Variational Auto-Encoders and Extensions

Diederik (Durk) Kingma

Max Welling

UNIVERSITEIT VAN AMSTERDAM

Parts of this talk

1. Basics
2. Extension with auxiliary variables
3. Recent advances

Variational auto-encoders

Basic problem setup

- **We assume:**
  - **A huge dataset** of observations

$$\mathrm{X} = \{\mathrm{x}^1, ..., \mathrm{x}^\mathrm{N}\}$$

  - There exists **a simple latent space** z:

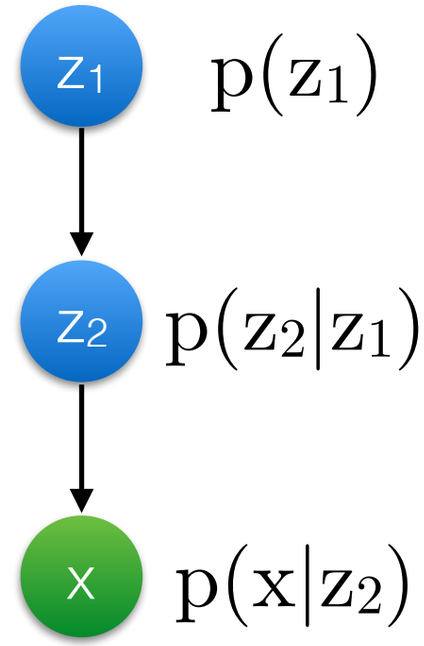
$$\mathrm{z} \sim \mathrm{p}_\Theta(\mathrm{z})$$
$$\mathrm{x}|\mathrm{z} \sim \mathrm{p}_\Theta(\mathrm{x}|\mathrm{z})$$

    - Exact posterior distribution $\mathrm{p}_\Theta(\mathrm{z}|\mathrm{x})$ **is intractable** so **EM is intractable**

- **We wish to:**
  - **efficiently learn** approx. maximum likelihood parameters Θ
  - **efficiently infer** latent variables for new observations x

Deep Latent-Variable Models

- **directed latent variables model**
  - can represent complicated **marginal distributions** over x

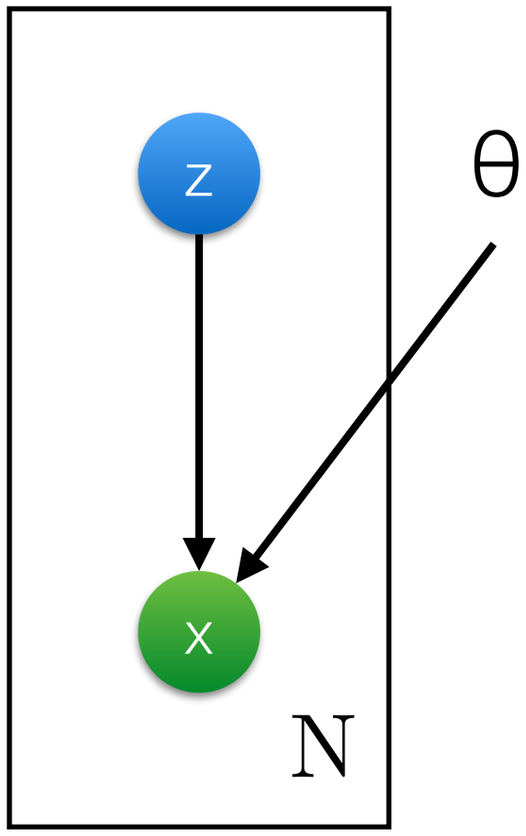


- **deep neural nets**
  - can represent complicated conditional dependencies

$$p(x,z_1,z_2) = p(x|z_2)p(z_2|z_1)p(z_1)$$

- We combine those strengths

Example model



- $\mathrm{p}(\mathrm{z}) = \mathrm{N}(0,\mathrm{I})$
- $\mathrm{p}_\theta(\mathrm{x}|\mathrm{z}) = \mathrm{N}(\mu,\sigma^2)$
  $\mu = \mathrm{f}_\theta(\mathrm{z}) =$ multilayer neural net
- With flexible neural net $f_\theta(z)$, $p_\theta(x)$ can be almost arbitrarily complicated / multi-modal distribution
- But intractable posterior distribution $\mathrm{p}(\mathrm{z}|\mathrm{x})$
  - Need approximate inference for learning

Non-variational approx. inference methods

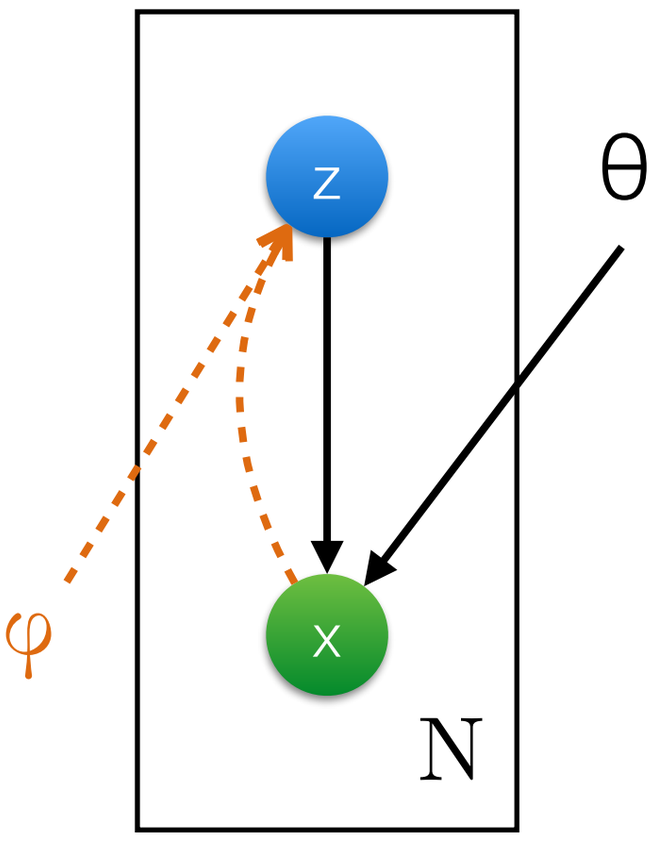
- Point estimate of $\mathrm{p}(\mathrm{z}|\mathrm{x})$ (MAP)
  - **Pro**: simple, fast
  - **Con**: too biased / approximate
- Markov Chain Monte Carlo (MCMC):
  - **Pro**: asymptotically unbiased
  - **Con:** often expensive, hard to assess convergence

Variational inference with Inference Networks

- Introduce **parametric model** $\mathrm{q}_\varphi(\mathrm{z}|\mathrm{x})$ of true posterior
  - φ: *variational parameters*
  - parameterised by neural networks
- Similar to Helmholtz Machine / Wake-Sleep (1994/1995, Dayan/Hinton/Frey/Neal, *Science*)
  - Also learns **parametric inference model** q(z|x) with SGD
  - But wake-sleep uses **different objective for** $\mathrm{q}_\varphi(\mathrm{z}|\mathrm{x})$ which doesn't optimise a bound log p(x)

Auto-Encoding Variational Bayes

[Kingma and Welling, 2013/2014]
[Rezende et al, 2014]



- $q_\varphi(z|x) = N(\mu,\sigma^2)$
  $[\mu, \sigma^2] = f^{(z|x)}(x,\varphi) = \text{multilayer neural net}$
- Objective: lower bound of log p(x).
  - Jointly optimized w.r.t. φ and θ
  - This is approx. maximum likelihood
  - Simple SGD:
    - Sampling small minibatches of data
    - Sampling from approx. posterior
- This also minimizes an expected KL divergence
  $D_{KL}(q_\varphi(z|x)||p(z|x))$
  -> gives us cheap approx. inference for new datapoints

Variational bound

$$\log p(x) = \mathcal{L}(x) + D_{KL}(q_\phi(z|x)||p(z|x))$$

$$\downarrow$$

Objective per datapoint:

$$\mathcal{L}(x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p(x,z) - \log q_\phi(z|x)\right]$$

Abbreviated:

$$\mathcal{L}(x) = \mathbb{E}_{q_\phi(z|x)}\left[f_\phi(x,z)\right]$$

Relatively new idea:

Stochastic Gradient-based Variational Inference

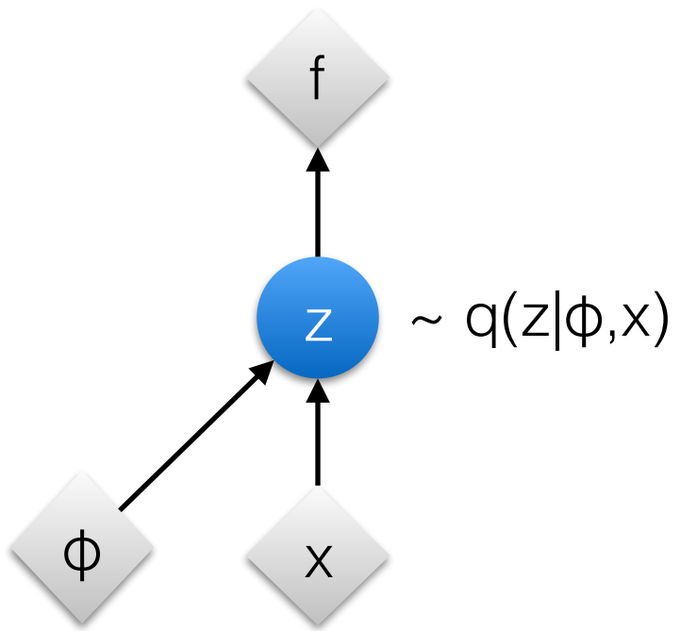
Objective per datapoint:

$$\mathcal{L}(x) = \mathbb{E}_{q_\phi(z|x)}\left[f_\phi(x, z)\right]$$

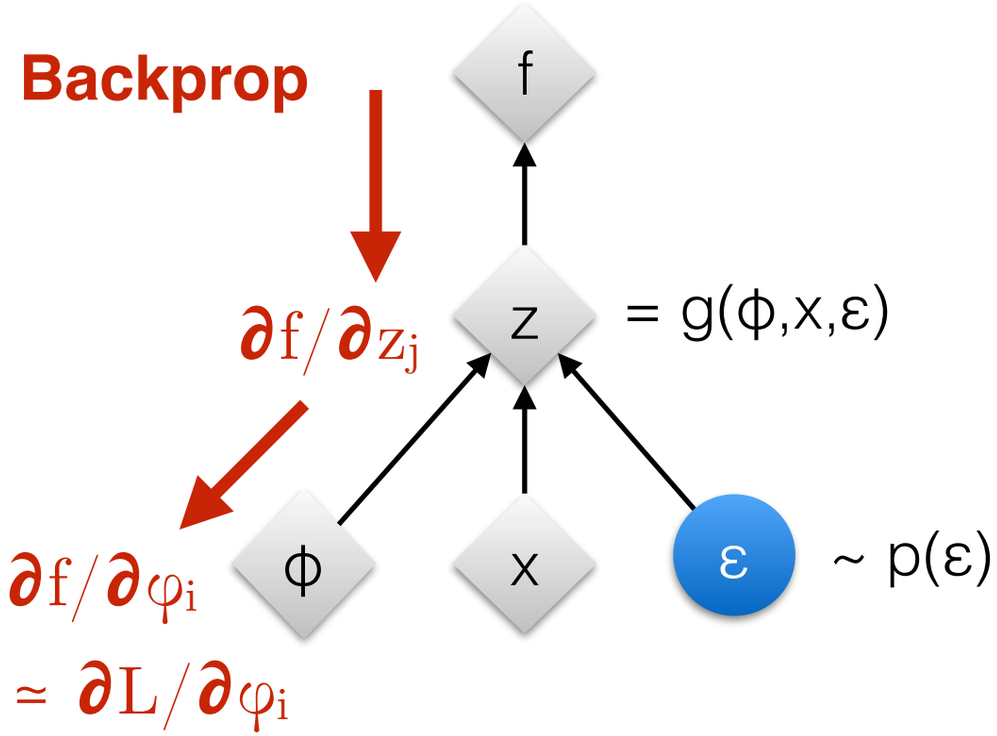
- Often no analytical solution to exact gradient $\nabla_\varphi \mathrm{L}$
- Solution: **(doubly) stochastic gradient ascent**
  - Only requires *unbiased estimates* of gradient
  - Can use *small minibatches of data*

Reparameterization trick

Original form



Reparameterised form



: Deterministic node

: Random node

[Kingma, 2013]
[Bengio, 2013]
[Kingma and Welling 2014]
[Rezende et al 2014]

Reparameterization trick

- Can be performed for a broad class of distributions, e.g.:

  - **Location-scale transforms**
    Normal, Laplace, Student t's, Logistic, etc.

  - **Inverse of CDF**
    Cauchy, Rayleight, Pareto, etc

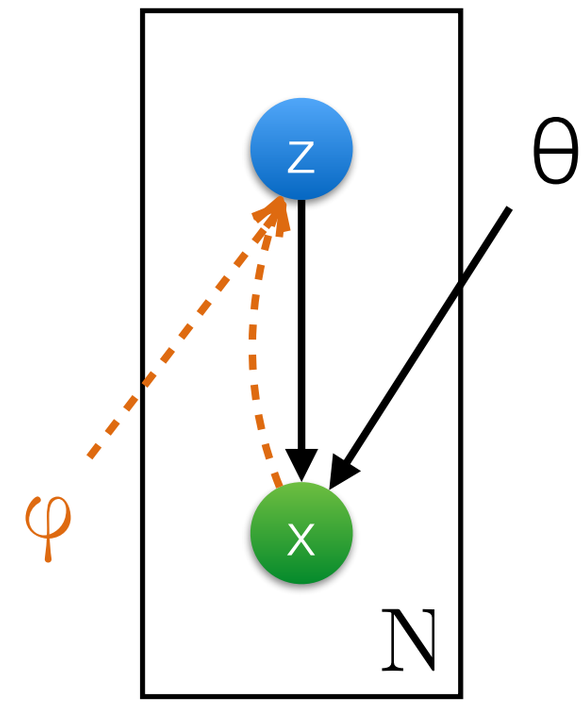
  - Other strategies exist
    Gamma, Dirichlet, Beta, Chi-Squared, etc

Stochastic Gradient Variational Inference

- **Variational inference by gradient ascent**
- **“Swiss army knife” for inference:**
  - Works with almost any $\mathrm{p}(\mathrm{x},\mathrm{z})$
  - Works with almost any $\mathrm{q}(\mathrm{z}|\mathrm{x})$
  - Just requires gradient ascent on single objective

Connection to auto-encoders

- Variational Auto-Encoder (**VAE**):
  - $\mathrm{q}(\mathrm{z}|\mathrm{x})$: stochastic neural encoder
    $\mathrm{p}(\mathrm{x}|\mathrm{z})$: stochastic neural decoder



Objective function

$$L = \underbrace{(\log p(x|z)}_{\text{Reconstruction error}} + \underbrace{\log p(z) - \log q(z|x))|_{z=g(\epsilon)}}_{\text{Regularization terms dictated by the bound}}$$

Conv. net as encoder/decoder, trained on faces

(trained by Alec Radford 2015)

Semi-Supervised Learning with Deep Generative Models

NIPS 2014

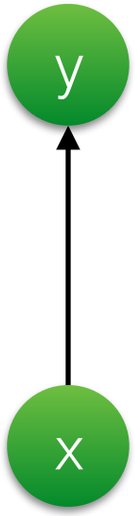
Diederik P. Kingma
Danilo J. Rezende
Shakir Mohamed
Max Welling

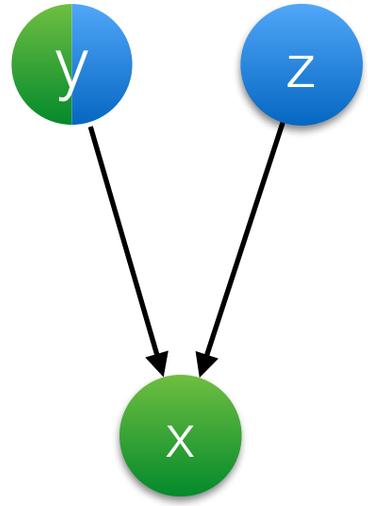
UNIVERSITEIT VAN AMSTERDAM

Google DeepMind

Classifier vs generative model

Classifier vs Generative model

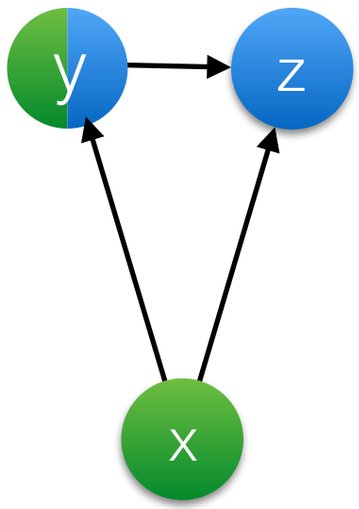




Each edge is parameterised as a deep neural net

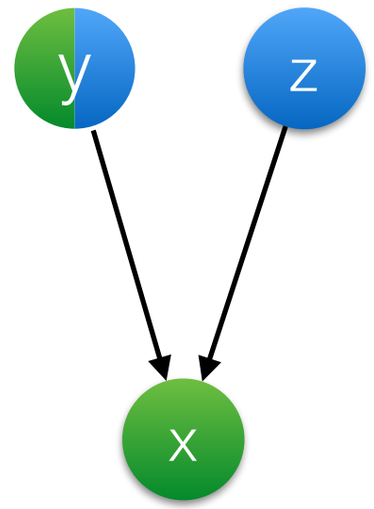
Generative model
for semi-supervised learning

Inference model



$$\mathrm{q(y|x)}$$
$$= \mathrm{classifier}$$

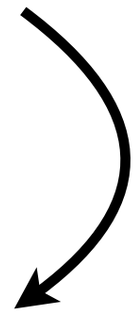
Generative model



Variational training includes optimisation of q(y|x)

VAEs are SOTA
on semi-supervised learning on MNIST

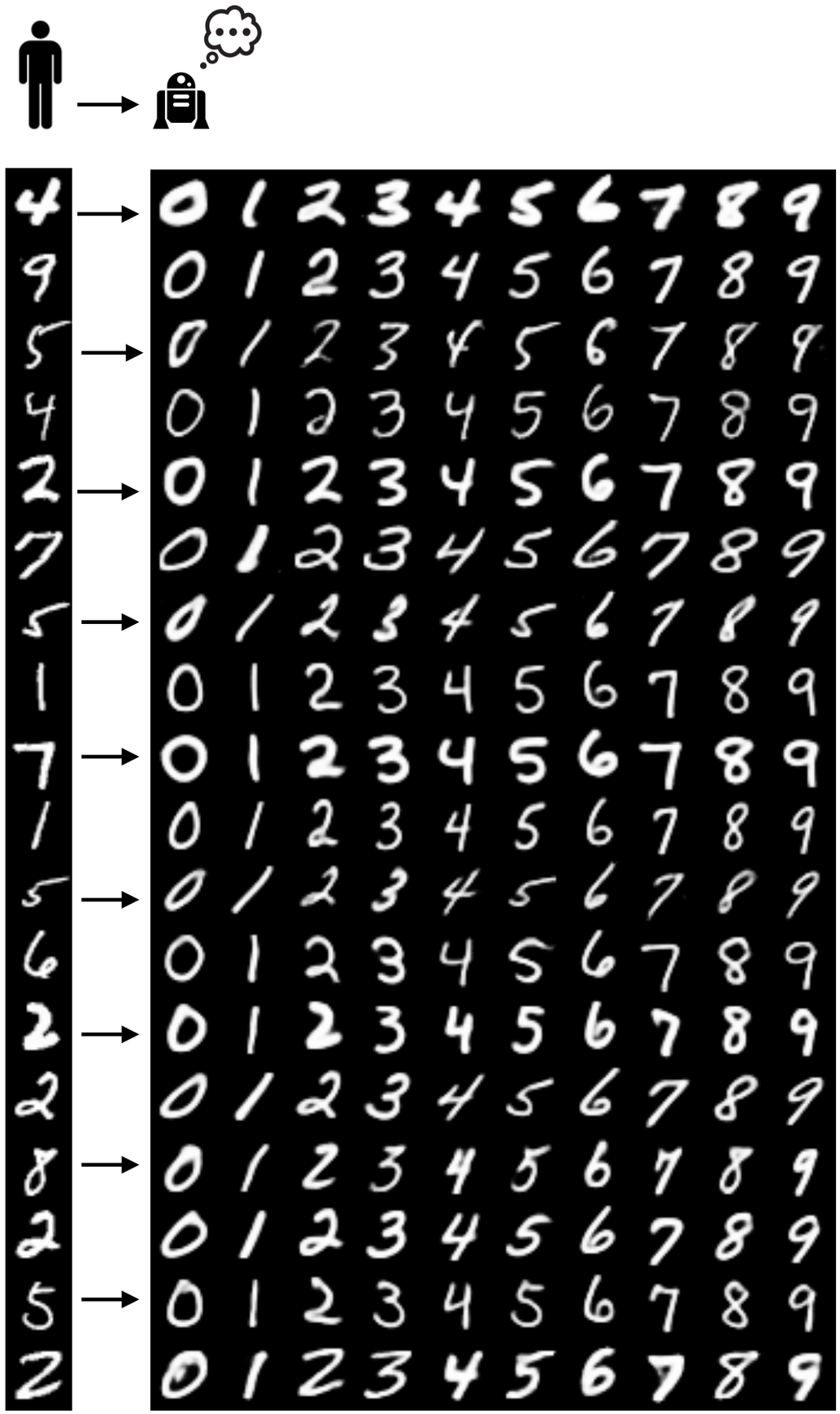
	100 labels
AtlasRBF (Pitelis et al., 2014)	8.10% (±0.95)
Deep Generative Model (M1+M2) (Kingma et al., 2014)	3.33% (±0.14)
Virtual Adversarial (Miyato et al., 2015)	2.12%
Ladder (Rasmus et al., 2015)	1.06% (±0.37)
Auxiliary Deep Generative Model (1 MC)	2.25% (± 0.08)
Auxiliary Deep Generative Model (10 MC)	**0.96% (± 0.02)**

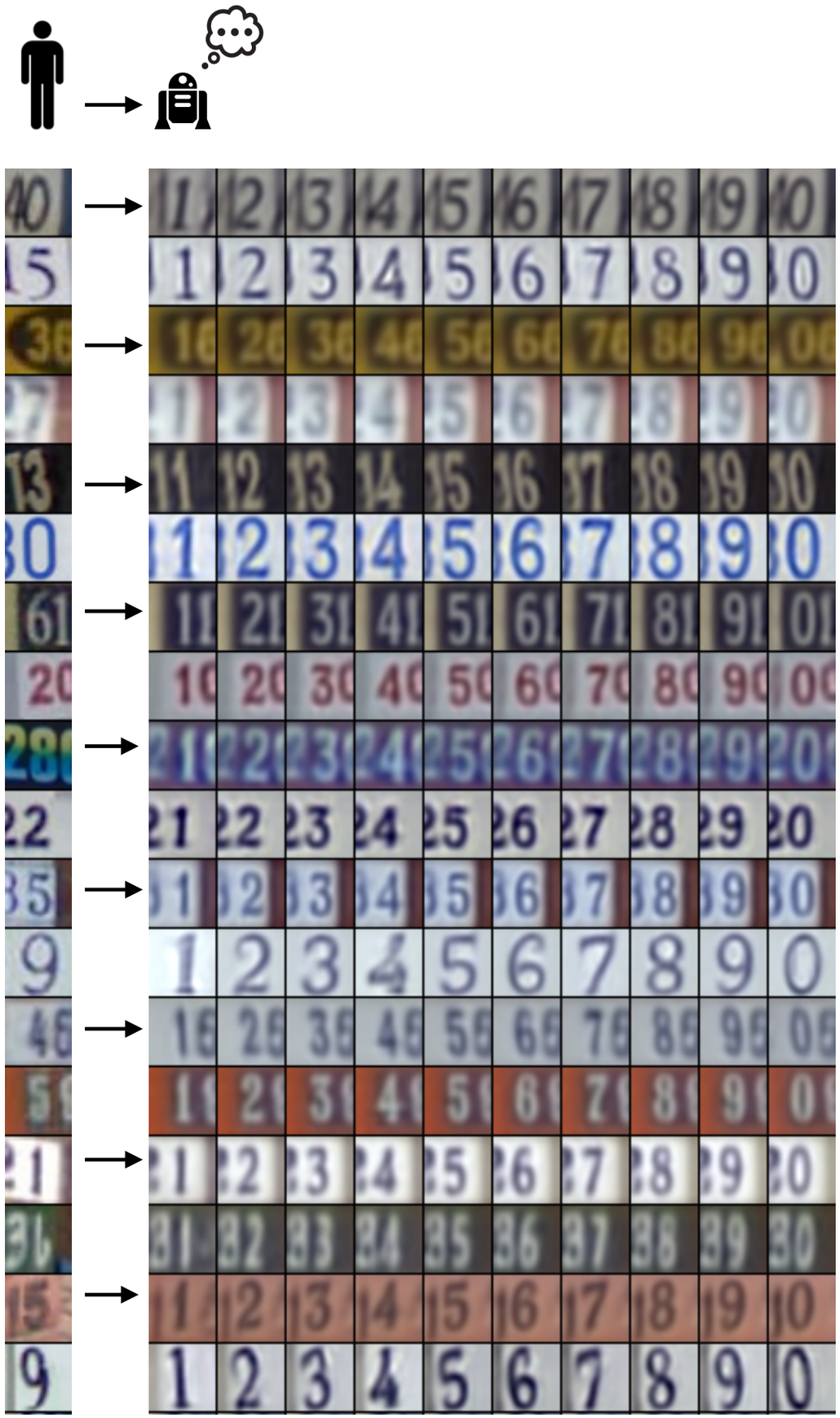


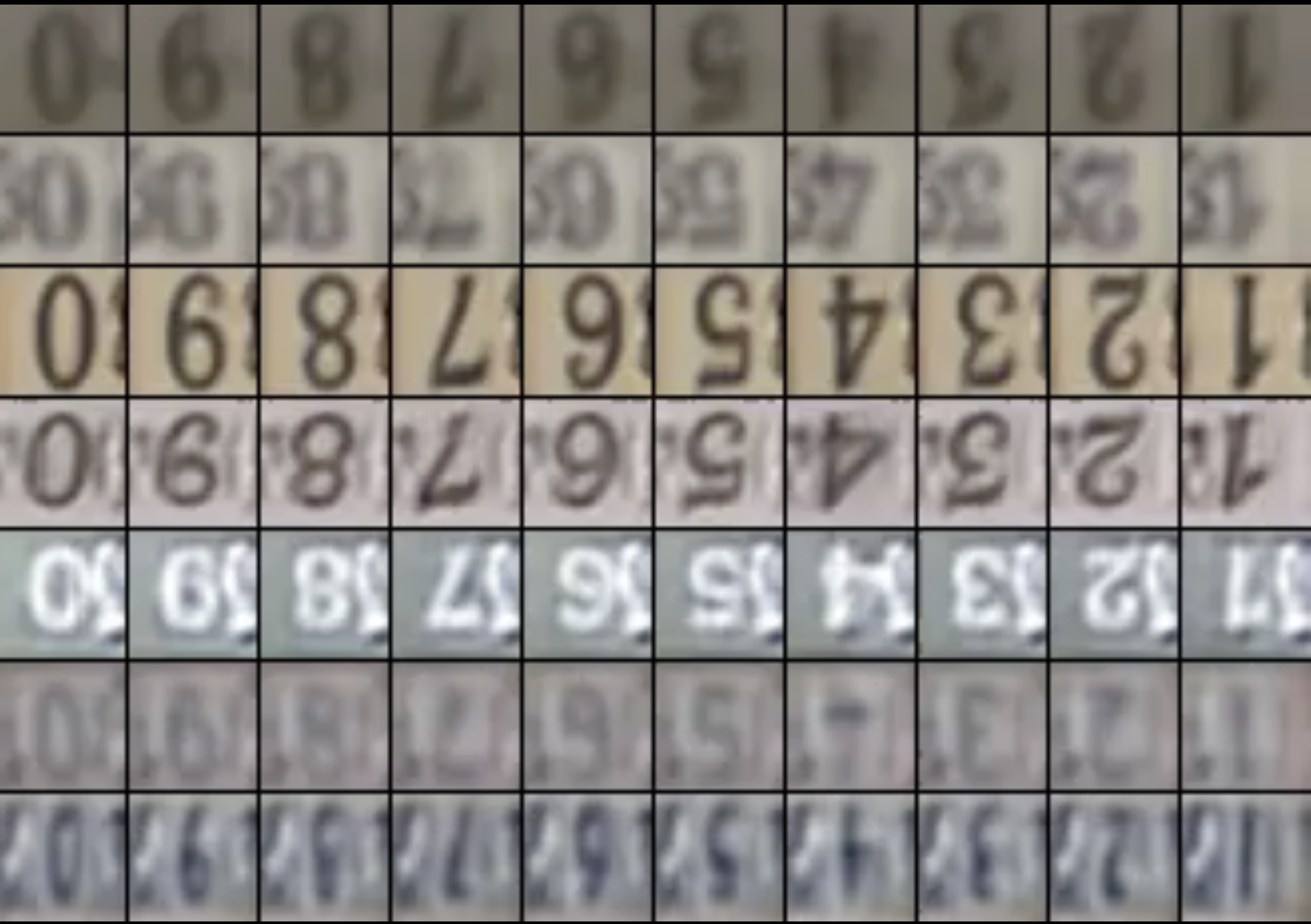
"Improving Semi-Supervised Learning with Auxiliary Deep Generative Models"

[Maaløe, Sønderby, Sønderby and Winter, 2015]

Analogy-making



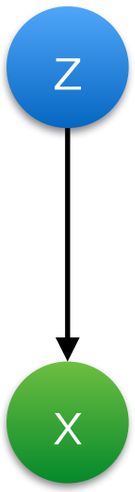




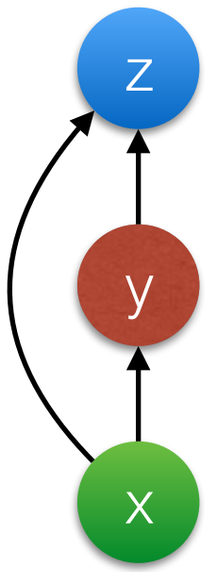
MCMC and Variational Inference: Bridging the Gap

Salimans, Kingma and Welling, ICML 2015

Generative model
$\mathrm{p}(\mathrm{x},\mathrm{z})$



Inference model
$\mathrm{q}(\mathrm{y},\mathrm{z}|\mathrm{x})$



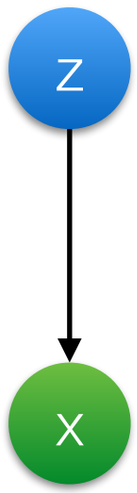
Implicit inference model
$\mathrm{q}(\mathrm{z}|\mathrm{x}) = \int \mathrm{q}(\mathrm{y},\mathrm{z}|\mathrm{x})\ \mathrm{dy}$

- Pro: **more accurate/flexible inference model** $\mathrm{q}(\mathrm{z}|\mathrm{x})$
- But: **intractable PDF** $\mathrm{q}(\mathrm{z}|\mathrm{x}) = \int \mathrm{q}(\mathrm{y},\mathrm{z}|\mathrm{x})\ \mathrm{dy}$

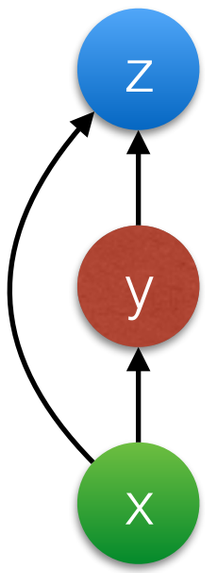
MCMC and Variational Inference: Bridging the Gap

Salimans, Kingma and Welling, ICML 2015

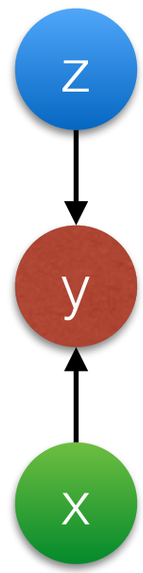
Generative model
$\mathrm{p}(\mathrm{x},\mathrm{z})$



Inference model
$\mathrm{q}(\mathrm{y},\mathrm{z}|\mathrm{x})$



auxiliary model
$\mathrm{r}(\mathrm{y}|\mathrm{x},\mathrm{z})$



new!

$$\mathcal{L}_{\text{aux}} = \mathbb{E}_{q_\phi(y,z|x)}\left[\log p(x,z) - \log q_\phi(y,z|x) + \log r_\phi(y|z,x)\right]$$

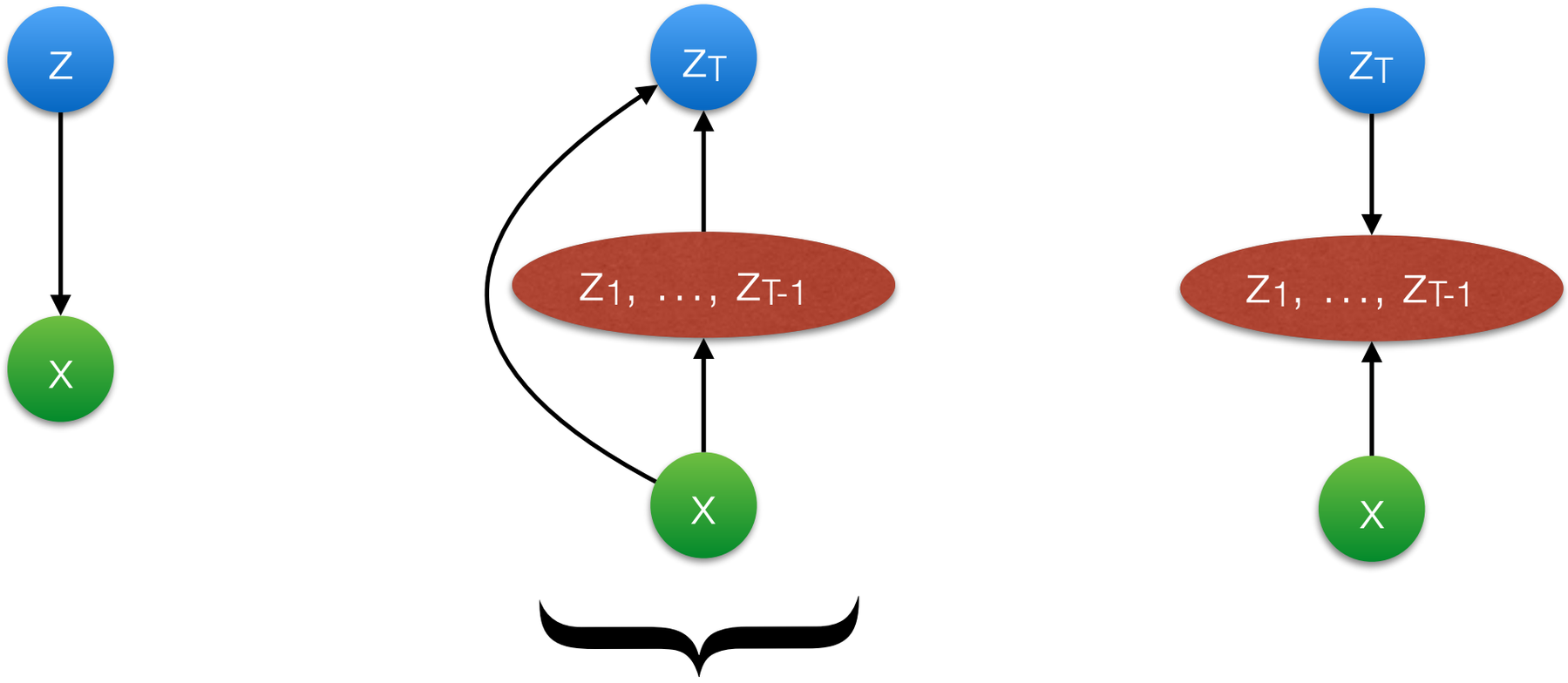
MCMC and Variational Inference: Bridging the Gap

Salimans, Kingma and Welling, ICML 2015

Generative model
$p(x,z)$

Inference model
$q(z_1,..,z_T|x)$

Auxiliary model
$r(z_1,...,z_{T-1}|z_T,x)$



MCMC chain, e.g. Hamiltonian Monte Carlo, as inference model with auxiliary variables

MCMC and Variational Inference: Bridging the Gap

Salimans, Kingma and Welling, ICML 2015

Table 1. Comparison of our approach to other recent methods in the literature. We compare the average marginal log-likelihood measured in nats of the digits in the MNIST test set. See section 3.2 for details.

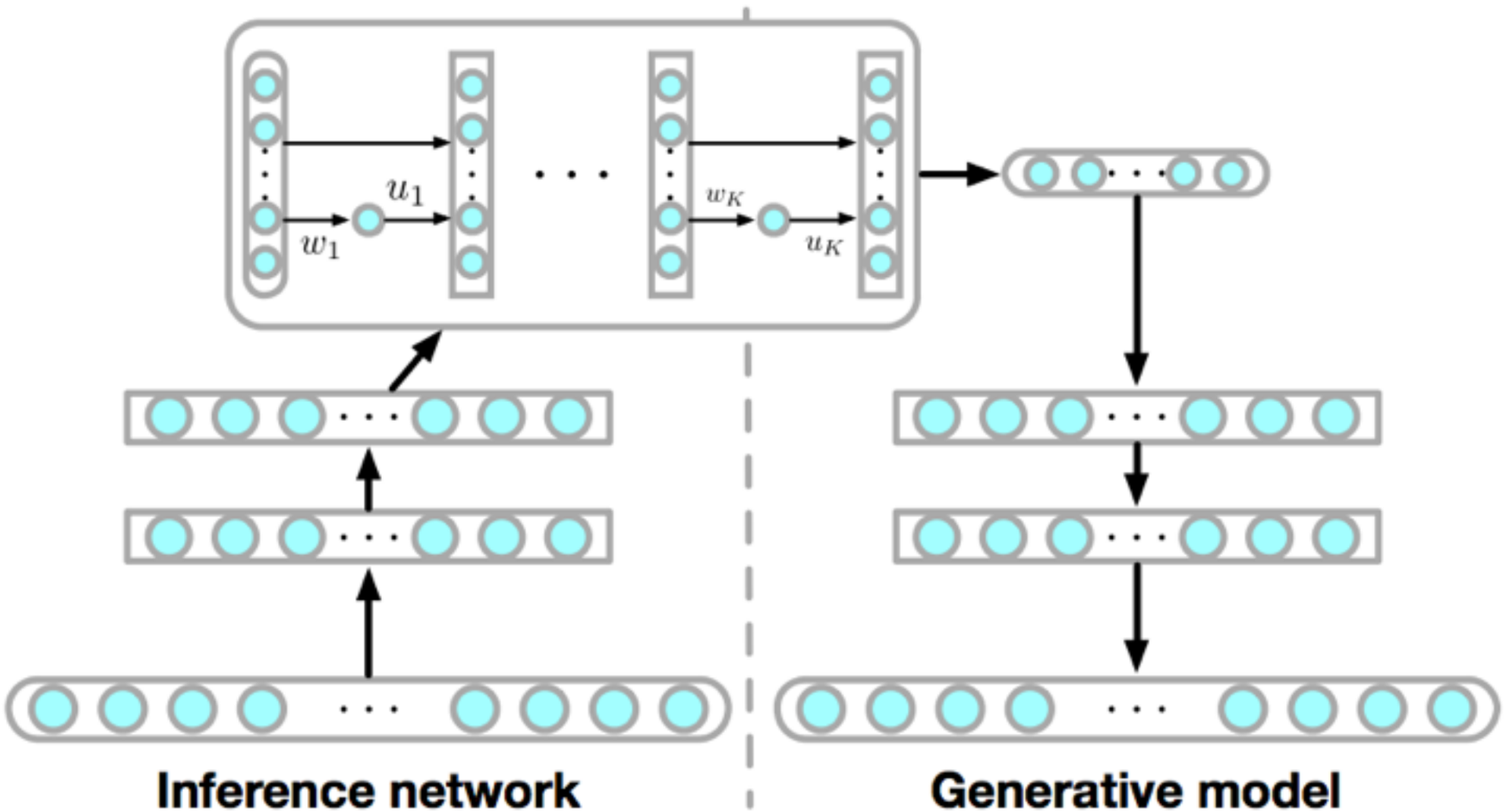
Model	$\log p(x) \geq -$	$\log p(x) = -$
HVI + fully-connected VAE:		
Without inference network:		
5 leapfrog steps	90.86	87.16
10 leapfrog steps	87.60	85.56
With inference network:		
No leapfrog steps	94.18	88.95
1 leapfrog step	91.70	88.08
4 leapfrog steps	89.82	86.40
8 leapfrog steps	88.30	85.51
HVI + convolutional VAE:		
No leapfrog steps	86.66	83.20
1 leapfrog step	85.40	82.98
2 leapfrog steps	85.17	82.96
4 leapfrog steps	84.94	82.78
8 leapfrog steps	84.81	82.72
16 leapfrog steps	84.11	82.22
16 leapfrog steps, $n_h = 800$	83.49	81.94
From (Gregor et al., 2015):		
DBN 2hl		84.55
EoNADE		85.10
DARN 1hl	88.30	84.13
DARN 12hl	87.72	
DRAW	80.97	

Large improvement in nats

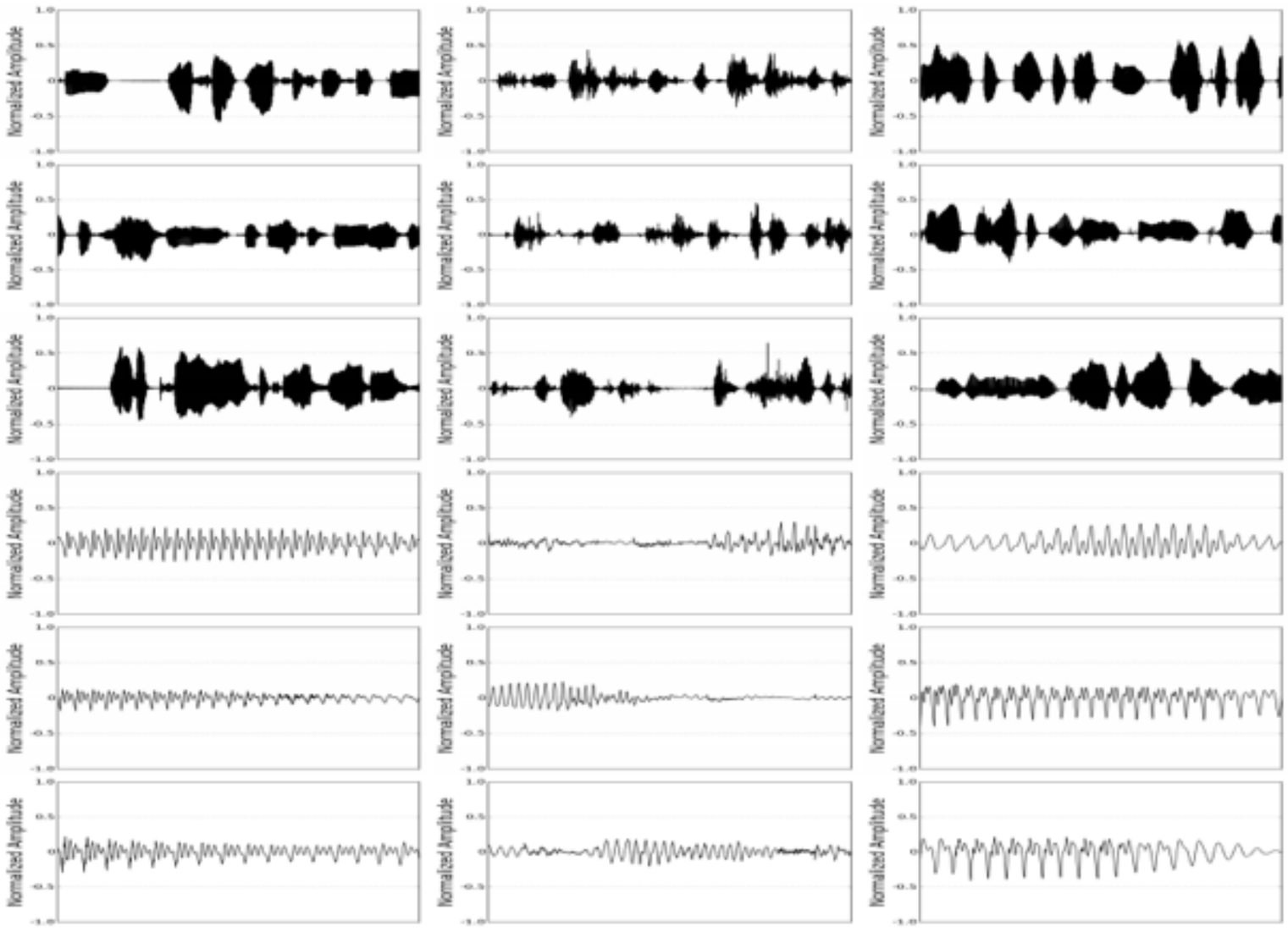
DRAW: Gregor et al, 2014
Recurrent VAE with attention

Variational Inference with Normalizing Flows

[Rezende and Mohamed, ICML 2015]



A Recurrent Latent Variable Model for Sequential Data
[Chung et al, 2015]



(a) Ground Truth (b) RNN-GMM (c) VRNN-Gauss

Generating speech with a variational RNN

Generating Images from Captions with Attention

[Mansimov et al, 2015]

(under submission at ICLR)

A stop sign is flying in blue skies.

A herd of elephants flying in the blue skies.

A toilet seat sits open in the grass field.

A person skiing on sand clad vast desert.

Importance Weighted Autoencoders
[Burda et al 2015]
(under submission at ICLR)

- Objective:

$$\mathcal{L}_k(\mathbf{x}) = \mathbb{E}_{\mathbf{h}_1,\ldots,\mathbf{h}_k \sim q(\mathbf{h}|\mathbf{x})} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p(\mathbf{x}, \mathbf{h}_i)}{q(\mathbf{h}_i|\mathbf{x})} \right]$$

- Equivalent to VAE objective for $\mathrm{k}=1$
- For $\mathrm{k}>1$, optimizes a tighter bound, at the expense of extra computation

Summary

- Variational Auto-Encoding: **scalable generative modeling**
  - Works with almost any model $\mathrm{p}(\mathrm{x},\mathrm{z})$
  - Works with almost any approx. posterior $\mathrm{q}(\mathrm{z}|\mathrm{x})$
  - Scales to huge datasets
  - Just requires gradient ascent on single objective
- Applications:
  - deep synthesis / analogic reasoning
  - nonlinear PCA
  - semi-supervised learning
  - optimizing MCMC hyper-parameters

https://github.com/dpkingma