered in MIS do not generalize to VI. However, recognizing that MISELBO is an expectation to be estimated by a mixture establishes clear connections between popular mixture sampling techniques, as described in Elvira et al. (2019), and the estimators used in BBVI mixture learning.

**Bayesian phylogenetics** In Bayesian phylogenetic inference, the posterior distribution over branch lengths and tree topologies is approximated jointly, given the observed sequence data (e.g., DNA). In Appendix E.3.1, we define the posterior and give more details on the generative model.

Many recent works have applied modern machine learning techniques to Bayesian phylogenetics (Zhang et al., 2018; Zhang, 2020b; Moretti et al., 2021; Zhang, 2023; Zhou et al., 2023). Notably, Kviman et al. (2023b) constructed a mixture of variational phylogenetic posterior approximations, achieving SOTA results.

However, in Kviman et al. (2023b), the inference time scales poorly with $A$, making it infeasible to learn mixture models with many components. In Sec. 6, we instead apply our two new estimators for learning the mixture parameters, decreasing the computational costs of the SOTA BBVI method. This takes the field closer to realistic application of mixture models in Bayesian phylogenetic and domains with even larger state spaces. These applications have traditionally been viewed as computationally demanding and, in practice, considered intractable within a Bayesian framework (e.g., species-tree reconciliation (Åkerborg et al., 2009))

## 4. Efficient Estimation of MISELBO

We now walk through the three approaches we consider for estimating Eq. (1), namely the All-to-All, and our two novel estimators, the Some-to-All, and Some-to-Some estimators.

**All-to-All** When implementing the All-to-All (A2A) estimator, a single latent variable is sampled from each of the $A$ available components, resulting in

$$\widetilde{\mathcal{L}}_{\text{A2A}} = \frac{1}{A}\sum_{a=1}^{A}\log\frac{p_\theta(x|z_a)p_\theta(z_a)}{\frac{1}{A}\sum_{a'=1}^{A}q_{\phi_{a'}}(z_a|x)}, \qquad (3)$$

where $z_a \sim q_{\phi_a(z_a|x)}$.

The computational cost of this estimator is proportional $A \times A$ and it connects to the N3 scheme in Elvira et al. (2019) since all $A$ components are used in the simulation of the $S = A$ samples and also appear in the denominator.

**Some-to-All** There are $A$ components in total, and for a given data point (or batch) we sample a subset, $\Phi$, of $S$

unique components (without replacement). No component is more likely to be selected *a priori*, and so we can consider sampling the subsets from a uniform distribution over all $\binom{A}{S}$ possible subsets, $\varphi(\Phi)$ (see Appendix A). Then, by obtaining $\Phi \sim \varphi(\Phi)$, we construct the Some-to-All (S2A) estimator

$$\widetilde{\mathcal{L}}_{\text{S2A}} := \frac{1}{S}\sum_{s=1}^{S} \log \frac{p_\theta(x|z_s)p_\theta(z_s)}{\frac{1}{A}\sum_{a=1}^{A} q_{\phi_a}(z_s|x)}, \tag{4}$$

where $z_s \sim q_{\phi_s(z_s|x)}$ for all $\phi_s \in \Phi$.

This estimator is linked to the R3 scheme in Elvira et al. (2019) since in both cases a subset of components is used to simulate the samples, while the unweighted mixture of all components appear in the denominator. The difference is that the R3 scheme samples exactly $S = A$ samples by selecting the components with multinomial resampling with replacement, while our S2A estimator samples $S < A$ samples by selecting the components with multinomial resampling without replacement instead.

A beneficial property of the S2A estimator is that it allows for sampling mixture components without replacement, resulting in a lower variance gradient estimator compared to when sampling with replacement. Moreover, the computational cost of the S2A estimator is $S \times A$ and it is an unbiased estimator of Eq. (1).

**Theorem 4.1.** *The Some-to-All estimator is an unbiased estimator of Eq.* (1)*.*

*Proof.* See Appendix A. □

Furthermore, its expectation is a lower bound on the marginal log-likelihood.

**Corollary 4.2.** *The expected value of the Some-to-All estimator is a lower bound on the marginal log-likelihood,*

$$\mathbb{E}\left[\widetilde{\mathcal{L}}_{S2A}\right] \leq \log p_\theta(x).$$

We leverage the unbiased property of S2A to substantially reduce the complexity involved in estimating the MISELBO objective. The computational cost of this estimator is proportional to $S \times A$, however, given that it is unbiased, we can keep $S$ small and instead increase $A$.

Although the focus of our work is on uniformly weighted components, we generalize Theorem 4.1 to hold for arbitrary mixture weights.

**Theorem 4.3.** *The Some-to-All estimator is an unbiased estimator of MISELBO for arbitrary mixture weights.*

*Proof.* See Appendix B. □

**Some-to-Some** The next estimator is inspired by the *a priori* partitioning approach in Elvira et al. (2019, Section 7.2). For a given data point, we, again obtain a $\Phi \sim \varphi(\Phi)$, where $|\Phi| = S$. In contrast to the S2A estimator, we here only evaluate the simulated latents on this subset of components, and we get the Some-to-Some (S2S) estimator

$$\widetilde{\mathcal{L}}_{\text{S2S}} := \frac{1}{S}\sum_{s=1}^{S} \log \frac{p_\theta(x|z_s)p_\theta(z_s)}{\frac{1}{S}\sum_{\phi_{s'}\in\Phi} q_{\phi_{s'}}(z_s|x)}, \tag{5}$$

where $z_s \sim q_{\phi_s(z_s|x)}$ for all $\phi_s \in \Phi$. The cost of computing this estimator is $S \times S$. Letting $S = 1$ is equivalent to inferring the parameters of an ensemble of variational approximations (Kviman et al., 2022).

The S2S estimator also connects with the R2 scheme in Elvira et al. (2019) since in both cases a subset of components is used to simulate the samples and this same subset of components appear in the denominator. Again, the difference is that the R2 scheme samples exactly $S = A$ samples by selecting the components with multinomial resampling with replacement, while our S2S samples $S < A$ samples by selecting the components with multinomial resampling without replacement.

**Theorem 4.4.** *The expected value of the Some-to-Some estimator is a lower bound on MISELBO, i.e.,*

$$\mathbb{E}\left[\widetilde{\mathcal{L}}_{S2S}\right] \leq \mathcal{L}_{\text{MIS}}.$$

*Proof.* See the supplementary material. □

**Corollary 4.5.** *The expected value of the Some-to-Some estimator is a lower bound on the marginal log-likelihood,*

$$\mathbb{E}\left[\widetilde{\mathcal{L}}_{S2S}\right] \leq \log p_\theta(x).$$

The S2S estimator has the smallest computational cost among the three presented here. However, what it gains in speed it trades off in joint inference among the mixture components—a component that is not in $\Phi$ will not affect the inference of $\phi_s \in \Phi$, which should affect cooperation. Yet, given a fixed $S$, we can improve performance by increasing $A$ without additional computational burden. This is because the S2S estimator can be viewed as an ensemble of mixtures, where we have access to $A$ models and select $S$ components for each instance of the ensemble.

**Summary** Our two new estimators have lower computational complexity than the A2A estimator. As we will demonstrate, the benefits gained in practice in terms of runtime will be particularly important if the numerator (the generative model) of Eq. (1) is expensive to compute. Comparing S2A and S2S, the latter will enjoy a shorter inference time if the entropy, or, alternatively, the denominator in Eq. (1), is expensive to compute.

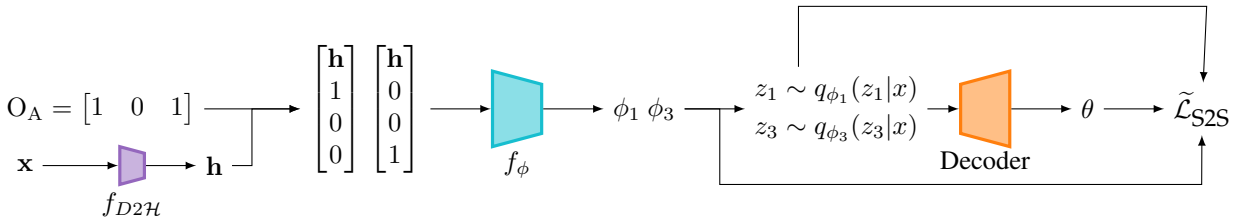



Figure 2: Block diagram depicting the estimation of MISELBO using **MISVAE** with the S2S estimator, with $S = 2$ and $A = 3$. First, $f_{D2\mathcal{H}}$ maps the data to an intermediate hidden space, producing a representation $h$. The next network, $f_\phi$, takes $h$ along with $S$ $A$-dimensional one-hot encodings, acting as signals of the $S$ mixtures used by the S2S estimator, as input, which are then mapped to the variational parameters, here $\phi_1$ and $\phi_2$, of the mixture components. Samples drawn from the $S$ mixtures are then passed to a decoding network to produce the parameters $\theta$ of the generative model. Collectively, the sampled latent variables, the variational parameters, and $\theta$ , are used to compute $\widetilde{\mathcal{L}}_{\text{S2S}}$. The diagram is explained in detail in Sec. 5. Corresponding diagrams for the S2A and A2A estimators can be found in Fig. 7.

## 5. Multiple Importance Sampling VAE

Impressive results have been obtained by naively expanding the parameter space with the number of mixtures. However, by carefully studying the problem at hand, similar, or even improved, performance gains can be achieved at negligible increases in parameter costs.

MISVAE is a new Mixture VAE architecture featuring an encoder network composed of two consecutive networks. The novelty of MISVAE lies in the second network, which parameterizes the mixture components using amortization.

The first network maps the data to an intermediate (deterministic) hidden space, $\mathcal{H}$; we refer to it as the D2$\mathcal{H}$ net and denote the function as $f_{D2\mathcal{H}}$. The second net is a mapping from the Cartesian product of $\mathcal{H}$ and the space of $A$-dimensional one-hot encodings, $\mathcal{O}_A$, to the parameters of a mixture component. We denote this as $f_\phi$ and call it the amortized mixture parameterization (AMP) net. We write

$$f_{D2\mathcal{H}} : \mathcal{X} \mapsto \mathcal{H}, \quad f_\phi : \mathcal{H} \times \mathcal{O}_A \mapsto \mathcal{Q}, \tag{6}$$

or, alternatively, $h = f_{D2\mathcal{H}}(x)$ and if $\mathcal{Q}$ is the family of Gaussians, $(\mu(h, \phi_s), \sigma(h, \phi_s)) = f_\phi(h, o_A(s))$, where $o_A(s)$ is an $A$ long one-hot encoding with the $s$-th element set to one.

The AMP net, $f_\phi$, is a sequence of neural networks (NNs), shared among all mixture components. The NNs take $o_A(s)$ as their biases. That is, to get the parameters of the $s$-th component, we pass $o_A(s)$ as a bias to the NNs. See Fig. 2 for a depiction of the MISVAE architecture.

## 6. Experiments

In this section, we infer variational parameters using MISVAE along with the S2S, S2A, and A2A estimators. We conduct comparisons among these methods and against SOTA approaches across a synthetic dataset, three image datasets, and eight phylogenetic datasets. All code necessary to replicate our experiments is publicly available at: `https://github.com/okviman/efficient-mixtures`.

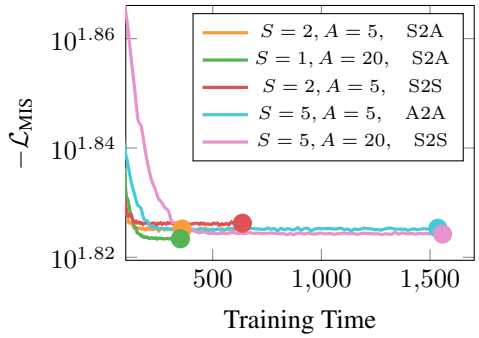



Figure 3: Comparison of MISELBO approximation performance and training runtimes across three distinct estimators under various settings of $S$ and $A$ in the Toy Experiment, trained for $50,000$ epochs.

### 6.1. Toy Example

Here we construct a toy example in order to evaluate the performances of the estimators when the posterior exhibits certain properties. Concretely, we design a generative model that is non-Gaussian, has an autoregressive likelihood function, and assume that the $A$ terms in the energy term, $\frac{1}{A}\sum_{a=1}^{A} \mathbb{E}_{q_{\phi_a}(z_a|x)}[\log p_\theta(z_a, x)]$, have to be computed sequentially.

These criteria are of interest as they naturally arise in many settings, including all our real-data experiments below: the posteriors are non-Gaussian, both the pixel-CNN decoder and the likelihood function in Bayesian phylogenetics are autoregressive, and the corresponding energy terms are not always parallelizable——the CIFAR-10 images are too big to parallelize over $A$ with a reasonable batch size, and the standard implementation of the dynamic programming re-

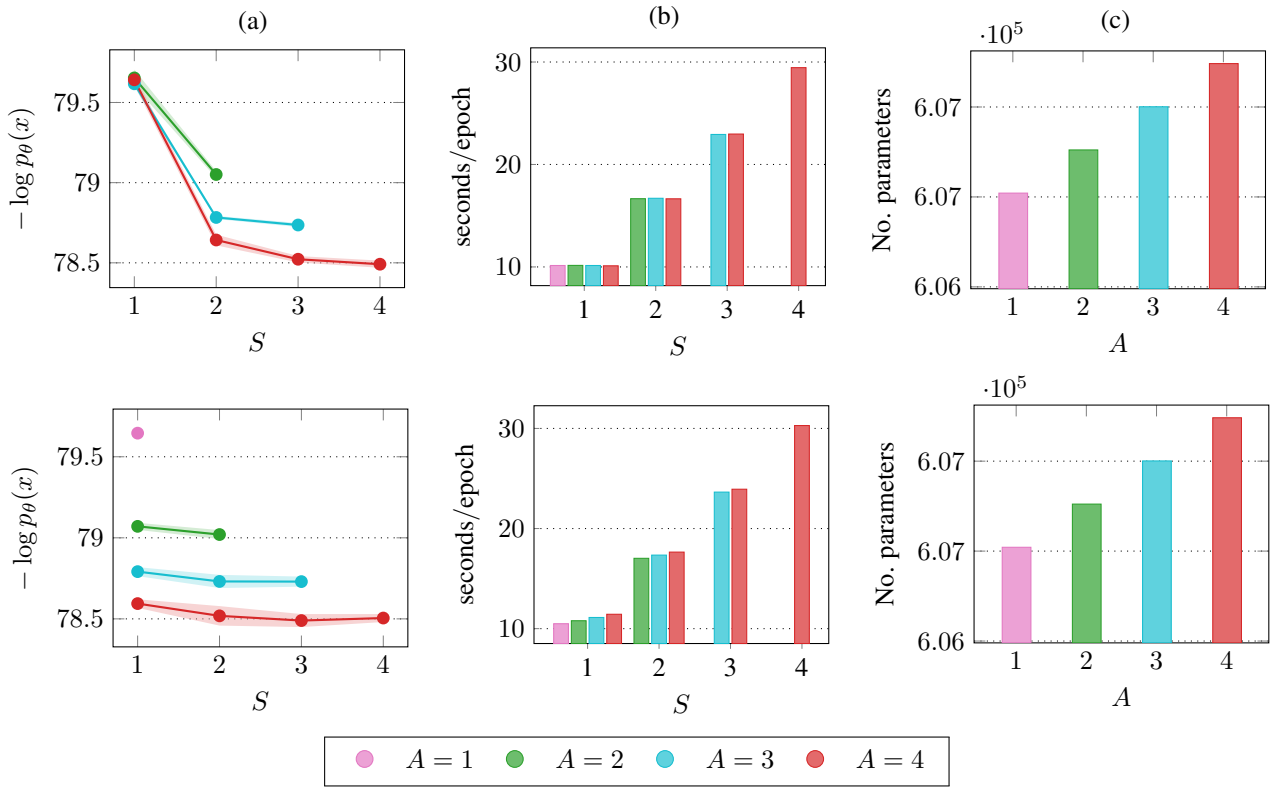



Figure 4: Results on **MNIST** for MISVAE trained with various combinations of $S$ and $A$, with the S2S estimator (top row) and the S2A estimator (bottom row). (a) Average (solid) NLL results computed over three runs with one standard deviation (opaque) displayed, (b) training time per epoch, and (c) the number of network parameters for MISVAE for increasing values of $A$. Using MISVAE, the number of network parameters increases by a small amount as we increase $A$. Also, with the S2S estimator, we can keep $S$ fixed and increase $A$, without impacting the number of seconds needed to complete an epoch and simultaneously improving the NLL. For S2A, we converge to an equivalent solution with $A$ held fixed for any $S < A$, meaning that in practice, we can scale up $A$ for small values of $S$ at a small extra computational cost per mixture component.

quired to compute the phylogenetic likelihood function does not necessarily parallelize.

With these criteria in mind, we let $p(x|z) = \prod_{n=1}^{N} p(x_n|z)$, where $x_n$ is the $n$-th $d_x$-dimensional data point, $N$ is the number of generated data points and $p(x_n|z) = \prod_{i=1}^{d_x} \text{Bernoulli}\left(x^{(i)} \middle| \text{sigmoid}\left(\theta^{(i)} + \sum_{j=1}^{i-1} \beta^{i-j} x^{(j)}\right)\right)$, with $\beta = 0.1$, superindices are within parentheses, and $\theta = Wz$, where

$$W \in \mathbb{R}^{d_x \times d_z}, \quad W_{u,v} \sim \log \mathcal{N}(0, 0.1). \tag{7}$$

Finally, as prior we use the hierarchical *Neal's funnel* model

$$p(z_2, z_1) = \mathcal{N}(z_2|0, e^{z_1/2})\mathcal{N}(z_1|0, 3). \tag{8}$$

We constrain our analysis to a 2-dimensional latent space for visualization purposes. An unnormlized posterior when $d_x = 20$ and $N = 5$ is depicted in Fig. 8.

**Results.** The posterior is non-Gaussian (and intractable) and has an autoregressive likelihood function. We artificially force the energy term to be computed sequentially, let $d_x = 20$ (a moderately large number) and chose $N = 5$. In Fig. 8 we visualize the unnormalized posterior when $d_x = 20$. All mixture components in the variational approximations are Gaussians with diagonal covariance matrices.

The purpose of this experiment is to compare the MISELBO values and total runtimes across different estimators, as well as to visualize the final variational approximations in the latent space. Accordingly, all models were subjected to training over $50,000$ epochs. Fig. 3 presents the evolution of MISELBO during the training process for these estimators, alongside the total runtime required to finish the epochs. If $d_x$ was to be further increased, so would the cost of evaluating the likelihood. Hence, we expect our new estimators to provide good approximations in shorter runtime than the A2A estimator when $d_x$ is sufficiently high.

The results in Fig. 3 confirm our expectations. When $(S = 2, A = 5)$, the S2A and A2A estimators achieve the same MISELBO scores, however, S2A requires only a

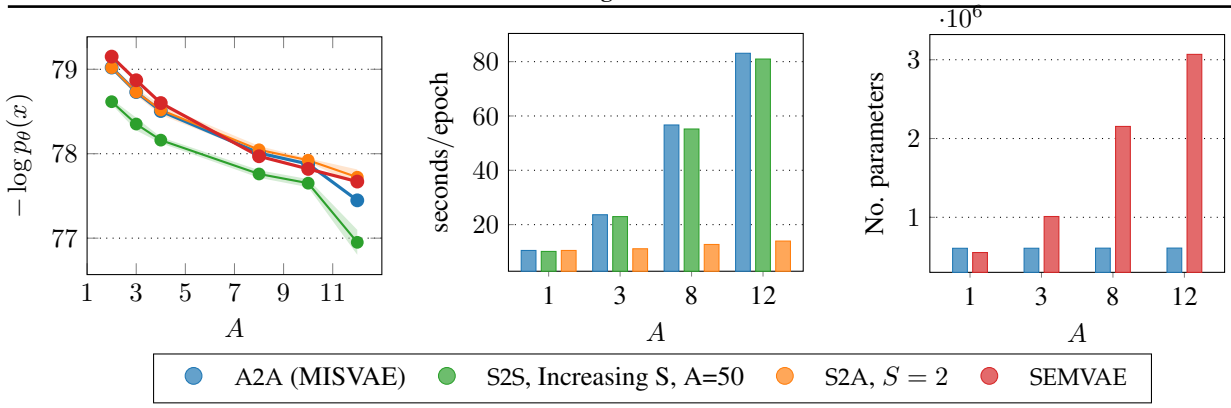



Figure 5: Comparison between SEMVAE and MISVAE using the S2S, A2A, and S2A estimators on **MNIST**: (a) NLL scores for increasing values of $A$, (b) training time per epoch, and (c) number of hyperparameters for increasing $A$ for SEMVAE compared to MISVAE. Note: The green curve represents the performance of MISVAE using the S2S estimator with $S$ increasing , such that S=A on the x-axis, while $A$ is held fixed at 50.

fraction of the runtime. Meanwhile, for the said $S$ and $A$, S2S performs worse in terms of MISELBO score, albeit in short runtime. Interestingly, as S2S can be regarded as an ensemble, S2S with $(S = 5, A = 20)$ can outperform A2A $(A = 5)$ in terms of MISELBO scores in approximately the same training time per epoch. Finally, as the S2A is an unbiased estimator of MISELBO for any $S < A$, it excels when utilizing a large number of mixtures $A$ with a small $S$. Notably, the configuration of S2A with $(S = 1, A = 20)$ not only boasts the fastest training time per epoch but also achieves the lowest overall negative MISELBO scores.

In Appendix E.1.1 we include further implementation details and visualizations of the approximations in the latent space.

### 6.2. Image Data

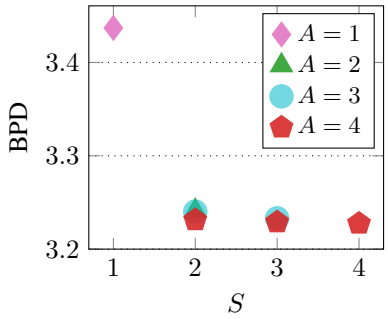
To make our results comparable to current SOTA mixture architectures, we use the same experiment setup and training-related hyperparameters as in Kviman et al. (2023a). We refer to the benchmark MISVAE with the Mixture VAE used in Kviman et al. (2023a), where separate encoder networks are used for each mixture component, as the separate encoder Mixture VAE (SEMVAE). We train on MNIST (LeCun & Cortes, 2010), FashionMNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky et al.). Additional details are provided in Appendix E

**Results.** The experiments conducted with real datasets confirm the insights gained from the Toy Experiment. In Figures 4 (MNIST) and 6 (CIFAR-10), we compare the NLL scores of MISVAE when trained using the S2S, S2A, and A2A estimators across progressively increasing values of $S$ and $A$. Note that the A2A estimator is used when $S = A$. We make two observations. First, for a given $S$, the S2S estimator consistently achieves lower NLL scores as $A$ increases, with this effect being more pronounced

Table 1: NLL statistics for SOTA VAE architectures on **MNIST**. The Composite model is a SEMVAE model with hierarchical models, NFs and the VampPrior. For IWAE, $L$ is the number of importance samples used during training.

| Model | NLL | No. Parameters | Seconds/epoch |
|---|---|---|---|
| IWAE ($L = 20$) (Burda et al., 2016) | 79.63 | 720,541 | 128.02 |
| Hierarchical VAE w. VampPrior (Tomczak & Welling, 2018) | 78.45 | 1,777,821 | - |
| NVAE (Vahdat & Kautz, 2020) | 78.01 | 33,363,134 | - |
| MAE (Ma et al., 2019) | 77.98 | 1,565,570 | |
| Ensemble NVAE (Kviman et al., 2022) | $77.77 \pm 0.2$ | - | - |
| Vanilla SEMVAE ($S = 12$; Kviman et al. (2023a)) | 77.67 | 8,549,344 | - |
| Composite SEMVAE ($S = 4$; Kviman et al. (2023a)) | $77.23 \pm 0.1$ | 5,212,065 | - |
| CR-NVAE (Sinha & Dieng, 2021) | 76.93 | - | - |
| MISVAE S2S ($A = 50, S = 20$; **our**) | 76.67 | **618,781** | 132.46 |
| MISVAE S2A ($A = 200, S = 1$; **our**) | 75.43 | 654,781 | **73.06** |
| MISVAE S2A ($A = 800, S = 1$; **our**) | **74.07** | 798,781 | 261.71 |

(a) Some-to-Some




(b) Some-to-All

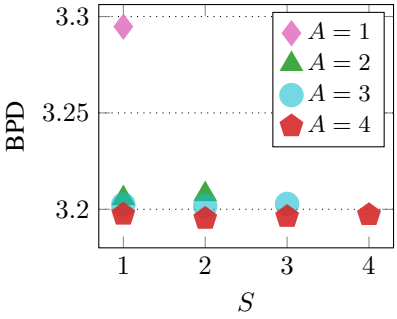



Figure 6: BPD results on **CIFAR-10** for MISVAEs trained with different estimators and combinations of $S$ and $A$.

for MNIST. Second, as depicted in Fig. 4b, the epoch completion time for the S2S estimator remains constant as $A$ increases, assuming $S$ is fixed. These observations collectively suggest the potential for improving performance by significantly increasing $A$ while maintaining a small $S$. This hypothesis is further examined in Fig. 13b, which shows that initially, increasing $A$ enhances performance. However, for large enough $A$, the performance gains start to wane, likely due to the decreasing probability of each component being updated in an epoch as $A$ increases.

We also make some useful observations regarding the S2A estimator. For a given value of $A$, the S2A estimator demonstrates approximately equivalent performance as $S$ varies, confirming its unbiasedness on both CIFAR-10 (Fig. 6b) and MNIST (Fig. 4a). Additionally, from Fig. 4b, we observe a marginal increase in epoch completion time for a fixed $S$ with increasing $A$. The combination of being unbiased and efficient suggests that the S2A estimator can be scaled to a substantially larger number of mixture components. In Fig. 1, we gradually increase $A$ up to hundreds of mixture components with $S = 1$ held fixed and achieve SOTA NLL scores on MNIST and FashionMNIST. In Table 1, we compare the number of network parameters and the NLL scores of our models against other competitive models in this domain. Notably, our models use far fewer network parameters at superior performance in terms of NLL.

Table 2: FID scores evaluated on the MNIST test set

| S | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| FID | 10.87 | 10.02 | 9.89 | 9.70 |

Finally, we compare MISVAE and our estimators against SEMVAE in Fig. 5, which reveals that S2A, even with $S = 2$ held constant, either outperforms or matches the performance of SEMVAE. Also, it achieves this with only a slight increase in inference time and network parameters as $A$ increases, unlike SEMVAE, where epoch completion time and network parameters escalate rapidly with $A$. We also see that S2S outperforms both S2A, albeit being considerably slower, and A2A, with approximately the same inference time. The superior performance of the S2S estimator is because it can be regarded as an ensemble of mixtures, thus providing a tighter bound compared to a single mixture model, as proven by Kviman et al. (2022).

**Generative performance.** In order to understand how the decoder contributes to the impressive NLL scores, we trained four MISVAEs using S2A with $A = 4$ and $S = 1, \dots, 4$. For each model, we evaluated the decoders generative performance via the FID score on the MNIST test dataset. The results in Table 2 demonstrate that the generative performance of the decoder increases when learned with increasingly expensive estimators. These results likely stem from the frequency of likelihood function evaluations. I.e., when $S \leq A$, the likelihood function is evaluated less frequently during gradient calculations with respect to the decoder weights. In Appendix E.4 we also include visualizations of generated images from MNIST and FashionMNIST.

### 6.3. Bayesian Phylogenetics

Kviman et al. (2023b) achieved SOTA results by developing a mixture of variational phylogenetic posterior approximations, learnt via the A2A estimator. As empirically demonstrated in (Kviman et al., 2023a;b), the NLL improves monotonically with an increasing number of components. Using a large number of mixtures in mixture variational Bayesian phylogenetic inference (VBPI), however, lead to significant computational overhead and slow convergence, especially with large datasets. We address these issues with the S2A and S2S estimators, showing that they achieve comparable results with reduced computational requirements. Overall, our estimators enable us to minimize computational time without compromising performance.

We precisely followed the training procedure outlined by Kviman et al. (2023b) and, like them, adopted the VBPI algorithm with NFs and mixture distributions. We performed experiments on eight popular datasets for Bayesian phylo-

Table 3: NLL estimates on eight Phylogenetic Datasets. All VBPI methods use 1000 importance samples, and the results are averaged over 100 runs and three independently trained models. The time evaluation for the likelihood function ($p$) and variational probability ($q$) was conducted on an i5-1130G7 CPU using one core using a CPU timer. #$p$ and #$q$ describe the number of times the likelihood and variational distribution needs to be evaluated respectively.

| Data | | | | DS1 | DS2 | DS3 | DS4 | DS5 | DS6 | DS7 | DS8 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| # Taxa | | | | 27 | 29 | 36 | 41 | 50 | 50 | 59 | 64 |
| # Sites | | | | 1949 | 2520 | 1812 | 1137 | 378 | 1133 | 1824 | 1008 |
| Run time $p$ (ms) | | | | 0.765 | 1.007 | 1.249 | 1.235 | 1.481 | 1.622 | 1.821 | 1.954 |
| Run time $q$ (ms) | | | | 0.778 | 1.082 | 1.425 | 1.427 | 1.816 | 1.652 | 2.130 | 2.359 |
| $A$ | | #$p$ | #$q$ | **VBPI with NFs and Mixtures using (A2A)** (Kviman et al., 2023b) | | | | | | | |
| 1 | | 1 | 1 | 7108.42 (.15) | 26367.72 (.06) | 33735.10 (.07) | 13330.00 (.23) | 8214.70 (.47) | 6724.50 (.45) | 37332.01 (.27) | 8650.68 (.46) |
| 2 | | 2 | 4 | 7108.40 (.10) | 26367.71 (.04) | 33735.10 (.05) | 13329.95 (.15) | 8214.62 (.26) | 6724.44 (.32) | 37331.96 (.19) | 8650.56 (.33) |
| 3 | | 3 | 9 | 7108.40 (.06) | 26367.70 (.03) | 33735.09 (.04) | 13329.94 (.11) | 8214.56 (.22) | 6724.40 (.23) | 37331.96 (.15) | 8650.54 (.30) |
| 4 | | 4 | 16 | 7108.40 (.05) | 26367.71 (.02) | 33735.09 (.02) | 13329.93 (.07) | 8214.57 (.16) | 6724.31 (.19) | 37331.92 (.13) | 8650.66 (.24) |
| $A$ | $S$ | #$p$ | #$q$ | **VBPI with NFs and Mixtures using (S2A)** | | | | | | | |
| 2 | 1 | 1 | 2 | 7108.54 (.11) | 26367.71 (.04) | 33735.10 (.05) | 13329.97 (.16) | 8214.92 (.44) | 6724.49 (.35) | 37331.99 (.19) | 8651.32 (.42) |
| 3 | 1 | 1 | 3 | 7108.77 (.08) | 26367.71 (.03) | 33735.10 (.04) | 13330.05 (.12) | 8215.34 (.42) | 6724.49 (.29) | 37331.98 (.15) | 8651.85 (.35) |
| 3 | 2 | 2 | 6 | 7108.44 (.08) | 26367.71 (.03) | 33735.09 (.04) | 13329.94 (.10) | 8214.62 (.28) | 6724.42 (.27) | 37331.95 (.17) | 8650.76 (.29) |
| 4 | 1 | 1 | 4 | 7108.92 (.09) | 26367.71 (.02) | 33735.09 (.03) | 13330.05 (.11) | 8217.98 (.73) | 6724.61 (.28) | 37331.96 (.13) | 8652.01 (.32) |
| 4 | 2 | 2 | 8 | 7108.40 (.06) | 26367.70 (.03) | 33735.10 (.03) | 13329.97 (.09) | 8214.69 (.21) | 6724.41 (.22) | 37331.98 (.14) | 8650.85 (.25) |
| 4 | 3 | 3 | 12 | 7108.40 (.05) | 26367.71 (.02) | 33735.09 (.03) | 13329.93 (.08) | 8214.59 (.17) | 6724.38 (.21) | 37331.96 (.11) | 8650.72 (.22) |
| $A$ | $S$ | #$p$ | #$q$ | **VBPI with NFs and Mixtures using (S2S)** | | | | | | | |
| 2 | 1 | 1 | 1 | 7108.41 (.10) | 26367.71 (.04) | 33735.09 (.06) | 13330.00 (.16) | 8214.77 (.38) | 6724.53 (.32) | 37332.00 (.20) | 8650.65 (.37) |
| 3 | 1 | 1 | 1 | 7108.41 (.08) | 26367.71 (.04) | 33735.09 (.05) | 13330.02 (.16) | 8214.90 (.35) | 6724.53 (.26) | 37332.00 (.18) | 8650.97 (.30) |
| 3 | 2 | 2 | 4 | 7108.42 (.08) | 26367.71 (.03) | 33735.09 (.04) | 13329.96 (.11) | 8214.63 (.23) | 6724.44 (.24) | 37331.98 (.13) | 8650.71 (.27) |
| 4 | 1 | 1 | 1 | 7108.74 (.15) | 26367.71 (.03) | 33735.09 (.04) | 13330.01 (.11) | 8215.73 (.48) | 6724.54 (.23) | 37332.00 (.15) | 8651.34 (.27) |
| 4 | 2 | 2 | 4 | 7108.41 (.06) | 26367.71 (.03) | 33735.09 (.04) | 13329.96 (.11) | 8214.72 (.20) | 6724.44 (.24) | 37331.98 (.13) | 8650.68 (.23) |
| 4 | 3 | 3 | 9 | 7108.40 (.06) | 26367.71 (.02) | 33735.09 (.03) | 13329.94 (.07) | 8214.59 (.16) | 6724.39 (.21) | 37331.94 (.15) | 8650.64 (.22) |
| | | | | **MCMC and VBPI with GNNs (scores from (Zhang & Matsen IV, 2019) and (Zhang, 2023))** | | | | | | | |
| $\text{MrBayes}_{\text{ss}}$ | | | | 7108.42 (.18) | 26367.57 (.48) | 33735.44 (.50) | 13330.06 (.54) | 8214.51 (.28) | 6724.07 (.86) | 37332.76 (2.42) | 8649.88 (1.75) |
| GGNN | | | | 7108.40 (.19) | 26367.73 (.10) | 33735.11 (.09) | 13329.95 (.19) | 8214.67 (.36) | 6724.38 (.42) | 37332.03 (.30) | 8650.68 (.48) |
| EDGE | | | | 7108.41 (.14) | 26367.73 (.07) | 33735.12 (.09) | 13329.94 (.19) | 8214.64 (.38) | 6724.37 (.40) | 37332.04 (.26) | 8650.65 (.45) |

genetics (Hedges et al., 1990; Garey et al., 1996; Yang & Yoder, 2003; Henk et al., 2003; Lakner et al., 2008; Zhang & Blackwell, 2001; Yoder & Yang, 2004; Rossman et al., 2001). As in Zhang & Matsen IV (2019); Zhang (2020a); Moretti et al. (2021); Koptagel et al. (2022); Zhang & Matsen IV (2022); Zhang (2023), we learn the approximations of branch-length and tree-topology distributions. As can be seen in Table 3, similar, or identical, NLL results are obtained with fewer likelihood and variational probability evaluations. The cost of each such operation is specified in the table. This implies that our estimators have successfully reduced the inference time of the SOTA model, while preserving the impressive NLL scores.

## 7. Future Work

A future avenue of research is to theoretically justify the results in this work by producing convergence results for mixtures. Convergence properties and guarantees for BBVI have previously been studied by Kim et al. (2023); Domke et al. (2024); Hotti et al. (2024) for when the variational family belongs to the location-scale family, which does not include mixtures. Since the mixture differential entropy can no longer be computed in closed form, one would need to consider stochastic estimates of both the energy and the differential entropy gradients of the variational objective. In this context, it would be interesting to consider the sticking-the-landing (STL) estimator, previously studied in the setting of BBVI by both (Kim et al., 2024) and (Domke et al., 2024). It turns out that with the STL estimator, a linear convergence rate can be achieved when the variational family contains the true posterior, which approximately holds for a mixture of Gaussians given a sufficient number of components.

Gradient-based inference of mixture weights in BBVI is non-trivial (Morningstar et al., 2021), and there are multiple approaches to reparameterized sampling of the components (Figurnov et al., 2018; Morningstar et al., 2021). However, an alternative is resampling-based inference, as in the adaptive IS literature. Recently, Kviman et al. (2024) proposed a new resampling methodology which could be applied in mixture BBVI to weight components such that the ELBO is maximized via a combinatorial optimization algorithm.

## 8. Conclusion

In this work, we have addressed the scalability and efficiency challenges faced by mixtures in BBVI, by introducing MISVAE and the novel estimators of the MISELBO: the Some-to-All and Some-to-Some estimators. Our contributions significantly decrease the number required learnable parameters and the computational costs associated with increasing the number of mixture components, enabling scalability of mixture models without compromising performance.

## Impact Statement

This paper presents work with the goal to advance the field of machine learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## Acknowledgments

First, we acknowledge the insightful comments provided by the reviewers, which have helped improve our work. This project was made possible through funding from the Swedish Foundation for Strategic Research grants BD15-0043 and ID19-0052, and from the Swedish Research Council grant 2018-05417_VR. The computations and data handling were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC), partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

## References

Åkerborg, Ö., Sennblad, B., Arvestad, L., and Lagergren, J. Simultaneous bayesian gene tree reconstruction and reconciliation analysis. *Proceedings of the National Academy of Sciences*, 106(14):5714–5719, 2009.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Burda, Y., Grosse, R., and Salakhutdinov, R. Importance weighted autoencoders, 2016.

Domke, J., Gower, R., and Garrigos, G. Provable convergence guarantees for black-box variational inference. *Advances in Neural Information Processing Systems*, 36, 2024.

Elvira, V. and Martino, L. Advances in importance sampling. *arXiv preprint arXiv:2102.05407*, 2021.

Elvira, V., Martino, L., Luengo, D., and Bugallo, M. F. Efficient multiple importance sampling estimators. *Signal Processing Letters, IEEE*, 22(10):1757–1761, 2015.

Elvira, V., Martino, L., Luengo, D., and Bugallo, M. F. Heretical multiple importance sampling. *IEEE Signal Processing Letters*, 23(10):1474–1478, 2016a.

Elvira, V., Martino, L., Luengo, D., and Bugallo, M. F. Multiple importance sampling with overlapping sets of proposals. In *2016 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 1–5. IEEE, 2016b.

Elvira, V., Martino, L., Luengo, D., and Bugallo, M. F. Generalized Multiple Importance Sampling. *Statistical Science*, 34(1):129 – 155, 2019. doi: 10.1214/18-STS668. URL https://doi.org/10.1214/18-STS668.

Figurnov, M., Mohamed, S., and Mnih, A. Implicit reparameterization gradients. *Advances in neural information processing systems*, 31, 2018.

Garey, J. R., Near, T. J., Nonnemacher, M. R., and others. Molecular evidence for acanthocephala as a subtaxon of rotifera. *Journal of Molecular*, 1996.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Hedges, S. B., Moberg, K. D., and others. Tetrapod phylogeny inferred from 18S and 28S ribosomal RNA sequences and a review of the evidence for amniote relationships. *Mol. Biol.*, 1990.

Henk, D. A., Weir, A., and Blackwell, M. Laboulbeniopsis termitarius, an ectoparasite of termites newly recognized as a member of the laboulbeniomycetes. *Mycologia*, 95 (4):561–564, 2003.

Hotti, A. M., Van der Goten, L. A., and Lagergren, J. Benefits of non-linear scale parameterizations in black box variational inference through smoothness results and gradient variance bounds. In *International Conference on Artificial Intelligence and Statistics*, pp. 3538–3546. PMLR, 2024.

Kim, K., Wu, K., Oh, J., Ma, Y.-A., and Gardner, J. R. On the convergence of black-box variational inference. In *Neural Information Processing Systems*, 2023. URL https://api.semanticscholar.org/CorpusID:258865567.

Kim, K., Ma, Y., and Gardner, J. Linear convergence of black-box variational inference: Should we stick the landing? In *International Conference on Artificial Intelligence and Statistics*, pp. 235–243. PMLR, 2024.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017.

Koptagel, H., Kviman, O., Melin, H., Safinianaini, N., and Lagergren, J. Vaiphy: a variational inference based algorithm for phylogeny. *Advances in Neural Information Processing Systems*, 2022.

Kostantinos, N. Gaussian mixtures and their applications to signal processing. *Advanced signal processing handbook:*

*theory and implementation for radar, sonar, and medical imaging real time systems*, pp. 3–1, 2000.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. Automatic differentiation variational inference. *Journal of machine learning research*, 2017.

Kviman, O., Melin, H., Koptagel, H., Elvira, V., and Lagergren, J. Multiple importance sampling elbo and deep ensembles of variational approximations. In *International Conference on Artificial Intelligence and Statistics*, pp. 10687–10702. PMLR, 2022.

Kviman, O., Molén, R., Hotti, A., Kurt, S., Elvira, V., and Lagergren, J. Cooperation in the latent space: the benefits of adding mixture components in variational autoencoders. In *International Conference on Machine Learning*, pp. 18008–18022. PMLR, 2023a.

Kviman, O., Molén, R., and Lagergren, J. Improved variational bayesian phylogenetic inference using mixtures. *arXiv preprint arXiv:2310.00941*, 2023b.

Kviman, O., Branchini, V., Elvira, N., and Lagergren, J. Variational resampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 3286–3294. PMLR, 2024.

Lakner, C., van der Mark, P., Huelsenbeck, J. P., Larget, B., and Ronquist, F. Efficiency of markov chain monte carlo tree proposals in bayesian phylogenetics. *Syst. Biol.*, 57 (1):86–103, February 2008.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Ma, X., Zhou, C., and Hovy, E. Mae: Mutual posterior-divergence regularization for variational autoencoders. *arXiv preprint arXiv:1901.01498*, 2019.

Moretti, A. K., Zhang, L., Naesseth, C. A., Venner, H., Blei, D., and Pe'er, I. Variational combinatorial sequential monte carlo methods for bayesian phylogenetic inference. In *Uncertainty in Artificial Intelligence*, pp. 971–981. PMLR, 2021.

Morningstar, W., Vikram, S., Ham, C., Gallagher, A., and Dillon, J. Automatic differentiation variational inference with mixtures. In *International Conference on Artificial Intelligence and Statistics*, pp. 3250–3258. PMLR, 2021.

Nalisnick, E., Hertel, L., and Smyth, P. Approximate inference for deep latent gaussian mixtures. In *NIPS Workshop on Bayesian Deep Learning*, volume 2, pp. 131, 2016.

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

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019. URL http://arxiv.org/abs/1912.01703.

Rainforth, T., Kosiorek, A., Le, T. A., Maddison, C., Igl, M., Wood, F., and Teh, Y. W. Tighter variational bounds are not necessarily better. In *International Conference on Machine Learning*, pp. 4277–4285. PMLR, 2018.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Roeder, G., Wu, Y., and Duvenaud, D. K. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. *Advances in Neural Information Processing Systems*, 30, 2017.

Rossman, A. Y., McKemy, J. M., Pardo-Schultheiss, R. A., and Schroers, H.-J. Molecular studies of the bionectriaceae using large subunit rDNA sequences. *Mycologia*, 93(1):100–110, January 2001.

Sadeghi, H., Andriyash, E., Vinci, W., Buffoni, L., and Amin, M. H. Pixelvae++: Improved pixelvae with discrete prior. *arXiv preprint arXiv:1908.09948*, 2019.

Sbert, M. and Elvira, V. Generalizing the balance heuristic estimator in multiple importance sampling. *Entropy*, 24 (2):191, 2022.

Sinha, S. and Dieng, A. B. Consistency regularization for variational auto-encoders. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12943–12954. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/6c19e0a6da12dc02239312f151072ddd-Paper.pdf.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. *Advances in neural information processing systems*, 29, 2016.

Tomczak, J. and Welling, M. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pp. 1214–1223. PMLR, 2018.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

Van Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.

Veach, E. and Guibas, L. Optimally combining sampling techniques for Monte Carlo rendering. *In SIGGRAPH 1995 Proceedings*, pp. 419–428, 1995.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL `http://arxiv.org/abs/1708.07747`.

Yang, Z. and Yoder, A. D. Comparison of likelihood and bayesian methods for estimating divergence times using multiple gene loci and calibration points, with application to a radiation of cute .... *Syst. Biol.*, 2003.

Yoder, A. D. and Yang, Z. Divergence dates for malagasy lemurs estimated from multiple gene loci: geological and evolutionary context. *Mol. Ecol.*, 13(4):757–773, April 2004.

Zhang, C. Improved variational bayesian phylogenetic inference with normalizing flows. *Advances in neural information processing systems*, 33:18760–18771, 2020a.

Zhang, C. Improved variational bayesian phylogenetic inference with normalizing flows. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18760–18771. Curran Associates, Inc., 2020b. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/d96409bf894217686ba124d7356686c9-Paper.pdf`.

Zhang, C. Learnable topological features for phylogenetic inference via graph neural networks. *arXiv preprint arXiv:2302.08840*, 2023.

Zhang, C. and Matsen IV, F. A. Variational bayesian phylogenetic inference. In *International Conference on Learning Representations*, 2018.

Zhang, C. and Matsen IV, F. A. Variational bayesian phylogenetic inference. In *ICLR*, 2019.

Zhang, C. and Matsen IV, F. A. A variational approach to bayesian phylogenetic inference. *arXiv preprint arXiv:2204.07747*, 2022.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

Zhang, N. and Blackwell, M. Molecular phylogeny of dogwood anthracnose fungus (discula destructiva) and the diaporthales. *Mycologia*, 2001.

Zhou, M., Yan, Z., Layne, E., Malkin, N., Zhang, D., Jain, M., Blanchette, M., and Bengio, Y. Phylogfn: Phylogenetic inference with generative flow networks. *arXiv preprint arXiv:2310.08774*, 2023.

## A. Expected Values of the Estimators

Here we prove the results presented in Section 4.

There are $A$ components in total and thus $\binom{A}{S}$ subsets $\Phi$ of cardinality $S$ (without replacement). Furthermore, summed over all subsets, an arbitrary component, $q_{\phi_k}$, is observed $\binom{A}{S}\frac{S}{A} = \binom{A-1}{S-1}$ times. Define a uniform distribution in the space of all $S$-subsets, $\Omega^S$,

$$\varphi\left(\Phi\right) = \frac{1}{\binom{A}{S}} \tag{9}$$

and the Some-to-All estimator as

$$\tilde{\mathcal{L}}^{M,S,J}_{\text{Some-to-All}} := \frac{1}{M}\sum_{m=1}^{M}\frac{1}{S}\sum_{\phi_k\in\Phi^m}\frac{1}{J}\sum_{j'=1}^{J}\log\frac{p(x,z_k^{j'})}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_k^{j'}|x)}, \quad \Phi^1,...,\Phi^M\sim\varphi(\Phi), \quad z_k^1,...,z_k^J\sim q_{\phi_k}(z|x). \tag{10}$$

**Theorem 4.1:** The Some-to-All estimator is an unbiased estimator of Eq. (1).

*Proof.* Taking expectations and utilizing that the expectation is a linear operator, the R.H.S. of Eq. (10) is

$$\frac{1}{M}\sum_{m=1}^{M}\mathbb{E}_{\varphi(\Phi)}\left[\frac{1}{S}\sum_{\phi_k\in\Phi}\frac{1}{J}\sum_{j'=1}^{J}\mathbb{E}_{q_{\phi_k}(z|x)}\left[\log\frac{p(x,z_k)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_k|x)}\right]\right] = \tag{11}$$

$$\mathbb{E}_{\varphi(\Phi)}\left[\frac{1}{S}\sum_{\phi_k\in\Phi}\mathbb{E}_{q_{\phi_k}(z|x)}\left[\log\frac{p(x,z_k)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_k|x)}\right]\right] = \tag{12}$$

$$\sum_{\Phi\in\Omega^S}\frac{1}{\binom{A}{S}}\frac{1}{S}\sum_{\phi_k\in\Phi}\mathbb{E}_{q_{\phi_k}(z|x)}\left[\log\frac{p(x,z_k)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_k|x)}\right] = \tag{13}$$

$$\frac{1}{\binom{A}{S}}\frac{1}{S}\sum_{\phi_k\in\Phi}\mathbb{E}_{q_{\phi_k}(z|x)}\left[\log\frac{p(x,z_k)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_k|x)}\right] = \tag{14}$$

$$\frac{1}{\binom{A}{S}}\frac{1}{S}\binom{A-1}{S-1}\sum_{k=1}^{A}\mathbb{E}_{q_{\phi_k}(z|x)}\left[\log\frac{p(x,z_k)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_k|x)}\right] = \tag{15}$$

$$\frac{1}{A}\sum_{k=1}^{A}\mathbb{E}_{q_{\phi_k}(z|x)}\left[\log\frac{p(x,z_k)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_k|x)}\right] = \mathcal{L}_{\text{MIS}}, \tag{16}$$

where the equality in Eq. (14) holds as

$$\sum_{\Phi\in\Omega^S}\sum_{\phi_k\in\Phi}q_{\phi_k} = \sum_{\Phi\in\Omega^S}\sum_{k=1}^{A}\mathbb{1}_{\{\phi_k\in\Phi\}}(\phi_k)q_{\phi_k}(z|x) = \binom{A-1}{S-1}\sum_{k=1}^{S}q_{\phi_k}(z|x),$$

following the statement in the beginning of this section; component $q_{\phi_k}$ will be observed $\binom{A}{S}\frac{S}{A}$ times in all possible subsets. □

From the theorem above we can provide the following corollary.

**Corollary 4.2:** The expected value of the Some-to-All estimator is a lower bound on the marginal log-likelihood,

$$\mathbb{E}\left[\widetilde{\mathcal{L}}_{\text{S2A}}\right] \leq \log p_\theta(x).$$

*Proof.* From Theorem 4.1, we have $\mathbb{E}\left[\tilde{\mathcal{L}}^{M,S,J}_{\text{S2A}}\right] = \mathcal{L}_{\text{MIS}}$ and from Kviman et al. (2022) it is known that $\mathcal{L}_{\text{MIS}} \leq \log p_\theta(x)$. □

Next, we turn to the examination of the expected value of the Some-to-Some estimator

$$\tilde{\mathcal{L}}_{\text{S2S}}^{M,S,J} := \frac{1}{M}\sum_{m=1}^{M}\frac{1}{S}\sum_{\phi_k\in\Phi^m}\frac{1}{J}\sum_{j'=1}^{J}\log\frac{p(x,z_k^{j'})}{\frac{1}{S}\sum_{\phi_j\in\Phi^m}q_{\phi_j}(z_k^{j'}|x)}, \quad \Phi^1,...,\Phi^M\sim\varphi(\Phi), \quad z_k^1,...,z_k^J\sim q_{\phi_k}(z|x). \tag{17}$$

Note that this estimator diverges from the S2A estimator merely in the denominator inside the logarithm. However, this change clearly implies that the S2S estimator is not an unbiased estimator of Eq. (1). In fact, its expected value is instead a lower bound on MISELBO, as we will show here.

**Theorem A.1.** *The expected value of the Some-to-Some estimator is a lower bound on MISELBO, i.e.*

$$\mathbb{E}\left[\tilde{\mathcal{L}}_{S2S}^{M,S,J}\right] \leq \mathcal{L}_{MIS}.$$

*Proof.* Using that $\mathbb{E}\left[\tilde{\mathcal{L}}_{\text{S2A}}^{M,S,J}\right] = \mathcal{L}_{\text{MIS}}$, we will directly check if $\mathbb{E}\left[\tilde{\mathcal{L}}_{\text{S2S}}^{M,S,J}\right] \leq \mathbb{E}\left[\tilde{\mathcal{L}}_{\text{S2A}}^{M,S,J}\right]$. From the formulation in Eq. (12), the inequality can equivalently be expressed as

$$\mathbb{E}_{\varphi(\Phi)}\left[\frac{1}{S}\sum_{\phi_s\in\Phi}\mathbb{E}_{q_{\phi_s}(z_s|x)}\left[\log\frac{p(x,z_s)}{\frac{1}{S}\sum_{\phi_k\in\Phi}q_{\phi_k}(z_s|x)}\right]\right] \leq \mathbb{E}_{\varphi(\Phi)}\left[\frac{1}{S}\sum_{\phi_s\in\Phi}\mathbb{E}_{q_{\phi_s}(z_s|x)}\left[\log\frac{p(x,z_s)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_s|x)}\right]\right]. \tag{18}$$

Subtracting the R.H.S. with the L.H.S., we get

$$\mathbb{E}_{\varphi(\Phi)}\left[\frac{1}{S}\sum_{\phi_s\in\Phi}\mathbb{E}_{q_{\phi_s}(z_s|x)}\left[\log\frac{p(x,z_s)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_s|x)}\right]\right] - \mathbb{E}_{\varphi(\Phi)}\left[\frac{1}{S}\sum_{\phi_s\in\Phi}\mathbb{E}_{q_{\phi_s}(z_s|x)}\left[\log\frac{p(x,z_s)}{\frac{1}{S}\sum_{\phi_k\in\Phi}q_{\phi_k}(z_s|x)}\right]\right] \tag{19}$$

$$=\mathbb{E}_{\varphi(\Phi)}\left[\frac{1}{S}\sum_{\phi_s\in\Phi}\mathbb{E}_{q_{\phi_s}(z_s|x)}\left[\log\frac{p(x,z_s)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_s|x)} - \log\frac{p(x,z_s)}{\frac{1}{S}\sum_{\phi_k\in\Phi}q_{\phi_k}(z_s|x)}\right]\right] \tag{20}$$

$$=\mathbb{E}_{\varphi(\Phi)}\left[\frac{1}{S}\sum_{\phi_s\in\Phi}\mathbb{E}_{q_{\phi_s}(z_s|x)}\left[\log\frac{\frac{1}{S}\sum_{\phi_k\in\Phi}q_{\phi_k}(z_s|x)}{\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z_s|x)}\right]\right] \tag{21}$$

$$=\mathbb{E}_{\varphi(\Phi)}\left[\text{KL}\left(\frac{1}{S}\sum_{\phi_k\in\Phi}q_{\phi_k}(z|x)\middle\|\frac{1}{A}\sum_{j=1}^{A}q_{\phi_j}(z|x)\right)\right] \geq 0, \tag{22}$$

where the final inequality holds as the KL, and thus the average of KLs, is non-negative. □

From the result above, it furthermore follows directly that the expected value of the S2S estimator is a lower bound on the marginal log-likelihood.

**Corollary A.2.** *The expected value of the Some-to-Some estimator is a lower bound on the marginal log-likelihood, i.e.*

$$\mathbb{E}\left[\tilde{\mathcal{L}}_{S2S}^{M,S,J}\right] \leq \log p_\theta(x).$$

**Corollary 4.1:** The expected value of the Some-to-All estimator is a lower bound on the marginal log-likelihood,

$$\mathbb{E}\left[\widetilde{\mathcal{L}}_{\text{S2A}}\right] \leq \log p_\theta(x).$$

*Proof.* Recalling that $\mathcal{L}_{\text{MIS}} \leq \log p_\theta(x)$, it follows from Theorem A.1 that $\mathbb{E}\left[\tilde{\mathcal{L}}_{\text{S2S}}^{M,S,J}\right] \leq \mathcal{L}_{\text{MIS}} \leq \log p_\theta(x)$. □

## B. Extension to Weighted Mixture

We now extend the unbiasedness result to the case of weighted mixtures. We repeat Theorem 4.3 from the main text.

**Theorem B.1.** *The Some-to-All estimator is an unbiased estimator of MISELBO for arbitrary mixture weights.*

*Proof.* First let us define the *weighted* A2A estimator

$$\widetilde{\mathcal{L}}_{\text{A2A}} = \sum_{k=1}^{A} \frac{w_k}{w_A} \log \frac{p_\theta(x|z_k)p_\theta(z_k)}{\sum_{a'=1}^{A} \frac{w_{a'}}{w_A} q_{\phi_{a'}}(z_k|x)}, \tag{23}$$

where $z_a \sim q_{\phi_a(z|x)}$, $w_a$ are the unnormalized weights and

$$w_A := \sum_{k=1}^{A} w_k.$$

Let $(I, \Sigma)$ be a measurable space with sample space

$$I = \left\{ H \in \{0,1\}^A : \sum_{j \in [A]} H_j = S \right\} \tag{24}$$

and probability measure $\varphi(H)$.

For the sake of generality we will consider an estimator of the importance weighted MISELBO (i.e. with $L \geq 1$). Let

$$\mathcal{L}_{S2A}^{M,S,\varphi} := \frac{1}{M} \sum_{m=1}^{M} \sum_{k=1}^{A} u_k H_k^m \log \frac{1}{L} \sum_{l=1}^{L} \frac{p(x, z_k^l)}{\sum_{a'=1}^{A} \frac{w_{a'}}{w_A} q_{\phi_{a'}}(z_k^l|x)} \quad (*)$$

where

$$H^1, \ldots, H^M \sim \varphi, \qquad z_k^l \sim q_{\phi_k}(z|x), \quad \text{and} \quad u_k := \frac{w_k}{w_A \mathbb{E}[H_k]}$$

Now we show that the expectation of $(*)$ is equal to the importance weighted MISELBO objective with arbitrary mixture weights.

$$\mathbb{E}[\mathcal{L}_{S2A}^{M,S,\varphi}] = \mathbb{E}_{H^1,\ldots,H^M \sim \varphi} \left[ \frac{1}{M} \sum_{m=1}^{M} \sum_{k=1}^{A} u_k H_k^m \mathbb{E}_{z_k^l \sim q_{\phi_k}(z|x)} \left[ \log \frac{1}{L} \sum_{l=1}^{L} \frac{p(x, z_k^l)}{\sum_{a'} \frac{w_{a'}}{w_A} q_{\phi_{a'}}(z_k^l|x)} \right] \right]$$

$$= \mathbb{E}_{H \sim \varphi} \left[ \sum_{k=1}^{A} u_k H_k \mathbb{E}_{z_k^l} \left[ \log \frac{1}{L} \sum_{l=1}^{L} \frac{p(x, z_k^l)}{\sum_{a'} \frac{w_{a'}}{w_A} q_{\phi_{a'}}(z_k^l|x)} \right] \right]$$

$$= \sum_{H \in I} \varphi(H) \left[ \sum_{k=1}^{A} u_k H_k \mathbb{E}_{z_k^l} \left[ \log \frac{1}{L} \sum_{l=1}^{L} \frac{p(x, z_k^l)}{\sum_{a'} \frac{w_{a'}}{w_A} q_{\phi_{a'}}(z_k^l|x)} \right] \right]$$

$$= \sum_{k=1}^{A} u_k \left( \sum_{H \in I^S} \varphi(H) H_k \right) \mathbb{E}_{z_k^l} \left[ \log \frac{1}{L} \sum_{l=1}^{L} \frac{p(x, z_k^l)}{\sum_{a'} \frac{w_{a'}}{w_A} q_{\phi_{a'}}(z_k^l|x)} \right]$$

$$= \sum_{k=1}^{A} u_k \mathbb{E}_\varphi[H_k] \mathbb{E}_{z_k^l} \left[ \log \frac{1}{L} \sum_{l=1}^{L} \frac{p(x, z_k^l)}{\sum_{a'} \frac{w_{a'}}{w_A} q_{\phi_{a'}}(z_k^l|x)} \right]$$

$$= \sum_{k=1}^{A} \frac{w_k}{w_A} \mathbb{E}_{z_k^l} \left[ \log \frac{1}{L} \sum_{l=1}^{L} \frac{p(x, z_k^l)}{\sum_{a'} \frac{w_{a'}}{w_A} q_{\phi_{a'}}(z_k^l|x)} \right].$$

That is the same as that of the weighted A2A. □

## C. Block Diagrams MISVAE - S2A and A2A

In Fig. 7 we display block diagrams for MISVAE with the S2A and A2A estimators.

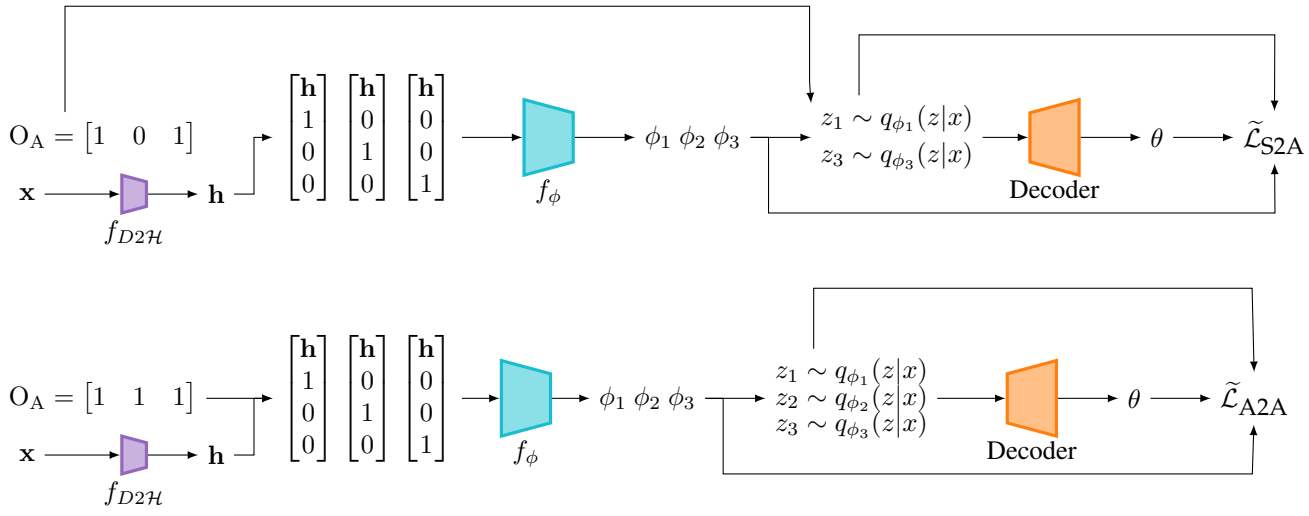



Figure 7: Block diagram illustrating the estimation of MISELBO with MISVAE, showcasing the S2A estimator (top) with $S = 2$ and $A = 3$, alongside the A2A estimator (bottom) with $A = 3$.

## D. Additional Experimental Details

### D.1. Training Infrastructure

All experiments were conducted on a NVIDIA RTX 4090s with 24 GiB of memory each using the PyTorch framework (Paszke et al., 2019).

**MNIST (LeCun & Cortes, 2010).** When training on the MNIST image dataset, $f_{D2\mathcal{H}}$ was defined as a sequence of five gated convolutional layers. To learn $f_\phi$, which amortizes the mean and covariance matrices of the variational posterior, we employed two separate non-linear networks. Each network consisted of an input layer, an output layer, and a hidden layer, the latter featuring 40 latent dimensions equipped with ReLU activation functions. We used a single layer Pixel CNN decoder. For optimization, we used Adam (Kingma & Ba, 2017), with a learning rate of 0.0005, and a batch size of 100 and initiated the process with a KL-warmup phase lasting 100 epochs.

**CIFAR-10 (Krizhevsky et al.).** For $f_{D2\mathcal{H}}$ we used a pre-trained ResNet model[2] with 20 layers. $f_\phi$ was defined the same way as on MNIST, except we used 128 latent dimensions. We optimized using Adam, with a learning rate of 0.001, and a batch size of 100 and initiated the training with KL-warmup during 500 epochs. We used a Pixel CNN decoder with 4 layers.

[2]https://github.com/akamaster/pytorch_resnet_cifar10/tree/master

# E. Additional Experimental Results

## E.1. Toy Experiment

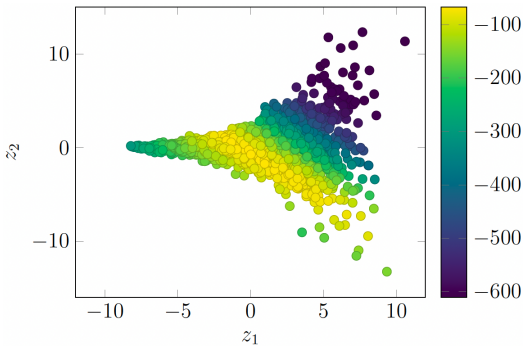



Figure 8: An unnormalized posterior from the toy example when $d_x = 20$ and $N = 5$. The plot is generated by sampling from the prior and evaluating the unnormalized log-probabilities of each sample (darker is lower).

### E.1.1. The Exponential-Decay-Bernoulli-Likelihood-and-Neal-Funnel-Prior Model

We generated a dataset $\mathcal{D} = \{x_n\}_{n=1}^5$ by sampling from the model. The approximations were learned using the Adam optimizer (Kingma & Ba, 2014) with learning rate equal to 0.001. The other optimizer parameters were set to their (PyTorch) default values. The variational parameters were initialized using the Kaiming-uniform initialization (He et al., 2015), the number of training iterations were 50k. All estimators used the same number of importance samples when estimating the MISELBO scores shown in the curve figures.

### E.1.2. Toy Experiment - Latent Space Visualization

The approximations are visualized in the latent spaces shown in figures 9-12. Notably, when $A = 5$, the approximations from S2A and A2A are identical (recall that S2A required substantially less inference time, see in Sec. 6). Meanwhile, the components in the S2S approximations were not able to sufficiently separate themselves.

Finally, in Fig. 12, we visualize how the approximation learned with the S2S estimator behaves when $S = 5$ and $S = 20$. The approximation obtains impressive MISELBO scores (shown in Sec. 6) although the components largely overlapping.

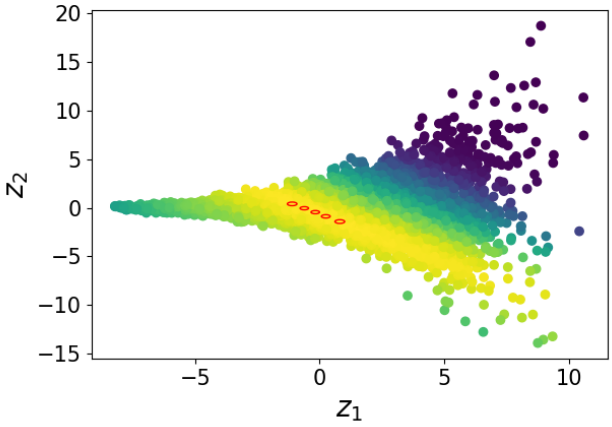



Figure 9: $dx = 20$: approximation when using the S2A estimator, $S = 2$ and $A = 5$.

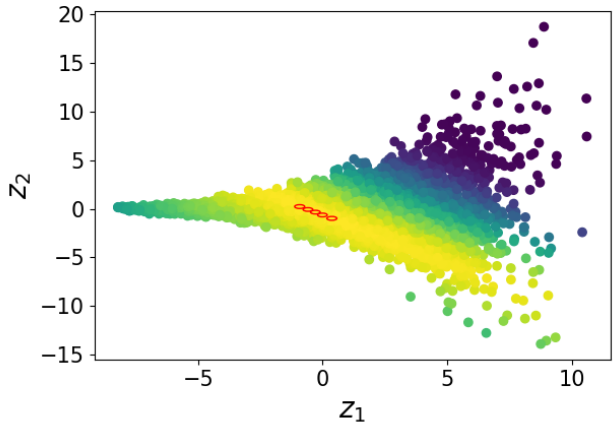



Figure 10: $dx = 20$: approximation when using the S2S estimator, $S = 2$ and $A = 5$.

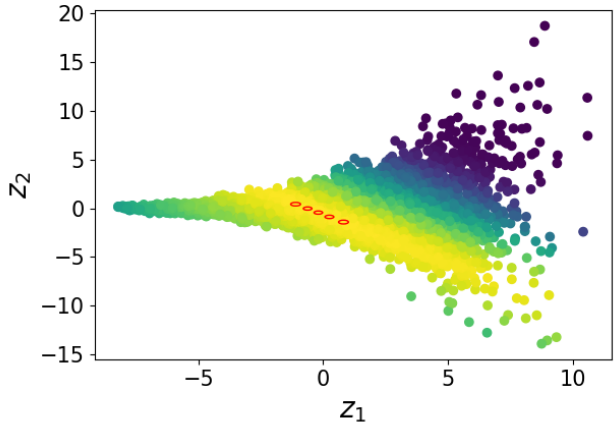



Figure 11: $dx = 20$: approximation when using the A2A estimator, $S = 5$ and $A = 5$.

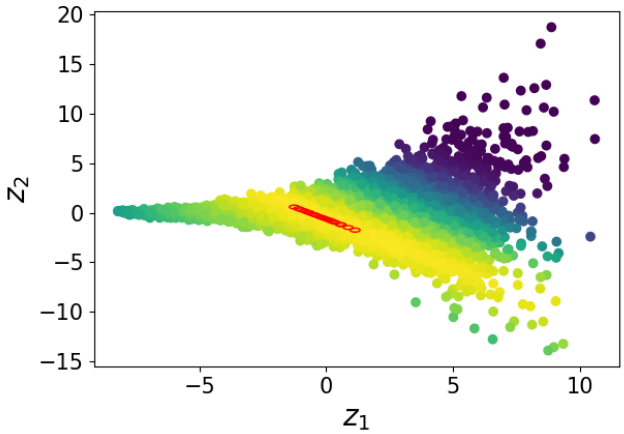



Figure 12: $dx = 20$: approximation when using the S2S estimator, $S = 5$ and $A = 20$.

### E.2. Additional S2S Results with big $A$

In Fig. 13a, we showcase how the final test set Negative Log Likelihood (NLL) value is impacted by setting the total number of mixtures to a large fixed value $A = 50$ and gradually increasing the number of components used by the S2S estimator. The performance of S2S is compared to the A2A estimator, where we instead let $A = S$ of the former estimator and gradually increase $A$. For small values of $S$, S2S exhibits a clear performance advantage both in terms of NLL and inference time per epoch (not shown here). However, as $S$ approaches $A$, the NLL performance advantage of S2S diminishes compared to A2A.

In Fig. 13b, we let $S$ be a fixed small value and gradually increase $A$ for the S2S estimator. For both curves, an initial increase in $A$ results in significant performance gains in terms of NLL; however, beyond a certain point, adding more mixtures yields no further improvements.

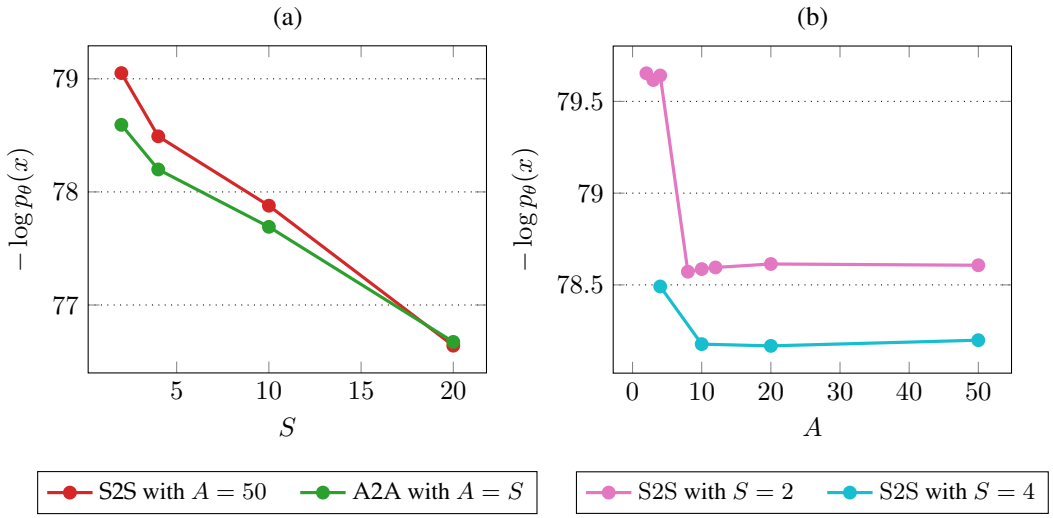



Figure 13: Analysis of MISELBO estimation using the S2S estimator. (a) With the number of mixtures $A$ set to a constant value, we incrementally increase the number of components $S$, observing the impact on estimation accuracy. (b) Conversely, we maintain a constant number of components $S$ while progressively increasing the number of mixtures $A$ to assess the benefits of additional mixtures on the estimation process.

### E.3. Phylogenetics Experiment

#### E.3.1. Details of Variational Bayesian Phylogenetic Inference using mixtures and black-box variational inference

The task in Bayesian phylogenetic inference, is to approximate the posterior distribution over branch lengths, $\mathcal{B}$, and tree topologies, $\tau$, given the observed sequence data (typically DNA data), $x$. The phylogenetic posterior is thus defined as

$$p(\tau, \mathcal{B}|x) = \frac{p(x|\tau, \mathcal{B})p(\mathcal{B}|\tau)p(\tau)}{p(x)}, \tag{25}$$

where the marginal likelihood, $p(x)$, is intractable.

Based on algorithms of (Zhang & Matsen IV, 2018; 2019; Zhang, 2020a; Zhang & Matsen IV, 2022; Zhang, 2023; Zhou et al., 2023) we utilized the S2A and S2S to speed up the improvements made with mixtures in (Kviman et al., 2023b). The improvements follow the same structure as the original paper. In summary, the Subsplit Bayesian Networks (SBNs; Zhang & Matsen IV (2018)) are utilized to learn tree topologies in a Bayesian phylogenetic context. An SBN employs a lookup table containing probabilities of subsplits (partial tree structures), referred to as a Conditional Probability Table (CPT). This table is learned through BBVI, and once the CPT is established, the SBN provides a tractable probability distribution over tree topologies, enabling sampling from this distribution. The parameters of SBNs are learned in the VBPI (Variational Bayesian Phylogenetic Inference) framework by maximizing MISELBO using VIMCO for variance reduction of the gradient.

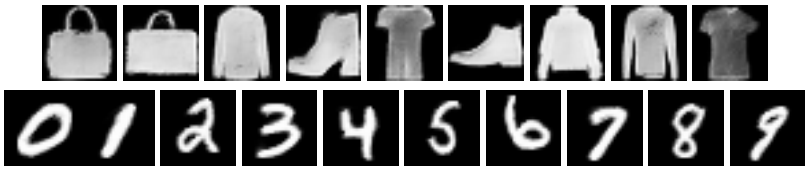


Figure 14: **Top:** Images generated with a variational posterior with 400 mixture components (A=400), trained with the S2A estimator and S=1 on FashionMNIST. **Bottom:** Images generated with a variational posterior with 600 mixture components (A=600), trained with the S2A estimator and S=1 on MNIST

VIMCO (Variational Inference for Monte Carlo Objectives), the VBPI-Mixtures algorithm is a novel development in Bayesian phylogenetics. It demonstrates that mixtures of SBNs can approximate distributions unattainable by a single SBN, providing more accurate models of complex phylogenetic datasets. The VIMCO estimator, specifically derived for mixtures, enhances this approach. This estimator enables the VBPI-Mixtures algorithm to jointly explore the tree-topology space more effectively, leading to state-of-the-art results on various real phylogenetics datasets. Thus, mixtures of SBNs, coupled with the VIMCO estimator, significantly improve the accuracy of approximations of the tree-topology posterior in Bayesian phylogenetic inference. In this article, we took these improvements and combined them with the novel improvements of S2A and S2S to significantly speed up the training process with minimal to no loss in performance, resulting in a scalable solution that can be used for more advance phylogenetic problems.

### E.4. Generated Images

To assess the generative capabilities of our models, we have included visualizations of images generated from the MNIST and FashionMNIST datasets in Fig. 14.

### E.5. Comparable performance on CIFAR-10

Table 4: NLL statistics for various SOTA VAE architectures on the **CIFAR**-10 dataset. The Composite model is a SEMVAE model which incorporates hierarchical models, NFs and the VampPrior. For IWAE $L$ is the number of importance samples used during training.

| Model | NLL |
|---|---|
| NVAE (Vahdat & Kautz, 2020) | 2.93 |
| PixelVAE++ (Sadeghi et al., 2019) | 2.90 |
| MAE (Ma et al., 2019) | 2.95 |
| Vanilla SEMVAE ($S = 12$; Kviman et al. (2023a)) | 4.83 |
| CR-NVAE (Sinha & Dieng, 2021) | 2.51 |
| MISVAE S2S ($A = 4, S = 2$; **our**) | 3.23 |
| MISVAE S2A ($A = 4, S = 2$; **our**) | 3.19 |