Hierarchical Variational Models

Rajesh Ranganath RAJESHR@CS.PRINCETON.EDU
Princeton University, 35 Olden St., Princeton, NJ 08540

Dustin Tran DUSTIN@CS.COLUMBIA.EDU
David M. Blei DAVID.BLEI@COLUMBIA.EDU
Columbia University, 500 W 120th St., New York, NY 10027

Abstract

Black box variational inference allows researchers to easily prototype and evaluate an array of models. Recent advances allow such algorithms to scale to high dimensions. However, a central question remains: How to specify an expressive variational distribution that maintains efficient computation? To address this, we develop hierarchical variational models (HVMs). HVMs augment a variational approximation with a prior on its parameters, which allows it to capture complex structure for both discrete and continuous latent variables. The algorithm we develop is black box, can be used for any HVM, and has the same computational efficiency as the original approximation. We study HVMs on a variety of deep discrete latent variable models. HVMs generalize other expressive variational distributions and maintains higher fidelity to the posterior.

1. Introduction

Black box variational inference (BBVI) is important to realizing the potential of modern applied Bayesian statistics. The promise of BBVI is that an investigator can specify any probabilistic model of hidden and observed variables, and then efficiently approximate its posterior without additional effort (Ranganath et al., 2014).

BBVI is a form of variational inference (Jordan et al., 1999). It sets up a parameterized family of distributions over the latent variables and then optimizes the parameters to be close to the posterior. Most applications of variational inference use the mean-field family. Each variable is independent and governed by its own parameters.

Though it enables efficient inference, the mean-field family is limited by its strong factorization. It cannot capture posterior dependencies between latent variables, dependencies which both improve the fidelity of the approximation and are sometimes of intrinsic interest.

To this end, we develop *hierarchical variational model*s *(*HVMs*)*, a class of families that goes beyond the mean-field and, indeed, beyond directly parameterized variational families in general. The main idea behind our method is to treat the variational family as a model of the latent variables and then to expand this model hierarchically. Just as hierarchical Bayesian models induce dependencies between data, hierarchical variational models induce dependencies between latent variables.

We develop an algorithm for fitting HVMs in the context of black box inference. Our algorithm is as general and computationally efficient as BBVI with the mean-field family, but it finds better approximations to the posterior. We demonstrate HVMs with a study of approximate posteriors for several variants of deep exponential families (Ranganath et al., 2015); HVMs generally outperform mean-field variational inference.

Technical summary. Consider a posterior distribution $p(\mathbf{z} \,|\, \mathbf{x})$, a distribution on d latent variables $\mathbf{z}_1, \ldots, \mathbf{z}_d$ conditioned on a set of observations $\mathbf{x}$. The mean-field family is a factorized distribution of the latent variables,

$$q_{\text{MF}}(\mathbf{z}; \boldsymbol{\lambda}) = \textstyle\prod_{i=1}^{d} q(z_i; \boldsymbol{\lambda}_i). \tag{1}$$

We fit its parameters $\boldsymbol{\lambda}$ to find a variational distribution that is close to the exact posterior.

By positing Eq. 1 as a model of the latent variables, we can expand it by placing a prior on its parameters. The result is a *hierarchical variational model*, a two-level distribution that first draws variational parameters from a prior $q(\boldsymbol{\lambda}; \boldsymbol{\theta})$ and then draws latent variables from the corresponding likelihood (Eq. 1). HVMs induce a family that

Proceedings of the 33^{rd} *International Conference on Machine Learning*, New York, NY, USA, 2016. JMLR: W&CP volume 48.

marginalizes out the mean-field parameters,

$$q_{\text{HVM}}(\mathbf{z};\boldsymbol{\theta}) = \int q(\boldsymbol{\lambda};\boldsymbol{\theta}) \prod_i q(z_i \,|\, \boldsymbol{\lambda}_i)\, \mathrm{d}\boldsymbol{\lambda}. \tag{2}$$

This expanded family can capture both posterior dependencies between the latent variables and more complex marginal distributions, thus better inferring the posterior. (We note that during inference the variational "posterior" $q(\boldsymbol{\lambda} \,|\, \mathbf{z}, \boldsymbol{\theta})$ will also play a role; it is the conditional distribution of the variational parameters given a realization of the hidden variables.)

Fitting an HVM involves optimizing the variational hyperparameters $\boldsymbol{\theta}$, and our algorithms for solving this problem maintain the computational efficiency of BBVI. Note the prior is a choice. As one example, we use mixture models as a prior of the mean-field parameters. As another, we use normalizing flows (Rezende and Mohamed, 2015), expanding their scope to a broad class of non-differentiable models.

2. Hierarchical Variational Models

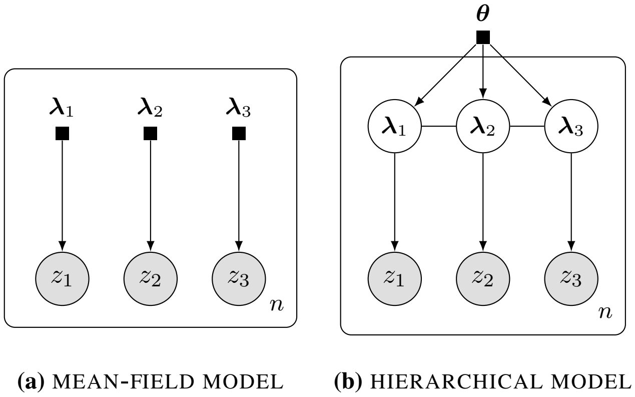
Recall, $p(\mathbf{z} \,|\, \mathbf{x})$ is the posterior. Variational inference frames posterior inference as optimization: posit a family of distributions $q(\mathbf{z};\boldsymbol{\lambda})$, parameterized by $\boldsymbol{\lambda}$, and minimize the KL divergence to the posterior distribution (Jordan et al., 1999; Wainwright and Jordan, 2008).

Classically, variational inference uses the mean-field family. In the mean-field family, each latent variable is assumed independent and governed by its own variational parameter (Eq. 1). This leads to a computationally efficient optimization problem that can be solved (up to a local optimum) with coordinate descent (Bishop, 2006; Ghahramani and Beal, 2001) or gradient-based methods (Hoffman et al., 2013; Ranganath et al., 2014).

Though effective, the mean-field factorization compromises the expressiveness of the variational family: it abandons any dependence structure in the posterior, and it cannot in general capture all marginal information. One of the challenges of variational inference is to construct richer approximating families—thus yielding high fidelity posterior approximations—and while still being computationally tractable. We develop a framework for such families.

2.1. Hierarchical variational models

Our central idea is to draw an analogy between probability models of data and variational distributions of latent variables. A probability model outlines a family of distributions over data, and how large that family is depends on the model's complexity. One common approach to expanding the complexity, especially in Bayesian statistics, is to expand a model hierarchically, i.e., by placing a prior on the parameters of the likelihood. Expanding a model hierarchically has distinct advantages: it induces new dependencies between the data, either through shrinkage or an explicitly correlated prior (Efron, 2012), and it enables us to reuse algorithms for the simpler model within algorithms for the richer model (Gelman and Hill, 2007).



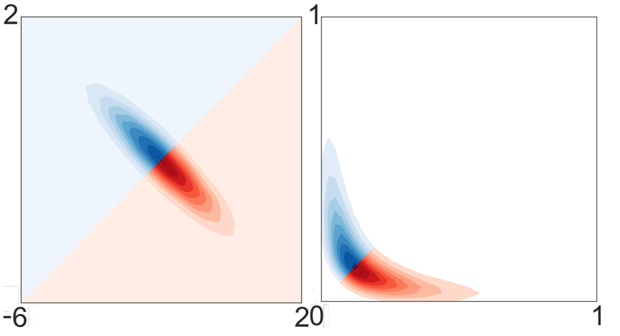
(a) MEAN-FIELD MODEL **(b)** HIERARCHICAL MODEL

Figure 1. Graphical model representation. **(a)** In mean-field models, the latent variables are strictly independent. **(b)** In hierarchical variational models, the latent variables are governed by a prior distribution on their parameters, which induces arbitrarily complex structure.

We use the same idea to expand the complexity of the mean-field variational family and to construct *hierarchical variational models* (HVMs). First, we view the mean-field family of Eq. 1 as a simple model of the latent variables. Next, we expand it hierarchically. We introduce a "variational prior" $q(\boldsymbol{\lambda};\boldsymbol{\theta})$ with "variational hyperparameters" $\boldsymbol{\theta}$ and place it on the mean-field model (a type of "variational likelihood"). Marginalizing out the prior gives $q_{\text{HVM}}(\mathbf{z};\boldsymbol{\theta})$, the hierarchical family of distributions over the latent variables in Eq. 2. This family enjoys the advantages of hierarchical modeling in the context of variational inference: it induces dependence among the latent variables and allows us to reuse simpler computation when fitting the more complex family.

Figure 1 illustrates the difference between the mean-field family and an HVM. Mean-field inference fits the variational parameters $\{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_d\}$ so that the factorized distribution is close to the exact posterior; this tries to match the posterior marginal for each variable. Using the same principle, HVM inference fits the variational hyperparameters so $q_{\text{HVM}}(\mathbf{z};\boldsymbol{\theta})$ is close to the exact posterior. This goes beyond matching marginals because of the shrinkage effects among the variables.

Figure 2 is a simple example. The variational family posits each z_i as a scalar from an exponential family. The variational parameters $\boldsymbol{\lambda}_i$ are the corresponding natural parameters, which are unconstrained. Now place a Gaussian prior



(a) $\mathrm{Normal}(\boldsymbol{\lambda};\boldsymbol{\theta})$ **(b)** $\prod_{i=1}^{2}\mathrm{Gamma}(z_i\,|\,\boldsymbol{\lambda}_i)$

Figure 2. **(a)** $q(\boldsymbol{\lambda};\boldsymbol{\theta})$: The (reparameterized) natural parameters assume a multivariate prior, with different areas indicated by red and blue. **(b)** $\prod_{i=1}^{2} q(z_i\,|\,\boldsymbol{\lambda}_i)$: The latent variables are drawn from a mean-field family, colorized according to its drawn parameters' color.

on the mean-field parameters, with a full covariance matrix. The resulting HVM is a two-level distribution: first draw the complete set of variational parameters $\{\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_d\}$ from a Gaussian (Figure 2a); then draw each z_i from its corresponding natural parameter (Figure 2b). The covariance on the variational parameters induces dependence among the z_i's, and the marginal of each z_i is an integrated likelihood; thus this HVM is more flexible than the mean-field family.

In general, if the HVM can capture the same marginals then $q_{\mathrm{HVM}}(\mathbf{z};\boldsymbol{\theta})$ is more expressive than the mean-field family.[1] As in the example, the HVM induces dependence among variables and also expands the family of possible marginals that it can capture. In Section 3 we see that, even with this more expressive family, we can develop a black box algorithm for HVMs. It exploits the mean-field structure of the variational likelihood and enjoys the corresponding computational advantages. First, we discuss how to specify an HVM.

2.2. Specifying an HVM

We can construct an HVM by placing a prior on any existing variational approximation. An HVM has two components: the variational likelihood $q(\mathbf{z}\mid\boldsymbol{\lambda})$ and the prior $q(\boldsymbol{\lambda};\boldsymbol{\theta})$. The likelihood comes from a variational family that admits gradients; here we focus on the mean-field family. As for the prior, the distribution of $\{\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_d\}$ should not have the same factorization structure as the variational likelihood—otherwise it will not induce dependence between latent variables. We outline several examples of variational priors.

Variational prior: Mixture of Gaussians. One option for a variational prior is to assume the mean-field parameters $\boldsymbol{\lambda}$ are drawn from a mixture of Gaussians. Let K be the number of components, $\boldsymbol{\pi}$ be a probability vector, $\boldsymbol{\mu}_k$, and Σ_k be the parameters of a d-dimensional multivariate Gaussian. The variational prior is

$$q(\boldsymbol{\lambda};\boldsymbol{\theta})=\sum_{i=1}^{K}\boldsymbol{\pi}_k\mathrm{N}(\boldsymbol{\mu}_k,\Sigma_k).$$

The parameters $\boldsymbol{\theta}$ contain the probability vector $\boldsymbol{\pi}$ as well as the component means $\boldsymbol{\mu}_k$ and variances Σ_k. The mixture locations $\boldsymbol{\mu}_k$ capture relationships between different latent variables. For example, a two-component mixture with two latent variables (and a mean field variational likelihood) can capture that the latent variables are either very positive or very negative.

Mixtures can approximate arbitrary distributions (given enough components), and have been considered as variational families (Jaakkola and Jordan, 1998; Lawrence, 2000; Gershman and Blei, 2012; Salimans et al., 2013). In the traditional setup, however, the mixtures form the variational appproximation on the latent variables directly. Here we use it on the variational parameters; this lets us use a mixture of Gaussians in many models, including those with discrete latent variables.

Variational prior: Normalizing flows. Mixtures offer flexible variational priors. However, in the algorithms we derive, the number of model likelihood evaluations scales with the number of mixture components; this is problematic in high dimensions. Further, in high dimensions the number of mixtures components can be impractical. We seek a prior whose computational complexity does not scale with its modeling flexibility. This motivates normalizing flows.

Normalizing flows are variational approximations for probability models with differentiable densities (Rezende and Mohamed, 2015). Normalizing flows build a parameterized probability distribution by transforming a simple random variable $\boldsymbol{\lambda}_0$ through a sequence of invertible differentiable functions f_1 to f_K. Each function transforms its input, so the distribution of the output is a complex warping of the original random variable $\boldsymbol{\lambda}_0$.

We can use normalizing flows as a variational prior. Let $\boldsymbol{\lambda}_k=f_k\circ\ldots\circ f_1(\boldsymbol{\lambda}_0)$; then the flow's density is

$$q(\boldsymbol{\lambda};\theta)=q(\boldsymbol{\lambda}_0)\prod_{k=1}^{K}\left|\det\left(\frac{\partial f_k}{\partial\boldsymbol{\lambda}_k}\right)\right|^{-1}.$$

With the normalizing flow prior, the latent variables become dependent because their variational parameters are deterministic functions of the same random variable. The

[1] Using an HVM to "regularize" the variational family, i.e., to induce dependence but limit the marginals, is an interesting avenue for future work. In the appendix, we relate HVMs to empirical Bayes and methods in reinforcement learning.

HVM expands the use of normalizing flows to non-differentiable latent variables, such as those with discrete, ordinal, and discontinuous support. In Section 4.2, we use normalizing flows to better approximate posteriors of discrete latent variables.

Other Variational Models. Many modeling tools can be brought to bear on building hierarchical variational models. For example, copulas explicitly introduce dependence among d random variables by using joint distributions on d-dimensional hypercubes (Nelsen, 2006). HVMs can use copulas as priors on either point mass or general mean-field likelihoods. As another example, we can replace the mixture model prior with a factorial mixture (Ghahramani, 1995). This leads to a richer posterior approximation.

2.3. Related work

There has been much work on learning posterior dependences. Saul and Jordan (1996); Ghahramani (1997) develop structured variational approximations: they factorize the variational family across subsets of variables, maintaining certain dependencies in the model. Unlike HVMs, however, structured approximations require model-specific considerations and can scale poorly when used with black box methods. For example, Mnih and Gregor (2014) develop a structured approximation for sigmoid belief networks—their approach is restricted to stochastic feed forward networks, and the variance of the stochastic gradients increases with the number of layers. In general, these families complement the construction of HVMs, and can be applied as a variational likelihood.

Within the context of more generic inference, Titsias and Lázaro-Gredilla (2014); Rezende and Mohamed (2015); Kucukelbir et al. (2016) propose rich approximating families in differentiable probability models. These methods work well in practice; however, they are restricted to probability models with densities differentiable with respect to their latent variables. For undirected models Agakov and Barber (2004) introduced the auxiliary bound for variational inference we derive. Salimans et al. (2015) derive the same bound, but limit their attention to differentiable probability models and auxiliary distributions defined by Markov transition kernels. Maaløe et al. (2016) study auxiliary distributions for semi-supervised learning with deep generative models. Tran et al. (2015) propose copulas as a way of learning dependencies in factorized approximations. Copulas can be efficiently extended to HVMs, whereas the full rank approach taken in Tran et al. (2015) requires computation quadratic in the number of latent variables. Giordano et al. (2015) use linear response theory to recover covariances from mean-field estimates. Their approach requires recovering the correct first order moments by mean-field inference and only provides estimates of smooth functions.

These generic methods can also be building blocks for HVMs, employed as variational priors for arbitrary mean-field factors. As in our example with a normalizing flow prior, this extends their scope to perform inference in discrete models (and, more generally, non-differentiable models). In other work, we use Gaussian processes (Tran et al., 2016b).

3. Optimizing HVMs

We derive a black box variational inference algorithm for a large class of probability models and using any hierarchical variational model as the posterior approximation. Our algorithm enables efficient inference by preserving both the computational complexity and variance properties of the stochastic gradients of the variational likelihood.

Hierarchical ELBO. We optimize over the parameters $\boldsymbol{\theta}$ of the variational prior to find the optimal distribution within the class of hierarchical variational models. Using the HVM, the ELBO is

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{q_{\text{HVM}}(\mathbf{z};\boldsymbol{\theta})}[\log p(\mathbf{x},\mathbf{z}) - \log q_{\text{HVM}}(\mathbf{z};\boldsymbol{\theta})]. \quad (3)$$

The expectation of the first term is tractable as long as we can sample from q and it has proper support. The expectation of the second term is the entropy. It contains an integral (Eq. 2) with respect to the variational prior, which is analytically intractable in general.

We construct a bound on the entropy. We introduce a distribution, $r(\boldsymbol{\lambda} \mid \mathbf{z}; \boldsymbol{\phi})$ with parameters $\boldsymbol{\phi}$ and apply the variational principle;

$$\begin{aligned} &-\mathbb{E}_{q_{\text{HVM}}}[\log q_{\text{HVM}}(\mathbf{z})] \quad (4) \\ &\quad \geq -\mathbb{E}_{q(\mathbf{z},\boldsymbol{\lambda})}[\log q(\boldsymbol{\lambda}) + \log q(\mathbf{z} \mid \boldsymbol{\lambda}) - \log r(\boldsymbol{\lambda} \mid \mathbf{z}; \boldsymbol{\phi})]. \end{aligned}$$

As in variational inference, the bound in Eq. 4 is exact when $r(\boldsymbol{\lambda} \mid \mathbf{z}; \boldsymbol{\phi})$ matches the *variational posterior* $q(\boldsymbol{\lambda} \mid \mathbf{z}; \boldsymbol{\theta})$. From this perspective, we can view r as a recursive variational approximation. It is a model for the posterior q of the mean-field parameters $\boldsymbol{\lambda}$ given a realization of the latent variables $\mathbf{z}$.

The bound is derived by introducing a term $\text{KL}(q \| r)$. Due to the asymmetry of KL-divergence, we can also substitute r into the first rather than the second argument of the KL divergence; this produces an alternative bound to Eq. 4. We can also extend the bound to multi-level hierarchical variational models, where now we model the posterior distribution q of the higher levels using higher levels in r. More details are available in the appendix.

Substituting the entropy bound (Eq. 4) into the ELBO (Eq. 3) gives a tractable lower bound. The *hierarchical*

ELBO is

$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{q(\mathbf{z}, \boldsymbol{\lambda}; \boldsymbol{\theta})}\Big[\log p(\mathbf{x}, \mathbf{z}) + \log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi}) \\ - \sum_{i=1}^{d} \log q(z_i \,|\, \boldsymbol{\lambda}_i) - \log q(\boldsymbol{\lambda}; \boldsymbol{\theta})\Big]. \tag{5}$$

The hierarchical ELBO is tractable, as all of the terms are tractable. We jointly fit q and r by maximizing Eq. 5 with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. Alternatively, the joint maximization can be interpreted as variational EM on an expanded probability model, $r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi}) p(\mathbf{z} \,|\, \mathbf{x})$. In this light, $\boldsymbol{\phi}$ are model parameters and $\boldsymbol{\theta}$ are variational parameters. Optimizing $\boldsymbol{\theta}$ improves the posterior approximation; optimizing $\boldsymbol{\phi}$ tightens the bound on the KL divergence by improving the recursive variational approximation.

We can also analyze Eq. 5 by rewriting it in terms of the mean-field ELBO,

$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_q[\mathcal{L}_{\text{MF}}(\boldsymbol{\lambda})] + \mathbb{E}_q[\log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi}) - \log q(\boldsymbol{\lambda}; \boldsymbol{\theta})].$$

where $\mathcal{L}_{\text{MF}} = \mathbb{E}_{q(\mathbf{z} \,|\, \boldsymbol{\lambda})}[\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \,|\, \boldsymbol{\lambda})]$. This shows that Eq. 5 is a sum of two terms: a Bayesian model average of the ELBO of the variational likelihood, with weights given by the variational prior $q(\boldsymbol{\lambda}; \boldsymbol{\theta})$; and a correction term that is a function of both the auxiliary distribution r and the variational prior. Since mixtures (i.e., convex combinations) cannot be sharper than their components, r must not be independent of $\mathbf{z}$ in order for this bound to be better than the original bound.

Stochastic Gradient of the ELBO. Before we describe how to optimize the hierarchical ELBO, we describe two types of stochastic gradients of the ELBO.

The score function estimator for the ELBO gradient applies to both discrete and continuous latent variable models. Let V be the score function, $V = \nabla_{\boldsymbol{\lambda}} \log q(\mathbf{z} \,|\, \boldsymbol{\lambda})$. The gradient of the ELBO is

$$\nabla^{score}_{\boldsymbol{\lambda}} \mathcal{L} = \mathbb{E}_{q(\mathbf{z} \,|\, \boldsymbol{\lambda})}[V(\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \,|\, \boldsymbol{\lambda}))]. \tag{6}$$

See Ranganath et al. (2014) for a derivation. We can construct noisy gradients from Eq. 6 by a Monte Carlo estimate of the expectation. In general, the score function estimator exhibits high variance.[2] Roughly, the variance of the estimator scales with the number of factors in the learning signal (Ranganath et al., 2014; Mnih and Gregor, 2014; Rezende et al., 2014).

In mean-field models, the gradient of the ELBO with respect to $\boldsymbol{\lambda}_i$ can be separated. Letting V_i be the local score $V_i = \nabla_{\boldsymbol{\lambda}} \log q(z_i \,|\, \boldsymbol{\lambda}_i)$, it is

$$\nabla_{\boldsymbol{\lambda}_i} \mathcal{L}_{\text{MF}} = \mathbb{E}_{q(z_i; \boldsymbol{\lambda}_i)}[V_i(\log p_i(\mathbf{x}, \mathbf{z}) - \log q(z_i; \boldsymbol{\lambda}_i))], \tag{7}$$

where $\log p_i(\mathbf{x}, \mathbf{z})$ are the components in the joint distribution that contain z_i. This update is not only local but it also drastically reduces the variance of Eq. 6. It makes stochastic optimization practical.

With differentiable latent variables, the estimator can take advantage of model gradients. One such estimator uses reparameterization: the ELBO is written in terms of a random variable $\boldsymbol{\epsilon}$, whose distribution $s(\boldsymbol{\epsilon})$ is free of the variational parameters, and such that $\mathbf{z}$ can be written as a deterministic function $\mathbf{z} = \mathbf{z}(\boldsymbol{\epsilon}; \boldsymbol{\lambda})$. Reparameterization allows gradients of variational parameters to move inside the expectation,

$$\nabla^{rep}_{\boldsymbol{\lambda}} \mathcal{L} = \mathbb{E}_{s(\boldsymbol{\epsilon})}[(\nabla_{\mathbf{z}} \log p(\mathbf{x}, \mathbf{z}) - \nabla_{\mathbf{z}} \log q(\mathbf{z}))\nabla_{\boldsymbol{\lambda}} \mathbf{z}(\boldsymbol{\epsilon}; \boldsymbol{\lambda})].$$

The reparameterization gradient constructs noisy gradients from this expression via Monte Carlo. Empirically, the reparameterization gradient exhibits lower variance than the score function gradient (Titsias, 2015). In the appendix, we show an analytic equality of the two gradients, which explains the observed difference in variances.

Stochastic Gradient of the Hierarchical ELBO. To optimize Eq. 5, we need to compute the stochastic gradient with respect to the variational hyperparameters $\boldsymbol{\theta}$ and auxiliary parameters $\boldsymbol{\phi}$. As long as we specify the variational prior $q(\boldsymbol{\lambda}; \boldsymbol{\theta})$ to be differentiable, we can apply the reparameterization gradient for the random variational parameters $\boldsymbol{\lambda}$. Let $\boldsymbol{\epsilon}$ be drawn from a distribution $s(\boldsymbol{\epsilon})$ such as the standard normal. Let $\boldsymbol{\lambda}$ be written as a function of $\boldsymbol{\epsilon}$ and $\boldsymbol{\theta}$ denoted $\boldsymbol{\lambda}(\boldsymbol{\epsilon}; \boldsymbol{\theta})$. The gradient of the hierarchical ELBO with respect to $\boldsymbol{\theta}$ is

$$\nabla_{\boldsymbol{\theta}} \widetilde{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}} \boldsymbol{\lambda}(\boldsymbol{\epsilon}) \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{MF}}(\boldsymbol{\lambda})] \\ + \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}} \boldsymbol{\lambda}(\boldsymbol{\epsilon}) \nabla_{\boldsymbol{\lambda}} [\log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi}) - \log q(\boldsymbol{\lambda}; \boldsymbol{\theta})]] \\ + \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}} \boldsymbol{\lambda}(\boldsymbol{\epsilon}) \mathbb{E}_{q(\mathbf{z} \,|\, \boldsymbol{\lambda})}[V \log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi})]]. \tag{8}$$

The first term is the gradient of the original variational approximation scaled by the chain rule from the reparameterization. Thus, hierarchical variational models inherit properties from the original variational approximation such as the variance reduced gradient (Eq. 7) from the mean-field factorization. The second and third terms try to match r and q. The second term is strictly based on reparameterization, thus it exhibits low variance.

The third term potentially involves a high variance gradient due to the appearance of all the latent variables in its gradient. Since the distribution $q(\mathbf{z} \,|\, \boldsymbol{\lambda}(\boldsymbol{\epsilon}; \boldsymbol{\theta}))$ factorizes (by definition) we can apply the same variance reduction for r as for the mean-field model. We examine this below.

[2] This is not surprising given that the score function estimator makes very few restrictions on the class of models, and requires access only to zero-order information given by the learning signal $\log p - \log q$.

Local Learning with r. The practicality of HVMs hinges on the variance of the stochastic gradients during optimization. Specifically, any additional variance introduced by r should be minimal. Let r_i be the terms $\log r(\boldsymbol{\lambda} \mid z_i)$ containing z_i. Then the last term in Eq. 8 can be rewritten as

$$\begin{aligned}&\mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[V\log r(\boldsymbol{\lambda}\mid\mathbf{z};\boldsymbol{\phi})]]\\ &=\mathbb{E}_{s(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}\left[\sum_{i=1}^{d}V_i\log r_i(\boldsymbol{\lambda}\mid\mathbf{z};\boldsymbol{\phi})\right]\right].\end{aligned}$$

We derive this expression (along with Eq. 8) in the appendix. When r_i does not depend on many variables, this gradient combines the computational efficiency of the mean-field with reparameterization, enabling fast inference for discrete and continuous latent variable models. This gradient also gives us the criteria for building an r that admits efficient stochastic gradients: r should be differentiable with respect to $\boldsymbol{\lambda}$, flexible enough to model the variational posterior $q(\boldsymbol{\lambda} \mid \mathbf{z})$, and factorize with respect to its dependence on each z_i.

One way to satisfy these criteria is by defining r to be a deterministic transformation of a factorized distribution. That is, let $\boldsymbol{\lambda}$ be the deterministic transform of $\boldsymbol{\lambda}_0$, and

$$r(\boldsymbol{\lambda}_0 \mid \mathbf{z}) = \prod_{i=1}^{d} r(\boldsymbol{\lambda}_{0i} \mid z_i). \tag{9}$$

Similar to normalizing flows, the deterministic transformation from $\boldsymbol{\lambda}_0$ to $\boldsymbol{\lambda}$ can be a sequence of invertible, differentiable functions g_1 to g_k. However unlike normalizing flows, we let the inverse functions g^{-1} have a known parametric form. We call this the *inverse flow*. Under this transformation, the log density of r is

$$\log r(\boldsymbol{\lambda} \mid \mathbf{z}) = \log r(\boldsymbol{\lambda}_0 \mid \mathbf{z}) + \sum_{k=1}^{K}\log\left(\left|\det(\frac{\partial g_k^{-1}}{\partial \boldsymbol{\lambda}_k})\right|\right).$$

The distribution r is parameterized by a deterministic transformation of a factorized distribution. We can quickly compute the sequence of intermediary $\boldsymbol{\lambda}$ by applying the known inverse functions—this enables us to quickly evaluate the log density of inverse flows at arbitrary points. This contrasts normalizing flows, where evaluating the log density of a value (not generated by the flow) requires inversions for each transformation.

This r meets our criteria. It is differentiable, flexible, and isolates each individual latent variable in a single term. It maintains the locality of the mean-field inference and is therefore crucial to the stochastic optimization.

Optimizing the Hierarchical ELBO with respect to $\boldsymbol{\phi}$. We derived how to optimize with respect to $\boldsymbol{\theta}$. Optimizing with respect to the auxiliary parameters $\boldsymbol{\phi}$ is simple. The expectation in the hierarchical ELBO (Eq. 5) does not depend on $\boldsymbol{\phi}$; therefore we can simply pass the gradient operator inside,

$$\nabla_{\boldsymbol{\phi}}\widetilde{\mathcal{L}} = \mathbb{E}_{q(\mathbf{z},\boldsymbol{\lambda})}[\nabla_{\boldsymbol{\phi}}\log r(\boldsymbol{\lambda} \mid \mathbf{z}, \boldsymbol{\phi})]. \tag{10}$$

Algorithm 1: Black box inference with an HVM

```
Input  : Model $\log p(\mathbf{x}, \mathbf{z})$,
         Variational model $q(\mathbf{z} \mid \boldsymbol{\lambda})q(\boldsymbol{\lambda}; \boldsymbol{\theta})$
Output: Variational Parameters: $\boldsymbol{\theta}$
Initialize $\boldsymbol{\phi}$ and $\boldsymbol{\lambda}$ randomly.
while not converged do
    Compute unbiased estimate of $\nabla_{\boldsymbol{\theta}}\widetilde{\mathcal{L}}$.    (Eq. 8)
    Compute unbiased estimate of $\nabla_{\boldsymbol{\phi}}\widetilde{\mathcal{L}}$.    (Eq. 10)
    Update $\boldsymbol{\phi}$ and $\boldsymbol{\lambda}$ using stochastic gradient ascent.
end
```

Algorithm. Algorithm 1 outlines the inference procedure, where we evaluate noisy estimates of both gradients using samples from the joint $q(\mathbf{z}, \boldsymbol{\lambda})$. In general, we can compute these gradients via automatic differentiation systems such as those available in Stan and Theano (Stan Development Team, 2015; Bergstra et al., 2010). This removes the need for model-specific computations (note that no assumption has been made on $\log p(\mathbf{x}, \mathbf{z})$ other than the ability to calculate it).

Table 1 outlines variational methods and their complexity requirements. HVMs, with a normalizing flow prior, have complexity linear in the number of latent variables, and the complexity is proportional to the length of the flow used to represent q and the inverse flow r.

Hierarchical variational models with multiple layers can contain both discrete and differentiable latent variables. Higher level differentiable variables follow directly from our derivation above. (See the appendix.)

Inference Networks. Classically, variational inference on models with latent variables associated with a data point requires optimizing over variational parameters whose number grows with the size of data. This process can be computationally prohibitive, especially at test time. Inference networks (Dayan, 2000; Stuhlmüller et al., 2013; Kingma and Welling, 2014; Rezende et al., 2014) amortize the cost of estimating these local variational parameters by tying them together through a neural network. Specifically, the data-point specific variational parameters are outputs of a neural network with the data point as input. The parameters of the neural network then become the variational parameters; this reduces the cost of estimating the parameters of all the data points to estimating parameters of the inference network. Inference networks can be applied to HVMs

Black box methods	Computation	Storage	Dependency	Class of models
BBVI (Ranganath et al., 2014)	$\mathcal{O}(d)$	$\mathcal{O}(d)$	✗	discrete/continuous
DSVI (Titsias and Lázaro-Gredilla, 2014)	$\mathcal{O}(d^2)$	$\mathcal{O}(d^2)$	✓	continuous-diff.
COPULA VI (Tran et al., 2015)	$\mathcal{O}(d^2)$	$\mathcal{O}(d^2)$	✓	discrete/continuous
MIXTURE (Jaakkola and Jordan, 1998)	$\mathcal{O}(Kd)$	$\mathcal{O}(Kd)$	✓	discrete/continuous
NF (Rezende and Mohamed, 2015)	$\mathcal{O}(Kd)$	$\mathcal{O}(Kd)$	✓	continuous-diff.
HVM w/ NF prior	$\mathcal{O}(Kd)$	$\mathcal{O}(Kd)$	✓	discrete/continuous

Table 1. A summary of black box inference methods, which can support either continuous-differentiable distributions or both discrete and continuous. d is the number of latent variables; for MIXTURE, K is the number of mixture components; for NF procedures, K is the number of transformations.

by making both the parameters of the variational model and recursive posterior approximation functions of their conditioning sets.

4. Empirical Study

We introduced a new class of variational families and developed efficient black box algorithms for their computation. We consider a simulated study on a two-dimensional discrete posterior; we also evaluate our proposed variational models on deep exponential families (Ranganath et al., 2015), a class of deep generative models which achieve state-of-the-art results on text analysis. In total, we train 2 variational models for the simulated study and 12 models over two datasets.[3]

4.1. Correlated Discrete Latent Variables

Consider a model whose posterior distribution is a pair of discrete latent variables defined on the countable support $\{0, 1, 2, \ldots, \} \times \{0, 1, 2, \ldots, \}$; Figure 3 depicts its probability mass in each dimension. The latent variables are correlated and form a complex multimodal structure. A mean-field Poisson approximation has difficulty capturing this distribution; it focuses entirely on the center mass. This contrasts hierarchical variational models, where we place a mixture prior on the Poisson distributions' rate parameters (reparameterized to share the same support). This HVM fits the various modes of the correlated Poisson latent variable model and exhibits a "smoother" surface due to its multimodality.

4.2. Deep Exponential Families

Deep exponential families (DEFs) (Ranganath et al., 2015) build a set of probability models from exponential families (Brown, 1986), whose latent structure mimic the architectures used in deep neural networks.

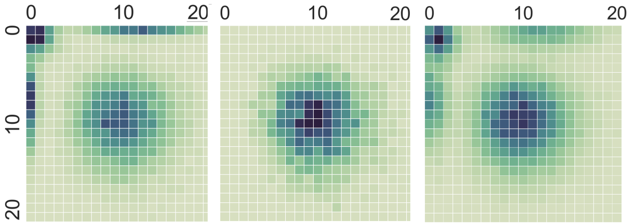


Figure 3. **(a)** The true posterior, which has correlated latent variables with countably infinite discrete support. **(b)** Mean-field Poisson approximation. **(c)** Hierarchical variational model with a mixture of Gaussians prior. Using this prior, the HVM exhibits high fidelity to the posterior as it capture multimodality on discrete surfaces.

Model. Exponential families are parameterized by a set of natural parameters. We denote a draw from an unspecified exponential family with natural parameter η as $\text{EXPFAM}(\eta)$. The natural parameter in deep exponential families are constructed from an inner product of the previous layer with weights, passed through a link function $g(\cdot)$.

Let L be the total number of layers, $\mathbf{z}_\ell$ be a vector of latent variables for layer ℓ (with $\mathbf{z}_{\ell,k}$ as an element), and $\mathbf{W}_{\ell,k}$ be shared weights across observations. DEFs use weights with priors, $\mathbf{W}_{\ell,k} \sim \text{EXPFAM}_W(\xi)$, and a prior at the top layer, $\mathbf{z}_{L,k} \sim \text{EXPFAM}_L(\eta)$. The generative process cascades: for each element k in layer $\ell = L - 1, \ldots, 1$,

$$\mathbf{z}_{\ell,k} \sim \text{EXPFAM}_\ell(g_\ell(\mathbf{W}_{\ell,k}^\top \mathbf{z}_{\ell+1}))$$
$$\mathbf{x} \sim \text{Poisson}(\mathbf{W}_0 \mathbf{z}_1).$$

We model count data with a Poisson likelihood on $\mathbf{x}$. We focus on DEFs with discrete latent variables.

The canonical example of a discrete DEF is the sigmoid belief network (SBN) (Neal, 1990). The SBN is a Bernoulli DEF, with $\mathbf{z}_{\ell,k} \in \{0, 1\}$. The other family of models we consider is the Poisson DEF, with

$$p(\mathbf{z}_{\ell,k} \,|\, \mathbf{z}_{l+1}, \mathbf{W}_{l,k}) \sim \text{Poisson}(\log(1 + \exp(\mathbf{z}_{l+1}^\top \mathbf{W}_{l,k}))),$$

for each element k in the layer ℓ. In the SBN, each observation either turns a feature on or off. In a Poisson DEF, each

[3] An implementation of HVMs is available in Edward (Tran et al., 2016a), a Python library for probabilistic modeling.

	Model	HVM	Mean-Field
Poisson	100	**3386**	3387
	100-30	**3396**	3896
	100-30-15	**3346**	3962
Bernoulli	100	**3060**	3084
	100-30	3394	**3339**
	100-30-15	**3420**	3575

Table 2. New York Times. Held-out perplexity (lower is better). Hierarchical variational models outperform mean-field in five models. Mean-field (Ranganath et al., 2015) fails at multi-level Poissons; HVMs make it possible to study multi-level Poissons.

	Model	HVM	Mean-Field
Poisson	100	**3327**	3392
	100-30	**2977**	3320
	100-30-15	**3007**	3332
Bernoulli	100	**3165**	3166
	100-30	**3135**	3195
	100-30-15	**3050**	3185

Table 3. Science. Held-out perplexity (lower is better). HVM outperforms mean-field on all six models. Hierarchical variational models identify that multi-level Poisson models are best, while mean-field does not.

observation counts each feature a positive integer number of times. This means Poisson DEFs are a multi-feature generalization of SBNs.

Variational Models. We consider the variational approximation that adds dependence to the $\mathbf{z}'s$. We parameterize each variational prior $q(\boldsymbol{\lambda}_{\mathbf{z}_i})$ with a normalizing flow of length 2, and use the inverse flow of length 10 for $r(\boldsymbol{\lambda}_{\mathbf{z}_i})$. We use planar transformations (Rezende and Mohamed, 2015). In a pilot study, we found little improvement with longer flow lengths. We compare to the mean-field approximation from Ranganath et al. (2015) which achieves state of the art results on text.

Data and Evaluation. We consider two text corpora of news and scientific articles— *The New York Times* (NYT) and *Science*. Both have 11K documents. NYT consists of 8K terms and *Science* consists of 5.9K terms. We train six models for each data set.

We examine held out perplexity following the same criteria as Ranganath et al. (2015). This is a document complete evaluation metric (Wallach et al., 2009) where the words are tested independently. As our evaluation uses data not included in posterior inference, it is possible for the mean-field family to outperform HVMs.

Results. HVMs achieve better performance over six models and two datasets, with a mean improvement in perplexity of 180 points. (Mean-field works better on only the two layer Bernoulli model on NYT.) From a data modeling viewpoint, we find that for *The New York Times* there is little advantage to multi-layer models, while on *Science* multi-layer models outperform their single layer counterparts. Overall, hierarchical variational models are less sensitive to inference in multi-layer models, as evidenced by the generally lower performance of mean-field with multiple layers. HVMs make it feasible to work with multi-level Poisson models. This is particularly important on *Science*, where hierarchical variational models identifies that multi-level Poisson models are best.

5. Discussion

We present hierarchical variational models, a rich class of posterior approximations constructed by placing priors on existing variational families. These priors encapsulate different modeling assumptions of the posterior and we explore several choices. We develop a black box algorithm can fit any HVM. There are several avenues for future work: studying alternative entropy bounds; analyzing HVMs in the empirical Bayes framework; and using other data modeling tools to build new variational models.

Acknowledgements. This work is supported by NSF IIS-0745520, IIS-1247664, IIS-1009542, ONR N00014-11-1-0651, DARPA FA8750-14-2-0009, N66001-15-C-4032, Facebook, Adobe, Amazon, NVIDIA, the Porter Ogden Jacobus Fellowship, the Seibel Foundation, and John Templeton Foundations.

References

Agakov, F. V. and Barber, D. (2004). An auxiliary variational method. In *Neural Information Processing Systems*.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (2010). Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer New York.

Brown, L. (1986). *Fundamentals of Statistical Exponential Families*. Institute of Mathematical Statistics, Hayward, CA.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Dayan, P. (2000). Helmholtz machines and wake-sleep learning. *Handbook of Brain Theory and Neural Network. MIT Press, Cambridge, MA*.

Efron, B. (2012). *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, volume 1. Cambridge University Press.

Efron, B. and Morris, C. (1973). Combining possibly related estimation problems. *Journal of the Royal Statistical Society, Series B*, 35(3):379–421.

Gelman, A. and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge Univ. Press.

Gershman, S. J. and Blei, D. M. (2012). A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1 – 12.

Ghahramani, Z. (1995). Factorial learning and the EM algorithm. In *Advances in Neural Information Processing Systems*, pages 617–624.

Ghahramani, Z. (1997). On structured variational approximations. Technical report.

Ghahramani, Z. and Beal, M. (2001). Propagation algorithms for variational Bayesian learning. In *Neural Information Processing Systems*.

Giordano, R. J., Broderick, T., and Jordan, M. I. (2015). Linear response methods for accurate covariance estimates from mean field variational bayes. In *Neural Information Processing Systems*.

Hoffman, M. and Blei, D. (2015). Stochastic structured variational inference. In *International Conference on Artifical Intelligence and Statistics*.

Hoffman, M., Blei, D., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research*, 14(1303–1347).

Jaakkola, T. S. and Jordan, M. I. (1998). Improving the Mean Field Approximation Via the Use of Mixture Distributions. In *Learning in Graphical Models*, pages 163–173. Springer Netherlands, Dordrecht.

Jordan, M., editor (1999). *Learning in Graphical Models*. MIT Press, Cambridge, MA.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999). Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233.

Kingma, D. and Welling, M. (2014). Auto-encoding variational Bayes. In *International Conference on Learning Representations*.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2016). Automatic differentiation variational inference. *arXiv preprint arXiv:1603.00788*.

Lawrence, N. (2000). *Variational Inference in Probabilistic Models*. PhD thesis.

Maaløe, L., Sønderby, C. K., Sønderby, S. K., and Winther, O. (2016). Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*.

Minka, T. (2001). *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology.

Mnih, A. and Gregor, K. (2014). Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*.

Neal, R. (1990). Learning stochastic feedforward networks. *Technical Repport CRG-TR-90-7: Department of Computer Science, University of Toronto*.

Nelsen, R. B. (2006). *An Introduction to Copulas (Springer Series in Statistics)*. Springer-Verlag New York, Inc.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *Artificial Intelligence and Statistics*.

Ranganath, R., Tang, L., Charlin, L., and Blei, D. M. (2015). Deep exponential families. In *Artificial Intelligence and Statistics*.

Rezende, D., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep latent Gaussian models. In *International Conference on Machine Learning*.

Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International Conference on Machine Learning*.

Robbins, H. (1964). The empirical bayes approach to statistical decision problems. *The Annals of Mathematical Statistics*, pages 1–20.

Rückstieß, T., Felder, M., and Schmidhuber, J. (2008). State-dependent exploration for policy gradient methods. In *Machine Learning and Knowledge Discovery in Databases*, pages 234–249. Springer.

Salimans, T., Kingma, D., and Welling, M. (2015). Markov chain Monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*.

Salimans, T., Knowles, D. A., et al. (2013). Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882.

Saul, L. K. and Jordan, M. I. (1996). Exploiting tractable substructures in intractable networks. *Neural Information Processing Systems*.

Sehnke, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., and Schmidhuber, J. (2008). Policy gradients with parameter-based exploration for control. In *Artificial Neural Networks-ICANN 2008*, pages 387–396. Springer.

Stan Development Team (2015). Stan: A C++ library for probability and sampling, version 2.8.0.

Stuhlmüller, A., Taylor, J., and Goodman, N. (2013). Learning stochastic inverses. In *Neural Information Processing Systems*.

Titsias, M. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational Bayes for non-conjugate inference. In *International Conference on Machine Learning*.

Titsias, M. K. (2015). Local expectation gradients for doubly stochastic variational inference. In *Neural Information Processing Systems*.

Tran, D., Blei, D. M., and Airoldi, E. M. (2015). Copula variational inference. In *Neural Information Processing Systems*.

Tran, D., Blei, D. M., Kucukelbir, A., Dieng, A., Rudolph, M., and Liang, D. (2016a). Edward: A library for probabilistic modeling, inference, and criticism.

Tran, D., Ranganath, R., and Blei, D. M. (2016b). The variational Gaussian process. In *International Conference on Learning Representations*.

Wainwright, M. and Jordan, M. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

Wallach, H., Murray, I., Salakhutdinov, R., and Mimno, D. (2009). Evaluation methods for topic models. In *International Conference on Machine Learning*.

A. Appendix

Relationship to empirical Bayes and RL. The augmentation with a variational prior has strong ties to empirical Bayesian methods, which use data to estimate hyperparameters of a prior distribution (Robbins, 1964; Efron and Morris, 1973). In general, empirical Bayes considers the fully Bayesian treatment of a hyperprior on the original prior—here, the variational prior on the original mean-field—and proceeds to integrate it out. As this is analytically intractable, much work has been on parametric estimation, which seek point estimates rather than the whole distribution encoded by the hyperprior. We avoid this at the level of the hyperprior (variational prior) via the hierarchical ELBO; however, our procedure can be viewed in this framework at one higher level. That is, we seek a point estimate of the "variational hyperprior" which governs the parameters on the variational prior.

A similar methodology also arises in the policy search literature (Rückstieß et al., 2008; Sehnke et al., 2008). Policy search methods aim to maximize the expected reward for a sequential decision-making task, by positing a distribution over trajectories and proceeding to learn its parameters. This distribution is known as the policy, and an upper-level policy considers a distribution over the original policy. This encourages exploration in the latent variable space and can be seen as a form of annealing.

Tractable bound on the entropy. Deriving an analytic expression for the entropy of q_{HVM} is generally intractable due to the integral in the definition of q_{HVM}. However, it is tractable when we know the distribution $q(\boldsymbol{\lambda} \,|\, \mathbf{z})$. This can be seen by noting from standard Bayes' rule that

$$q(\mathbf{z})q(\boldsymbol{\lambda} \,|\, \mathbf{z}) = q(\boldsymbol{\lambda})q(\mathbf{z} \,|\, \boldsymbol{\lambda}), \tag{11}$$

and that the right hand side is specified by the construction of the hierarchical variational model. Note also that $q(\boldsymbol{\lambda} \,|\, \mathbf{z})$ can be interpreted as the posterior distribution of the original variational parameters $\boldsymbol{\lambda}$ given the model, thus we will denote it as $q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z})$.

In general, computing $q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z})$ from the specification of the hierarchical variational model is as hard as the integral needed to compute the entropy. Instead, we approximate q_{POST} with an auxiliary distribution $r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi})$ parameterized by $\boldsymbol{\phi}$. This yields a bound on the entropy in terms of the analytically known distributions $r(\boldsymbol{\lambda} \,|\, \mathbf{z})$, $q(\mathbf{z} \,|\, \boldsymbol{\lambda})$, and $q(\boldsymbol{\lambda})$.

First note that the KL-divergence between two distributions is greater than zero, and is precisely zero only when the two distributions are equal. This means the entropy can be bounded as follows:

$$\begin{aligned}
&-\mathbb{E}_{q_{\text{HVM}}}[\log q_{\text{HVM}}(\mathbf{z})] \\
&\quad = -\mathbb{E}_{q_{\text{HVM}}}[\log q_{\text{HVM}}(\mathbf{z}) - \text{KL}(q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z}) || q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z}))] \\
&\quad \geq -\mathbb{E}_{q_{\text{HVM}}}[\log q_{\text{HVM}}(\mathbf{z}) + \text{KL}(q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z}) || r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi}))]] \\
&\quad = -\mathbb{E}_{q_{\text{HVM}}}[\mathbb{E}_{q_{\text{POST}}}[\log q_{\text{HVM}}(\mathbf{z}) + \log q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z}) \\
&\qquad - \log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi})]] \\
&\quad = -\mathbb{E}_{q(\mathbf{z},\boldsymbol{\lambda})}[\log q_{\text{HVM}}(\mathbf{z}) + \log q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z}) - \log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi})].
\end{aligned}$$

Then by Eq. 11, the bound simplifies to

$$\begin{aligned}
&-\mathbb{E}_{q_{\text{HVM}}}[\log q_{\text{HVM}}(\mathbf{z})] \\
&\quad \geq -\mathbb{E}_{q(\mathbf{z},\boldsymbol{\lambda})}[\log q(\boldsymbol{\lambda}) + \log q(\mathbf{z} \,|\, \boldsymbol{\lambda}) - \log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi})].
\end{aligned}$$

A similar bound in derived by Salimans et al. (2015) directly for $\log p(x)$.

In the above derivation, the approximation r to the variational posterior $q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z})$ is placed as the second argument of a KL-divergence term. Replacing the first argument instead yields a different tractable upper bound as well.

$$\begin{aligned}
&-\mathbb{E}_{q_{\text{HVM}}}[\log q(\mathbf{z})] \\
&\quad = \mathbb{E}_{q_{\text{HVM}}}[-\log q(\mathbf{z}) + \text{KL}(q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z}) || q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z}))] \\
&\quad \leq \mathbb{E}_{q_{\text{HVM}}}[-\log q(\mathbf{z}) + \text{KL}(r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi}) || q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z}))]] \\
&\quad = \mathbb{E}_{q_{\text{HVM}}}[\mathbb{E}_r[-\log q(\mathbf{z}) - \log q_{\text{POST}}(\boldsymbol{\lambda} \,|\, \mathbf{z}) + \log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi})]] \\
&\quad = \mathbb{E}_{q_{\text{HVM}}}[\mathbb{E}_r[-\log q(\mathbf{z}) - \log \frac{q(\mathbf{z} \,|\, \boldsymbol{\lambda})q(\boldsymbol{\lambda})}{q(\mathbf{z})} + \log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi})]] \\
&\quad = \mathbb{E}_{q_{\text{HVM}}}[\mathbb{E}_r[-\log q(\boldsymbol{\lambda}) - \log q(\mathbf{z} \,|\, \boldsymbol{\lambda}) + \log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi})]].
\end{aligned}$$

The bound is also tractable when r and q_{HVM} can be sampled and all distributions are analytic. The derivation of these two bounds parallels the development of expectation propagation (Minka, 2001) and variational Bayes (Jordan, 1999) which are based on alternative forms of the KL-divergence[4]. Exploring the role and relative merits of both bounds we derive in the context of variational models will be an important direction in the study of variational models with latent variables.

The entropy bound is tighter than the trivial conditional entropy bound of $\mathbb{H}[q_{\text{HVM}}] \geq \mathbb{H}[q \,|\, \boldsymbol{\lambda}]$ (Cover and Thomas, 2012). This bound is attained when specifying the recursive approximation to be the prior; i.e., it is the special case when $r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi}) = q(\boldsymbol{\lambda}; \boldsymbol{\theta})$.

Gradient Derivation. We derive the gradient of the hierarchical ELBO using its mean-field representation:

$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_q[\mathcal{L}(\boldsymbol{\lambda})] + \mathbb{E}_q[(\log r(\boldsymbol{\lambda} \,|\, \mathbf{z}; \boldsymbol{\phi}) - \log q(\boldsymbol{\lambda}; \boldsymbol{\theta}))].$$

[4] Note that the first bound, which corresponds to the objective in expectation propagation (EP), directly minimizes $\text{KL}(q\|r)$ whereas EP only minimizes this locally.

Using the reparameterization $\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})$, where $\boldsymbol{\epsilon} \sim s$, this is

$$\begin{aligned}\widetilde{\mathcal{L}}(\boldsymbol{\theta},\boldsymbol{\phi}) &= \mathbb{E}_{s(\boldsymbol{\epsilon})}[\mathcal{L}(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta}))] \\ &\quad + \mathbb{E}_{s(\boldsymbol{\epsilon})}[\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[(\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})] - \log q(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta});\boldsymbol{\theta}))].\end{aligned}$$

We now differentiate the three additive terms with respect to $\boldsymbol{\theta}$. As in the main text, we suppress $\boldsymbol{\theta}$ in the definition of $\boldsymbol{\lambda}$ when clear and define the score function

$$V = \nabla_{\boldsymbol{\lambda}} \log q(\mathbf{z}\,|\,\boldsymbol{\lambda}).$$

By the chain rule the derivative of the first term is

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{s(\boldsymbol{\epsilon})}[\mathcal{L}(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta}))] = \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\nabla_{\boldsymbol{\lambda}}\mathcal{L}(\boldsymbol{\lambda})].$$

We now differentiate the second term:

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{s(\boldsymbol{\epsilon})}[\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})]] \\ &= \nabla_{\boldsymbol{\theta}}\mathbb{E}_{s(\boldsymbol{\epsilon})}\left[\int q(\mathbf{z}\,|\,\boldsymbol{\lambda})\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})\,\mathrm{d}\mathbf{z}\right] \\ &= \mathbb{E}_{s(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\theta}}\left[\int q(\mathbf{z}\,|\,\boldsymbol{\lambda})\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})\,\mathrm{d}\mathbf{z}\right]\right] \\ &= \mathbb{E}_{s(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\nabla_{\boldsymbol{\lambda}}\left[\int q(\mathbf{z}\,|\,\boldsymbol{\lambda})\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})\,\mathrm{d}\mathbf{z}\right]\right].\end{aligned}$$

Applying the product rule to the inner derivative gives

$$\begin{aligned}&\nabla_{\boldsymbol{\lambda}}\left[\int q(\mathbf{z}\,|\,\boldsymbol{\lambda})\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})\,\mathrm{d}\mathbf{z}\right] \\ &\quad= \int \nabla_{\boldsymbol{\lambda}} q(\mathbf{z}\,|\,\boldsymbol{\lambda})\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})\,\mathrm{d}\mathbf{z} \\ &\quad\quad+ \int q(\mathbf{z}\,|\,\boldsymbol{\lambda})\nabla_{\boldsymbol{\lambda}}\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})\,\mathrm{d}\mathbf{z} \\ &\quad= \int \nabla_{\boldsymbol{\lambda}}\log q(\mathbf{z}\,|\,\boldsymbol{\lambda})q(\mathbf{z}\,|\,\boldsymbol{\lambda})\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})\,\mathrm{d}\mathbf{z} \\ &\quad\quad+ \int q(\mathbf{z}\,|\,\boldsymbol{\lambda})\nabla_{\boldsymbol{\lambda}}\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})\,\mathrm{d}\mathbf{z} \\ &\quad= \mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[V\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})] \\ &\quad\quad+ \mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[\nabla_{\boldsymbol{\lambda}}\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})].\end{aligned}$$

Substituting this back into the previous expression gives the gradient of the second term

$$\begin{aligned}&\mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[V\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})]] \\ &\quad+ \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[\nabla_{\boldsymbol{\lambda}}\log r(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\,|\,\mathbf{z};\boldsymbol{\phi})]]\end{aligned}$$

The third term also follows by the chain rule

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\mathbb{E}_{s(\boldsymbol{\epsilon})}[\log q(\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta});\boldsymbol{\theta})] \\ &\quad= \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\nabla_{\boldsymbol{\lambda}}\log q(\boldsymbol{\lambda};\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}}\log q(\boldsymbol{\lambda};\boldsymbol{\theta})] \\ &\quad= \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\nabla_{\boldsymbol{\lambda}}\log q(\boldsymbol{\lambda};\boldsymbol{\theta})]\end{aligned}$$

where the last equality follows by

$$\mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\log q(\boldsymbol{\lambda};\boldsymbol{\theta})] = \mathbb{E}_{q(\boldsymbol{\lambda};\boldsymbol{\theta})}[\nabla_{\boldsymbol{\theta}}\log q(\boldsymbol{\lambda};\boldsymbol{\theta})] = \mathbf{0}.$$

Combining these together gives the total expression for the gradient

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\widetilde{L}(\boldsymbol{\theta},\boldsymbol{\phi}) &= \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\nabla_{\boldsymbol{\lambda}}\mathcal{L}_{\text{MF}}(\boldsymbol{\lambda})] \\ &\quad+ \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\nabla_{\boldsymbol{\lambda}}[\log r(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi}) - \log q(\boldsymbol{\lambda};\boldsymbol{\theta})]] \\ &\quad+ \mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[V\log r(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi})]].\end{aligned}$$

Introducing r_i to the gradient. One term of the gradient involves the product of the score function with all of r,

$$\mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[V\log r(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi})]].$$

Localizing (Rao-Blackwellizing) the inner expectation as in Ranganath et al. (2014); Mnih and Gregor (2014) can drastically reduce the variance. Recall that

$$q(\mathbf{z}\,|\,\boldsymbol{\lambda}) = \prod_{i=1}^{d} q(z_i\,|\,\boldsymbol{\lambda}_i).$$

Next, we define V_i to be the score functions of the factor. That is

$$V_i = \nabla_{\boldsymbol{\lambda}}\log q(z_i\,|\,\boldsymbol{\lambda}_i).$$

This is a vector with nonzero entries corresponding to $\boldsymbol{\lambda}_i$. Substituting the factorization into the gradient term yields

$$\mathbb{E}_{s(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon})\sum_{i=1}^{d}\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[V_i\log r(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi})]\right]. \tag{12}$$

Now we define r_i to be the terms in $\log r$ containing z_i and r_{-i} to be the remaining terms. Then the inner expectation in the gradient term is

$$\begin{aligned}&\sum_{i=1}^{d}\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[V_i(\log r_i(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi}) + \log r_{-i}(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi}))] \\ &\quad= \sum_{i=1}^{d}\mathbb{E}_{q(z_i\,|\,\boldsymbol{\lambda})}[V_i\mathbb{E}_{q(\mathbf{z}_{-i}\,|\,\boldsymbol{\lambda})}[\log r_i(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi}) + \log r_{-i}(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi})]], \\ &\quad= \sum_{i=1}^{d}\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[V_i\log r_i(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi})],\end{aligned}$$

where the last equality follows from the expectation of the score function of a distribution is zero. Substituting this back into Eq. 12 yields the desired result

$$\begin{aligned}&\mathbb{E}_{s(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}[V\log r(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi})]] \\ &= \mathbb{E}_{s(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\epsilon};\boldsymbol{\theta})\mathbb{E}_{q(\mathbf{z}\,|\,\boldsymbol{\lambda})}\left[\sum_{i=1}^{d}V_i\log r_i(\boldsymbol{\lambda}\,|\,\mathbf{z};\boldsymbol{\phi})\right]\right].\end{aligned}$$

Equality of Two Gradients. We now provide a direct connection between the score gradient and the reparameterization gradient. We carry this out in one-dimension for clarity, but the same principle holds in higher dimensions. Let Q be the cumulative distribution function (CDF) of q and let $z = T(\mathbf{z}_0; \boldsymbol{\lambda})$ be reparameterizable in terms of a uniform random variable $\mathbf{z}_0$ (inverse-CDF sampling). We focus on the one dimensional case for simplicity. Recall integration by parts computes a definite integral as

$$\int_{\text{supp}(\mathbf{z})} w(\mathbf{z}) \, \mathrm{d}v(\mathbf{z}) \\ = |w(\mathbf{z})v(\mathbf{z})|_{\text{supp}(\mathbf{z})} - \int_{\text{supp}(\mathbf{z})} v(\mathbf{z}) \, \mathrm{d}w(\mathbf{z}),$$

where the $| \cdot |$ notation indicates evaluation of a portion of the integral. In the subsequent derivation, we let $w(\mathbf{z}) = \log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z})$, and let $dv(\mathbf{z}) = \nabla_{\boldsymbol{\lambda}} \log q(\mathbf{z}) q(\mathbf{z}) = \nabla_{\boldsymbol{\lambda}} q(\mathbf{z})$.

Recall that we assume that we can CDF-transform $\mathbf{z}$ and that the transformation is differentiable. That is, when $\mathbf{u}$ is a standard uniform random variable, $\mathbf{z} = Q^{-1}(\mathbf{u}, \boldsymbol{\lambda})$. Then

$$\begin{aligned}
\nabla_{\boldsymbol{\lambda}}^{score} \mathcal{L} &= \mathbb{E}_{q(\mathbf{z} \mid \boldsymbol{\lambda})} [\nabla_{\boldsymbol{\lambda}} \log q(\mathbf{z} \mid \boldsymbol{\lambda})(\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid \boldsymbol{\lambda}))] \\
&= \int_{supp(\mathbf{z})} \nabla_{\boldsymbol{\lambda}} q(\mathbf{z} \mid \boldsymbol{\lambda})(\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid \boldsymbol{\lambda}))] \, \mathrm{d}\mathbf{z} \\
&= \left| \left[\int_{\mathbf{z}} \nabla_{\boldsymbol{\lambda}} q(\mathbf{z} \mid \boldsymbol{\lambda}) \, \mathrm{d}\mathbf{z} \right] (\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid \boldsymbol{\lambda})) \right|_{supp(\mathbf{z})} \\
&\quad - \int \left[\int_{\mathbf{z}} \nabla_{\boldsymbol{\lambda}} q(\mathbf{z} \mid \boldsymbol{\lambda}) \, \mathrm{d}\mathbf{z} \right] \nabla_{\mathbf{z}} [\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid \boldsymbol{\lambda})] \, \mathrm{d}\mathbf{z} \\
&= |\nabla_{\boldsymbol{\lambda}} Q(\mathbf{z} \mid \boldsymbol{\lambda})(\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid \boldsymbol{\lambda}))|_{supp(\mathbf{z})} \\
&\quad - \int \nabla_{\boldsymbol{\lambda}} \left[Q(\mathbf{z} \mid \boldsymbol{\lambda}) \right] \nabla_{\mathbf{z}} [\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid \boldsymbol{\lambda})] \, \mathrm{d}\mathbf{z} \\
&= |\nabla_{\boldsymbol{\lambda}} Q(\mathbf{z} \mid \boldsymbol{\lambda})(\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid \boldsymbol{\lambda}))|_{supp(\mathbf{z})} \\
&\quad + \int q(\mathbf{z} \mid \boldsymbol{\lambda}) \nabla_{\boldsymbol{\lambda}} \left[\mathbf{z} \right] \nabla_{\mathbf{z}} [\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid \boldsymbol{\lambda})] \, \mathrm{d}\mathbf{z} \\
&= |\nabla_{\boldsymbol{\lambda}} Q(\mathbf{z} \mid \boldsymbol{\lambda})(\log p(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z} \mid \boldsymbol{\lambda}))|_{supp(\mathbf{z})} \\
&\quad + \nabla_{\boldsymbol{\lambda}}^{rep} \mathcal{L},
\end{aligned}$$

where the second to last equality follows by the derivative of the CDF function (Hoffman and Blei, 2015). By looking at the Monte Carlo expression of both sides, we can see the reduction in variance that the reparameterization gradient has over the score gradient comes from the analytic computation of the gradient of the definite integral (which has value **0**).

Hyperparameters and Convergence. We study one, two, and three layer DEFs with 100, 30, and 15 units respectively and set prior hyperparameters following Ranganath et al. (2015). For HVMs, we use Nesterov's accelerated gradient with momentum parameter of 0.9, combined with RMSProp with a scaling factor of 10^{-3}, to maximize the lower bound. For the mean-field family, we use the learning rate hyperparameters from the original authors'. The HVMs converge faster on Poisson models relative to Bernoulli models. The one layer Poisson model was the fastest to infer.

Multi-level $q(\boldsymbol{\lambda}; \boldsymbol{\theta})$ and Optimizing with Discrete Variables in the Variational Prior. As mentioned in the main text Hierarchical variational models with multiple layers can contain both discrete and differentiable latent variables. Higher level differentiable variables follow directly from our derivation above. Discrete variables in the prior pose a difficulty due to high variance, as the learning signal contains the entire model. Local expectation gradients (Titsias, 2015) provide an efficient gradient estimator for variational approximations over discrete variables with small support—done by analytically marginalizing over each discrete variable individually. This approach can be combined with the gradient in Equation 8 of the main text to form an efficient gradient estimator.

In the setting where the prior has discrete variables, optimization requires a little more work. First we note that in a non-degenerate mean-field setup that the $\boldsymbol{\lambda}$'s are differentiable parameters of probability distributions. This means they will always, conditional on the discrete variables, be differentiable in the variational prior. This means that we can both compute the gradients for these parameters using the technique from above and that the discrete variables exist at a higher level of the hierarchical variational model; these discrete variables can be added to r conditional on everything else. The gradients of discrete variables can be computed using the score gradient, but Monte Carlo estimates of this will have high variance due to no simplification of the learning signal (like in the mean-field). We can step around this issue by using local expectation gradients (Titsias, 2015) which marginalize out one variable at a time to get low variance stochastic gradients. This is generally tractable when the discrete variables have small support such as the binary variables in the factorial mixture