Variational Inference with Continuously-Indexed Normalizing Flows Supplementary Material

Anthony Caterini[1] **Rob Cornish**[1] **Dino Sejdinovic**[1] **Arnaud Doucet**[1]

[1]Department of Statistics, University of Oxford

A SINGLE-LAYER DENSITY AND OBJECTIVE FUNCTION

Here we demonstrate how to obtain the single-layer CIF density and associated objective function.

Single-layer density We can derive the joint density $q_{Z,U}$ by first considering the density over (Z, U, W) and integrating out W:

$$q_{Z,U}(z, u) = \int q_{Z,U,W}(z, u, w) \, \mathrm{d}w = \int q_W(w) \cdot q_{U|W}(u \mid w) \cdot \delta(z - G(w; u)) \, \mathrm{d}w.$$

Now if we perform the change of variable $w = G^{-1}(z'; u)$, we get $\mathrm{d}w = |\det \mathrm{D}_z G^{-1}(z'; u)| \, \mathrm{d}z'$, which then gives

$$\begin{aligned} q_{Z,U}(z, u) &= \int q_W(G^{-1}(z'; u)) \cdot q_{U|W}(u \mid G^{-1}(z'; u)) \cdot \delta(z - z') \cdot |\det \mathrm{D}_z G^{-1}(z'; u)| \, \mathrm{d}z' \\ &= q_W(G^{-1}(z; u)) \cdot q_{U|W}(u \mid G^{-1}(z; u)) \cdot |\det \mathrm{D}_z G^{-1}(z; u)|. \end{aligned}$$

Single-layer objective First, we substitute our model $q_{Z,U}$ into (3) to obtain

$$\mathcal{L}(x) = \mathbb{E}_{(z,u)\sim q_{Z,U}} \left[\frac{p_{X,Z}(x, z) \cdot r_{U|Z}(u \mid z)}{q_W(G^{-1}(z; u)) \cdot q_{U|W}(u \mid G^{-1}(z; u)) \cdot |\det \mathrm{D}_z G^{-1}(z; u)|} \right].$$

Now, noting that $z = G(w; u)$ for some $w \sim q_W$ as per the sampling procedure (4), we can rewrite the above objective instead as an expectation over $q_{W,U}$ (using LOTUS) to obtain

$$\mathcal{L}(x) = \mathbb{E}_{(w,u)\sim q_{W,U}} \left[\log \frac{p_{X,Z}(x, G(w; u)) \cdot r_{U|Z}(u \mid G(w; u))}{q_W(w) \cdot q_{U|W}(u \mid w) \cdot |\det \mathrm{D}_w G(w; u)|^{-1}} \right],$$

since $\mathrm{D}_z G^{-1}\left(G(w; u); u\right) = \mathrm{D}_w G(w; u)$, which recovers (9).

B MULTI-LAYER DENSITY AND OBJECTIVE FUNCTION

This section is much like the previous section, except this time for the multi-layer model. We demonstrate how to recursively calculate the density and provide the objective function for a multi-layer model in both the un-amortized and amortized settings.

Recursive multi-layer density We can derive the full joint density $q_{Z,U_{1:L}}$ by first considering an intermediate density $q_{W_\ell,U_{1:\ell}}$ for $\ell \in \{1, \dots, L\}$, then integrating over the variable $W_{\ell-1}$:

$$\begin{aligned} q_{W_\ell,U_{1:\ell}}(w_\ell, u_{1:\ell}) &= \int q_{W_\ell,U_\ell,W_{\ell-1},U_{1:\ell-1}}(w_\ell, u_\ell, w_{\ell-1}, u_{1:\ell-1}) \, \mathrm{d}w_{\ell-1} \\ &= \int q_{W_{\ell-1},U_{1:\ell-1}}(w_{\ell-1}, u_{1:\ell-1}) \cdot q_{U_\ell|W_{\ell-1}}(u_\ell \mid w_{\ell-1}) \cdot \delta(w_\ell - G_\ell(w_{\ell-1}; u_\ell)) \, \mathrm{d}w_{\ell-1}, \end{aligned}$$

where in the second line we use the fact that U_ℓ is conditionally independent of $U_{1:\ell-1}$ given $W_{\ell-1}$ (note that this fact is also used to derive the structure of the auxiliary posterior $q_{U_{1:L}|Z}$), and the base case of the recursion is given by $q_{W_0,U_{1:0}}(w_0,-) \coloneqq q_{W_0}(w_0)$. Now, as in the previous section, we use the change of variable $w_{\ell-1} = G_\ell^{-1}(w'_\ell;u_\ell)$ to obtain $\mathrm{d}w_{\ell-1} = |\det \mathrm{D}_{w_\ell} G_\ell^{-1}(w'_\ell;u_\ell)|\,\mathrm{d}w'_\ell$, and thus

$$q_{W_\ell,U_{1:\ell}}(w_\ell,u_{1:\ell}) = q_{W_{\ell-1},U_{1:\ell-1}}\left(G_\ell^{-1}(w_\ell;u_\ell),u_{1:\ell-1}\right)\cdot q_{U_\ell|W_{\ell-1}}\left(u_\ell|G_\ell^{-1}(w_\ell;u_\ell)\right)\cdot|\det \mathrm{D}_{w_\ell}G_\ell^{-1}(w_\ell;u_\ell)|.$$

We obtain our full inference model as the L^{th} step of the recursion, i.e. $q_{Z,U_{1:L}} \equiv q_{W_L,U_{1:L}}$.

Multi-layer objective function Given our joint model $q_{Z,U_{1:L}}$ and the factorized auxiliary inference model $r_{U_{1:L}|Z}$ from (8), we can write the objective function from (2) as

$$\mathcal{L}(x) = \mathbb{E}_{(z,u_{1:L})\sim q_{Z,U_{1:L}}}\left[\log\frac{p_{X,Z}(x,z)\cdot r_{U_{1:L}|Z}(u_{1:L}\mid z)}{q_{Z,U_{1:L}}(z,u_{1:L})}\right].$$

However, as in the single-layer case, it is difficult to calculate unbiased gradients of the objective – as written in this form – with respect to the parameters of the bijections G_ℓ, as these bijections are also appearing in the distribution over which we take the expectation. Thus we write $w_\ell = G_\ell(w_{\ell-1};u_\ell)$ recursively for $\ell \in \{1,\ldots,L\}$, with $z \coloneqq w_L$, to rewrite the objective function instead as an expectation over $q_{W_0,U_{1:L}}(u_{1:L},w_0) \coloneqq q_{W_0}(w_0)\cdot\prod_{\ell=1}^L q_{U_\ell|W_{\ell-1}}(u_\ell|w_{\ell-1})$ as below:

$$\begin{aligned}\mathcal{L}(x) &= \mathbb{E}_{(w_0,u_{1:L})\sim q_{W_0,U_{1:L}}}\left[\log\frac{p_{X,Z}(x,z)\cdot\prod_{\ell=1}^L r_{U_\ell|W_\ell}(u_\ell\mid w_\ell)}{q_{W_0}(w_0)\cdot\prod_{\ell=1}^L\left\{q_{U_\ell|W_{\ell-1}}(u_\ell\mid w_{\ell-1})\cdot|\det \mathrm{D}_{w_{\ell-1}}G_\ell(w_{\ell-1};u_\ell)|^{-1}\right\}}\right]\\ &= \mathbb{E}_{(w_0,u_{1:L})\sim q_{W_0,U_{1:L}}}\left[-\log q_{W_0}(w_0)+\sum_{\ell=1}^L\log\frac{r_{U_\ell|W_\ell}(u_\ell\mid w_\ell)\cdot|\det \mathrm{D}_{w_{\ell-1}}G_\ell(w_{\ell-1};u_\ell)|}{q_{U_\ell|W_{\ell-1}}(u_\ell\mid w_{\ell-1})}+\log p_{X,Z}(x,z)\right].\end{aligned}\tag{1}$$

The form of objective given in (1) demonstrates how Algorithm 1 works: initialize with $-\log q_{W_0}(w_0)$, collect $r_\ell, \mathrm{D}G_\ell$, and q_ℓ terms at each step ℓ, and then finish by evaluating the joint target at the realized z value.

Amortization When using amortization, we can redefine the generative process (7) given X as follows:

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

where $Z \coloneqq W_L$. Now, additionally conditioning our auxiliary inference model $r_{U_{1:L}|Z,X}$ on X, we can write the objective (1) for the amortized case as

$$\mathcal{L}(x) = \mathbb{E}_{(w_0,u_{1:L})\sim q_{W_0,U_{1:L}|X}}\left[-\log q_{W_0|X}(w_0\mid x)+\sum_{\ell=1}^L\log\frac{r_{U_\ell|W_\ell,X}(u_\ell\mid w_\ell,x)\cdot|\det \mathrm{D}_{w_{\ell-1}}G_\ell(w_{\ell-1};u_\ell)|}{q_{U_\ell|W_{\ell-1}}(u_\ell\mid w_{\ell-1})}+\log p_{X,Z}(x,z)\right].\tag{2}$$

Multi-layer CIF as a single-layer CIF Lastly, we show here how the multi-layer model (7) 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 $G^\ell(\cdot;u_1,\ldots,u_\ell) \coloneqq G_\ell(\cdot;u_\ell)\circ\cdots\circ G_1(\cdot;u_1)$, and then take $W \coloneqq W_0, U \coloneqq U_1,\ldots,U_L), q_{U|W}(u\mid w) \coloneqq \prod_\ell q_{U_\ell|W_{\ell-1}}\left(u_\ell\mid G^\ell(w;u_{1:\ell})\right)$, and $G \coloneqq G^L$ in (4) to arrive at (7).

C CONTINUOUSLY-INDEXED FLOWS VERSUS BASELINE NORMALIZING FLOWS

Here, we provide a proof of Proposition 3.1. Throughout the proof, we consider x such that $\mathcal{L}_2^\phi(x) \geq \mathcal{L}_2^\psi(x)$ as per our assumption.

Let us first denote the normalizing flow objective (11) as $\mathcal{L}_1$. It is not hard to show that $\mathcal{L}_2^\psi$ reduces to $\mathcal{L}_1$:

$$\begin{aligned}\mathcal{L}_2^\psi(x) &= \mathbb{E}_{(w,u)\sim q^\phi_{W,U}}\left[\log\frac{p_{X,Z}(x,G_\psi(w;u))\cdot r_{U|Z}(u\mid G_\psi(w;u))}{q_W(w)\cdot q^\psi_{U|W}(u\mid w)\cdot|\det \mathrm{D}_w G_\psi(w;u)|^{-1}}\right],\\ &= \mathbb{E}_{w\sim q_W}\,\mathbb{E}_{u\sim\rho}\left[\log\frac{p_{X,Z}(x,g(w))\cdot\rho(u)}{q_W(w)\cdot\rho(u)\cdot|\det \mathrm{D}g(w)|^{-1}}\right]\\ &= \mathcal{L}_1(x).\end{aligned}$$

From (3), we also know that

$$\mathbb{E}_{z\sim q_Z^\phi}\left[\log p_{X,Z}(x,z) - \log q_Z^\phi(z)\right] \geq \mathcal{L}_2^\phi(x),$$

as the left-hand-side is the intractable marginal ELBO obtained when using $q_Z^\phi(z) \coloneqq \int q_{Z,U}^\phi(z,u)\,\mathrm{d}u$ as the variational distribution.

We get the final result by exploiting the standard relationship between the ELBO and the KL divergence:

$$\begin{aligned}
D_{\mathrm{KL}}\Big(q_Z^\phi \mid\mid p_{Z|X}(\cdot|x)\Big) &= \log p_X(x) - \mathbb{E}_{z\sim q_Z^\phi}\left[\log p_{X,Z}(x,z) - \log q_Z^\phi(z)\right] \\
&\leq \log p_X(x) - \mathcal{L}_2^\phi(x) \\
&\leq \log p_X(x) - \mathcal{L}_1(x) \\
&= D_{\mathrm{KL}}\big(g_\# q_W \mid\mid p_{Z|X}(\cdot|x)\big)\,.
\end{aligned}$$

D RELATIONSHIP BETWEEN CONTINUOUSLY-INDEXED FLOWS FOR DENSITY ESTIMATION AND VARIATIONAL INFERENCE

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

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

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

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

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 $x \in \mathcal{D}$:

$$\mathcal{L}(x) = \mathbb{E}_{u\sim q_{U|X}(\cdot|x)}\left[\log r_{X,U}(x,u) - \log q_{U|X}(u \mid x)\right]. \tag{3}$$

Note that instead of maximizing the average of (3) over the dataset, we could instead theoretically maximize the average of (3) 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

$$\mathbb{E}_{x\sim q_X^*}\left[\mathcal{L}(x)\right] - \mathbb{E}_{x\sim q_X^*}\left[\log q_X^*(x)\right] \tag{4}$$

since q_X^* is independent of the parameters of the model. If we substitute (3) into this expression and expand the definition for $r_{X,U}$, we have

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

which derives (12), where we define $q_{X,U}(x,u) \coloneqq q_X^*(x) \cdot q_{U|X}(u \mid x)$.

Note also that maximizing (4) is equivalent to minimizing an upper bound on $D_{\mathrm{KL}}(q_X^* \mid\mid r_X)$:

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

This is not surprising but at least motivates the use of (4) as a theoretical objective.

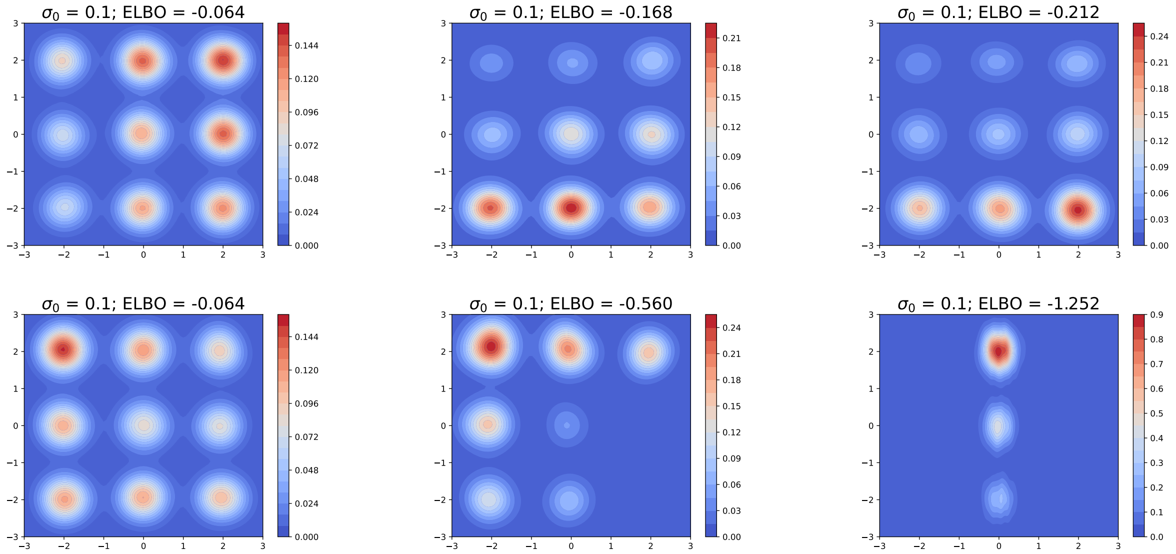

Figure 1: All runs of the mixture of Gaussians experiment for $\sigma_0 = 0.1$. CIF models are in the top row, NSF models in the bottom row.

E FURTHER EXPERIMENT DETAILS

We have included all details about the experiments from the main text in this section. We first include the approximate posterior plots from *all* runs of the $K = 9$ mixture of Gaussians experiment and not just the best of three random seeds. We next discuss the setup of both the image and mixture of Gaussians problems, then the specific structures used to build the baseline flow models, CIF extensions, 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.

E.1 MORE 2D MIXTURE PLOTS

We have included visualizations of the trained approximate posteriors for all three runs of each setting of σ_0 in Figure 1, Figure 2, and Figure 3. We notice consistently better performance from the trained CIF models, with the CIFs learning to cover the modes in all cases.

E.2 SETUP OF SPECIFIC PROBLEMS

E.2.1 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:

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

Then, $p_{Z|X_{1:n}}(\cdot \mid x_{1:n}) = \sum_k \omega_k(x_{1:n}) \cdot \mathcal{N}(\tilde{\mu}_k(\bar{x}), \tilde{\Sigma}_k)$ is another mixture of Gaussians for some weights ω_k and modified parameters $\tilde{\mu}_k, \tilde{\Sigma}_k$. 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 $\Sigma_k := \frac{1}{4^2}\mathbf{I}$ for all $k \in \{1, \ldots, K\}$ 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 $\Sigma_k := \frac{1}{4^2}\mathbf{I}$ for all $k \in \{1, \ldots, K\}$.

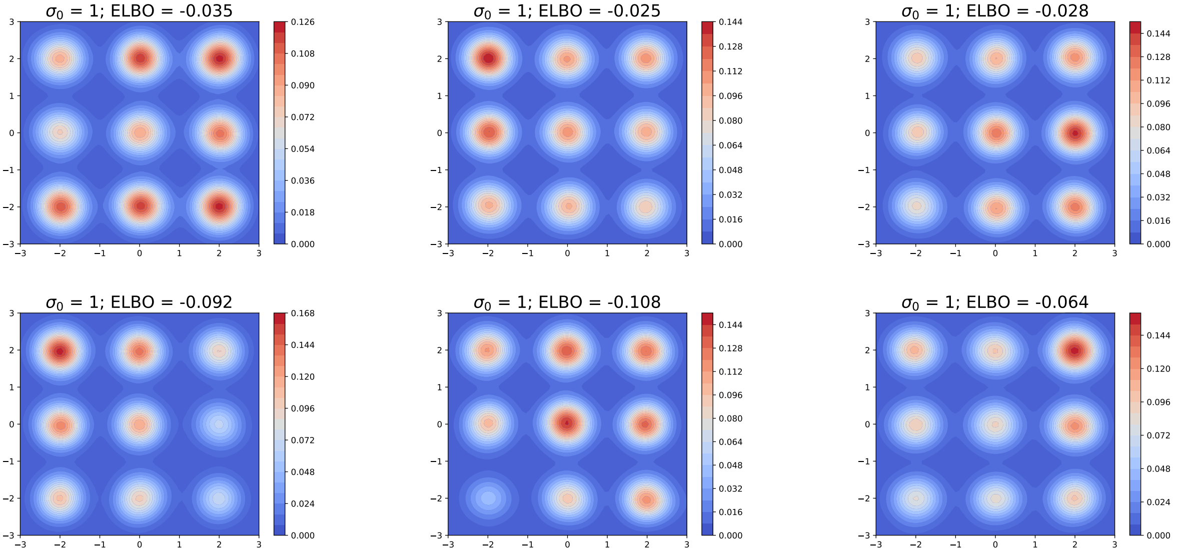

Figure 2: All runs of the mixture of Gaussians experiment for $\sigma_0 = 1$. CIF models are in the top row, NSF models in the bottom row.

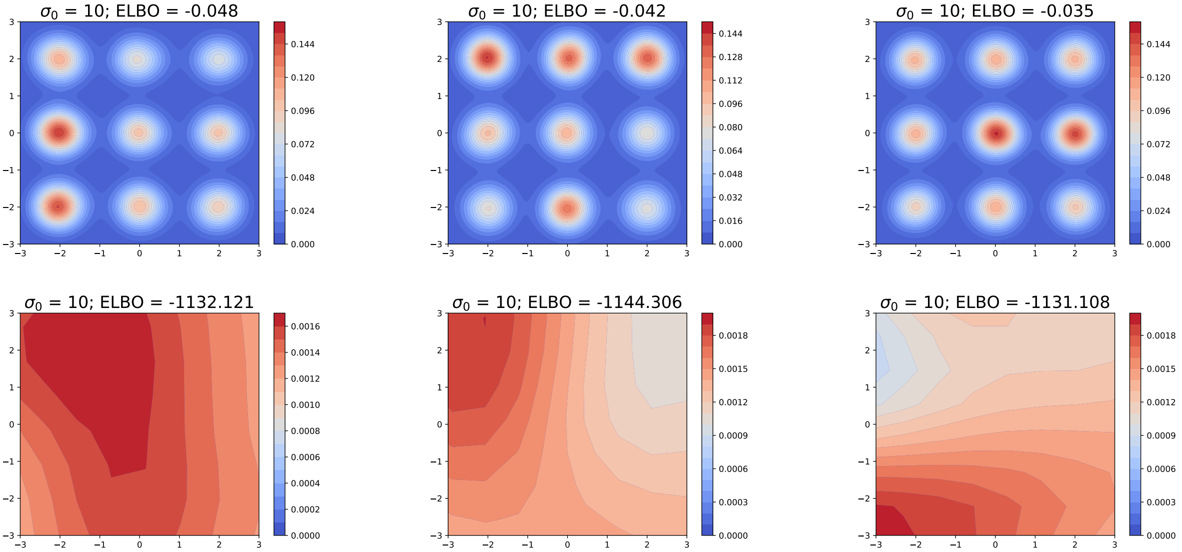

Figure 3: All runs of the mixture of Gaussians experiment for $\sigma_0 = 10$. CIF models are in the top row, NSF models in the bottom row.

Table 1: Hyperparameters used in the MAF bijections throughout the paper.

Hyperparameter	**Value**
Flow steps	5
Autoregressive networks	512×2 for Large MAF, 420×2 for all others (including CIF-MAF)
Batch normalization	`True` for MAFs, `False` for CIF-MAFs

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	44 for Large NSF, 32 for all others (including CIF-NSF)
Bins	8
Dropout	0.0
Tail bound (B)	3

E.2.2 Image Experiment

We first re-iterate that we set $n_Z := 20$ for the small decoder, and $n_Z := 32$ for the large one, so that $\mathcal{Z} := \mathbb{R}^{n_Z}$.

The small decoder is a single-hidden-layer transposed convolutional network. It applies a fully-connected layer with $\tanh$ nonlinearity to transform the n_Z-dimensional latent variables into 8 feature maps of size 14×14, and then applies a zero-padded transposed convolution with a 4×4 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.

The large decoder exactly matches the form from Durkan et al. (2019); indeed, we simply used the `ConvDecoder` directly from their codebase (`https://github.com/bayesiains/nsf`).

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.

E.3 MODEL DETAILS

Here we discuss the details of the models used. We note that all of the MAF, NSF, CIF-MAF, and CIF-NSF models use a distribution specified by the VAE encoder (Subsubsection E.3.4) as the initial distribution $q_{W|X}$ for the image experiments.

E.3.1 Masked Autoregressive Flow Bijection Settings

We note the hyperparameter settings that we used for the masked autoregressive flow bijections throughout the paper in Table 1. We insert batch normalization layers between flow steps in the MAF models as per the recomnmendation of Papamakarios et al. (2017), but do not use them in CIFs as the form of G_ℓ makes them unnecessary.

E.3.2 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. We clip the gradients at a norm of 5 in all models using NSF bijections as recommended by Durkan et al. (2019).

E.3.3 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_ℓ, t_ℓ (from (6)) for $\ell \in \{1, \ldots, L\}$, which we describe below.

For all experiments, we define the densities $q_{U_\ell|W_{\ell-1}}(\cdot \mid w) := \mathcal{N}\left(\mu_\ell^u(w), \text{diag}\left(\sigma_\ell^u(w)^2\right)\right)$ for all $\ell \in \{1, \ldots, L\}$ 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_ℓ and t_ℓ 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: $r_{U_\ell|W_\ell}(\cdot \mid w) := \mathcal{N}\left(\mu_\ell^r(w), \text{diag}\left(\sigma_\ell^r(w)^2\right)\right)$ for all $\ell \in \{1, \ldots, L\}$ 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 $r_{U_\ell|W_\ell,X}(\cdot \mid w, x) := \mathcal{N}\left(\mu_\ell^r(w,x), \text{diag}\left(\sigma_\ell^r(w,x)^2\right)\right)$ for all $\ell \in \{1, \ldots, L\}, w \in \mathcal{Z}$, 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 E.3.4.

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 for the Gaussian mixture example, 10 for the CIF-MAF, and 20 for the CIF-NSF.

E.3.4 VAE Encoder Settings

The structure of the encoder used in the VAE model essentially mirrors the structure of the small decoder network from Subsubsection E.2.2. In particular, given a $1 \times 28 \times 28$ image, a zero-padded convolution is performed using a 4×4 filter and stride length 2 with the $\tanh$ nonlinearity applied afterwards, outputting 8 feature maps each of size 14×14. 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.

E.4 OPTIMIZATION HYPERPARAMETERS

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

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.

E.5 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

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 the small decoder experiment. For each run of a particular model on a particular dataset, we calculate the estimator (either (5) or (6)) 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 Small VAE models trained with different random seeds on the MNIST dataset. For each of these models, we first calculate (5) three separate times obtaining the empirical variance of these estimates, and then we average the empirical variances across the 3 Small VAE models trained with different random seeds.

Model	**MNIST**	**Fashion-MNIST**
Small VAE	2.60×10^{-3}	2.51×10^{-3}
Small MAF	2.85×10^{-3}	6.77×10^{-3}
Large MAF	2.59×10^{-3}	4.93×10^{-3}
CIF-MAF	2.38×10^{-3}	6.45×10^{-4}
Small NSF	2.78×10^{-4}	3.84×10^{-3}
Large NSF	1.84×10^{-3}	$2,27 \times 10^{-3}$
CIF-NSF	9.78×10^{-4}	4.47×10^{-3}

dataset $\mathcal{D}_{\text{test}} = \{x_i\}_{i=1}^{N_{\text{test}}}$ and a number of samples S, the average log-likelihood for a single run is given by

$$\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \log \left(\frac{1}{S} \sum_{s=1}^{S} \frac{p_{X,Z}(x_i, z_i^{(s)})}{q_{Z|X}(z_i^{(s)} \mid x)} \right), \qquad \text{where } z_i^{(s)} \sim q_{Z|X}(\cdot \mid x_i), \tag{5}$$

for explicit models $q_{Z|X}$ (e.g. VAEs and normalizing flows), and

$$\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \log \left(\frac{1}{S} \sum_{s=1}^{S} \frac{p_{X,Z}(x_i, z_i^{(s)}) \cdot r_{U|Z,X}(u_i^{(s)} \mid z_i^{(s)}, x_i)}{q_{Z,U|X}(z_i^{(s)}, u_i^{(s)} \mid x)} \right), \qquad \text{where } z_i^{(s)}, u_i^{(s)} \sim q_{Z,U|X}(\cdot, \cdot \mid x_i), \tag{6}$$

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

F FURTHER DETAILS ON THE MARGINAL ELBO ESTIMATOR

Here is the full version of the (positively) biased, but still consistent, estimator of (2) described in Subsection 4.1:

$$\widehat{\mathcal{L}}(x) := \frac{1}{N} \sum_{i=1}^{N} \left(\log p_{X,Z}(x, z_i) - \log \left\{ \frac{1}{M} \sum_{j=1}^{M} \frac{q_{Z,U}(z_i, u_{i,j})}{r_{U|Z}(u_{i,j} \mid z_i)} \right\} \right), \tag{7}$$

Table 5: Average plus/minus standard deviation of 20 estimates of the **marginal** ELBO (7) from a single trained model of the mixture of Gaussians experiment. We vary the values of N and select M as either a linear ($M = N$) or square-root ($M = \sqrt{N}$) function of N. We include Monte Carlo estimates of the auxiliary ELBO from the sample of z_i for reference.

N	$M = N$		$M = \sqrt{N}$	
	Marginal	**Auxiliary**	**Marginal**	**Auxiliary**
1	-0.213 ± 0.665	-0.237 ± 0.668	-0.302 ± 0.361	-0.317 ± 0.363
5	-0.165 ± 0.180	-0.166 ± 0.183	-0.178 ± 0.221	-0.186 ± 0.223
10	-0.189 ± 0.180	-0.190 ± 0.187	-0.156 ± 0.136	-0.160 ± 0.145
50	-0.190 ± 0.078	-0.198 ± 0.076	-0.162 ± 0.094	-0.167 ± 0.096
100	-0.177 ± 0.057	-0.182 ± 0.059	-0.181 ± 0.056	-0.185 ± 0.053
500	-0.176 ± 0.022	-0.180 ± 0.022	-0.172 ± 0.018	-0.177 ± 0.019
1,000	-0.160 ± 0.018	-0.165 ± 0.019	-0.169 ± 0.020	-0.174 ± 0.020
5,000	-0.169 ± 0.006	-0.174 ± 0.006	-0.174 ± 0.006	-0.179 ± 0.006
10,000	-0.171 ± 0.006	-0.175 ± 0.006	-0.170 ± 0.005	-0.175 ± 0.005
50,000	-0.169 ± 0.002	-0.174 ± 0.002	-0.170 ± 0.002	-0.175 ± 0.002

where $(z_i, u_i') \overset{\text{i.i.d.}}{\sim} q_{Z,U}$ for $i \in \{1, \dots, N\}$, and for each i, $u_{i,j} \overset{\text{i.i.d.}}{\sim} r_{U|Z}(\cdot \mid z_i)$ for $j \in \{1, \dots, M\}$. We need to be careful to make M large enough so that our estimates are not too biased, and at the same time make N large enough so that our estimates are not dominated by variance. We explore the relationship between values of the estimator (7) for a single trained model, and the settings of M and N in Table 5. For $N = 10{,}000$ and $M = 100$, we have both acceptably low variance and low bias as the estimator appears to not be significantly lowered by increasing M.