Efficient Inference and Learning with Intractable Posteriors? Yes, Please.

Diederik (Durk) Kingma

Max Welling

UNIVERSITEIT VAN AMSTERDAM

Parts of this talk

1. Basics of Stochastic Gradient VI (SGVI)

2. Learning deep latent-variable models

3. Semi-supervised learning / analogic reasoning

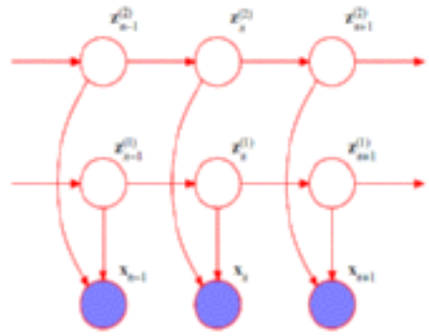
4. SGVI with auxiliary variables

5. SGVI for model parameters
   (local reparameterization)

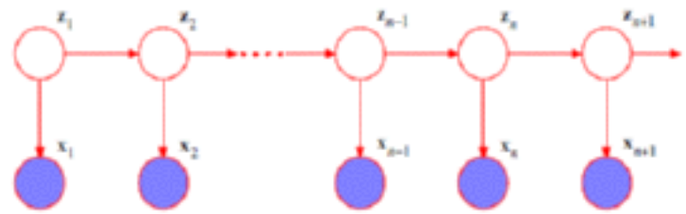
Stochastic Gradient Variational Inference

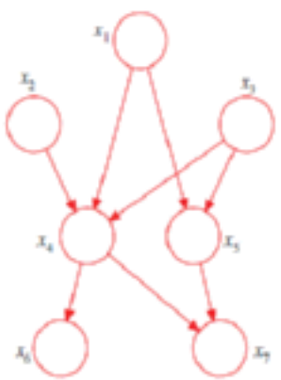
Bayesian inference problem setup

- x: *observed data*
  z: *unobserved/latent variables* or parameters
  $\mathrm{p}(\mathrm{x},\mathrm{z})$: probabilistic model, often factorized



- We are (very) interested in inferring a *posterior distribution* $\mathrm{p}(\mathrm{z}|\mathrm{x})$
  e.g.:
  - Full distribution over model parameters
  - Enables learning parameters in latent-variable models





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

Variational inference

- Introduce **parametric model** $\mathrm{q}_\varphi(\mathrm{z})$ or $\mathrm{q}_\varphi(\mathrm{z}|\mathrm{x})$ of true posterior
  - φ: *variational parameters*
- Objective: **minimize** w.r.t. φ the KL-divergence: $\mathrm{D}_{\mathrm{KL}}(\mathrm{q}_\varphi(\mathrm{z}|\mathrm{x})||\mathrm{p}(\mathrm{z}|\mathrm{x}))$
  - $\mathrm{D}_{\mathrm{KL}}(\mathrm{q}_\varphi(\mathrm{z}|\mathrm{x})||\mathrm{p}(\mathrm{z}|\mathrm{x})) = 0$ at optimum

Variational bound

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

$$\downarrow$$

Objective:

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

Abbreviated:

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

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

- 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. 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:

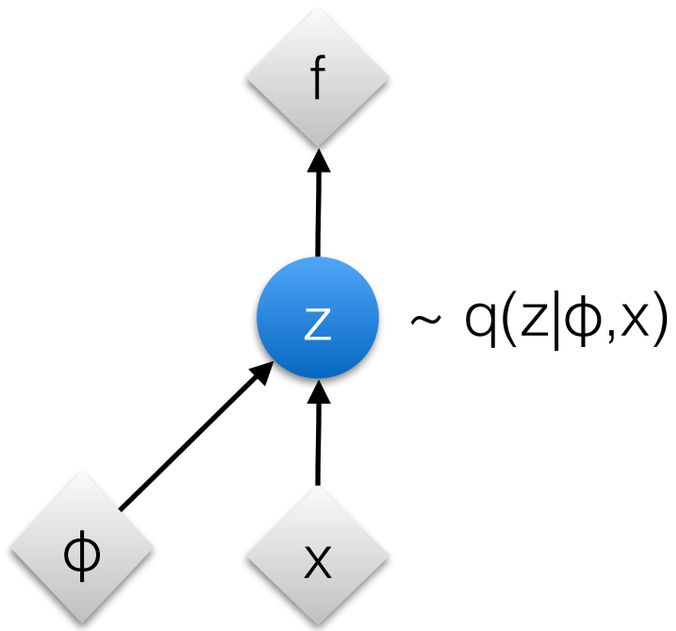
$$\begin{aligned}
&\text{Step 1a: sample } \epsilon^l \text{ from } p(\epsilon) \\
&\text{Step 1b: } z^l = g_\phi(\epsilon), \quad \text{such that } z^l \sim q_\phi(z|x) \\
&\text{Step 2: } L \simeq f_\phi(x, z^l) \\
&\text{Step 3: } \nabla_\phi \mathcal{L} \simeq \nabla_\phi f_\phi(x, z^l)
\end{aligned}$$

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

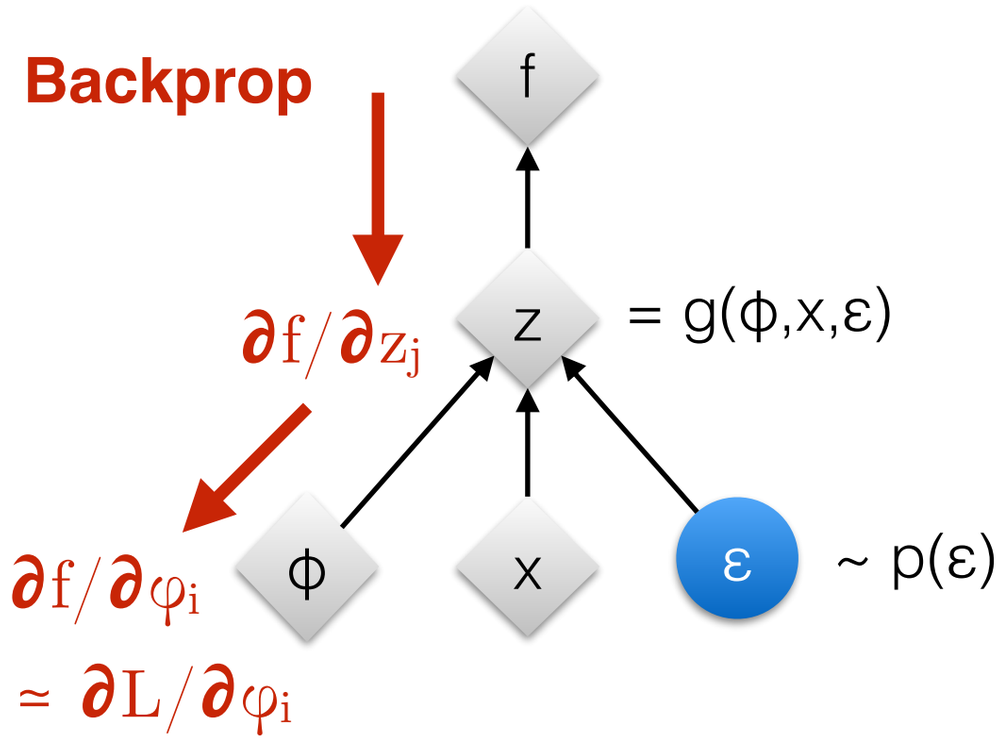
[Kingma and Welling, 2013/2014] Auto-encoding Variational Bayes
[Rezende et al, 2014] Stochastic Backpropagation and Variational Inference in DLVMs
[Titsias and Lázaro-Gredilla, 2014] Doubly Stochastic VB for non-Conjugate Inference

Reparameterization trick

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

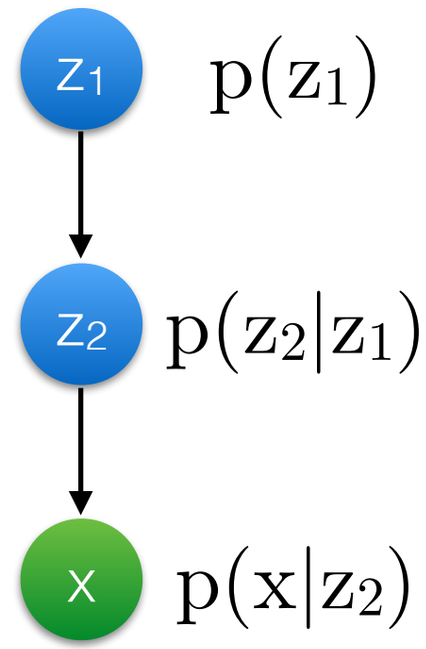
- **Variational inference by gradient ascent**
- **“Swiss army knife” for inference:**
  - Works with almost any $\mathrm{p}(\mathrm{x},\mathrm{z})$
  - Works with almost any $\mathrm{q}(\mathrm{z}|\mathrm{x})$
  - Just requires gradient ascent on single objective

Deep latent-variable models

Deep Latent-Variable Models

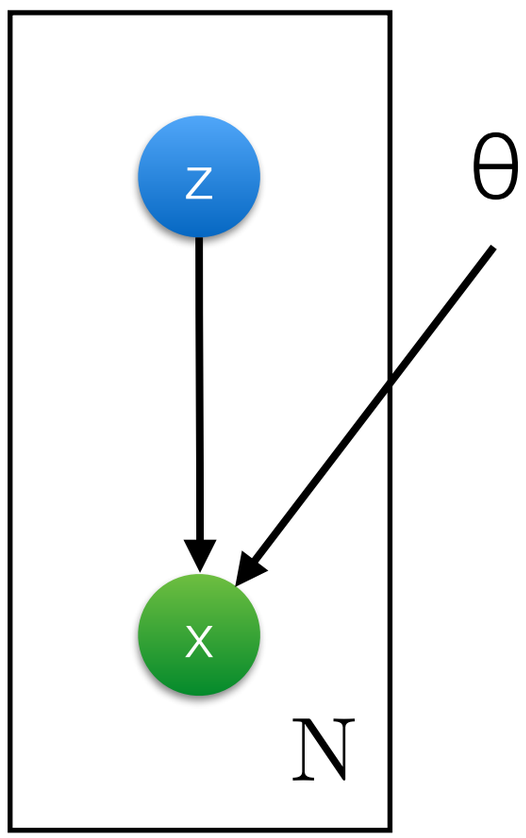
- We combining the strengths of **deep neural nets** with those of **latent-variable models**
  - **directed latent variables models**: can represent complicated **marginal distributions** over x
    - probabilistic **deep neural nets**: can represent complicated conditional dependencies $\mathrm{p}(\mathrm{y}|\mathrm{x})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

  $=>$ Approximate inference



$$\mathrm{p}(\mathrm{x},\mathrm{z}_1,\mathrm{z}_2) = \mathrm{p}(\mathrm{x}|\mathrm{z}_2)\mathrm{p}(\mathrm{z}_2|\mathrm{z}_1)\mathrm{p}(\mathrm{z}_1)$$

Example model



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

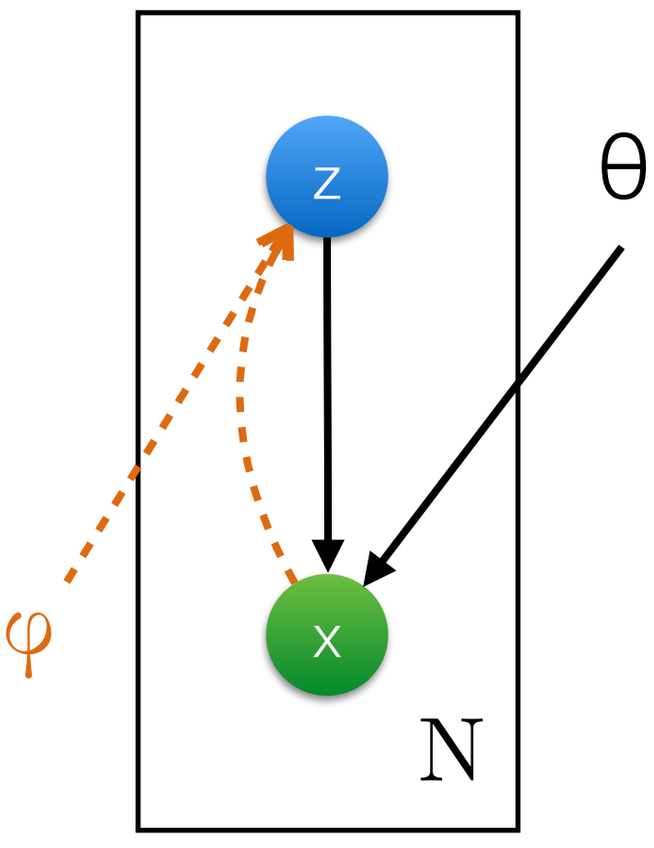
x: e.g. Face

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

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

Auto-Encoding Variational Bayes

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



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

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

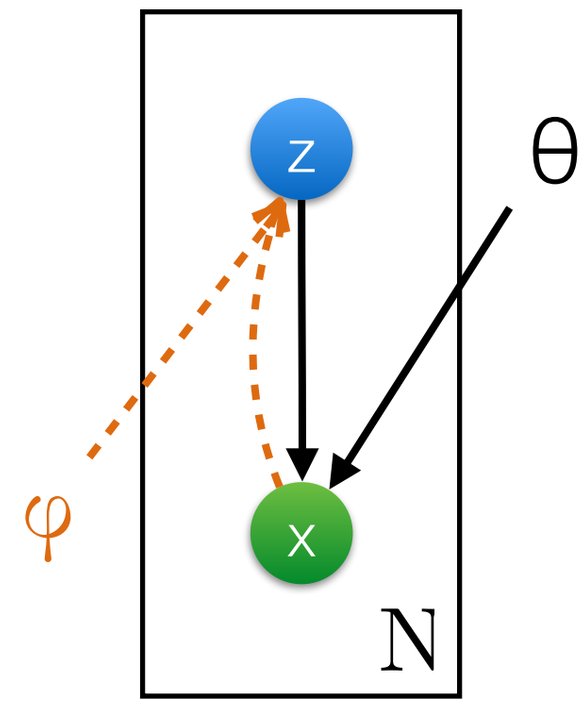
$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^i)} \left[\log p_\theta(x^i, z) - \log q_\phi(z|x^i)\right]$$

- Jointly optimized w.r.t. φ and θ

- Doubly stochastic optimization:
  - Sampling small minibatches of data
  - Sampling from approx posterior

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

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

Connection to Helmholtz Machine / Wake-Sleep

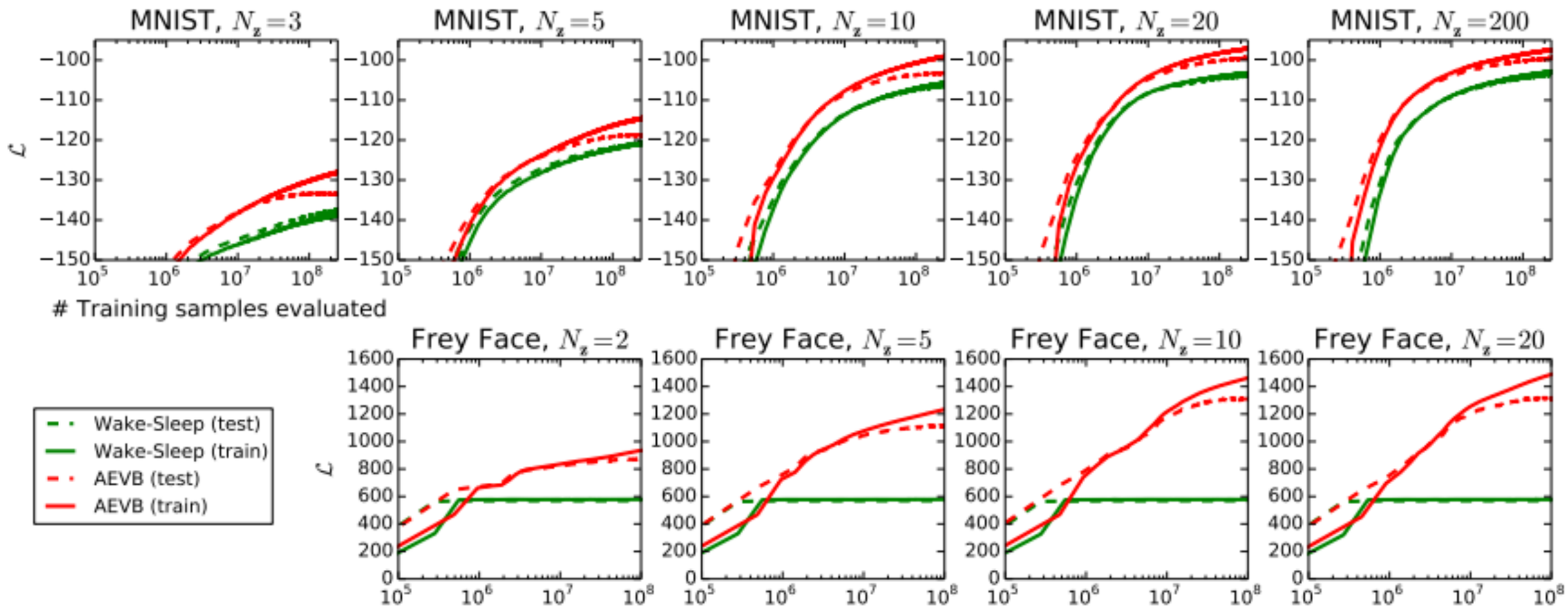
(1994/1995, Dayan/Hinton/Frey/Neal, *Science*)

- Also introduced **parametric inference model** q(z|x) learned with **gradient ascent**

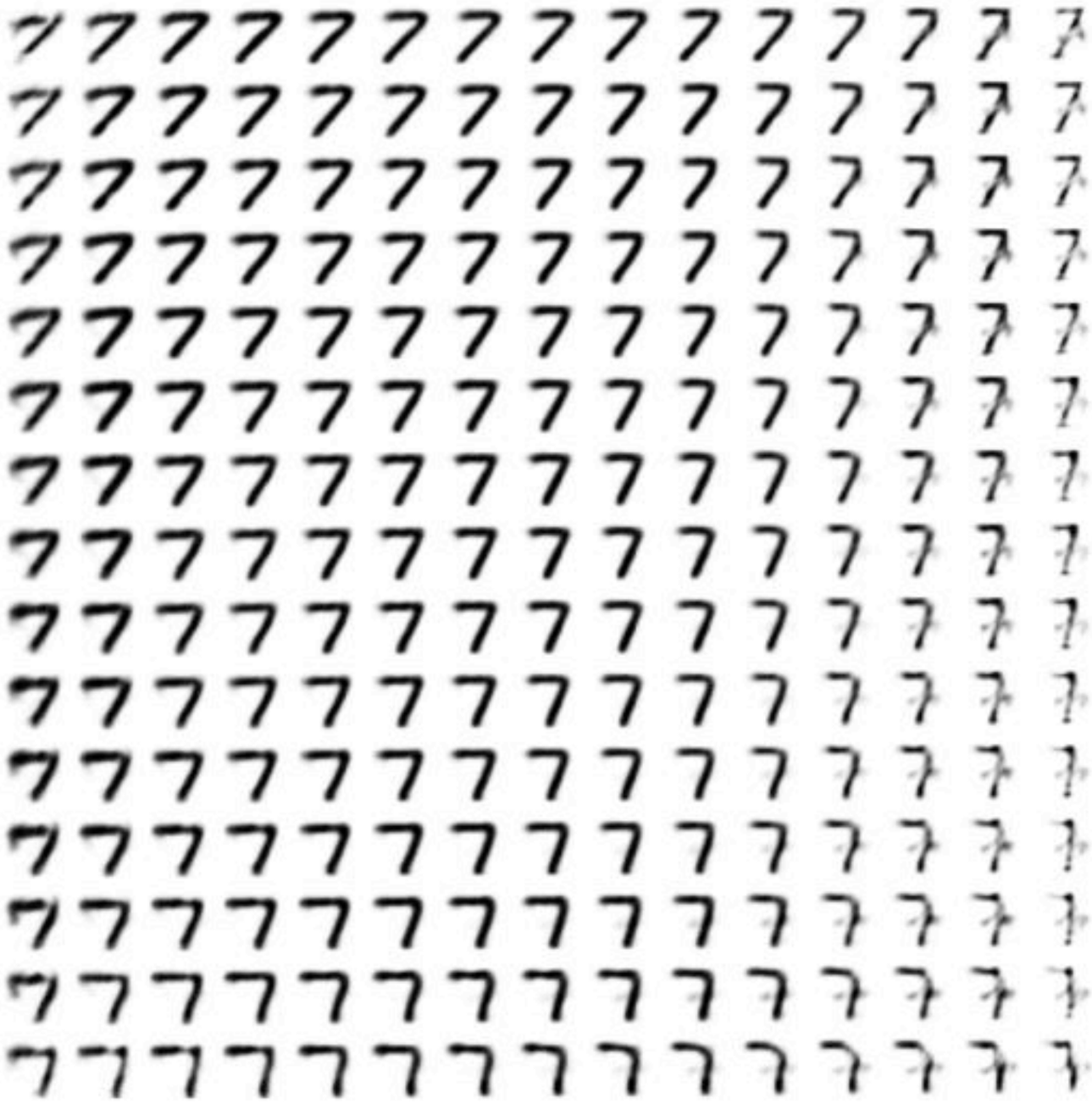
- But the wake-sleep algorithm used **incorrect gradient for *q***

- **Main difference is: now we know how to learn Q correctly with gradient ascent**

AEVB vs Wake-Sleep



3D latent space



Conv. net as encoder/decoder, trained on faces

(trained by Alec Radford 2015)

Semi-Supervised Learning with Deep Generative Models

NIPS 2014

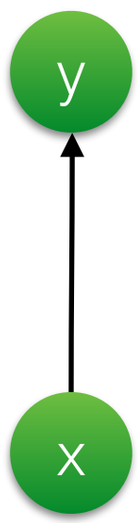
Diederik P. Kingma
Danilo J. Rezende
Shakir Mohamed
Max Welling

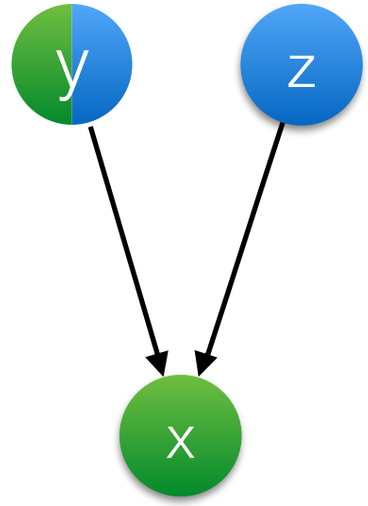
UNIVERSITEIT VAN AMSTERDAM

Google DeepMind

Classifier vs generative model

Classifier vs Generative model

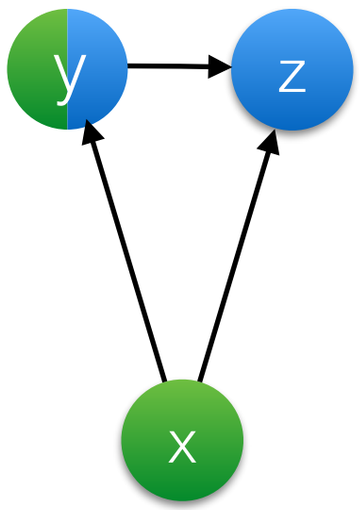




Each edge is parameterised as a deep neural net

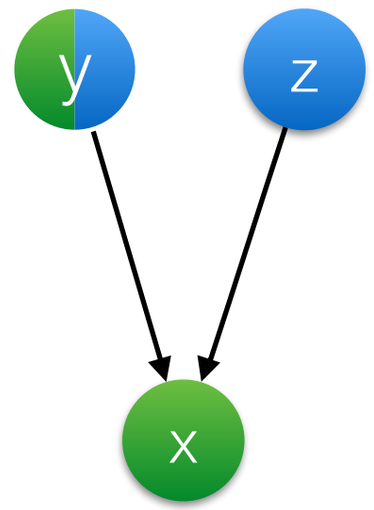
Generative model for semi-supervised learning

Inference model



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

Generative model

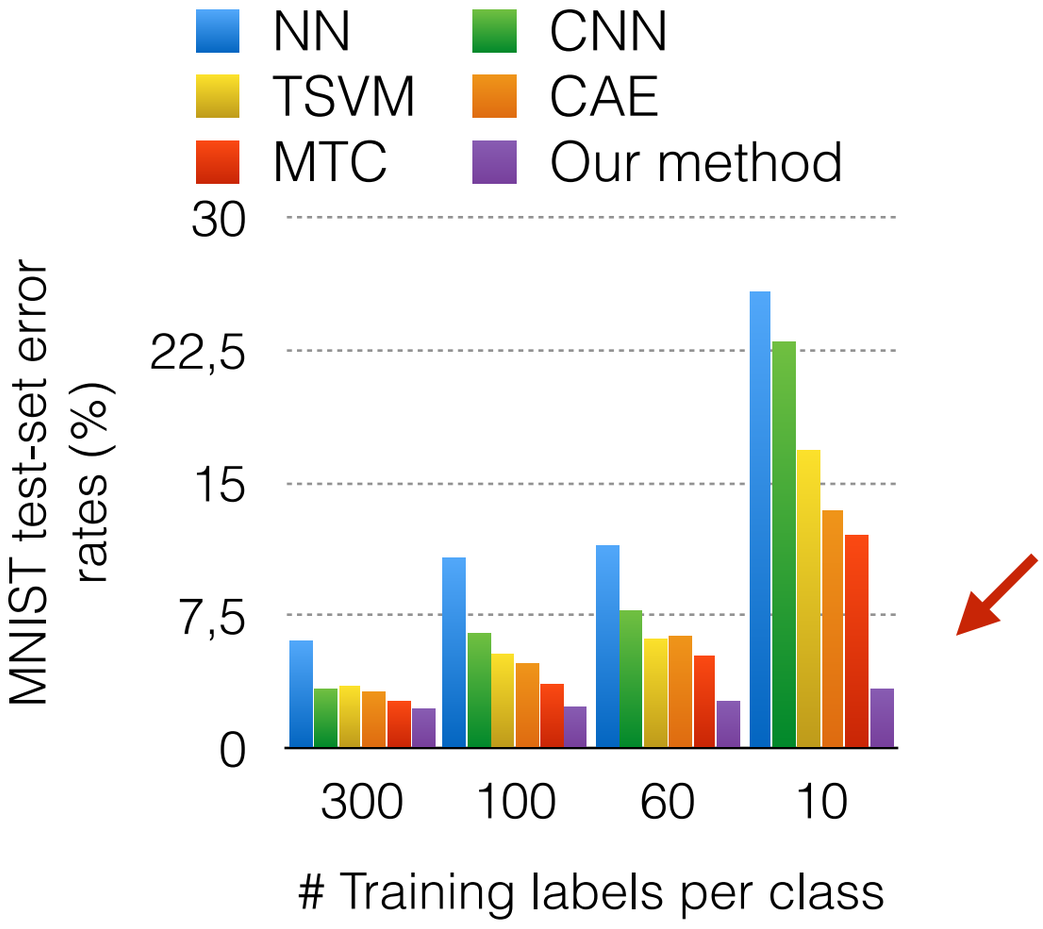


Variational training includes optimisation of q(y|x)

Classification Results
(SOTA in 2014)

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)



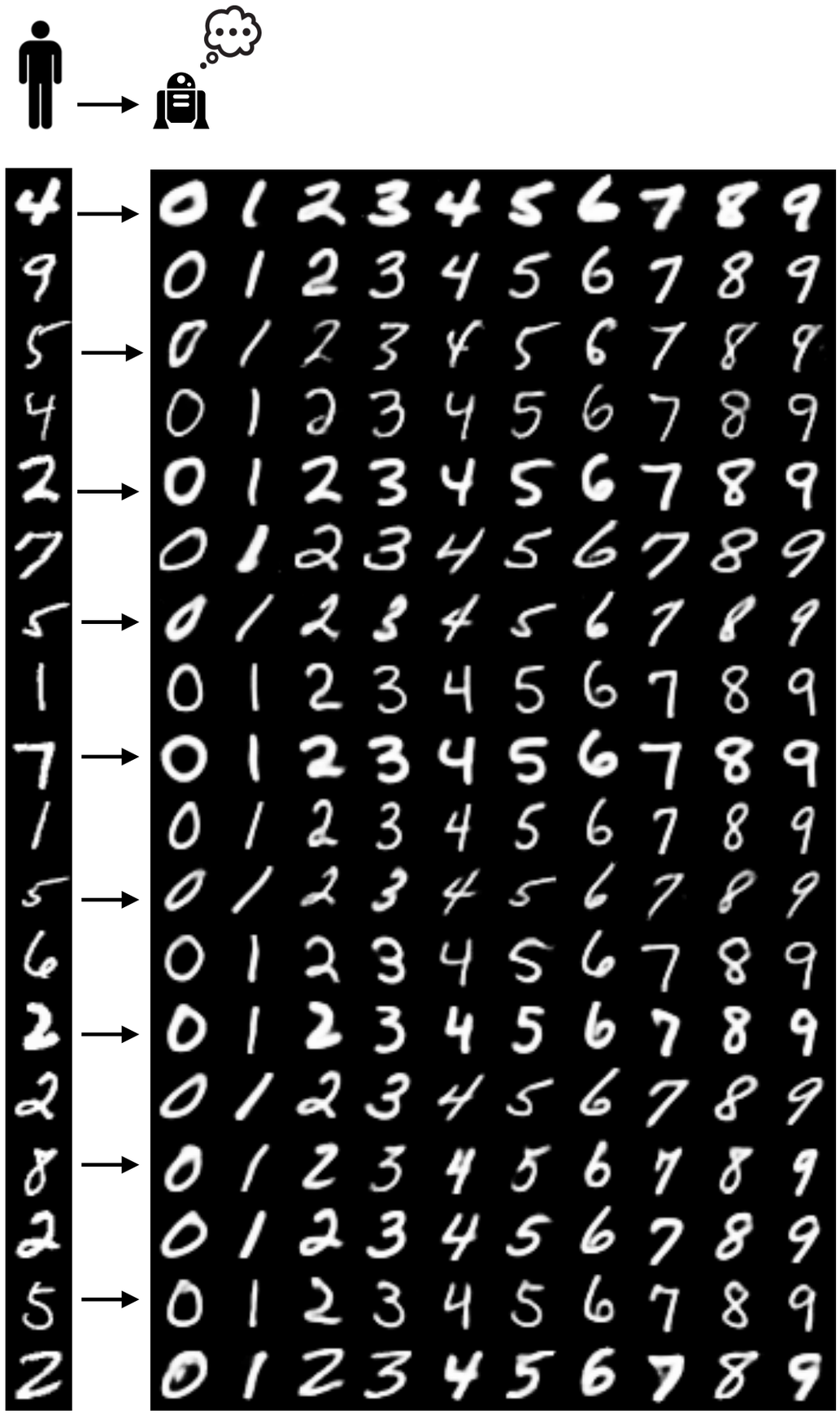
SVHN

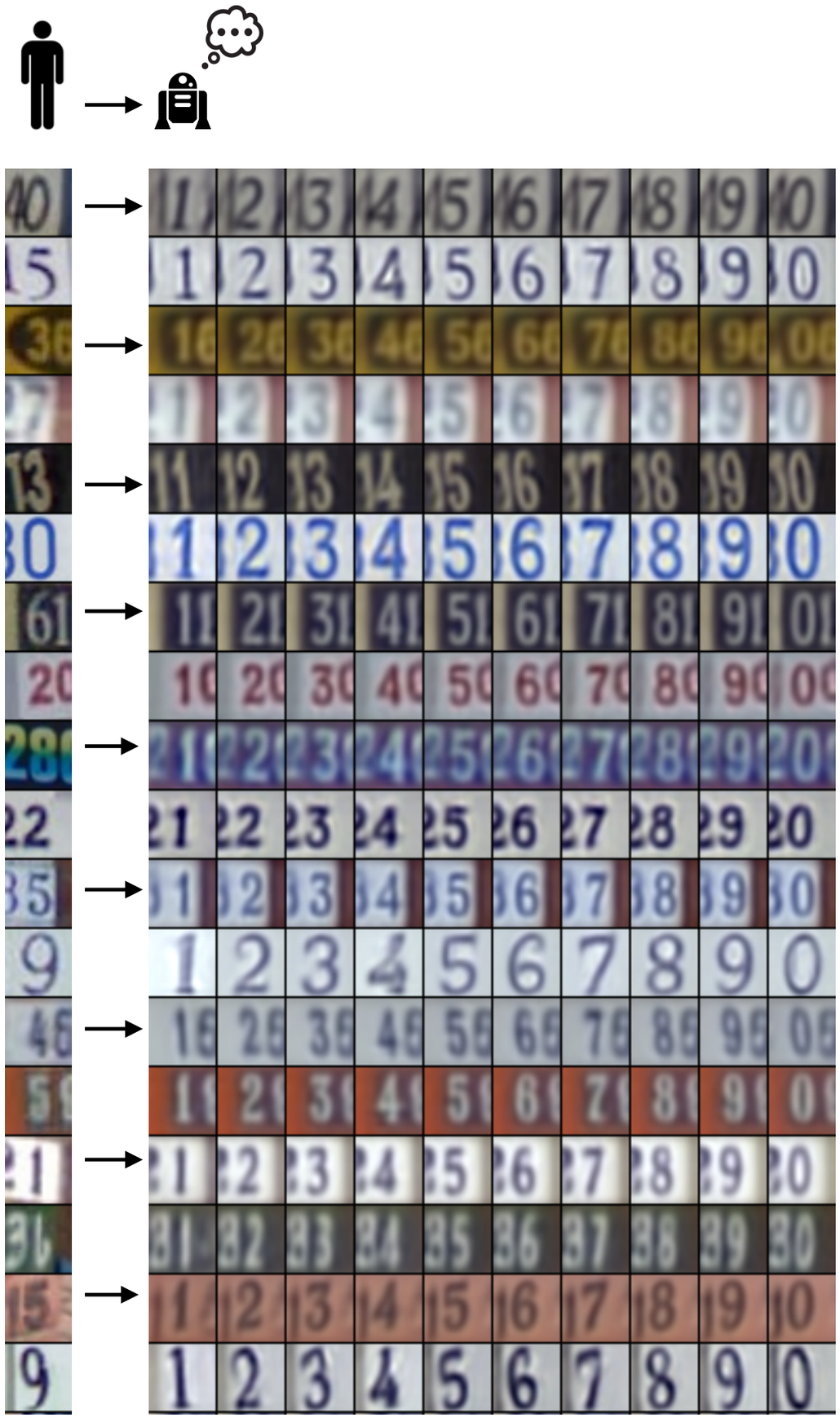
KNN	TSVM	M1+KNN	M1+TSVM	M1+M2
77.93 ($\pm$ 0.08)	66.55 ($\pm$ 0.10)	65.63 ($\pm$ 0.15)	54.33 ($\pm$ 0.11)	**36.02** ($\pm$ 0.10)

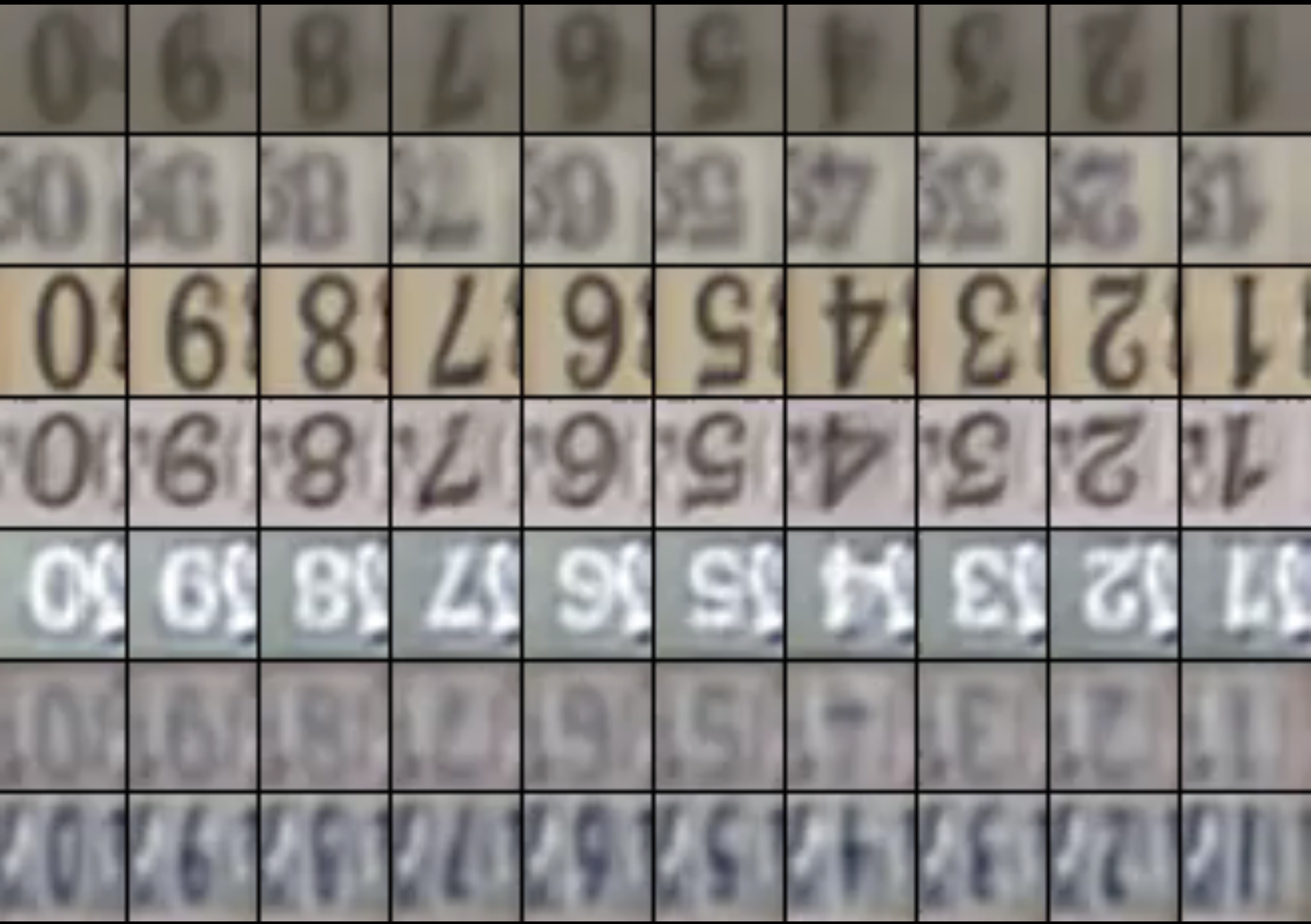
NORB

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







SGVI with auxiliary variables

SGVI with auxiliary variables (1)

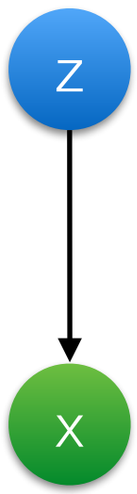
- Main idea 1: Flexible inference models through **auxiliary variables**
  - optimization through a **new bound**
- Main idea 2: Use **MCMC chain** as VI with auxiliary variables
  - **gradient-based optimization of MCMC kernels**
- Examples: Gibbs with over-relaxation, Hamiltonian VI

MCMC and Variational Inference: Bridging the Gap

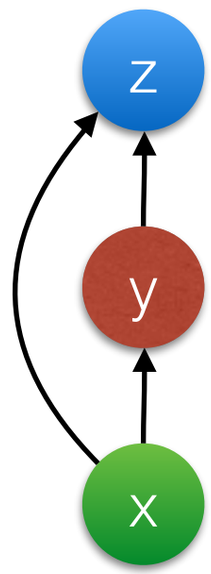
Tim Salimans, **Diederik P. Kingma**, Max Welling, ICML 2015

SGVI with auxiliary variables (2)

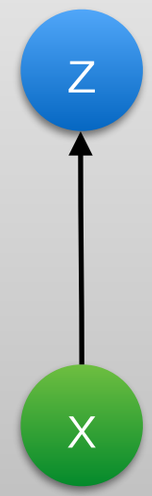
Generative model
$\mathrm{p}(\mathrm{x},\mathrm{z})$



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

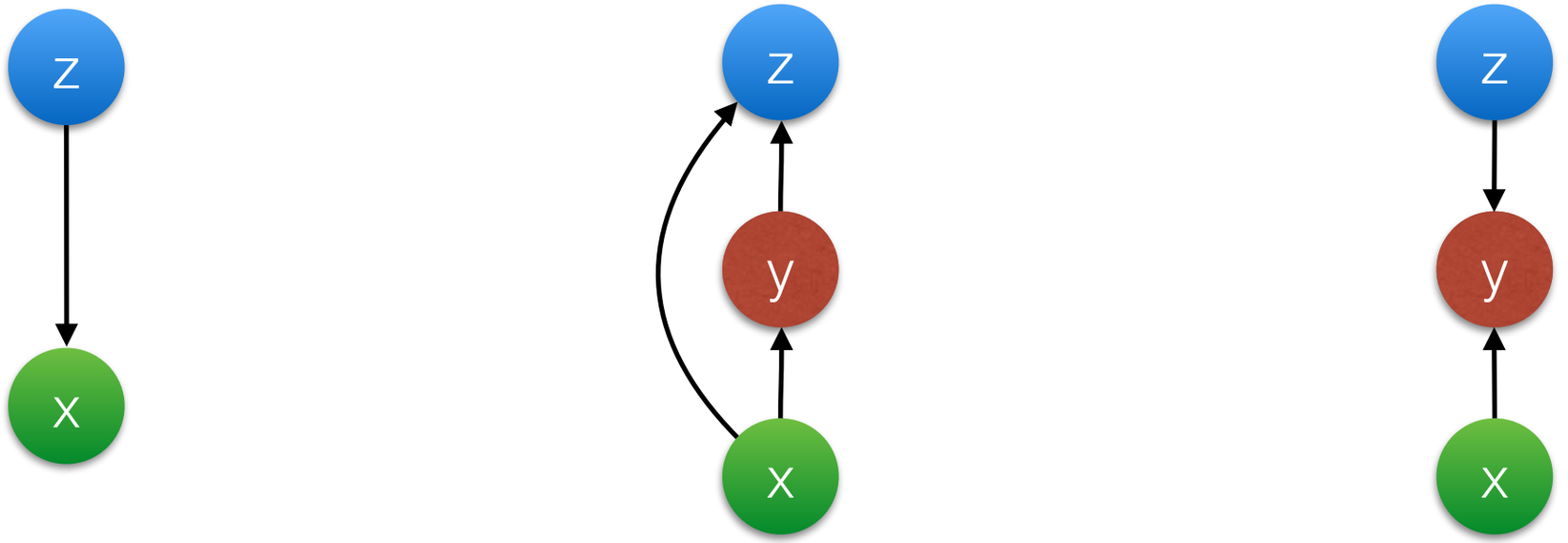
SGVI with auxiliary variables (3)

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!



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

SGVI with auxiliary variables (4)

Regular bound:

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

Bound with auxiliary variables:

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

where: $q_\phi(z|x) = \int q_\phi(z,y|x)\,dy$

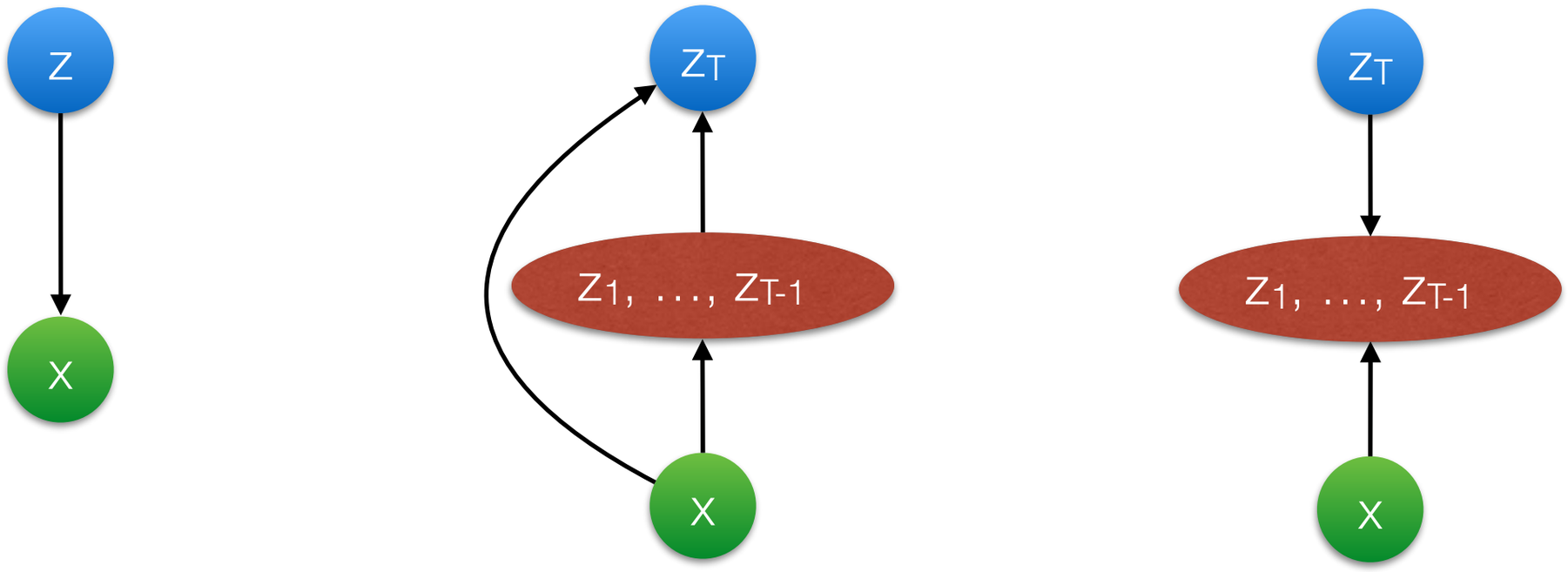
- Better than regular bound
  if $\mathrm{D_{KL}(q||p)}$ sufficiently improves
  and $\mathrm{E[D_{KL}(q|||r)]}$ remains sufficiently small
  -> requires flexible $\mathrm{r(y|z,x)}$

Markov Chain Variational Inference (MCVI)

Generative model
$p(x,z)$

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

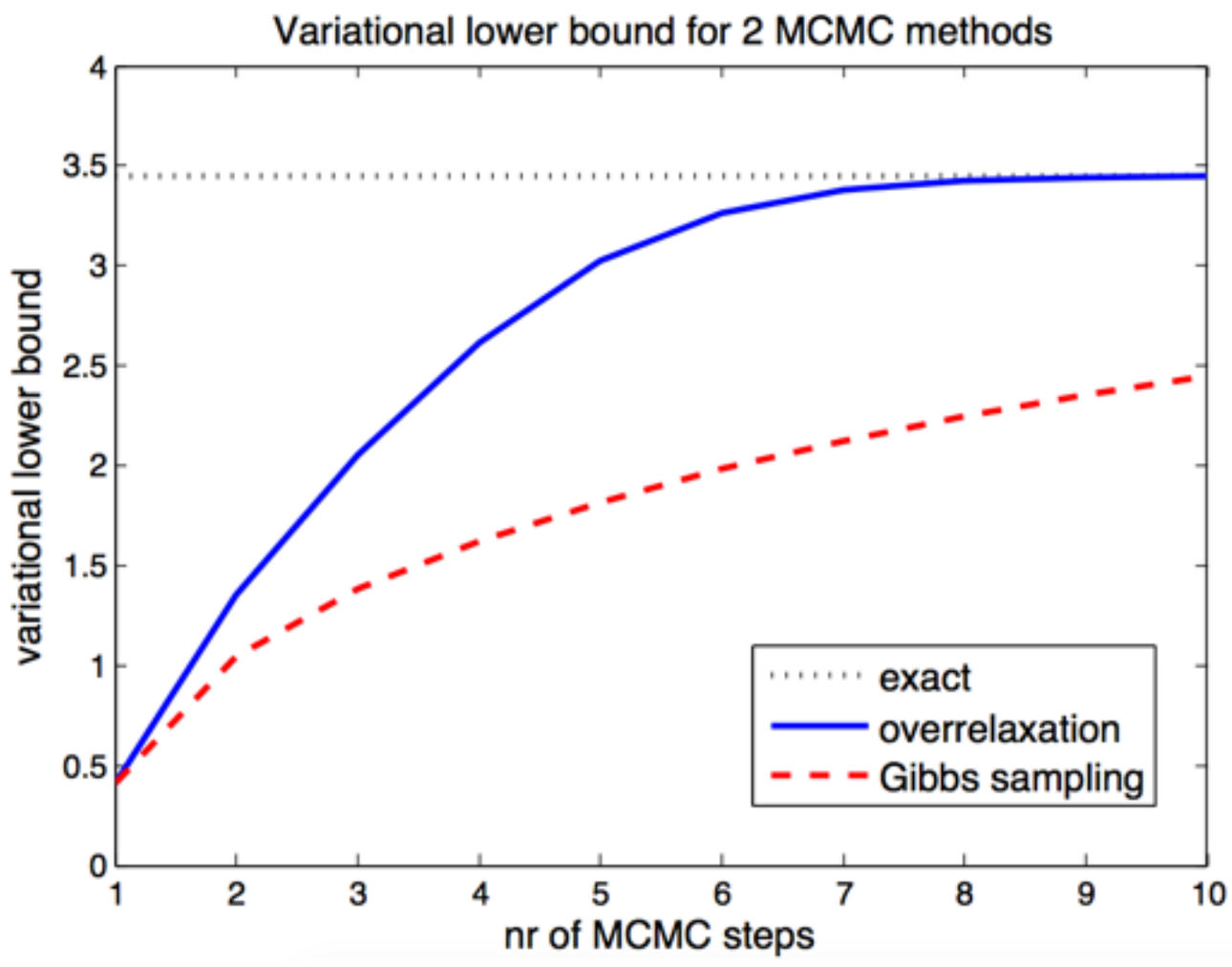
Auxiliary model
$r(z_1,...,z_{T-1}|z_T,x)$



MCMC chain
as inference model
with auxiliary variables

Example 1: Gibbs sampling with over-relaxation

- (toy) Posterior: bivariate Gaussian
- Inference model: Gibbs sampling with over-relaxation **[Adler ’81]**
- Auxiliary (inverse) model: linear Gaussian



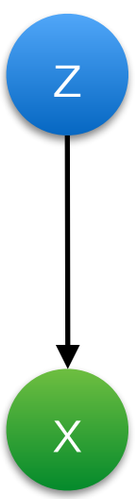
Hamiltonian VI without MH (HVI)

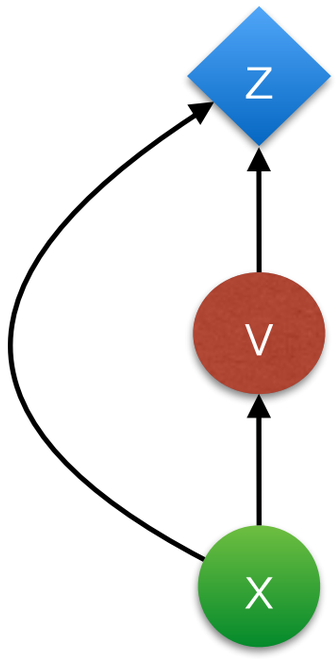
Generative model
$\mathrm{p(x,z)}$

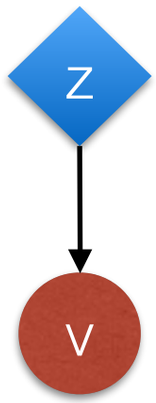
Inference model
$\mathrm{q(v|x)}$

Auxiliary model
$\mathrm{r(v|z)}$

$$\mathrm{z = Hamiltonian\_Dynamics(v,x)}$$







Example: HVI applied to beta-binomial

Gaussian since
$\mathrm{q(v|x)} = \mathrm{Normal()}$

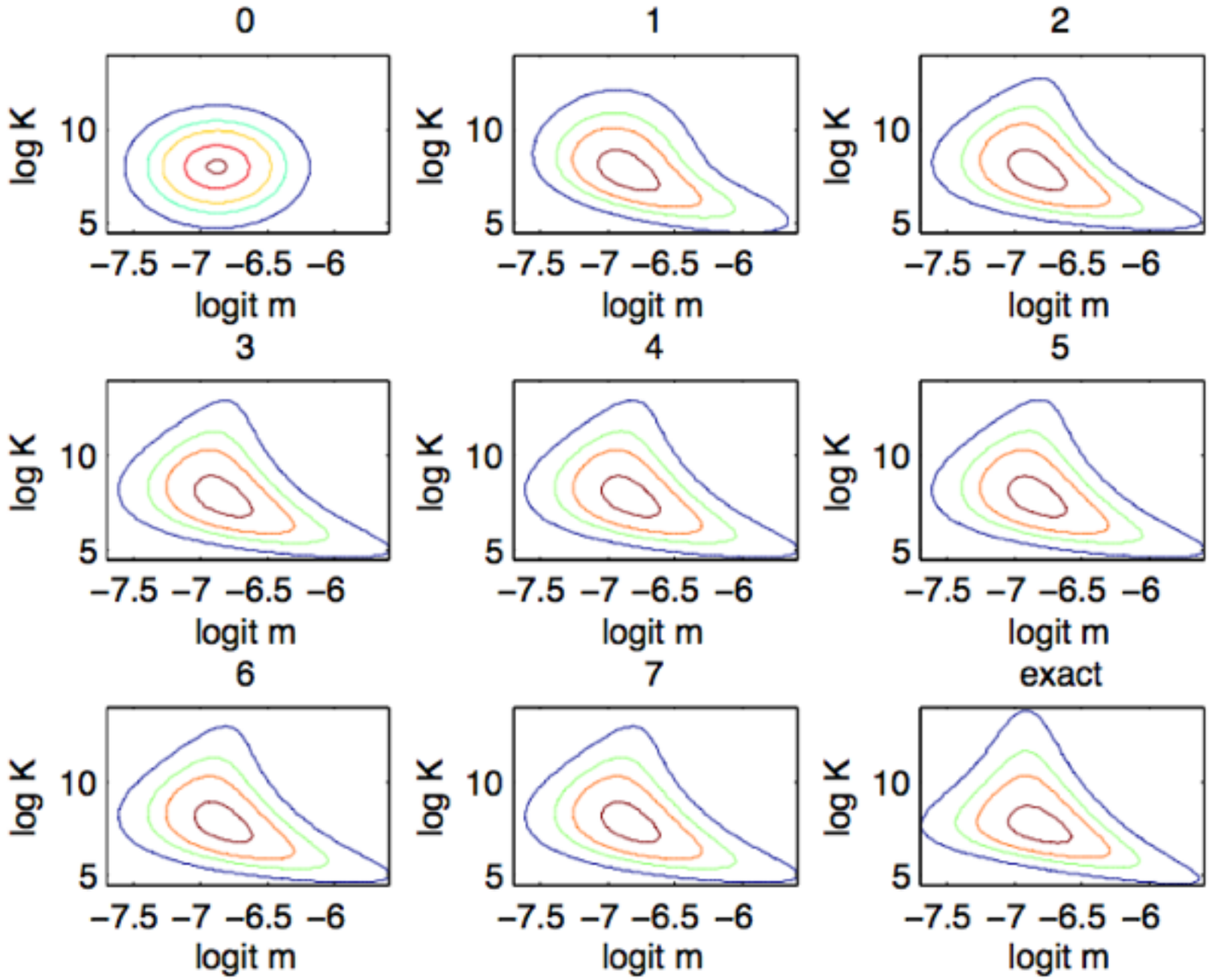


Table 1. Comparison of our approach to other recent methods in the literature. We compare the average marginal log-likelihood measured in nats of the digits in the MNIST test set. See section 3.2 for details.

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	

Auxiliary variables: Summary

- Main idea 1:
  VI with auxiliary variables through new bound
  => more flexible inference

- Main idea 2:
  MCMC chain as VI with auxiliary variables
  => optimize MCMC parameters through the variational bound

- Examples: Gibbs with over-relaxation, Hamiltonian VI

SGVB for parameters of deep neural architectures

SGVB for parameters of deep neural architectures

- Most parameters reside in **linear operations**
  - Fully connected layers, convolutional layers
- SGVI efficiency depends on:
  1. **Variance** of gradients
  2. Hardware utilisation in multicore/GPU architectures
- **Goal:** Low variance, full hardware utilisation

SGVB for global parameters

- $(\mathrm{X},\mathrm{Y}) = \{\mathrm{x}^{\mathrm{i}},\mathrm{y}^{\mathrm{i}}\}_{\mathrm{i}=1..\mathrm{N}}$: dataset
- $\mathrm{p}(\mathrm{y}|\mathrm{x},\mathrm{w})$: classification model (e.g. deep net)
- $\mathrm{q}_{\varphi}(\mathrm{w})$: variational approx. posterior

$$\mathcal{L}(\phi) = -D_{KL}(q_\phi(\mathbf{w})||p(\mathbf{w})) + L_{\mathcal{D}}(\phi)$$

$$\text{where} \quad L_{\mathcal{D}}(\phi) = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \mathbb{E}_{q_\phi(\mathbf{w})}\left[\log p(\mathbf{y}|\mathbf{x},\mathbf{w})\right]$$

Variance of bound estimator

Minibatch-based estimator:

$$L_{\mathcal{D}}^{\mathrm{SGVB}}(\phi) = \frac{N}{M} \sum_{i=1}^{M} L_i(\epsilon, \phi)$$

Variance of estimator:

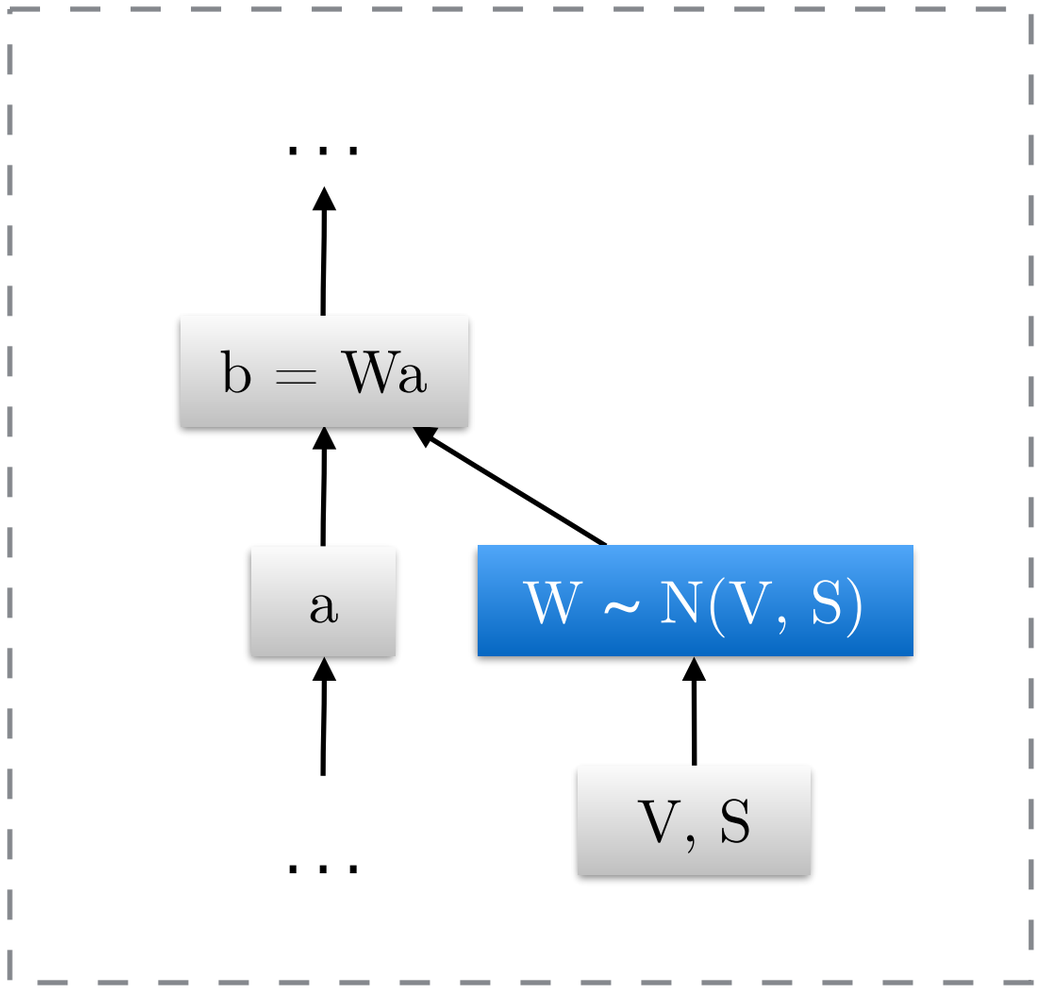
$$\mathrm{Var}_{\epsilon,S}\left[L_{\mathcal{D}}^{\mathrm{SGVB}}(\phi)\right] = \frac{N}{M}\left(\sum_{i=1}^{M} \mathbb{E}_{\epsilon}\left[L_i\right]^2 + \mathrm{Var}_{\epsilon}\left[L_i\right]\right) + 2\sum_{i>j} \mathrm{Cov}_{\epsilon}\left[L_i, L_j\right]$$

0 for independent $\mathrm{L_i}$

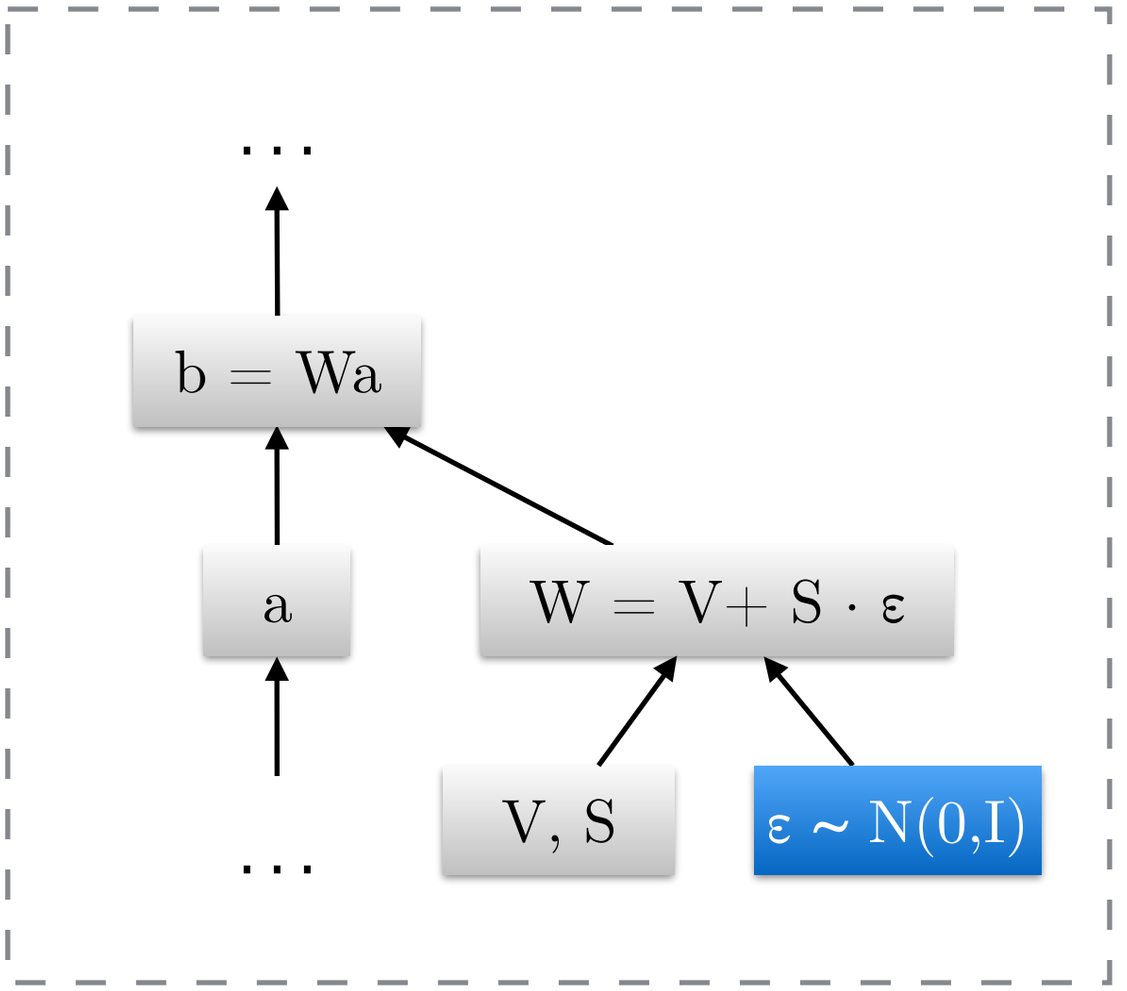
=> We want independent noise per datapoint

Non-local parameterisation

Centered (CP)

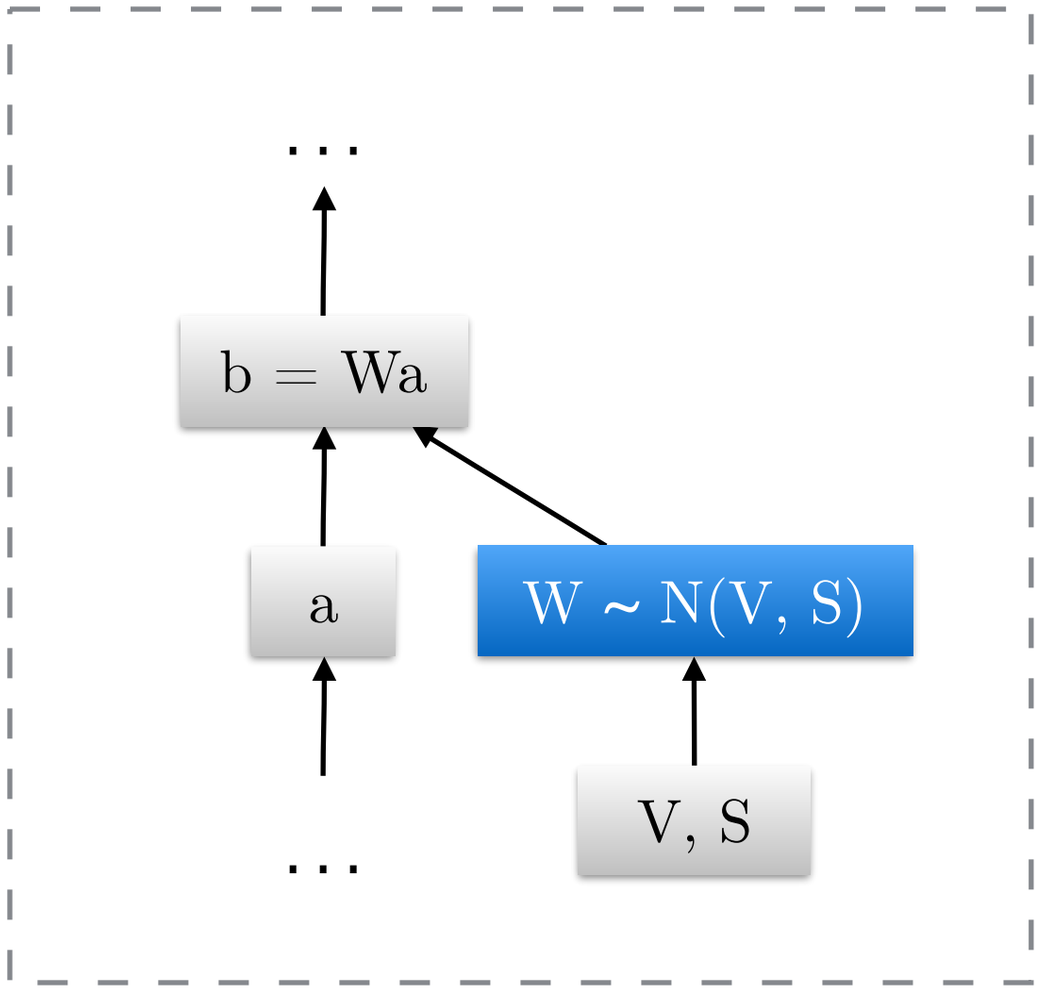


Non-centered (NCP)

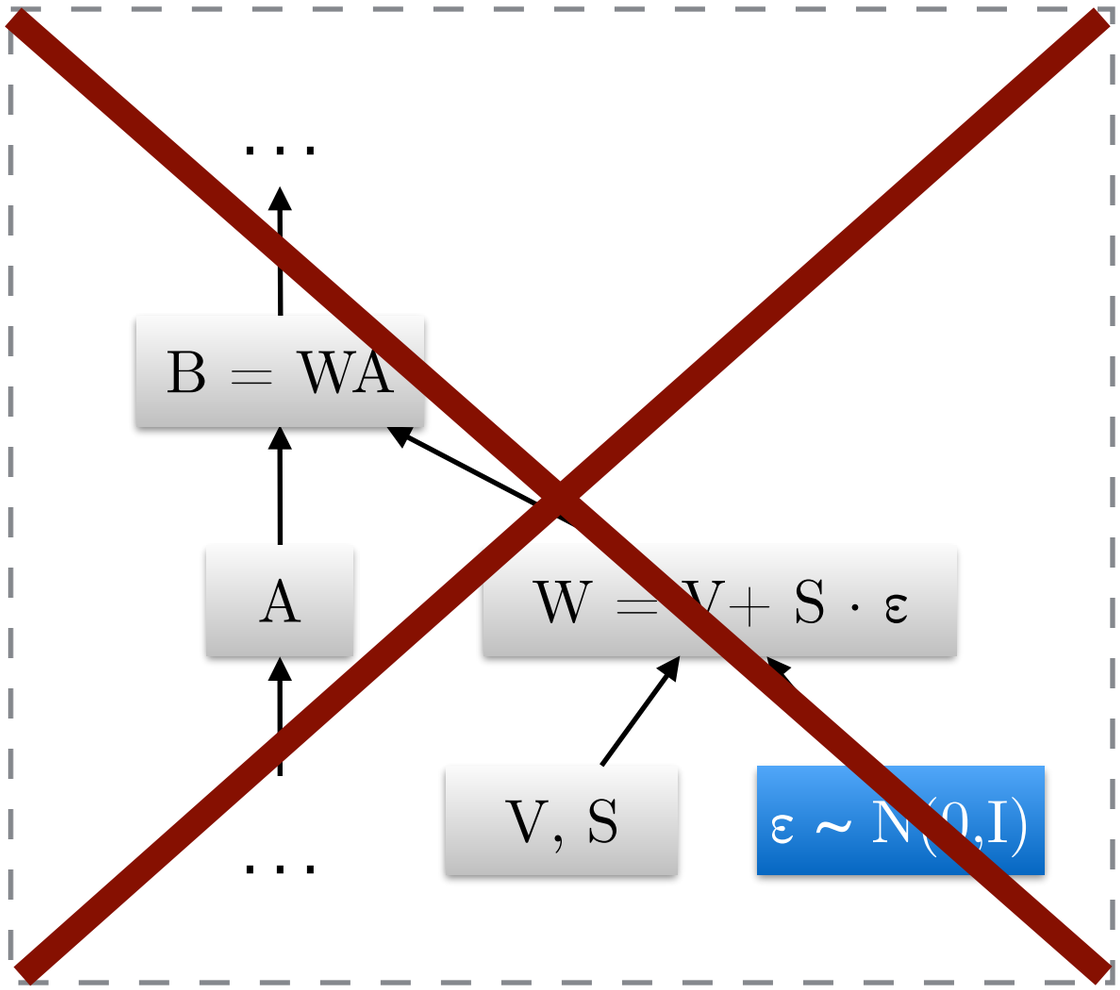


Non-local parameterisation

Centered (CP)



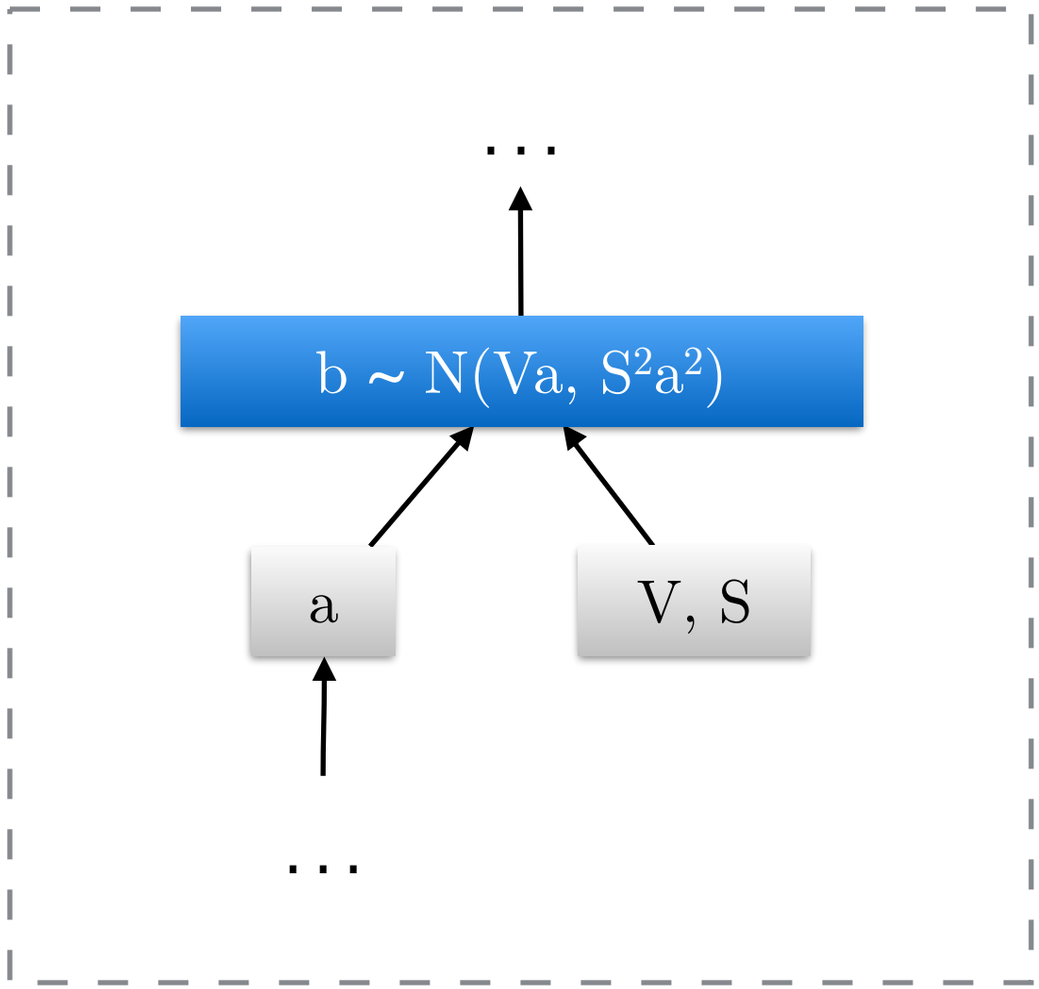
Non-centered (NCP)



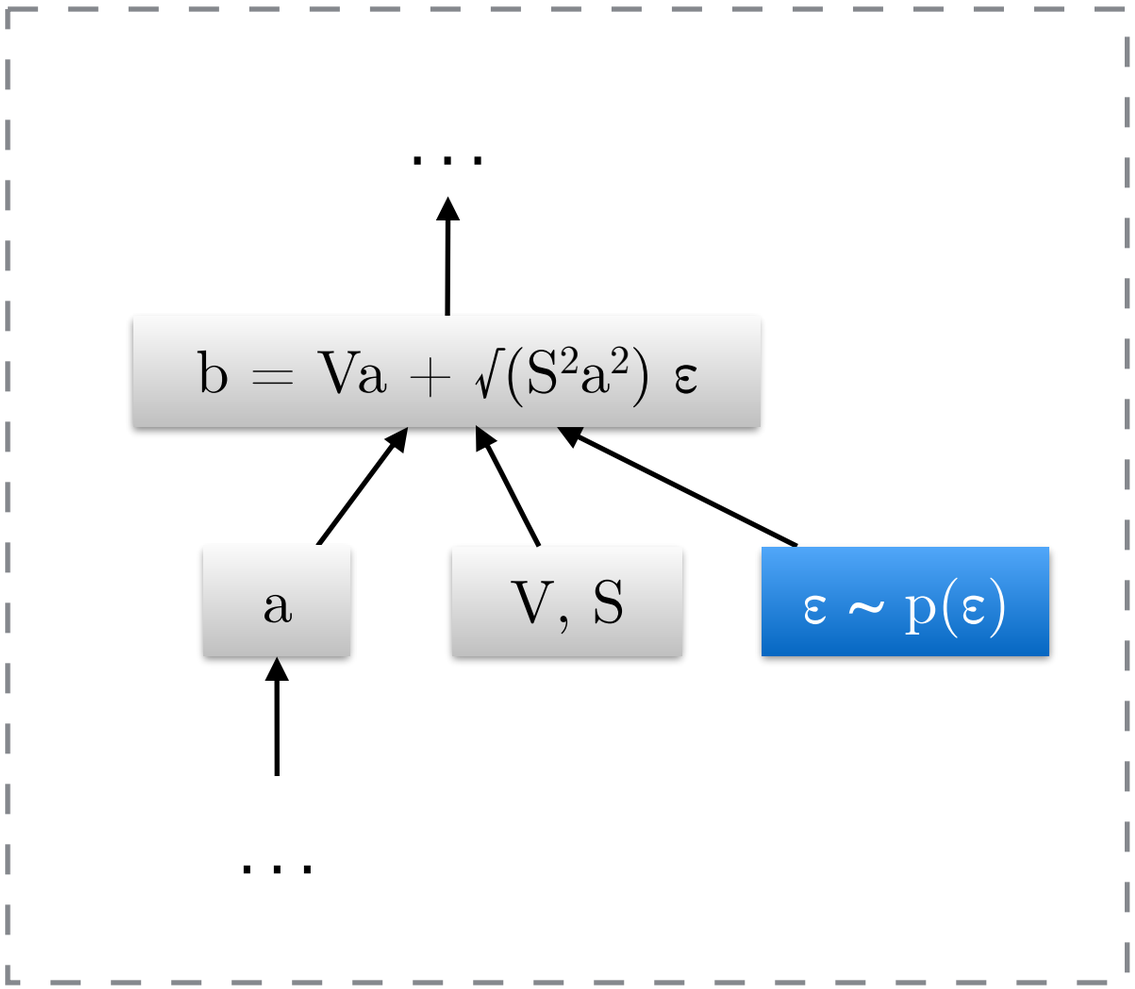
No minibatch parallelism AND independent noise

Local parameterisation

Centered (CP)

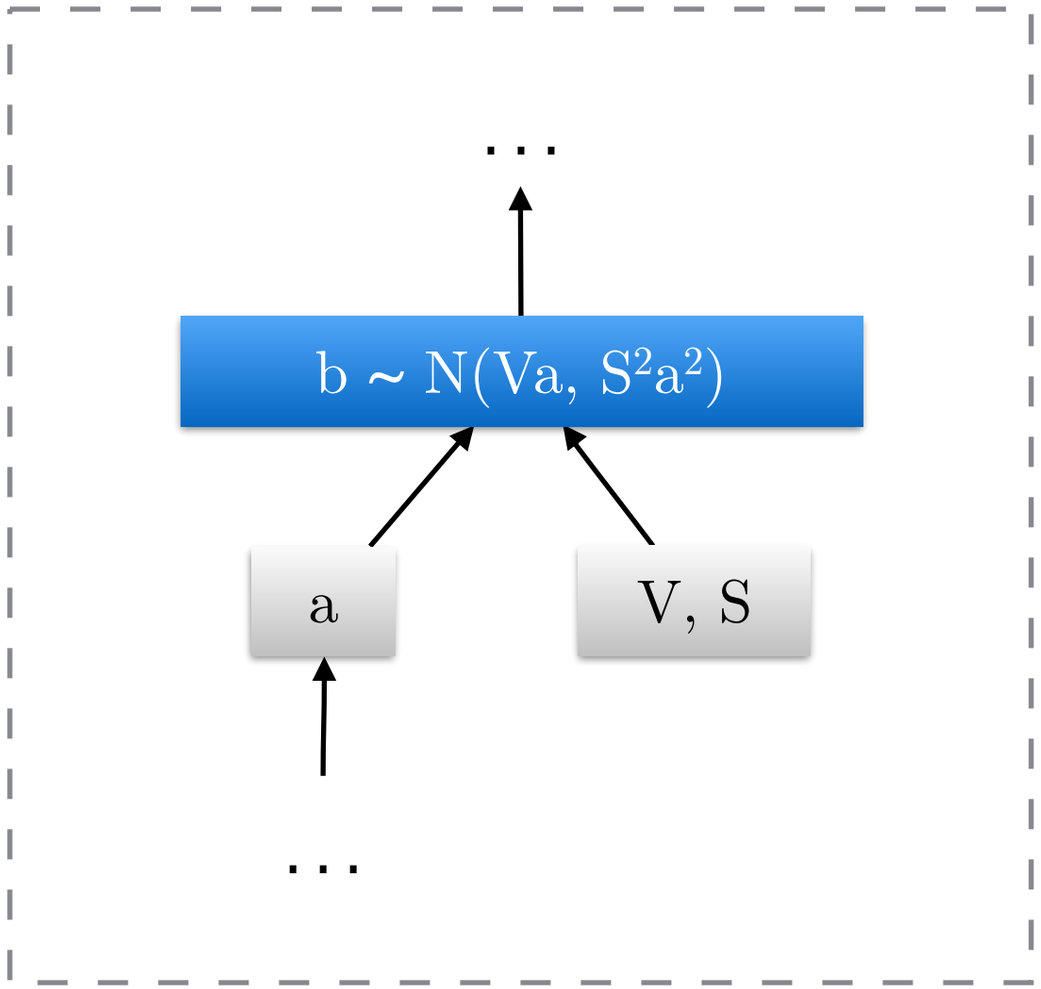


Non-centered (NCP)

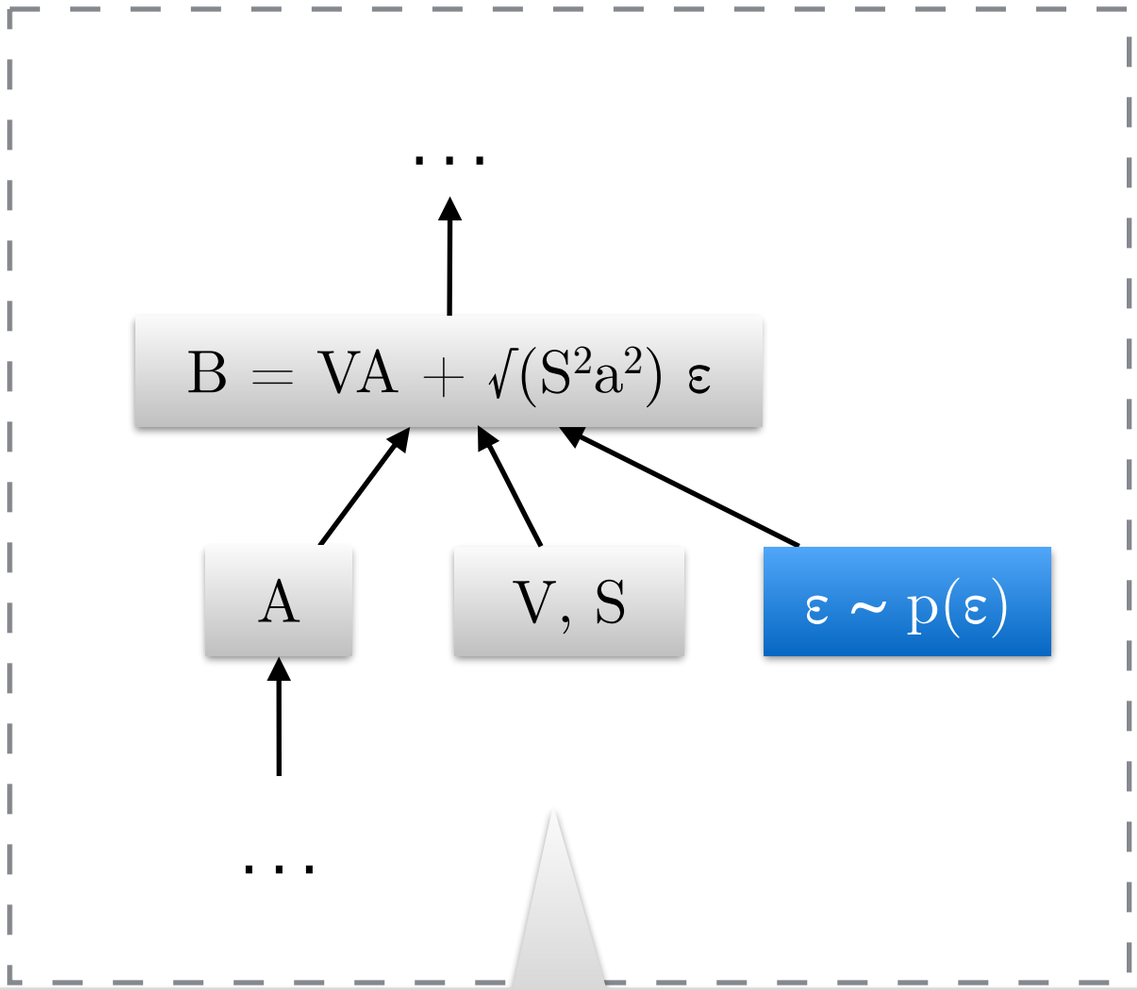


Parallelised local parameterisation

Centered (CP)



Non-centered (NCP)



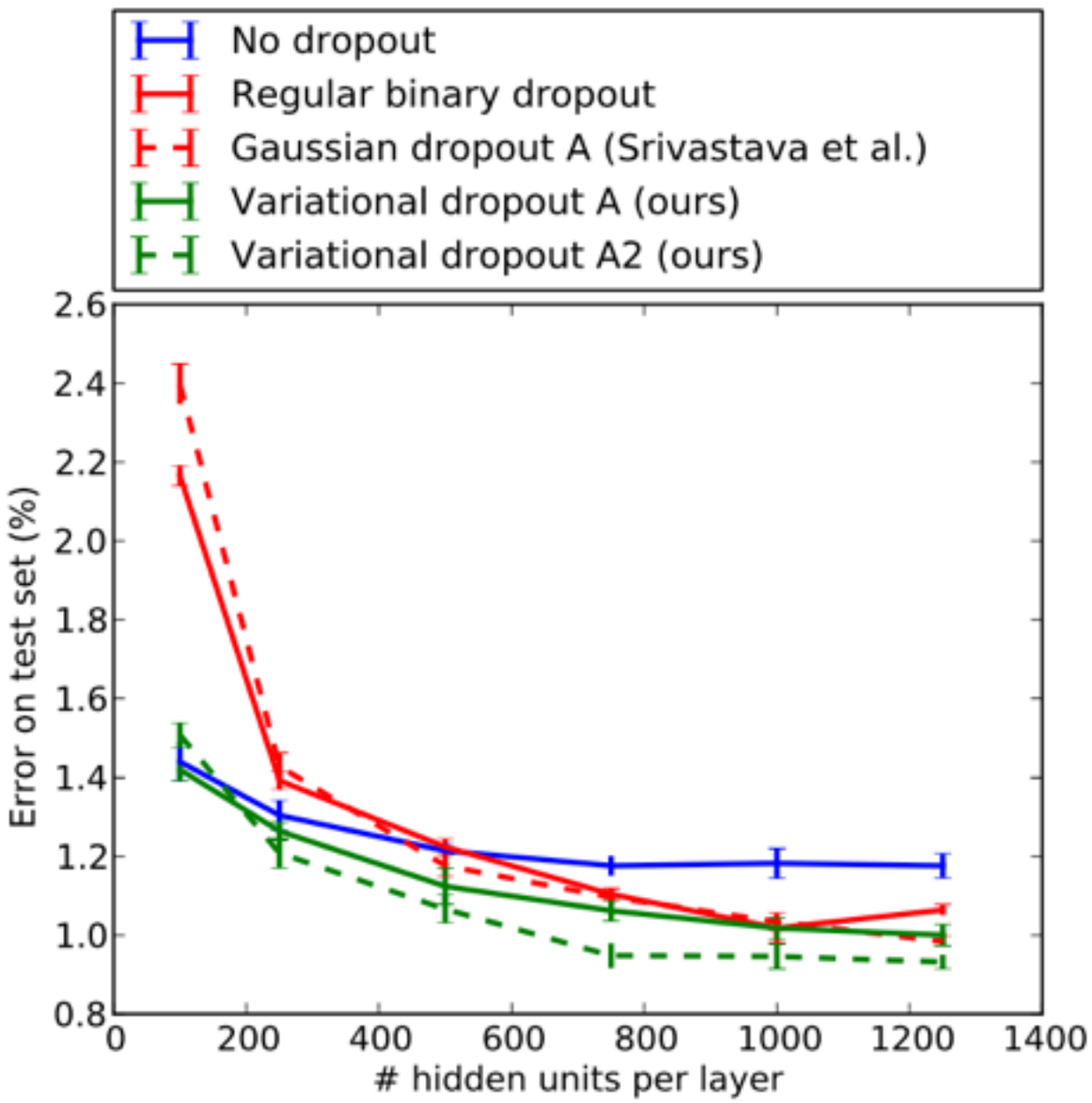
Each row of A and B is element of minibatch

Trivial minibatch parallism (GEMM, conv, …)

Learning Gaussian dropout rates

- Gaussian dropout is a special case of SGVB with scale-invariant prior $\mathrm{p}(\mathrm{w})$

- Variational interpretation allows optimizing the dropout rates



SGVB for global parameters

- Main idea 1:
  improve efficiency of Stochastic Gradient Variational Bayes (SGVB) through local parameterisation
  => reduced variance

- Main idea 2:
  treat Gaussian dropout as SGVB with local noise
  => optimize dropout parameters towards bound
  => better generalisation

“Variational Dropout and the Local Reparameterization Trick”
Diederik P. Kingma, Tim Salimans, Max Welling, NIPS 2015

Summary

- Stochastic Gradient Variational Inference (SGVI):
  “Swiss army knife” for inference
  - Works with almost any model $\mathrm{p}(\mathrm{x},\mathrm{z})$
  - Works with almost any approx. posterior $\mathrm{q}(\mathrm{z}|\mathrm{x})$
  - Just requires gradient ascent on single objective
- Applications:
  - variational auto-encoders
  - semi-supervised learning
  - optimizing MCMC hyper-parameters, dropout rates

Current project: **full-scale images, then video, audio**

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