

Variational Inference with Normalizing Flows

ATML Group 12

Department of Computer Science, University of Oxford

1032626, 1034125, 1034129, 1036969



Introduction

- Calculating the true posterior distribution of inference tasks is in most cases an intractable problem.
- Lots of research on approaches for efficient approximation of the posterior, however the resulting classes prove to be of limited expressiveness.
- The authors in [1] introduce the notion of normalizing flows, sequences of invertible transformations applied to a simple initial density, to efficiently create more expressive families of candidate posteriors to be used for variational inference.
- We compare the performance of different types of normalizing flows on the MNIST dataset.

Our Work

- Reproduced experiment on MNIST using Linear Normalizing Flows
- Reproduced experiment on MNIST using NICE
- Extended the ideas of the paper and experimented with Invertible Convolutional Flows
- Created open-source Github repository with code and results: github.com/ATML-Group-12/normalising_flows

Theoretical Background

Normalizing flows are sequences of invertible, smooth mappings $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$. We define a flow of length K to be K such transformations as follows:

$$\mathbf{z}_K = f_K \circ \dots \circ f_2 \circ f_1(\mathbf{z}_0)$$

where \mathbf{z}_0 is a random variable that has a simple initial distribution, and \mathbf{z}_K is the corresponding random variable after applying the transformations. The properties of normalizing flows allow us to calculate the log-density of \mathbf{z}_K efficiently, using the change of variable theorem:

$$\ln q_K(\mathbf{z}_K) = \ln q_0(\mathbf{z}_0) - \sum_{k=1}^K \ln \left| \det \frac{\partial f_k}{\partial \mathbf{z}_{k-1}} \right|$$

The normalizing flow is defined as the path of the successive distributions q_K .

The first class of flows that we use in our experiments is that of linear flows. More specifically, we consider **planar** and **radial** flows, which perform series of contractions and expansions in the direction perpendicular to a fixed hyperplane and around a reference point respectively.

Theoretical Background

The authors compare their work with **NICE**, which introduces coupling layers. A coupling layer f is a neural network layer with easy to compute inverse and a trivial Jacobian. For an input vector $\mathbf{z} \in \mathbb{R}^D$, we have

$$f(\mathbf{z}) = (\mathbf{z}_A, g(\mathbf{z}_B, h(\mathbf{z}_A))) \quad (1)$$

$$f^{-1}(\mathbf{z}) = (\mathbf{z}_A, g^{-1}(\mathbf{z}_B, h(\mathbf{z}_A))) \quad (2)$$

where (A, B) is a partition of $\{1, 2, \dots, D\}$, h is a neural network with input size $|A|$, and g is a coupling law, a function that is invertible for the first argument given the second. When $g(a, b) = a + b$, the Jacobian is the identity matrix, so f is classified as a volume-preserving flow.

Experiments

We compare expressivity of normalizing flows with NICE coupling layers. We also added in diagonal scaling for NICE layers since this was not accounted for in the original results.

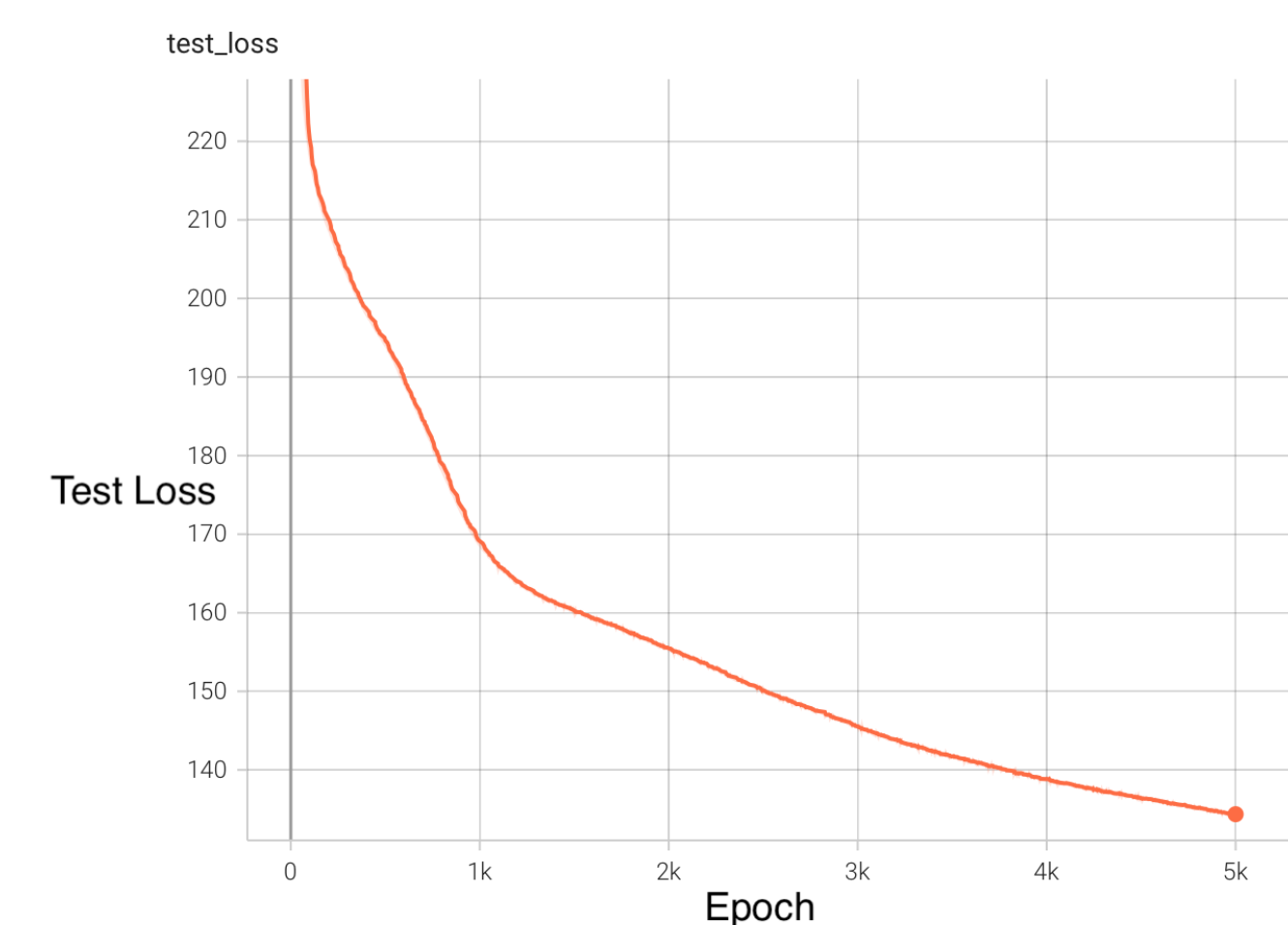
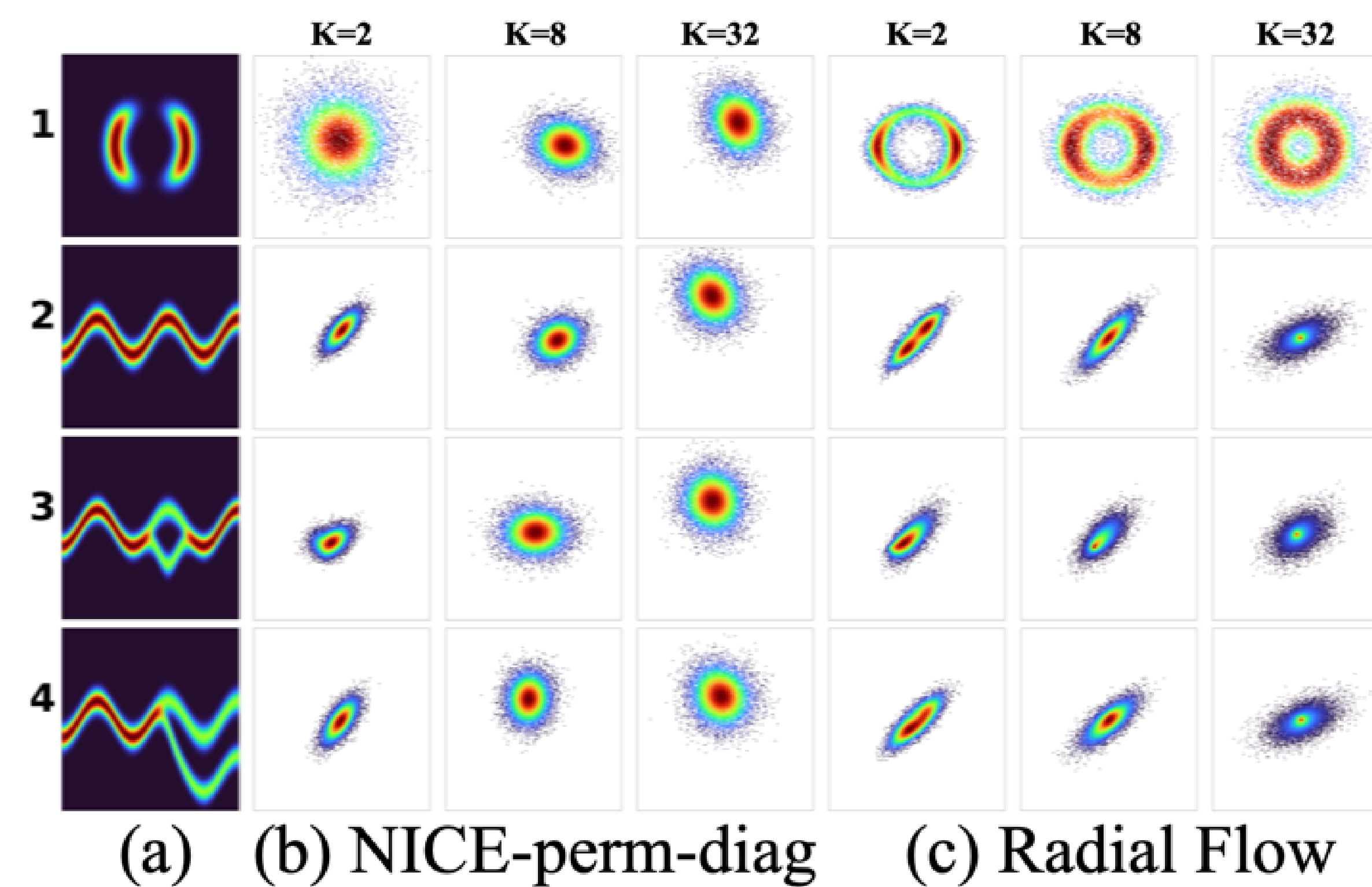


Figure: Test Loss over 5000 epochs using Circular Convolutional Flows

Results

The range of variational bounds is close to those from the paper, but the shapes are visibly inconsistent. We suspect this is due to the “parameter updates” being iterations or epochs, as well as the model being not well-specified in the paper. We do note that among NICE-based models, those with diagonal scaling do perform better than their counterparts.

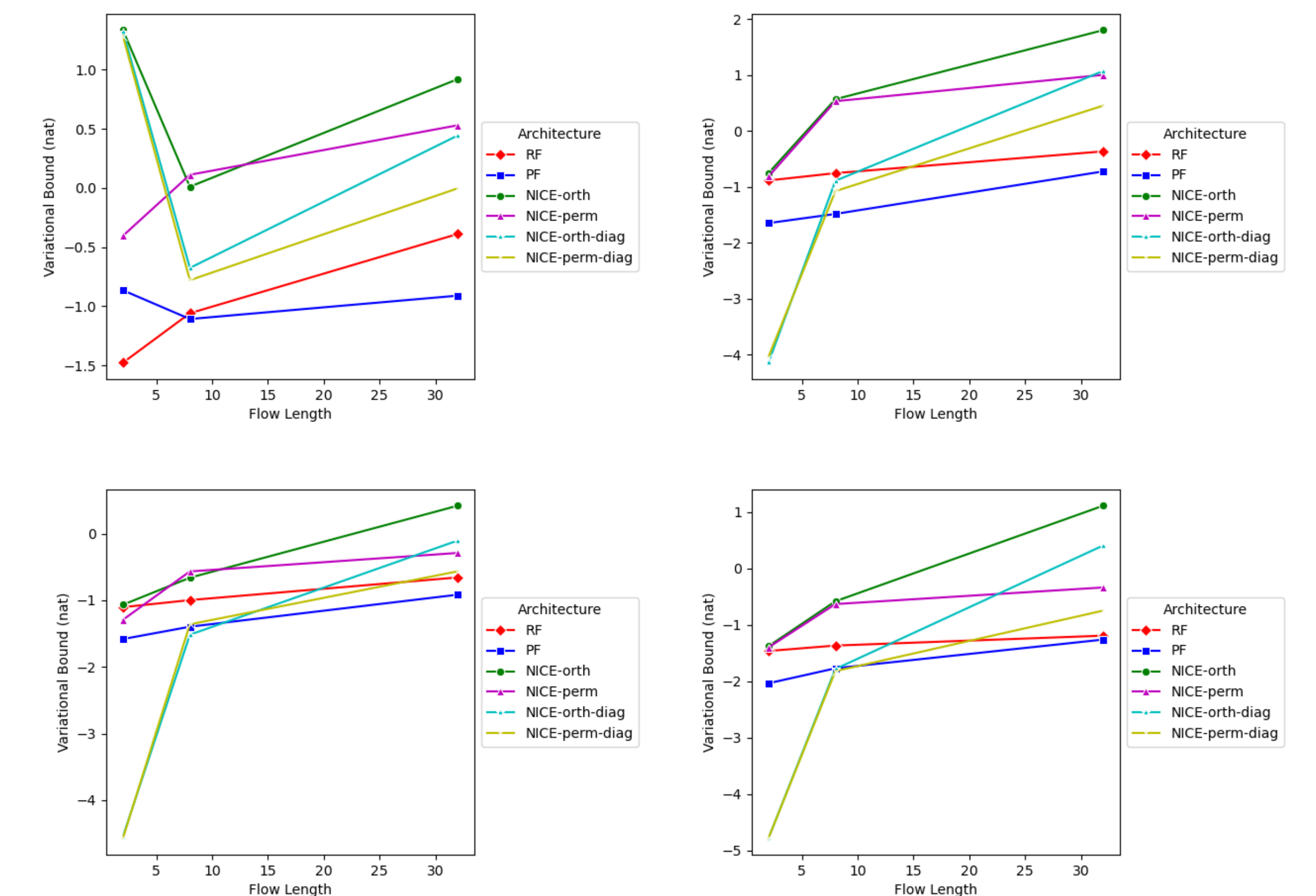


Figure: (d) Variational bounds

Our Improvements and Extensions

To extend the results of the paper, we also looked to what other kinds of flows we could implement. The authors of Invertible Convolutional Flows have used transformations whose determinants are *circulant matrices*, which will allow computation of the log determinant, as well as the convolution and its inverse, to be calculated in time $\mathcal{O}(N \log N)$ by making use of Discrete Fourier Transforms (DTFs). Since it had no public implementations, we decided to implement this - Circular Convolutional Flows.

References

- [1] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2015.
- [2] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2014.