Stochastic Backpropagation, Variational Inference, and Semi-Supervised Learning


Diederik (Durk) Kingma


Danilo J. Rezende


Shakir Mohamed


Max Welling

(*) Universiteit van Amsterdam

(**) Google DeepMind

Stochastic Gradient Variational Inference

Bayesian inference problem setup

- x: *observed data*

  z: *unobserved/latent variables* or parameters

  $\mathrm{p(x,z)}$: probabilistic model, often factorized

- We are (very) interested in inferring a *posterior distribution* $\mathrm{p(z|x)}$

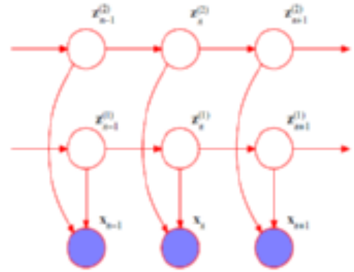
  e.g.:

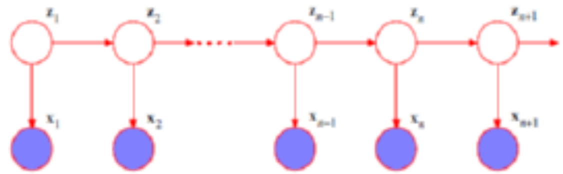
  - Full distribution over model parameters

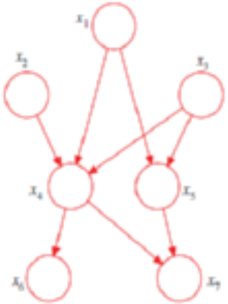
  - Enables learning parameters in latent-variable models

- $\mathrm{p(z|x) = p(x,z)/p(x)}$ most often *intractable*

  - Need good approximations







Non-variational approx. inference methods

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

Variational inference

- Introduce *parametric model* $q_\varphi(z)$ or $q_\varphi(z|x)$ of true posterior
  - φ: *variational parameters*
- Objective: optimize φ w.r.t. the KL-divergence: $D_{KL}(q_\varphi(z|x)||p(z|x))$
  - $q_\varphi(z|x) = p(z|x)$ when $D_{KL}(q_\varphi(z|x)||p(z|x)) = 0$

Variational bound

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

$$\downarrow$$

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

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

- Non-gradient-based optimisation technique:
  Mean-Field VB with fixed-point equations

  Pro: efficiency
  Con: intractable / not applicable in many cases

Relatively new idea:

Stochastic Gradient-based Variational Inference

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

- Often no analytical solution to exact gradient $\nabla_\varphi \mathrm{L}$
- Solution: **stochastic gradient ascent**
  Only requires *unbiased estimates* of gradient

Strategy 1: **Standard gradient estimator**

Objective:

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

Gradient:

$$\begin{aligned}\nabla_\phi \mathcal{L} &= \mathbb{E}_{q_\phi(z|x)}\left[(\nabla_\phi \log q_\phi(z|x)) f_\phi(x,z)\right] \\ &\simeq (\nabla_\phi \log q_\phi(z^l|x)) f_\phi(x,z^l) \\ &\text{where } z^l \sim q_\phi(z|x)\end{aligned}$$

Pro: Valid for almost any $\mathrm{q(z|x)}$.

Con: Variance. Often requires variance reduction techniques.

[**Hoffman et al, 2013**] Stochastic Variational Inference
[**Blei et al, 2013**] Variational bayesian inference with stochastic search.
[**Ranganath et al, 2014**] Black Box Variational Inference.
[**Mnih and Gregor, 2014**] Neural Variational Inference and Learning in Belief Networks.

Objective:

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

Gradient:

$$\begin{aligned}
&\text{Step 1: sample } z^l \text{ from } q_\phi(z|x)\\
&\text{Step 2: } \mathcal{L} \simeq f_\phi(x, z^l)\\
&\text{Step 3: } \nabla_\phi \mathcal{L} \simeq \nabla_\phi f_\phi(x, z^l)
\end{aligned}$$

- Problem: requires backpropagation through sampling process

Strategy 2: **Reparameterized gradient estimator**

Objective:

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

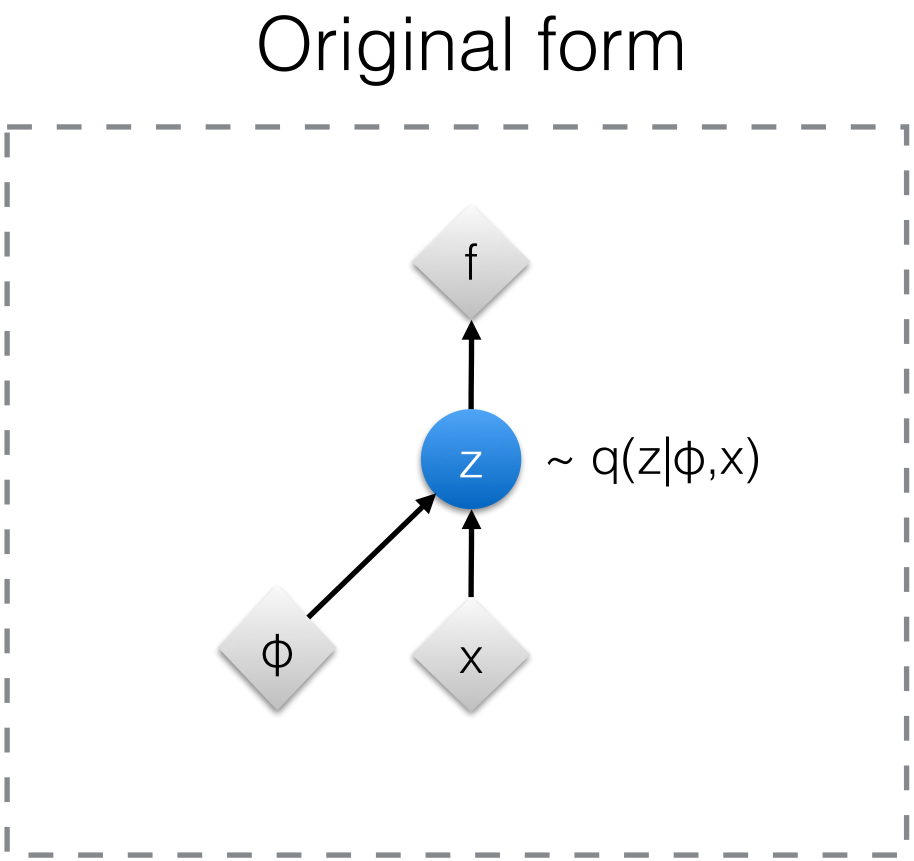
Gradient:

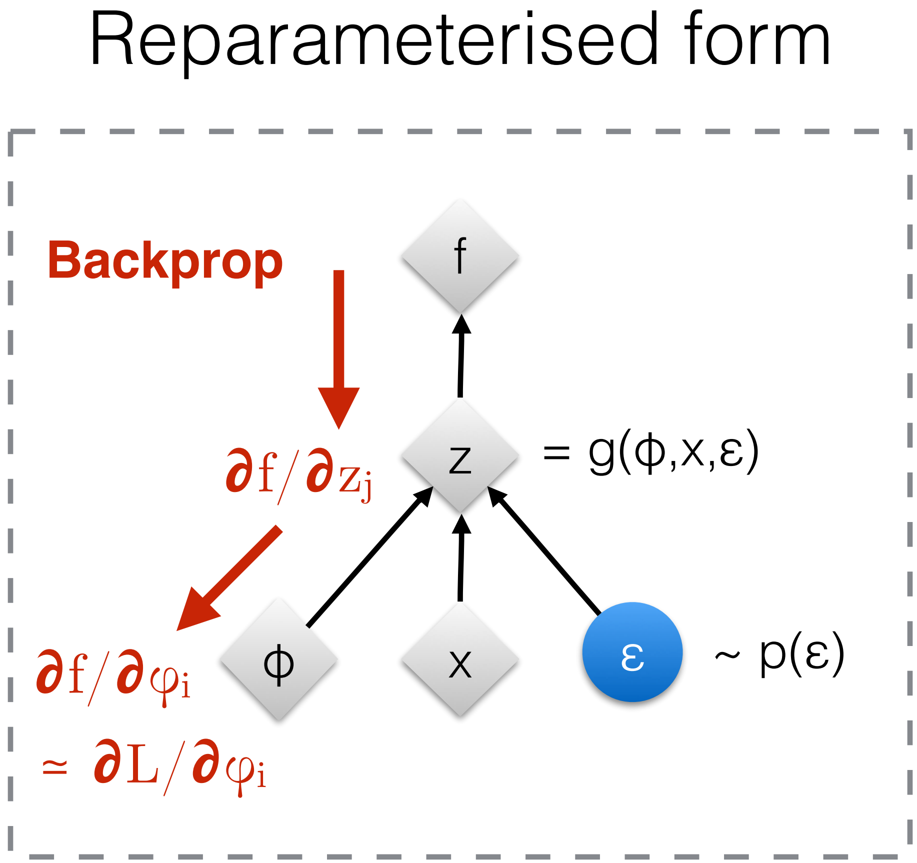
$$\begin{aligned}
&\text{Step 1: sample } \epsilon^l \text{ from } p(\epsilon)\\
&\text{Step 2: } z^l = g_\phi(\epsilon), \quad \text{such that } z^l \sim q_\phi(z|x)\\
&\text{Step 3: } L \simeq f_\phi(x, z^l)\\
&\text{Step 4: } \nabla_\phi \mathcal{L} \simeq \nabla_\phi f_\phi(x, z^l)
\end{aligned}$$

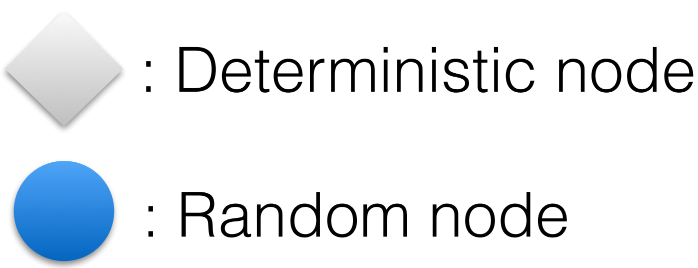
- Simple, low variance.
- Can be combined with standard gradient for discrete vars.

[Kingma and Welling, 2013/2014] Auto-encoding Variational Bayes
[Rezende et al, 2014] Stochastic Backpropagation and Variational Inference in DLVMs
[Titsias and Lázaro-Gredilla, 2014] Doubly Stochastic VB for non-Conjugate Inference

Reparameterization trick







[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 $p(x,z)$
  - Works with almost any $q(z|x)$
  - Just requires gradient ascent on single objective

Deep latent-variable models

Deep Latent-Variable Models

- We combining the strengths of **deep neural nets** with those of **latent-variable models**
  - **directed latent variables models**: can represent complicated **marginal distributions** over x
  - probabilistic **deep neural nets**: can represent complicated conditional dependencies $p(y|x) = f(x,y)$
- Intractable posterior distribution $p(z|x)$

  $=>$ Approximate inference

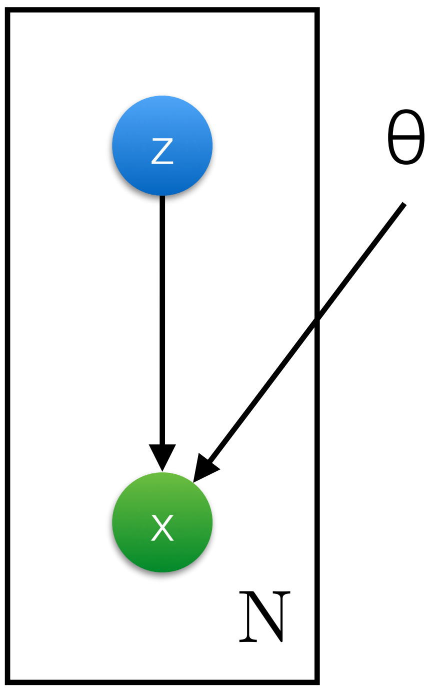
$p(z_1)$

$p(z_2|z_1)$

$p(x|z_2)$

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

Example model



- z: latent variable (low-dimensional)
  x: observed variable
  θ: parameters

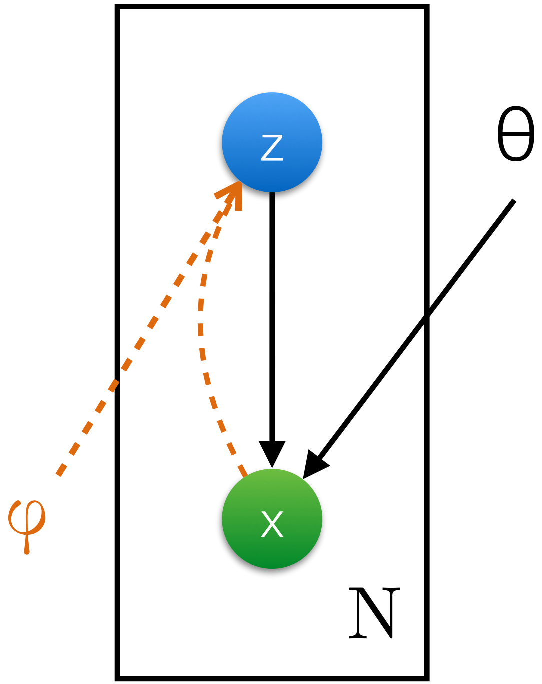
- $p(z) = N(0,I)$

- $p_\theta(x|z) = N(\mu,\sigma^2)$
  $[\mu,\ \sigma^2] = f^{(x|z)}(z;\theta) =$ multilayer neural net

x: e.g. Face

Auto-Encoding Variational Bayes

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



- φ: variational parameters
  $q_\varphi(z|x) = N(\mu, \sigma^2)$
  $[\mu, \sigma^2] = f^{(z|x)}(x, \varphi) =$ multilayer neural net

- Objective: lower bound of log p(x).

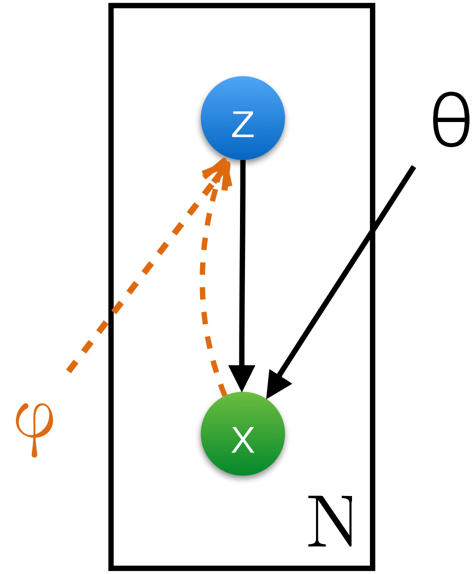
$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^i)} \left[\log p_\theta(x^i, z) - \log q_\phi(z|x^i)\right]$$

- Jointly optimized w.r.t. φ and θ

- Doubly stochastic optimization:
  - Using small minibatches of data
  - Using our proposed gradient estimator

Connection to auto-encoders

- "Variational Auto-Encoder":
  - $q(z|x)$: stochastic encoder
    $p(x|z)$: stochastic decoder
  - Variational bound decomposes as negative reconstruction error plus regularisation terms



Objective function

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

Reconstruction error

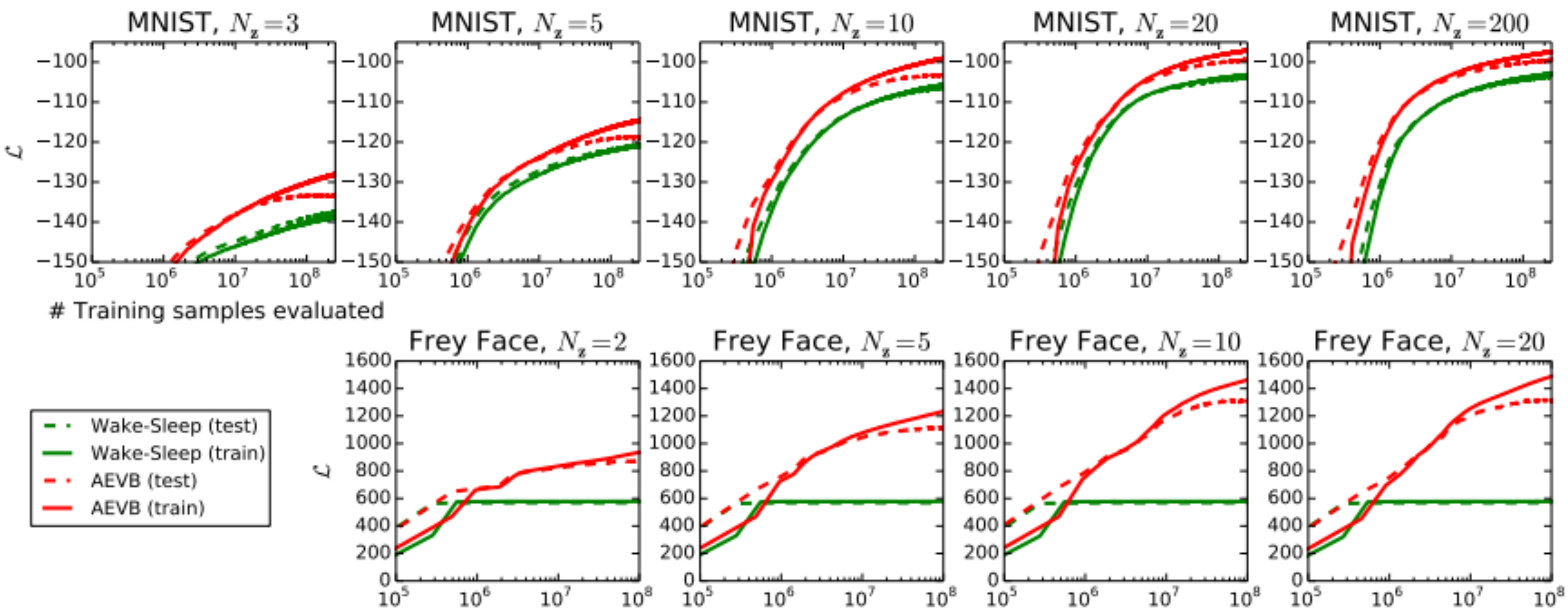
Regularization terms dictated by the bound

Connection to Helmholtz Machine / Wake-Sleep

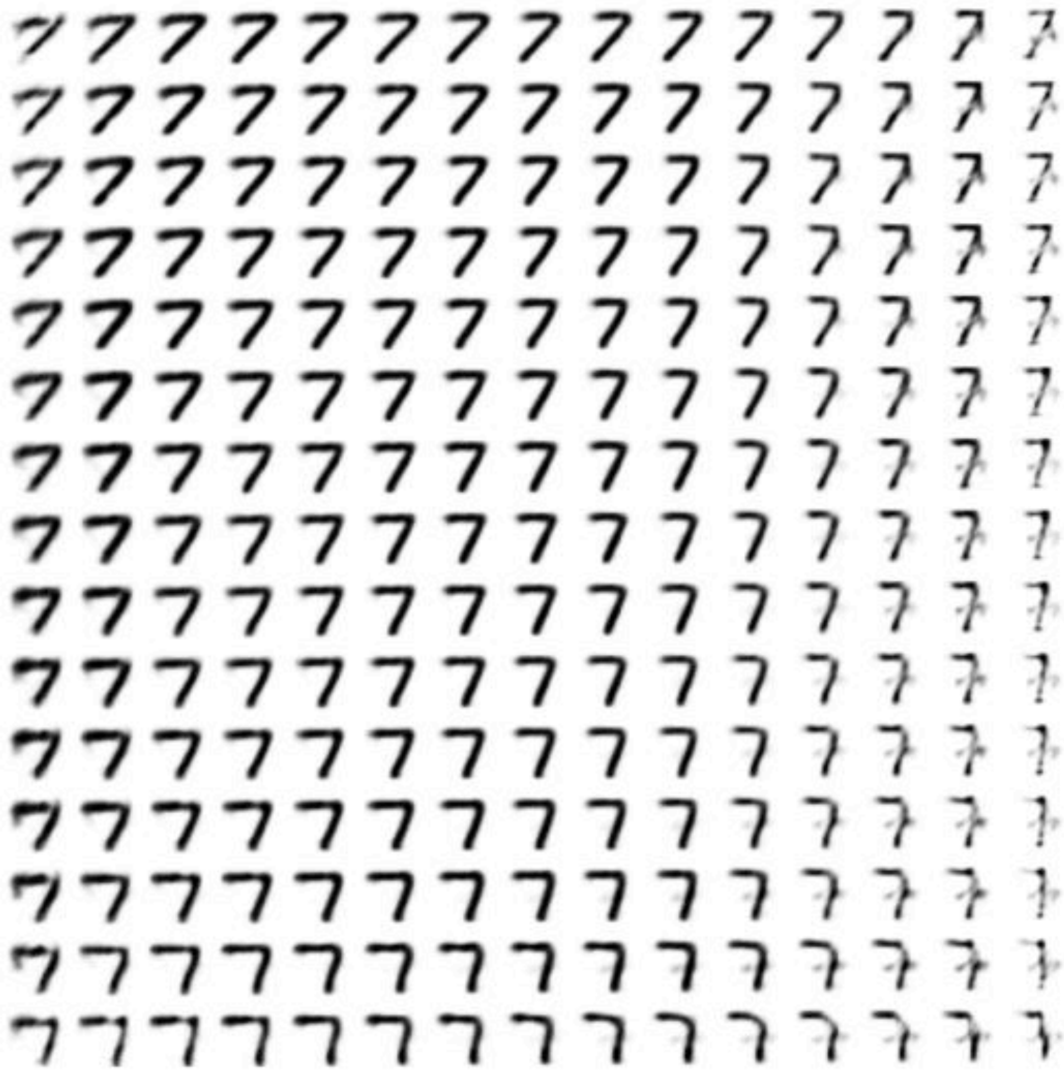
(1994/1995, Dayan/Hinton/Frey/Neal, *Science*)

- Also introduced **parametric inference model** $q(z|x)$ learned with **gradient ascent**
- But the wake-sleep algorithm used **incorrect gradient for *q***
- **Main difference is: now we know how to learn Q correctly with gradient ascent**

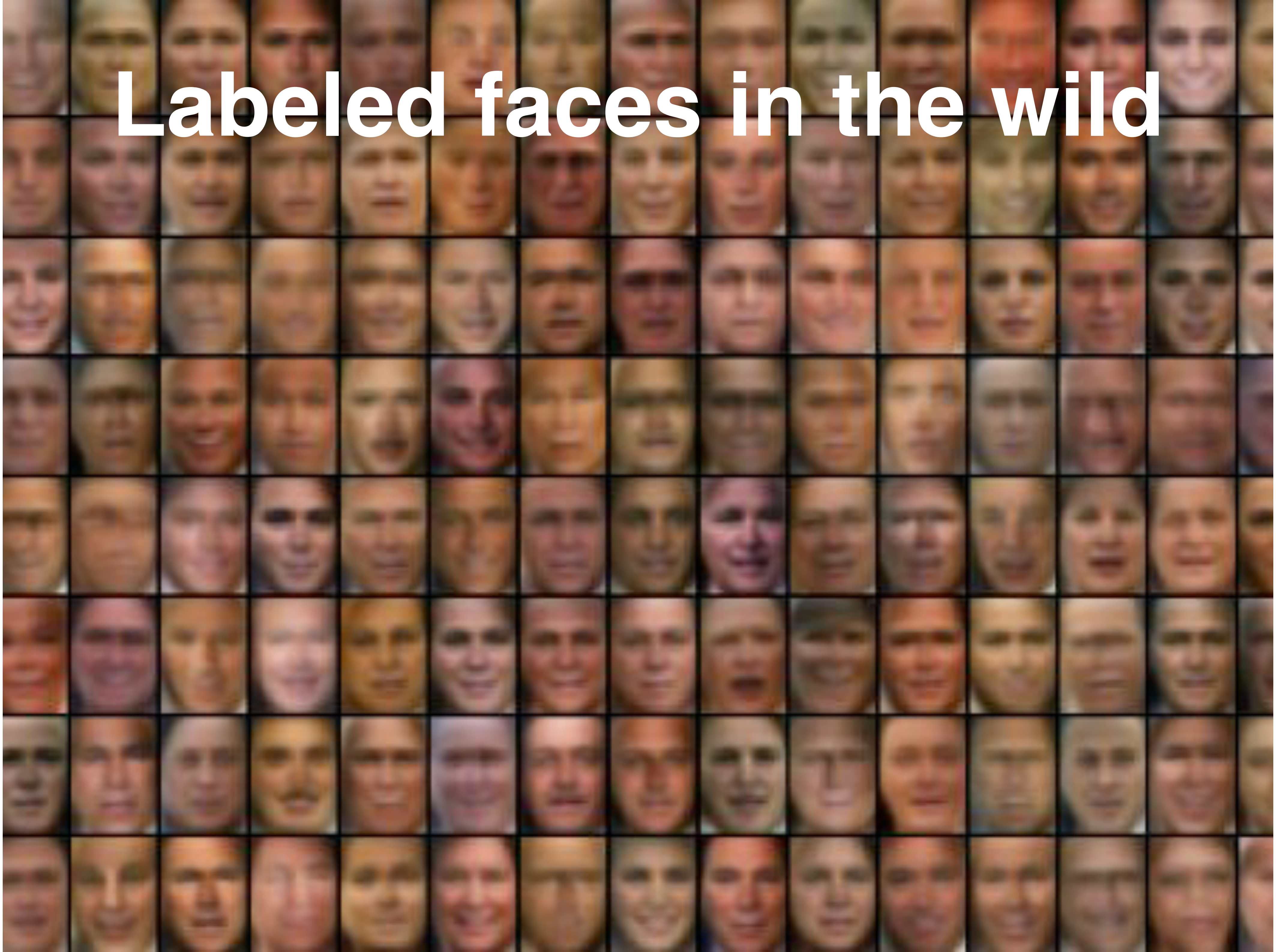
AEVB vs Wake-Sleep



3D latent space



Labeled faces in the wild


Labeled faces in the wild

Semi-Supervised Learning with Deep Generative Models

Diederik P. Kingma (*)
Danilo J. Rezende (**)
Shakir Mohamed (**)
Max Welling (*)

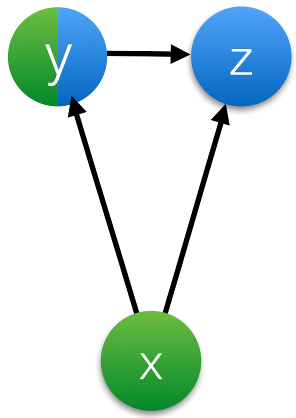
Classifier vs generative model

Classifier vs Generative model

Each edge is parameterised as a deep neural net

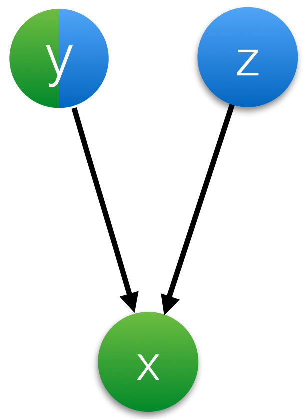
Generative model for semi-supervised learning

Inference model



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

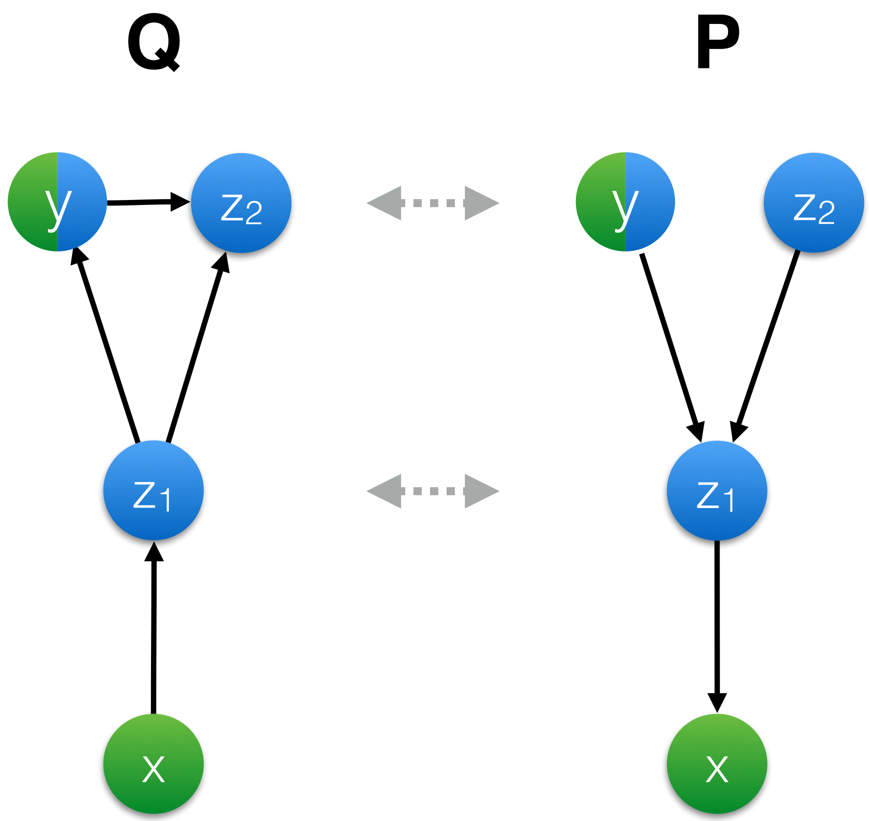
Generative model



Each edge is parameterised as a deep neural net

Deeper Approach

Stacked semi-supervised learner

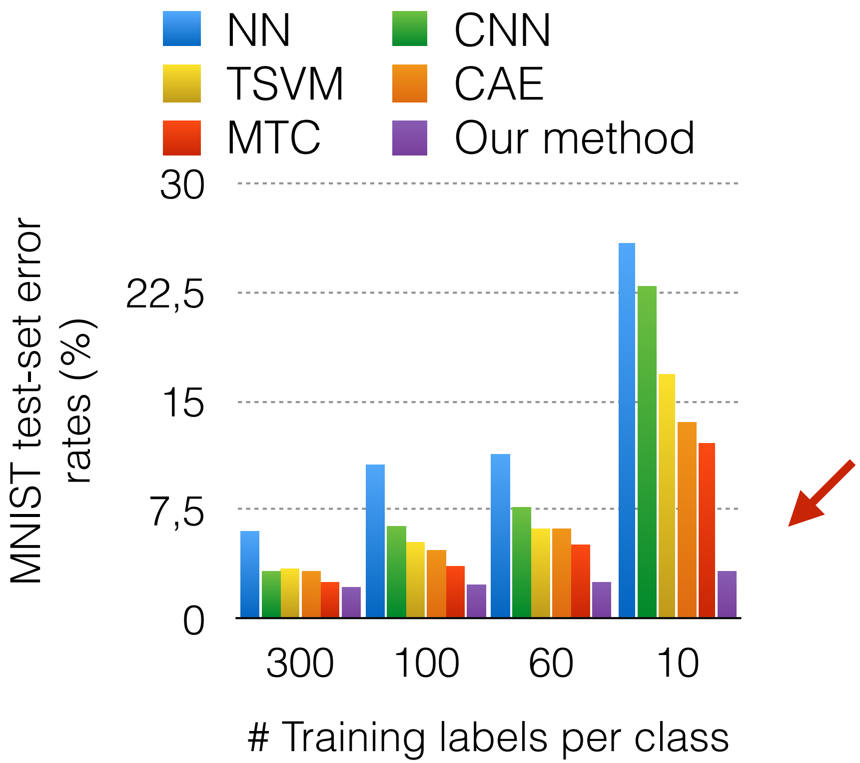


Each edge is parameterised as a deep neural net

Classification Results

MNIST

N	NN	CNN	TSVM	CAE	MTC	AtlasRBF	M1+TSVM	M2	M1+M2
100	25.81	22.98	16.81	13.47	12.03	8.10 ($\pm$ 0.95)	11.82 ($\pm$ 0.25)	11.97 ($\pm$ 1.71)	**3.33** ($\pm$ 0.14)
600	11.44	7.68	6.16	6.3	5.13	–	5.72 ($\pm$ 0.049)	4.94 ($\pm$ 0.13)	**2.59** ($\pm$ 0.05)
1000	10.7	6.45	5.38	4.77	3.64	3.68 ($\pm$ 0.12)	4.24 ($\pm$ 0.07)	3.60 ($\pm$ 0.56)	**2.40** ($\pm$ 0.02)
3000	6.04	3.35	3.45	3.22	2.57	–	3.49 ($\pm$ 0.04)	3.92 ($\pm$ 0.63)	**2.18** ($\pm$ 0.04)



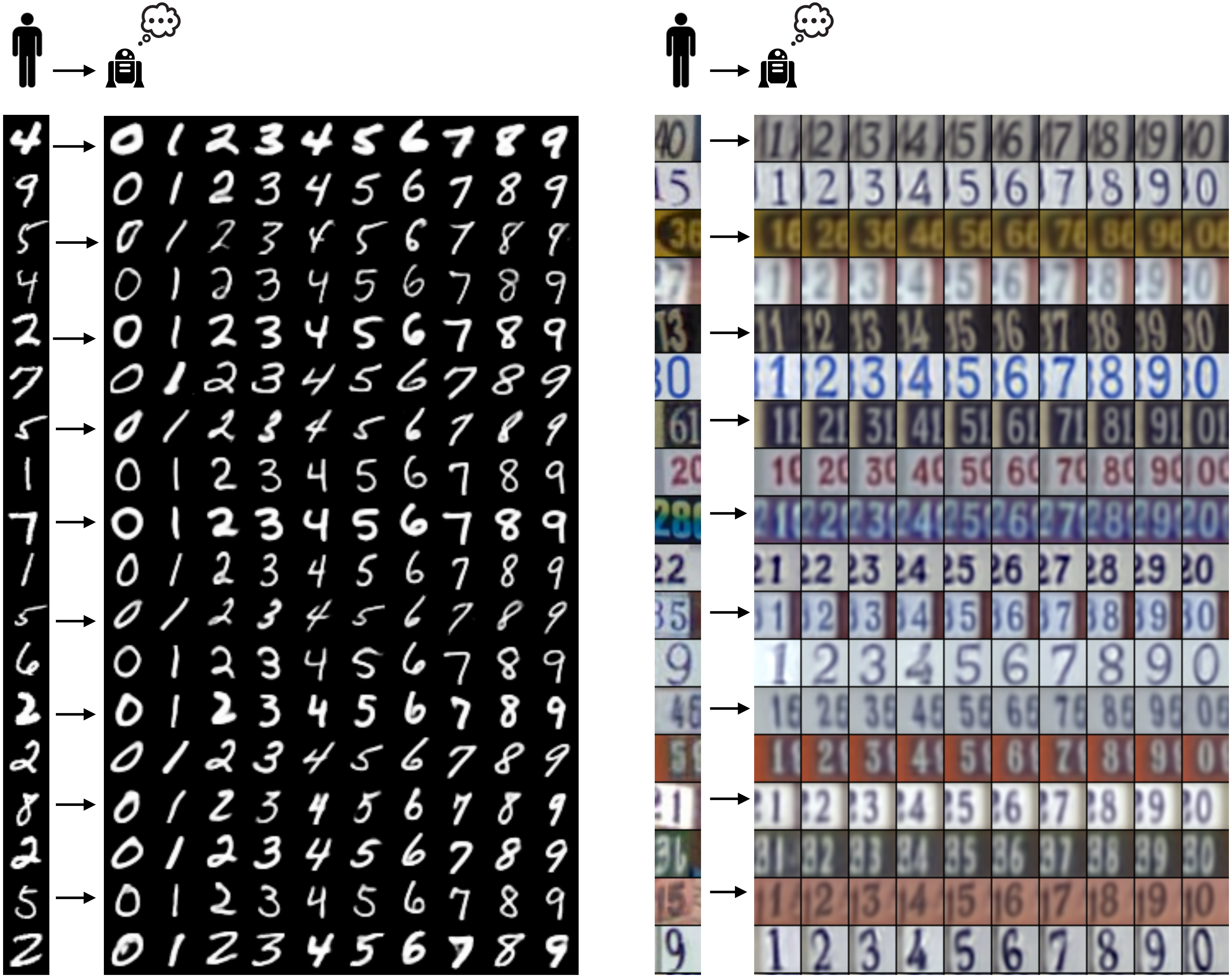
SVHN

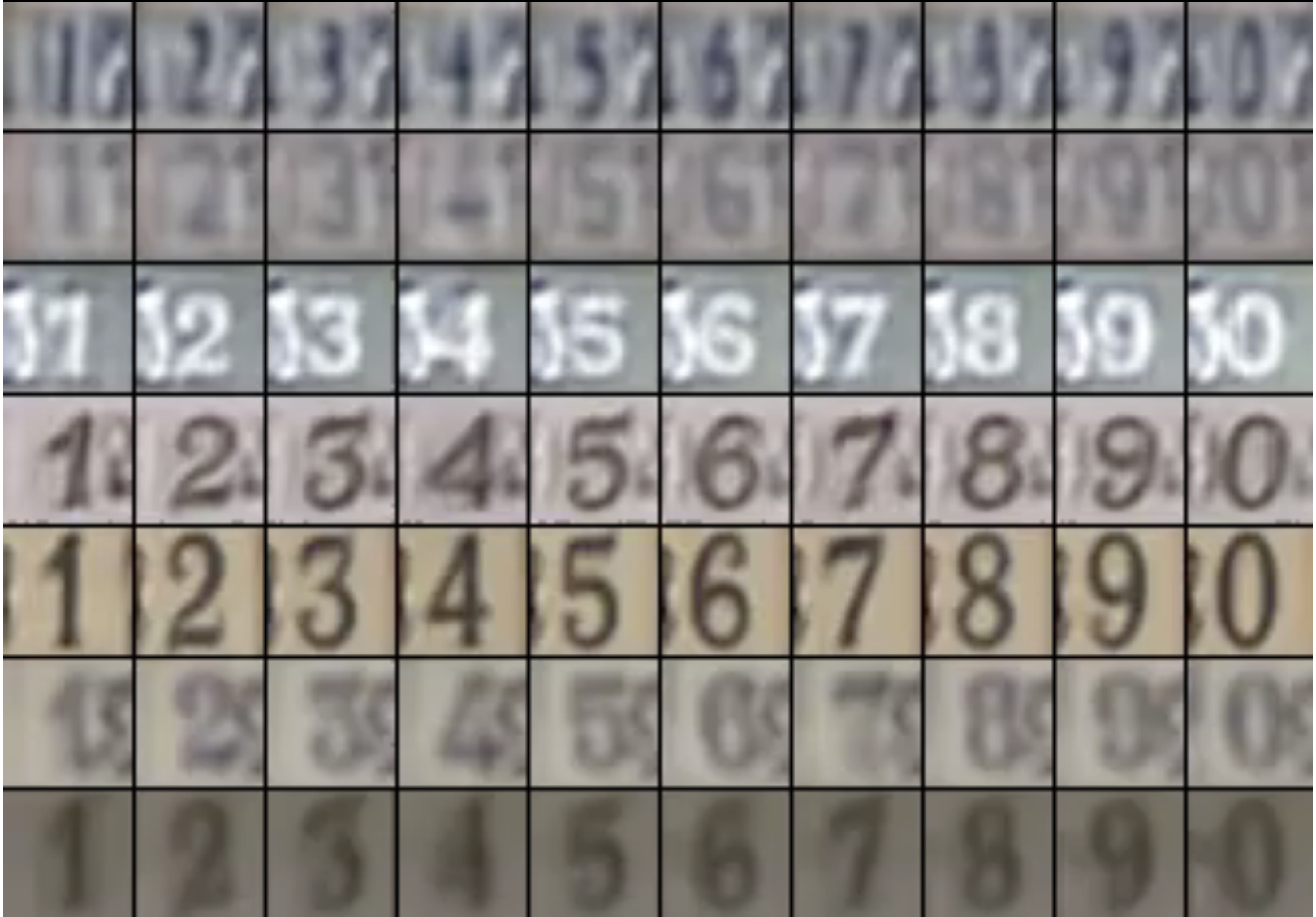
KNN	TSVM	M1+KNN	M1+TSVM	M1+M2
77.93 ($\pm$ 0.08)	66.55 ($\pm$ 0.10)	65.63 ($\pm$ 0.15)	54.33 ($\pm$ 0.11)	**36.02** ($\pm$ 0.10)

NORB

KNN	TSVM	M1+KNN	M1+TSVM
78.71 ($\pm$ 0.02)	26.00 ($\pm$ 0.06)	65.39 ($\pm$ 0.09)	**18.79** ($\pm$ 0.05)

Analogy-making





Summary

- Stochastic Gradient Variational Inference:
  “Swiss army knife” for inference
  - Works with almost any model $p(x,z)$
  - Works with almost any approx. posterior $q(z|x)$
  - Just requires gradient ascent on single objective
- Applications:
  - any continuous posterior inference problem
  - deep latent-variable models / **Helmholtz machines**
  - **semi-supervised learning**

https://github.com/dpkingma/nips14-ssl