Variational Inference with Stein Mixtures

Eric Nalisnick
Department of Computer Science
University of California, Irvine
enalisni@uci.edu

Padhraic Smyth
Department of Computer Science
University of California, Irvine
smyth@ics.uci.edu

1 Introduction

Obtaining uncertainty estimates is increasingly important in modern machine learning, especially as models are given an increasing amount of responsibility. Yet, as the tasks undertaken by automation become more complex, so do the models and accompanying inference strategies. In fact, exact inference is often impossible in practice for modern probabilistic models. Thus, performing *variational inference* (VI) [12] accurately and at scale has become essential. As posteriors are often complicated—exhibiting multiple modes, skew, correlations, symmetries—recent VI research has focused on using either implicit [20, 23, 18] or mixture approximations [10, 1, 7, 19, 9]. The former makes no strong parametric assumption, using either samples from a black-box function [20] or particles [16, 22] to represent the posterior, and the latter uses a convex combination of parametric forms as the variational distribution. While, in theory, these methods are expressive enough to capture any posterior, practical issues can prevent them from being a good approximation. For instance, the discrete approximation that makes implicit methods so flexible does not scale with dimensionality, as an exponential number of samples/particles are needed to cover the posterior. Mixtures, on the other hand, can scale well, but their optimization objective is problematic and requires using a lower bound to the *evidence lower bound* (ELBO) (i.e. a lower bound of a lower bound on the marginal likelihood) [10, 7, 21].

In this paper, we propose a scalable but nonetheless free-form variation inference method that performs Stein variational gradient descent (SVGD) [16] on *distributions*. Specifically, we describe how to perform SGVD on particles forming the second-level of a hierarchical variational model [21]. As the resulting marginal posterior is a mixture, we call the approximation a *Stein mixture*. Just as SVGD reduces to MAP estimation when using one particle, optimizing a Stein mixture with one component reduces to maximizing the ELBO. With more than one component, Stein mixtures can be optimized without having to resort to any lower bounds. In fact, given enough samples, the method performs gradient descent on a direct estimate of the marginal likelihood. Furthermore, we can leverage black-box inference models to create *amortized Stein mixtures*: mixture models without a fixed number of components. This allows for a small number of components to be used during training (for computational efficiency) but an unlimited number to be used for prediction. We demonstrate the method's effectiveness in both small and large scale experiments.

2 Background

Variational Inference. When we encounter an intractable posterior, we can make progress by proposing a distribution—call it $q(\boldsymbol{\theta}; \boldsymbol{\psi})$, with parameters $\boldsymbol{\psi}$—and fitting this distribution as an approximation to the true posterior. The optimization is usually done via maximization of the model evidence: $\log p(\mathbf{X}) = \log\ \mathbb{E}_q[p(\mathbf{X}, \boldsymbol{\theta})/q(\boldsymbol{\theta}; \boldsymbol{\psi})]$. Analytical evaluation of the expectation above is intractable but can be done by importance sampling [8, 2]. A more common approach is to optimize a lower bound, know as the *evidence lower bound* (ELBO) [12]. It can be obtained by straightforward application of Jensen's inequality to the expectation above: $\log p(\mathbf{X}) \geq \mathbb{E}_q[\log p(\mathbf{X}, \boldsymbol{\theta}) - \log q(\boldsymbol{\theta}; \boldsymbol{\psi})]$.

Advances in Approximate Bayesian Inference, NIPS 2017 Workshop, Long Beach, CA, USA.

Mixture Approximations. Using mixture distributions as approximations allows for the posterior's usual multi-modality to be captured and thus has been of interest from the genesis of variational Bayesian inference [10]. For a mixture approximation $q(\boldsymbol{\theta}) = \sum_k \pi_k q_k(\boldsymbol{\theta})$—where π_k denotes the weights and q_k denotes the component distributions—the ELBO is derived to be: $\log p(\mathbf{X}) \geq \sum_k \pi_k \mathbb{E}_{q_k}[\log p(\mathbf{X}, \boldsymbol{\theta})] + \mathbb{H}[q(\boldsymbol{\theta})]$. The first term is a weighted sum of expectations under each component and can be evaluated and optimized just as the similar term is for non-mixtures. However, the $\mathbb{H}[q(\boldsymbol{\theta})]$ term, the mixture model entropy, is intractable.

Previous work has used one of two strategies to deal with the entropy term. Jaakkola & Jordan (1998) [10] derive a lower bound to the ELBO and introduce auxiliary parameters that must be optimized to tighten the bound. Gershman et al. (2012) [7] use Jensen's inequality to derive the following bound: $\mathbb{H}[q(\boldsymbol{\theta})] \geq -\sum_k \pi_k \log \sum_j \pi_j \int_{\boldsymbol{\theta}} q_k(\boldsymbol{\theta}) q_j(\boldsymbol{\theta})\, d\boldsymbol{\theta}$. While this bound is appealing in that it does not introduce more parameters, it is almost always looser than the Jaakkola & Jordan bound.

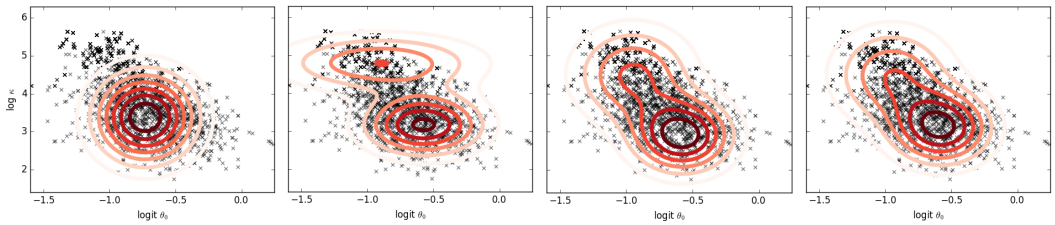
Stein Variational Gradient Descent Another appealing class of posterior approximations is particle methods [5, 22, 16]: the posterior is represented by a discrete set of points, i.e. $q(\theta) = \frac{1}{K}\sum_k \delta[\boldsymbol{\theta}_k]$. Particle methods are free of parametric assumptions and therefore are highly expressive. A recently proposed particle method is *Stein variational gradient descent* (SVGD) [16]. It exploits a connection between Stein's identity and the derivative of the Kullback-Leibler divergence (KLD) to efficiently transport the particles through iterative, deterministic updates. The update for the kth particle is given as $\boldsymbol{\theta}_k^{t+1} = \boldsymbol{\theta}_k^t + \alpha \nabla_{\boldsymbol{\theta}_k} \text{KLD}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{X})]$ where α is a learning rate and the KLD derivative is given by: $\nabla_{\boldsymbol{\theta}_k} \text{KLD}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{X})] = \frac{1}{K}\sum_{j=1}^K k(\boldsymbol{\theta}_j, \boldsymbol{\theta}_k) \nabla_{\boldsymbol{\theta}_j} \log p(\boldsymbol{\theta}_j|\mathbf{X}) + \nabla_{\boldsymbol{\theta}_j} k(\boldsymbol{\theta}_j, \boldsymbol{\theta}_k)$ where $k(\cdot,\cdot)$ is a proper kernel function. While the first term pulls particles towards the maxima of the log joint, the second term, $\nabla_{\boldsymbol{\theta}_j} k(\boldsymbol{\theta}_j, \boldsymbol{\theta}_k)$, encourages the particles to repulse one another, leading to beneficial posterior exploration. SVGD also has the nice property that reducing to one particle recovers MAP estimation, as the kernel terms become constants. Yet, like many particle methods, SVGD becomes problematic in high dimensions. The problem simply stems from the curse of dimensionality: the number of points required to adequately represent the posterior could be exponential in the dimension. In turn, using a large number of particles makes the $\mathcal{O}(K^2)$ nature of the updates computationally burdensome.

3 Stein Mixtures

We now turn to our proposed method: using SVGD to estimate a mixture approximation to the posterior, which we call a *Stein mixture* (SM). SMs combine the free-form nature of (implicit) particle approximations with mixtures, resulting in a method that better scales to high dimensions. By changing the operating objects from particles to *distributions*, we need fewer objects for a good approximation since each distribution can do the work of several particles. This relieves SVGD's K^2 computational costs with little loss in approximation quality as mixtures are still highly expressive. Moreover, as will be shown later in the section, the SGVB updates retain the nice simplification property: using one particle reduces to ELBO optimization.

Variational Model. We define the posterior approximation as the following hierarchical variational model [21]: $\boldsymbol{\psi} \sim q(\boldsymbol{\psi}),\ \boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\psi),\ q(\boldsymbol{\psi}|\mathbf{X}) = \frac{1}{K}\sum_{k=1}^K \delta[\boldsymbol{\psi}_k]$ where $q(\boldsymbol{\psi})$ and $q(\boldsymbol{\theta}|\psi)$ together form the variational model, with the former being the prior and the latter being the first-level distribution. $q(\boldsymbol{\psi}|\mathbf{X})$, then, is the posterior distribution over the second-level parameters and is made up of particles that will be optimized by SVGD. The marginal posterior is a mixture of the form $q(\boldsymbol{\theta}; \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_K) = \frac{1}{K}\sum_k q(\boldsymbol{\theta}|\boldsymbol{\psi}_k)$. Notice that this is a restricted mixture: each component has the same weight, which is a byproduct of SGVB using uniformly weighted particle distributions.

Update Derivation. To optimize the SM model, the SVGD update is derived as follows. The objective function is: $\text{KLD}\left[q(\boldsymbol{\psi}|\mathbf{X}) \,||\, p(\boldsymbol{\psi}|\mathbf{X})\right] = \text{KLD}[q(\boldsymbol{\psi}) \,||\, p(\mathbf{X}|\boldsymbol{\psi})q(\boldsymbol{\psi})/p(\mathbf{X})]$ where $q(\boldsymbol{\psi})$ and $q(\boldsymbol{\psi}|\mathbf{X})$ are as defined above and $p(\mathbf{X}|\boldsymbol{\psi}) = \mathbb{E}[p(\mathbf{X}, \boldsymbol{\theta})/q(\boldsymbol{\theta}|\boldsymbol{\psi})]$ is the marginal likelihood *as a function of the variational parameters*. While the marginal likelihood is often thought of as an expression of solely the generative model, for hierarchical approximations, the line between the generative and variational components becomes blurred [21]. Lastly $p(\mathbf{X})$ is the 'true' marginal likelihood with the variational parameters integrated out. The SVGD update is given by $\boldsymbol{\psi}_k^{t+1} =$



(a) $K = 1$ (b) $K = 2$ (c) $K = 3$ (d) $K = 4$

Figure 1: *Baseball Model Posterior Approximation*. The figures above show how the Stein mixture (red contours) improves with the addition of components (K) when approximating the posterior distribution of $\text{logit}(\theta_0)$ vs $\log \kappa$. The black points represent 2000 posterior samples drawn via `Stan`.

$\boldsymbol{\psi}_k^t + \alpha\, \boldsymbol{\phi}[\boldsymbol{\psi}_k^t]$ where $\boldsymbol{\phi}[\boldsymbol{\psi}]$ is defined as :

$$\frac{1}{K}\sum_{j=1}^{K} k(\boldsymbol{\psi}_j, \boldsymbol{\psi}) \nabla_{\boldsymbol{\psi}_j} \log \mathbb{E}[p(\mathbf{X}, \boldsymbol{\theta})/q(\boldsymbol{\theta}|\boldsymbol{\psi}_j)] \; + k(\boldsymbol{\psi}_j, \boldsymbol{\psi}) \nabla_{\boldsymbol{\psi}_j} \log q(\boldsymbol{\psi}_j) + \nabla_{\boldsymbol{\psi}_j} k(\boldsymbol{\psi}_j, \boldsymbol{\psi}). \quad (1)$$

Notice that the $\nabla_{\boldsymbol{\psi}_j} \log q(\boldsymbol{\psi}_j)$ term acts as regularization on the variational parameters, and from here forward we assume it is sufficiently weak and can be ignored.

Importance Weighted Gradients. One last but crucial detail to address about the update in Equation 1 is how to evaluate the $\nabla_{\boldsymbol{\psi}_j} \log \mathbb{E}[p(\mathbf{X}, \boldsymbol{\theta})/q(\boldsymbol{\theta}|\boldsymbol{\psi})]$ term. We employ two recent advances in variational inference: *differentiable non-centered parametrizations* (DNCPs) [14] and importance weighted Monte Carlo gradients [2]. The first calculates integrals by drawing differentiable samples: $\hat{\boldsymbol{\theta}} = g(\boldsymbol{\psi}, \hat{\boldsymbol{\xi}})$, $\boldsymbol{\xi} \sim p_0$, where $g(\cdot)$ is the differentiable function and $\hat{\boldsymbol{\xi}}$ is a sample from some fixed distribution p_0. The second technique is to use an importance weighted estimate of the marginal likelihood [2]. We draw S samples from the DNCP of our variational posterior and then calculate importance weights $w_s = p(\mathbf{X}, \hat{\boldsymbol{\theta}}_s)/q(\hat{\boldsymbol{\theta}}_s|\boldsymbol{\psi})$ for each sample $\hat{\boldsymbol{\theta}}_s$. The marginal likelihood's gradient estimator is then given as [2]: $\nabla_{\boldsymbol{\psi}_j} \log p(\mathbf{X}|\boldsymbol{\psi}_j) \approx \sum_s \tilde{w}_s \nabla_{\boldsymbol{\psi}_j} \log \frac{p(\mathbf{X}, \hat{\boldsymbol{\theta}}_s)}{q(\hat{\boldsymbol{\theta}}_s|\boldsymbol{\psi}_j)}$ where $\tilde{w}_s = w_s / \sum_{i=1}^{S} w_i$ is a normalized importance weight. Combining this equation with the general update in Equation 1 yields the final, black-box update:

$$\tilde{\boldsymbol{\phi}}[\boldsymbol{\psi}] = \frac{1}{K}\sum_{j=1}^{K} k(\boldsymbol{\psi}_j, \boldsymbol{\psi}) \sum_s \tilde{w}_s \nabla_{\boldsymbol{\psi}_j} \log \frac{p(\mathbf{X}, \hat{\boldsymbol{\theta}}_s)}{q(\hat{\boldsymbol{\theta}}_s|\boldsymbol{\psi}_j)} \; + \nabla_{\boldsymbol{\psi}_j} k(\boldsymbol{\psi}_j, \boldsymbol{\psi}). \quad (2)$$

Unfortunately, the importance weighted estimate of the marginal likelihood is biased downward [2]. Yet, this is not a significant problem since (1) the bias is strictly downward, guaranteeing we are optimizing an upper bound on $\text{KLD}[q(\boldsymbol{\psi}) \;||\; p(\mathbf{X}|\boldsymbol{\psi})q(\boldsymbol{\psi})/p(\mathbf{X})]$, and (2) the bias never increases for each additional sample and disappears as $S \to \infty$ [2].

Reduced Objectives. Just as SVGD reduces to MAP estimation when using one particle, Equation 2 also has satisfying behavior when reducing the number of particles (K) and the number of samples (S). Starting with the former, we see that when $K = 1$, the kernel terms become constants and the optimization objective reduces to an importance sampled estimate of the marginal likelihood—the same objective used in Burda et al. (2016). When $K = 1$ and $S = 1$, the objective reduces to a one-sample estimate of the ELBO. Unfortunately, it is hard to analyze SVGD's behavior when $K > 1$, and characterizing its solutions is an open problem [15].

Kernel Function. For the kernel function in Equation 2, any standard kernel can be used. However, small changes in some parameters may result in drastic changes in the variational distributions; for example, changes to a Gaussian's scale affects the distribution more fundamentally than a change in its mean. Thus, we use a kernel between probability distributions to naturally account for the information geometry. We use the *probability product kernel* [11], which has the form: $k(\boldsymbol{\psi}_j, \boldsymbol{\psi}_k; \rho) = \int_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{\psi}_j)^\rho p(\boldsymbol{\theta}|\boldsymbol{\psi}_k)^\rho \, d\boldsymbol{\theta}$ for two distributions $p(\boldsymbol{\theta}|\boldsymbol{\psi}_j)$ and $p(\boldsymbol{\theta}|\boldsymbol{\psi}_k)$ and where ρ is a hyperparameter. Note the kernel's similarity to Gershman et al. (2012)'s entropy bound.

Amortized Stein Mixtures. Notice that the SM model, in its particle assumption, does not require a parametric density for $\boldsymbol{\psi}$. Therefore, we can leverage *amortization* [6] as Feng et al. (2017) do for

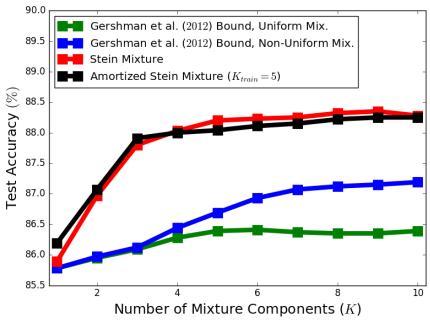


Figure 2: *IMDB Review Sentiment Classification*. Test set classification accuracy is reported for Stein (red) and Gershman et al. (2012) (green and blue) mixtures of size K. The black line denotes an amortized Stein mixture that used $K = 5$ during training but used the number of components according to the x-axis during testing. We see that both versions of the Stein mixture (red and black) out perform the alternative optimization scheme.

regular SVGD [4]: use a black-box function to generate the particles and update the shared function instead of the particles themselves. At each optimization step, we sample K particles from a neural network $f(\cdot)$ with parameters $\boldsymbol{\lambda}$ by passing in a random seed $\boldsymbol{\epsilon}_k$: $\hat{\boldsymbol{\psi}}_k = f(\hat{\boldsymbol{\epsilon}}_k; \boldsymbol{\lambda})$, $\boldsymbol{\epsilon}_k \sim p_0$. The neural network is then optimized to find fixed-points of the SVGD update: $\boldsymbol{\lambda}^* = \text{argmin}_{\boldsymbol{\lambda}} ||\tilde{\boldsymbol{\phi}}[\hat{\boldsymbol{\psi}}_k]||_2^2$. The number of samples K can be kept small at training time, but $f(\hat{\boldsymbol{\epsilon}}_k; \boldsymbol{\lambda})$ can be sampled from without bound when generating predictions.

4 Experiments

We demonstrate the effectiveness of Stein mixtures (SMs) on both small and large scale tasks. For all gradient updates, we used AdaM [13] with a cross-validated learning rate. For the probability product kernel, we used $\rho = 1$ for the sentiment classification task and manually tuned it for the baseball dataset experiment based on visual inspection of the posterior. We use Normal distributions for the components of all mixture approximations, performing transformations on bounded variables (such as $\text{logit}(x \in (0, 1))$) when necessary.

Baseball Model. Our first experiment is a small-scale sanity check in which we visualize the SM posterior in comparison to MCMC samples. We apply a hierarchical binomial regression model to the Efron-Morris baseball dataset [3], which consists of the number of hits y_i eighteen payers (indexed by i) obtained out of N_i at bats: $y_i \sim \text{Binomial}(N_i, \theta_i)$, $\theta_i \sim \text{Beta}(\phi \cdot \kappa, (1 - \phi) \cdot \kappa)$, $\kappa \sim \text{Pareto}(1, 1.5)$, $\phi \sim \text{Uniform}(0, 1)$. Figure 1 shows the SM approximation (red contours) plotted over 2000 MCMC samples (black points) from `Stan` for an increasing number of components (K). We see that, as expected, adding components gradually improves the posterior approximation.

Sentiment Classification. We next experimented on a large scale classification task, using the IMDB sentiment analysis dataset of Maas et al. (2011) [17], which contains 25,000 instances for both train and test splits. We reduced the vocabulary to the 20,000 most frequent words. We model the sentiment labels $y_i \in \{0, 1\}$ with a Bayesian logistic regression model of the form: $y_i \sim \text{Bernoulli}(f(\boldsymbol{\beta}\mathbf{x}_i))$, $\beta_d \sim \text{Normal}(0, 1/\tau)$, $\tau \sim \text{Gamma}(1, 0.1)$. We compare SMs for $K \in [1, 10]$ against two mixture approximations optimized via Gershman et al. (2012)'s ELBO lower bound (one with uniform weights, one with learned weights). We did not compare against Jaakkola & Jordan's (1998) bound because the auxiliary model would have added $80,000$ parameters. We also report results for an amortized SM using $K = 5$ during training. Test set accuracy is reported in Figure 2 via ensembling 100 posterior samples. We see that both SMs, amortized (black line) and non- (red line), out perform the Gershman et al. mixtures for all values of K. Interestingly, even though the SMs use uniform weights, they have higher accuracy than when the mixture weights are optimized (blue line). Furthermore, we see that the amortized SM performs better than the un-amortized version for $K < 5$ (as expected) and even performs similarly for $K > 5$ (albeit slightly worse).

5 Conclusions

We propose a variational inference method entitled *Stein mixtures* (SMs). SMs uniquely offer an approximation that is both scalable and expressive. Moreover, the mixture can be made dynamic, having a variable number of components, by incorporating an amortization model. Future work includes performing more experiments and analysis of the SM SVGD update for finite samples.

References

[1] Christopher M Bishop, Neil D Lawrence, Tommi Jaakkola, and Michael I Jordan. Approximating posterior distributions in belief networks using mixtures. In *Advances in neural information processing systems*, pages 416–422, 1998.

[2] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *International Conference on Learning Representations (ICLR)*, 2016.

[3] 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.

[4] Yihao Feng, Dilin Wang, and Qiang Liu. Learning to draw samples with amortized stein variational gradient descent. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2017.

[5] Andrew Frank, Padhraic Smyth, and Alexander T Ihler. Particle-based variational inference for continuous systems. In *Advances in Neural Information Processing Systems*, pages 826–834, 2009.

[6] Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the Cognitive Science Society*, volume 36, 2014.

[7] Samuel J Gershman, Matthew D Hoffman, and David M Blei. Nonparametric variational inference. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 235–242. Omnipress, 2012.

[8] Zoubin Ghahramani and Matthew J Beal. Variational inference for bayesian mixtures of factor analysers. In *Advances in neural information processing systems*, pages 449–455, 2000.

[9] Fangjian Guo, Xiangyu Wang, Kai Fan, Tamara Broderick, and David B Dunson. Boosting variational inference. *ArXiv Preprint*, 2016.

[10] 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.

[11] Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5(Jul):819–844, 2004.

[12] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2014.

[14] Diederik Kingma and Max Welling. Efficient gradient-based inference through transformations between bayes nets and neural nets. In *International Conference on Machine Learning*, pages 1782–1790, 2014.

[15] Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, 2017.

[16] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Neural Information Processing Systems (NIPS)*, 2016.

[17] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 142–150, 2011.

[18] Lars M. Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[19] Andrew C Miller, Nicholas Foti, and Ryan P Adams. Variational boosting: Iteratively refining posterior approximations. *Proceedings of the 34th International Conference on Machine Learning (ICML-17)*, 2017.

[20] Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator variational inference. In *Advances in Neural Information Processing Systems*, pages 496–504, 2016.

[21] Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

[22] Ardavan Saeedi, Tejas D. Kulkarni, Vikash K. Mansinghka, and Samuel J. Gershman. Variational particle approximations. *Journal of Machine Learning Research*, 18(69):1–29, 2017.

[23] Dustin Tran, Rajesh Ranganath, and David M Blei. Deep and hierarchical implicit models. *Advances in Neural Information Processing Systems*, 2017.