2nd Symposium on Advances in Approximate Bayesian Inference, 2019 1–11

Rapid Model Comparison by Amortizing Across Models

Lily H. Zhang LILYZHANG@ALUMNI.HARVARD.EDU
Indico Data Solutions

Michael C. Hughes MHUGHES@CS.TUFTS.EDU
Tufts University, Department of Computer Science

Abstract

Comparing the inferences of diverse candidate models is an essential part of model checking and escaping local optima. To enable efficient comparison, we introduce an amortized variational inference framework that can perform fast and reliable posterior estimation across models of the same architecture. Our Any Parameter Encoder (APE) extends the encoder neural network common in amortized inference to take both a data feature vector and a model parameter vector as input. APE thus reduces posterior inference across unseen data and models to a single forward pass. In experiments comparing candidate topic models for synthetic data and product reviews, our Any Parameter Encoder yields comparable posteriors to more expensive methods in far less time, especially when the encoder architecture is designed in model-aware fashion.

Keywords: Variational inference, amortized inference, model comparison, topic models

1. Introduction

We consider the problem of approximate Bayesian inference for latent variable models, such as topic models (Blei et al., 2003), embedding models (Mohamed et al., 2009), and dynamical systems models (Shumway and Stoffer, 1991). An important step in using such probabilistic models to extract insight from large datasets is model checking and comparison. While many types of comparison are possible (Gelman et al., 2013), we focus on a problem that we call within-model comparison. Given several candidate parameter vectors $\theta_1, \theta_2, \ldots$, all from the same space $\Theta \subseteq \mathbb{R}^D$, our goal is to efficiently determine which parameter θ_m is best at explaining a given dataset of N examples $\{x_n\}_{n=1}^N$.

Multiple ways exist to rank candidate parameters, including performance on heldout data or human-in-the-loop inspection. A principled choice is to select the parameter that maximizes the data's marginal likelihood: $\sum_{n=1}^N \log p(x_n|\theta_m)$. For our latent variable models of interest, computing this likelihood requires marginalizing over a hidden variable h_n: $p(x_n|\theta_m) = \int p(x_n|h_n, \theta_m)p(h_n|\theta_m)dh_n$. This integral is challenging even for a single example n and model m. One promising solution is variational inference (VI). Using VI, we can estimate an approximate posterior $q(h_n|x_n, \theta_m)$ over hidden variables. Approximate posteriors q can be used to compute lower bounds on marginal likelihood, and can also be helpful for human inspection of model insights and uncertainties. However, it is expensive to estimate a separate q at each example n and model m. In this paper, we develop new VI tools[1] that enable rapid-yet-effective within-model comparisons for large datasets.

1. Public Code Release: https://github.com/tufts-ml/any_parameter_encoder

The need for within-model comparison (and our methods) is present in many practical modeling tasks. Here we discuss two possible scenarios, with some details specialized to our intended topic modeling applications (Blei, 2012). First, in human-in-the-loop scenarios, a domain expert may inspect some estimated parameter θ and then suggest an alternative parameter θ' that improves interpretability. In topic modeling, this may mean removing "intruder words" to make topics more coherent (Chang et al., 2009). Second, in automated parameter learning scenarios, many algorithms propose data-driven transformations of the current solution θ into a new candidate θ', in order to escape the local optima common in non-convex optimization objectives for latent variable models (Roberts et al., 2016), Examples include split-merge proposal moves (Ueda and Ghahramani, 2002; Jain and Neal, 2004) or evolutionary algorithms (Sundararajan and Mengshoel, 2016). Across both these scenarios, new candidates θ' arise repeatedly over time, and estimating approximate posteriors for each is essential to assess fitness yet expensive to perform for large datasets.

Our contribution is the Any Parameter Encoder (APE), which amortizes posterior inference across models θ_m and data x_n. We are inspired by efforts to scale a single model to large datasets by using an encoder neural network (NN) to amortize posterior inference across data examples (Rezende et al., 2014; Kingma and Welling, 2014). Our key idea is that to additionally generalize across *models*, we feed model parameter vector θ_m and data feature vector x_n as input to the encoder. APE is applicable to any model with continuous hidden variables for which amortized inference is possible via the reparameterization trick.

2. Methods

We consider a general family of probabilistic models that use parameter vector θ_m to generate a dataset of N continuous hidden variables h_n and observations x_n via a factorized distribution: $\prod_{n=1}^{N} p(h_n|\theta_m)p(x_n|h_n, \theta_m)$. Our goal is fast-yet-accurate estimation of each example's local posterior $p(h_n|x_n, \theta_m)$ for a range of model parameters $\theta_1, \theta_2, \ldots \in \Theta$.

Topic Models. As a sample application, we focus on the Logistic Normal topic model from Srivastava and Sutton (2017). Given known vocabulary size V, we observe N documents represented by count vectors x_n (vector of size V counting the types of all T_n words in document n). We model each x_n as a mixture of K possible topics. Let hidden variable h_{nk} be the probability that a word in document n is produced by topic k. Thus, $h_n \in \Delta^K$ is a non-negative vector of size K that sums to one. We model h_n with a Logistic Normal prior, with mean and covariance set to be similar to a sparse Dirichlet(0.01) prior (Hennig et al., 2012) for interpretability. Given h_n, we model the observed word-count vector for document n with a Multinomial likelihood: $x_n \sim \text{Mult}(T_n, \sum_{k=1}^{K} h_{nk}\theta_k)$. This is a document-specific mixture of topics, where each topic k is defined by a word probability vector $\theta_k \in \Delta^V$. Our parameter of interest is the topic-word probability vector $\theta = \{\theta_k\}_{k=1}^{K}$.

VI Approximations for the Single Example Posterior. While the true posterior $p(h_n|x_n, \theta_m)$ is usually intractable, we use variational inference (VI) (Wainwright and Jordan, 2008) to approximate it. We choose a simpler density $q(h_n|\lambda_n)$ and optimize parameter λ_n to minimize KL divergence from the true posterior. Inference reduces to the well-known evidence lower bound (ELBO) optimization problem given example x_n and model θ_m:

$$\text{Inference:} \quad \lambda_n \leftarrow \arg\max_{\lambda_n} \mathbb{E}_q \left[\log p(x_n, h_n|\theta_m) - \log q(h_n|\lambda_n) \right], \tag{1}$$

Given several parameters of interest, we can perform model comparison by solving the above optimization problem separately for each θ_m. However, this is expensive. Solving Eq. (1) for a model θ_m requires dozens of iterative updates of gradient ascent for each example.

VI Amortized across Data Examples. Previously, Rezende et al. (2014) and Kingma and Welling (2014) have sped up inference by setting per-example variational parameters λ_n to the output of an encoder neural network (NN) instead of an iterative optimization procedure. The "Standard" encoder, with weights parameters ϕ, takes input data x_n and produces variational parameters $\lambda_\phi^{\text{NN}}(x_n)$. Inference for example n reduces to one fast forward pass: $\lambda_n \leftarrow \lambda_\phi^{\text{NN}}(x_n)$. While encoders often produce λ_n with worse ELBO scores than optimal solutions to Eq. (1) (Krishnan et al., 2018), they are preferred for their speed. However, for our model comparison goals the standard encoder is *expensive*, because for each parameter θ_m of interest we must train separate specialized NN weights ϕ_m.

Contribution: VI Amortized over Model Parameters. Our goal is to enable rapid estimation of posteriors $p(h_n|x_n, \theta_m)$ for many possible parameters $\theta_1, \theta_2, \ldots \in \Theta$ (not all known in advance). We thus consider a family of approximating densities q that explicitly conditions on *both* a given data vector x_n and the query parameter vector θ_m. Again, we use a neural network to transform these inputs into the variational parameters, $\lambda_n \leftarrow \lambda_\phi^{\text{NN}}(x_n, \theta_m)$. We call this the Any Parameter Encoder. Unlike the earlier Standard Encoder, which trains ϕ for one specific θ, our approach can directly generalize to many θ.

Encoder Architecture Design for Topic Models. Given the difficulty of posterior inference even for a single parameter θ, developing an effective Any Parameter Encoder requires careful selection of a NN architecture that can transform its two inputs, data x_n and model θ, to produce accurate approximate posteriors. Following previous work (Kingma and Welling, 2014), we use multi-layer perceptrons. We further suggest that an architecture designed to capture structure in the generative model should improve results further.

Our baseline "naive" architecture defines the input of the neural net as simply the concatenation of vector x_n and vector θ. While simple, we suggest this will be difficult to train effectively given the size of the input ($(K+1)V$ for the topic model) and lack of inductive bias to prioritize the model's needed interactions between entries of x_n and θ.

As an improvement, we consider a *model-aware* encoder architecture. Our design is motivated by a view of posterior inference as roughly moment-matching when data is plentiful. For our topic model, each document's Multinomial likelihood has a mean vector equal to $\sum_k h_{nk}\theta_k = \theta h_n$, writing θ as a $V \times K$ matrix. This mean vector should be (roughly) equal to the observed word-frequency vector $\frac{1}{T_n} x_n$. If μ_n is the mean of $q(h_n)$ and used as a plug-in estimate for h_n, then we want to satisfy $\frac{1}{T_n} x_n \approx \theta \mu_n$. Solving for μ_n via least squares, we get $\mu_n \approx \frac{1}{T_n}(\theta^T\theta)^{-1}\theta^T x_n$, which we might simplify to a non-linear function of $\theta^T x_n$. Thus, we suggest using the following model-aware encoder architecture:

$$q(h_n|x_n, \theta) = \text{LogisticNormal}(\mu_\phi^{\text{NN}}(\theta^T x_n), \text{diag}(\sigma_\phi^{\text{NN}}(\theta^T x_n))) \tag{2}$$

This model-aware architecture has encoder input dimension K, which is much smaller than $(K+1)V$ for the naive approach (and thus hopefully easier to train). Furthermore, this

	Synthetic Data ($K = 20$ topics)				Product Reviews ($K = 30$ topics)			
	Test Likelihood.	Time (sec.)	Agreement		Test Likelihood	Time (sec.)	Agreement	
Standard Encoder	-9.3672 ± 0.0436	0.01	26/45	58%	-6.1966 ± 0.0254	0.05	13/45	29%
APE Encoder (Naive Arch.)	-5.9508 ± 0.0170	0.11	28/45	62%	-5.4006 ± 0.0016	0.22	14/45	31%
APE Encoder (Model-aware Arch.)	-4.2384 ± 0.0029	0.04	43/45	96%	-5.2450 ± 0.0142	0.05	34/45	75%
VI (non-amortized)	-3.9047 ± 0.0023	5.09	45/45	100%	-4.9758 ± 0.0058	216.30	45/45	100%
NUTS	-4.0888 ± 0.0018	2044.91	-	-	-5.0803 ± 0.0008	23148.40	-	

Table 1: Comparison of per-word posterior predictive log likelihood (higher is better) and inference time (lower is better) for heldout test sets of 300 document-model combinations. 'Agreement' measures how often a method's ELBO ranking of pairs θ, θ' matches VI's ranking.

should provide desirable inductive bias to produce useful mean and covariance estimates. We emphasize that this design is specialized to the topic model, and further work is needed to develop model-aware architecture design strategies for general latent variable models.

Training the Encoder. Training our encoder parameters ϕ requires an available set of M parameter vectors $\{\theta_m\}_{m=1}^M$ of interest. We choose these to be representative of the subset of Θ we wish to generalize well to. We then maximize ELBO across all M models:

$$\max_\phi \mathbb{E}_q \left[\sum_{m=1}^M \sum_{n=1}^N \log p(x_n, h_n | \theta_m) - \log q(h_n | \lambda_\phi^{\text{NN}}(x_n, \theta_m)) \right] \tag{3}$$

We use stochastic gradient ascent to solve for ϕ, using the reparameterization trick to estimate gradients for a minibatch of examples and models at each step. We can interpret this objective as an expectation over samples θ_m from a target distribution over parameters.

Related Work. Recent meta-learning VAEs (Wu et al., 2019; Gordon et al., 2019) also try to generalize across models, but their encoders take sampled datasets from a model as input, not model parameters θ. Other work focuses on inverting the structure of the graphical model to improve amortization (Webb et al., 2018). Our approach offers a simpler, more direct approach to model comparison. For topic models, Yao et al. (2009) present early amortization efforts, and Srivastava and Sutton (2017) first used VAE approaches.

3. Experiments: Topic Models for Synthetic Data and Product Reviews

We compare our proposed Any Parameter Encoder (APE) to several other inference methods on two topic modeling tasks. For all VI methods, we choose q to be a Logistic Normal parameterized by a mean and a diagonal covariance. The appendix has complete details.

APE. We consider both naive and model-aware encoder architectures described above. Both use MLPs with 2 layers with 100 units per hidden layer, selected via grid search.

Baselines. We consider three baselines implemented in Pyro (Bingham et al., 2018) and PyTorch (Paszke et al., 2017). First, **Variational Inference (VI)** uses gradient ascent to optimize Eq. (1). Second, we use **Standard encoder** VAEs for topic models (Srivastava and Sutton, 2017). This encoder is specialized to a single parameter θ, with architecture size selected via grid search (similar to APE). Finally, we run Pyro's off-the-shelf implementation of Hamiltonian Monte Carlo with the No U-Turn Sampler (**NUTS**) (Hoffman and Gelman, 2014), though we expect specialized implementations to be more performant.

Synthetic Data Experiments. We consider a $V = 100$ vocabulary dataset inspired by the "toy bars" of Griffiths and Steyvers (2004). Using $K = 20$ true topics θ^*, we sample

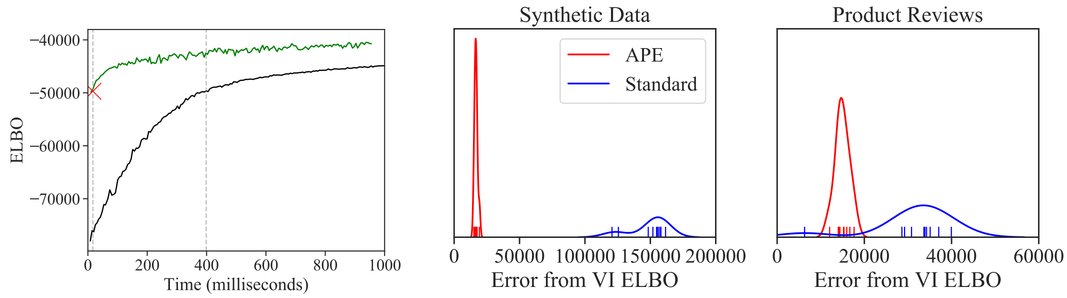


Figure 1: *Left:* ELBO vs. elapsed time for VI on a test set of 300 document-θ_m combinations on synthetic data. We show a randomly-initialized run (black) and a warm-started run (green) initialized via our Any-Parameter Encoder (red "X"). The randomly-initialized VI would require over 400 milliseconds (vertical line) to reach the quality our APE achieved in <20 ms. *Right:* Kernel Density Estimation of absolute difference between encoder ELBO and VI ELBO across different topics. APE results (red) are closer to VI (i.e. less error).

a 500-document dataset. We consider $M = 50,000$ possible model parameters $\{\theta_m\}_{m=1}^M$, sampled from a symmetric, sparse Dirichlet prior over the vocabulary. Typical θ_m look *unlike* the true topics θ^*, as shown in the supplement, so inference must handle diversity well. We train our APE on 25 million possible document-θ pairs for two epochs, then evaluate on unseen document-θ pairs drawn from the same generative process.

Product Reviews. We model 6,343 text documents of consumer product reviews (Blitzer et al., 2007). We use the $V = 3000$ most frequent vocabulary terms and $K = 30$ topics. We generate training topics in the same way as in the synthetic data experiments, and we evaluate on test topics found via Gibbs sampling with several separately initialized runs.

Results: Encoder Design. Results comparing naive and model-aware encoder architectures are in Table 1. Our proposed model-aware input layer yields better heldout likelihoods than the naive alternative, which we suggest is due to its more effective inductive bias.

Results: Quality-vs-Time Tradeoff. Comparing results across Table 1 and Fig. 1, we see that while the Standard Encoder understandably fails to generalize across models, our Any Parameter Encoder achieves quality close to non-amortized VI and NUTS with a speed up factor of over 100-1000x. APE can also provide a useful warm start initialization to VI.

Results: Agreement in model comparison. Motivated by the need to rapidly assess proposal moves that escape local optima, we gather 10 different models and measure whether each encoder's ranking of a pair θ, θ' on the test set agrees with VI's ranking. Table 1 shows that APE agrees with VI in 75% of 45 cases in the real data scenario, while Standard Encoder agrees just 29% of the time. This suggests APE may be trustworthy for accept/reject decisions, though further work is needed to improve this number further.

4. Conclusion

Across two datasets and many model parameters, our Any Parameter Encoder produces posterior approximations that are nearly as good as expensive VI, but over 100x faster. Future opportunities include simultaneous training of parameters and encoders, and handling Bayesian nonparametric models where θ changes size during training (Hughes et al., 2015).

Acknowledgments

This research supported in part by the National Science Foundation under awards ISS-1908617 and HDR-1934553

References

J. Aitchison and S. M. Shen. Logistic-Normal Distributions: Some Properties and Uses. *Biometrika*, 67(2):261–272, 1980. http://www.jstor.org/stable/2335470.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep Universal Probabilistic Programming. *ArXiv e-prints*, 2018. https://arxiv.org/abs/1810.09538.

David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

John Blitzer, Mark Dredze, Fernando Pereira, et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*, 2007.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems*, 2009.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC Press, 2013.

Jonathan Gordon, John Bronskill, Matthias Bauer, Sebastian Nowozin, and Richard Turner. Meta-Learning Probabilistic Inference for Prediction. In *International Conference on Learning Representations (ICLR)*, 2019. https://openreview.net/forum?id=HkxStoC5F7.

Tom L. Griffiths and Mark Steyvers. Finding Scientific Topics. *Proceedings of the National Academy of Sciences*, 2004.

Philipp Hennig, David H Stern, Ralf Herbrich, and Thore Graepel. Kernel Topic Models. In *Artificial Intelligence and Statistics*, 2012.

Matthew D Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, page 31, 2014. http://jmlr.org/papers/volume15/hoffman14a/hoffman14a.pdf.

Michael C. Hughes, Dae Il Kim, and Erik B. Sudderth. Reliable and Scalable Variational Inference for the Hierarchical Dirichlet Process. In *Artificial Intelligence and Statistics*, 2015. http://proceedings.mlr.press/v38/hughes15.html.

S. Jain and R.M. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182, 2004.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014. http://arxiv.org/abs/1312.6114.

Rahul G Krishnan, Dawen Liang, and Matthew D Hoffman. On the challenges of learning with inference networks on sparse, high-dimensional data. In *Artificial Intelligence and Statistics*, page 9, 2018. http://proceedings.mlr.press/v84/krishnan18a/krishnan18a.pdf.

Shakir Mohamed, Zoubin Ghahramani, and Katherine A Heller. Bayesian exponential family PCA. In *Advances in Neural Information Processing Systems 21*. 2009. URL http://papers.nips.cc/paper/3532-bayesian-exponential-family-pca.pdf.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. https://openreview.net/forum?id=BJJsrmfCZ.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning*, pages 1278–1286, 2014. http://proceedings.mlr.press/v32/rezende14.pdf.

Margaret E. Roberts, Brandon M. Stewart, and Dustin Tingley. Navigating the local modes of big data. *Computational Social Science*, 51, 2016. http://scholar.harvard.edu/files/dtingley/files/multimod.pdf.

R. H. Shumway and D. S. Stoffer. Dynamic linear models with switching. *Journal of the American Statistical Association*, 86(415):763–769, 1991.

Leslie N Smith. Cyclical learning rates for training neural networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

Akash Srivastava and Charles Sutton. Autoencoding Variational Inference For Topic Models. In *International Conference on Learning Representations*, 2017. https://openreview.net/pdf?id=BybtVK9lg.

Priya Krishnan Sundararajan and Ole J Mengshoel. A Genetic Algorithm for Learning Parameters in Bayesian Networks using Expectation Maximization. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, volume PMLR 52, 2016. http://proceedings.mlr.press/v52/sundararajan16.html.

N. Ueda and Z. Ghahramani. Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, 15(1):1223–1241, 2002.

Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008. ISSN 1935-8237, 1935-8245. https://people.eecs.berkeley.edu/~wainwrig/Papers/WaiJor08_FTML.pdf.

Stefan Webb, Adam Goliński, Robert Zinkov, N. Siddharth, Tom Rainforth, Yee Whye Teh, and Frank Wood. Faithful inversion of generative models for effective amortized inference. In *Neural Information Processing Systems*, 2018. URL https://papers.nips.cc/paper/7570-faithful-inversion-of-generative-models-for-effective-amortized-inference.

Mike Wu, Kristy Choi, Noah Goodman, and Stefano Ermon. Meta-Amortized Variational Inference and Learning. *ArXiv e-prints*, 2019. http://arxiv.org/abs/1902.01950.

Limin Yao, David Mimno, and Andrew McCallum. Efficient Methods for Topic Model Inference on Streaming Document Collections. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

Appendix A. Method Details

A.1. Topic Model Prior Details

Hidden variable h_{nk} defines the probability that any word in document n is produced by topic k. Thus, h_n is a non-negative vector of size K that sums to one. Commonly, h_n is given a symmetric Dirichlet prior with concentration $\alpha > 0$. Smaller α values lead to sampled vectors h_n with more sparsity. Instead, we model h_n with a Logistic Normal prior (Aitchison and Shen, 1980), and choose the mean and covariance to be close (in a KL sense) to a $\mathrm{Dir}(\alpha)$ prior, as in (Hennig et al., 2012)).

$$h_n \sim \mathrm{LogisticNormal}\left([\mu_1(\alpha) \ \ldots \ \mu_K(\alpha)], \mathrm{diag}([v_1(\alpha) \ \ldots \ v_K(\alpha)])\right) \tag{4}$$

$$\mu_k(\alpha) = \log \alpha_k - \frac{1}{K} \log \alpha_k, \quad v_k(\alpha) = \frac{1}{\alpha_k}(1 - \frac{2}{K}) + \frac{1}{K^2} \sum_k \frac{1}{\alpha_k}. \tag{5}$$

We set $\alpha_k = 0.01$ for all topics k. The choice of Logistic Normal prior enables the use of the reparameterization trick (Kingma and Welling, 2014), which produces unbiased and lower variance Monte Carlo estimation of the ELBO objective and its gradient with respect to encoder parameters.

We compare our Any-Parameter Encoder (APE) inference method to several baseline methods, all implemented for Logistic Normal topic models using Pyro (Bingham et al., 2018) with PyTorch backend (Paszke et al., 2017). We use CPU for all methods during evaluation for a consistent time comparison across methods.

A.2. Inference Implementation Details.

Approximate Posterior Family. For all variational methods, we chose approximate posterior q to belong to the Logistic Normal family parameterized by a mean vector $\mu_n \in \mathbb{R}^K$ and a diagonal covariance matrix with diagonal $\sigma_n^2 \in \mathbb{R}_+^K$.

$$q(h_n|\lambda_n{=}\{\mu_n, \sigma_n\}) = \mathrm{LogisticNormal}_K(\mu_n, \mathrm{diag}(\sigma_n^2)) \tag{6}$$

For encoder methods, the parameters $\{\mu_n, \log \sigma_n^2\}$ are the output of a shared encoder NN. For VI, these are free parameters of the optimization problem.

Variational Inference (VI). We perform using gradient ascent to maximize the objective in Eq. (1), learning a per-example mean and variance variational parameter. We run gradient updates until our moving average loss (window of 10 steps) has improved by less than 0.001% of its previous value. For our VI runs from random initializations, we use the Adam optimizer with an initial learning rate of .01, decaying the rate by 50% every 5000 steps. For our warm-started runs, we use an initial learning rate of 0.0005. In practice, we ran VI multiple times with different learning rate parameters and took the best one. Table 1 only reports the time to run the best setting, not the total time which includes various restarts.

Standard encoder. We use a standard encoder that closely matches the VAE for topic models in Srivastava and Sutton (2017). The only architectural difference is the addition of a temperature parameter on the μ_n vector before applying the softmax to ensure the means lie on the simplex. We found that the additional parameter sped up training by allowing the peakiness of the posterior to be directly tuned by a single parameter. We use a feedforward encoder with two hidden layers, each 100 units. We chose the architecture via hyperparameter sweeps. The total number of trainable parameters in the model is 24,721 on the synthetic data and 316,781 on the real data; this is compared to 16,721 and 19,781 parameters for model-aware APE.

NUTS. For the Hamiltonian Monte Carlo (HMC) with the No U-Turn Sampler (NUTS) (Hoffman and Gelman, 2014), we use a step size of 1 adapted during the warmup phase using Dual Averaging scheme. Upon inspection, we find that the method's slightly lower posterior predictive log likelihood relative to VI is due to its wider posteriors. We also find that the Pyro implementation is (understandably) quite slow and consequently warm-start the NUTS sampler using VI to encourage rapid mixing. We are aware that there exist faster, more specialized implementations, but we decided to keep our tooling consistent for scientific purposes.

Appendix B. Experiment Details

B.1. Synthetic Data Generation

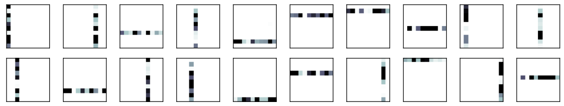
We generate a set of different models $\{\theta_0, \theta_1, ...\theta_M\}$ from a symmetric Dirichlet prior with $\alpha = 0.1$. We train our Any-Parameter Encoder in random batches of document-topic combinations. With 500 documents and 50,000 topics (i.e. $D = 500, M = 50,000$), we have 25 million combinations in total.

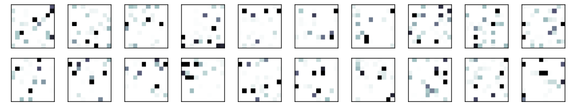
The topics used to generate the synthetic data represent "toy bars", inspired by (Griffiths and Steyvers, 2004). See Figure 2 for a visualization. We use this same toy bars-biased prior to generate all our topics in the holdout set, though the order of the topics is random. See Figures 2(a) and 2(b) for a visualization of the true topics θ^* and a representative model θ in the training set, respectively. Figure 2(c) shows sample reconstructions from each inference technique (rows) on different document-topic combinations (columns). The reconstructions are a qualitative assessment of the ability of the inference method to generate good posteriors and demonstrate the ability of APE to reach close-to comparable results with more expensive methods.

B.2. Encoder Training Details

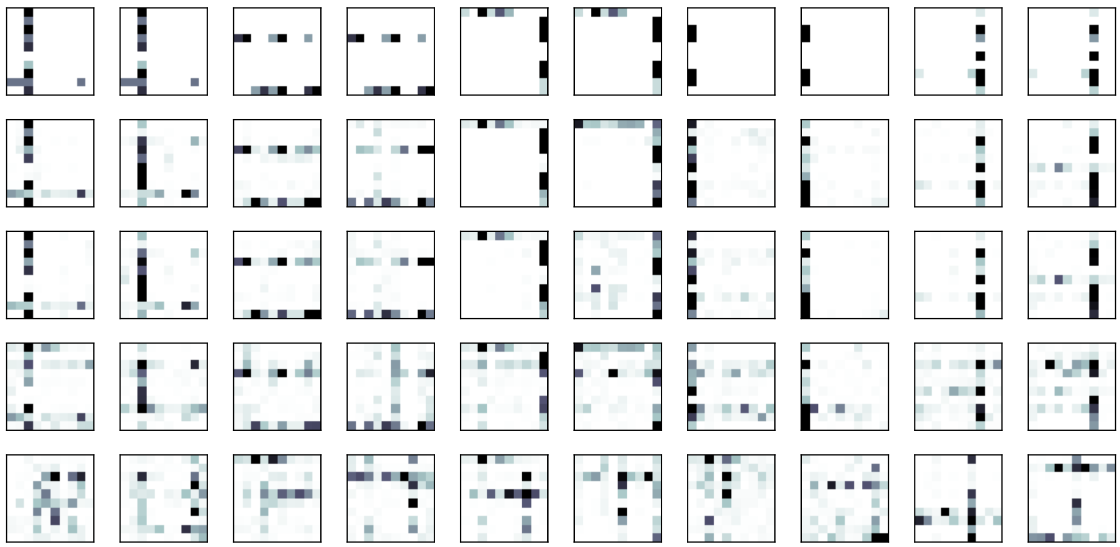
For training both APE and the Standard encoder on the synthetic data, we use Adam with an exponential decay learning schedule, a starting learning rate of 0.01, and a decay rate of .8 every 50,000 steps. We find that this schedule tends to be fairly robust; these hyperparameters were used for both APE and the Standard encoder on both the synthetic and real data. We chose our initial learning rate via a learning rate finder posed in Smith (2017), and we train for 2 epochs with a batch size of 100.

We train our standard VAE encoder on a single model with parameters θ drawn randomly from a symmetric Dirichlet prior with $\alpha = 0.1$. To train the standard encoder, we pass in our model of interest to the decoder, holding its weights fixed as we perform stochastic backpropagation to update the encoder weights. The same thing happens for APE, though the same topics are additionally included as part of the input into the encoder.





(a) True topics θ^* that generated the data. (b) Sample topics θ_m.



(c) Reconstructions of the documents. Rows (from top to bottom): Original document, No U-Turn Sampler, Variation inference, Any-Parameter encoder, Standard encoder. Columns are different document topic pairs. The Any-Parameter Encoder produces reconstructions close to comparable with the other more expensive procedures, while the Standard encoder fails to generalize to sensical posteriors.

Figure 2: Toy Bars Qualitative Visualizations. We represent document and the individual topics of a model by a 10x10, where each pixel in the plot represents a word in the vocabulary. The darkness of each pixel represents the density of the word (for topics) or the relative occurrence of the word (for documents). *(a)* and *(b)* show topics, and *(c)* shows documents and corresponding reconstructions from posteriors obtained through the various methods.

B.3. TLDR

We develop VAEs where the encoder takes a model parameter vector as additional input, so we can do rapid inference for many models