Supplementary Material: Sylvester Normalizing Flows for Variational Inference

A Architecture

In the experiments we used convolutional layers for both the encoder and the decoder. Moreover, we used the gated activation function for convolutional layers:

$$\mathbf{h}_l = (\mathbf{W}_l * \mathbf{h}_{l-1} + \mathbf{b}_l) \odot \sigma(\mathbf{V}_l * \mathbf{h}_{l-1} + \mathbf{c}_l),$$

where $\mathbf{h}_{l-1}$ and $\mathbf{h}_l$ are inputs and outputs of the l-th layer, respectively, $\mathbf{W}_l, \mathbf{V}_l$ are weights of the l-th layer, $\mathbf{b}_l, \mathbf{c}_l$ denote biases, $*$ is the convolution operator, and $\sigma(\cdot)$ is the sigmoid activation function.

We used the following architecture of the encoder (k is a kernel size, p is a padding size, and s is a stride size):[1]

$$\begin{aligned}
&\mathrm{Conv}(\mathrm{in}=1, \mathrm{out}=32, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=1)\\
&\mathrm{Conv}(\mathrm{in}=32, \mathrm{out}=32, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=2)\\
&\mathrm{Conv}(\mathrm{in}=32, \mathrm{out}=64, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=1)\\
&\mathrm{Conv}(\mathrm{in}=64, \mathrm{out}=64, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=2)\\
&\mathrm{Conv}(\mathrm{in}=64, \mathrm{out}=64, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=1)\\
&\mathrm{Conv}(\mathrm{in}=64, \mathrm{out}=64, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=1)\\
&\mathrm{Conv}(\mathrm{in}=64, \mathrm{out}=256, \mathrm{k}=7, \mathrm{p}=0, \mathrm{s}=1)
\end{aligned}$$

Notice the last layer acts as a fully-connected layer. Eventually, fully-connected linear layers were used to parameterized diagonal Gaussian distribution and amortized parameters of a flow.

The decoder mirrors the structure of the encoder with transposed convolutional layers (op is an outer padding):

$$\begin{aligned}
&\mathrm{ConvT}(\mathrm{in}=64, \mathrm{out}=64, \mathrm{k}=7, \mathrm{p}=0, \mathrm{s}=1)\\
&\mathrm{ConvT}(\mathrm{in}=64, \mathrm{out}=64, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=1)\\
&\mathrm{ConvT}(\mathrm{in}=64, \mathrm{out}=32, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=2, \mathrm{op}=1)\\
&\mathrm{ConvT}(\mathrm{in}=32, \mathrm{out}=32, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=1)\\
&\mathrm{ConvT}(\mathrm{in}=32, \mathrm{out}=32, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=2, \mathrm{op}=1)\\
&\mathrm{ConvT}(\mathrm{in}=32, \mathrm{out}=32, \mathrm{k}=5, \mathrm{p}=2, \mathrm{s}=1)\\
&\mathrm{ConvT}(\mathrm{in}=32, \mathrm{out}=1, \mathrm{k}=1, \mathrm{p}=0, \mathrm{s}=1)
\end{aligned}$$

B Description of datasets

In the experimetns we used the following four image datasets: static MNIST[2], OMNIGLOT[3], Caltech 101 Silhouettes[4], and Frey Faces[5]. Frey Faces contains images of size 28 × 20 and all other datasets contain 28 × 28 images.

MNIST consists of hand-written digits split into 60,000 training datapoints and 10,000 test sample points. In order to perform model selection we put aside 10,000 images from the training set.

OMNIGLOT is a dataset containing 1,623 hand-written characters from 50 various alphabets. Each character is represented by about 20 images that makes the problem very challenging. The dataset is split into 24,345 training datapoints and 8,070 test images. We randomly pick 1,345 training examples for validation. During training we applied dynamic binarization of data similarly to dynamic MNIST.

Caltech 101 Silhouettes contains images representing sil-

[1] We use a PyTorch convention of defining convolutional layers.

[2] `http://yann.lecun.com/exdb/mnist/`

[3] `https://github.com/yburda/iwae/blob/master/datasets/OMNIGLOT/chardata.mat`.

[4] `https://people.cs.umass.edu/~marlin/data/caltech101_silhouettes_28_split1.mat`.

[5] `http://www.cs.nyu.edu/~roweis/data/frey_rawface.mat`

houettes of 101 object classes. Each image is a filled, black polygon of an object on a white background. There are 4,100 training images, 2,264 validation datapoints and 2,307 test examples. The dataset is characterized by a small training sample size and many classes that makes the learning problem ambitious.

Frey Faces is a dataset of faces of a one person with different emotional expressions. The dataset consists of nearly 2,000 gray-scaled images. We randomly split them into 1,565 training images, 200 validation images and 200 test images. We repeated the experiment 3 times.

C MNIST experiments

The exact numbers for the evidence lower bound as shown in Fig. 3 are listed in Table 1.

Table 1: Negative evidence lower bounds for the test set on MNIST. All results are obtained with stochastic hidden units of size 64.

Model	**-ELBO**
VAE	86.51 ± 0.11
Planar ($K = 4$)	86.40 ± 0.08
Planar ($K = 8$)	86.37 ± 0.006
Planar ($K = 16$)	85.71 ± 0.20
IAF ($W = 320$, $K = 4$)	85.04 ± 0.07
IAF ($W = 320$, $K = 8$)	84.70 ± 0.08
IAF ($W = 320$, $K = 16$)	84.50 ± 0.11
IAF ($W = 640$, $K = 4$)	84.58 ± 0.06
IAF ($W = 640$, $K = 8$)	84.50 ± 0.08
IAF ($W = 640$, $K = 16$)	84.29 ± 0.09
IAF ($W = 1280$, $K = 4$)	84.96 ± 0.25
IAF ($W = 1280$, $K = 8$)	84.55 ± 0.08
IAF ($W = 1280$, $K = 16$)	84.30 ± 0.16
O-SNF ($M = 16$, $K = 4$)	84.08 ± 0.04
O-SNF ($M = 16$, $K = 8$)	83.71 ± 0.07
O-SNF ($M = 16$, $K = 16$)	83.52 ± 0.09
O-SNF ($M = 32$, $K = 4$)	83.76 ± 0.02
O-SNF ($M = 32$, $K = 8$)	83.58 ± 0.03
O-SNF ($M = 32$, $K = 16$)	83.32 ± 0.06
H-SNF ($K = 4$, $H = 4$)	83.73 ± 0.05
H-SNF ($K = 8$, $H = 4$)	83.49 ± 0.08
H-SNF ($K = 16$, $H = 4$)	83.36 ± 0.04
H-SNF ($K = 4$, $H = 8$)	83.72 ± 0.03
H-SNF ($K = 8$, $H = 8$)	83.52 ± 0.01
H-SNF ($K = 16$, $H = 8$)	83.35 ± 0.05
T-SNF ($K = 4$)	83.74 ± 0.04
T-SNF ($K = 8$)	83.48 ± 0.03
T-SNF ($K = 16$)	83.35 ± 0.06