
Approximating exponential family models (not single distributions) with a two-network architecture

Sean R Bittner¹ John P Cunningham¹

Abstract

In most settings, the goal of deep generative modeling has been to find a *particular member* of that model family: optimized parameters index a distribution that is close to a target distribution. Much less attention, however, has been paid to learning the *model itself*. Here we introduce a two-network architecture (one of which is invertible) and optimization procedure for learning intractable exponential family models (*not* a single distribution from those models). These exponential families are learned accurately, allowing operations like posterior inference to be executed directly and generically with an input choice of natural parameters, rather than performing inference via optimization for each particular distribution within that model.

1. Introduction

Much recent work has focused on deep generative models, which map a latent random variable $w \sim q_0$ through a member of a highly expressive function family $\mathcal{G} = \{g_\theta : \theta \in \Theta\}$, the composition resulting in an implicit probability model $\mathcal{M} = \{q(g_\theta(w)) : \theta \in \Theta\}$. Choosing \mathcal{G} to be a parameter-indexed family of neural networks has both a rich history (Dayan et al., 1995; MacKay & Gibbs, 1997), and has recently been used to produce exciting results for density estimation (Uria et al., 2013; Rippel & Adams, 2013; Papamakarios et al., 2017), generation of complex data (Goodfellow et al., 2014), variational inference (Kingma & Welling, 2013; Rezende et al., 2014; Titsias & Lázaro-Gredilla, 2014), and more.

On the other hand, since these models have been chosen to be generic and flexible, they can lack the classic stipulation that a model instantiates existing domain knowledge (Gelman et al., 2014; Tenenbaum et al., 2006; McCullagh, 2002). There are well known drawbacks of fitting such flex-

ible models to finite (albeit large) data sets, which contrast with the bias-variance benefits that come from working in a restricted model space (Hastie et al., 2001, §7.3). Work on generalization and compressibility in deep networks suggests that this broad class of function families are indeed quite large, perhaps problematically so (Zhou et al., 2018).

When performing inference on a restricted model, it is increasingly common to deploy an implicit “recognition network” model for variational inference (Kingma & Welling, 2013), which finds a $q_{\theta^*}(z) \in \mathcal{M}$ such that an evidence bound is optimized with respect to the true posterior $p(z|X)$. However, it is widely understood that many such true posteriors $p(z|X)$ are exponential families (albeit intractable, due to the choice of sufficient statistics $t(z)$) of the form: $\mathcal{P} = \left\{ \frac{h(z)}{A(\eta)} \exp \{ \eta^\top t(z) \} : \eta \in H \right\}$ (Wainwright et al., 2008). Should we be able to learn a tractable approximation to this exponential family model, we would in the very least get the bias-variance benefits of an intelligently restricted model space, and at best would get inference “for free” in the sense that we could evaluate approximate posteriors directly without separate optimization for each dataset encountered (a novel form of *amortized inference* (Gershman & Goodman, 2014; Kingma & Welling, 2013; Rezende et al., 2014; Stuhlmüller et al., 2013)). In this paper, we aim to learn a restricted model $\mathcal{Q} = \{q(z; \eta) : \eta \in H\}$ that will be a strict subset of \mathcal{M} and will closely approximate a target exponential family \mathcal{P} . Note the critical difference between this aim and much of the literature that seeks to learn a density $q_\theta^* \in \mathcal{M}$ (we explore this distinction in depth both algorithmically and empirically). To proceed, we specify a set of invertible deep generative models $\mathcal{Q} = \{Q_\phi : \phi \in \Phi\}$, from which we can learn a single model Q_{ϕ^*} . We restrict Θ , the parameter space of \mathcal{M} , to be itself the image of a second deep *parameter network* family $\mathcal{F} = \{f_\phi : \phi \in \Phi\}$, such that $\{f_\phi(\eta) : \eta \in H\} \subset \Theta$.

We define this two-network architecture, which we term an *exponential family network* (EFN), and we specify a stochastic optimization procedure over an ELBO-like variant of the typical Kullback-Leibler divergence. We then demonstrate the ability of EFNs to approximately learn exponential families and the benefits of approximating distributions in such restricted model spaces. Finally we demonstrate the com-

¹Columbia University, New York City, USA. Correspondence to: Sean R Bittner <srb2201@columbia.edu>.

putational savings afforded by this approach when learning the posterior family of point-process latent intensities, given neural spike trains recorded in a neuroscience experiment.

2. Exponential family networks

To define exponential family networks (EFNs), we begin with relevant context for our modeling choice of exponential families (§2.1). We then describe the network architectural constraint and the background we use to satisfy that constraint (§2.2). We then introduce EFN in detail, including the optimization algorithm used for learning (§2.3). The similarities with variational inference are then explored in depth in (§2.4).

2.1. Exponential families as target model \mathcal{P}

We will focus on a fundamental problem setup in probabilistic inference, that of a latent variable $z \in \mathcal{Z}$ with prior belief $p_0(z)$, and where we observe a dataset $X = \{x_1, \dots, x_N\} \subset \mathcal{X}$ as conditionally independent draws given z . Updating our belief with data produces the posterior $p(z|X) \propto p_0(z) \prod_{i=1}^N p(x_i|z)$. This setup is shown as a graphical model in Figure 1A.

If we restrict our attention to priors and likelihoods that belong to exponential families $\mathcal{P} = \left\{ \frac{h(\cdot)}{A(\eta)} \exp \{ \eta^\top t(\cdot) \} : \eta \in H \right\}$, the posterior can also be viewed as an exponential family, albeit an intractable one (Wainwright et al., 2008). For simplicity we will hereafter suppress the base measure $h(\cdot)$. Consider:

$$p_0(z) = \frac{1}{A_0(\alpha)} \exp \{ \alpha^\top t_0(z) \} \quad (1)$$

$$p(x_i|z) = \frac{1}{A(z)} \exp \{ \nu(z)^\top t(x_i) \}, \quad (2)$$

where $t(\cdot)$ is the sufficient statistic vector, and $\nu(z)$ is the natural parameter of the likelihood in natural form (Robert, 2007). The posterior then has the form:

$$p(z|x_1, \dots, x_N) \propto \exp \left\{ \left[\begin{array}{c} \alpha \\ \sum_i t(x_i) \\ -N \end{array} \right]^\top \left[\begin{array}{c} t_0(z) \\ \nu(z) \\ \log A(z) \end{array} \right] \right\}, \quad (3)$$

which again is an intractable exponential family.

To give a concrete example, consider the hierarchical Dirichlet – a Dirichlet prior $z \sim \text{Dir}(\alpha)$ (of dimension $|\mathcal{Z}|$) with conditionally iid Dirichlet draws $x_i|z \sim \text{Dir}(\beta z)$ (MacKay & Peto, 1995; Teh et al., 2006; Blei et al., 2003; Pritchard et al., 2000). Figure 1B shows the prior for a given α (top), and three examples of datasets that could arise via this generative model (middle). A set of basic manipulations shows the hierarchical Dirichlet posterior $p(z|X)$ to be itself an exponential family with natural parameter $\eta = [\alpha - 1, \sum_i \log(x_i), -N]^\top$ and sufficient statistic

$t(z) = [\log(z), \beta z, \log(B(\beta z))]^\top$. The corresponding posteriors are shown in Figure 1B (bottom). Note importantly that, because the likelihood was chosen to be an exponential family (which is closed under sampling), this form will not change for any choice of $|\mathcal{Z}|$ -dimensional hierarchical Dirichlet – any draw from the prior, any N , or any particular realization of observed data X (technically the prior need not be exponential family, but we leave it as such for simplicity). The exponential family is clearly sufficient for this property, and the Pitman-Koopman Lemma further clarifies that it is also necessary (under reasonable conditions) (Robert, 2007, §3.3.3).

The critical observation here is that, if we can approximately learn an intractable exponential family (the model itself), then it becomes trivial to perform posterior inference: we simply use the dataset to index into the natural parameter η of the intractable family, and the posterior distribution is produced. This is the goal of EFNs.

2.2. Density networks as approximating family \mathcal{M}

Invertible deep generative models, which we will use for our approximating model family \mathcal{M} , can be defined by any base random variable $w \sim p_0$ mapped through any bijective, parameter-indexed function family $\mathcal{G} = \{g_\theta : \theta \in \Theta\}$, with induced density on $z = g_\theta(w)$ as $q_\theta(z)$. This is a well-established idea that has recently seen many variants and applications (MacKay & Gibbs, 1997; Baird et al., 2005; Tabak et al., 2010; Rippel & Adams, 2013; Uria et al., 2013; Rezende & Mohamed, 2015; Dinh et al., 2016; Papamakarios et al., 2017; Jacobsen et al., 2018). Specifically, let $z = g_\theta(w) = g_L \circ \dots \circ g_1(w)$ for bijective vector-valued functions g_ℓ (suppressing θ), and denote $J_\theta^\ell(z)$ as the Jacobian of the function g_ℓ at the layer activation corresponding to z . Then we have:

$$q_\theta(z) = q_0(g_1^{-1} \circ \dots \circ g_L^{-1}(z)) \prod_{\ell=1}^L \frac{1}{|J_\theta^\ell(z)|}. \quad (4)$$

The specific form of the layers g_ℓ can be chosen based on empirical considerations; we clarify our choice in §3. For the remainder (and to avoid confusion when we introduce a second network), we call this deep bijective neural architecture the *density network*; this network is shown vertically oriented (flowing from w down to z) in Figure 1C.

This density network induces the model $\mathcal{M} = \{q(g_\theta(w)) : \theta \in \Theta\}$, which previous work has searched to find a single optimized distribution q_{θ^*} (such as a posterior or data generative density), on the assumption and subsequent empirical evidence that the target exponential family member is close to (or approximately belongs to) \mathcal{M} . We make the same assumption for the exponential family itself and seek to intelligently restrict \mathcal{M} in order to learn the exponential family.

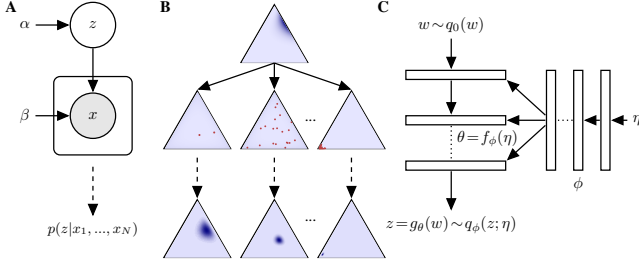


Figure 1. (A) Probabilistic graphical model. (B) Hierarchical Dirichlets: a Dirichlet prior with conditionally iid Dirichlet draws. (top) prior $p_0(z)$, (middle) three sample conditional Dirichlet datasets X of $N = 2, 20$, and 100 , and (bottom) the three posteriors that themselves belong to an exponential family \mathcal{P} . (C) Architecture for exponential family network (EFN): density network running top to bottom; parameter network right to left.

2.3. Exponential family networks as approximating model \mathcal{Q}

Having introduced our target model \mathcal{P} , an exponential family with natural parameters $\eta \in H$, and the density network family \mathcal{M} , we now seek to learn $\mathcal{Q} \approx \mathcal{P}$, where $\mathcal{Q} \subset \mathcal{M}$. To do so we will parameterize θ , the parameters of the density network, as the image of a second *parameter network* family $\mathcal{F} = \{f_\phi : H \rightarrow \Theta, \phi \in \Phi\}$. This network is shown flowing from right to left in Figure 1C. Using a second meta-network to aid or restrict network learning has been used in a variety of settings; a few examples include parameterizing the optimization algorithm in the “learning to learn” setting (Andrychowicz et al., 2016), and a more closely related work that used a second network to condition on observations for local latent variational inference (Rezende & Mohamed, 2015), a connection which we explore closely in Appendix A.

Any choice of parameter network parameters ϕ induces a $|H|$ -dimensional submanifold (the image $f_\phi(H)$) of the density network parameter space Θ , and as such defines a restricted model $\mathcal{Q}_\phi = \{q_{f_\phi}(z; \eta) : \eta \in H\} \subset \mathcal{M}$; by our choice of H as the natural parameter space of the exponential family target \mathcal{P} , this model restriction is at least of the correct dimensionality. Our goal then is to search over the implied set of models $\mathcal{Q} = \{\mathcal{Q}_\phi : \phi \in \Phi\}$ to find an optimal ϕ^* such that $\mathcal{Q}_{\phi^*} \approx \mathcal{P}$.

Given the connections between the exponential family and Shannon entropy, we will measure the error between \mathcal{Q}_ϕ and \mathcal{P} with Kullback-Leibler divergence. Consider for the moment a fixed choice of natural parameter η ; we seek to minimize, over ϕ :

$$D(q_\phi(z; \eta) || p(z; \eta)) = \mathbb{E}_{q_\phi} \left(\log q_\phi(z; \eta) - \eta^\top t(z) + \log(A(z)) \right) \quad (5)$$

which is equivalent to minimizing

$$\mathbb{E}_{q_\phi} \left(\log q_\phi(z; \eta) - \eta^\top t(z) \right) = \mathbb{E}_{q_\phi} \left(\log q_0(g_\theta^{-1}(z)) + \sum_{\ell=1}^L \log |J_\theta^\ell(z)| - \eta^\top t(z) \right), \quad (6)$$

where again we note that $\theta = f_\phi(\eta)$, and thus for a fixed η , this objective depends only on ϕ . Indeed, the target $\eta^\top t(z)$ is linear in η (an obvious restatement of the log-linear exponential family form), giving us some hope that we may be able to learn this model¹.

Of course we seek to approximate not just a single target exponential family member $p(z; \eta)$ for a fixed η , but rather the entire model $\mathcal{P} = \{p(z; \eta) : \eta \in H\}$. For optimization we thus need to introduce a distribution $p(\eta)$ (for stochastic optimization), leading to the objective:

$$\begin{aligned} \argmin_{\phi} \mathbb{E}_{p(\eta)} (D(q_\phi(z; \eta) || p(z; \eta))) \\ = \argmin_{\phi} D(q_\phi(z; \eta)p(\eta) || p(z; \eta)p(\eta)). \end{aligned} \quad (7)$$

Unbiased estimates of this objective are immediate. $q_\phi(z; \eta)$ is sampled by computing the density network parameters $\theta = f_\phi(\eta)$ (using the parameter network), sampling the latent $w \sim q_0(w)$, and running that w through the density network. $p(\eta)$ is user defined and chosen such that it is trivial to sample. Stochastic optimization can then be carried out on the estimator:

$$\begin{aligned} \mathbb{L}(\phi) = \frac{1}{K} \frac{1}{M} \sum_{k=1}^K \sum_{m=1}^M \left(\log q_0(g_{\theta^k}^{-1}(z^m)) \right. \\ \left. + \sum_{\ell=1}^L \log |J_{\theta^k}^\ell(z^m)| - \eta_k^\top t(z^m) \right), \end{aligned} \quad (8)$$

where $\theta^k = f_\phi(\eta_k)$. Successful optimization over ϕ should thus result in $\mathcal{Q}_{\phi^*} \in \mathcal{M}$ that accurately approximates the target exponential family; that is, $\mathcal{Q} \approx \mathcal{P}$. We call this two-network architecture and optimization an exponential family network (EFN). What remains for empirical implementation is to make particular choices of hyperparameters, network layers, and optimization algorithm, which we specify in §3 below.

3. Results

To investigate the performance of EFNs, we assess approximation fidelity on some tractable exponential families, examine the benefits of learning in a regularized model space, and characterize data analysis scenarios in which training an EFN is computationally advantageous. First, we test the

¹This objective can also produce approximations of the log partition (as the intercept term implied by this linear target), which we have found to be reasonably accurate, though nuanced schemes are likely appropriate (Papamakarios & Murray, 2015).

ability of EFNs to approximate the target model \mathcal{P} when this model is a known, tractable exponential family: this choice provides a simple ground truth and calibrates us to expected performance vs alternatives. Additionally, tractable exponential families allow us to measure the relative accuracy of single distribution approximations in isolation versus indexed members of trained EFNs. The main advantage of learning an EFN is to make tractable a previously intractable exponential family (at least approximately). This confers major benefits in terms of test-time: for example, rather than optimization needing to be run for variational inference with each particular dataset realized from a model class, EFN will allow immediate lookup. This benefit is orders of magnitude and is not instructive to view, so we show a decision boundary among neural data analysis scenarios, in which training an EFN is computationally advantageous to approximating several distributions through VI optimization individually. Most often, training an EFN has striking computational advantages.

To compare model approximations by EFNs to standard methodology, we alternatively train density networks to approximate members of the target model family. Since η will not change, we dispose of the parameter network and train the density network directly over θ (again with a deterministic choice of a single η). When the distribution being approximated is a posterior, this procedure is variational inference. This is the key comparison for the EFN model, and we refer to this alternative as NF for normalizing flow.

We also must make some particular architectural choices for these experiments. We considered a variety of density network architectures. For each exponential family, we searched through some candidate architectures which consisted of cascades of normalizing flow layers such as planar and radial flows introduced in (Rezende & Mohamed, 2015), a structured spinner flows inspired by (Bojarski et al., 2016), and a single affine transformation.

The parameter network was given tanh nonlinearities. In many of the results below we will analyze EFNs across a range of model dimensionality D (that is, $z \in \mathcal{Z} \subseteq \mathbb{R}^D$).

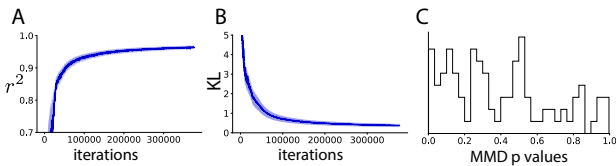


Figure 2. 50-dimensional Dirichlet exponential family network. (A) Distribution of r^2 between log density of EFN samples and ground truth across choices of η throughout optimization. (B) Distribution of KL divergence throughout optimization. (C) Distribution of maximum mean discrepancy p-values between EFN samples and ground truth after optimization.

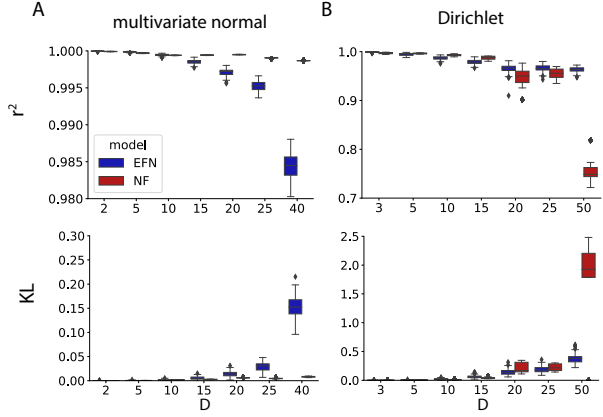


Figure 3. Scaling exponential family networks: D denotes the dimensionality of the family being learned, and comparisons are between EFN and its alternative NF (see text). (A) Multivariate normal family (B) Dirichlet family.

In all cases then we have also D flow layers in the density network (except when the affine transformation is optimal). In analyses where D was less than 20, 20 flow layers were used. The number of layers in the parameter network scaled as the square root of D , with a minimum of 4 layers, and the number of units per layer scaled linearly from the input to the number of density network parameters. Models were trained using the Adam optimizer algorithm (Kingma & Ba, 2014), with learning rates ranging from 10^{-3} to 10^{-5} . Optimizations ran for at least 50,000 iterations, and completed once there was a subthreshold increase in ELBO. These choices were made so that model performance saturated, and were held constant within comparative analyses.

All code was implemented in tensorflow, and is available at <https://github.com/cunningham-lab/efn>.

3.1. Tractable exponential families

Here we study the multivariate Gaussian and Dirichlet families, which offer a known ground truth and intuition about the range of performance that EFN – learning a model – has with respect to its single-distribution counterpart NF.

First, to validate the basic EFN approach, we train the $D = 50$ -dimensional Dirichlet family. We chose $p(\eta)$, the prior on the α parameter vector of the Dirichlet, as $\alpha_i \sim U[.5, 5.0]$. The number of η samples K at each iteration was 100, and the minibatch size in z was $M = 1000$. Figure 2 shows a high accuracy fit to this Dirichlet model: Figures 2A and 2B shows rapid convergence to high coefficient of determination r^2 and low Kullback-Leibler divergence. Since we are doing distribution regression, r^2 is a convenient metric calculated as the coefficient of determination between the model predictions $\log(q_\phi(z_i; \eta_k))$ and their known targets $\eta_k^\top t(z_i)$. We can then perform a

standard MMD-based kernel two-sample test (Gretton et al., 2012) between distributions chosen from \mathcal{P} and \mathcal{Q}_{ϕ^*} : the unstructured distribution of p values clarifies that the EFN model \mathcal{Q}_{ϕ^*} is not statistically significantly different than the true target Dirichlet family \mathcal{P} (using a test with 100 samples).

Second, in Figure 3 we consider how this performance scales across dimensionality. Consider EFN vs NF, where again the only difference is that EFN attempts to learn the entire model (as in $\eta \in H$), whereas NF chooses a single η and thus learns a single distribution optimizing the density network parameters θ directly. One might expect a noticeable deficit in approximation by EFNs, since they are generalizing the expressivity of the density network across $p(\eta)$. Accordingly, this deficit is apparent when modeling the multivariate normal family (Fig. 3A). In low dimensions, we have nearly exact model approximation by EFNs (blue) and distributional approximations by NFs (red). The distributions learned by NFs were drawn from the same η prior as the EFN was trained. However, as dimensionality increases EFN distributional approximations become significantly worse than the nearly perfect approximations learned by NFs. The η prior of the multivariate normal was specified as an isotropic normal on the mean parameter $\mu_i \sim \mathcal{N}(0, 0.1)$, and an inverse-Wishart distribution on the covariance $\Sigma \sim IW(n, \Psi)$ with degrees of freedom $n = 5$ and $\Psi = nDI$.

However, learning the model with EFN does not necessarily harm the distributional approximation relative to NF. In fact, conventional wisdom suggests that learning in a restricted model space is beneficial for regularization. Here, the expansiveness of the η prior determines the necessary degree of generalization of the EFN assigning a weight in the objective to the approximation loss of each distribution. By requiring the parameter network to learn generalizations of the density network across the η prior, local minima may be avoided that NFs would otherwise be susceptible to. This is in fact what we see when modeling the Dirichlet distribution (Fig. 3B). In low dimensions, NF performs better than EFN, but from 20 dimensions and greater, the restricted model space of the EFN confers superior optimization convergence relative to NF, which is more susceptible to local minima.

3.2. Lookup inference in an intractable exponential family

Of course the main interest of an EFN is to learn intractable exponential families. The Gaussian family is the ubiquitous prior for real valued parameters, but it does not match well with the nonnegativity requirements of the intensity measure required of certain distributions, most notably the Poisson. Log Gaussian Cox Processes have been used numerous times in machine learning, and all have required attention to

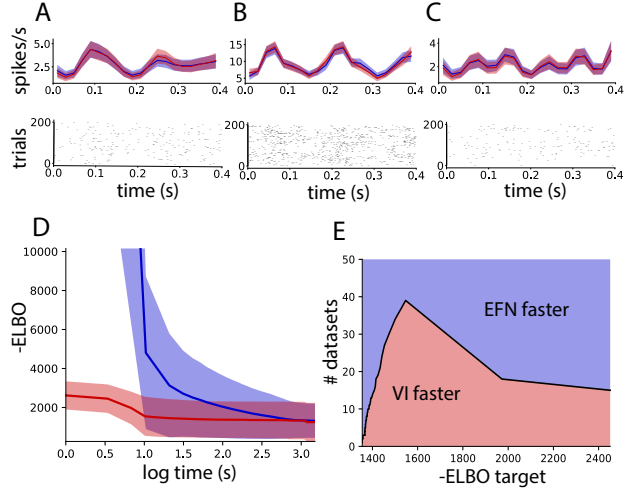


Figure 4. Lookup inference in a log-Gaussian Poisson model with V1 responses to drift grating stimuli. (A-C) Top: Inferred latent intensities from a single EFN (blue) or variational inference (red) run individually for each dataset. Shading denotes the standard deviation of the posterior. Bottom: Corresponding V1 spiking responses. (D) Distribution of -ELBO throughout training across a held out test group of 100 datasets for the EFN (blue), and across 298 datasets fit with NF (red). (E) Decision boundary for what number of datasets for a given target approximation accuracy it is advantageous to train an EFN rather than run variational inference individually for each dataset.

approximate inference in this fundamentally nonconjugate model. Furthermore, many of these examples have been used to analyze the latent firing intensity of neural spike train data (Cunningham et al., 2008b;a; Adams et al., 2009; Gao et al., 2016).

We demonstrate the utility of a log-Gaussian Poisson EFN for inferring latent firing intensities of neurons recorded in primary visual cortex of anesthetized macaques in response to 6.25 Hz drift grating stimuli (Smith & Kohn, 2008). 200 spike train responses were recorded for each neuron in response to 12 different grating orientations. Spiking responses were binned into 20ms intervals from 280ms-680ms following stimulus onset (to avoid the effects of transient neural dynamics). The frequency of the drift grating stimulus suggests a 25ms squared-exponential gaussian process prior on the 20-dimensional latent space, with mean and variance calculated across log firing rates of all neural responses (or "datasets"). Across three experimental subjects, 247 neurons with signal-to-noise ratios greater than 1.5 and mean firing rates greater than 1 Hz were considered, resulting in 2,964 total datasets for inferring latent intensities. By training an EFN on this log-Gaussian Poisson family, we have a model of the posterior distribution for this prior covariance, and some chosen spiking responses.

We can compare the posterior distribution learned with standard variational inference with NF (red) for a given neuron's response, to the posterior distribution we get with immediate lookup by supplying the spiking responses of a neuron and the chosen prior (the natural parameters of the posterior) as input to a trained EFN (blue) (Fig. 4A-C). As a reminder, NF is learning a single member of an exponential family. If that exponential family is in fact an intractable posterior distribution (such as the hierarchical Dirichlet or log-Gaussian Poisson examples already discussed), then indeed NF is *precisely* performing variational inference with a normalizing flow recognition network, as in (Rezende & Mohamed, 2015; Dinh et al., 2016; Papamakarios et al., 2017). Both EFNs and NFs were trained with 30 planar flow layers. These posteriors are very similar, and neither appears to fit the data better than the other. The high quality of these lookup posteriors is an incredible feat by the EFN. Now that we have trained this EFN, we have immediate posterior inference for all remaining and future neural recordings.

Training an EFN understandably takes more time than an NF (Fig. 4D), but once the EFN is trained we have immediate posterior inference lookup. If we have a target level of approximation (ELBO target) we can determine when it is faster to get posterior inference on a number of datasets by training an EFN and then using the immediate lookup feature or by running variational inference independently for each distribution. By computing the amount of computational time it takes to reach the ELBO target on average for both EFN and NF, and then counting how many datasets it would take to learn with NF before eclipsing the training time for the EFN. This results in a decision boundary (Fig. 4E), where an EFN is more computationally efficient for running posterior inference, and we have infinite computational savings for each additional dataset. As the ELBO target increases from its minimum value on the right of Fig. 4E, the extra time it takes an EFN to reach this ELBO target relative to NF increases initially. At some point, the EFN ELBO and NF ELBO distributions begin to converge (Fig. 4D), and the gap of time between learning an EFN and an NF for a given ELBO target begins to decrease. For some posterior distributions, the EFN learning approach may confer a mean ELBO greater than achievable by traditional variational inference due to the benefits of learning in a restricted model class. In the case where EFN achieves a greater mean ELBO than NF, it is always advantageous to use EFN.

Our ability to approximate an intractable exponential family model with an EFN is very encouraging. We have shown in the applied setting of inferring neural firing rates that learning the posterior inference model with an EFN can confer enormous computational savings. There is nothing unique about this application, insofar as we expect the power of learning exponential family models to translate to appli-

cations of intractable exponential family models in other settings. One can imagine downloading a pre-trained EFN for an intractable exponential family model, and being able to do posterior inference immediately given an arbitrary choice of prior and dataset.

4. Conclusion

We have approached the problem of learning an exponential family using a deep generative network, the parameters of which are the image of the natural parameters of the target exponential family under another deep neural network. We demonstrated high quality empirical performance across a range of dimensionalities, the potential for better approximations when learning in a restricted model space, and computational savings afforded by immediate posterior inference lookup.

References

- Adams, R. P., Murray, I., and MacKay, D. J. Tractable non-parametric bayesian inference in poisson processes with gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 9–16. ACM, 2009.
- Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., and de Freitas, N. Learning to learn by gradient descent by gradient descent. In *NIPS*, pp. 3981–3989, 2016.
- Baird, L., Smalenberger, D., and Ingkiriwang, S. One-step neural network inversion with pdf learning and emulation. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, volume 2, pp. 966–971. IEEE, 2005.
- Barber, D. and Wiergerinck, W. Tractable variational structures for approximating graphical models. In *NIPS*, pp. 183–189, 1999.
- Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022, 2003.
- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.
- Bojarski, M., Choromanska, A., Choromanski, K., Fagan, F., Gouy-Pailler, C., Morvan, A., Sakr, N., Sarlos, T., and Atif, J. Structured adaptive and random spinners for fast machine learning computations. *arXiv preprint arXiv:1610.06209*, 2016.

- Cunningham, J. P., Byron, M. Y., Shenoy, K. V., and Sahani, M. Inferring neural firing rates from spike trains using gaussian processes. In *NIPS*, pp. 329–336, 2008a.
- Cunningham, J. P., Shenoy, K. V., and Sahani, M. Fast gaussian process methods for point process intensity estimation. In *Proceedings of the 25th international conference on Machine learning*, pp. 192–199. ACM, 2008b.
- Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.
- Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.
- Gao, Y., Archer, E. W., Paninski, L., and Cunningham, J. P. Linear dynamical neural population models through nonlinear embeddings. In *NIPS*, pp. 163–171, 2016.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. *Bayesian data analysis*, volume 2. CRC press Boca Raton, FL, 2014.
- Gershman, S. and Goodman, N. Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 36, 2014.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *NIPS 27*, pp. 2672–2680. Curran Associates, Inc., 2014. URL <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>.
- Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.
- Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.
- Hoffman, M. and Blei, D. Stochastic structured variational inference. In *Artificial Intelligence and Statistics*, pp. 361–369, 2015.
- Jacobsen, J.-H., Smeulders, A., and Oyallon, E. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv*, 12 2013. URL <https://arxiv.org/pdf/1312.6114>.
- MacKay, D. J. and Peto, L. C. B. A hierarchical dirichlet language model. *Natural language engineering*, 1(3): 289–308, 1995.
- MacKay, D. J. C. and Gibbs, M. N. Density networks. In *Statistics and Neural Networks*, pp. 129–146. Oxford, 1997.
- McCullagh, P. What is a statistical model? *The Annals of Statistics*, 30(5):1225–1267, 2002. ISSN 00905364. URL <http://www.jstor.org/stable/1558705>.
- Papamakarios, G. and Murray, I. Distilling intractable generative models. In *Probabilistic Integration Workshop at Neural Information Processing Systems*, 2015.
- Papamakarios, G., Murray, I., and Pavlakou, T. Masked autoregressive flow for density estimation. In *NIPS*, pp. 2335–2344, 2017.
- Pritchard, J. K., Stephens, M., and Donnelly, P. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.
- Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.
- Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- Rippel, O. and Adams, R. P. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.
- Robert, C. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.
- Saul, L. K. and Jordan, M. I. Exploiting tractable substructures in intractable networks. In *NIPS*, pp. 486–492, 1996.
- Smith, M. A. and Kohn, A. Spatial and temporal scales of neuronal correlation in primary visual cortex. *Journal of Neuroscience*, 28(48):12591–12603, 2008.
- Stuhlmüller, A., Taylor, J., and Goodman, N. Learning stochastic inverses. In *NIPS*, pp. 3048–3056, 2013.
- Tabak, E. G., Vanden-Eijnden, E., et al. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

- Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006. doi: 10.1198/016214506000000302. URL <https://doi.org/10.1198/016214506000000302>.
- Tenenbaum, J. B., Griffiths, T. L., and Kemp, C. Theory-based bayesian models of inductive learning and reasoning. *Trends in cognitive sciences*, 10(7):309–318, 2006.
- Titsias, M. and Lázaro-Gredilla, M. Doubly stochastic variational bayes for non-conjugate inference. In *International Conference on Machine Learning*, pp. 1971–1979, 2014.
- Tran, D., Blei, D., and Airoldi, E. M. Copula variational inference. In *NIPS*, pp. 3564–3572, 2015.
- Uria, B., Murray, I., and Larochelle, H. Rnade: The real-valued neural autoregressive density-estimator. In *NIPS*, pp. 2175–2183, 2013.
- Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.
- Zhou, W., Veitch, V., Austern, M., Adams, R. P., and Orbanz, P. Compressibility and generalization in large-scale deep learning. *arXiv preprint arXiv:1804.05862*, 2018.

A. Relation to variational inference

A tremendous amount of work in recent years has gone into variational inference (VI), and its similarity to EFN warrants careful attention. In the following, we aim to carefully (and somewhat pedantically) dissect this question. As such, though EFN can address any target exponential family, to bring us closest to VI let us here restrict the EFN target model \mathcal{P} to be a family of posterior distributions (such as for example the log-Gaussian Poisson example in Section 4.2.)

The typical role of variational inference is to infer an approximate posterior $q_\phi(z) \approx p(z|X)$. In this setting, the difference with EFN is stark, in so much as VI learns this single posterior approximation, whereas the main goal of the EFN is to approximate the model $\mathcal{P} = p_\eta(z|X) : \eta \in H$: to learn the family of distributions. More recently, much focus has gone into the particular instance of VI for local variables z_i , for example $\prod_{i=1}^N p(z_i)p(x_i|z_i)$ (such as a variational autoencoder (Kingma & Welling, 2013)) or $p(u) \prod_{i=1}^N p(z_i|u)p(x_i|z_i)$ (latent Dirichlet allocation being a canonical example (Blei et al., 2003; 2017)), the result of which is often an amortized inference/recognition network that produces a local variational distribution $q_{\phi^*}(z_i|x_i)$.

This local variational distribution is typically parameterized explicitly: the inference network $\mu_\phi(x_i)$ induces a local parametric distribution, often a Gaussian $q(z_i|x_i) \sim \mathcal{N}(z_i; \mu_\phi(x_i))$ (Kingma & Welling, 2013, for example). Viewed this way, local-latent-variable VI methods induce a model $\{q_{\phi^*}(z_i|x_i) : x_i \in X\}$ for a finite dataset X . In that sense, EFN and VI are similar ‘model learning’ approaches. Even more closely, as part of a long-standing desire to add structure to VI beyond mean-field (classically (Saul & Jordan, 1996; Barber & Wiergerinck, 1999); more recently (Hoffman & Blei, 2015; Tran et al., 2015), to name but a few), in several cases an inference network has been used to parameterize a deep implicit model (in a two-network inference architecture, to say nothing of whether or not the generative model itself is a deep generative model); closest to the EFN architecture is (Rezende & Mohamed, 2015) (cf. Figure 2 of (Rezende & Mohamed, 2015) with Figure 1C here). Thus EFN (when used for posterior families) can be seen as a close generalization of VI.

Even accepting this VI-as-a-model view, the difference between the finite dataset X and the natural parameter space H persists when viewed at a mechanical level; well-known are the overfitting/generalization issues associated with a finite dataset compared with access to a distribution $p(\eta)$. Thus one goal of EFN is to allow the model $\mathcal{Q}_{\phi^*} \approx \mathcal{P}$ to be learned in the absence of a finite dataset, such that inference on that dataset can then be executed without concerns of overfitting to that set (and of course without having to run a VI optimization for every new dataset; we will demonstrate this benefit of EFN in the experiments). Perhaps more importantly, the “model” implied by VI is parameterized by x_i , and indeed the inference network takes x_i as input. The EFN on the other hand is considerably more general; the posterior includes the natural parameters of the prior (Equation 3). This allows the EFN architecture to learn across a more general setting that VI cannot, since any VI inference network is only parameterized by data. One final difference made clear by Equation 3 is that the observations are given to the EFN *in natural form* (that is, $t(x_i)$, not x_i) (Robert, 2007). This choice is a novel insight: by exploiting the known sufficiency of $t(x_i)$ in the target model \mathcal{P} , some difference in performance for VI may be observed. Accordingly, while EFN and VI do at a high level bear multiple similarities, the differences are both material and provoke interesting speculation about means to improve both VI and EFN.