



Boulder

# Generative Networks; Variational Auto-Encoders



[YouTube Playlist](#)

**Maziar Raissi**

**Assistant Professor**

Department of Applied Mathematics

University of Colorado Boulder

[maziar.raissi@colorado.edu](mailto:maziar.raissi@colorado.edu)



Boulder

# Auto-Encoding Variational Bayes



[YouTube Video](#)

$X = \{x^i\}_{i=1}^N \rightarrow$  dataset ( $N$  i.i.d samples of some continuous or discrete random variable  $x$ )

$z \rightarrow$  unobserved (latent) continuous random variable

$z^i \rightarrow$  generated from  $\underbrace{p_{\theta^*}(z)}$   
prior distribution

$x^i \rightarrow$  generated from some conditional distribution  $\underbrace{p_{\theta^*}(x|z)}$   
likelihood (generative model or probabilistic decoder)

$p_{\theta}(x) = \int p_{\theta}(z)p_{\theta}(x|z)dz \rightarrow$  marginal likelihood (interactive)

$p_{\theta}(z|x) = \frac{p_{\theta}(x|z)p_{\theta}(z)}{p_{\theta}(x)} \rightarrow$  posterior (interactive)

Image denoising, inpainting, and super-resolution

$\underbrace{q_{\phi}(z|x)}$   $\rightarrow$  an approximation to the posterior distribution  
recognition model (probabilistic encoder)

$$\log p_{\theta}(x^1, \dots, x^N) = \sum_{i=1}^N \log p_{\theta}(x^i)$$

$$\log p_{\theta}(x^i) = \underbrace{KL(q_{\phi}(z|x^i)||p_{\theta}(z|x^i))}_{\geq 0} + \underbrace{\mathbb{E}_{q_{\phi}(z|x^i)}[-\log q_{\phi}(z|x^i) + \log p_{\theta}(x^i, z)]}_{\mathcal{L}(\theta, \phi; x^i)}$$



reparameterization trick

$z \sim q_{\phi}(z|x)$  auxiliary noise variable

$z = \underbrace{g_{\phi}(\epsilon, x)}$  with  $\overbrace{\epsilon \sim p(\epsilon)}$   
differentiable transformation

$$\mathcal{L}(\theta, \phi; x^i) \approx \frac{1}{L} \sum_{\ell=1}^L [\log p_{\theta}(x^i, z^{i,\ell}) - \log q_{\phi}(z^{i,\ell}|x^i)]$$

$$z^{i,\ell} = g_{\phi}(\epsilon^{\ell}, x^i), \epsilon^{\ell} \sim p(\epsilon)$$

Variational Auto-Encoder

$$p_{\theta}(z) = \mathcal{N}(z; 0, I)$$

$$\log q_{\phi}(z|x^i) = \log \mathcal{N}(z; \mu(x^i), \sigma^2(x^i)I)$$

$$g_{\phi}(\epsilon, x) = \mu(x) + \sigma(x)\epsilon$$

$$\epsilon \sim \mathcal{N}(\epsilon; 0, I)$$

$\log p_{\theta}(x|z)$  depends on the type of data  
(e.g., Gaussian or Bernoulli)

6	6	/	7	8	1	4	8	2	8
9	6	8	3	9	6	8	3	1	9
5	3	7	1	3	6	9	1	7	9
8	9	0	8	6	9	1	9	6	3
9	2	3	3	3	1	3	8	6	
6	9	9	8	6	1	6	6	6	
9	5	2	6	6	5	1	8	9	9
9	9	9	1	3	1	2	8	2	3
0	4	6	1	2	3	2	0	8	9
9	7	5	4	9	3	4	8	5	1



Boulder

# Stochastic Backpropagation and Approximate Inference in Deep Generative Models

Deep Latent Gaussian Models (DLGMs)

$\xi_l \sim \mathcal{N}(\xi_l | \mathbf{0}, \mathbf{I})$ ,  $l = 1, \dots, L$   $\rightarrow$  mutually independent Gaussian variables

$$\mathbf{h}_L = \mathbf{G}_L \xi_L,$$

$$\mathbf{h}_l = T_l(\mathbf{h}_{l+1}) + \mathbf{G}_l \xi_l, \quad l = 1 \dots L-1$$

$$\mathbf{v} \sim \pi(\mathbf{v} | T_0(\mathbf{h}_1)),$$

$\mathbf{G}_l \rightarrow$  matrices

$T_l \rightarrow$  MLPs (multi-layer perceptrons)

$\pi \rightarrow$  any appropriate distribution

$\boldsymbol{\theta}^g \rightarrow$  parameters of the MLPs and the matrices

$p(\boldsymbol{\theta}^g) = \mathcal{N}(\boldsymbol{\theta} | \mathbf{0}, \kappa \mathbf{I}) \rightarrow$  Gaussian prior

Scalable Inference in DLGMs

$\mathbf{V} \rightarrow$  full dataset of size  $N \times D$

$\mathbf{v}_n = [v_{n1}, \dots, v_{nD}]^\top \rightarrow$  observations

$$\mathcal{L}(\mathbf{V}) = -\log p(\mathbf{V}) = -\log \int p(\mathbf{V} | \boldsymbol{\xi}, \boldsymbol{\theta}^g) p(\boldsymbol{\xi}, \boldsymbol{\theta}^g) d\boldsymbol{\xi}$$

$$= -\log \int \frac{q(\boldsymbol{\xi})}{q(\boldsymbol{\xi})} p(\mathbf{V} | \boldsymbol{\xi}, \boldsymbol{\theta}^g) p(\boldsymbol{\xi}, \boldsymbol{\theta}^g) d\boldsymbol{\xi}$$

$$\leq \mathcal{F}(\mathbf{V}) = D_{KL}[q(\boldsymbol{\xi}) \| p(\boldsymbol{\xi})] - \mathbb{E}_q [\log p(\mathbf{V} | \boldsymbol{\xi}, \boldsymbol{\theta}^g) p(\boldsymbol{\theta}^g)]$$

$$q(\boldsymbol{\xi} | \mathbf{V}, \boldsymbol{\theta}^r) = \prod_{n=1}^N \prod_{l=1}^L \mathcal{N}(\boldsymbol{\xi}_{n,l} | \boldsymbol{\mu}_l(\mathbf{v}_n), \mathbf{C}_l(\mathbf{v}_n)) \rightarrow \text{recognition model}$$

$$D_{KL}[\mathcal{N}(\boldsymbol{\mu}, \mathbf{C}) \| \mathcal{N}(\mathbf{0}, \mathbf{I})] = \frac{1}{2} [\text{Tr}(\mathbf{C}) - \log |\mathbf{C}| + \boldsymbol{\mu}^\top \boldsymbol{\mu} - D]$$

$$\mathcal{F}(\mathbf{V}) = -\sum_n \mathbb{E}_q [\log p(\mathbf{v}_n | \mathbf{h}(\boldsymbol{\xi}_n))] + \frac{1}{2\kappa} \|\boldsymbol{\theta}^g\|^2$$

$$+ \frac{1}{2} \sum_{n,l} [\|\boldsymbol{\mu}_{n,l}\|^2 + \text{Tr}(\mathbf{C}_{n,l}) - \log |\mathbf{C}_{n,l}| - 1]$$

$$\nabla_{\theta_j^g} \mathcal{F}(\mathbf{V}) = -\mathbb{E}_q [\nabla_{\theta_j^g} \log p(\mathbf{V} | \mathbf{h})] + \frac{1}{\kappa} \theta_j^g$$

$$\nabla_{\boldsymbol{\theta}^r} \mathcal{F}(\mathbf{v}) = ?$$

Stochastic Backpropagation

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\theta}}} [f(\boldsymbol{\xi})]$$

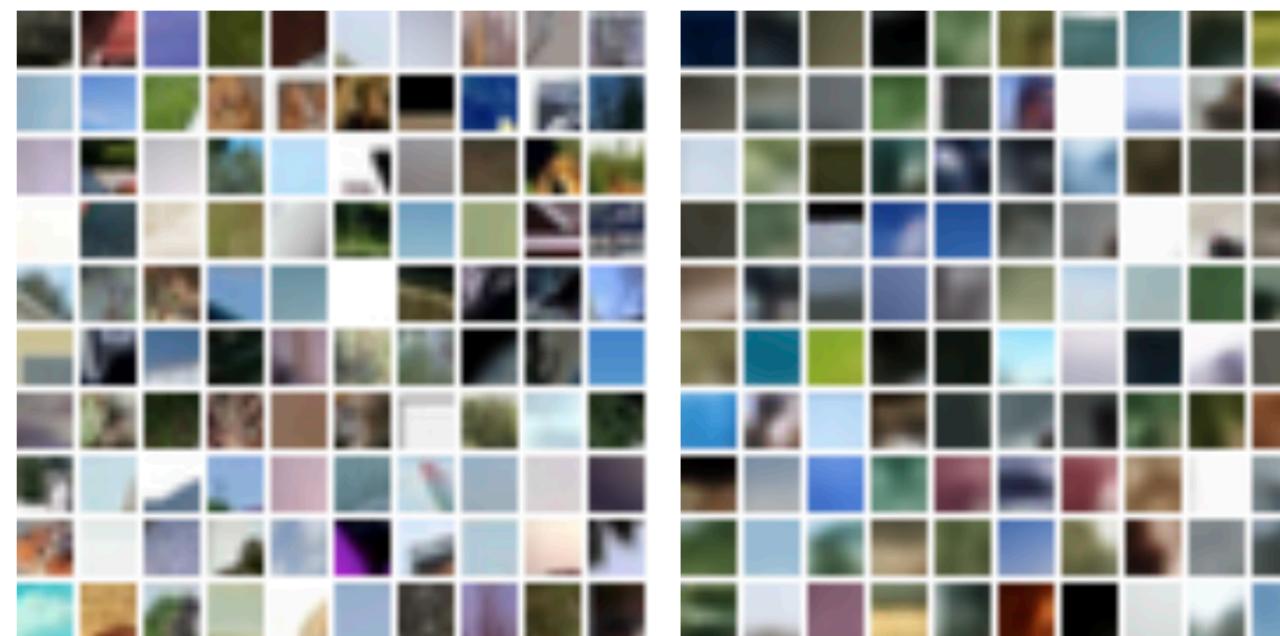
$$f(\boldsymbol{\xi}) = \log p(\mathbf{v} | \mathbf{h}(\boldsymbol{\xi}))$$

$$\nabla_{\mathbf{R}} \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})} [f(\boldsymbol{\xi})] = \nabla_{\mathbf{R}} \mathbb{E}_{\mathcal{N}(0, I)} [f(\boldsymbol{\mu} + \mathbf{R}\boldsymbol{\epsilon})] = \mathbb{E}_{\mathcal{N}(0, I)} [\boldsymbol{\epsilon} \mathbf{g}^\top]$$

$\mathbf{g} \rightarrow$  gradient of the function  $f(\boldsymbol{\xi})$

$$\mathbf{C} = \mathbf{R} \mathbf{R}^\top$$

$$\nabla_{\boldsymbol{\theta}^r} \mathcal{F}(\mathbf{v}) = \nabla_{\boldsymbol{\mu}} \mathcal{F}(\mathbf{v})^\top \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\theta}^r} + \text{Tr} \left( \nabla_{\mathbf{R}} \mathcal{F}(\mathbf{v}) \frac{\partial \mathbf{R}}{\partial \boldsymbol{\theta}^r} \right)$$



(b) CIFAR

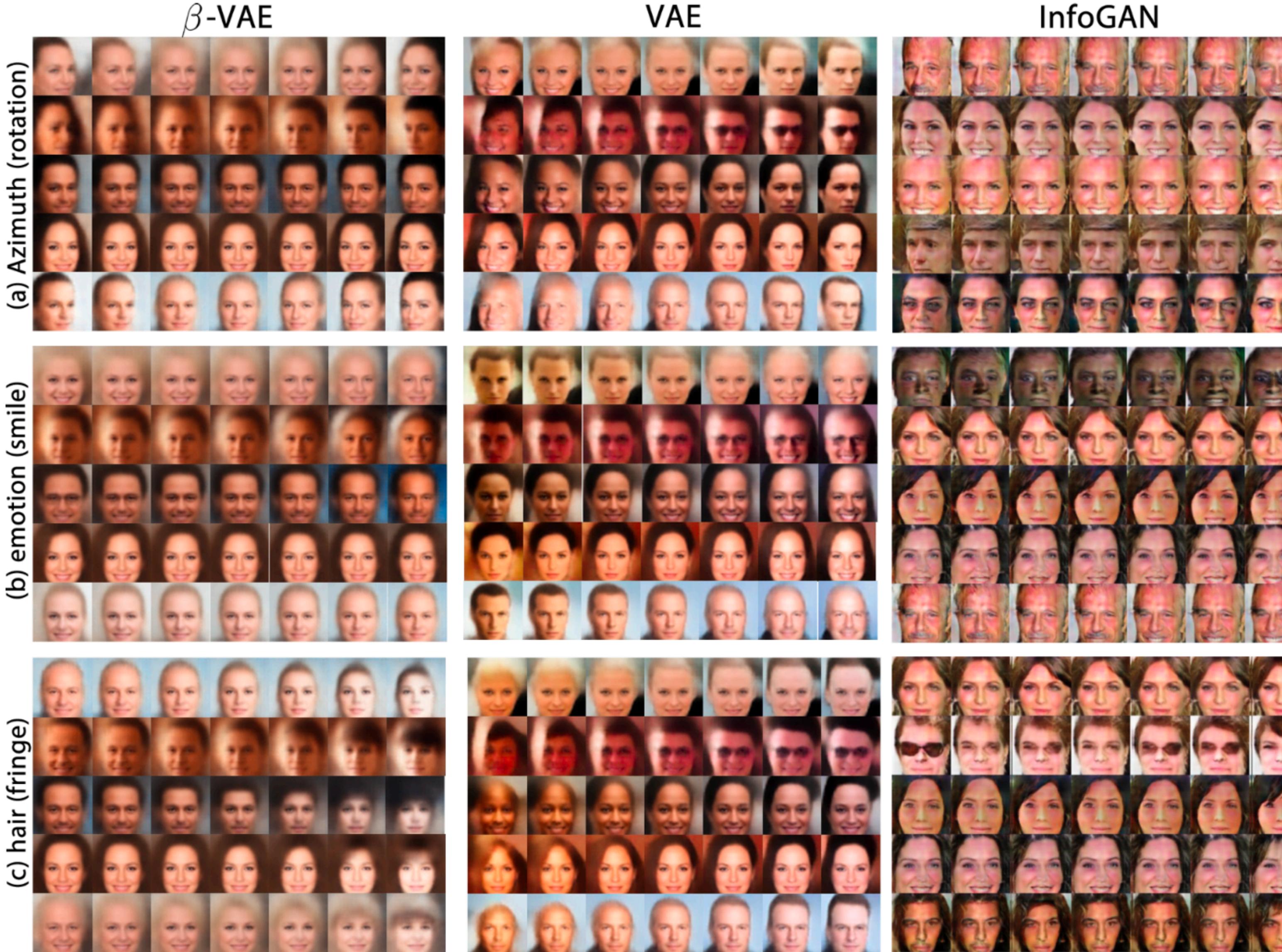


(c) Frey



Boulder

# beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework



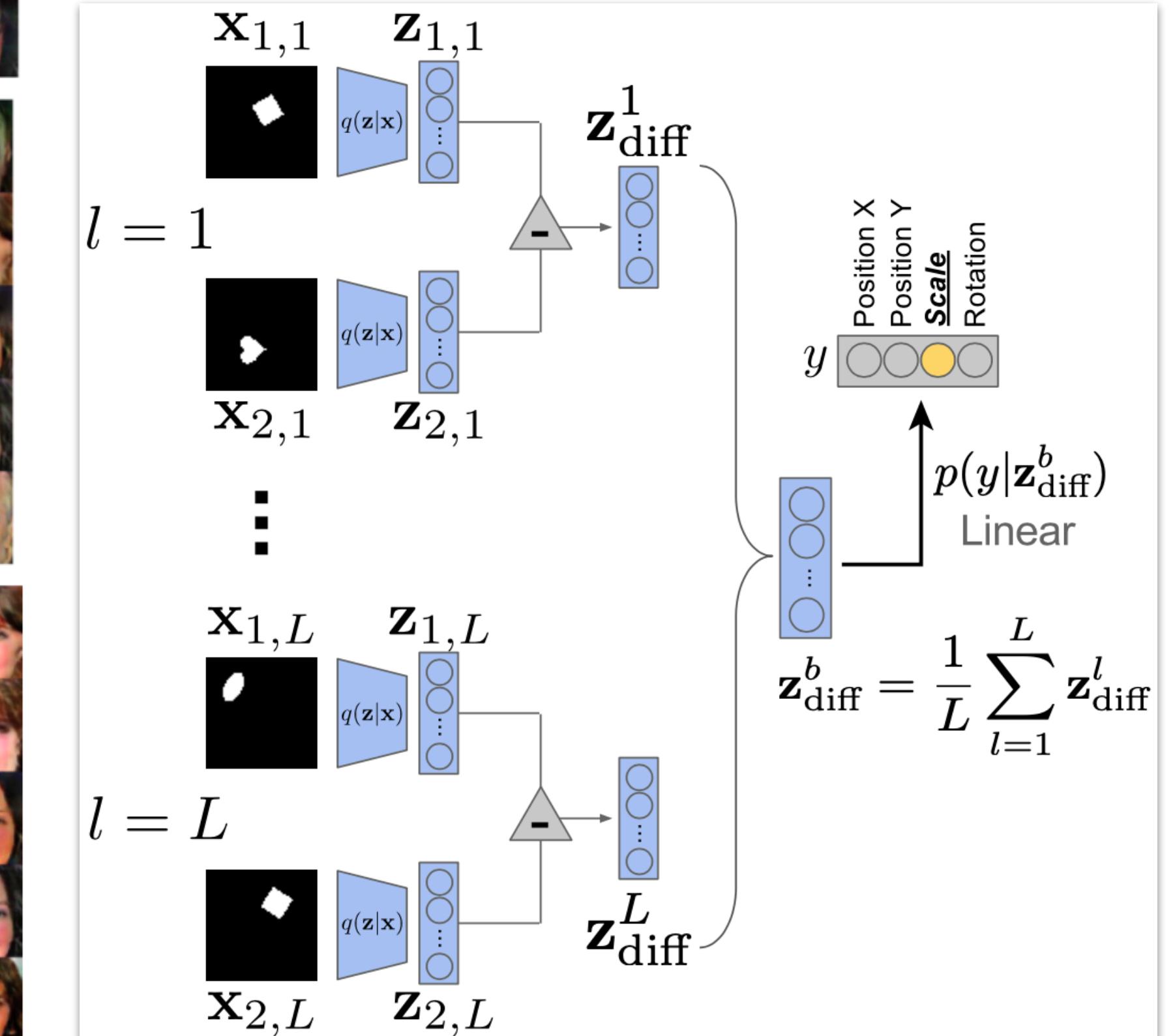
$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}, \beta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \beta D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

set the prior to be an isotropic unit Gaussian ( $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, I)$ )

$\beta \rightarrow$  degree of learned disentanglement

Estimated using either a “disentanglement metric”

(see the paper) in the semi-supervised case or through visual inspection heuristics in the unsupervised case.





Boulder

# Categorical Reparameterization with Gumbel-Softmax

## Score Function-based Gradient Estimators (SF)

REINFORCE or likelihood ratio estimator

$$\nabla_{\theta} p_{\theta}(z) = p_{\theta}(z) \nabla_{\theta} \log p_{\theta}(z)$$

$$\nabla_{\theta} \mathbb{E}_z [f(z)] = \mathbb{E}_z [f(z) \nabla_{\theta} \log p_{\theta}(z)]$$

does not require backpropagating through  $f$  or the sample  $z$

suffers from high variance

$b(z) \rightarrow$  control variate

$$\mu_b = \mathbb{E}_z [b(z) \nabla_{\theta} \log p_{\theta}(z)] \rightarrow$$
 analytical expectation

$$\nabla_{\theta} \mathbb{E}_z [f(z)] = \mathbb{E}_z [(f(z) - b(z)) \nabla_{\theta} \log p_{\theta}(z)] + \mu_b$$

NVIL, DARN, MuProp, VIMCO

## Gumbel-Softmax distribution

$z \rightarrow$  categorical variable with class probabilities

$$\pi = (\pi_1, \dots, \pi_k)$$

$k$ -dimensional one-hot vectors lying on the corners of the  $(k - 1)$ -dimensional simplex  $\Delta^{k-1}$

$$\mathbb{E}_p[z] = [\pi_1, \dots, \pi_k] \rightarrow$$
 element-wise mean

## Gumbel-Max Trick

a simple and efficient way to draw samples  $z$  from a categorical distribution with class probabilities  $\pi$

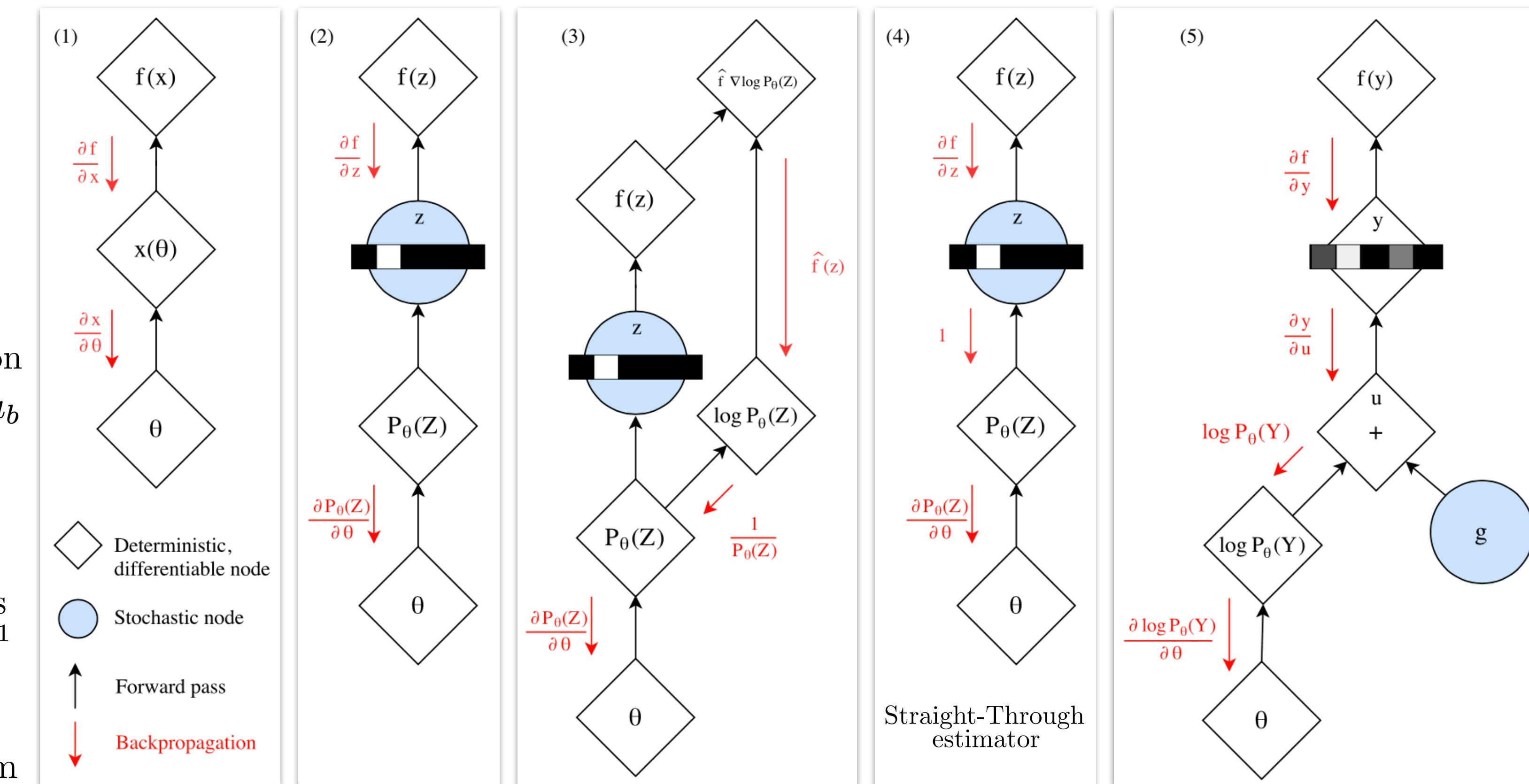
$$z = \text{one\_hot} \left( \arg \max_i [g_i + \log \pi_i] \right)$$

$g_1, \dots, g_k \rightarrow$  i.i.d samples drawn from  $\text{Gumbel}(0, 1)$

inverse transform sampling

$$u \sim \text{Uniform}(0, 1) \implies g = -\log(-\log(u))$$

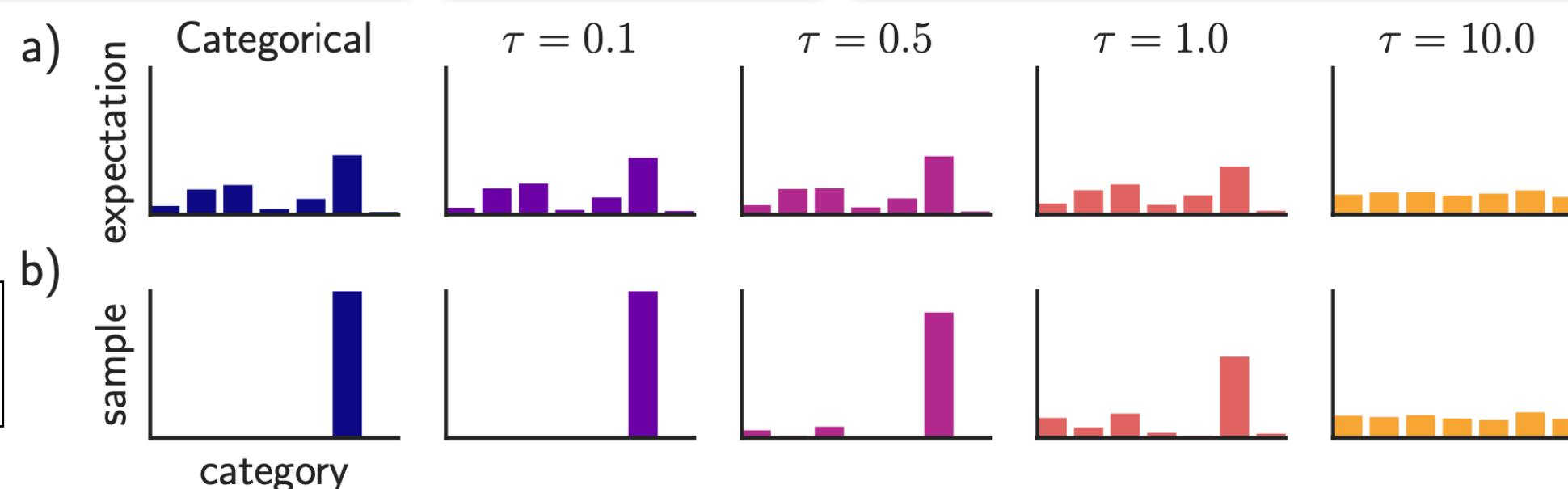
softmax: continuous, differentiable approximation to argmax



$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^k \exp((\log(\pi_j) + g_j)/\tau)}$$

for  $i = 1, \dots, k$ .

$$p_{\pi, \tau}(y_1, \dots, y_k) = \Gamma(k) \tau^{k-1} \left( \sum_{i=1}^k \pi_i / y_i^\tau \right)^{-k} \prod_{i=1}^k (\pi_i / y_i^{\tau+1})$$





Boulder



# Questions?

[YouTube Playlist](#)

---