

---

# Variational Inference with Continuously-Indexed Normalizing Flows

---

Anthony Caterini<sup>1</sup> Rob Cornish<sup>1</sup> Dino Sejdinovic<sup>1</sup> Arnaud Doucet<sup>1</sup>

## Abstract

Continuously-indexed flows (CIFs) have recently achieved improvements over baseline normalizing flows in a variety of density estimation tasks. In this paper, we adapt CIFs to the task of variational inference (VI) through the framework of auxiliary VI, and demonstrate that the advantages of CIFs over baseline flows can also translate to the VI setting for both sampling from posteriors with complicated topology and performing maximum likelihood estimation in latent-variable models.

## 1 Introduction

Variational inference (VI) has emerged as a fast alternative to Markov chain Monte Carlo for Bayesian inference, and as a method to facilitate maximum likelihood optimization of probabilistic latent variable models. VI approaches approximate a target posterior by minimizing the KL divergence over a parametrized family of distributions. The expressiveness of this family is essential for good performance, with under-expressive models leading to increased bias and underestimation of posterior variance (Yin & Zhou, 2018).

If the density of the approximate posterior is available in closed-form, then the variational family is said to be *explicit*. Explicit models allow for straightforward estimation of the VI objective, but can often lead to reduced expressiveness which limits their performance overall. Mean-field VI (Blei et al., 2017), for example, imposes restrictive independence assumptions between the variables of interest. Normalizing flows (Rezende & Mohamed, 2015) can be used to improve the expressiveness of underlying mean-field distributional families, but these too can have limitations, particularly when considering targets with complex topological structure (Cornish et al., 2020).

To improve expressiveness, it is therefore natural to consider *implicit* variational methods, which do not require the

---

<sup>1</sup>University of Oxford, Oxford, United Kingdom. Correspondence to: Anthony Caterini <anthony.caterini@stats.ox.ac.uk>.

approximate posterior to be available in closed form and so offer greater freedom and flexibility in specifying a variational family. Implicit models have achieved impressive results in a variety of tasks including VI (Tran et al., 2017), as well as generative modelling (Brock et al., 2018) and Bayesian experimental design (Kleinegesse et al., 2020). These models are often well-motivated by, and frequently used for, various applications within the physical sciences (Tran et al., 2017; Kleinegesse et al., 2020).

In many situations, implicit methods require the estimation of density ratios (Mohamed & Lakshminarayanan, 2016), which becomes increasingly difficult in higher dimensions (Sugiyama et al., 2012). We can circumvent these difficulties within the context of VI using the framework of *auxiliary variational inference* (AVI) (Agakov & Barber, 2004), in which the variational posterior is still defined as an intractable marginalization, but now over a tractable joint distribution. AVI methods have demonstrated improved expressiveness over their explicit counterparts in a number of settings (Burda et al., 2016; Yin & Zhou, 2018; Lawson et al., 2019), while skirting many of the difficulties associated with other types of implicit models.

In this work, we explore a novel approach to AVI which uses continuously-indexed flows (CIFs) (Cornish et al., 2020) to define the approximate posterior. CIFs, which were originally proposed for density estimation, introduce auxiliary variables to relax the constraint of bijectivity imposed by standard normalizing flows. We describe how CIFs can be used as the variational family in an AVI framework, and empirically demonstrate the advantages of using CIFs over normalizing flows for both modelling posteriors with complicated topologies and performing maximum likelihood estimation in generative modelling.

## 2 CIFs for Auxiliary Variational Inference

### 2.1 Background

Given a joint probabilistic model  $p_{X,Z}(x, z)$ , with data  $x \in \mathcal{X}$  and latent variable  $z \in \mathcal{Z}$ , variational inference (VI) provides us with a means to both approximate the intractable posterior  $p_{Z|X}(z|x)$  and maximize the marginal likelihood  $p_X(x) = \int p_{X,Z}(x, z) dz$  with respect to the parameters of the joint model. This is accomplished by introducing a

parametrized approximate posterior<sup>1</sup>  $q_Z(z)$  and maximizing the Evidence Lower Bound (ELBO)

$$\mathcal{L}_1(x) := \mathbb{E}_{q_Z} \left[ \log \frac{p_{X,Z}(x, z)}{q_Z(z)} \right] \leq \log p_X(x). \quad (1)$$

For a fixed model  $p_{X,Z}$ , this is equivalent to minimizing the KL divergence between  $q_Z$  and the true posterior  $p_{Z|X}(\cdot|x)$ .

To gain expressiveness over explicit methods, we can use an *implicit* distribution  $q_Z(z) = \int q_{Z,U}(z, u) du$  as the approximate posterior, where  $u \in \mathcal{U}$  are *auxiliary* variables. If  $q_{Z,U}$  can be sampled from and evaluated pointwise, our VI method falls within the framework of *auxiliary variational inference* (AVI). Using auxiliary variables allows for the specification of more complex models but loses the tractability of  $\mathcal{L}_1$ . However, we can tractably lower-bound (1) as per e.g. Agakov & Barber (2004); Lawson et al. (2019), arriving at the objective

$$\mathcal{L}_2(x) := \mathbb{E}_{q_{Z,U}} \left[ \log \frac{p_{X,Z}(x, z) \cdot r_{U|Z}(u|z)}{q_{Z,U}(z, u)} \right], \quad (2)$$

where  $r_{U|Z}(u|z)$  is an *auxiliary* inference distribution which can be evaluated pointwise. It is straightforward to show

$$\mathcal{L}_1(x) = \mathcal{L}_2(x) + \mathbb{E}_{q_Z} [D_{\text{KL}}(q_{U|Z}(\cdot|z) \parallel r_{U|Z}(\cdot|z))], \quad (3)$$

which suggests that  $r_{U|Z}$  should be expressive enough to closely approximate the typically intractable  $q_{U|Z}$ .

## 2.2 CIFs in this Framework

We now show how continuously-indexed flows (CIFs) (Cornish et al., 2020) can be used to define an implicit variational posterior over auxiliary variables. Given  $G : \mathcal{Z} \times \mathcal{U} \rightarrow \mathcal{Z}$  such that  $G(\cdot; u)$  is a bijection for each  $u \in \mathcal{U}$ , as well as densities  $q_W$  and  $q_{U|W}$ , the CIF generative process

$$W \sim q_W, \quad U \sim q_{U|W}(\cdot|W), \quad Z = G(W; U), \quad (4)$$

admits the following joint density over  $\mathcal{Z} \times \mathcal{U}$ :

$$\begin{aligned} q_{Z,U}(z, u) &= q_W(G^{-1}(z; u)) \\ &\times q_{U|W}(u|G^{-1}(z; u)) |\det D_z G^{-1}(z; u)|. \end{aligned} \quad (5)$$

Here  $W$  typically originates from a mean-field density.

Given the sampling process (4) and the resulting joint distribution (5), we can rewrite the objective (2) as

$$\mathcal{L}_C(x) := \mathbb{E}_{q_{W,U}} \left[ \log \frac{p_{X,Z}(x, G(w; u)) \cdot r_{U|Z}(u|G(w; u))}{q_W(w) \cdot q_{U|W}(u|w) \cdot |\det D_w G(w; u)|^{-1}} \right], \quad (6)$$

<sup>1</sup>We may also *amortize*  $q_Z$  and replace it with the conditional  $q_{Z|X}$  when considering the case of generative modelling.

where  $q_{W,U}(w, u) := q_W(w) \cdot q_{U|W}(u|w)$ , and the auxiliary inference model  $r_{U|Z}$  is another parametrized density.<sup>2</sup> Assuming  $q_{W,U}$  can be reparametrized (Kingma & Welling, 2014), we can optimize this objective with respect to the parameters of  $p, q, r$ , and  $G$  via stochastic gradient descent.

Many design choices used by Cornish et al. (2020) can be applied out-of-the-box in this context. In particular, we can take both  $q_{U|W}$  and  $r_{U|Z}$  to be conditionally Gaussian with parametrized mean and covariance, and set

$$G(w; u) := e^{s(u)} \odot g(w) + t(u), \quad (7)$$

where  $g : \mathcal{Z} \rightarrow \mathcal{Z}$  is a base bijection,  $s, t : \mathcal{U} \rightarrow \mathcal{Z}$  are unrestricted neural networks, and  $\odot$  denotes elementwise multiplication. We will see that this choice of indexed bijection can mitigate scaling pathologies in certain VI problems, and indeed generalizes the BatchNorm transformation (Ioffe & Szegedy, 2015) often used to stabilize the training of normalizing flows (Dinh et al., 2017). One difference is that the VI setting requires taking expectations with respect to  $q_W$ , whereas in density estimation the corresponding term only needs to be evaluated pointwise.

Like Cornish et al. (2020), we can also stack the generative process (4) to gain a more expressive multi-layer model. We can view this multi-layer model as an instance of (4) for certain choices of  $q_{U|W}$  and  $G$ , and as such we focus on the single-layer case in the exposition. Both of these points are further expounded in Appendix A. Of particular note is Algorithm 1, which describes how to compute an unbiased estimator of the multi-layer objective from which we can then obtain unbiased gradients via automatic differentiation.

## 3 Comparison to Related Work

### 3.1 Normalizing Flows for VI

Normalizing flows (NFs) were originally constructed as a method for increasing the expressiveness of approximate posteriors in VI (Rezende & Mohamed, 2015). NF methods define  $q_Z$  via the generative process

$$W \sim q_W, \quad Z = g(W), \quad (8)$$

where  $g : \mathcal{Z} \rightarrow \mathcal{Z}$  is a bijection. Using the change of variable formula, we can rewrite (1) for this model as

$$\mathcal{L}_N(x) = \mathbb{E}_{q_W} \left[ \log \frac{p_{X,Z}(x, g(w))}{q_W(w) \cdot |\det Dg(w)|^{-1}} \right]. \quad (9)$$

When  $G$  is independent of  $u$ , the CIF ELBO (6) indeed reduces to (9).<sup>3</sup> This shows that CIFs generalize NFs not only in the density estimation setting, but also in VI.

<sup>2</sup>Again, we may wish to use amortization, in which case  $q_W$  and  $r_{U|Z}$  would be replaced by  $q_{W|X}$  and  $r_{U|Z,X}$  throughout, respectively. Further details can be found in Appendix A.

<sup>3</sup>See Appendix B for a discussion.

Explicit NFs provide a framework for increasing the expressiveness of any trainable explicit density model. However, the bijectivity constraint can lead to problems when modelling a density that is concentrated on a region with complicated topological structure (Cornish et al., 2020, Corollary 2.2), and may cause flows to become numerically non-invertible in this case (Behrmann et al., 2020). Many models such as neural spline flows (NSFs) (Durkan et al., 2019) and *universal* flows (Huang et al., 2018; Jaini et al., 2019) have been proposed to improve expressiveness within the standard framework based on a single bijection. CIFs, on the other hand, use auxiliary variables to provide a mechanism for circumventing the limitations of using a single bijection, but lose analytical tractability as a result.

### 3.2 Implicit VI Methods

CIFs fall within the class of auxiliary VI models (Agakov & Barber, 2004), which includes methods such as importance-weighted auto-encoders (Burda et al., 2016), hierarchical variational models (Ranganath et al., 2016), and semi-implicit VI (Yin & Zhou, 2018). CIFs are distinguished from these approaches by their use of parametrized bijections to improve the inference procedure. To the best of our knowledge, the only other auxiliary VI methods relying on bijections are Hamiltonian-based (Salimans et al., 2015; Caterini et al., 2018), although these models are instead bijective over the extended space  $\mathcal{Z} \times \mathcal{U}$ , and have greatly increasing computational requirements as the number of parameters in  $p_{X,Z}$  grows since they require  $D_z \log p_{X,Z}(x, z)$  at every forward flow step.

A separate class of implicit VI models proposes expressive but intractable joint densities which require density ratio estimation to train (Huszár, 2017; Tran et al., 2017). CIFs, along with other AVI methods, avoid density ratio estimation by instead constructing a tractable joint density.

### 3.3 CIFs for Density Estimation

The inference process required to train CIFs for density estimation is very closely related to the generative process (4). In particular, if we index the forward CIF model for density estimation by  $r$  (instead of  $p$  used by Cornish et al. (2020)), the ELBO objective for a single-layer model becomes<sup>4</sup>

$$\mathbb{E}_{q_{X,U}} \left[ \log \frac{r_Z(G(x; u)) \cdot r_{U|Z}(u|G(x; u))}{q_X^*(x) \cdot q_{U|X}(u|x) \cdot |\det D_x G(x; u)|^{-1}} \right], \quad (10)$$

where  $q_X^*$  is the unknown data-generating distribution from which we have i.i.d. samples, and  $q_{X,U}(x, u) := q_X^*(x) \cdot q_{U|X}(u|x)$ . Comparing this with (6), we see that CIFs for density estimation may be interpreted as performing AVI targeting  $r_Z$  with an inference model having generative

process

$$X \sim q_X^*, \quad U \sim q_{U|X}(\cdot|X), \quad Z = G(X; U). \quad (11)$$

## 4 Experiments

In this section, we investigate the use of CIFs in both posterior sampling and maximum likelihood estimation of generative models. We compare inference models based on the auto-regressive variant of the Neural Spline Flow (NSF) (Durkan et al., 2019) to CIF-based extensions. NSFs are a good choice of baseline because they empirically provide good performance in general-purpose density estimation. Throughout, we use the ADAM optimizer (Kingma & Ba, 2015). Hyperparameters for all experiments are available in Appendix D. Code will be made available at <https://github.com/anthonycaterini/cif-vi>.

### 4.1 Toy Mixture of Gaussians

Our first example looks at using VI to sample from a toy mixture of Gaussians, similar to the first target considered in Duan (2019). Given component means  $\{\mu_k\}_k$  and covariances  $\{\Sigma_k\}_k$ , we directly define the “posterior”<sup>5</sup>  $p_{Z|X}(z|x) := \sum_{k=1}^K \mathcal{N}(z; \mu_k, \Sigma_k) / K$ , where  $K$  is the total number of components, so that the joint target is  $p_{X,Z}(x, z) \propto p_{Z|X}(z|x)$ . We work in two dimensions with component means adequately spaced out in a square lattice. Although the support of  $p_{Z|X}$  is all of  $\mathbb{R}^2$ , it is concentrated on a subset of  $K$  disconnected components which is not homeomorphic to  $\mathbb{R}^2$ , and thus we anticipate difficulties in using just a normalizing flow as the approximate posterior.

The initial distribution for both the NSF and CIF models is given by  $q_W := \mathcal{N}(0, \sigma_0^2 \mathbf{I})$ , with  $\sigma_0$  taken as either a fixed hyperparameter or a trainable variational parameter. The CIF extension includes an auxiliary variable  $u \in \mathbb{R}$  at each layer, adding 8.5% more parameters on top of the baseline.

The results of our experiment with  $K = 9$  and  $\sigma_0$  fixed throughout training are visible in Figure 1. We notice that the performance of the NSF model (bottom row) widely varies given the choice of  $\sigma_0$ . The CIF-NSF model (top row), however, is more robust to this choice, as the form of (7) allows the model to directly control the noise of the outputted samples and maintain fairly consistent performance over a wider range of  $\sigma_0$ . Of course we might allow  $\sigma_0$  to be learned during training, and we experiment with this on a more challenging problem ( $K = 16$ ). We find that the trained CIF models again outperform the baseline NSF models (estimated ELBO over 3 runs of  $-0.123 \pm 0.011$  vs.  $-0.564 \pm 0.003$ ), thus demonstrating the increased expressiveness of CIFs beyond just rescaling the noise.

<sup>5</sup>Note that there is no data  $x$  in this example – we define the “posterior” directly. Details are in Appendix D.

<sup>4</sup>See Appendix C for a derivation.

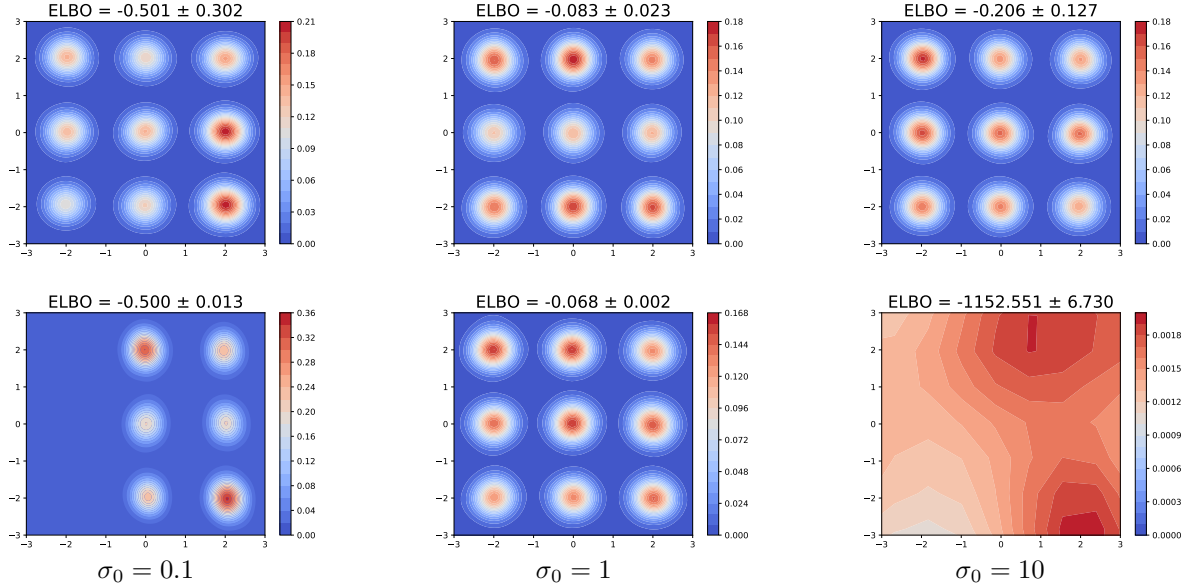


Figure 1. Samples from the trained inference models visualized using a KDE plot for a range of  $\sigma_0$  values. We ran each configuration 3 times, displaying the best model of the three in the image, with the average plus/minus standard error of the ELBO across the three runs shown in the title of the plot (higher is better). Models in the top row are CIF-NSFs, and those in the bottom row are baseline NSFs. We see that when  $\sigma_0 = 0.1$ , the NSF does not have enough noise to cover the target, and when  $\sigma_0 = 10$ , the NSF has too much noise and cannot locate the target. The CIF-NSF provides good coverage of the target in all cases.

Table 1. Test-set marginal log-likelihood averaged over three runs.

	VAE	NSF	CIF-NSF
MNIST	$-87.37 \pm 0.15$	$-82.95 \pm 0.11$	<b><math>-82.22 \pm 0.13</math></b>
FMNIST	$-217.82 \pm 0.07$	$-215.45 \pm 0.08$	<b><math>-214.50 \pm 0.11</math></b>

## 4.2 Generative Modelling of Images

For our second example, we use VI to build a generative model of the dynamically-binarized versions of the MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017) datasets. Given a neural network “decoder”  $\pi : \mathcal{Z} \rightarrow [0, 1]^d$ , the generative model of an image  $X \in \{0, 1\}^d$  is given by  $Z \sim \mathcal{N}(0, \mathbf{I})$ ,  $X \sim \prod_{j=1}^d \text{Ber}(\cdot | \pi_j(Z))$ . For all experiments we use a 20-dimensional latent space.

We consider three models of inference to facilitate maximum likelihood estimation of the parameters of  $\pi$ . First we use a variational auto-encoder (VAE) (Kingma & Welling, 2014) inference model, where  $q_{Z|X}(\cdot | x) := \mathcal{N}(\mu_Z(x), \text{diag} \sigma_Z^2(x))$  with an “encoder” network outputting both  $\mu_Z$  and  $\sigma_Z$ . Next, we consider an NSF model with a VAE encoder for  $q_{W|X}$ . The third model is a CIF which further builds on the NSF model and has an amortized auxiliary inference model as described in Appendix A. For computational purposes, we use only a small encoder and decoder which may restrict the expressiveness of the VAE.

The results of the experiment are available in Table 1.

We see that the CIF-NSF model provides the highest log-likelihood for both datasets, demonstrating the increased expressive power of auxiliary variables over the baseline NSF. In fairness, we note that the CIF-NSF model has 28.7% more parameters than the baseline; however this was the only configuration we tried and thus we anticipate being able to replicate these results in a more equitable setting.

## 5 Conclusion and Future Work

In this work, we have presented a novel method for variational inference based on continuously-indexed normalizing flows, and have demonstrated its ability to both sample from complicated target distributions and facilitate maximum likelihood estimation of generative models.

There are many interesting avenues for future work directly stemming from this result. Experimentally, it is worthwhile to assess the ability of CIFs to improve on a wider range of explicit NF baselines, and to directly compare against other methods for auxiliary VI such as semi-implicit variational inference (Yin & Zhou, 2018). On the theoretical side, it would be interesting to perform a deeper comparison between the CIF objective and the explicit NF objective, both on the value of the objective and the variance of its estimators. It would also be interesting to consider the topological implications of using bijections on the extended space (e.g. Hamiltonian methods) as opposed to CIF’s indexed functions which are only bijective over the latent space.



This work can also serve as a template for applying CIFs more generally in applications where NFs have proven effective, such as compression (Ho et al., 2019) and approximate Bayesian computation (Papamakarios et al., 2019). These approaches may require the formulation of surrogate objectives, but the expressiveness gains could overcome the additional costs (as in VI and density estimation) and should therefore be investigated.

## References

- Agakov, F. V. and Barber, D. An auxiliary variational method. In *International Conference on Neural Information Processing*, pp. 561–566. Springer, 2004.
- Behrmann, J., Vicol, P., Wang, K.-C., Grosse, R. B., and Jacobsen, J.-H. On the invertibility of invertible neural networks, 2020. URL <https://openreview.net/forum?id=BJlVeyHFwH>.
- 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.
- Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *6th International Conference on Learning Representations*, 2018.
- Burda, Y., Grosse, R. B., and Salakhutdinov, R. Importance weighted autoencoders. In *4th International Conference on Learning Representations*, 2016.
- Caterini, A. L., Doucet, A., and Sejdinovic, D. Hamiltonian variational auto-encoder. In *Advances in Neural Information Processing Systems*, pp. 8167–8177, 2018.
- Cornish, R., Caterini, A. L., Deligiannidis, G., and Doucet, A. Relaxing bijectivity constraints with continuously-indexed normalising flows. In *International Conference on Machine Learning*, 2020.
- Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *5th International Conference on Learning Representations*, 2017.
- Duan, L. L. Transport Monte Carlo. *arXiv preprint arXiv:1907.10448*, 2019.
- Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In *Advances in Neural Information Processing Systems*, pp. 7509–7520, 2019.
- Ho, J., Lohn, E., and Abbeel, P. Compression with flows via local bits-back coding. In *Advances in Neural Information Processing Systems*, pp. 3874–3883, 2019.
- Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows. In *International Conference on Machine Learning*, pp. 2078–2087, 2018.
- Huszár, F. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.
- Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.
- Jaini, P., Selby, K. A., and Yu, Y. Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, pp. 3009–3018, 2019.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.
- Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *2nd International Conference on Learning Representations*, 2014.
- Kleinegesse, S., Drovandi, C., and Gutmann, M. U. Sequential Bayesian experimental design for implicit models via mutual information. *arXiv preprint arXiv:2003.09379*, 2020.
- Lawson, J., Tucker, G., Dai, B., and Ranganath, R. Energy-inspired models: Learning with sampler-induced distributions. In *Advances in Neural Information Processing Systems*, pp. 8499–8511, 2019.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Mohamed, S. and Lakshminarayanan, B. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.
- Papamakarios, G., Sterratt, D., and Murray, I. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 837–848, 2019.
- Ranganath, R., Tran, D., and Blei, D. Hierarchical variational models. In *International Conference on Machine Learning*, pp. 324–333, 2016.
- Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015.
- Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Salimans, T., Kingma, D. P., and Welling, M. Markov chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pp. 1218–1226, 2015.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Tran, D., Ranganath, R., and Blei, D. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, pp. 5523–5533, 2017.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yin, M. and Zhou, M. Semi-implicit variational inference. In *International Conference on Machine Learning*, pp. 5660–5669, 2018.

---

# Variational Inference with Continuously-Indexed Normalizing Flows: Appendix

---

## A Stacking Layers of CIFs

In this section we describe how to stack the CIF generative process (4) to gain further expressiveness in our models. First we discuss how to do this when our approximate posterior is not amortized, which is useful in more classical VI for Bayesian inference. We then proceed with the amortized case which is useful for generative modelling – this requires additional considerations when inverting the inference model to ensure the correct form of auxiliary inference distribution. The results of this section are then summarized by [Algorithm 1](#), which provides a procedure to compute an unbiased estimator of both the amortized and non-amortized loss functions; from these we can calculate unbiased gradient estimators with automatic differentiation and thus perform gradient-based optimization.

It is worth noting that we use multi-layer objectives throughout our experiments section: the mixture of Gaussians model from [Subsection 4.1](#) uses a multi-layer CIF without amortization, and the variational posterior in [Subsection 4.2](#) uses an amortized multi-layer CIF.

### A.1 Stacked Inference Model Without Amortization

When we do not want to amortize our variational posterior, we can essentially adapt the formulation of [Cornish et al. \(2020, Section 3\)](#) to our setting. We can repeatedly apply the last two steps of the single-layer generative process (4) to produce the  $L$ -layer generative process

$$W_0 \sim q_{W_0}, \quad U_\ell \sim q_{U_\ell|W_{\ell-1}}(\cdot|W_{\ell-1}), \quad W_\ell = G_\ell(W_{\ell-1}; U_\ell), \quad (12)$$

where the final two steps are repeated sequentially over  $\ell \in \{1, \dots, L\}$ , and  $Z := W_L$ . We can thus write the joint density of  $(W_\ell, U_{1:\ell})$  recursively for  $\ell \in \{1, \dots, L\}$ :

$$q_{W_\ell, U_{1:\ell}}(w_\ell, u_{1:\ell}) = q_{W_{\ell-1}, U_{1:\ell-1}}(G_\ell^{-1}(w_\ell; u_\ell), u_{1:\ell-1}) \cdot q_{U_\ell|W_{\ell-1}}(u_\ell|G_\ell^{-1}(w_\ell; u_\ell)) \cdot |\det DG_\ell^{-1}(w_\ell; u_\ell)|,$$

where  $q_{W_0, U_{1:0}}(w_0, -) \equiv q_{W_0}(w_0)$ .

We now need to introduce an auxiliary inference distribution  $r_{U_{1:L}|Z}$ . From (3), we know that the optimal choice would be  $q_{U_{1:L}|Z}$ , which factorizes as

$$q_{U_{1:L}|Z}(u_{1:L}|z) = \prod_{\ell=1}^L q_{U_\ell|W_\ell}(u_\ell|w_\ell),$$

where  $w_L := z$  and  $w_\ell := G_{\ell+1}^{-1}(w_{\ell+1}; u_{\ell+1})$  for  $\ell \in \{1, \dots, L-1\}$ . Although this gives us the form of  $q_{U_{1:L}|Z}$ , the individual distributions  $q_{U_\ell|W_\ell}$  are intractable. However this does at least motivate us to structure  $r_{U_{1:L}|W}$  similarly as

$$r_{U_{1:L}|Z}(u_{1:L}|z) = \prod_{\ell=1}^L r_{U_\ell|W_\ell}(u_\ell|w_\ell)$$

for parametrized (but not necessarily reparametrizable) distributions  $r_{U_\ell|W_\ell}$ .

Altogether, we can write the objective function (2) as

$$\mathcal{L}_C(x) = \mathbb{E}_{q_{U_{1:L}, W_0}} \left[ \log \frac{p_{X,Z}(x, z) \cdot \prod_{\ell=1}^L r_{U_\ell|W_\ell}(u_\ell|w_\ell)}{q_{W_0}(w_0) \cdot \prod_{\ell=1}^L \{q_{U_\ell|W_{\ell-1}}(u_\ell|w_{\ell-1}) \cdot |\det DG_\ell(w_{\ell-1}; u_\ell)|^{-1}\}} \right], \quad (13)$$

where  $q_{U_{1:L}, W_0}(u_{1:L}, w_0) := q_{W_0}(w_0) \cdot \prod_{\ell=1}^L q_{U_\ell|W_{\ell-1}}(u_\ell|w_{\ell-1})$ , and  $w_\ell = G_\ell(w_{\ell-1}; u_\ell)$  recursively for  $\ell \in \{1, \dots, L\}$  with  $z := w_L$ ; in other words, the expectation is taken over all steps of (12). Since each step of the process is assumed to be reparametrizable, we can compute unbiased gradients of (13) given the unbiased estimators from [Algorithm 1](#).

Finally, we note that the multi-layer model (12) corresponds to an instance of (4) for an  $L$ -layered extended space and bijection (as per Cornish et al. (2020, Section 3.1)): first define  $G^\ell(\cdot; u_1, \dots, u_\ell) := G_\ell(\cdot; u_\ell) \circ \dots \circ G_1(\cdot; u_1)$ , and then take  $W = W_0, U = (U_1, \dots, U_L), q_{U|W}(u|w) = \prod_\ell q_{U_\ell|W_{\ell-1}}(u_\ell|G^\ell(w; u_{1:\ell}))$ , and  $G = G^L$  in (4) to arrive at (12).

## A.2 Stacked Inference Model With Amortization

Alternatively, we might want to use an amortized approximate posterior when the true posterior  $p_{Z|X}(\cdot|x)$  varies across the dataset  $\mathcal{D} = \{x_i\}_{i=1}^N$ , such as in the generative process for a variational auto-encoder (Kingma & Welling, 2014). We can easily redefine the generative process (12) given  $X$  as follows:

$$W_0 \sim q_{W_0|X}(\cdot|X), \quad U_\ell \sim q_{U_\ell|W_{\ell-1}}(\cdot|W_{\ell-1}), \quad W_\ell = G_\ell(W_{\ell-1}; U_\ell), \quad (14)$$

where  $Z := W_L$ . Again we need to introduce an auxiliary inference distribution  $r_{U_{1:L}|Z,X}$  – this time conditional on  $X$  – which should be made to match the form of  $q_{U_{1:L}|Z,X}$ , given by

$$q_{U_{1:L}|Z,X}(u_{1:L}|z, x) = \prod_{\ell=1}^L q_{U_\ell|U_{\ell+1:L}, Z, X}(u_\ell|u_{\ell+1:L}, z, x) = \prod_{\ell=1}^L q_{U_\ell|W_\ell, X}(u_\ell|w_\ell, x)$$

since, given  $W_\ell, U_\ell$  is independent of  $U_{\ell+1:L}$  but not  $X$ . This motivates structuring  $r_{U_{1:L}|Z,X}$  as

$$r_{U_{1:L}|Z,X}(u_{1:L}|z, x) = \prod_{\ell=1}^L r_{U_\ell|W_\ell, X}(u_\ell|w_\ell, x)$$

for densities  $r_{U_\ell|W_\ell, X}$  which can be evaluated pointwise.

Now, we can write the amortized objective similarly to (13), but with additional conditioning on  $x$  throughout:

$$\mathcal{L}_C(x) = \mathbb{E}_{q_{U_{1:L}, W_0|X}(\cdot|x)} \left[ \log \frac{p_{X,Z}(x, z) \cdot \prod_{\ell=1}^L r_{U_\ell|W_\ell, X}(u_\ell|w_\ell, x)}{q_{W_0|X}(w_0|x) \cdot \prod_{\ell=1}^L \{q_{U_\ell|W_{\ell-1}}(u_\ell|w_{\ell-1}) \cdot |\det \text{DG}_\ell(w_{\ell-1}; u_\ell)|^{-1}\}} \right] \quad (15)$$

with  $z := w_L$ , where now  $q_{U_{1:L}, W_0|X}(u_{1:L}, w_0|x) := q_{W_0|X}(w_0|x) \cdot \prod_{\ell=1}^L q_{U_\ell|W_{\ell-1}}(u_\ell|w_{\ell-1})$ . Again, we assume each step of the process (14) is reparametrizable, meaning that we can calculate unbiased gradients of (15) given the unbiased estimators from Algorithm 1.

## A.3 Algorithm to Compute Stacked Loss Functions

Algorithm 1 below presents a procedure for computing an unbiased estimator of either (13) or (15). Recall that we assume  $q_0$  and  $q_{U_\ell|W_{\ell-1}}$  below are reparametrizable for all  $\ell \in \{1, \dots, L\}$ , so that unbiased gradients can be calculated from the output of the algorithm using automatic differentiation methods.

## B CIFs Generalize Normalizing Flows in Variational Inference

Consider the single-layer CIF objective (6). If we suppose  $G$  is now independent of  $u$ , i.e.  $G(\cdot; u) = g(\cdot)$  for all  $u \in \mathcal{U}$  (which is certainly achievable if we assume the form (7)), we can see that

$$\begin{aligned} \mathcal{L}_C(x) &= \mathbb{E}_{q_{w,u}} \left[ \log \frac{p_{X,Z}(x, g(w)) \cdot r_{U|Z}(u|g(w))}{q_W(w) \cdot q_{U|W}(u|w) \cdot |\det \text{D}g(w)|^{-1}} \right], \\ &= \mathbb{E}_{q_w} \left[ \log \frac{p_{X,Z}(x, g(w))}{q_W(w) \cdot |\det \text{D}g(w)|^{-1}} + \mathbb{E}_{q_{u|w}(\cdot|w)} \left[ \log \frac{r_{U|Z}(u|g(w))}{q_{U|W}(u|w)} \right] \right] \\ &= \mathcal{L}_N(x) - \mathbb{E}_{q_w} [D_{\text{KL}}(q_{U|W}(\cdot|w) \parallel r_{U|Z}(\cdot|g(w)))]. \end{aligned}$$

Now, given the typical Gaussian structure of  $q_{U|W}$  and  $r_{U|Z}$ , we would expect that the optimization algorithm would ignore the conditioning variables in  $q_{U|W}$  and  $r_{U|Z}$ , and thus be able to drive the KL term to zero. This means that we can reasonably expect to achieve  $\mathcal{L}_C(x) = \mathcal{L}_N(x)$  when  $G$  is independent of  $u$  (which we can also certainly achieve as mentioned above). Thus, we will theoretically achieve at least the performance of an explicit normalizing flow in VI when using a CIF for VI.



**Algorithm 1** Unbiased estimation of  $\mathcal{L}_C(x)$ 


---

```

function ELBO( $x$ , amortized)
  if amortized then
     $q_0 \leftarrow q_{W_0|X}(\cdot|x)$ 
  else
     $q_0 \leftarrow q_{W_0}$ 
  end if
   $w_0 \sim q_0$ 
   $\Delta \leftarrow -\log q_0(w)$ 
  for  $\ell = 1, \dots, L$  do
     $u \sim q_{U_\ell|W_{\ell-1}}(\cdot|w_{\ell-1})$ 
     $w_\ell \leftarrow G_\ell(u; w_{\ell-1})$ 
    if amortized then
       $r_\ell \leftarrow r_{U_\ell|W_\ell, X}(\cdot|w_\ell, x)$ 
    else
       $r_\ell \leftarrow r_{U_\ell|W_\ell}(\cdot|w_\ell)$ 
    end if
     $\Delta \leftarrow \Delta + \log r_\ell(u) - \log q_{U_\ell|W_{\ell-1}}(u|w_{\ell-1}) + \log |\det DG_\ell(w_{\ell-1}; u)|$ 
  end for
  return  $\Delta + \log p_{X,Z}(x, w_L)$ 
end function
    
```

---

## C Relationship Between CIFs for Density Estimation and Variational Inference

When we are using CIFs for density estimation, we can write the single-layer generative process as

$$Z \sim r_Z, \quad U|Z \sim r_{U|Z}(\cdot|Z), \quad X = G^{-1}(Z; U),$$

so that  $r_X(x) = \int r_{X,U}(x, u) du$  is the proposed density model of a dataset  $\mathcal{D} = \{x_i\}_{i=1}^N$ , with

$$r_{X,U}(x, u) = r_Z(G(x; u)) \cdot r_{U|Z}(u|G(x; u)) \cdot |\det DG(x; u)|.$$

Our goal is to maximize the average likelihood of  $r_X(x)$  over the dataset, i.e.  $\max \frac{1}{N} \sum_{i=1}^N \log r_X(x_i)$ . However, since  $r_X$  is intractable, we must introduce a reparametrizable inference distribution  $q_{U|X}$  and instead maximize the ELBO, given here for a datapoint  $x \in \mathcal{D}$ :

$$\mathcal{L}(x) = \mathbb{E}_{q_{U|X}(\cdot|x)} [\log r_{X,U}(x, u) - \log q_{U|X}(u|x)]. \quad (16)$$

Note that instead of maximizing the average of (16) over the dataset, we could instead theoretically maximize the average of (16) over the unknown “true” data-generating distribution – here denoted  $q_X^*$  – which admits the objective  $\max \mathbb{E}_{q_X^*} \mathcal{L}(x)$ . Maximizing this objective is equivalent to maximizing

$$\mathbb{E}_{q_X^*} [\mathcal{L}(x)] - \mathbb{E}_{q_X^*} [\log q_X^*(x)] \quad (17)$$

since  $q_X^*$  is independent of the parameters of the model. If we substitute (16) into this expression and expand the definition for  $r_{X,U}$ , we have

$$\begin{aligned}
 & \mathbb{E}_{q_X^*} [\mathcal{L}(x)] - \mathbb{E}_{q_X^*} [\log q_X^*(x)] \\
 &= \mathbb{E}_{q_X^*} \left[ \mathbb{E}_{q_{U|X}(\cdot|x)} [\log r_{X,U}(x, u) - \log q_{U|X}(u|x)] - \log q_X^*(x) \right] \\
 &= \mathbb{E}_{q_X^*} \left[ \mathbb{E}_{q_{U|X}(\cdot|x)} [\log r_Z(G(x; u)) + \log r_{U|Z}(u|G(x; u)) + \log |\det D_x G(x; u)| - \log q_{U|X}(u|x)] - \log q_X^*(x) \right] \\
 &= \mathbb{E}_{q_{X,U}} \left[ \log \frac{r_Z(G(x; u)) \cdot r_{U|Z}(u|G(x; u))}{q_X^*(x) \cdot q_{U|X}(u|x) \cdot |\det D_x G(x; u)|^{-1}} \right],
 \end{aligned}$$

which derives (10), where we define  $q_{X,U}(x, u) := q_X^*(x) \cdot q_{U|X}(u|x)$ .

Note also that maximizing (17) is equivalent to minimizing an upper bound on  $D_{\text{KL}}(q_X^* \parallel r_X)$ :

$$\begin{aligned} \mathbb{E}_{q_X^*} [\log q_X^*(x)] - \mathbb{E}_{q_X^*} [\mathcal{L}(x)] &= \mathbb{E}_{q_X^*} \left[ \log q_X^*(x) - \mathbb{E}_{q_{U|X}(\cdot|x)} [\log r_{X,U}(x, u) - \log q_{U|X}(u|x)] \right] \\ &\geq \mathbb{E}_{q_X^*} [\log q_X^*(x) - \log r_X(x)] \quad (\text{Jensen}) \\ &= D_{\text{KL}}(q_X^* \parallel r_X). \end{aligned}$$

This is not at all surprising but at least motivates the use of (17) as a theoretical objective.

## D Further Experiment Details

We have included all details about the experiments from the main text in this section. We discuss the setup of both the image and mixture of Gaussians problems, then the specific structures used to build the NSF, CIF-NSF, and VAE models, then discuss the details of the optimization, and finally describe the log-likelihood estimator used to generate the values in Table 1.

### D.1 Setup of Specific Problems

#### D.1.1 MIXTURE OF GAUSSIANS EXPERIMENT

First of all, we note that the Mixture of Gaussians experiment may seem a bit unusual because we directly define the posterior and have no actual “data”  $x$  in the problem. However, we can easily imagine a Bayesian generative process which would essentially create such a posterior:

$$z \sim \sum_k \alpha_k \cdot \mathcal{N}(\mu_k, \Sigma_k), \quad x_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(z, \Sigma).$$

Then,  $p_{Z|X_{1:n}}(\cdot|x_{1:n}) = \sum_k \omega_k(x_{1:n}) \cdot \mathcal{N}(\tilde{\mu}_k(\bar{x}), \tilde{\Sigma}_k)$  is another mixture of Gaussians. Instead of defining the model in this way, we just directly specify the posterior as a mixture of Gaussians and perform inference.

For the  $K = 9$  experiment, we evenly space the means out in a lattice within the  $[-2, 2]^2$  square, i.e.  $\{\mu_k\}_{k=1}^9 := \{-2, 0, 2\} \times \{-2, 0, 2\}$ , and we select  $\Sigma_k := \frac{1}{42} \mathbf{I}$  for all  $k \in \{1, \dots, K\}$  so that the components had enough separation.

For the  $K = 16$  experiment, we again evenly space the means out in a lattice, but this time within the  $[-3, 3]^2$  square, i.e.  $\{\mu_k\}_{k=1}^{16} := \{-3, -1, 1, 3\} \times \{-3, -1, 1, 3\}$ , and again set  $\Sigma_k := \frac{1}{42} \mathbf{I}$  for all  $k \in \{1, \dots, K\}$ .

#### D.1.2 IMAGE EXPERIMENT

We first re-iterate that we are using a latent dimension of size 20, i.e.  $\mathcal{Z} := \mathbb{R}^{20}$ .

We use only a single-hidden-layer neural network as the decoder  $\pi$  for computational reasons. It applies a fully-connected layer with tanh nonlinearity to transform the latent variables into 8 feature maps of size  $14 \times 14$ , and then applies a zero-padded transposed convolution with a  $4 \times 4$  kernel and stride of 2 to project into size  $1 \times 28 \times 28$  (the same size as the MNIST or Fashion-MNIST data). We use this output to directly parametrize the logits of a Bernoulli distribution.

We use the standard train/test split for both MNIST and Fashion-MNIST, with 60,000 training points and 10,000 test points in each dataset. Of the 60,000 training points in each, we set aside 10% as validation points for early stopping.

### D.2 Model Details

Here we discuss the details of the models used. We note that both the NSF and CIF-NSF models use a distribution specified by the VAE encoder (Subsubsection D.2.3) as the initial distribution  $q_{W|X}$  for the image experiments.

#### D.2.1 NEURAL SPLINE FLOW BIJECTION SETTINGS

We note the hyperparameter settings that we used for the neural spline flow bijections throughout the paper in Table 2. Additionally, we note that we did not use the  $1 \times 1$  convolutions between NSF layers, electing to just randomly permute the channels for simplicity. Finally, we clip the gradients at a norm of 5 in all models using NSF bijections as recommended by Durkan et al. (2019).

Table 2. Hyperparameters used in the NSF bijections throughout the paper. Parameters have the same meaning as those from Durkan et al. (2019, Table 5), although we have additionally noted the tail bound used for the splines.

HYPERPARAMETER	VALUE
FLOW STEPS	5 FOR GAUSSIAN MIXTURE, 10 FOR IMAGES
RESIDUAL BLOCKS	2
HIDDEN FEATURES	32
BINS	8
DROPOUT	0.0
TAIL BOUND ( $B$ )	3

### D.2.2 CONTINUOUSLY-INDEXED FLOW SETTINGS

In this section, we describe the network configurations and hyperparameter settings that we use for the CIF extensions to the NSF bijections. Beyond what is required for the baseline flow, a multi-layer CIF additionally requires definitions of  $q_{U_\ell|W_{\ell-1}}, r_{U_\ell|W_\ell}$  ( $r_{U_\ell|W_\ell, X}$  when amortized), and  $s_\ell, t_\ell$  (from (7)) for  $\ell \in \{1, \dots, L\}$ , which we describe below.

For all experiments, we define the densities  $q_{U_\ell|W_{\ell-1}}(\cdot|w) := \mathcal{N}(\mu_\ell^u(w), \text{diag}(\sigma_\ell^u(w)^2))$  for all  $\ell \in \{1, \dots, L\}$  and  $w \in \mathcal{Z}$ , where  $\mu_\ell^u(w)$  and  $\sigma_\ell^u(w)$  are outputs of the same neural network: a 2-hidden-layer MLP with 10 hidden units in each layer. Similarly,  $s_\ell$  and  $t_\ell$  are two outputs of a 2-hidden-layer MLP with 10 hidden units in each layer.

The auxiliary inference model for the Gaussian mixture experiment is essentially the same as  $q$  above:  $r_{U_\ell|W_\ell}(\cdot|w) := \mathcal{N}(\mu_\ell^r(w), \text{diag}(\sigma_\ell^r(w)^2))$  for all  $\ell \in \{1, \dots, L\}$  and  $w \in \mathcal{Z}$ , where  $\mu_\ell^r(w)$  and  $\sigma_\ell^r(w)$  are outputs of a 2-hidden-layer MLP with 10 hidden units in each layer.

For the image experiment, the auxiliary inference model is now amortized, with  $r_{U_\ell|W_\ell, X}(\cdot|w, x) := \mathcal{N}(\mu_\ell^r(w, x), \text{diag}(\sigma_\ell^r(w, x)^2))$  for all  $\ell \in \{1, \dots, L\}, w \in \mathcal{Z}$ , and  $x \in \mathcal{X}$ , where  $\mu_\ell^r(w, x)$  and  $\sigma_\ell^r(w, x)$  are again two outputs of the same neural network. However, this network has a more complicated structure as it is taking in both vector-valued and image-valued inputs; we describe the steps of the network in the list below:

1. Use a linear layer to project  $w$  into a shape amenable to upsampling into an image channel (here we selected  $1 \times 7 \times 7$  as this shape).
2. Bilinearly upsample by a factor of 4 to size  $1 \times 28 \times 28$  and append as an additional channel to the input  $x$  to get a new input  $\tilde{x} \in \mathbb{R}^{2 \times 28 \times 28}$ .
3. Feed  $\tilde{x}$  into a network of the same form as the VAE encoder in Subsubsection D.2.3.

The encoder will output the parameters of the normal distribution as required. We note that the linear layer step could likely be made more parameter-efficient (e.g. map to  $1 \times 4 \times 4$  and upsample by a factor of 7), and there are likely other ways to combine vector-valued  $w$  and image-valued  $x$  more sensibly. Nevertheless, the design choices made here performed well in practice.

We also need to specify the  $u$  dimension for a CIF: we add  $u \in \mathbb{R}$  at each layer for the Gaussian mixture example, and  $u \in \mathbb{R}^2$  for the image datasets. This provides a total  $u$  dimension of 5 and 20, respectively, across the two examples.

### D.2.3 VAE ENCODER SETTINGS

The structure of the encoder used in the VAE model essentially mirrors the structure of the decoder network from Subsubsection D.1.2. In particular, given a  $1 \times 28 \times 28$  image, a zero-padded convolution is performed using a  $4 \times 4$  filter and stride length 2 with the tanh nonlinearity applied afterwards, outputting 8 feature maps each of size  $14 \times 14$ . Then, a fully-connected linear layer is applied to map the feature maps to an output which is two times the size of the latent dimension, giving us the mean and (log) standard deviation of the approximate posterior.

## D.3 Optimization Hyperparameters

Table 3 notes the parameters used for optimizing the models across experiments. There are a few things to note:

Table 3. Optimization hyperparameters used for each experiment. Note that an “epoch” for the mixture of Gaussians example is simply a single optimization step for a specified number of samples, as there is no “data”.

HYPERPARAMETER	MIXTURE OF GAUSSIANS	IMAGES
LEARNING RATE	$10^{-3}$	$10^{-3}$
WEIGHT DECAY	0	0
TRAINING BATCH SIZE	N/A	100
$q$ SAMPLES PER STEP	1,000	1
EARLY STOPPING	NO	YES
EARLY STOPPING EPOCHS	N/A	50
MAXIMUM EPOCHS	20,000	1,000

Table 4. Average variance in log-likelihood estimators across models and datasets. For each run of a particular model on a particular dataset, we calculate the estimator (either (18) or (19)) 3 separate times, and calculate the empirical variance across the outputted estimates. Then we average this variance across the original 3 runs for each model-dataset combination, arriving at the numbers in the table. For example, we have 3 VAE models trained with different random seeds on the MNIST dataset. For each of these models, we calculate (18) three separate times and calculate the empirical variance of these estimates, and then we average the empirical variances across the 3 VAE models trained with different random seeds.

	MNIST	F MNIST
VAE	$1.20 \times 10^{-4}$	$9.61 \times 10^{-4}$
NSF	$1.94 \times 10^{-4}$	$1.02 \times 10^{-3}$
CIF-NSF	$1.85 \times 10^{-4}$	$1.08 \times 10^{-3}$

1. An “epoch” for the mixture of Gaussians example is simple a single stochastic optimization step for a specified number of samples from the approximate posterior since there is no “data” in this example.
2. None of the image experiments actually reached the maximum number of epochs.
3. The hyperparameter choices below were essentially default choices.

#### D.4 Estimation of Marginal Log-Likelihood

To generate the log-likelihood outputs in Table 1, we use an importance-sampling-based estimate as in e.g. Rezende et al. (2014, Appendix E) for each run, and then average the results of this estimator across three runs. Specifically, given the test dataset  $\mathcal{D}_{\text{test}} = \{x_i\}_{i=1}^m$  and a number of samples  $S$ , the average log-likelihood for a single run is given by

$$\frac{1}{m} \sum_{i=1}^m \log \left( \frac{1}{S} \sum_{s=1}^S \frac{p_{X,Z}(x_i, z_i^{(s)})}{q_{Z|X}(z_i^{(s)}|x)} \right), \quad \text{where } z_i^{(s)} \sim q_{Z|X}(\cdot|x_i), \quad (18)$$

for explicit models  $q_{Z|X}$  (i.e. VAE and NSF), and

$$\frac{1}{m} \sum_{i=1}^m \log \left( \frac{1}{S} \sum_{s=1}^S \frac{p_{X,Z}(x_i, z_i^{(s)}) \cdot r_{U|Z,X}(u_i^{(s)}|z_i^{(s)}, x_i)}{q_{Z,U|X}(z_i^{(s)}, u_i^{(s)}|x)} \right), \quad \text{where } z_i^{(s)}, u_i^{(s)} \sim q_{Z,U|X}(\cdot, \cdot|x_i), \quad (19)$$

for implicit models  $q_{Z,U|X}$  (i.e. CIF-NSF). We take  $S = 1000$  in practice, finding that this provides adequately low-variance estimators as noted in Table 4.