Journal of Machine Learning Research 27 (2026) 1-34 Submitted 10/25; Revised 6/26; Published 8/26

Gradient Estimation for Mixture Variational Inference

Javier Burroni* JAVIER.BURRONI@GMAIL.COM

Daniel Sheldon SHELDON@CS.UMASS.EDU
College of Information & Computer Sciences
University of Massachusetts Amherst
Amherst, MA 01003-9264, USA

Editor: Silvia Chiappa

Abstract

Mixture distributions are expressive variational families for black-box VI, but their discrete component choices complicate gradient estimation. We systematize reparameterization-based estimators for mixtures in a common notation, giving self-contained derivations and extending several to new settings. In particular, we provide an elementary derivation of a single-sample *post-stratified* estimator—previously derived via transport equations—and prove a variance reduction relative to simple random sampling. We also broaden the applicability of implicit reparameterization and reduce its computational complexity. Across different benchmarks, we find that stratified estimators are consistently robust when feasible; among single-sample methods, the post-stratified estimator frequently perform best, while implicit reparameterization is the most computationally demanding. Our analysis clarifies when each method should be used and provides efficient algorithms that make mixture-based variational inference practical.

Keywords: variational inference, mixture models, gradient estimation, reparameterization trick

1. Introduction

Mixture distributions offer a conceptually appealing method for creating an expressive family of distributions. By mixing multiple simpler unimodal building blocks, complex multimodal posterior approximations can be constructed, which has attracted the attention of the variational inference community (Jaakkola and Jordan, 1998).

However, their use as approximating distributions in variational objectives is challenging because computing gradients through mixture expectations is complex—especially when the mixture weights are also learned (our focus). Score-function estimators can handle this, but their high variance makes optimization difficult. The reparameterization trick (Fu, 2006; Rezende et al., 2014; Kingma and Welling, 2013), a key tool in variational inference, does not apply directly because sampling from mixtures involves discrete choices. In principle, one can write the mixture expectation as a convex combination of component expectations and then reparameterize each component (Roeder et al., 2017), yielding a stratified estimator; yet this requires multiple model evaluations per gradient step and can be prohibitive as the

∗. This work was done while Javier Burroni was at the University of Massachusetts Amherst.

number of components grows. This motivates *single-sample* estimators that keep one model evaluation per draw while retaining reparameterization ideas (Figurnov et al., 2018; Graves, 2016; Jankowiak and Karaletsos, 2019).

This paper unifies and clarifies the landscape of gradient estimators for mixture distributions in variational inference, and contributes new methodology and analysis. Concretely:

- We state all estimators in common notation and provide self-contained derivations (Table 1). The derivations include both the stratified estimator (Section 3) and single-sample estimators (Section 4).
- Among the single-sample estimators, we provide a novel, elementary derivation of a "hybrid" estimator, briefly mentioned by Jankowiak and Karaletsos (2019), sidestepping the formalisms of vector fields and transport equations used previously. We call this the *post-stratified estimator* due to its connection to stratified and post-stratified sampling of mixture distributions (Section 4.3). This connection lets us prove that the estimator reduces variance relative to simple random sampling.
- We expand the class of distributions that admit implicit reparameterization (Figurnov et al., 2018) and show how to reduce its computational cost (Section 4.5).
- Finally, we provide a systematic empirical comparison of different estimators across various settings (Section 5). Our main findings are: (i) the stratified estimator is robust when applicable; (ii) for single-sample estimators, the post-stratified estimator and a representative estimator from Jankowiak and Karaletsos (2019) (Section 6) are often the best options, though performance varies by problem; (iii) implicit reparameterization is typically the most computationally expensive.

2. Problem Formulation and Summary of Estimators

In variational inference, we choose parameters θ by optimizing the expectation $\mathbb{E}_{q_\theta}[\mathcal{L}_\theta]$ of a test function $\mathcal{L}_\theta(z)$, so that $q_\theta(z)$ closely approximates the posterior $p(z \mid x)$ for observed data x. Estimating gradients of expectations of this form is a basic primitive across machine learning—arising in variational inference, reinforcement learning, and beyond—and has no canonical solution (Mohamed et al., 2020). Since optimization proceeds by gradient methods, a central task is to estimate, by differentiating under the integral sign,

$$\nabla_\theta \, \mathbb{E}_{q_\theta}[\mathcal{L}_\theta] = \int (\nabla_\theta \mathcal{L}_\theta(z)) \, q_\theta(z) \, dz + \int \mathcal{L}_\theta(z) \, \nabla_\theta q_\theta(z) \, dz. \tag{1}$$

The first term is itself an expectation under q_θ and is easily estimated by sampling; the difficulty is the second term, where θ enters through the distribution q_θ that we sample from. Two standard strategies address it. The *score-function* (or REINFORCE) estimator (Williams, 1992) uses the identity $\nabla_\theta q_\theta = q_\theta \nabla_\theta \ln q_\theta$ to rewrite the troublesome term as an ordinary expectation, $\mathbb{E}_{q_\theta}[\mathcal{L}_\theta \nabla_\theta \ln q_\theta]$; it applies to any differentiable density but is notorious for high variance. The *reparameterization trick* (Fu, 2006; Rezende et al., 2014; Kingma and Welling, 2013) instead writes a sample as a deterministic, differentiable transform of parameter-free noise: if $z = \mathcal{T}_\theta(\epsilon)$ with $\epsilon \sim q_{\text{base}}$ and q_{base} independent of θ, then

$$\mathbb{E}_{q_\theta}[\mathcal{L}_\theta(z)] = \mathbb{E}_{\epsilon \sim q_{\text{base}}}\big[\mathcal{L}_\theta(\mathcal{T}_\theta(\epsilon))\big], \qquad \text{so} \qquad \nabla_\theta \, \mathbb{E}_{q_\theta}[\mathcal{L}_\theta] = \mathbb{E}_{\epsilon \sim q_{\text{base}}}\big[\nabla_\theta \mathcal{L}_\theta(\mathcal{T}_\theta(\epsilon))\big]. \tag{2}$$

Because the base distribution does not depend on θ, the gradient commutes with the expectation and acts on the smooth map $\mathcal{T}_\theta$. For a Gaussian $q_\theta = \mathcal{N}(\mu, \sigma^2)$ with $\theta = (\mu, \sigma)$, one takes $q_{\text{base}} = \mathcal{N}(0, 1)$ and $\mathcal{T}_\theta(\epsilon) = \mu + \sigma\epsilon$. This estimator typically has far lower variance, making it the workhorse of variational inference; for a comprehensive treatment of both strategies, we refer the reader to Mohamed et al. (2020).

We study the case where q_θ is a K-component mixture, in which each component has a differentiable density $q_{k,\theta} \colon \mathcal{Z} \to \mathbb{R}$ and the overall density is

$$q_\theta \colon z \mapsto \sum_{k=1}^{K} \pi_{k,\theta} q_{k,\theta}(z),$$

where $\theta \in \mathbb{R}^d$ collects all parameters of the mixture and its components, and $\pi_{k,\theta}$ are nonnegative weights that sum to 1. Sampling from a mixture first selects a component and then draws from it; it is this discrete step that places reparameterization (2) out of direct

Estimator	**Expression**	**Sampling**
Stratified, $\hat{g}^{(K)}$ (Roeder et al., 2017) Estimator 1, Section 3	$\sum_{k=1}^{K} \nabla_\theta(\pi_{k,\theta}\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k)))$	$\epsilon_{1:K} \overset{\text{iid}}{\sim} q_{\text{base}}$
Stratified, alternative form Estimator 1b, Section 3	$\sum_{k=1}^{K} \pi_{k,\theta}\Big(\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k)) + \mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\nabla_\theta \ln \pi_{k,\theta}\Big)$	$\epsilon_{1:K} \overset{\text{iid}}{\sim} q_{\text{base}}$
Randomized components, $\hat{g}^{\mathrm{R}\kappa}$ Estimator 4, Section 4.2	$\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))\big\vert_{\epsilon=\mathcal{T}_{k,\theta}^{-1}(z_k)} + \mathcal{L}_\theta(z_k)\nabla_\theta \ln \pi_{k,\theta}$	$k \sim \mathrm{Cat}(\pi_{1:K,\theta})$ $z \sim q_{k,\theta}$
Post-stratified, $\hat{g}^{\mathrm{PS}}$ Estimator 5, Section 4.3	$\sum_{k=1}^{K} q_\theta(k \mid z)\Big(\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))\big\vert_{\epsilon=\mathcal{T}_{k,\theta}^{-1}(z)} + \mathcal{L}_\theta(z)\nabla_\theta \ln \pi_{k,\theta}\Big)$	$z \sim q_\theta$
J&K, $\hat{g}^{\mathrm{JK}}$ (Jankowiak and Karaletsos, 2019) Estimator 6, Section 4.4	$\sum_{k=1}^{K} q_\theta(k \mid z)\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))\big\vert_{\epsilon=\mathcal{T}_{k,\theta}^{-1}(z)} + \mathbf{V}^\theta(z) \cdot \nabla_z\mathcal{L}_\theta(z)$	$z \sim q_\theta$
Implicit reparameterization, $\hat{g}^{\mathrm{IR}}$ (Figurnov et al., 2018) Estimator 8, Section 4.5	$-\nabla_\theta\mathcal{S}_\theta(z)(\nabla_z\mathcal{S}_\theta(z))^{-1}\nabla_z\mathcal{L}_\theta(z) + \nabla_\theta\mathcal{L}_\theta(z)$	$z \sim q_\theta$
Score-function, $\hat{g}^{\mathrm{SF}}$ (Williams, 1992) Estimator 3, Section 4.1	$\mathcal{L}_\theta(z)\sum_{k=1}^{K} q_\theta(k \mid z)\nabla_\theta \ln(\pi_{k,\theta}q_{k,\theta}(z)) + \nabla_\theta\mathcal{L}_\theta(z)$	$z \sim q_\theta$

Table 1: Overview of estimators and derivations. See Section 2 for notation and background.

reach. For a mixture, the gradient (1) specializes to

$$\nabla_\theta \, \mathbb{E}_{q_\theta} \, \mathcal{L}_\theta = \nabla_\theta \int \mathcal{L}_\theta(z) \Big(\sum_{k=1}^{K} \pi_{k,\theta} q_{k,\theta}(z) \Big) \, dz. \tag{3}$$

We aim to estimate this gradient with low variance at minimal computational cost.

Table 1 summarizes the estimators using the notation defined in this section; the estimators themselves will be derived in later sections. The estimators typically only require $\mathcal{L}_\theta$ be differentiable in both z and θ. The Sticking-the-Landing estimator additionally requires that the expectation equal the evidence lower bound (ELBO); that is, $\mathcal{L}_\theta(z) = \ln p(x, z) - \ln q_\theta(z)$, where $p(x, z)$ is the joint distribution of the data x and latent variables z.

Notation *For a scalar-valued $g(z) \colon \mathbb{R}^M \to \mathbb{R}$, we denote by $\nabla_z g(z)$ the column vector of partial derivatives of g with respect to z evaluated at z. For a vector-valued $h(\theta) \colon \mathbb{R}^D \to \mathbb{R}^M$, we extend the notation such that $(\nabla_\theta h(\theta))_{ij} = \frac{\partial h_j(\theta)}{\partial \theta_i}$; i.e., the jth column is $\nabla_\theta h_j(\theta)$. Hence, $\nabla_\theta g(h(\theta)) = \nabla_\theta h(\theta) \cdot \nabla_z g(z)\big|_{z=h(\theta)}$. We also use the Jacobian $J_\theta h(\theta) = (\nabla_\theta h(\theta))^\top$.*

We consider two settings: *pure inference* problems, where $p(z \,|\, x)$ is fixed, and *learning* problems, where the model $p_\phi(x \,|\, z)$ is optimized with respect to ϕ. Working in either setting does not substantially complicate estimator derivations: when optimizing parameters that do not affect q_θ, gradients with respect to ϕ commute with the expectation under q_θ.

For most estimators, we assume each mixture component is reparameterizable and shares a common base distribution, as stated in the following assumption.

Assumption 1 *There exists a distribution q_{base} and bijections $(\mathcal{T}_{k,\theta} \colon \mathcal{Z} \to \mathcal{Z})_k$, such that $\mathcal{T}_{k,\theta}(\epsilon) \sim q_{k,\theta}$ whenever $\epsilon \sim q_{\text{base}}$. Moreover, we assume that the matrix of partial derivatives $\nabla_\theta \mathcal{T}_{k,\theta}$ exists and is continuous.*

For the implicit reparameterization estimator, we assume a standardizing function $\mathcal{S}_\theta$ mapping q_θ to the base distribution: if $z \sim q_\theta$, then $\mathcal{S}_\theta(z) \sim q_{\text{base}}$; see Section 4.5.

3. Stratified Estimator

We present the *stratified estimator* and its properties. The name reflects an analogy with *stratified sampling* from survey methodology (Lohr, 2021): components serve as strata, weights $\pi_{k,\theta}$ as stratum sizes, and the gradient is estimated by drawing one sample per stratum (Morningstar et al., 2021). This generalizes the reparameterization trick to mixtures.

Notice that, under Assumption 1, the mixture expectation can be expressed as

$$\mathbb{E}_{q_\theta} \, \mathcal{L}_\theta = \sum_{k=1}^{K} \pi_{k,\theta} \, \mathbb{E}_{q_{k,\theta}} \, \mathcal{L}_\theta = \sum_{k=1}^{K} \pi_{k,\theta} \, \mathbb{E}_{\epsilon \sim q_{\text{base}}} [\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))]. \tag{4}$$

Estimator 1 [Stratified] Let $\epsilon_1, \ldots, \epsilon_K \sim q_{\text{base}}$ be independent samples from the base distribution. Then, the estimator

$$\hat{g}^{(K)} \colon (\epsilon_1, \ldots, \epsilon_K) \mapsto \sum_{k=1}^{K} \nabla_\theta (\pi_{k,\theta} \mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))) \tag{5}$$

is an unbiased estimator of $\nabla_\theta \, \mathbb{E}_{q_\theta} \, \mathcal{L}_\theta$.

Proof This is a consequence of the *reparameterization trick* and follows because q_{base} does not depend on θ. We can write

$$\begin{aligned}\mathbb{E}_{\epsilon_{1:K}\sim q_{\text{base}}}[\hat{g}^{(K)}(\epsilon_1,\dots,\epsilon_K)] &= \mathbb{E}_{\epsilon_{1:K}\sim q_{\text{base}}}\Big[\sum_{k=1}^{K}\nabla_\theta(\pi_{k,\theta}\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k)))\Big]\\ &= \nabla_\theta\sum_{k=1}^{K}\pi_{k,\theta}\,\mathbb{E}_{\epsilon\sim q_{\text{base}}}\big[\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))\big] && ;\ (\star)\\ &= \nabla_\theta\,\mathbb{E}_{q_\theta}\,\mathcal{L}_\theta, && ;\ \text{by (4)}\end{aligned}$$

where $(\star)$ uses linearity of the expectation and the fact that q_{base} is independent of θ. ■

We also write an alternative form of the stratified estimator that will be useful later:

Estimator 1b Take $\epsilon_1,\dots,\epsilon_K \sim q_{\text{base}}$. Then, an equivalent form of the stratified estimator is given by

$$\hat{g}^{(K)}\colon(\epsilon_1,\dots,\epsilon_K)\mapsto\sum_{k=1}^{K}\pi_{k,\theta}\Big(\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))+\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\nabla_\theta\ln\pi_{k,\theta}\Big). \tag{6}$$

Proof This follows from applying the product rule to each term in Estimator 1:

$$\begin{aligned}\nabla_\theta(\pi_{k,\theta}\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))) &= \pi_{k,\theta}\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))+(\nabla_\theta\pi_{k,\theta})\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))\\ &= \pi_{k,\theta}\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))+\pi_{k,\theta}\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))\nabla_\theta\ln\pi_{k,\theta}.\end{aligned}$$ ■

Weakness of the stratified estimator. Although the stratified estimator is straightforward to derive, it has a major limitation: when we chain mixtures together, the computational cost grows exponentially with the chain length. Consider a simple time series example with data $x_1,\dots,x_T$ and latent variables z_t at each time step, where the model factorizes as $p(x_1,\dots,x_T,z_1,\dots,z_T) = p(x_1\,|\,z_1)p(z_1)\prod_{t=2}^{T}p(x_t\,|\,z_t)p(z_t\,|\,z_{t-1})$. For simplicity, take $T=2$. When approximating the posterior $p(z_1,z_2\,|\,x_1,x_2)$, we preserve the structure of the model by using a mixture distribution to approximate each conditional distribution, leading to the following variational distribution:

$$q_\theta(z_1,z_2)=q_\theta(z_1)q_\theta(z_2\,|\,z_1)=\Big(\sum_{k_1=1}^{K_1}\pi_{k_1,\theta}q_{k_1,\theta}(z_1)\Big)\Big(\sum_{k_2=1}^{K_2}\pi_{k_2,\theta}q_{k_2,\theta}(z_2\,|\,z_1)\Big),$$

where $q_{k_1,\theta}$ and $q_{k_2,\theta}$ are the components of the mixture distribution for z_1 and $z_2\,|\,z_1$, respectively. Assume the components of the mixture are reparameterizable; that is, if $\epsilon_1,\epsilon_2\sim q_{\text{base}}$ are independent samples from the base distribution, then

$$z_1=\mathcal{T}_{k_1,\theta_1}(\epsilon_1)\sim q_{k_1,\theta},\qquad\text{and}\qquad z_2=\mathcal{T}_{k_2,\theta}(z_1,\epsilon_2)\sim q_{k_2,\theta|z_1}.$$

A version of the stratified estimator for this model is given by

$$\hat{g}^{(K_1,K_2)}\colon(\epsilon_1,\epsilon_2)\mapsto\sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}\nabla_\theta\pi_{k_1,\theta}\pi_{k_2,\theta}\mathcal{L}_\theta(\mathcal{T}_{k_1,\theta}(\epsilon_1),\mathcal{T}_{k_2,\theta}(\mathcal{T}_{k_1,\theta}(\epsilon_1),\epsilon_2)),$$

which requires $K_1 \times K_2$ model evaluations per gradient. Its derivation mirrors the proof of the stratified estimator. The exponential growth in evaluations is clear: the joint distribution of z_1 and z_2 is a $K_1 \times K_2$-component mixture, and the stratified estimator evaluates the model at each (k_1, k_2) component. In Section 5.3, we illustrate this behavior with a deep Markov model for music generation, which uses a chain of mixtures to model the latent variables. Even for sequences as short as $T = 60$, the required evaluations render the stratified estimator infeasible. This motivates single-sample estimators, which we explore in the following sections.

A note on noise samples. Estimator 1 draws K samples from the base distribution, following Morningstar et al. (2021). The original stratified estimator for variational inference (Roeder et al., 2017) instead uses a single base sample to generate the mixture-component samples. Despite this difference, both have the same expectation and the same limitation—even with only one sample from the noise distribution, $\mathcal{L}_\theta$ is evaluated at K distinct points.

3.1 Sticking-the-Landing

Our primary focus is ELBO optimization, for which Roeder et al. (2017) recommend using the Sticking-the-Landing (STL) modification. This approach eliminates a term that, despite having zero expectation, contributes to the variance of the estimator. As a result, the modified estimator has zero variance when q_θ matches the true posterior, in contrast to the unmodified version, offering a (potential) variance reduction. We introduce the estimator.

Notation *Define $\mathcal{L}_{\bar{\theta}} \colon z \mapsto \ln p(x, z) - \ln q_{\bar{\theta}}(z)$. Here, $q_{\bar{\theta}}(z) = \sum_{k=1}^{K} \pi_{k,\bar{\theta}} q_{k,\bar{\theta}}(z)$ denotes the density parameterized by θ with gradients stopped. This notation indicates that, within $\mathcal{L}_{\bar{\theta}}$, the parameter θ is treated as a constant with respect to differentiation, effectively ignoring the dependency of $\mathcal{L}$ on θ through q_θ in gradient computation.*

Estimator 2 [Stratified w/STL] Let $\mathcal{L}_\theta(z) = \ln p(x, z) - \ln q_\theta(z)$ and let $\epsilon_1, \ldots, \epsilon_K \sim q_{\text{base}}$ be i.i.d. from the base distribution. Then the estimator

$$\hat{g}^{\text{STL}_K} \colon (\epsilon_1, \ldots, \epsilon_K) \mapsto \nabla_\theta \sum_{k=1}^{K} \pi_{k,\theta} \mathcal{L}_{\bar{\theta}}(\mathcal{T}_{k,\theta}(\epsilon_k)), \tag{7}$$

is an unbiased estimator of $\nabla_\theta \, \mathbb{E}_{q_\theta} \, \mathcal{L}_\theta$.

Proof Since $\mathcal{L}_\theta(z) = \ln p(x, z) - \ln q_\theta(z)$, and using the unbiasedness of the stratified estimator $\hat{g}^{(K)}$ (Estimator 1), we have

$$\begin{aligned}
\nabla_\theta \, \mathbb{E}_{q_\theta} \, \mathcal{L}_\theta &= \mathbb{E}_{\epsilon_{1:K}}[\hat{g}^{(K)}(\epsilon_1, \ldots, \epsilon_K)] \qquad (8) \\
&= \mathbb{E}_{\epsilon_{1:K}}\Big[\nabla_\theta \sum_{k=1}^{K} \pi_{k,\theta}\big(\ln p(\mathcal{T}_{k,\theta}(\epsilon_k), x) - \ln q_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\big)\Big] \\
&= \mathbb{E}_{\epsilon_{1:K}}\Big[\nabla_\theta \sum_{k=1}^{K} \pi_{k,\theta} \ln p(\mathcal{T}_{k,\theta}(\epsilon_k), x)\Big] - \mathbb{E}_{\epsilon_{1:K}}\underbrace{\Big[\nabla_\theta \sum_{k=1}^{K} \pi_{k,\theta} \ln q_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\Big]}_{(\ddagger)}.
\end{aligned}$$

Let us focus on (‡). For every ϵ_k, it holds that

$$\begin{aligned}\nabla_\theta(\pi_{k,\theta}\ln q_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))) &= (\nabla_\theta\pi_{k,\theta})\ln q_\theta(\mathcal{T}_{k,\theta}(\epsilon_k)) + \pi_{k,\theta}\nabla_\theta \ln q_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\\ &= (\nabla_\theta\pi_{k,\theta})\ln q_\theta(\mathcal{T}_{k,\theta}(\epsilon_k)) + \pi_{k,\theta}\nabla_\theta\mathcal{T}_{k,\theta}(\epsilon_k)\cdot\nabla_z \ln q_\theta(z)\big|_{z=\mathcal{T}_{k,\theta}(\epsilon_k)}\\ &\quad + \pi_{k,\theta}\nabla_\theta \ln q_\theta(z)\big|_{z=\mathcal{T}_{k,\theta}(\epsilon_k)}\\ &= \nabla_\theta(\pi_{k,\theta}\ln q_{\bar\theta}(\mathcal{T}_{k,\theta}(\epsilon_k))) + \pi_{k,\theta}\nabla_\theta \ln q_\theta(z)\big|_{z=\mathcal{T}_{k,\theta}(\epsilon_k)}.\end{aligned}$$

After taking expectations, Eq. (8) becomes

$$\begin{aligned}\nabla_\theta\,\mathbb{E}_{q_\theta}\,\mathcal{L}_\theta = \mathbb{E}_{\epsilon_{1:K}}\Big[\nabla_\theta\sum_{k=1}^K \pi_{k,\theta}\ln p(\mathcal{T}_{k,\theta}(\epsilon_k),x)\Big]\\ - \mathbb{E}_{\epsilon_{1:K}}\Big[\nabla_\theta\sum_{k=1}^K \pi_{k,\theta}\ln q_{\bar\theta}(\mathcal{T}_{k,\theta}(\epsilon_k))\Big] - \mathbb{E}_{\epsilon_{1:K}}\Big[\sum_{k=1}^K \pi_{k,\theta}\nabla_\theta \ln q_{\bar\theta}(z)\big|_{z=\mathcal{T}_{k,\theta}(\epsilon_k)}\Big].\end{aligned}$$

It then follows from $\mathcal{L}_{\bar\theta}(\mathcal{T}_{k,\theta}(\epsilon_k)) = \ln p(\mathcal{T}_{k,\theta}(\epsilon_k),x) - \ln q_{\bar\theta}(\mathcal{T}_{k,\theta}(\epsilon_k))$ that

$$\begin{aligned}\nabla_\theta\,\mathbb{E}_{q_\theta}\,\mathcal{L}_\theta &= \mathbb{E}_{\epsilon_{1:K}}\Big[\nabla_\theta\sum_{k=1}^K \pi_{k,\theta}\mathcal{L}_{\bar\theta}(\mathcal{T}_{k,\theta}(\epsilon_k))\Big] - \mathbb{E}_{\epsilon_{1:K}}\Big[\sum_{k=1}^K \pi_{k,\theta}\nabla_\theta \ln q_{\bar\theta}(z)\big|_{z=\mathcal{T}_{k,\theta}(\epsilon_k)}\Big]\\ &= \mathbb{E}_{\epsilon_{1:K}}\Big[\hat g^{\mathrm{STL}_K}(\epsilon_1,\dots,\epsilon_K)\Big] - \sum_{k=1}^K \pi_{k,\theta}\,\mathbb{E}_{\epsilon_k\sim q_{\mathrm{base}}}\Big[\nabla_\theta \ln q_{\bar\theta}(z)\big|_{z=\mathcal{T}_{k,\theta}(\epsilon_k)}\Big].\end{aligned}$$

To finish, it suffices to show that the expectation of the second term is zero. Indeed,

$$\begin{aligned}\sum_{k=1}^K \pi_{k,\theta}\,\mathbb{E}_{\epsilon_k}[\nabla_\theta \ln q_\theta(z)|_{z=\mathcal{T}_{k,\bar\theta}(\epsilon_k)}] &= \sum_{k=1}^K \pi_{k,\bar\theta}\,\mathbb{E}_{z_k\sim q_{k,\bar\theta}}[\nabla_\theta \ln q_\theta(z_k)]\\ &= \mathbb{E}_{z\sim q_{\bar\theta}}[\nabla_\theta \ln q_\theta(z)\big|_{\theta=\bar\theta}]\\ &= \nabla_\theta\int_{\mathcal{Z}} q_\theta(z)\big|_{\theta=\bar\theta}\,dz = \nabla_\theta 1 = 0.\end{aligned}$$

■

Note *The STL modification applies whenever the gradient of the expectation contains the term $\nabla_\theta \ln q_\theta(z)$, which holds for several estimators in the following sections. Nevertheless, we present each estimator in its full form so the discussion is not confined to the ELBO objective. In the Experiments section, we compare both variants for every estimator and find that the benefits of STL are mixed.*

4. Single Sample Estimators

To address the stratified estimator's limitations, we seek estimators requiring a number of samples independent of the number of mixture components. This section focuses on single-sample estimators that evaluate $\mathcal{L}_\theta$ or its gradient at just one point. This approach helps not only with chained mixtures—where sample requirements grow exponentially with chain length (Figurnov et al., 2018)—but also when evaluating $\mathcal{L}_\theta$ is expensive. In practice, they made feasible methods such as the variational marginal particle filter of Lai et al. (2022).

Single-sample estimators involve different trade-offs, and no single estimator works best for all scenarios. Their performance and resource requirements vary significantly depending on the problem's structure. We organize them into two main types: score-function estimators, which do not require reparameterization (Section 4.1); and reparameterization-based estimators—a.k.a. pathwise gradient estimators. The latter category includes the post-stratified estimator (Section 4.3), which applies whenever components are reparameterizable; the estimator of Jankowiak and Obermeyer (2018) (Section 4.4), a specialized version for diagonal Gaussians with `softmax` weights; and the implicit reparameterization estimator of Figurnov et al. (2018) (Section 4.5), which handles cases where explicit reparameterization is unavailable.

4.1 Score-Function Estimator: Non-Reparameterizable Components

In the most general case, we require only that mixture components and weights be differentiable in their parameters. In this setting, we use the score-function estimator (Williams, 1992) to estimate the gradient of $\mathbb{E}_{q_\theta}[\mathcal{L}_\theta]$ with respect to the mixture parameters. We specialize the score-function estimator to mixtures, which will be relevant later when we discuss the post-stratified estimator.

Estimator 3 [Score-function estimator] Take a sample z drawn from the mixture distribution q_θ, and let $q_\theta(k \mid z) = \frac{\pi_{k,\theta} q_{k,\theta}(z)}{q_\theta(z)}$ denote the probability of z being drawn from the kth component of the mixture. Then, for any differentiable test function $\mathcal{L}_\theta$, the estimator

$$\hat{g}_\theta^{\mathrm{SF}} \colon z \mapsto \mathcal{L}_\theta(z) \sum_{k=1}^{K} q_\theta(k \mid z) \nabla_\theta \ln(\pi_{k,\theta} q_{k,\theta}(z)) + \nabla_\theta \mathcal{L}_\theta(z) \tag{9}$$

is an unbiased estimator of $\nabla_\theta \, \mathbb{E}_{q_\theta} \, \mathcal{L}_\theta$.

Although this estimator's unbiasedness is well known, we include a proof because we present it in an unconventional form to highlight the specifics of mixtures.

Proof Notice first that

$$\begin{aligned}
\nabla_\theta \ln q_\theta(z) = \frac{1}{q_\theta(z)} \nabla_\theta q_\theta(z) &= \frac{1}{q_\theta(z)} \sum_{k=1}^{K} \nabla_\theta (\pi_{k,\theta} q_{k,\theta}(z)) \\
&= \frac{1}{q_\theta(z)} \sum_{k=1}^{K} \pi_{k,\theta} q_{k,\theta}(z) \frac{\nabla_\theta (\pi_{k,\theta} q_{k,\theta}(z))}{\pi_{k,\theta} q_{k,\theta}(z)} \\
&= \sum_{k=1}^{K} q_\theta(k \mid z) \nabla_\theta \ln(\pi_{k,\theta} q_{k,\theta}(z)).
\end{aligned} \tag{10}$$

We can now follow the usual proof. Differentiating under the integral,

$$\begin{aligned}
\nabla_\theta \, \mathbb{E}_{z \sim q_\theta}[\mathcal{L}_\theta(z)] &= \int_{\mathcal{Z}} \big(\mathcal{L}_\theta(z) \nabla_\theta q_\theta(z) + (\nabla_\theta \mathcal{L}_\theta(z)) q_\theta(z) \big) \, dz \\
&= \mathbb{E}_{z \sim q_\theta} \Big[\mathcal{L}_\theta(z) \nabla_\theta \ln q_\theta(z) + \nabla_\theta \mathcal{L}_\theta(z) \Big].
\end{aligned}$$

To complete the proof, we plug in (10) to obtain

$$\nabla_\theta \mathbb{E}_{z\sim q_\theta}[\mathcal{L}_\theta(z)] = \mathbb{E}_{z\sim q_\theta}\Big[\mathcal{L}_\theta(z)\sum_{k=1}^{K} q_\theta(k\,|\,z)\nabla_\theta \ln(\pi_{k,\theta}q_{k,\theta}(z)) + \nabla_\theta\mathcal{L}_\theta(z)\Big] = \mathbb{E}_{q_\theta}[\hat{g}_\theta^{\mathrm{SF}}]. \quad \blacksquare$$

When $\mathcal{L}_\theta$ is the ELBO, i.e., $\mathcal{L}_\theta(z) = \ln p(x,z) - \ln q_\theta(z)$ with $p(x,z)$ independent of θ, we have $\mathbb{E}_{q_\theta}[\nabla_\theta\mathcal{L}_\theta] = -\,\mathbb{E}_{q_\theta}[\nabla_\theta \ln q_\theta] = \mathbf{0}$, and the second term in (9) vanishes in expectation. Dropping it recovers the familiar REINFORCE form $\hat{g}_\theta^{\mathrm{SF}}(z) = \mathcal{L}_\theta(z)\nabla_\theta \ln q_\theta(z)$, which is the form we used in the experiments.

4.2 Randomized Components

The stratified estimator is limited by the need to evaluate the test function $\mathcal{L}_\theta$ at K points, one per mixture component. One way to avoid this cost is to sample a component k, draw z_k from $q_{k,\theta}$, and compute the gradients via the reparameterization of z_k. This procedure can reduce the computation to as few as a single evaluation of $\mathcal{L}_\theta$. In the survey sampling analogy of Section 3, this two-stage procedure corresponds to *simple random sampling*.

Estimator 4 [Randomized components] Let $k \sim \mathrm{Cat}(\pi_{1,\theta},\dots,\pi_{K,\theta})$ be a sample from the categorical distribution with probabilities $\pi_{1,\theta},\dots,\pi_{K,\theta}$, and let $z_k \sim q_{k,\theta}$ be a sample from the kth component of the mixture. Then, the estimator

$$\hat{g}_\theta^{\mathrm{R}\kappa}\colon (k,z_k) \mapsto \nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\big|_{\epsilon_k=\mathcal{T}_{k,\theta}^{-1}(z_k)} + \mathcal{L}_\theta(z_k)\nabla_\theta \ln \pi_{k,\theta}$$

is an unbiased estimator of $\nabla_\theta \mathbb{E}_{q_\theta}\mathcal{L}_\theta$.

Proof By Estimator 1b, the estimator and stratified estimator have equal expectations:

$$\begin{aligned}
\mathbb{E}_k\big[\mathbb{E}_{z_k}\big[\hat{g}_\theta^{\mathrm{R}\kappa}(k,z_k)\big]\big] &= \mathbb{E}_k\big[\mathbb{E}_{\epsilon_k}\big[\hat{g}_\theta^{\mathrm{R}\kappa}(k,\mathcal{T}_{k,\theta}(\epsilon_k))\big]\big] \\
&= \mathbb{E}_{\epsilon_{1:K}}\Big[\sum_{k=1}^{K}\pi_{k,\theta}\big(\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k)) + \mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\nabla_\theta\ln\pi_{k,\theta}\big)\Big] \\
&= \mathbb{E}_{\epsilon_{1:K}}[\hat{g}^{(K)}(\epsilon_1,\dots,\epsilon_K)]. \qquad \blacksquare
\end{aligned}$$

4.3 Post-Stratified Estimator

An alternative way to overcome the weakness of the stratified estimator is to sample z from the mixture q_θ. Then, for each stratum k, consider what the gradient of $\mathcal{L}_\theta$ would look like had z been drawn from k. This estimator, sketched in Jankowiak and Karaletsos (2019) as the *hybrid estimator*, is unbiased and has lower variance than the randomized components estimator. We provide a more elementary derivation that does not invoke vector fields or the continuity equation.

Estimator 5 [Post-stratified] Let $z \sim q_\theta$ and define $q_\theta(k\,|\,z) = \frac{\pi_{k,\theta}q_{k,\theta}(z)}{q_\theta(z)}$, the posterior probability that z comes from component k; assume that $\mathcal{T}_{k,\theta}$ is invertible for each k. Then

$$\hat{g}_\theta^{\mathrm{PS}}\colon z \mapsto \sum_{k=1}^{K} q_\theta(k\,|\,z)\Big(\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\big|_{\epsilon=\mathcal{T}_{k,\theta}^{-1}(z)} + \mathcal{L}_\theta(z)\nabla_\theta\ln\pi_{k,\theta}\Big) \tag{11}$$

is an unbiased estimator of $\nabla_\theta \mathbb{E}_{q_\theta}\mathcal{L}_\theta$.

Proof Observe that for integrable functions h, the following property holds:

$$\pi_{k,\theta}\,\mathbb{E}_{q_{k,\theta}}[h(z)] = \int_{\mathcal{Z}} h(z)\frac{\pi_{k,\theta}q_{k,\theta}(z)}{q_\theta(z)}q_\theta(z)\,dz = \mathbb{E}_{q_\theta}[h(z)q_\theta(k\,|\,z)]. \tag{12}$$

To derive the result, we use the alternative form of the stratified estimator in Eq. (6), i.e.,

$$\begin{aligned}
\nabla_\theta\,\mathbb{E}_{q_\theta}\,\mathcal{L}_\theta &= \mathbb{E}_{\epsilon_{1:K}\sim q_{\text{base}}}\Big[\sum_{k=1}^{K}\pi_{k,\theta}\Big(\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k)) + \mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\nabla_\theta\ln\pi_{k,\theta}\Big)\Big] \\
&= \sum_{k=1}^{K}\mathbb{E}_{z_k\sim q_{k,\theta}}\Big[\pi_{k,\theta}\Big(\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\big|_{\epsilon_k=\mathcal{T}_{k,\theta}^{-1}(z_k)} + \mathcal{L}_\theta(z_k)\nabla_\theta\ln\pi_{k,\theta}\Big)\Big] \\
&= \sum_{k=1}^{K}\mathbb{E}_{z\sim q_\theta}\Big[q_\theta(k\,|\,z)\big(\nabla_\theta\mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k)) + \mathcal{L}_\theta(z)\nabla_\theta\ln\pi_{k,\theta}\big)\big|_{\epsilon=\mathcal{T}_{k,\theta}^{-1}(z)}\Big] \quad ;\text{ by (12)} \\
&= \mathbb{E}_{z\sim q_\theta}[\hat{g}_\theta^{\text{PS}}].
\end{aligned}$$

■

Relation to survey sampling. In survey sampling, *post-stratification* reweights units after collection so that weighted counts match known population totals. In our setting, we also reweight post-sample, treating each sampled unit as fractionally belonging to all components according to its membership probabilities and averaging the component-specific gradients. Both adjust contributions after sampling, but the targets differ: post-stratification matches population composition, whereas we target the expected output under the mixture. This modification, which permits using as few as one sample, shares the spirit of the *endogenous post-stratification* of Breidt and Opsomer (2008); hence the *post-stratified* moniker.

Practical computation. To realize the benefit of a single-sample estimator, we must compute it using one evaluation of $\mathcal{L}_\theta$ and its gradient. Accordingly, we express the estimator as the sum of a reparameterization term and a score-function term. That is,

$$\begin{aligned}
\hat{g}_\theta^{\text{PS}}(z) = \overbrace{\nabla_\theta\mathcal{L}_\theta(z) + \Big(\sum_{k=1}^{K} q_\theta(k\,|\,z)\nabla_\theta\mathcal{T}_{k,\theta}(\epsilon_k)\big|_{\epsilon_k=\mathcal{T}_{k,\theta}^{-1}(z)}\Big)\cdot\nabla_z\mathcal{L}_\theta(z)}^{\text{reparameterization term}} \\
\underbrace{+\,\mathcal{L}_\theta(z)\sum_{k=1}^{K} q_\theta(k\,|\,z)\nabla_\theta\ln\pi_{k,\theta}}_{\text{score function term}}\,.
\end{aligned} \tag{13}$$

The reparameterization term can be efficiently computed via automatic differentiation using a surrogate objective S_θ^{PS}. Define noise variables $\tilde{\epsilon}_k = \mathcal{T}_{k,\theta}^{-1}(z)$ so that $\mathcal{T}_{k,\theta}(\tilde{\epsilon}_k) = z$ for each k. The surrogate function is:

$$S_\theta^{\text{PS}}\colon \tilde{\epsilon}_1,\ldots,\tilde{\epsilon}_K \mapsto \mathcal{L}_\theta\Big(\sum_{k=1}^{K} q_{\bar{\theta}}(k\,|\,z)\mathcal{T}_{k,\theta}(\epsilon_k)\Big),$$

where $q_{\bar{\theta}}(k\,|\,z)$ is the probability that z comes from component k, treated as constant with respect to θ. The gradient of $S_\theta^{\text{PS}}(\tilde{\epsilon}_1,\ldots,\tilde{\epsilon}_K)$ recovers the reparameterization term of $\hat{g}_\theta^{\text{PS}}$.

Variance comparison. The following proposition proves that the post-stratified estimator $\hat{g}_\theta^{\mathrm{PS}}$ has variance no larger than the randomized components estimator $\hat{g}_\theta^{\mathrm{R}\kappa}$. Both estimators require similar computation—evaluation of $\mathcal{L}_\theta$ and its gradients—but the post-stratified adjustments reduce variance. The proof, related to the Rao-Blackwell theorem, also interprets the post-stratified estimator as the conditional expectation of the randomized components estimator. The proposition is stated for scalar-valued estimators; for vector-valued estimators, the result holds componentwise.

Proposition 2 *Suppose the gradient is computed with respect to a scalar quantity. Then, the post-stratified estimator is the conditional expectation of the randomized components estimator given the sample z. That is,*

$$\hat{g}_\theta^{\mathrm{PS}}(z) = \mathbb{E}_{\hat{g}_\theta^{\mathrm{R}\kappa} \mid z}[\hat{g}_\theta^{\mathrm{R}\kappa} \mid z].$$

Moreover, the variance of the post-stratified estimator $\hat{g}_\theta^{\mathrm{PS}}$ is bounded by that of the randomized components estimator $\hat{g}_\theta^{\mathrm{R}\kappa}$:

$$\mathrm{Var}[\hat{g}_\theta^{\mathrm{PS}}] \leq \mathrm{Var}\big[\hat{g}_\theta^{\mathrm{R}\kappa}\big].$$

Proof For a given z, consider the random variable given by the randomized components Estimator 4 condition on z, that is $\hat{g}_\theta^{\mathrm{R}\kappa} \mid z$. This random variable will take the value of the estimator $\hat{g}_\theta^{\mathrm{R}\kappa}(k, z)$ with probability $q_\theta(k \mid z)$. It follows directly that

$$\begin{aligned}
\mathbb{E}_{\hat{g}_\theta^{\mathrm{R}\kappa} \mid z}[\hat{g}_\theta^{\mathrm{R}\kappa} \mid z] &= \sum_{k=1}^{K} q_\theta(k \mid z) \hat{g}_\theta^{\mathrm{R}\kappa}(k, z) \\
&= \sum_{k=1}^{K} q_\theta(k \mid z) \big(\nabla_\theta \mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon_k))\big|_{\epsilon = \mathcal{T}_{k,\theta}^{-1}(z_k)} + \mathcal{L}_\theta(z_k) \nabla_\theta \ln \pi_{k,\theta} \big) = \hat{g}_\theta^{\mathrm{PS}}(z).
\end{aligned}$$

The second part then follows from the law of total variance. That is,

$$\begin{aligned}
\mathrm{Var}_{\hat{g}_\theta^{\mathrm{R}\kappa}}\big[\hat{g}_\theta^{\mathrm{R}\kappa}\big] &= \mathbb{E}_{z \sim q_\theta}\big[\mathrm{Var}_{\hat{g}_\theta^{\mathrm{R}\kappa} \mid z}\big[\hat{g}_\theta^{\mathrm{R}\kappa} \mid z\big]\big] + \mathrm{Var}_{z \sim q_\theta}\big[\mathbb{E}_{\hat{g}_\theta^{\mathrm{R}\kappa} \mid z}\big[\hat{g}_\theta^{\mathrm{R}\kappa} \mid z\big]\big] \\
&\geq \mathrm{Var}_{z \sim q_\theta}\big[\mathbb{E}_{\hat{g}_\theta^{\mathrm{R}\kappa} \mid z}\big[\hat{g}_\theta^{\mathrm{R}\kappa} \mid z\big]\big] \\
&= \mathrm{Var}_{z \sim q_\theta}\big[\hat{g}_\theta^{\mathrm{PS}}(z)\big].
\end{aligned}$$

■

4.4 Special Cases with `softmax` Mixture Weights: J&K Estimator

An alternative method for deriving the gradient estimators uses the transport equation [cf. Villani (2021); Ambrosio et al. (2005)]. Jankowiak and Obermeyer (2018) initiated this approach for variational inference, replacing $\nabla_\theta \mathcal{T}_\theta$ in $\nabla_\theta \mathcal{T}_\theta \cdot \nabla_z \mathcal{L}_\theta(z)\big|_{z=\mathcal{T}_\theta} + \nabla_\theta \mathcal{L}_\theta(z)\big|_{z=\mathcal{T}_\theta}$ with a vector field $\mathbf{V}^\theta(z)$ that satisfies the transport equation. Jankowiak and Karaletsos (2019) later derived vector fields satisfying the transport equation for both component parameters and mixture weights. Their construction yields a family of estimators, one for each of four diagonal-Gaussian approximating subfamilies with `softmax`-parameterized weights: per-component diagonal covariance, shared diagonal covariance, Gaussian scale mixture, and zero-mean Gaussian scale mixture. We take the most general instance, with per-component diagonal covariance, as a representative for evaluation and comparison, and refer to it throughout as the *J&K estimator*.

Estimator 6 [Jankowiak & Karaletsos—diagonal Gaussian] Suppose q_θ is a mixture of diagonal Gaussians, and let ℓ_k be the `softmax` logits satisfying $\pi_k = e^{\ell_k}/\sum_{j=1}^K e^{\ell_j}$. Further, assume that each component is parameterized with D-dimensional vectors $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ such that, if $z_k \sim q_{k,\theta}$, then $z_{kd} \sim \mathcal{N}(\mu_{kd}, \sigma_{kd}^2)$. Define the vector field matrix $\mathbf{V}^\theta(z)$, with rows at each z given by:

$$\mathbf{V}^\theta_{\cdot,r} \colon z \mapsto \begin{cases} \mathbf{v}^{\ell_k}(z) & \text{if } \theta_r \text{ corresponds to the } \texttt{softmax} \text{ logit } \ell_k, \\ \mathbf{0} & \text{otherwise,} \end{cases}$$

where $\mathbf{v}^{\ell_k}(z)$ will be defined later. Then, the estimator

$$\hat{g}_\theta^{\mathrm{JK}} \colon z \mapsto \sum_{k=1}^K q_\theta(k \mid z) \nabla_\theta \mathcal{L}_\theta(\mathcal{T}_{k,\theta}(\epsilon))\big|_{\epsilon=\mathcal{T}_{k,\theta}^{-1}(z)} + \mathbf{V}^\theta(z) \cdot \nabla_z \mathcal{L}_\theta(z)$$

is an unbiased[1] estimator of $\nabla_\theta \mathbb{E}_{q_\theta} \mathcal{L}_\theta$.

We now define the vector field $\mathbf{v}^{\ell_k}(z)$, which provides the gradients for the mixture weight parameters, by introducing intermediate quantities. For each dimension d, let $\sigma_d^0 \equiv \min(\sigma_{1d}, \sigma_{2d}, \ldots, \sigma_{Kd})$ denote the smallest standard deviation across all components, and form the vector $\boldsymbol{\sigma}^0$ from these minimum values. For each input z and for all k and d, define:

$$\tilde{\epsilon}_{kd} = \frac{z_d - \boldsymbol{\mu}_{kd}}{\boldsymbol{\sigma}_{kd}}, \qquad \bar{\epsilon}_{kd} = \frac{z_d - \boldsymbol{\mu}_{kd}}{\boldsymbol{\sigma}_d^0}, \qquad \text{and} \qquad r_{kd}^2 = \sum_{i<d} \tilde{\epsilon}_{ki}^2 + \sum_{i>d} \bar{\epsilon}_{ki}^2.$$

Here $\tilde{\epsilon}_{kd}$ is the standardized deviation of z_d from component k, $\bar{\epsilon}_{kd}$ uses the minimum standard deviation across components, and r_{kd}^2 combines squared deviations from other dimensions. The vector field $\mathbf{v}^{\ell_k}(z)$ is then:

$$\check{\mathbf{v}}^k \colon z \mapsto \sum_d \frac{(\Phi(\tilde{\epsilon}_{kd}) - \Phi(\bar{\epsilon}_{kd}))\phi(r_{kd}^2)}{q_\theta \prod_{i<d} \sigma_{ki} \prod_{i>d} \sigma_d^0} \mathbf{e}_d, \quad \text{and} \quad \mathbf{v}^{\ell_k} \colon z \mapsto \pi_k \Big(\check{\mathbf{v}}^k(z) - \sum_j \pi_j \check{\mathbf{v}}^j(z) \Big),$$

where Φ and ϕ are the standard Gaussian CDF and PDF, and $\mathbf{e}_d$ is the dth unit vector.

4.4.1 Fixed mixture weights

A simpler variant fixes the mixture weights $\pi_1, \ldots, \pi_K$ to constants, either uniform ($1/K$) or randomly chosen. With constant mixture weights, only gradients with respect to the component parameters are needed. Here, the post-stratified and J&K estimators coincide and apply to any reparameterizable components, not just diagonal Gaussians.

Estimator 7 [Jankowiak & Karaletsos $\hat{g}^{\mathrm{JK}}$with fixed weights π] Let q_θ be a mixture distribution with fixed mixture weights $\bar{\pi}_1, \ldots, \bar{\pi}_K$, where each component k is reparameterizable through $\mathcal{T}_{k,\theta}$. Let $z \sim q_\theta$ and let $q_\theta(k \mid z)$ denote the posterior probability that z came from component k. Then the estimator

$$\hat{g}_\theta^{\mathrm{JK}-\bar{\pi}} \colon z \mapsto \nabla_\theta \mathcal{L}_\theta(z) + \Big(\sum_{k=1}^K q_\theta(k \mid z) \nabla_\theta \mathcal{T}_{k,\theta}(\epsilon)\big|_{\epsilon=\mathcal{T}_{k,\theta}^{-1}(z)} \Big) \cdot \nabla_z \mathcal{L}_\theta(z) \tag{14}$$

is unbiased for $\nabla_\theta \mathbb{E}_{q_\theta} \mathcal{L}_\theta$.

1. See Jankowiak and Karaletsos (2019) for the proof of unbiasedness and the vector field expressions.

Proof This is the post-stratified estimator of Eq. (13) with fixed weights, so the score-function term vanishes since $\nabla_\theta \ln \pi_{k,\theta} = \mathbf{0}$ for all k. ■

4.5 Implicit Reparameterization Gradients

The last single-sample estimator we consider is the *implicit reparameterization gradients estimator*, introduced in general form by Figurnov et al. (2018). Several authors independently applied similar ideas to more restricted settings: Jankowiak and Obermeyer (2018); Domke and Sheldon (2018); Salimans and Knowles (2013); Hoffman and Blei (2015) for univariate distributions or specific families like elliptical distributions. Notably, Graves (2016) developed implicit reparameterization for mixtures of multivariate distributions. The method, which applies beyond mixture distributions, uses implicit differentiation with an invertible standardization function $\mathcal{S}_\theta$ that satisfies $\mathcal{S}_\theta(z) \sim q_{\text{base}}$ whenever $z \sim q_\theta$. In other words, $\mathcal{S}_\theta$ maps samples from q_θ to a base distribution q_{base}, the reverse of the typical reparameterization map $\mathcal{T}_\theta$. This approach is useful when the reparameterization map is unavailable but the standardization function is known.

We start by stating a proposition needed for the derivation of the estimator. We then develop the estimator from the general case through efficient implementations for mixtures of Gaussian, Student's t, and Cauchy components.

Proposition 3 (Implicit Reparameterization) *Suppose $\mathcal{S}_\theta(z)$ satisfies $\mathcal{S}_\theta(z) \sim q_{base}$ whenever $z \sim q_\theta$. Assume $\mathcal{S}_\theta(z)$ is differentiable with respect to z and θ, that for each θ the function $\mathcal{S}_\theta$ is invertible with respect to z, and that the inverse $\mathcal{S}_\theta^{-1}(\epsilon)$ is differentiable with respect to θ. Then*

$$\nabla_\theta \mathcal{S}_\theta^{-1}(\epsilon)\big|_{\epsilon=S_\theta(z)} = -\nabla_\theta \mathcal{S}_\theta(z)(\nabla_z \mathcal{S}_\theta(z))^{-1}. \tag{15}$$

The proposition allows us to compute the gradient of the inverse $\mathcal{T}_\theta = \mathcal{S}_\theta^{-1}$ with respect to θ, even when an explicit form for $\mathcal{T}_\theta$ is unavailable. We follow Figurnov et al. (2018).

Proof Using the identity $\epsilon = \mathcal{S}_\theta(\mathcal{T}_\theta(\epsilon))$, we derive that

$$0 = \nabla_\theta \epsilon = \nabla_\theta(\mathcal{S}_\theta(\mathcal{T}_\theta(\epsilon))) = \nabla_\theta \mathcal{T}_\theta(\epsilon) \nabla_z \mathcal{S}_\theta(z)\big|_{z=\mathcal{T}_\theta(\epsilon)} + \nabla_\theta \mathcal{S}_\theta(z)\big|_{z=\mathcal{T}_\theta(\epsilon)}.$$

That is,

$$\nabla_\theta \mathcal{T}_\theta(\epsilon) = -\nabla_\theta \mathcal{S}_\theta(z)(\nabla_z \mathcal{S}_\theta(z))^{-1}\Big|_{z=\mathcal{T}_\theta(\epsilon)}.$$

Set $\epsilon = \mathcal{S}_\theta(z)$ to finish the proof. ■

Estimator 8 [Implicit reparameterization] Let $\mathcal{S}_\theta$ be a standardization function satisfying the conditions of Proposition 3. Then for any $z \sim q_\theta$, the estimator

$$\hat{g}^{\text{IR}} \colon z \mapsto -\nabla_\theta \mathcal{S}_\theta(z)(\nabla_z \mathcal{S}_\theta(z))^{-1} \nabla_z \mathcal{L}_\theta(z) + \nabla_\theta \mathcal{L}_\theta(z)$$

is unbiased for $\nabla_\theta \mathbb{E}_{q_\theta} \mathcal{L}_\theta$.

Proof Let $\mathcal{T}_\theta = \mathcal{S}_\theta^{-1}$. Then

$$\begin{aligned}
\mathbb{E}_{q_\theta}\, \hat{g}^{\mathrm{IR}} &= \mathbb{E}_{z\sim q_\theta}\Big[-\nabla_\theta \mathcal{S}_\theta(z)(\nabla_z \mathcal{S}_\theta(z))^{-1}\nabla_z \mathcal{L}_\theta(z) + \nabla_\theta \mathcal{L}_\theta(z)\Big] \\
&= \mathbb{E}_{z\sim q_\theta}\Big[\nabla_\theta \mathcal{T}_\theta(\mathcal{S}_\theta(z))\nabla_z \mathcal{L}_\theta(z) + \nabla_\theta \mathcal{L}_\theta(z)\Big] && ;\ \text{by Prop. 3} \\
&= \mathbb{E}_{\epsilon\sim q_{\mathrm{base}}}\big[\nabla_\theta \mathcal{T}_\theta(\epsilon)\nabla_z \mathcal{L}_\theta(\mathcal{T}_\theta(\epsilon)) + \nabla_\theta \mathcal{L}_\theta(z)\big|_{z=\mathcal{T}_\theta(\epsilon)}\big] && ;\ \mathcal{S}_\theta(z)\sim q_{\mathrm{base}} \\
&= \mathbb{E}_{\epsilon\sim q_{\mathrm{base}}}\big[\nabla_\theta \mathcal{L}_\theta(\mathcal{T}_\theta(\epsilon))\big] \\
&= \nabla_\theta\, \mathbb{E}_{q_\theta}\, \mathcal{L}_\theta.
\end{aligned}$$

■

Although the technique could apply more broadly in principle, we have not found cases where implicit reparameterization works but the post-stratified estimator (which requires explicit component reparameterizations) does not. Still, the two approaches differ in approximation quality, robustness, and computational cost; see the experiments section.

The estimator requires a tractable standardization map $\mathcal{S}_\theta$; the remainder of this section constructs these for specific mixture families with efficient gradient algorithms.

4.5.1 Scalar-valued Z

For scalar-valued Z, let $F_\theta(\bar{z}) = \int_{-\infty}^{\bar{z}} q_\theta(z)\, dz$ denote the CDF of q_θ, and let $F_{k,\theta}$ denote the CDF of each component $q_{k,\theta}$. Assume each $F_{k,\theta}$ is differentiable with respect to θ. The CDF F_θ is a natural standardization function, with $q_{\mathrm{base}} = \mathcal{U}(0,1)$.

Estimator 9 [Implicit reparameterization, scalar-valued Z] Let $F_{k,\theta}$ denote the CDF of the component $q_{k,\theta}$, and assume $\frac{\partial F_{k,\theta}}{\partial \theta}$ can be computed for each k. Let $F_\theta = \sum_{k=1}^K \pi_{k,\theta} F_{k,\theta}$ be the CDF of q_θ. Then for any $z \sim q_\theta$, the estimator

$$\hat{g}^{\mathrm{IR}} \colon z \mapsto -\frac{\nabla_\theta F_\theta(z)}{q_\theta(z)}\frac{\partial \mathcal{L}_\theta}{\partial z}(z) + \nabla_\theta \mathcal{L}_\theta(z)$$

is unbiased for $\nabla_\theta\, \mathbb{E}_{q_\theta}\, \mathcal{L}_\theta$.

Proof The result follows from Estimator 8 with $\mathcal{S}_\theta = F_\theta$, because $F_\theta(z) \sim \mathcal{U}(0,1)$ when $z \sim q_\theta$. In particular, $\nabla_z \mathcal{S}_\theta(z) = \partial F_\theta(z)/\partial z = q_\theta(z)$ and $\nabla_\theta \mathcal{S}_\theta(z) = \nabla_\theta F_\theta(z)$. ■

4.5.2 Vector valued Z: multivariate Gaussian, Student's t and Cauchy

Proposition 3 extends to vector-valued $Z \in \mathbb{R}^D$ with gradients replaced by Jacobian transposes where appropriate. The extension requires a standardization function $\mathcal{S}$ satisfying the proposition's conditions. However, the multivariate CDF is unsuitable since it maps $\mathbb{R}^D$ to a scalar and is non-invertible. The key insight of Graves (2016), used by Figurnov et al. (2018), is to use *conditional* CDFs instead. For $Z = (Z^{(1)}, \ldots, Z^{(D)})$, the conditional CDF of $Z^{(d)}$ given $z^{<d} = (z^{(1)}, \ldots, z^{(d-1)})$ is:

$$F^{(d)}(z^{(d)} \mid z^{<d}) = \mathbb{P}(Z^{(d)} \le z^{(d)} \mid Z^{(1)} = z^{(1)}, \ldots, Z^{(d-1)} = z^{(d-1)})$$

These conditional CDFs define the map:

$$\begin{aligned}\mathcal{S}\colon \mathbb{R}^D \times \Theta &\to [0,1]^D \\ (z^{(1)}, \ldots, z^{(D)}, \theta) &\mapsto \big(F_\theta^{(1)}(z^{(1)}), F_\theta^{(2)}(z^{(2)} \mid z^{(1)}), \ldots, F_\theta^{(D)}(z^{(D)} \mid z^{<D})\big),\end{aligned}$$

which is invertible for each θ and satisfies Proposition 3.

This construction works for mixtures of multivariate Gaussian, Student's t, and Cauchy distributions. Student's t mixtures are especially relevant: they capture heavy-tailed distributions in images (van den Oord and Schrauwen, 2014) and improve robustness to outliers. Domke and Sheldon (2018) proposed elliptical distributions, including Student's t, as variational families for black-box VI; mixtures extend this flexibility. The Cauchy distribution provides a practical example of a distribution without a well-defined mean or variance.

4.5.3 Structure of Mixture Conditional CDFs

For mixture distributions, the conditional CDF of $z^{(d)}$ given $z^{<d} = (z^{(1)}, \ldots, z^{(d-1)})$ is a convex combination of the components' conditional CDFs.

Proposition 4 (Graves) *Let $F_{k,\theta}^{(d)}$ denote the conditional CDF of the dth dimension in component k, and $F_\theta^{(d)}$ the conditional CDF of the dth dimension of the mixture. Then*

$$F_\theta^{(d)}(z^{(d)} \mid z^{<d}) = \sum_{k=1}^{K} w_{k,\theta}^{<d} F_{k,\theta}^{(d)}(z^{(d)} \mid z^{<d}), \tag{16}$$

where $w_{k,\theta}^{<d} = \frac{\pi_{k,\theta} q_{k,\theta}(z^{<d})}{\sum_{k'} \pi_{k',\theta} q_{k',\theta}(z^{<d})} = \mathbb{P}_\theta(k \mid z^{<d})$.

Proof Following Graves (2016), the conditional density is a mixture of component conditional densities:

$$\begin{aligned}q_\theta(z^{(d)} \mid z^{<d}) = \frac{q_\theta(z^{(d)}, z^{<d})}{q_\theta(z^{<d})} &= \frac{\sum_{k=1}^K \pi_{k,\theta}\, q_{k,\theta}(z^{(d)}, z^{<d})}{\sum_{k'=1}^K \pi_{k',\theta}\, q_{k',\theta}(z^{<d})} \\ &= \sum_{k=1}^K \frac{\pi_{k,\theta}\, q_{k,\theta}(z^{<d})}{\sum_{k'=1}^K \pi_{k',\theta}\, q_{k',\theta}(z^{<d})} \frac{q_{k,\theta}(z^{(d)}, z^{<d})}{q_{k,\theta}(z^{<d})} \\ &= \sum_{k=1}^K \mathbb{P}_\theta(k \mid z^{<d})\, q_{k,\theta}(z^{(d)} \mid z^{<d}).\end{aligned}$$

The result follows from the definition of conditional CDF. ■

4.5.4 Computing Conditionals: Concrete Cases

The main challenge of this approach is computing, for each component k, the marginal densities $q_{k,\theta}(z^{(1)}, \ldots, z^{(d)})$ and conditional CDFs $F_{k,\theta}^{(d)}(z^{(d)} \mid z^{<d})$ needed to construct $\mathcal{S}$. The component's distribution family can differ across components as long as these quantities are

available. Prior work computed these for independent dimensions (Graves, 2016; Figurnov et al., 2018) and multivariate Gaussians (Figurnov et al., 2018). We extend these results to multivariate Student's t and Cauchy distributions.

Independent dimensions. Independence yields:

$$q_{k,\theta}(z^{(1)}, \ldots, z^{(d)}) = \prod_{d'=1}^{d} q_{k,\theta}(z^{(d')}), \quad \text{and} \quad F^{(d)}_{k,\theta}(z^{(d)} \mid z^{<d}) = F^{(d)}_{k,\theta}(z^{(d)}).$$

Gaussian, Student's t, and Cauchy. These distributions are *consistent elliptical distributions* (Kano, 1994)—elliptical distributions closed under marginalization. As location-scale families, we write them with parameters $\mu \in \mathbb{R}^D$ and covariance $\Sigma \succ \mathbf{0}$: $\mathcal{N}(\mu, \Sigma)$ for Gaussian, $\mathcal{ST}(\mu, \Sigma, \nu)$ for Student's t with degrees of freedom $\nu > 0$, and $\mathcal{C}(\mu, \Sigma)$ for Cauchy (the case $\nu = 1$). Marginalization preserves the family: $(z^{(1)}, \ldots, z^{(d)})$ has parameters $\mu_{1:d}$ and $\Sigma_{1:d,1:d}$ (and the same ν for Student's t). This property extends to mixtures and provides the marginal densities.

To compute conditional CDFs, define the notation:

$$\begin{bmatrix} \mu_{<d} \\ \mu_d \end{bmatrix} = \begin{bmatrix} \mu_{1:d-1} \\ \mu_d \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} \Sigma_{<d,<d} & \Sigma_{<d,d} \\ \Sigma_{d,<d} & \Sigma_{d,d} \end{bmatrix} = \begin{bmatrix} \Sigma_{1:d-1,1:d-1} & \Sigma_{(1:d-1,d)} \\ \Sigma_{d,1:d-1} & \Sigma_{d,d} \end{bmatrix}.$$

While elliptical conditionals are elliptical, their parameters require care (Kelker, 1970; Cambanis et al., 1981). Define:

$$\mu_{d|\cdot} = \mu_d + \Sigma_{d,<d}\Sigma^{-1}_{<d,<d}(z^{<d} - \mu_{<d}), \quad \text{and} \quad \zeta^2_{d|\cdot} = \Sigma_{d,d} - \Sigma_{d,<d}\Sigma^{-1}_{<d,<d}\Sigma_{<d,d}. \tag{17}$$

The conditional distributions for each family are:

Multivariate Gaussian. The conditional distribution is:

$$z^{(d)} \mid z^{<d} \sim \mathcal{N}\big(\mu_{d|\cdot}, \zeta^2_{d|\cdot}\big).$$

Multivariate Student's t. The conditional distribution is more complex; see Ding (2016) for an elementary derivation:

$$z^{(d)} \mid z^{<d} \sim \mathcal{ST}\Big(\mu_{d|\cdot}, \frac{\nu + \|z^{<d}\|^2_\Sigma}{\nu + d - 1}\zeta^2_{d|\cdot}, \nu + d - 1\Big),$$

where

$$\|z^{<d}\|^2_\Sigma = (z^{<d} - \mu_{<d})^{\mathrm{T}}\Sigma^{-1}_{<d,<d}(z^{<d} - \mu_{<d}). \tag{18}$$

Multivariate Cauchy. As a special case of Student's t with $\nu = 1$, the formula above applies with modified parameters. Note that the conditional is not necessarily Cauchy but rather Student's t with d degrees of freedom.

Remark 5 *An open problem is whether this approach extends to other consistent elliptical distributions, such as those of Bånkestad et al. (2020).*

4.5.5 Algorithmic considerations

Implicit reparameterization gradients have significant computational costs that are often overlooked. The estimator computes $\mathcal{S}$ and applies Proposition 3 to estimate ELBO gradients. To evaluate

$$\underbrace{-\nabla_\theta \mathcal{S}_\theta(z)}_{\text{(a)}} \overbrace{(\nabla_z \mathcal{S}_\theta(z))^{-1}}^{\text{(b)}},$$

we need the gradient of $\mathcal{S}$.

Term (a) can be computed by adding a surrogate term before calling $\mathcal{L}_\theta(\,\cdot\,)$. Since automatic differentiation systems typically return the Jacobian $J.(\,\cdot\,)$ (the transpose of the gradient), let X_θ solve $\overline{(J_z \mathcal{S}_\theta(z))}\, X_\theta = -\mathcal{S}_\theta(\bar{z})$, where $\overline{(\,\cdot\,)}$ denotes stopped gradients. Then $\bar{z} + X_\theta - \overline{X_\theta}$ evaluates to $\bar{z}$, but its gradient yields the required term:

$$\begin{aligned}\nabla_\theta \mathcal{L}_\theta(\bar{z} + X_\theta - \overline{X_\theta}) &= \nabla_\theta X_\theta \, \nabla_z \mathcal{L}_\theta(\bar{z}) + \nabla_\theta \mathcal{L}_\theta(\bar{z}) \\ &= -\nabla_\theta \mathcal{S}_\theta(z) \, (\nabla_z \mathcal{S}_\theta(z))^{-1} \, \nabla_z \mathcal{L}_\theta(\bar{z}) + \nabla_\theta \mathcal{L}_\theta(\bar{z}).\end{aligned}$$

To evaluate (b), we need the gradient of $\mathcal{S}$ with respect to z. Since $\mathcal{S}\colon \mathbb{R}^D \times \Theta \to [0,1]^D$, computing this gradient via automatic differentiation requires D passes through $\mathcal{S}$. Let $C_\mathcal{S}$ denote the cost of evaluating $\mathcal{S}$ once. Then the total cost is $\mathcal{O}(D \cdot C_\mathcal{S})$ for the gradient, plus $\mathcal{O}(D^2)$ to solve the lower-triangular system for the inverse. Efficient implementations of $\mathcal{S}$ are therefore essential.

Computing $\mathcal{S}$. The main computational challenge is evaluating $\mathcal{S}$ efficiently.

Algorithm 1 Compute $\mathcal{S}$ for a mixture of dense Gaussian components.

Require: Batched vectors $z \in \mathbb{R}^D$, $\mu \in \mathbb{R}^{K \times D}$ and $\pi = (\pi_1, \ldots, \pi_K)$ mixture weights, and a batch of lower triangular matrices $L \in \mathbb{R}^{K \times D \times D}$.
Ensure: Standardization function $\mathcal{S}$

```
W ← Weights(z, π)  [Eq. (16)]                                   ▷ O(KD²)
ϵ ← Batch solve Lϵ = z − μ                                      ▷ O(KD²)
Conditionals ← ConditionalsDist(ϵ, μ, L)  [Eq. (19)]            ▷ O(KD²)
WeightedCDFs ← Conditionals.CDF(z) × W                          ▷ O(KD)
Result ← Sum(WeightedCDFs, dim = components)                    ▷ O(KD)
return Result
```

For diagonal Gaussian covariances, Jankowiak and Karaletsos (2019) showed that $\mathcal{S}$ achieves the lower bound with $C_\mathcal{S} = \mathcal{O}(D)$ operations, making the overall complexity $\mathcal{O}(D^2)$. However, efficient algorithms for dense covariance settings remain an open problem.

We address this gap for mixtures of dense multivariate Gaussian components, with algorithms adaptable to Student's t and Cauchy. Computing the D sets of conditional parameters $\mu_{d|\cdot}$ and $\zeta^2_{d|\cdot}$ via (17) requires inverting $\Sigma_{<d,<d}$ at each dimension. Naively, this yields $C_\mathcal{S} = \mathcal{O}(D^4)$ and overall complexity $\mathcal{O}(D^5)$ for the full gradient.

We can do better by exploiting the Cholesky parameterization. In practice, Gaussian and Student's t covariances are parameterized as $\Sigma = LL^\mathrm{T}$, where L is lower triangular

with positive diagonal. Rather than inverting $\Sigma_{<d,<d}$ at each dimension, we solve one linear system $L\epsilon = z - \mu$ and reuse ϵ for all conditional parameters:

Proposition 6 *Let L be the Cholesky decomposition of Σ. Define $\mathring{L}$ as L with zero diagonal, and let $\epsilon \in \mathbb{R}^D$ solve $L\epsilon = z - \mu$. Then the conditional parameters in* (17) *are:*

$$\mu_{\cdot|\cdot} = \mu + \mathring{L}\epsilon, \qquad \text{and} \qquad \zeta^2_{\cdot|\cdot} = \mathbf{diag}\,(L)^2\,. \tag{19}$$

This computation vectorizes trivially across all K components.

Proof Write Σ in block form as

$$\Sigma = \begin{bmatrix} \Sigma_{<d,<d} & \Sigma_{<d,d} \\ \Sigma_{d,<d} & \Sigma_{d,d} \end{bmatrix} = \begin{bmatrix} L_{<d,<d} & \mathbf{0} \\ L_{d,<d} & L_{d,d} \end{bmatrix} \begin{bmatrix} L^{\mathrm{T}}_{<d,<d} & L^{\mathrm{T}}_{d,<d} \\ \mathbf{0} & L^{\mathrm{T}}_{d,d} \end{bmatrix}.$$

Both $\mu_{d|\cdot}$ and $\zeta^2_{d|\cdot}$ in (17) involve $\Sigma_{d,<d}\Sigma^{-1}_{<d,<d}$, which simplifies to:

$$\Sigma_{d,<d}\Sigma^{-1}_{<d,<d} = L_{d,<d}L^{\mathrm{T}}_{<d,<d}(L_{<d,<d}L^{\mathrm{T}}_{<d,<d})^{-1} = L_{d,<d}L^{-1}_{<d,<d}. \tag{$\star$}$$

From the block structure of $L\epsilon = z - \mu$:

$$\begin{bmatrix} L_{<d,<d} & \mathbf{0} \\ L_{\geq d,<d} & L_{\geq d,\geq d} \end{bmatrix} \begin{bmatrix} \epsilon^{<d} \\ \epsilon_{\geq d} \end{bmatrix} = \begin{bmatrix} z^{<d} - \mu_{<d} \\ z_{\geq d} - \mu_{\geq d} \end{bmatrix},$$

we have $\epsilon^{<d} = L^{-1}_{<d,<d}(z^{<d} - \mu_{<d})$. Substituting ($\star$) into (17) gives:

$$\mu_{d|\cdot} = \mu_d + \Sigma_{d,<d}\Sigma^{-1}_{<d,<d}(z^{<d} - \mu_{<d}) = \mu_d + L_{d,<d}\epsilon^{<d}.$$

The dth component of $\mu_{\cdot|\cdot}$ is:

$$\mu^{(d)}_{\cdot|\cdot} = \mu_d + (\mathring{L}\epsilon)^{(d)} = \mu_d + \mathring{L}_{d,\cdot}\epsilon = \mu_d + L_{d,<d}\epsilon^{<d} = \mu_{d|\cdot},$$

where the third equality uses that $\mathring{L}$ is lower triangular with zero diagonal.

For $\zeta^2_{d|\cdot}$, since $L_{d,d}$ is scalar:

$$\begin{aligned}\zeta^2_{d|\cdot} &= \Sigma_{d,d} - \Sigma_{d,<d}\Sigma^{-1}_{<d,<d}\Sigma_{<d,d} = L_{d,<d}L^{\mathrm{T}}_{d,<d} + L_{d,d}L^{\mathrm{T}}_{d,d} - L_{d,<d}L^{-1}_{<d,<d}L_{<d,<d}L^{\mathrm{T}}_{d,<d} \\ &= L_{d,d}L^{\mathrm{T}}_{d,d} = L^2_{d,d}.\end{aligned}$$

Vectorization across components is immediate since operations are independent. ■

With this formulation, solving the lower-triangular system $L\epsilon = z - \mu$ costs only $\mathcal{O}(D^2)$ operations. For Student's t, ϵ also provides $\|z^{<d}\|^2_\Sigma = \|\epsilon^{<d}\|^2_2$. Thus, the Cholesky-based approach achieves $C_{\mathcal{S}} = \mathcal{O}(D^2)$ and overall complexity $\mathcal{O}(D^3)$—a substantial improvement over the naive $\mathcal{O}(D^5)$.

5. Experiments

We compared the estimators across three experiments, adapting setups from Jankowiak and Karaletsos (2019) to highlight their differences.

i) Baseball model: a purely inferential task—an important statistical use case for variational inference—with evidence that mixture distributions better approximate the posterior than mean-field alternatives.

ii) Deep Markov model (DMM): because the stratified estimator cannot handle chain-structured models, we trained a deep Markov model for music generation and used mixture distributions to approximate each conditional distribution.

iii) MNIST-like datasets: variational autoencoders trained on MNIST, FashionMNIST, and KMNIST to test performance in higher-dimensional latent spaces, a setting that is challenging for the implicit reparameterization estimator.

For each experiment, we assess the posterior approximation quality (ELBO) and total optimization time.

We compare five estimators: the stratified estimator $\hat{g}^{(K)}$ (baseline when applicable), the implicit reparameterization estimator $\hat{g}^{\mathrm{IR}}$, the score-function estimator $\hat{g}^{\mathrm{SF}}$, the post-stratified estimator $\hat{g}^{\mathrm{PS}}$, and the J&K estimator $\hat{g}^{\mathrm{JK}}$ (applicable only to mixtures of diagonal Gaussians with `softmax` mixture weights). All but $\hat{g}^{(K)}$ are single-sample estimators; $\hat{g}^{\mathrm{SF}}$ is the only one not using reparameterization.

Setup. We tested three types of approximating distributions: mixtures of diagonal Gaussians, mixtures of Gaussians with dense covariance matrices, and mixtures of multivariate Student's t-distributions ($\nu = 4$ degrees of freedom) with dense covariance.

We distinguish between *particles* (the minimal estimation units) and *samples* (evaluations of $\mathcal{L}_\theta$ and its gradients). The stratified estimator requires K samples per particle (one per component), while single-sample estimators require only one sample per particle. This convention matches that used in Pyro. We varied both the number of mixture components and particles, averaging estimates across particles for the final gradient estimate.

Using Pyro (Bingham et al., 2019), we implemented the implicit reparameterization, stratified, and post-stratified estimators (not previously available in Pyro) and used Pyro's native J&K and score-function estimators. All models were optimized using Adam (Kingma and Ba, 2015).

We ran experiments across multiple random seeds: 10 for Baseball and DMM, and 6 for MNIST-like datasets. Each configuration (particles, distribution, estimator, seed) was run at multiple learning rates (see the appendix). For per-iteration comparisons, we use the ELBO envelope across learning rates: at each iteration, we compute the mean ELBO across seeds for each learning rate, then report the maximum across learning rates. For aggregate comparisons, we selected the learning rate reaching the highest median ELBO across seeds and compared estimators using the mean ELBO at that rate.

We used the `softmax` function for mixture weights: $w_k = \exp(\ell_k)/\sum_{j=1}^{K}\exp(\ell_j)$. For diagonal Gaussians, $z_k = \boldsymbol{\mu} + \exp(\ln \boldsymbol{\sigma}) \odot \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$. For dense Gaussians, $z_k = \boldsymbol{\mu} + L\epsilon$, where L is the Cholesky factor with positive diagonal elements via `softplus`. For Student's t with ν degrees of freedom, if $\tilde{z} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and $u \sim \chi^2_\nu$, then $z = \boldsymbol{\mu} + \tilde{z}\sqrt{\nu/u}$ is

Student's t with mean $\boldsymbol{\mu}$ and covariance $\frac{\nu}{\nu-2}\Sigma$ (for $\nu \leq 2$ the same parameterization holds, but Σ is not a covariance matrix).

5.1 General Findings

We first summarize findings common across all experiments, then analyze each setting.

First, the implicit reparameterization estimator is consistently the slowest single-sample method and can even be slower than the stratified estimator. Second, the stratified estimator is the most robust when feasible; on Baseball and MNIST-like datasets it delivered solid ELBOs with predictable run times. Third, no single-sample estimator consistently achieves the best ELBO—there is no one-size-fits-all. The J&K and post-stratified estimators complement each other: each outperforms the other in different settings. Fourth, numerical instabilities affected the implicit reparameterization and J&K estimators. Finally, using STL substantially affects performance in some settings but not others; see Section 5.5.

We conjecture that ELBO performance depends on the optimal weight distribution. When weights spread across many components, implicit reparameterization and J&K tend to outperform post-stratified; when weights concentrate on few components, post-stratified tends to win. This pattern aligns with the stratified estimator's inefficiency: concentrating most probability mass on few components wastes samples on low-contribution components.

The estimators also differ in applicability. Post-stratified applies more broadly than implicit reparameterization, which in turn applies more broadly than J&K (which requires `softmax` parameterization).

5.2 Baseball

To evaluate the estimators in a purely inferential setting—approximating a fixed posterior $p(z \mid x)$—we used the Baseball dataset. This hierarchical model estimates batting abilities for 18 players from their batting averages (Efron and Morris, 1975). For player i, the number of hits in 45 at-bats follows a binomial distribution with success probability ϕ_i. Each ϕ_i is drawn from $\text{Beta}(\alpha, \beta)$ with shared parameters satisfying $\alpha = \kappa\xi$ and $\beta = \kappa(1-\xi)$, where $\kappa \sim \text{Pareto}(1, 1.5)$ and $\xi \sim \text{Beta}(1, 1)$. (See Gelman et al. 2013 and the Stan documentation for details.) We approximate the posterior over latent variables κ, ξ, and ϕ_i using a mixture distribution. We used Pyro's implementation, matching Jankowiak and Karaletsos (2019). Because the model is small, we used CPU, where additional samples—via more particles or the stratified estimator—are more expensive than in GPU-accelerated experiments.

Figure 1 shows that the stratified estimator reaches higher ELBOs in fewer iterations. The post-stratified and J&K estimators are competitive, especially with more particles. With one particle, implicit reparameterization slightly outperforms score-function, but the gap narrows with more particles. This small advantage comes at significant runtime cost: implicit reparameterization is the slowest single-sample estimator (Figure 2). The stratified estimator also slows as components increase due to more objective evaluations per iteration. For some mixture families, implicit reparameterization is not only the slowest single-sample method but slower than the stratified estimator—suggesting that the cubic complexity from Section 4.5.5 can be prohibitive even at $D = 20$. By contrast, post-stratified and J&K runtimes remain stable across components and mixture families.

The stratified estimator's faster convergence partly reflects drawing more samples, but also benefits from sampling once per component, which suits *this* posterior well. To test this empirically, we compared the stratified estimator using one particle per component (K samples total) to single-sample estimators with one and with K particles. Even matching the total number of samples, the stratified estimator achieves higher ELBOs, confirming the value of per-component sampling. However, for post-stratified and J&K, this gap matters only briefly during optimization (Figure 5, appendix).

With single-sample estimators, the ELBO peaks at $K = 21$ components, indicating that the increased expressive power is beneficial. The mean ELBO with diagonal Gaussian ($K = 1$) is -55.54, versus -54.70 (J&K) and -54.86 (post-stratified) at $K = 21$. A t-test comparing $K = 1$ and $K = 21$ gives $p < 10^{-8}$ for both. Across all configurations, the STL estimator achieves the highest ELBO, which is discussed later in this section.

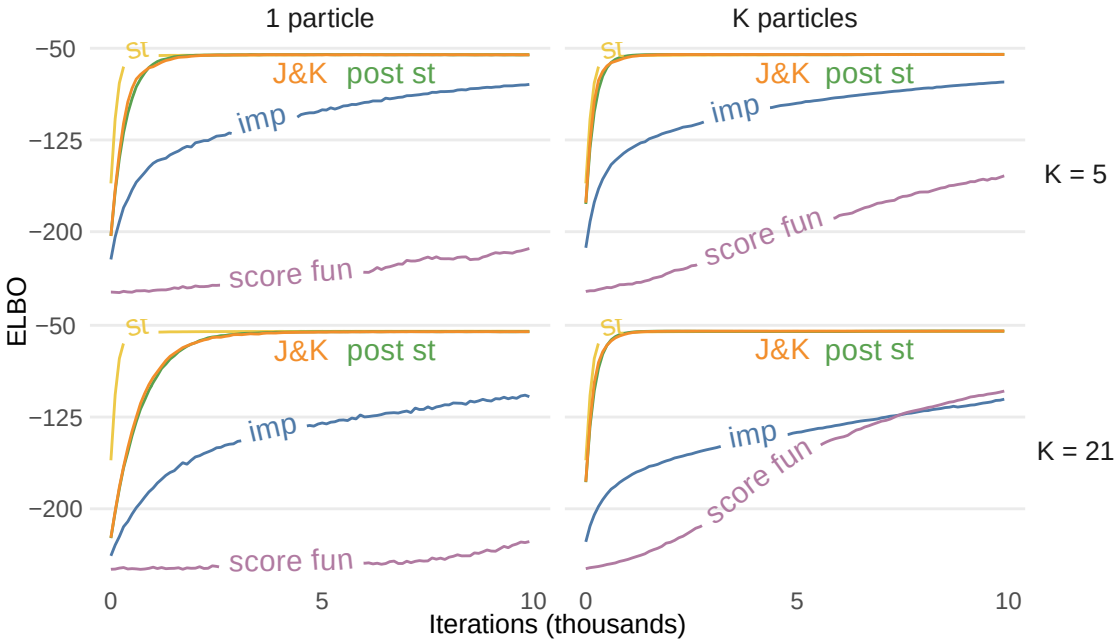

Figure 1: **Test ELBO** vs. iteration on Baseball for mixtures of diagonal Gaussians. Per-iteration envelope of mean test ELBO across learning rates. Components $K \in \{5, 21\}$; particles $n \in \{1, K\}$. The n=K setting matches the K samples used by stratified [cf. Figure 5]. Stratified converges fastest; post-stratified and J&K are competitive, especially with more particles. Notably, the score function estimator benefits more from increasing n than implicit reparameterization.

5.3 Deep Markov Model

Although efficient on the Baseball dataset, the stratified estimator has a severe weakness for chains of mixtures: the number of objective evaluations grows exponentially with the number of mixtures in the chain (Section 3). To illustrate, we use a deep Markov model for polyphonic music following Krishnan et al. (2017) and the JSB dataset of 4-voice chorales in the style of J.S. Bach (Boulanger-Lewandowski et al., 2012), creating a setting in which

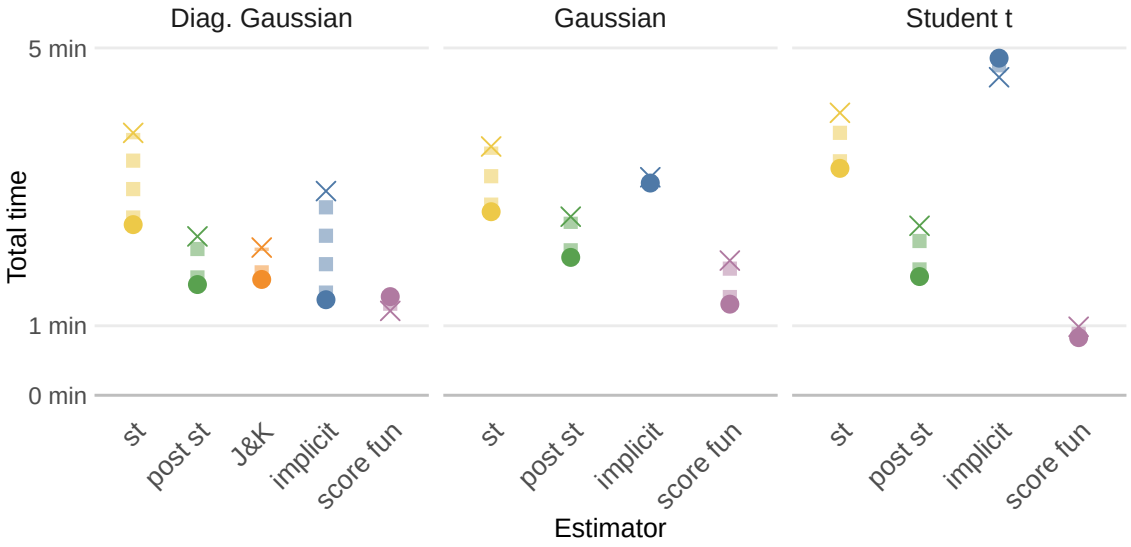

Figure 2: **Running time** for optimizing Baseball with different approximating distributions and estimators (1 particle, 10,000 iterations). Line endpoints denote the median across seeds and learning rates; markers: ● for $K = 5$, × for $K = 21$. The implicit reparameterization estimator is substantially slower than others, especially with full-covariance components (Gaussian, multivariate Student's t). The stratified estimator slows as K grows (it draws K samples per step). In contrast, post-stratified and J&K runtimes are relatively stable across K and families.

the stratified estimator cannot be used. At time step t, the probability of a note is governed by the latent variable z_t ($D = 32$), which depends only on z_{t-1} from the previous step. We approximate the posterior $p_\phi(z_t \mid x_t, z_{t-1})$ with a mixture $q_\theta(z_t \mid x_t, z_{t-1})$ of diagonal Gaussian distributions, with $K \in \{2, 3, 5\}$ components. The dataset contains 229 sequences with an average length of 60 time steps. This setup produces a chain of mixtures that makes the stratified estimator infeasible: in some cases more than K^{60} samples would be required to compute it. Accordingly, we compare only single-sample estimators: the implicit reparameterization estimator, the J&K estimator, the post-stratified estimator, and the score-function estimator as the baseline.

We adopt the experimental setup and implementation of Jankowiak and Karaletsos (2019) and refer readers to that work for model and training details. To simplify comparisons, we omit KL-annealing.[2] This choice affected the quality of the learned models but allowed us to focus on the estimators' relative performance. See the appendix for details.

Several results emerge from this experiment (Table 2): (i) The three estimators using some form of reparameterization achieve a higher test ELBO than the score-function estimator, though none improves on a single-Gaussian baseline ($K = 1$: ELBO -7.61, STL effect $+0.16$). (ii) The implicit reparameterization and J&K estimators perform similarly and substantially outperform the post-stratified estimator. (iii) The best-performing estimators benefit from using STL, yielding statistically significant improvements in ELBO; the

2. KL-annealing gradually increases the weight of the KL term (between the approximation and the prior) in the ELBO; see Pyro's deep Markov model documentation.

post-stratified estimator does not. A different pattern appears for running time. Relative to the score-function, (iv) the J&K estimator takes 20 % longer, the post-stratified estimator 25 % longer, and the implicit reparameterization estimator 64 % to 68 % longer. This large overhead for implicit reparameterization is consistent with the discussion in Section 4.5.5.

Finally, recall that the post-stratified and J&K estimators differ only in how they estimate the gradient with respect to the mixture weights—the post-stratified uses a score-function term. This difference, together with points (ii) and (iii), suggests that the variance of the gradient for the mixture weights greatly impacts performance in cases involving a chain of mixtures.

Table 2: **Test ELBO** (higher is better) for the deep Markov model with diagonal-Gaussian mixtures, and the **STL effect** (ΔELBO from applying STL). Blue indicates significance (t-test, $\alpha = 0.05$).

K	Implicit rep.		J&K		Post-stratified		Score function
		STL effect		STL effect		STL effect	
2	$-7.75^{\dagger}$	0.22	-7.72	0.16	-10.64	0.12	-11.26
3	-7.81	0.25	-7.75	0.13	-11.09	0.01	-11.25
5	-7.83	0.15	-7.85	0.22	-11.20	0.12	-11.24

† For reference, using $K = 1$ leads to an ELBO of -7.61 and an STL effect of 0.16.

5.4 MNIST, FashionMNIST, and KMNIST

Section 4.5.5 shows that the implicit reparameterization estimator scales polynomially with the latent dimension D. To examine this empirically, we use a higher-dimensional latent space ($D = 50$) than in earlier experiments and train a variational autoencoder (VAE) on MNIST, FashionMNIST, and KMNIST. In all cases, we set the number of components K to 3, 4, and 5, and use either 1 or 5 particles for estimation.

Figure 3 reports the mean ELBO across seeds after 500 training epochs. We focus on mixtures of diagonal Gaussian distributions and reparameterizable estimators. For readability, we plot ELBO differences relative to the stratified estimator; a positive value indicates that an estimator performs better than stratified for that configuration.

Several patterns are apparent in the results: (i) Across all datasets and configurations, the post-stratified estimator matches the stratified estimator and in some cases surpasses it, confirming that post-stratified is an excellent alternative in this setting. (ii) The estimator of J&K diverges on the KMNIST dataset in *all* configurations. (iii) The implicit reparameterization estimator appears more robust than the estimator of J&K, almost always achieving a higher ELBO, though it still falls short of the post-stratified estimators. (iv) The implicit reparameterization estimator is considerably slower than the other single-sample estimators (Table 3). We discuss this slowdown in detail later.

The observation in point (i) challenges the assumption that the stratified estimator is always best. At $K = 5$, we equalize computational cost by comparing the stratified estimator with a single particle to the post-stratified estimator with 5 particles. Using the

stratified estimator, we obtain mean ELBOs of -235.40 for FashionMNIST, -186.28 for KMNIST, and -93.85 for MNIST; the post-stratified estimator improves these by 0.82, 1.07, and 0.86, respectively. The positive differences are statistically significant with $\alpha = 0.05$. We conjecture that in this setting the optimal mixture has a single component, and that the post-stratified estimator is more efficient because it avoids sampling from components that do not contribute to the objective. To support this claim, we compared the final ELBO achieved by the stratified estimator using a single particle across $K \in \{1, 3, 4, 5\}$ components and found nearly identical values, consistent with a single-component optimum. This pattern persists for the STL version of the estimator, although the specific ELBOs differ.

Many particles and diverging estimators. In addition to the experiments in Figure 3, we also optimized with 100 particles. Under this setting, the only single-sample estimators that converged consistently were the post-stratified estimator and the score-function estimator; the implicit reparameterization and the J&K estimators almost always failed to converge. This outcome is counterintuitive because increasing the number of particles reduces the estimator's variance. Inspecting the runs, we traced the failure to numerical instability in Pyro's implementation, not necessarily to the J&K estimator itself; more particles increased the chance of triggering this failure mode.[3] The implicit reparameterization estimator likewise failed to converge in most cases; the cause is less clear, but the method requires solving several linear systems, which can be numerically unstable, especially with many particles.

Running time. Measuring optimization time is difficult because it depends on the hardware and on transient factors such as machine load and temperature. We therefore measured, for each combination of estimator, number of particles, and number of components, the time for one epoch with one seed and one learning rate on the three MNIST-like datasets. Table 3 shows that the estimators of J&K and the post-stratified estimator take only slightly longer than the score-function estimator yet yield a much better-quality estimation, whereas the implicit reparameterization is considerably slower. These patterns hold across numbers of particles and components and, for the post-stratified estimator, also across mixture families. By contrast, the implicit reparameterization estimator degrades rapidly, requiring more than three times the time of the score-function estimator with five particles and a mixture of Student's t distributions with dense covariance matrices. It also increases memory pressure: with many particles and dense-covariance mixtures, we ran out of memory when using the implicit reparameterization estimator.

Additional families of mixtures. In this section we primarily evaluate mixtures of diagonal Gaussian distributions. We also test mixtures of Gaussian distributions with dense covariance matrices and mixtures of Student's t distributions with dense covariance matrices; Table 3 reports their running times. For these families, the implicit reparameterization estimator failed to complete optimization. Among single-sample estimators, only the post-stratified estimator usually completed the optimizations (the J&K estimator does not apply), yet its ELBO never exceeded that of the mixture of diagonal Gaussian distributions. By contrast, the stratified estimator sometimes achieved a higher ELBO than it did for the

3. The issue is a `float32` underflow in the backward pass; after submission, we proposed a log-space fix upstream.

diagonal-Gaussian mixtures. Overall, these experiments offer little evidence of benefits from using the additional families of mixtures. A likely cause of the weak single-sample performance is numerical instability introduced by dense covariance matrices, particularly when solving the linear systems required to compute log-densities. These instabilities represent one of the most important open problems identified in this work, which we discuss later.

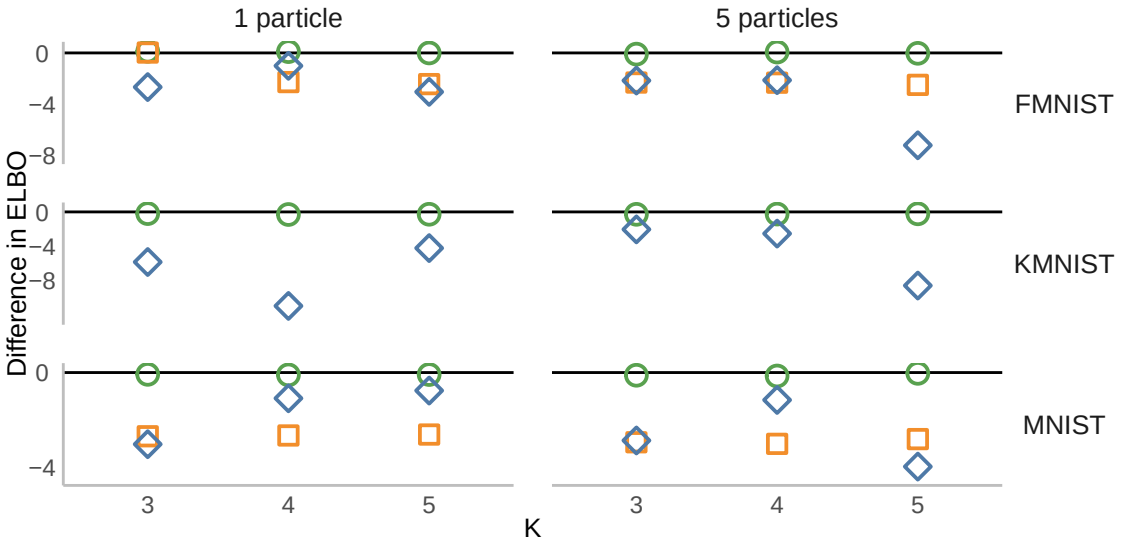

Figure 3: **Differences in mean ELBO** relative to the stratified estimator (negative favors stratified). Rows: FashionMNIST, KMNIST, MNIST. Columns: particle counts. The post-stratified estimator matches or sometimes exceeds the results of the stratified. Implicit reparameterization consistently yields higher ELBOs than J&K; on KMNIST, J&K diverges while implicit remains stable yet below post-stratified. The score-function estimator is omitted due to very large gaps.

5.5 Effect of the Sticking-the-Landing Modification

Across all experiments, we evaluated each estimator both in its standard form and with the Sticking-the-Landing (STL) modification. Prior work has analyzed STL for mixture variational distributions; in particular, Roeder et al. (2017) used the stratified estimator to illustrate its implementation. Here we extend the analysis to additional estimators and settings. The results are *puzzling*: there is no clear pattern predicting when STL helps or hurts, nor its magnitude. For example, on the Baseball dataset, STL can improve approximation quality, though not consistently. Figure 4 shows that, even in the same setting (same number of components and a single particle), STL increases the ELBO with the implicit reparameterization estimator yet has a negative effect for the stratified, post-stratified, and J&K estimators. Using the same J&K estimator, the effect is markedly positive in the deep Markov model setting. For the MNIST-like datasets, the results were more homogeneous but still inconclusive: with a mixture of diagonal Gaussian distributions and a single particle, STL yielded a statistically significant improvement in 24 of 32 configurations that completed optimization (75 %). Notably, the effect size is usually large. These findings are intriguing because, once a setting is fixed, both the posterior and the approximating distri-

Table 3: **Running time per epoch** of the score-function estimator (last column, seconds) and **percent increase** over that baseline for other estimators on the MNIST-like datasets; cross-dataset differences are negligible.

	K	Implicit rep.	J&K	Post-stratified	Stratified	Score-function
		%	%	%	%	seconds
1 particle						
Diagonal Gaussian	3	16.30 %	6.15 %	8.09 %	16.78 %	7.00 s
	4	13.62 %	4.22 %	4.89 %	13.02 %	7.03 s
	5	15.18 %	5.82 %	5.88 %	12.76 %	6.98 s
Gaussian	3	25.52 %	—	5.61 %	11.60 %	7.28 s
	4	29.02 %	—	5.34 %	13.75 %	7.18 s
	5	34.77 %	—	4.16 %	12.04 %	7.31 s
Student's t	3	53.00 %	—	6.84 %	14.94 %	7.26 s
	4	61.46 %	—	7.95 %	17.31 %	7.15 s
	5	65.42 %	—	5.50 %	16.43 %	7.34 s
5 particles						
Diagonal Gaussian	3	30.38 %	4.85 %	5.95 %	20.19 %	7.13 s
	4	31.81 %	4.65 %	5.30 %	19.31 %	7.13 s
	5	38.32 %	5.11 %	5.17 %	20.52 %	7.14 s
Gaussian	3	82.89 %	—	6.84 %	15.74 %	7.31 s
	4	108.04 %	—	8.89 %	18.99 %	7.26 s
	5	129.87 %	—	7.79 %	17.06 %	7.44 s
Student's t	3	151.19 %	—	7.07 %	14.97 %	7.50 s
	4	192.70 %	—	9.84 %	15.55 %	7.57 s
	5	230.09 %	—	9.41 %	16.95 %	7.73 s

bution family remain unchanged, and only the gradient estimator varies. Hence, differences in STL's impact are solely due to the choice of estimator. This is especially striking given that STL's theoretical motivation, i.e., zero variance when q_θ matches the true posterior, is itself an estimator-agnostic property: every STL-modified estimator inherits it, yet the practical consequences vary widely. Given that mixture distributions as variational distributions typically do not yield unique solutions (except in uninteresting cases), the estimator can substantially steer optimization. Our configurations depart from those recently used to study STL (Kim et al., 2024). Since STL is straightforward to implement, we recommend evaluating both versions of each estimator during optimization.

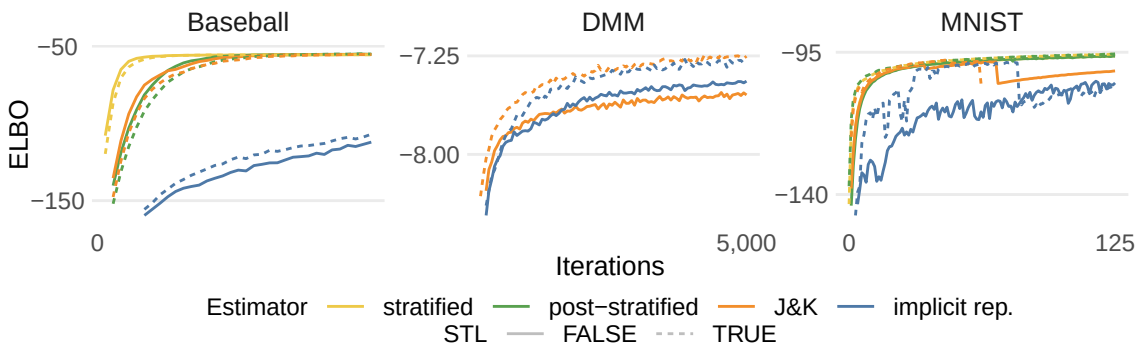

Figure 4: **Test ELBO** over iterations ($K = 5$ components, single particle), with/without STL: (left) Baseball, (center) deep Markov model, (right) MNIST. On Baseball, STL helps implicit reparameterization but hinders stratified, post-stratified, and J&K. On the deep Markov model, STL benefits J&K. On MNIST, STL improves stratified and post-stratified with mixed effects on J&K and implicit reparameterization. Score-function not shown (STL not applicable).

6. Related and Future Work

The variational inference community has long been interested in using mixture distributions for more expressive approximating families. For a review of early work through 2018, see Section 5.4 of Zhang et al. (2019).

Many different approaches have been proposed for incorporating mixture distributions into VI, distinct from those we present here. This diversity of methods reflects the problem's inherent complexity. We now briefly discuss other approaches found in the literature. One of the clearest examples of these initial works is Variational Boosting, independently proposed by Guo et al. (2017) and Miller et al. (2017). In this approach, the approximation is refined by adding new components to the mixture in a greedy way, while keeping the already learned components fixed.

Another more recent example, which is closer to our work, comes from Morningstar et al. (2021). They introduced a new objective using ideas from importance sampling and IWAEs. In addition to suggesting the moniker "stratified" for the stratified estimator, they presented the SIWAE objective, which requires at least one sample from each component—hence, the 'S' stands for stratified. Concurrently, Kviman et al. (2022) introduced the MISELBO objective, which is closely related to the SIWAE objective and was originally developed for mixtures with fixed and uniform mixture weights. Recent developments related to this objective have been significant, with empirical studies such as Kviman et al. (2023) showing the benefits of mixtures for VAEs. Additionally, Hotti et al. (2024) have scaled the MISELBO to hundreds of components in the fixed mixture weights setting by eliminating the need to use one sample from each component. One of the methods they proposed involves randomly selecting a subset of components to sample from, similar to the randomized component estimator (Estimator 4) we presented in Section 4.2.

Our work differs in two key ways: (a) we study gradient estimators when mixture weights are learned, a key technical challenge, and (b) we focus on general objectives, including the ELBO.

Future work. There are several promising directions to extend the research presented in this work. One such direction, which was beyond the scope of this study, involves exploring the application of the post-stratified estimator in settings with hundreds of components but fixed mixture weights, such as those explored by Hotti et al. (2024). Given that we have proved the post-stratified estimator's ability to reduce estimation variance compared to the randomized component estimator, it seems plausible to extend these results to similar settings.

A second direction, related to the stratified estimator, is to intensify the parallelism with surveying and explore different allocation strategies for the strata. It would be interesting to consider the use of Neyman's allocation [see, e.g., Lohr, 2021] in the stratified estimator.

During this work, we encountered numerical stability issues with distributions using dense location-scale matrices—a problem beyond mixture distributions. As mentioned in the experiments section, we represent the covariance matrix Σ using Cholesky decomposition: $LL^{\mathrm{T}} = \Sigma$. Computing the log-density for mean μ and sample z requires solving the linear system $LX = z - \mu$. Since L is only constrained to be lower triangular with positive diagonal elements, it can become ill-conditioned and cause numerical instability. Addressing how to parameterize the covariance matrix to ensure both numerical stability and efficient gradient computation remains an interesting open problem. Unlike simple multivariate Gaussian or Student's t distributions, where the entropy term $\mathbb{E}_{q_\theta}[\ln q_\theta]$ can be computed analytically or by adjusting the density of a base distribution, mixture distributions require computing the density for each component, potentially triggering instability. Interestingly, using the STL modification to estimate gradients with simple Gaussian or Student's t approximations can also trigger this issue, as solving the linear system is required.

7. Conclusion

This work presents a comprehensive study of reparameterization-based gradient estimators for mixture models in variational inference. The stratified estimator proves highly robust when feasible—that is, when the number of components is small enough to sample from each. For single-sample estimators, no single method dominates; the best choice depends on the problem, typically favoring either the post-stratified estimator or the J&K estimator (which requires diagonal Gaussian components with softmax-parameterized weights). A key insight across all estimators is the importance of accounting for component probabilities: either by weighting component gradients with mixture weights $\pi_{k,\theta}$ (stratified) or with posterior probabilities $q_\theta(k \mid z)$ of the realized sample (post-stratified and J&K).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. 1749854 and 1908577.

Appendix A. Additional Experimental Details

A.1 Baseball dataset

For Baseball, we use mixtures with $K \in \{5, 21\}$ components and run 10,000 iterations with $n \in \{1, K\}$ particles, sweeping Adam learning rates 5×10^{-3}, 10^{-3}, and 10^{-4}.

Architecture, batching, and exact loss. Following Efron and Morris (1975), we use Pyro's tutorial implementation of the partially-pooled Beta–Binomial. The priors are $m \sim \text{Uniform}(0, 1)$ and $\kappa \sim \text{Pareto}(1, 1.5)$, and a plate over 18 players carries $\phi_i \sim \text{Beta}(m\kappa, (1-m)\kappa)$ with observations $\text{Binomial}(45, \phi_i)$. Variational family: a mixture ($K \in \{5, 21\}$) over $(m, \kappa, \phi_1, \ldots, \phi_{18}) \in \mathbb{R}^{20}$ in unconstrained coordinates, with the standard bijections mapping back to $\phi_i \in (0, 1)$ and $\kappa > 1$. No mini-batching: each SVI step uses all 18 observations. An "epoch" in the training loop is a block of 100 SVI steps, used only as the unit for periodic evaluation. Loss computed by Pyro's `Trace_ELBO` for the score-function, J&K, post-stratified, and implicit reparameterization estimators, and by `TraceEnum_ELBO` for the stratified estimator (which enumerates the component index).

Initialization, stopping, and hardware. Autoguide initialization (custom `AutoMix` subclass of Pyro's `AutoContinuous`): component means at the per-site prior median (Pyro's `init_to_median`) plus a $0.1\mathcal{N}(0, I)$ perturbation per component, component scales at 0.1 under a softplus-positive constraint, component logits drawn from Uniform$[0, 1)$. Fixed budget of 10,000 SVI steps, no early stopping, 10 seeds. ELBO evaluated every 100 steps as the mean of 100 fresh single-sample estimates on the full dataset. Reported run: best final mean ELBO across the three swept Adam learning rates. All runs on CPU.

A.2 Deep Markov Model

Architecture. Model and inference network from Krishnan et al. (2017), packaged in Pyro's deep Markov model example, with the single diagonal-Gaussian guide replaced by a mixture of $K \in \{2, 3, 5\}$ diagonal Gaussians. Latent $z_t \in \mathbb{R}^{32}$, observation $x_t \in \{0, 1\}^{88}$. Emitter for $p(x_t \mid z_t)$: two-hidden-layer 100-unit ReLU MLP. Gated transition for $p(z_t \mid z_{t-1})$: 200-unit ReLU MLP with a sigmoid gate over a non-linear "proposed mean" and an identity-initialized linear path, softplus on the diagonal scale. Guide combiner over a unidirectional ReLU RNN (hidden size 600, run on the time-reversed sequence): tanh projection of z_{t-1} averaged with the RNN hidden state, then three linear outputs (sizes $32 \cdot K$ means, $32 \cdot K$ softplus scales, K component logits).

Batching and exact loss. Mini-batches of 20 sequences (Pyro default), padded and masked to the longest sequence so padded-position contributions are zero. Loss: ELBO via Pyro's `Trace_ELBO`, summed over time and observations and averaged across the mini-batch, with both prior and posterior log-densities at z_t evaluated on the sampled trace (no analytic KL). KL-annealing disabled.

Score-function estimator. Pyro's `Trace_ELBO` Rao–Blackwellizes the score-function gradient across the data plate (each of the 20 sequences in a mini-batch is an independent score-function term), but applies no further Rao–Blackwellization within a sequence.

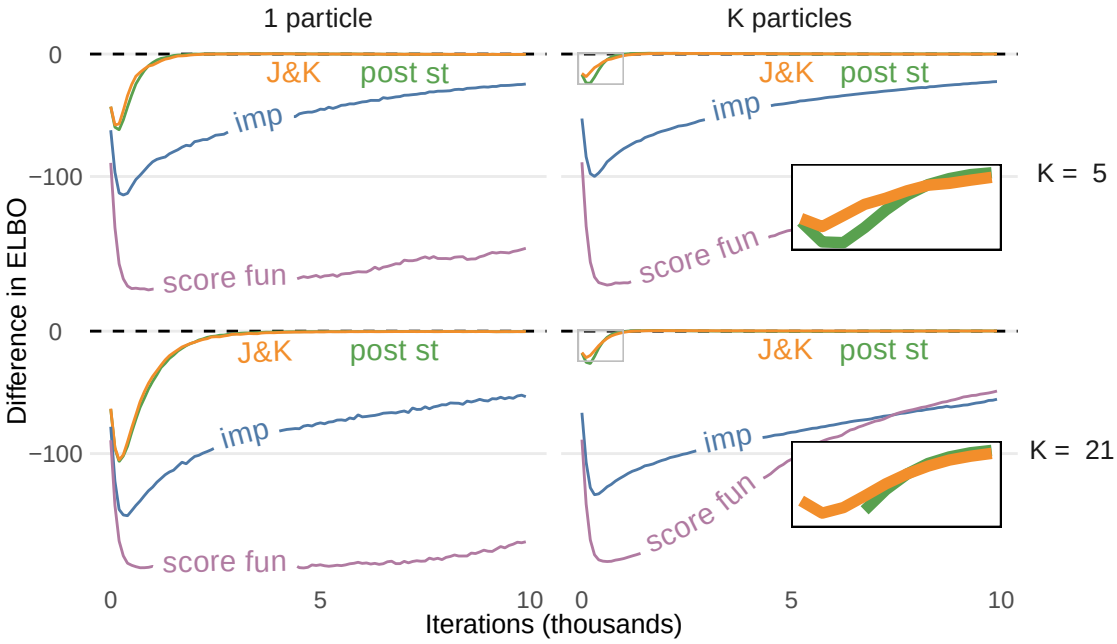

Figure 5: **ELBO differences** vs. stratified on Baseball over iterations, for diagonal-Gaussian mixtures with $K \in \{5, 21\}$. At each iteration we plot the envelope of the test ELBO across learning rates (negative favors stratified). Stratified uses one particle per iteration; other estimators use $n \in \{1, K\}$, with $n = K$ matching stratified's total samples. Insets highlight regions where post-stratified and J&K underperform most.

Parameter initialization. PyTorch defaults (Kaiming-uniform weights and biases) with two exceptions from the reference implementation: in the gated transition, the linear "identity" path on z_{t-1} starts at the identity matrix with zero bias; the trainable initial states z_0, z_0^q, and RNN initial hidden state h_0 start at zero. The combiner's K component-logit head uses the default initialization.

Optimizer and stopping criteria. Pyro's `ClippedAdam` ($\beta_1 = 0.96$, $\beta_2 = 0.999$, gradient norm clipped to 10, weight decay 2.0, learning-rate decay 0.99996 per step). Adam learning rate swept over $\{5 \times 10^{-4}, 3 \times 10^{-4}, 10^{-4}, 3 \times 10^{-5}, 10^{-5}\}$, 10 seeds. Fixed budget of 1000 epochs (no early stopping); validation/test ELBO every 50 epochs on the JSB-provided splits. Reported number: mean ELBO across seeds at the learning rate with the highest median validation ELBO. Running-time numbers use the same setup but 100 iterations at a single learning rate on identical GPUs. The STL figure (Figure 4) extends the budget to 5000 epochs.

A.3 MNIST, FashionMNIST, and KMNIST datasets

For MNIST, FashionMNIST, and KMNIST, we use the architecture from Pyro's variational autoencoder example. We sweep $K \in \{3, 4, 5\}$ components and Adam learning rates 10^{-3}, 5×10^{-4}, 10^{-4}, using 6 random seeds.

Architecture, batching, and exact loss. Latent dimension $D = 50$, hidden size 400, image size 784. Decoder: $z \mapsto \text{Bernoulli}(\sigma(W_2\,\text{softplus}(W_1 z + b_1) + b_2))$. Encoder: a $784 \to 400$ softplus first layer followed by separate linear outputs for the mixture parameters. For diagonal-Gaussian mixtures: two $400 \to 50 \cdot K$ outputs (means and exponentiated scales) and one $400 \to K$ output (component logits). For dense-covariance Gaussian and Student's t mixtures: the scale output is $400 \to 50^2 \cdot K$, reshaped into K lower-triangular factors with softplus on the diagonal (SoftplusLowerCholesky-style). Mini-batches of 128 images, shuffled each epoch, dynamically binarized by a single Bernoulli(x) draw with $x \in [0,1]^{784}$. Loss: per-image negative ELBO under $\mathcal{N}(0, I_{50})$ prior and Bernoulli observation. Test ELBO is the dataset average over the test split, computed once per epoch.

Initialization and stopping criteria. PyTorch defaults (Kaiming-uniform weights and biases); no pre-training or warm-starting from the reference VAE example's weights. Fixed budget of 500 epochs, no early stopping or learning-rate scheduling.

References

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

Maria Bånkestad, Jens Sjölund, Jalil Taghia, and Thomas Schön. The elliptical processes: a family of fat-tailed stochastic processes. *arXiv preprint arXiv:2003.07201*, 2020.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul A. Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20:28:1–28:6, 2019. URL `http://jmlr.org/papers/v20/18-403.html`.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1881–1888, 2012.

F. Jay Breidt and Jean D. Opsomer. Endogenous post-stratification in surveys: Classifying with a sample-fitted model. *The Annals of Statistics*, 36(1):403 – 427, 2008. doi:10.1214/009053607000000703.

Stamatis Cambanis, Steel Huang, and Gordon Simons. On the theory of elliptically contoured distributions. *Journal of Multivariate Analysis*, 11(3):368–385, 1981.

Peng Ding. On the conditional distribution of the multivariate t distribution. *The American Statistician*, 70(3):293–295, 2016.

Justin Domke and Daniel R Sheldon. Importance weighting and variational inference. *Advances in neural information processing systems*, 31, 2018.

Bradley Efron and Carl Morris. Data analysis using Stein's estimator and its generalizations. *Journal of the American Statistical Association*, 70(350):311–319, 1975.

Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. *Advances in neural information processing systems*, 31, 2018.

Michael C. Fu. Gradient estimation. *Handbooks in operations research and management science*, 13:575–616, 2006.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, New York, 3 edition, 2013. ISBN 9780429113079. doi:10.1201/b16018. URL https://doi.org/10.1201/b16018.

Alex Graves. Stochastic backpropagation through mixture density distributions. *arXiv preprint arXiv:1607.05690*, 2016.

Fangjian Guo, Xiangyu Wang, Kai Fan, Tamara Broderick, and David B. Dunson. Boosting variational inference, 2017.

Matthew Hoffman and David Blei. Stochastic structured variational inference. In *Artificial Intelligence and Statistics*, pages 361–369. PMLR, 2015.

Alexandra Hotti, Oskar Kviman, Ricky Molén, Víctor Elvira, and Jens Lagergren. Efficient mixture learning in black-box variational inference. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=Grrydzui3A.

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.

Martin Jankowiak and Theofanis Karaletsos. Pathwise derivatives for multivariate distributions. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 333–342. PMLR, 2019.

Martin Jankowiak and Fritz Obermeyer. Pathwise derivatives beyond the reparameterization trick. In *International conference on machine learning*, pages 2235–2244. PMLR, 2018.

Yutaka Kano. Consistency property of elliptic probability density functions. *Journal of Multivariate Analysis*, 51(1):139–147, 1994.

Douglas Kelker. Distribution theory of spherical distributions and a location-scale parameter generalization. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 419–430, 1970.

Kyurae Kim, Yian Ma, and Jacob Gardner. Linear convergence of black-box variational inference: Should we Stick the Landing? In *International Conference on Artificial Intelligence and Statistics*, pages 235–243. PMLR, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Rahul Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Oskar Kviman, Harald Melin, Hazal Koptagel, Victor Elvira, and Jens Lagergren. Multiple importance sampling elbo and deep ensembles of variational approximations. In *International Conference on Artificial Intelligence and Statistics*, pages 10687–10702. PMLR, 2022.

Oskar Kviman, Ricky Molén, Alexandra Hotti, Semih Kurt, Vıctor Elvira, and Jens Lagergren. Cooperation in the latent space: The benefits of adding mixture components in variational autoencoders. In *International Conference on Machine Learning*, pages 18008–18022. PMLR, 2023.

Jinlin Lai, Justin Domke, and Daniel R. Sheldon. Variational marginal particle filters. In *International Conference on Artificial Intelligence and Statistics*, pages 875–895. PMLR, 2022.

Sharon L. Lohr. *Sampling: design and analysis*. Chapman and Hall/CRC, 2021.

Andrew C Miller, Nicholas J. Foti, and Ryan P. Adams. Variational boosting: Iteratively refining posterior approximations. In *International Conference on Machine Learning*, pages 2420–2429. PMLR, 2017.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

Warren Morningstar, Sharad Vikram, Cusuh Ham, Andrew Gallagher, and Joshua Dillon. Automatic differentiation variational inference with mixtures. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. PMLR, 2021.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

Geoffrey Roeder, Yuhuai Wu, and David K. Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Tim Salimans and David A. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837 – 882, 2013. doi:10.1214/13-BA858.

Stan Development Team. *Stan Modeling Language Users Guide and Reference Manual*, 2017. URL `http://mc-stan.org/users/documentation/case-studies/pool-binary-trials.html`.

Aäron van den Oord and Benjamin Schrauwen. The student-t mixture as a natural image patch prior with application to image compression. *J. Mach. Learn. Res.*, 15(1):2061–2086, 2014.

Cédric Villani. *Topics in optimal transportation*, volume 58. American Mathematical Soc., 2021.

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

Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (8):2008–2026, 2019. doi:10.1109/TPAMI.2018.2889774.