NEURAL INVERSE CDF SAMPLING

ANDREW STIRN: andrew.stirn@columbia.edu
TONY JEBARA: jebara@cs.columbia.edu

THE PROBLEM

Variational Bayesian inference frequently relies on Monte Carlo (MC) integration when intractable expectations appear in the evidence lower bound (ELBO). Variational inference evaluates these expectations wrt the proposed posterior approximations. In many cases, such as in Variational Auto-Encoders, we maximize the ELBO via gradient methods, however, to do so in conjunction MC integration requires that samples from the posterior approximations are differentiable wrt their respective parameters. Closed-form "reparameterization tricks" unfortunately only exist for a subset of distribution families (e.g. Gaussian).

INVERSE CDF SAMPLING

For continuous univariate distribution, inverse CDF sampling allows us to reparameterize a sample from a Uniform distribution to the target distribution:

$$\epsilon \sim \text{Uniform}(0,1)$$
$$z = F^{-1}(\epsilon;\ \theta)$$

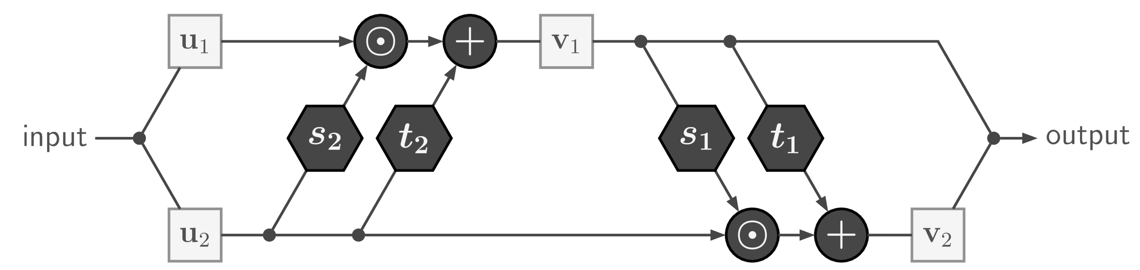
Unfortunately, not all continuous univariate distributions have a closed form quantile function, F^{-1}. In these cases, $\frac{\partial z}{\partial \theta}$ does not exist, prohibiting gradient methods.

OUR CONTRIBUTIONS

We leverage invertible neural network (INN) layers [1] to approximate continuous univariate CDF's, allowing for inverse CDF sampling while ensuring the existence of $\frac{\partial z}{\partial \theta}$. Because the INN blocks are analytically invertible, our approximation maintains *monotonicity* and *one-to-oneness* ensuring: $z = \hat{F}^{-1}(\hat{F}(z;\theta);\theta)$.

INVERTIBLE NEURAL NETWORK: AFFINE COUPLING LAYER

Forward Operation [2]

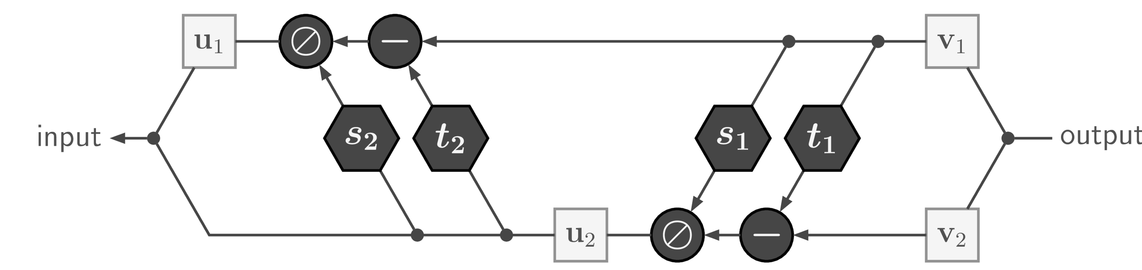


Bifurcate Input $= [\boldsymbol{u_1}, \boldsymbol{u_2}]$

$$\boldsymbol{v_1} = \boldsymbol{u_1} \odot \exp(s_2(\boldsymbol{u_2})) + t_2(\boldsymbol{u_2})$$
$$\boldsymbol{v_2} = \boldsymbol{u_2} \odot \exp(s_1(\boldsymbol{v_1})) + t_1(\boldsymbol{v_1})$$

Recombine Output $= [\boldsymbol{v_1}, \boldsymbol{v_2}]$

Inverse Operation [2]



Bifurcate Output $= [\boldsymbol{v_1}, \boldsymbol{v_2}]$

$$\boldsymbol{u_2} = (\boldsymbol{v_2} - t_1(\boldsymbol{v_1})) \odot \exp(-s_1(\boldsymbol{v_1}))$$
$$\boldsymbol{u_1} = (\boldsymbol{v_1} - t_2(\boldsymbol{u_2})) \odot \exp(-s_2(\boldsymbol{u_2}))$$

Recombine Input $= [\boldsymbol{u_1}, \boldsymbol{u_2}]$

The main requirement of this layer is that $\dim(\boldsymbol{u_i}) = \dim(\boldsymbol{v_i})$, such that, when cascading these layers, dimensions must be fixed and identical across layers. The main benefit of this layer is that invertibility is maintained even when s_i and t_i are non-invertible functions (i.e. affine + ReLU).

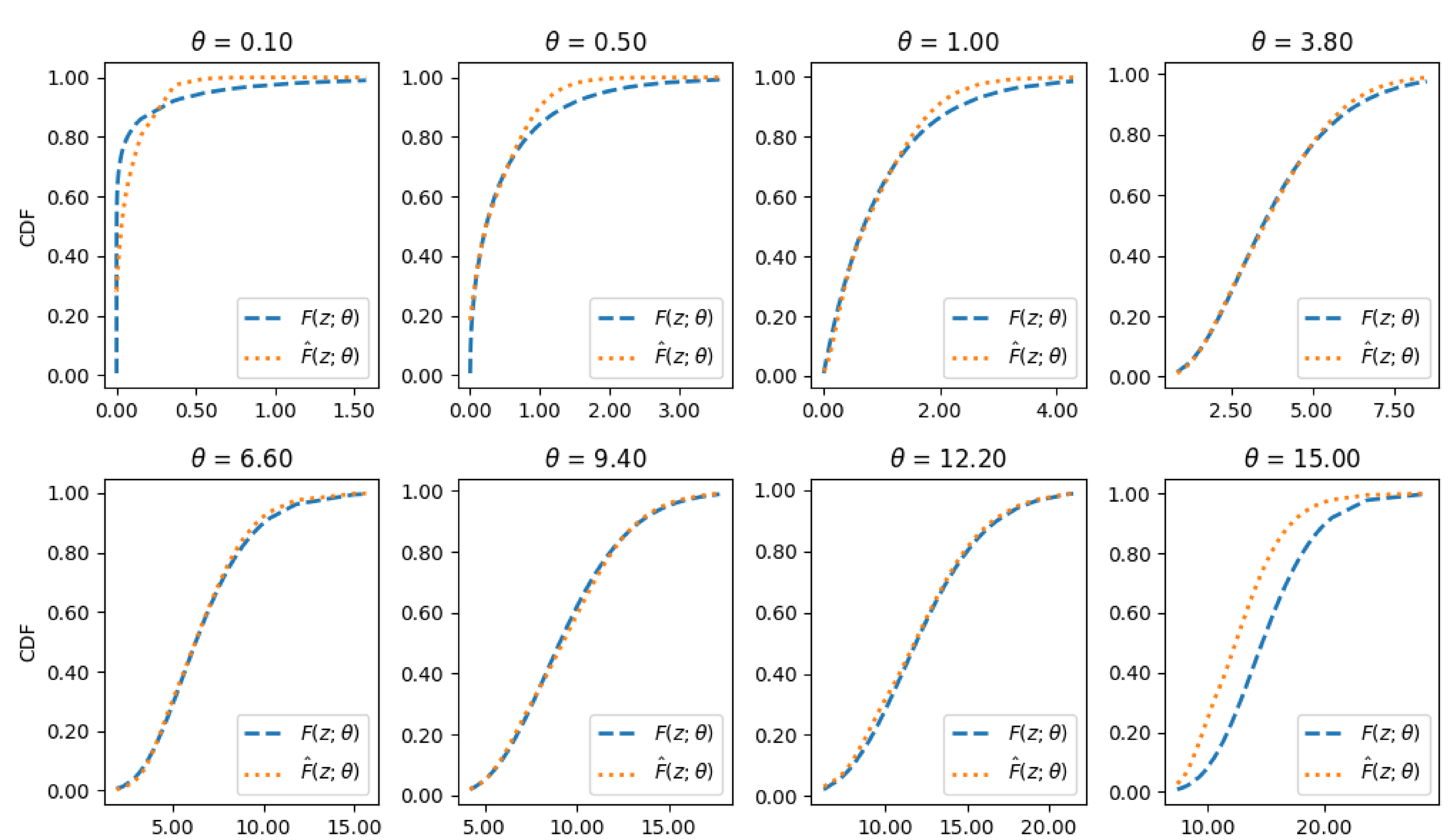
INN MODIFICATIONS

Replacing $\exp(\cdot)$ with $b^{(\cdot)}$, improves numerical stability ($b < e$) and maintains invertibility. Let $f_i(x) = \text{elu}(W_{f_i} \cdot x + b_{f_i})$, where $f \in \{s, t\}$, and W_{f_i} and b_{f_i} are the outputs of multi-layer perceptrons operating on the distribution parameters θ. Let loss be $|F(z_i;\ \theta_n) - \hat{F}(z_i;\ \theta_n)|$.

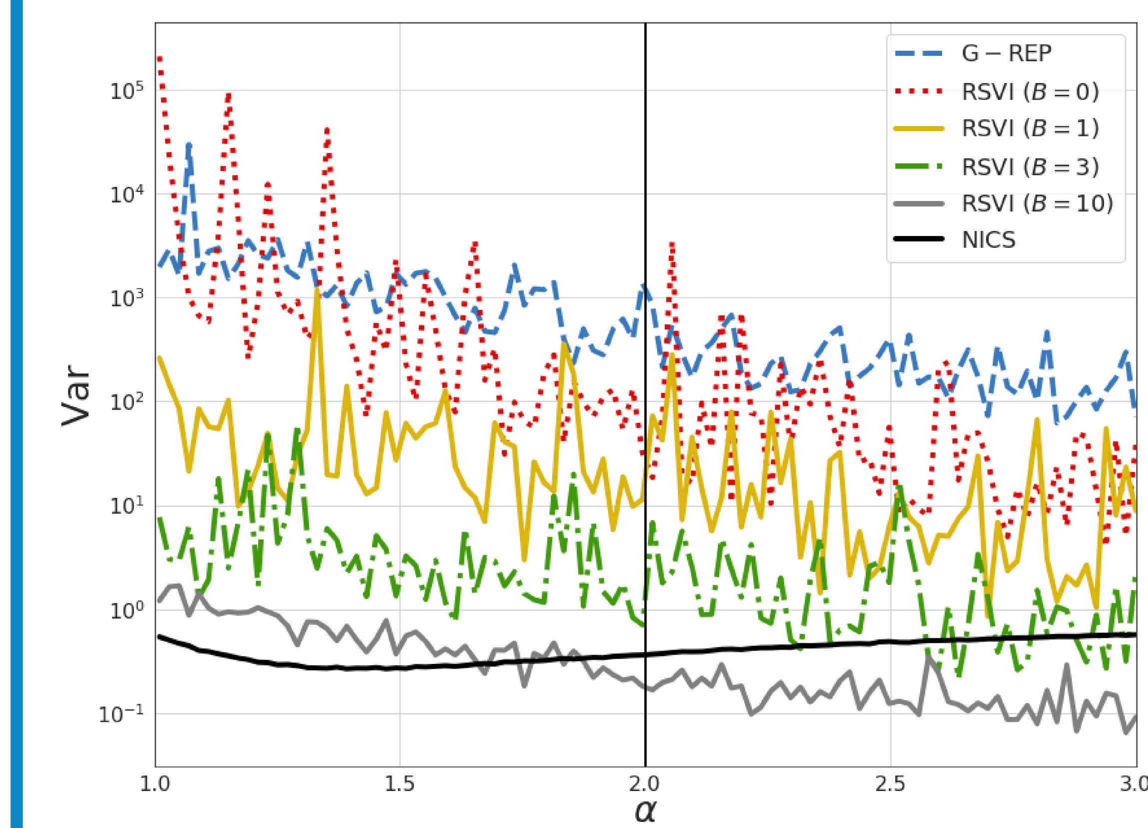
GAMMA CDF TRAINING BATCH

Initialize batch $\mathcal{B} = \{\emptyset\}$
for $n \in \{1, \ldots, N_{\text{batch}}\}$ **do**
  Sample: $\theta_n \overset{iid}{\sim} p(\theta)$
  for $i \in \{1, \ldots, N_z\}$ **do**
    Sample: $z_i \overset{iid}{\sim} \text{Gamma}(\theta_n, 1)$
    Append: $\mathcal{B} := \mathcal{B} \cup (z_i, \theta_n, F(z_i;\ \theta_n))$
  end for
end for

CDF APPROXIMATION FOR GAMMA(θ,1) DISTRIBUTION: $\theta \in (0, 15]$



GRADIENT REPARAMETERIZATION



RESULTS

Perplexity Scores	Softmax	NGS
NVLDA	1099.96	1103.77
ProDLDA	1161.69	1099.94

REFERENCES & CODE

[1] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," *CoRR*, vol. abs/1605.08803, 2016. [Online]. Available: http://arxiv.org/abs/1605.08803

[2] V. L. L. Heidelberg, "Analyzing inverse problems with invertible neural networks," 2018. [Online]. Available: https://hci.iwr.uni-heidelberg.de/vislearn/inverse-problems

Source code available at: `github.com/astirn/neural-inverse-cdf-sampling`