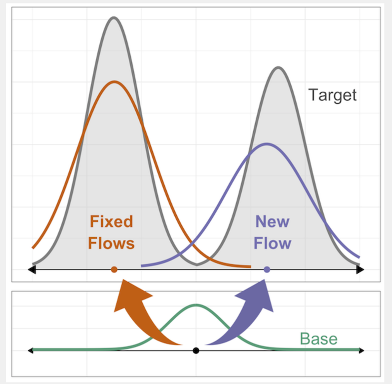
Gradient Boosted Normalizing Flows

Robert Giaquinto   Arindam Banerjee

University of Minnesota, Twin Cities

Summary

- A *wider* approach to training normalizing flows (NF).
- Successively add NF components trained with boosting.
- Resembles a mixture, but has many advantages.
- For density estimation and variational inference.



Normalizing Flows

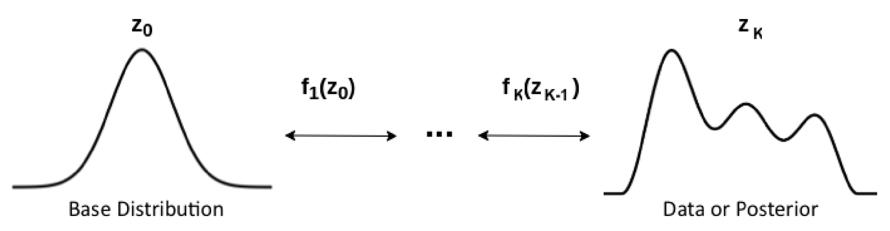


Figure 1. Flows create flexible distributions via a chain of smooth, invertible mappings.

- Exact and efficient likelihood computation, and data generation [3].
- Change of variables on sequence of K invertible transformations:

$$\log p_X(\mathbf{x} = \mathbf{z}_K) = \log p_Z(\mathbf{z}_0) + \sum_{k=1}^{K} \log \left| \det \frac{d\mathbf{z}_k}{d\mathbf{z}_{k-1}} \right|$$

- Recent trend: deeper, more complex transformation chains.

Training Flows with Gradient Boosting

Component c=1: Fit with traditional objective — no boosting!

Component c>1: Have fixed components $G^{(c-1)}$, and new component $g^{(c)}$

1. Train $g^{(c)}$ via Frank-Wolfe linear approximation [1].
2. Optimize component weight $\rho_c \in [0,1]$, ensures a valid probability.

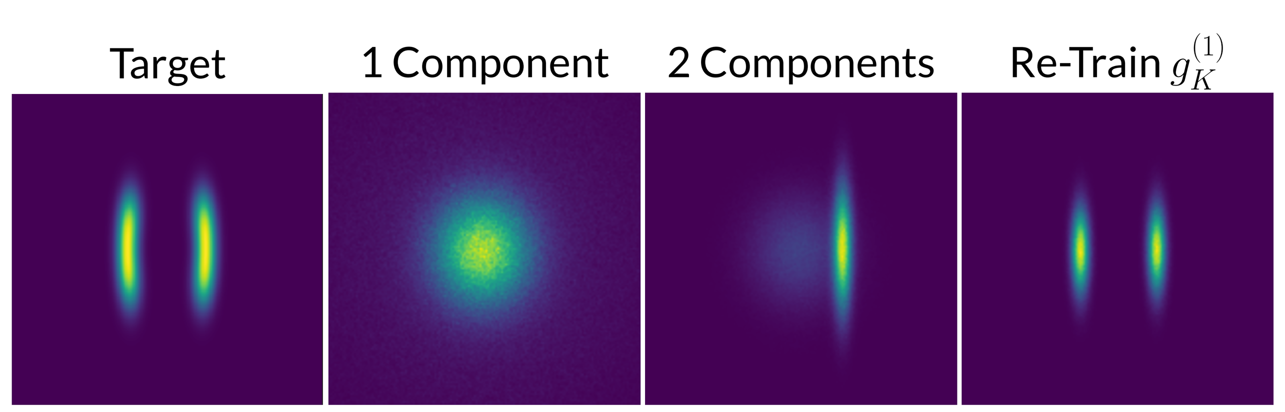


Figure 2. Toy example. A simple `scale+shift` flow cannot model the target and leads to mode-covering (**1 Component**). GBNF introduces a second component (**2 Components**) which seeks a region of high probability that is not well modeled by the first component. Additional *boosted* training corrects the first component, shifting it to the left ellipsoid (**Re-Train** $g_K^{(1)}$).

Gradient Boosted Normalizing Flows

Model: Convex combination of fixed and new flow components:

$$G_K^{(c)}(\mathbf{x}) = \psi\left((1-\rho_c)\psi^{-1}(G_K^{(c-1)}(\mathbf{x})) + \rho_c \psi^{-1}(g_K^{(c)}(\mathbf{x})) \right) / \Gamma_{(c)}$$

- Components assigned weight $\rho_c \in [0,1]$.
- Special Cases:
  1. $\psi(a) = a \implies$ *additive* mixture model (and partition function $\Gamma_{(c)} = 1$).
  2. $\psi(a) = \exp(a) \implies$ *multiplicative* mixture model.

Advantages:

- Flexibility increases by adding more components.
  - An alternative to more complex, deeper flows.
- Mixture-like structure, but focus is only on learning a new $g_K^{(c)}$ (easier!).
- Fast, parallel sampling and prediction.
- Compliments many existing normalizing flows.

Density Estimation with Multiplicative GBNF

Goal: Minimize $KL\left(p^*(\mathbf{x}) \,||\, G_K^{(c)}(\mathbf{x})\right)$, for finite samples $\{\mathbf{x}_i\}$ corresponds to:

$$\text{Loss: } \mathcal{F} = -\frac{1}{N}\sum_{i=1}^{N}\left[\left(\log(G_K^{(c-1)}(\mathbf{x}_i)) + \rho_c \log(g_K^{(c)}(\mathbf{x}_i)) \right) - \log \Gamma_{(c)} \right]$$

Approach: Fit each new $g_K^{(c)}$ based on gradient boosting, yields objective:

$$g_K^{(c)} = \arg\max_{g_K \in \mathcal{G}_K} \mathbb{E}_{p^*}\left[\log g_K(\mathbf{x})\right] - \log \mathbb{E}_{G_K^{(c-1)}}\left[g_K(\mathbf{x})\right]$$

Direct solution:

$$g_K^{(c)}(\mathbf{x}) = \frac{p^*(\mathbf{x})}{G_K^{(c-1)}(\mathbf{x})}$$

$\implies$ More attracted to data that are poorly modeled by fixed components $G_K^{(c-1)}$.

Variational Inference with GBNF

Goal: Augment VAE with a GBNF posterior $G_K^{(c)}$. Minimize negative-ELBO:

$$\mathcal{F}(\mathbf{x}) = \mathbb{E}_{G_K^{(c)}}\left[\log G_K^{(c)}(\mathbf{z}_K \mid \mathbf{x}) - \log p_\theta(\mathbf{x}, \mathbf{z}_K) \right]$$

Approach: Fit new $g_K^{(c)}$ to $\nabla_G \mathcal{F}(\mathbf{x}_i)$, the functional gradient w.r.t. $G_K^{(c)}$ at $\rho_c \to 0$:

$$g_K^{(c)} = \arg\min_{g_K \in \mathcal{G}_K} \sum_{i=1}^{n} \mathbb{E}_{g_K(\mathbf{z}_K|\mathbf{x}_i)}\left[\nabla_G \mathcal{F}(\mathbf{x}_i)\right]$$

- Must add entropy regularization $\mathbb{E}_{g_K(\mathbf{z}_K|\mathbf{x}_i)}[\lambda \log g_K(\mathbf{z}_K \mid \mathbf{x}_i)]$ to avoid degenerate solutions [2], controlled by hyperparameter $\lambda > 0$.
- Similar to training VAE, but down-weight reconstructions explained by $G_K^{(c-1)}$

"Decoder Shock:" Abrupt Changes to Approximate Posterior

Problem: VAE decoder is shared by all GBNF components—what happens when a new component is introduced?

- Sudden shift in posterior distribution leads to (temporary) jumps in reconstruction loss by decoder.
- Unique to VAEs augmented by GBNF.

Solution: Periodically sample from the fixed components, helping the decoder remember past components.

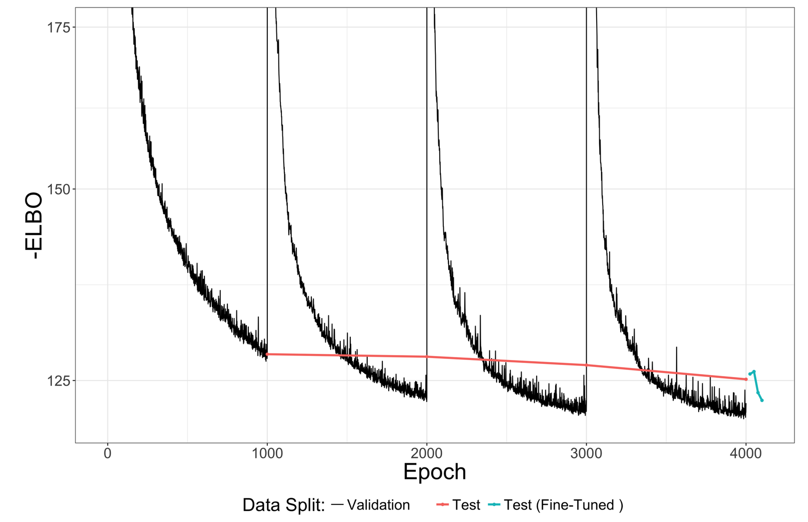


Figure 3. "Decoder Shock." Loss jumps when a new component is added, coinciding with sudden change in samples passed to decoder.

Experiments

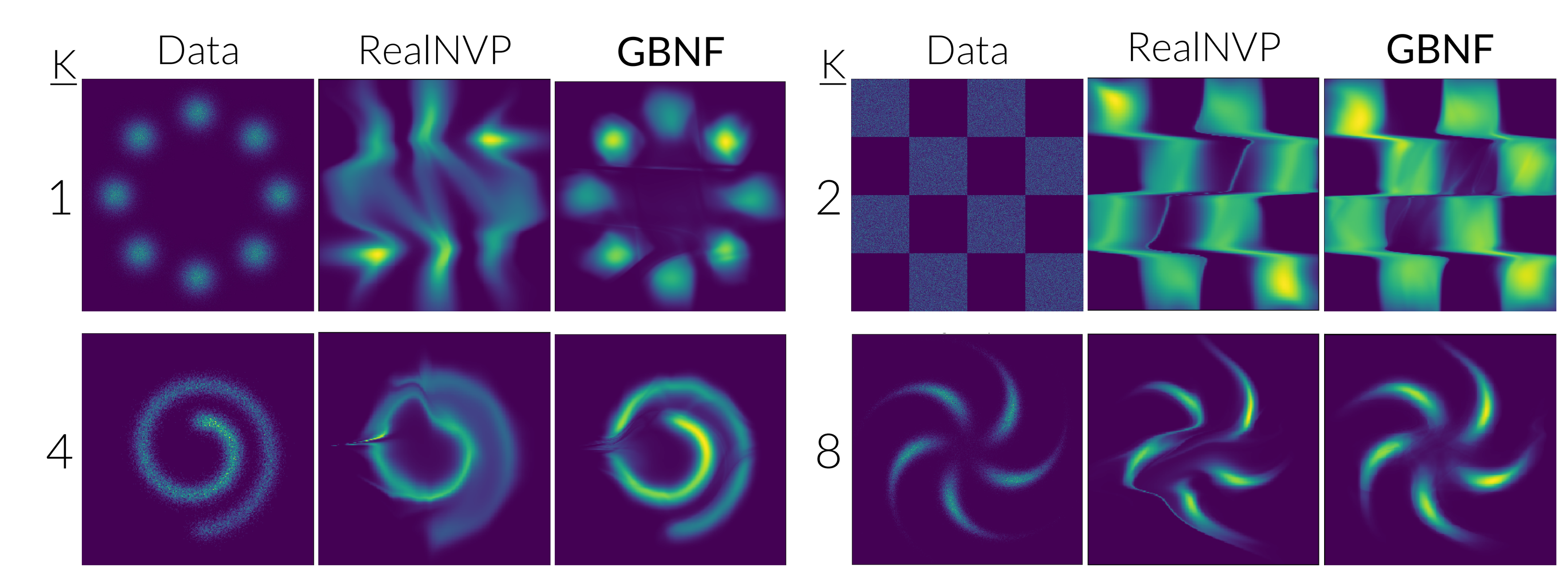


Figure 4. Density estimation for 2D toy data. For **GBNF** each component is a RealNVP flow with $K = 1, 2, 4$ or 8 flow steps. The single **RealNVP** is equivalent to GBNF's first component.

Model	POWER↑ d=6;n=2,049,280	GAS↑ d=8;n=1,052,065	HEPMASS↑ d=21;n=525,123	MINIBOONE↑ d=43;n=36,488	BSDS300↑ d=63;n=1,300,000
RealNVP	$0.17_{\pm.01}$	$8.33_{\pm.14}$	$-18.71_{\pm.02}$	$-13.55_{\pm.49}$	$153.28_{\pm1.78}$
Boosted RealNVP	$0.27_{\pm0.01}$	$9.58_{\pm.04}$	$-18.60_{\pm0.06}$	$-10.69_{\pm0.07}$	$154.23_{\pm2.21}$
Glow	$0.17_{\pm.01}$	$8.15_{\pm.40}$	$-18.92_{\pm.08}$	$-11.35_{\pm.07}$	$155.07_{\pm.03}$
Boosted Glow	$0.24_{\pm0.01}$	$9.95_{\pm0.11}$	$-17.81_{\pm0.12}$	$-10.76_{\pm0.02}$	$154.68_{\pm0.34}$

Table 1. Density estimation. Log-likelihood on tabular data.

Model	MNIST↓	MNIST↓	MNIST↓	MNIST↓
VAE	$84.97_{\pm0.01}$	$4.78_{\pm0.07}$	$103.16_{\pm0.01}$	$108.43_{\pm1.81}$
Planar	$83.16_{\pm0.07}$	**4.60** $_{\pm0.04}$	**100.18** $_{\pm0.01}$	**104.23** $_{\pm1.60}$
Sylvester	**81.99** $_{\pm0.02}$	**4.49** $_{\pm0.03}$	**98.54** $_{\pm0.29}$	**100.38** $_{\pm1.20}$
IAF	**83.14** $_{\pm0.06}$	$4.70_{\pm0.05}$	$100.97_{\pm0.07}$	$108.41_{\pm1.31}$
RealNVP	$83.36_{\pm0.09}$	$4.62_{\pm0.16}$	$100.43_{\pm0.19}$	$113.00_{\pm1.70}$
GBNF	**82.59** $_{\pm0.03}$	**4.41** $_{\pm0.01}$	**99.09** $_{\pm0.17}$	**106.40** $_{\pm0.54}$

Table 2. Variational inference. Negative log-likelihood for image data.

References and Acknowledgements

The research was supported by NSF grants OAC-1934634, IIS-1908104, IIS-1563950, IIS-1447566, IIS-1447574, IIS-1422557, CCF-1451986. We thank the University of Minnesota Supercomputing Institute for technical support.

[1] Zac Cranko and Richard Nock. Boosting Density Estimation Remastered. *Proceedings of the 36th International Conference on Machine Learning*, 97:1416–1425, 2019.

[2] Fangjian Guo, Xiangyu Wang, Kai Fan, Tamara Broderick, and David B. Dunson. Boosting Variational Inference. In *Advances in Neural Information Processing Systems*, Barcelona, Spain, 2016.

[3] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing Flows for Probabilistic Modeling and Inference. *arXiv:1912.02762 [cs, stat]*, December 2019.