Semi-Conditional Normalizing Flows for Semi-Supervised Learning

Andrei Atanov, Alexandra Volokhova, Arsenii Ashukha,
Ivan Sosnovik, Dmitry Vetrov

SAMSUNG AI Center - Moscow

p(B|A)yesgroup.ru

Contributions

Propose a ***Semi-Conditional Normalizing Flows***.

- **Efficiently** evaluate marginal distribution of unlabelled data.
- **Flexible** conditional coupling layers.
- **Data obfuscation** by class-independent latent representations.

Semi-Supervised Learning with Deep Generative models

- Model joint distribution over object and label:

$$p_\theta(x, y) = p_\theta(x|y)p(y)$$

Let's use Normalizing Flows as they provide **tractable log-likelihood** and its gradients estimations!

- Use marginal density for unlabelled objects:

$$p_\theta(x) = \sum_{k=1}^{K} p_\theta(x, y=k)$$

Inference for unlabelled objects is **K times slower!**
Let's **condition only a small part** of a flow!

- **Objective** for training model parameters:

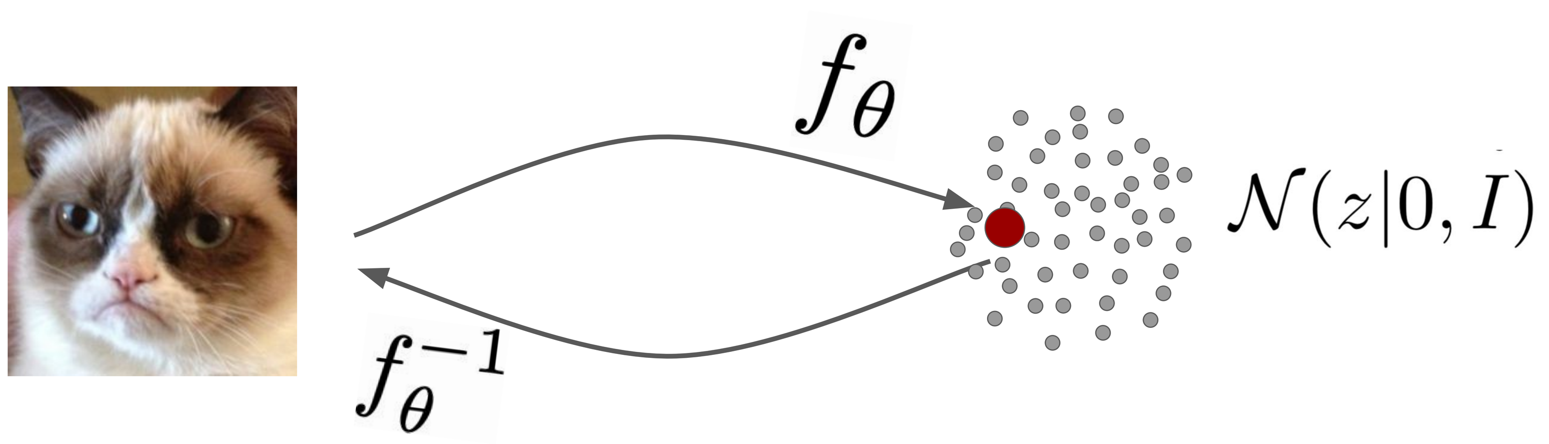
$$L(\theta) = \underbrace{\sum \log p_\theta(x_i, y_i)}_{\textit{labelled}} + \underbrace{\sum \log p_\theta(x_j)}_{\textit{unlabelled}}$$

- **Prediction** can be done by the posterior:

$$p_\theta(y|x) = \frac{p_\theta(x,y)}{p_\theta(x)}$$

Normalizing Flows

- Model data as an invertible transformation of a simple random variable:



- Change of variables formula to compute the density:

$$\log p_\theta(x|y) = \log\left|\frac{\partial f_\theta(x; y)}{\partial x^T}\right| + \log p(z)$$

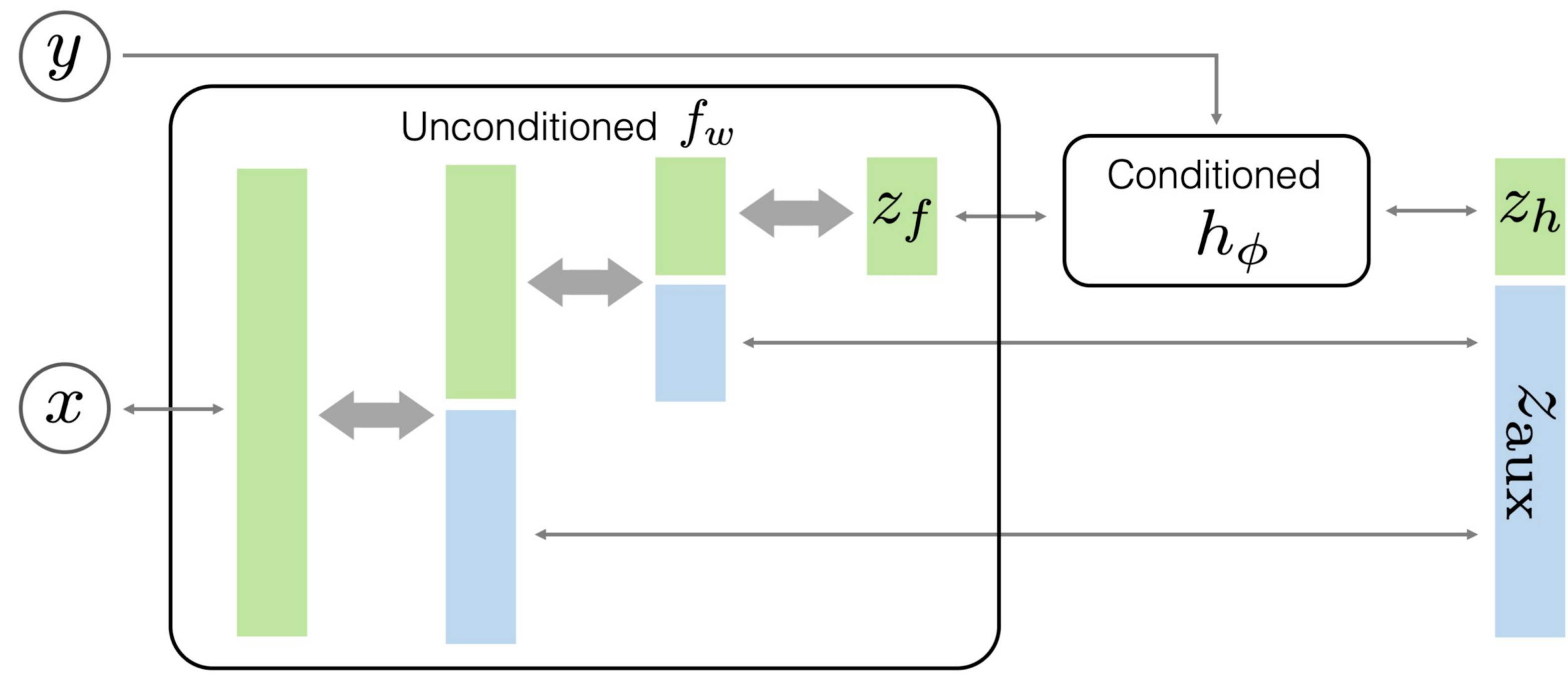
- Use conditional coupling layer for flexible conditioning on a class label:

$$z_1 = x_1, \quad z_2 = x_2 \odot \exp\left(s\left(x_1, \mathbf{y}\right)\right) + t\left(x_1, \mathbf{y}\right)$$

arbitrary neural networks

Semi-Conditional Normalizing Flows

- Condition only a small part of a flow to decrease the dimension of a representation and speed up inference.



- Single pass of large flow to compute marginal density:

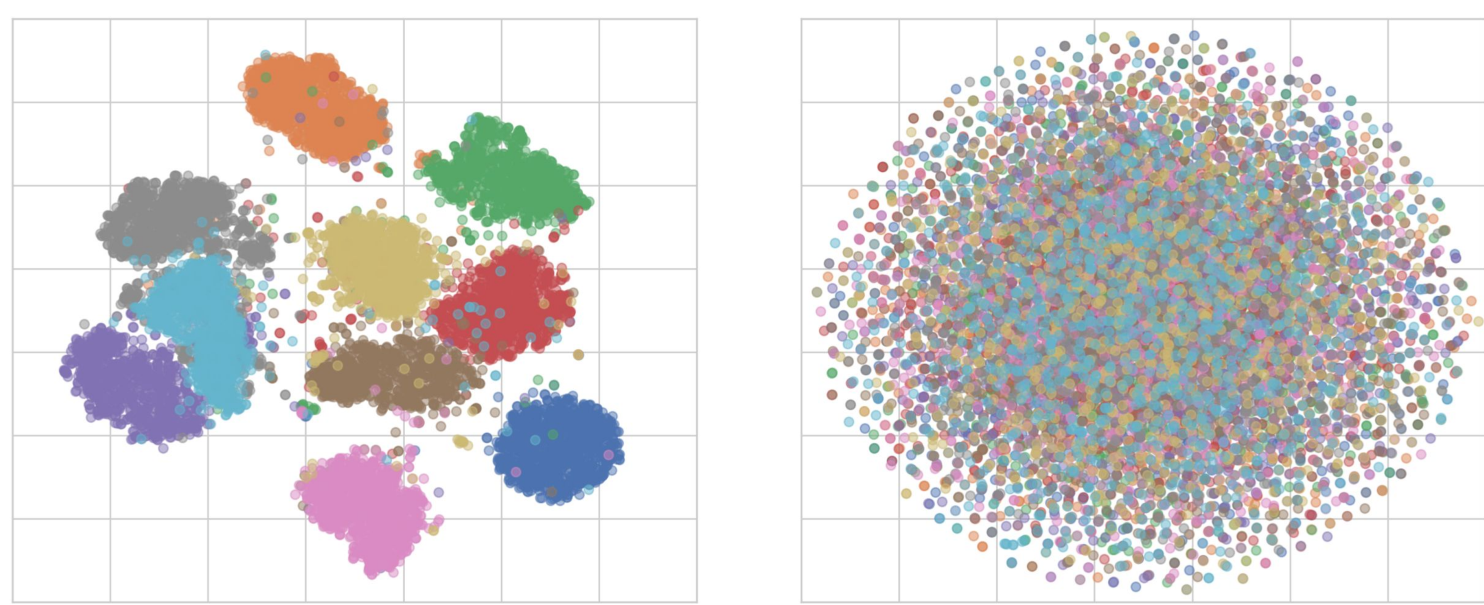
$$\log p_\theta(x) = \overbrace{\boxed{\log\left|\frac{\partial f_w(x)}{\partial x^T}\right|}}^{\text{high cost}} + \log \mathcal{N}(z_{\text{aux}}|0, I)$$
$$+ \sum_{y=1}^{K} \underbrace{\boxed{\log\left|\frac{\partial h_\phi(z_f; y)}{\partial z_f^T}\right|}}_{\text{low cost}} + \log\mathcal{N}(z_h|0, I) + \log p(y)$$

MNIST Semi-Supervised Classification

Model	Optimisation	L_{clf}	Error, %	Bits/dim
Kingma et al. (2014)	VI	✓	3.3 ± 0.1	-
SCNF-GLOW (Ours)	SGD	✗	1.9 ± 0.3	1.145 ± 0.004
	EM-SGD	✓	2.0 ± 0.1	1.151 ± 0.010
		✗	1.9 ± 0.0	1.146 ± 0.002
SCNF-GMM (Ours)	SGD	✗	14.2 ± 2.4	1.143 ± 0.011
	EM-SGD	✓	16.9 ± 5.3	1.141 ± 0.006
		✗	13.4 ± 2.8	1.145 ± 0.005

- **100 labelled** objects (the rest are unlabelled).
- **EM-SGD** --- an Expectation Maximization algorithm for maximizing marginal likelihood.
- Kingma et al. (2014) --- stacked M1+M2 model based on variational autoencoders.

Semi-Supervised Data Obfuscation

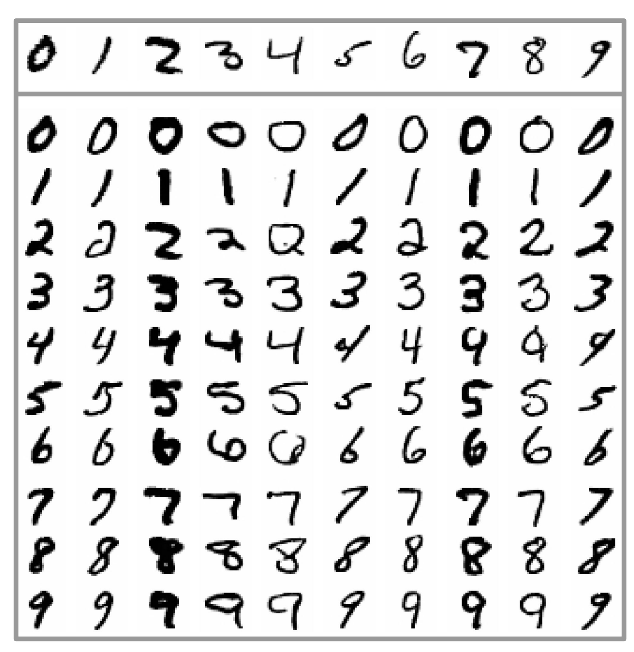


(a) t-SNE Embeddings of z_f (b) t-SNE Embeddings of z_h

- The objective favours z_h to be independent of class variable:

$$\log p_\theta(z_f|y) = \underbrace{\boxed{\log\left|\frac{\partial h_\phi(z_f; y)}{\partial z_f^T}\right|}}_{\text{enforce increasing volume}} + \underbrace{\boxed{\log\mathcal{N}(z_h|0, I)}}_{\text{push z in a high density region}}$$

- However, z_h contains all the other information since the invertibility.



Reconstructions with different class labels