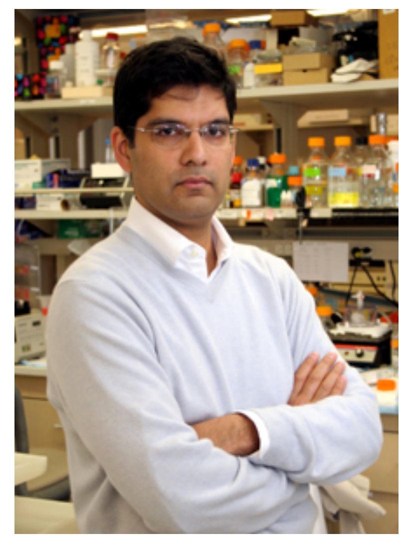
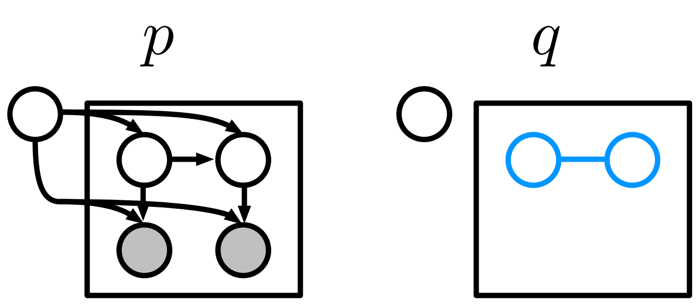
Learning representations for efficient inference

Matt Johnson, David Duvenaud, Alex Wiltschko, Bob Datta, Ryan Adams



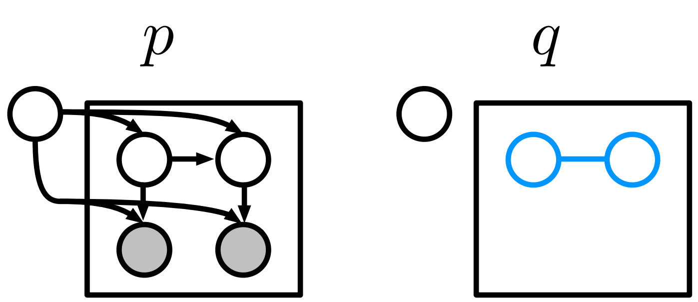
HARVARD INTELLIGENT PROBABILISTIC SYSTEMS

HARVARD MEDICAL SCHOOL



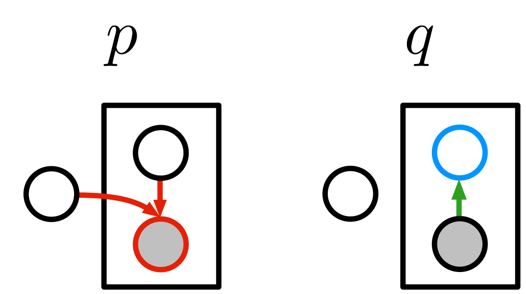
$$\color{blue}{q^*(x)} \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta)q(x) \,]$$

Natural gradient SVI
for nice PGMs



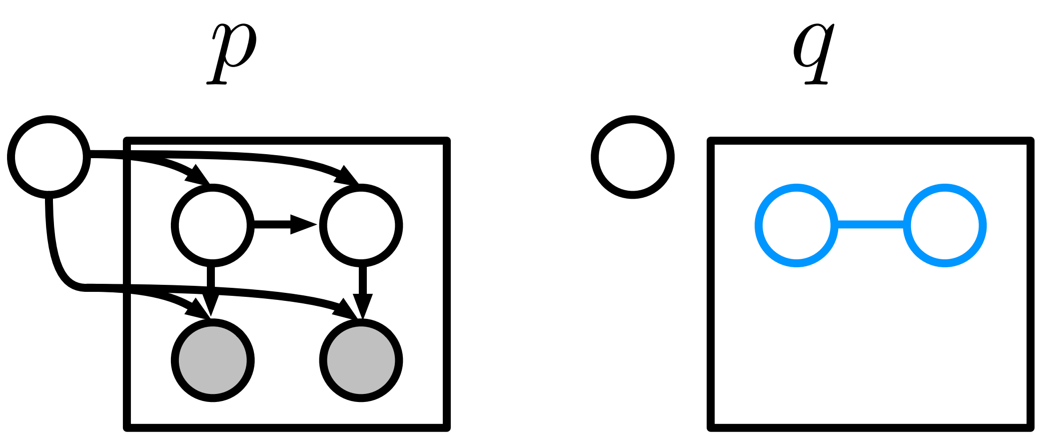
$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta)q(x) \,]$$

Natural gradient SVI
for nice PGMs



$$q^*(x) \triangleq \mathcal{N}(x \,|\, \mu(y;\phi), \Sigma(y;\phi))$$

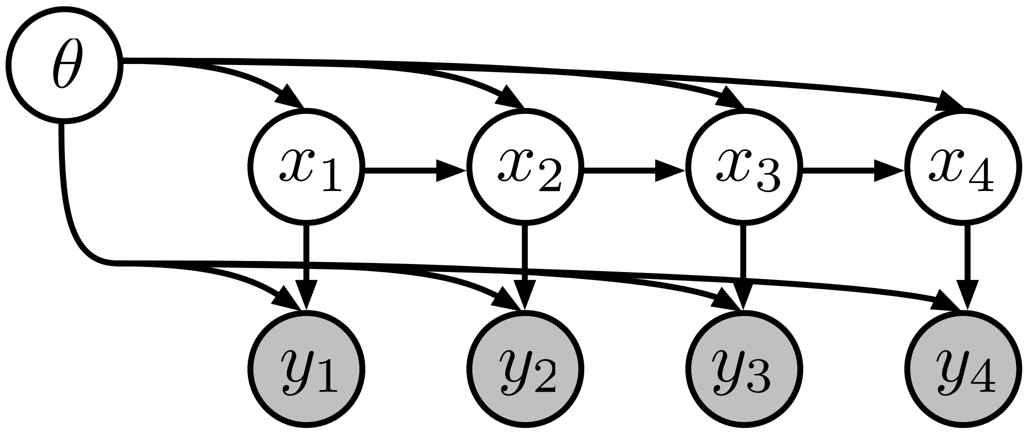
Variational autoencoders
and inference networks



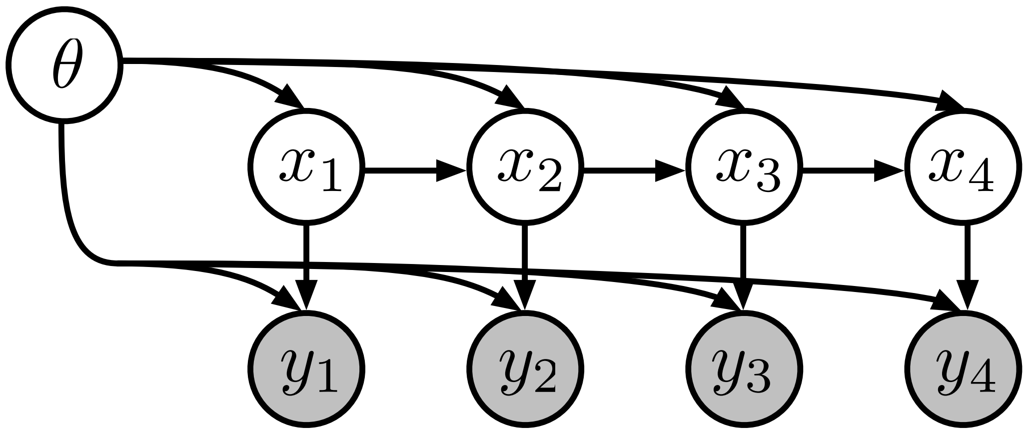
$$q^*(x) \triangleq \underset{q(x)}{\arg\max}\, \mathcal{L}[\, q(\theta)q(x)\,]$$

Natural gradient SVI
for nice PGMs [1,2]

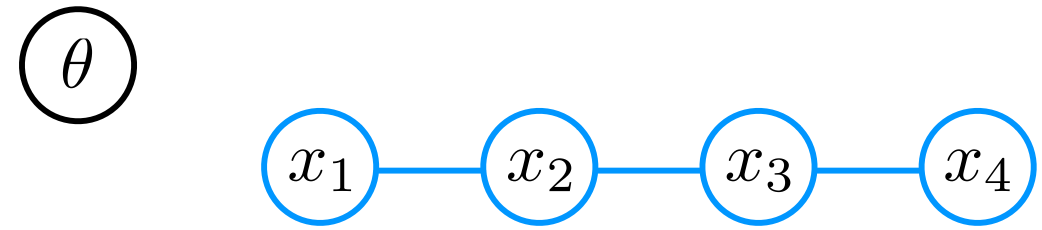
[1] Hoffman, Bach, Blei. Online learning for Latent Dirichlet Allocation. NIPS 2010.
[2] Hoffman, Blei, Wang, and Paisley. Stochastic variational inference. JMLR 2013.



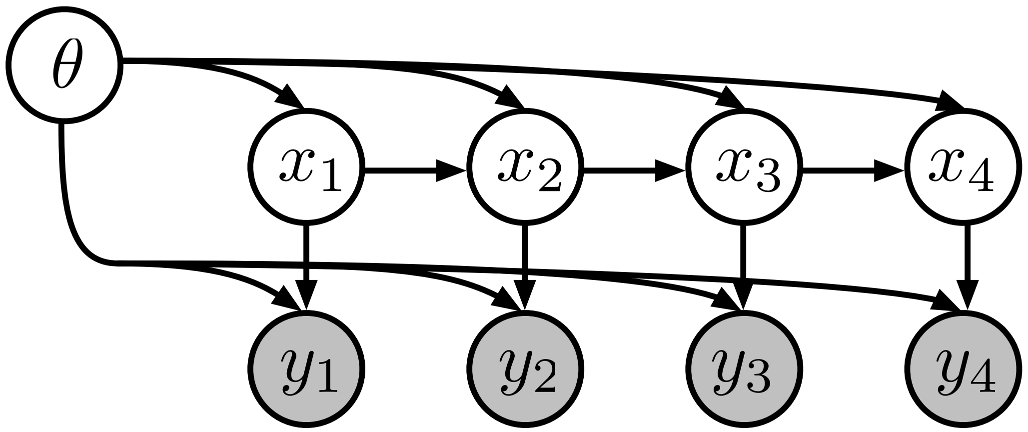
$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \theta)$ is a linear-Gaussian observation
$p(\theta)$ is a conjugate prior

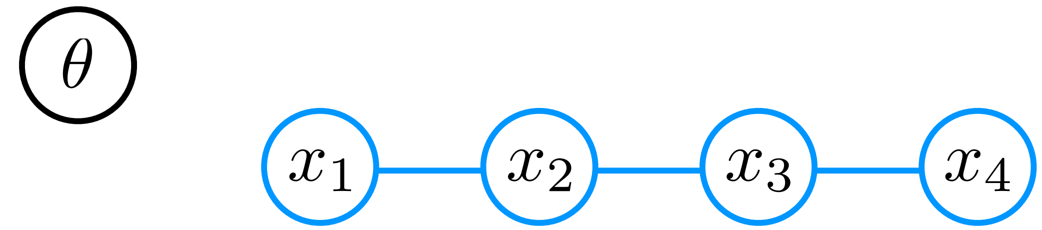


$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \theta)$ is a linear-Gaussian observation
$p(\theta)$ is a conjugate prior



$$q(\theta)q(x) \approx p(\theta, x \mid y)$$

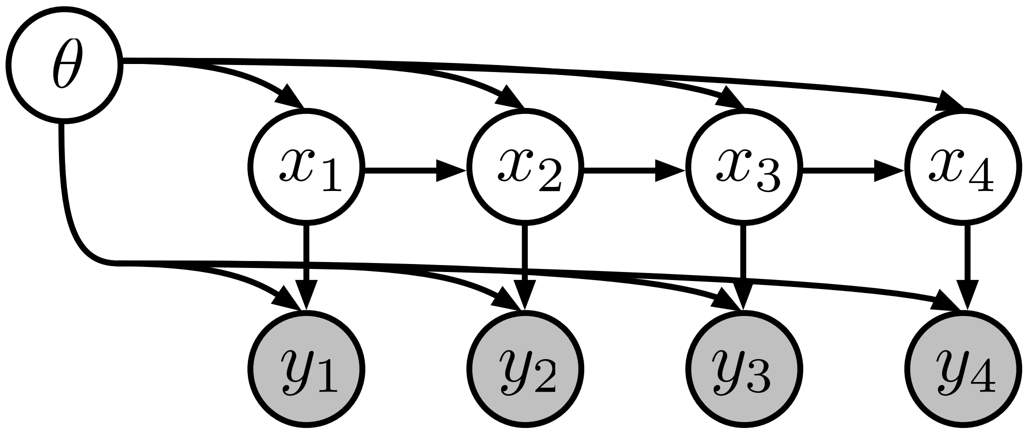


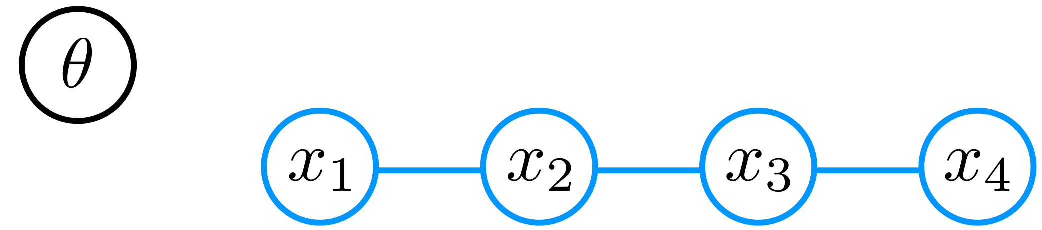


$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \theta)$ is a linear-Gaussian observation
$p(\theta)$ is a conjugate prior

$$q(\theta)q(x) \approx p(\theta, x \mid y)$$

$$\mathcal{L}(\eta_\theta, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(x)}\left[\log \frac{p(\theta, x, y)}{q(\theta)q(x)}\right]$$



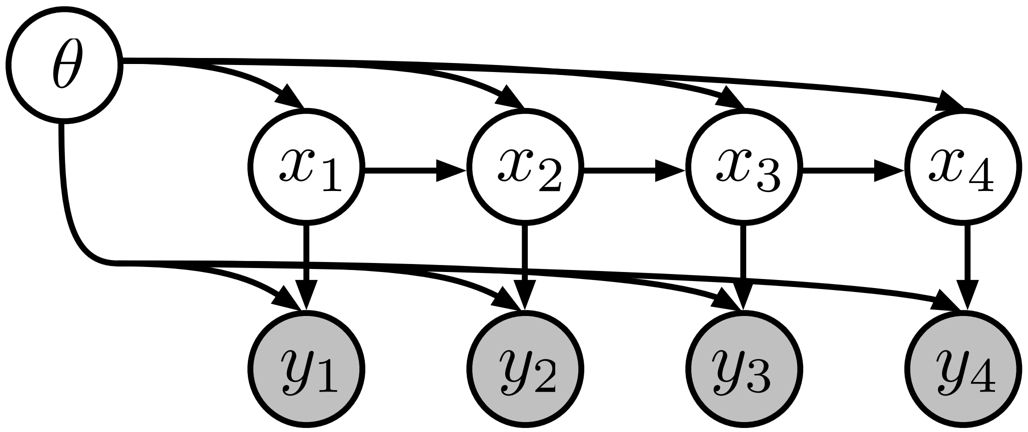


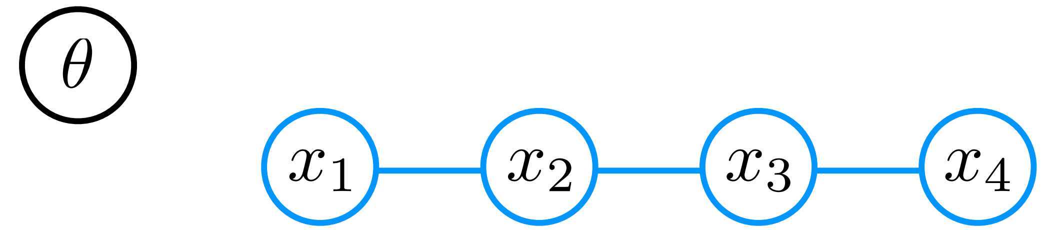
$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \theta)$ is a linear-Gaussian observation
$p(\theta)$ is a conjugate prior

$$q(\theta)q(x) \approx p(\theta, x \mid y)$$

$$\mathcal{L}(\eta_\theta, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(x)}\left[\log \frac{p(\theta, x, y)}{q(\theta)q(x)}\right]$$

$$\eta_x^*(\eta_\theta) \triangleq \arg\max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_x) \qquad \mathcal{L}_{\text{SVI}}(\eta_\theta) \triangleq \mathcal{L}(\eta_\theta, \eta_x^*(\eta_\theta))$$





$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \theta)$ is a linear-Gaussian observation
$p(\theta)$ is a conjugate prior

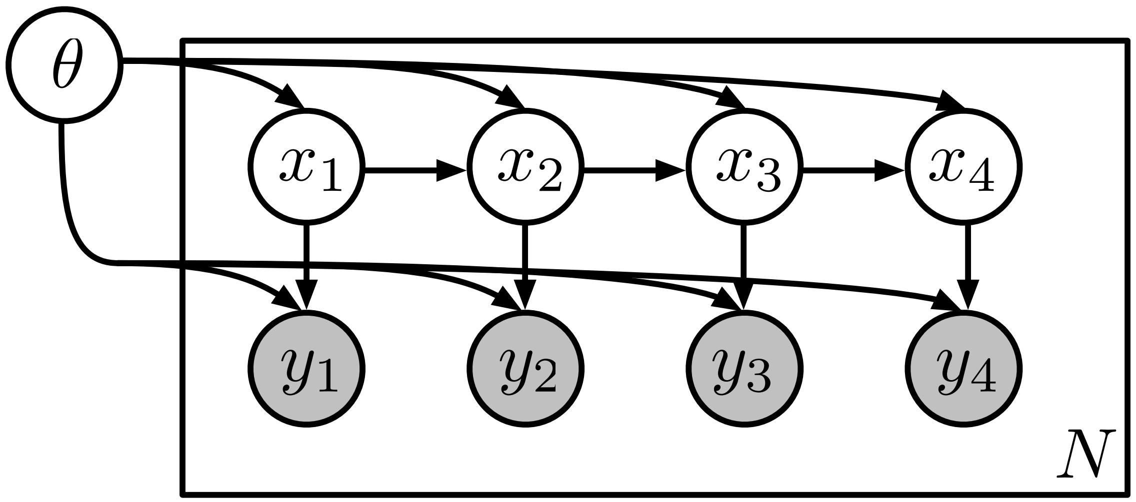
$$q(\theta)q(x) \approx p(\theta, x \mid y)$$

$$\mathcal{L}(\eta_\theta, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(x)}\left[\log \frac{p(\theta, x, y)}{q(\theta)q(x)}\right]$$

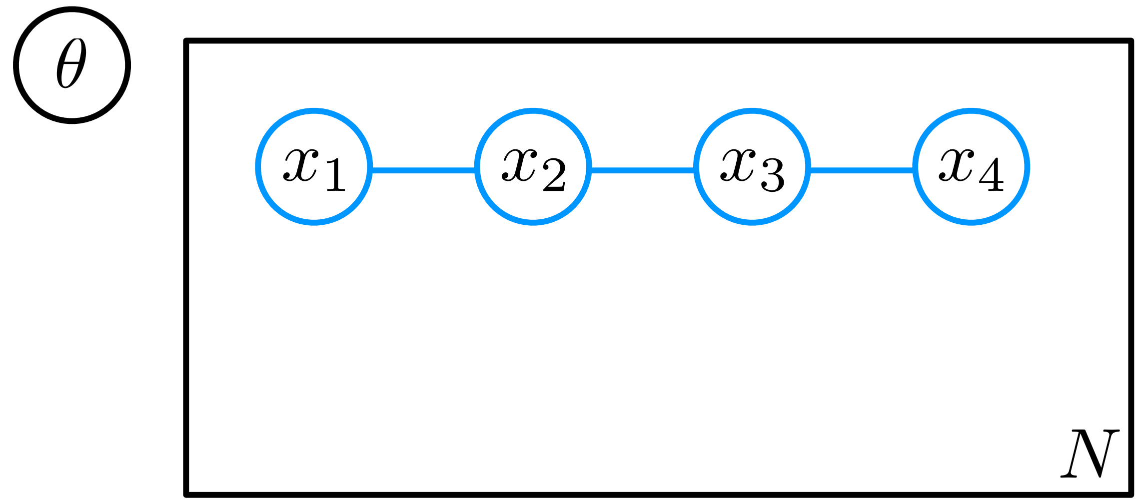
$$\eta_x^*(\eta_\theta) \triangleq \arg\max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_x) \qquad \mathcal{L}_{\text{SVI}}(\eta_\theta) \triangleq \mathcal{L}(\eta_\theta, \eta_x^*(\eta_\theta))$$

Proposition (natural gradient SVI of Hoffman et al. 2013)

$$\widetilde{\nabla}\mathcal{L}_{\text{SVI}}(\eta_\theta) = \eta_\theta^0 + \mathbb{E}_{q^*(x)}(t_{xy}(x, y), 1) - \eta_\theta$$



$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \theta)$ is a linear-Gaussian observation
$p(\theta)$ is a conjugate prior



$$q(\theta)q(x) \approx p(\theta, x \mid y)$$

$$\mathcal{L}(\eta_\theta, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(x)}\left[\log \frac{p(\theta,x,y)}{q(\theta)q(x)}\right]$$

$$\eta_x^*(\eta_\theta) \triangleq \arg\max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_x) \qquad \mathcal{L}_{\text{SVI}}(\eta_\theta) \triangleq \mathcal{L}(\eta_\theta, \eta_x^*(\eta_\theta))$$

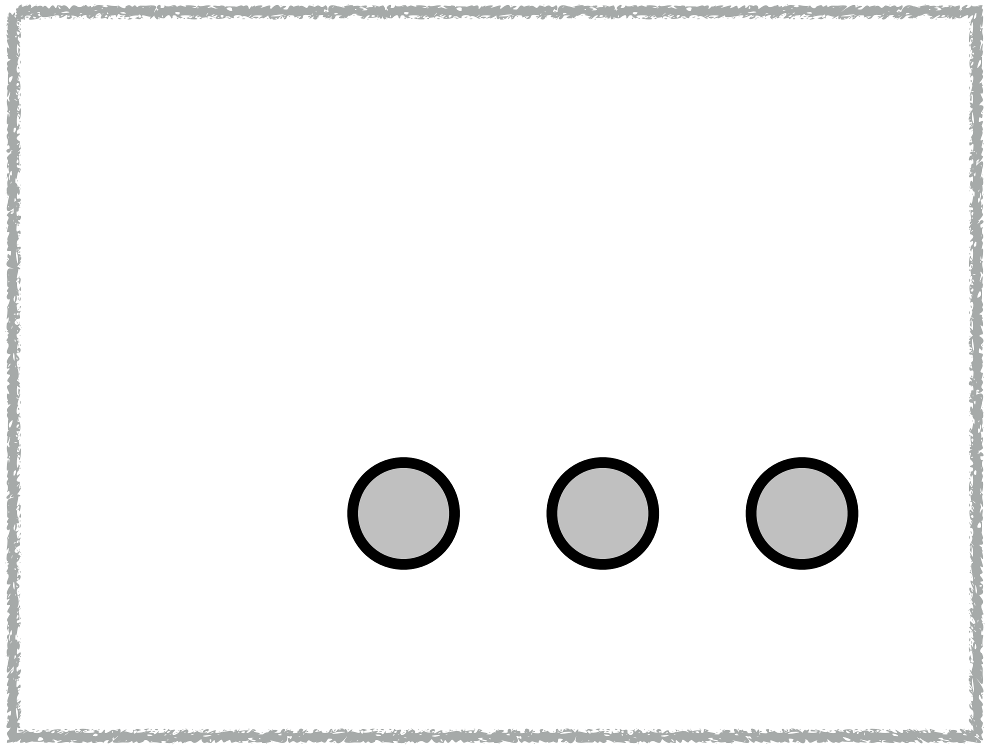
Proposition (natural gradient SVI of Hoffman et al. 2013)

$$\widetilde{\nabla}\mathcal{L}_{\text{SVI}}(\eta_\theta) = \eta_\theta^0 + \sum_{n=1}^{N} \mathbb{E}_{q^*(x_n)}(t_{xy}(x_n, y_n), 1) - \eta_\theta$$

Step 1: compute evidence potentials

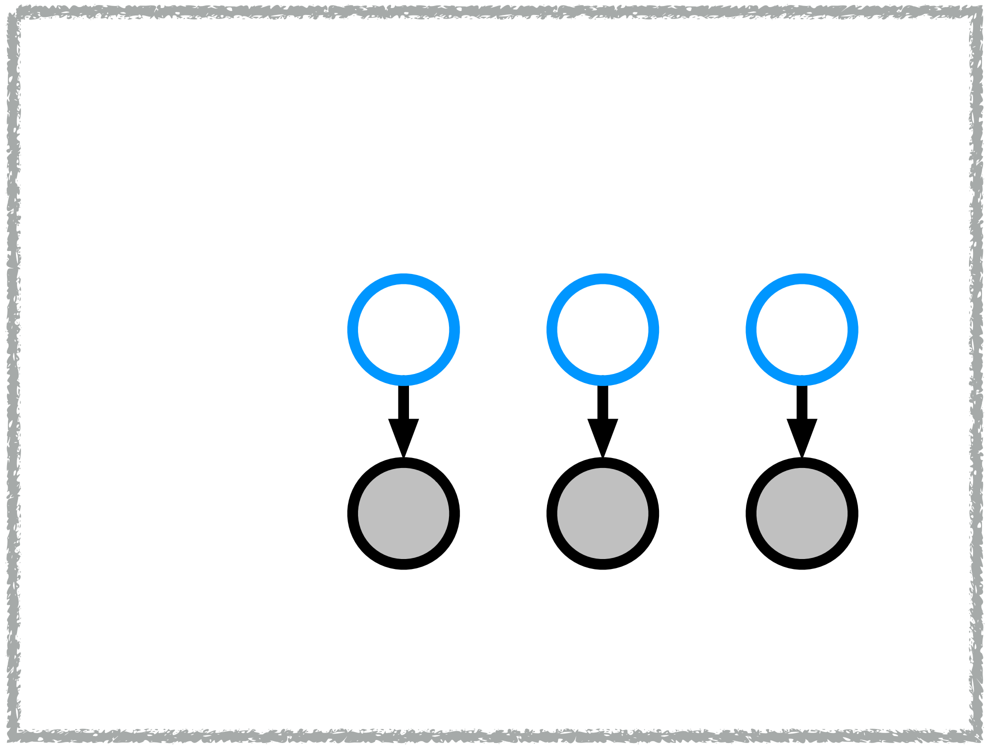
[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

Step 1: compute evidence potentials



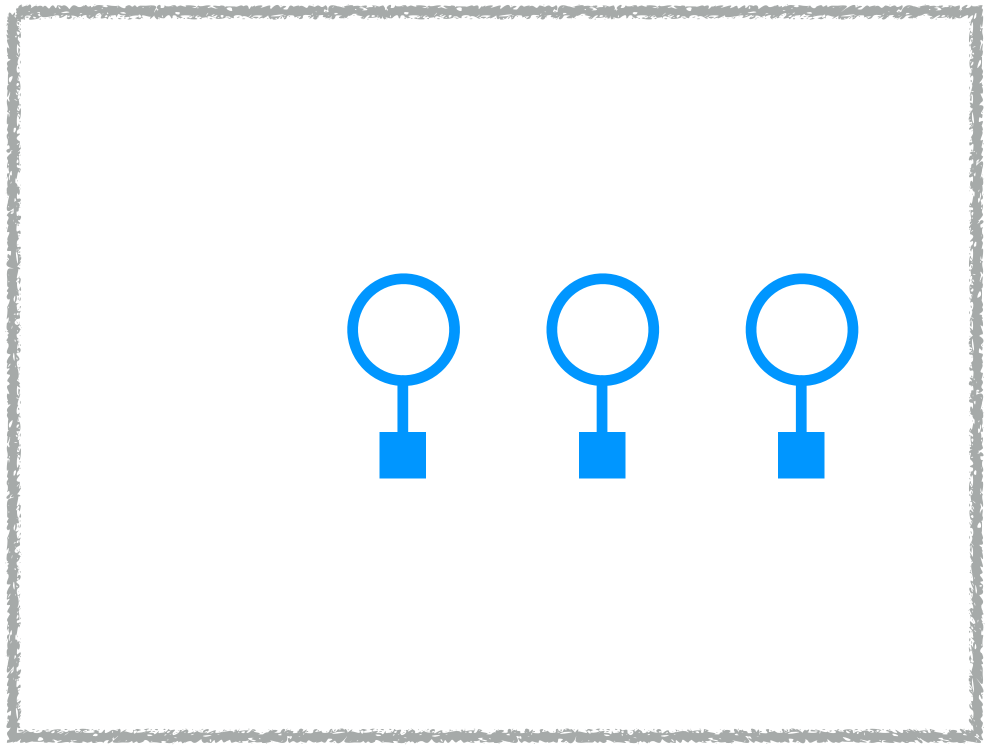
[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

Step 1: compute evidence potentials



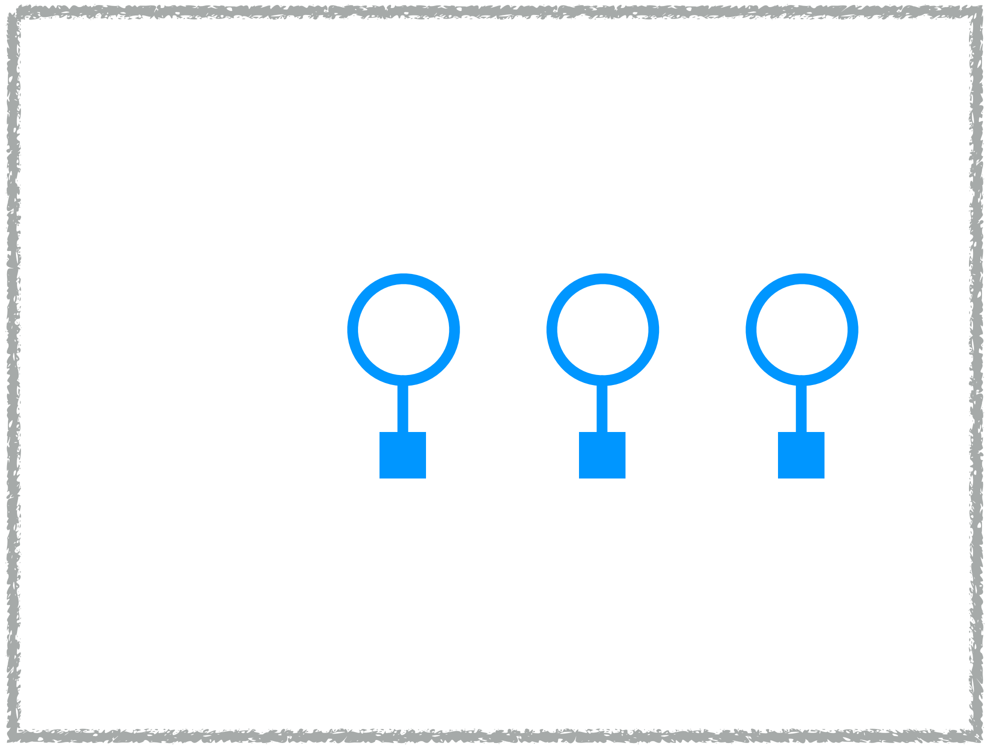
[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

Step 1: compute evidence potentials



[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

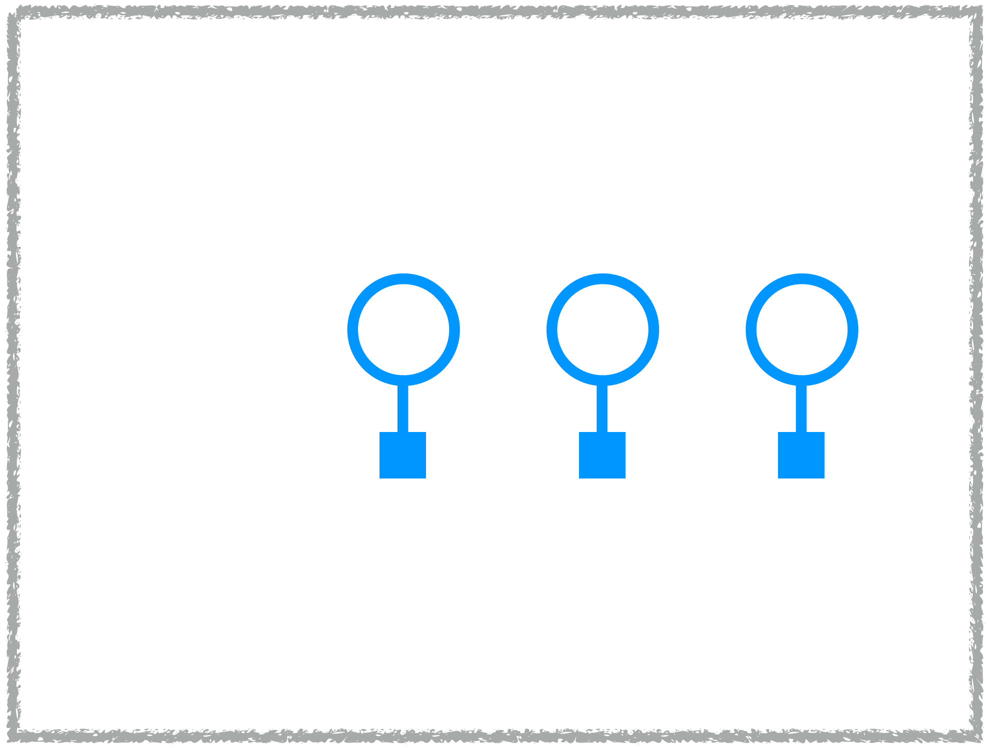
Step 1: compute evidence potentials



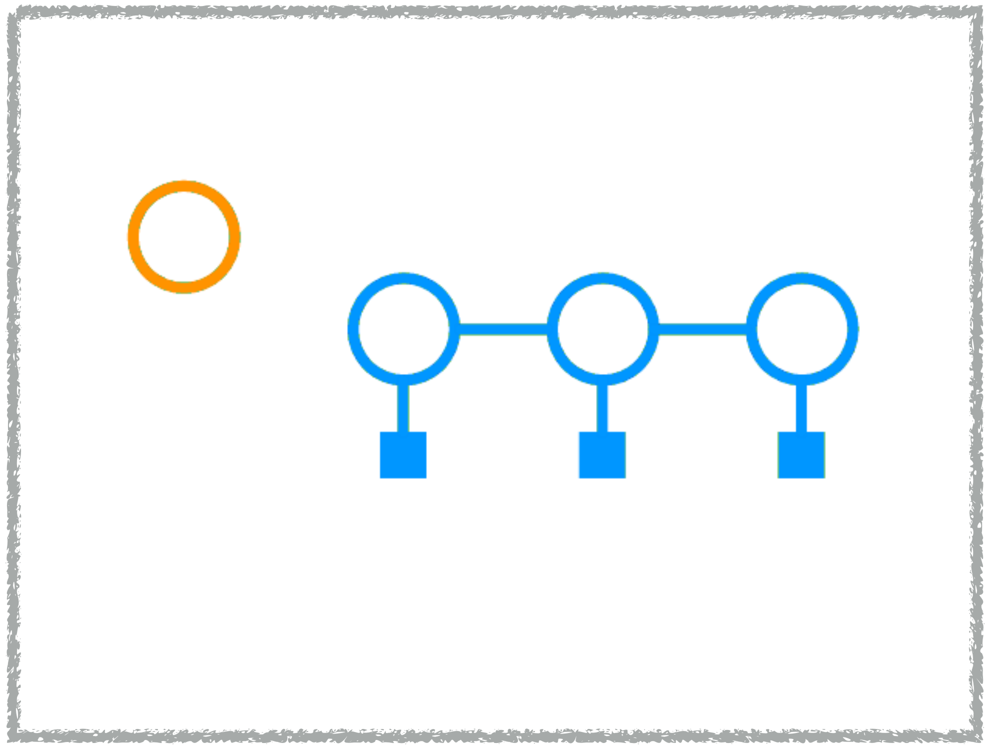
Step 2: run fast message passing

[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

Step 1: compute evidence potentials

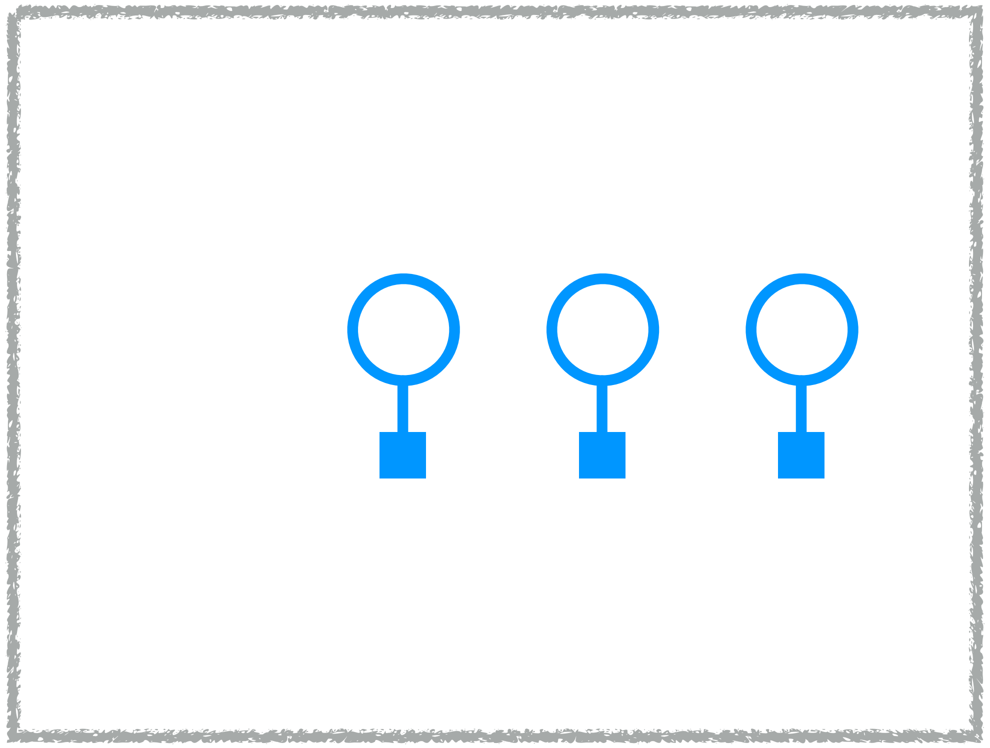


Step 2: run fast message passing

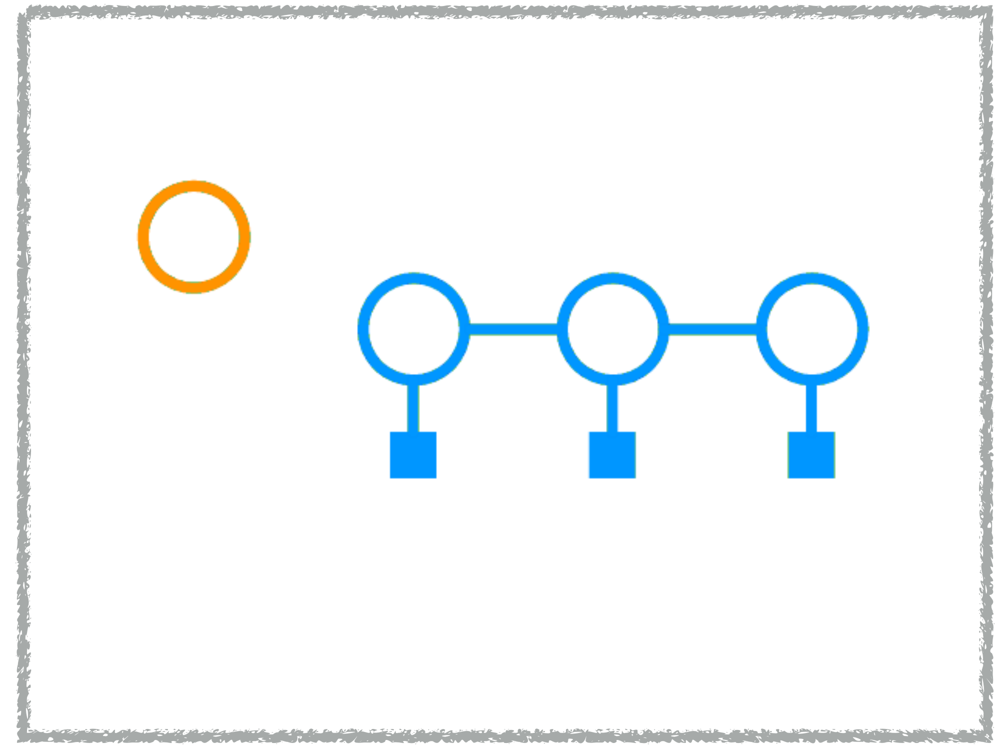


[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

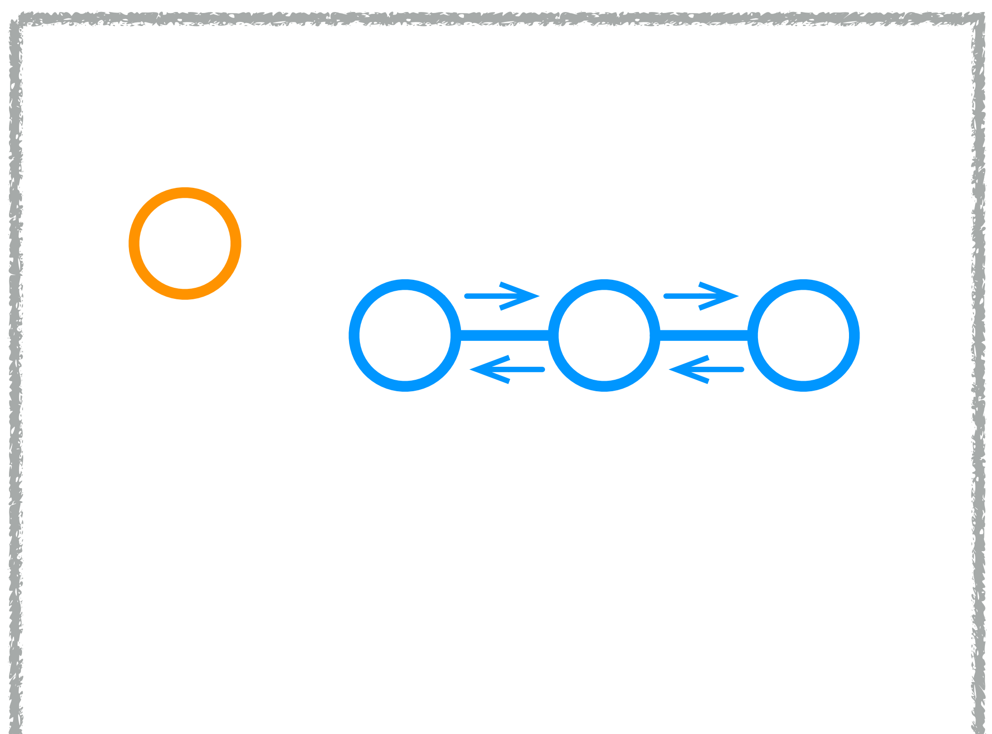
Step 1: compute evidence potentials



Step 2: run fast message passing

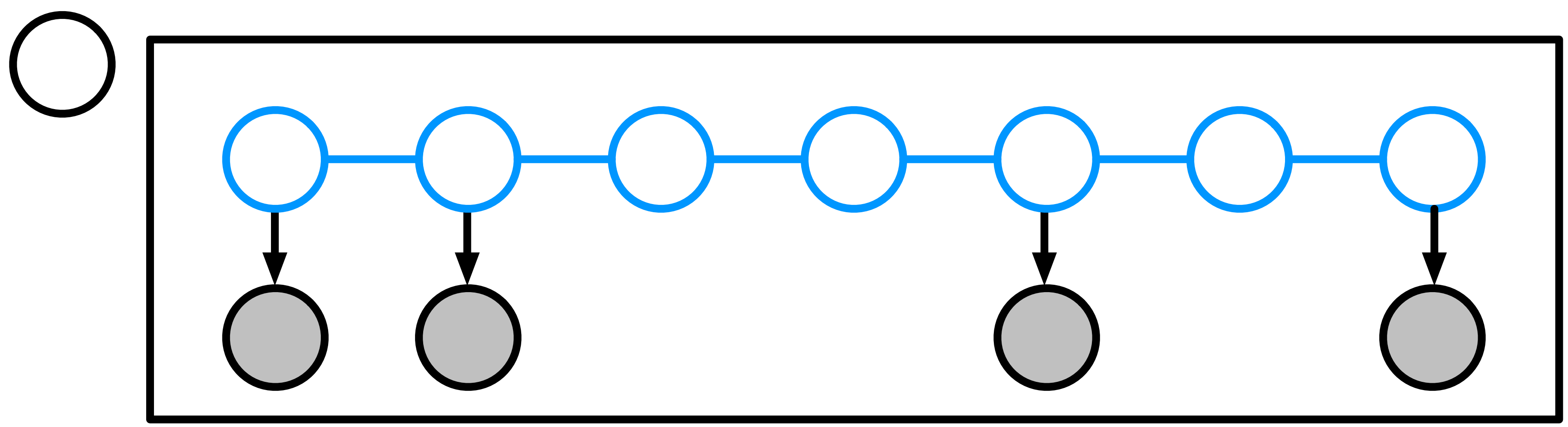


Step 3: compute natural gradient

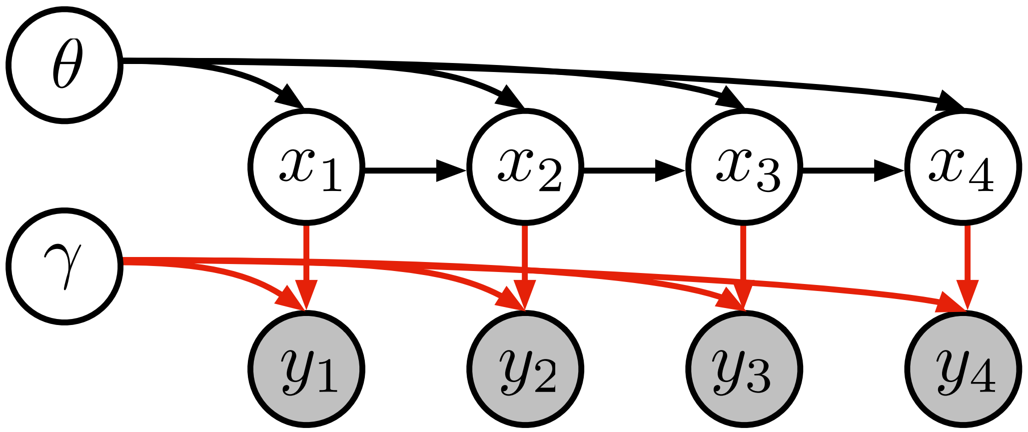


[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

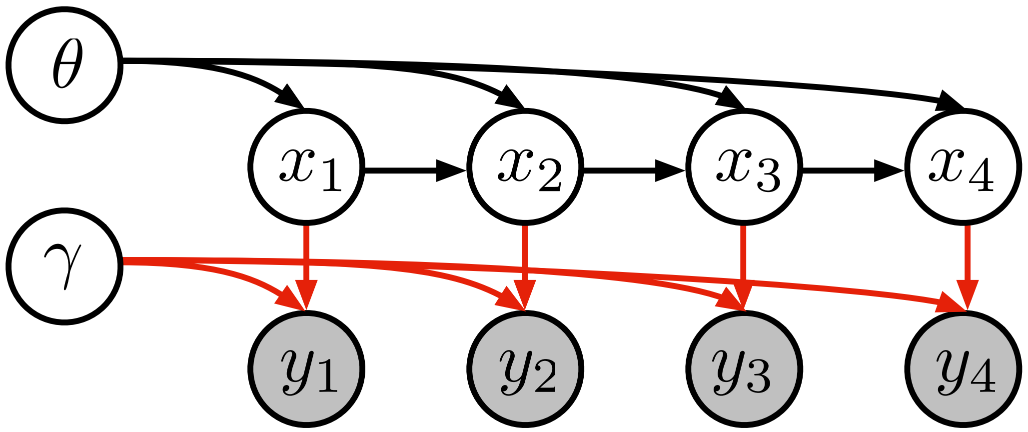
arbitrary inference queries

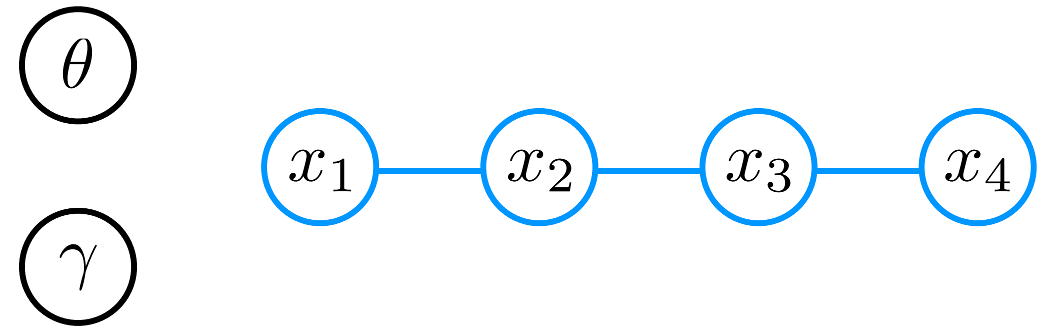


What about more general observation models?



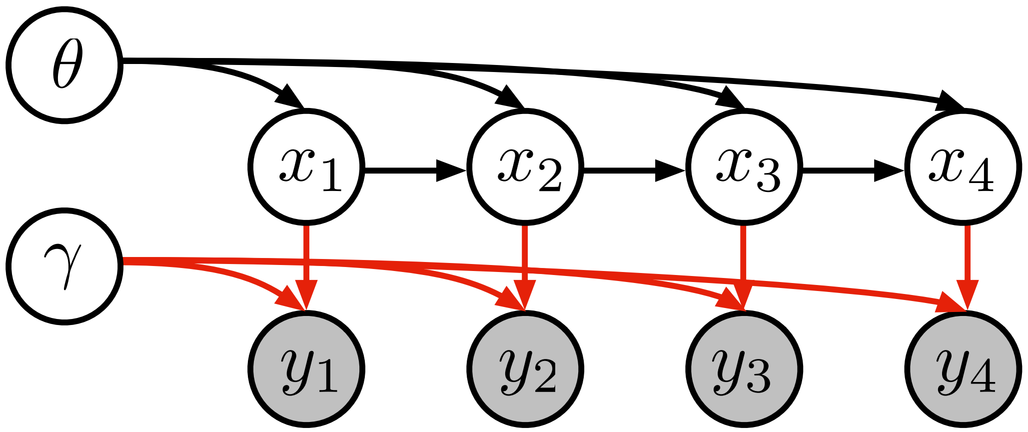
$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \gamma)$ is a neural network decoder
$p(\theta)$ is a conjugate prior, $p(\gamma)$ is generic

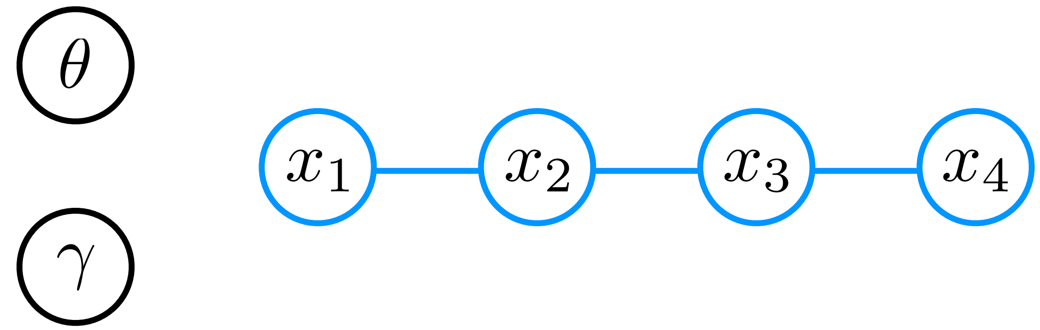




$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \gamma)$ is a neural network decoder
$p(\theta)$ is a conjugate prior, $p(\gamma)$ is generic

$$q(\theta)q(\gamma)q(x) \approx p(\theta, \gamma, x \mid y)$$





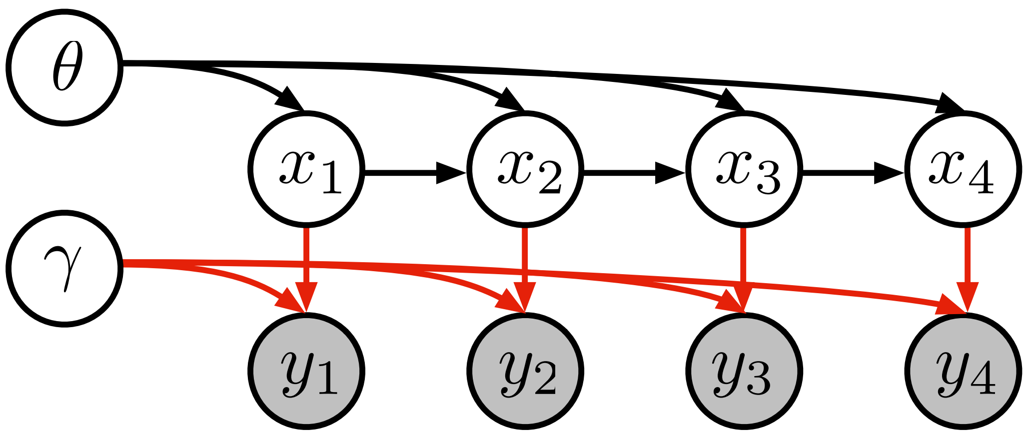
$p(x \mid \theta)$ is a linear dynamical system

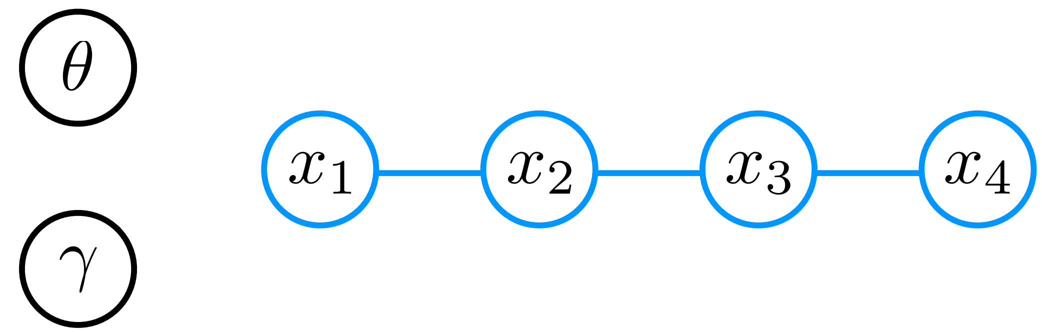
$p(y \mid x, \gamma)$ is a neural network decoder

$p(\theta)$ is a conjugate prior, $p(\gamma)$ is generic

$$q(\theta)q(\gamma)q(x) \approx p(\theta, \gamma, x \mid y)$$

$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\theta)q(x)}\left[\log \frac{p(\theta, \gamma, x)p(y \mid x, \gamma)}{q(\theta)q(\gamma)q(x)}\right]$$





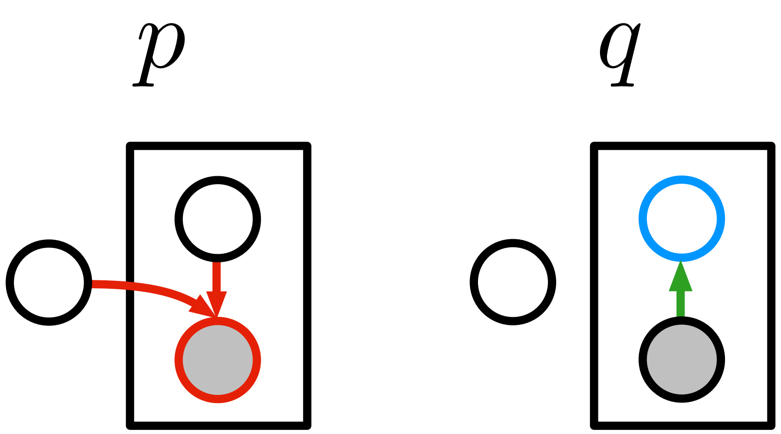
$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \gamma)$ is a neural network decoder
$p(\theta)$ is a conjugate prior, $p(\gamma)$ is generic

$$q(\theta)q(\gamma)q(x) \approx p(\theta, \gamma, x \mid y)$$

$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\theta)q(x)}\left[\log \frac{p(\theta, \gamma, x)p(y \mid x, \gamma)}{q(\theta)q(\gamma)q(x)}\right]$$

$$\eta_x^\star(\eta_\theta, \eta_\gamma) \triangleq \arg\max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x)$$

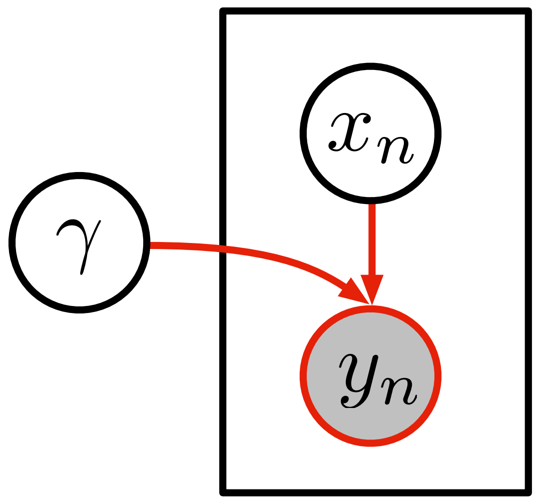
$$\mathcal{L}_{\mathrm{SVI}}(\eta_\theta, \eta_\gamma) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^\star(\eta_\theta, \eta_\gamma))$$



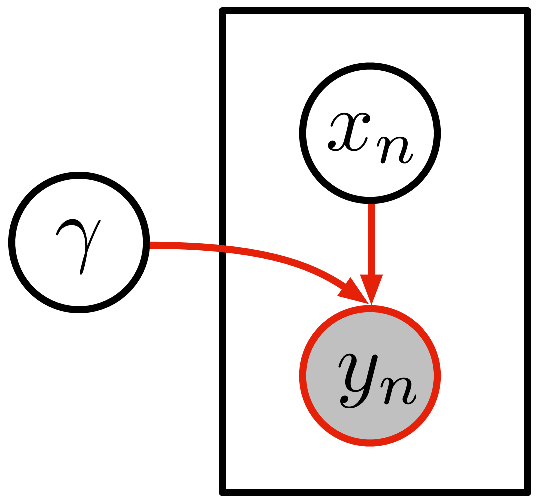
$$q^*(x) \triangleq \mathcal{N}(x \mid \mu(y; \phi), \Sigma(y; \phi))$$

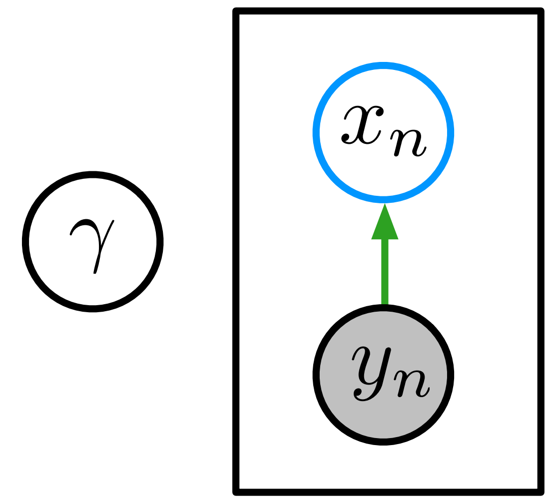
Variational autoencoders and inference networks [1,2]

[1] Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014.
[2] Rezende, Mohamed, and Wierstra. Stochastic backpropagation and approximate inference in deep generative models. ICML 2014

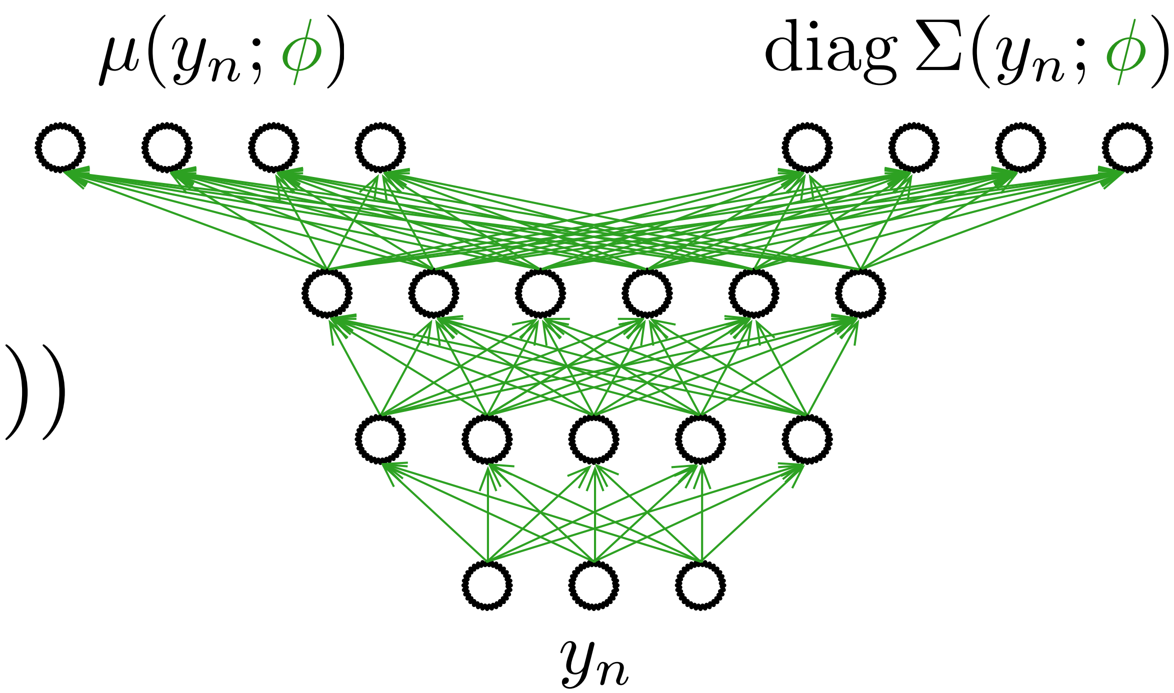

x_n
γ
y_n

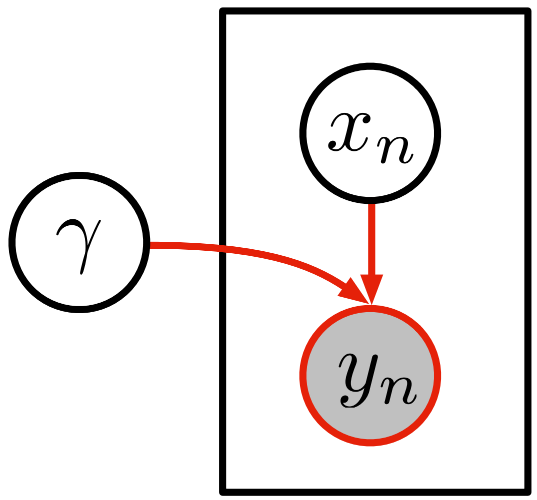
$$q^{\star}(x_n) \triangleq \mathcal{N}(x_n \,|\, \mu(y_n; \phi),\, \Sigma(y_n; \phi))$$

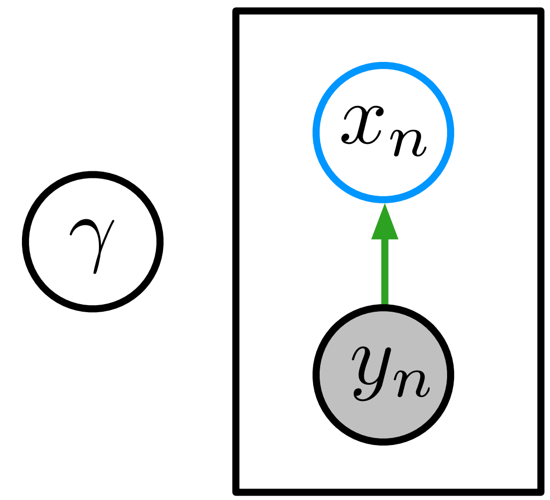


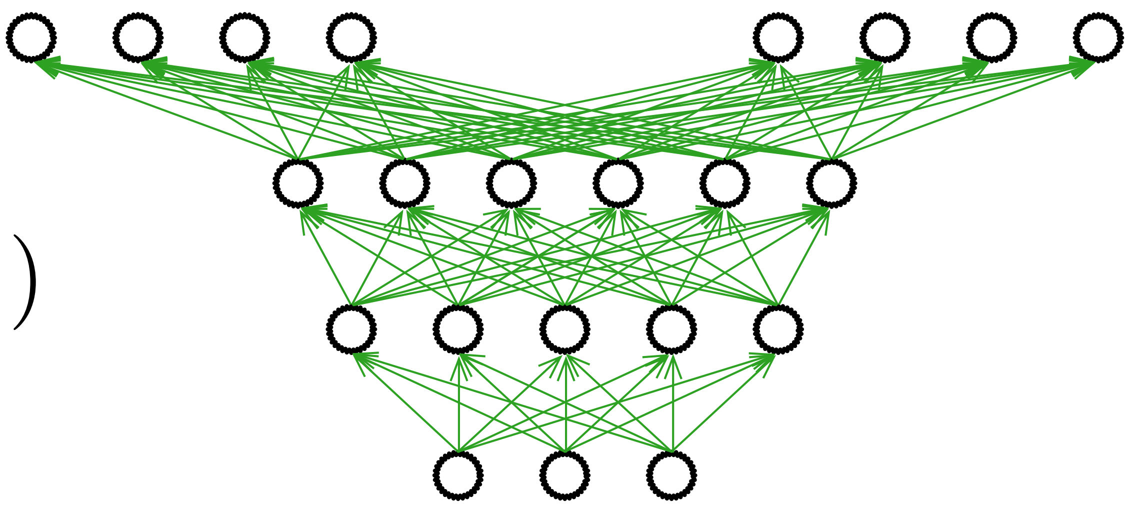


$$q^{\star}(x_n) \triangleq \mathcal{N}(x_n \mid \mu(y_n;\phi),\ \Sigma(y_n;\phi))$$









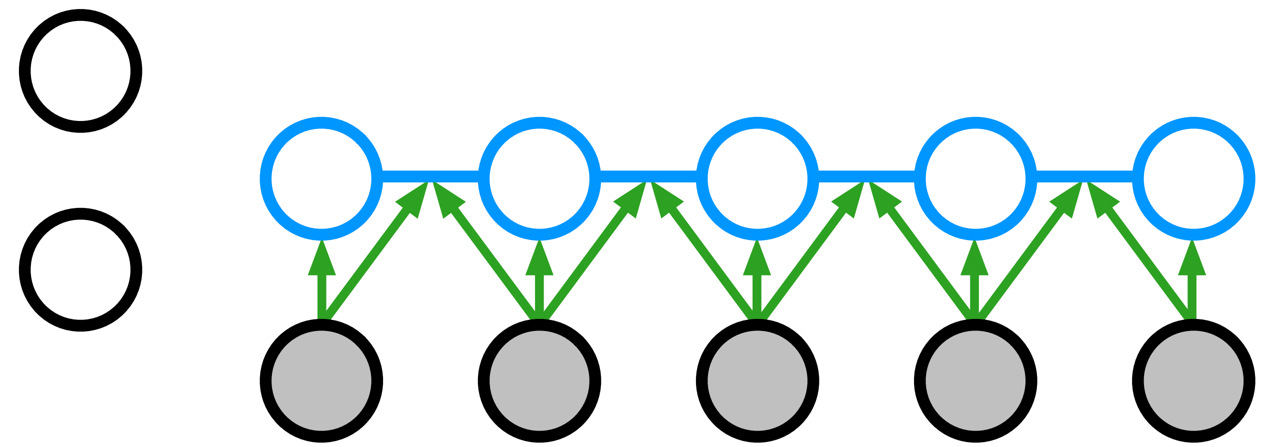
$$q^\star(x_n) \triangleq \mathcal{N}(x_n \mid \mu(y_n; \phi), \, \Sigma(y_n; \phi))$$

$$\mathcal{L}_{\text{VAE}}(\eta_\gamma, \phi) \triangleq \mathcal{L}(\eta_\gamma, \eta_x^\star(\phi))$$

$$\mu_t(y_t; \phi_\mu)$$

[1,2] $$J_{t,t}(y_t; \phi_D)$$

$$J_{t,t+1}(y_t, y_{t+1}; \phi_B)$$



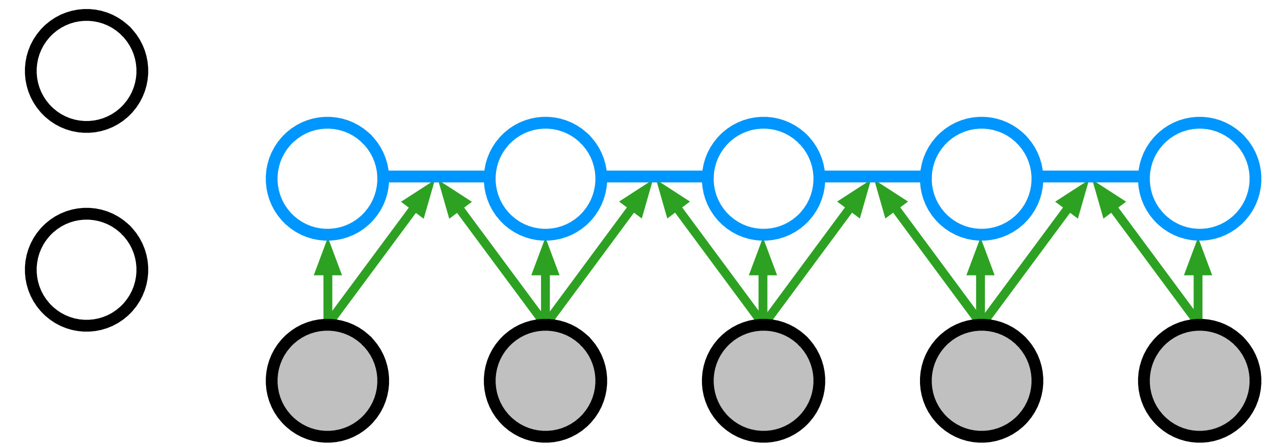
[1] Archer, Park, Buesing, Cunningham, Paninski. Black box variational inference for state space models. ICLR 2016 Workshops.
[2] Gao*, Archer*, Paninski, Cunningham. Linear dynamical neural population models through nonlinear embeddings. NIPS 2016.

[1,2]

$$\mu_t(y_t; \phi_\mu)$$

$$J_{t,t}(y_t; \phi_D)$$

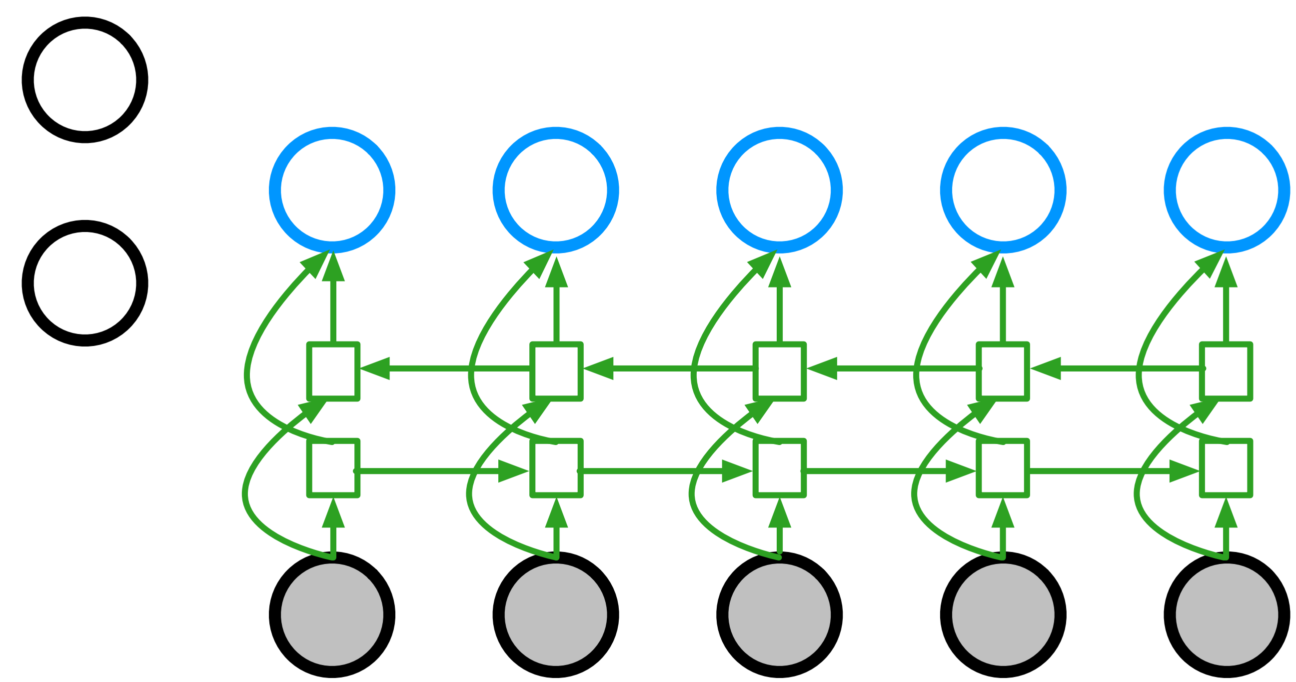
$$J_{t,t+1}(y_t, y_{t+1}; \phi_B)$$



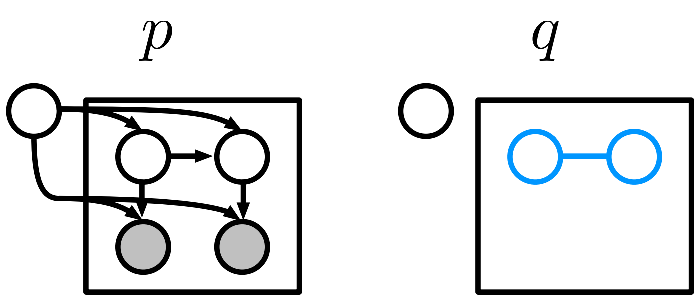
[3]

$$\mu_t(y_{1:T}, \hat{x}_{t-1}; \phi)$$

$$\Sigma_t(y_{1:T}, \hat{x}_{t-1}; \phi)$$

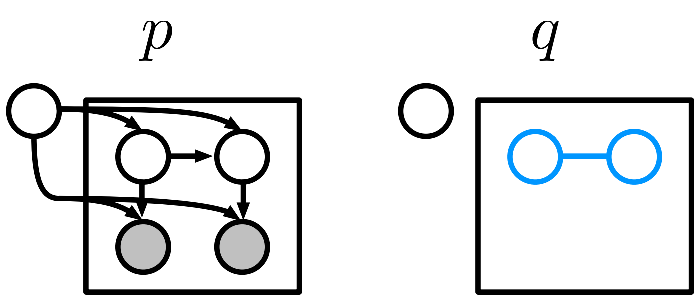


[1] Archer, Park, Buesing, Cunningham, Paninski. Black box variational inference for state space models. ICLR 2016 Workshops.
[2] Gao*, Archer*, Paninski, Cunningham. Linear dynamical neural population models through nonlinear embeddings. NIPS 2016.
[3] Krishnan, Shalit, Sontag. Structured inference networks for nonlinear state space models. AISTATS 2017.



$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta) q(x) \,]$$

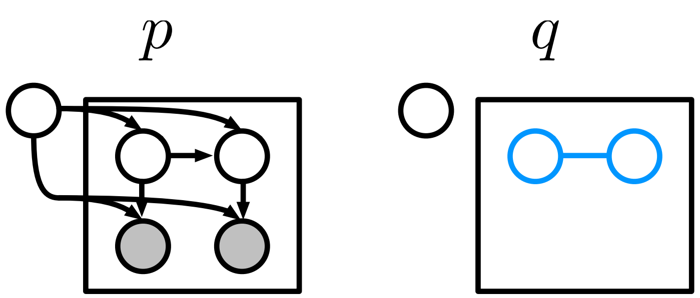
Natural gradient SVI



$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta)q(x)\,]$$

Natural gradient SVI

– expensive for general obs.

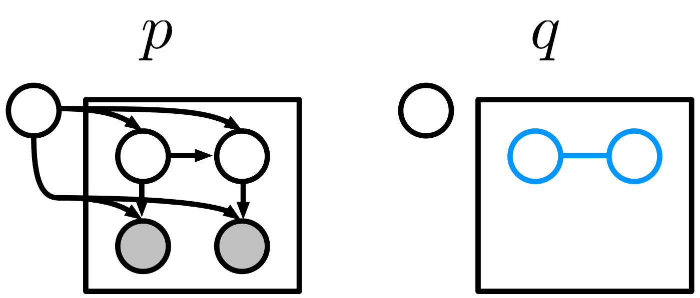


$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta)q(x) \,]$$

Natural gradient SVI

– expensive for general obs.

+ optimal local factor



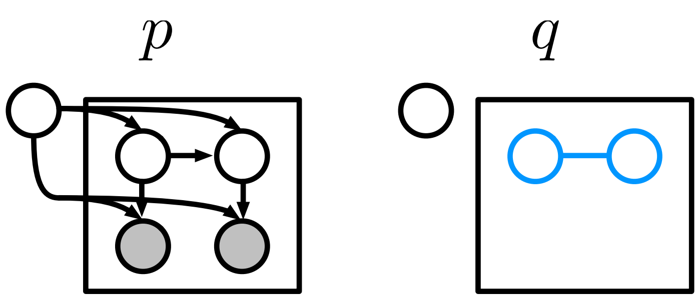
$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta)q(x) \,]$$

Natural gradient SVI

– expensive for general obs.

+ optimal local factor

+ exploits conj. graph structure



$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta)q(x) \,]$$

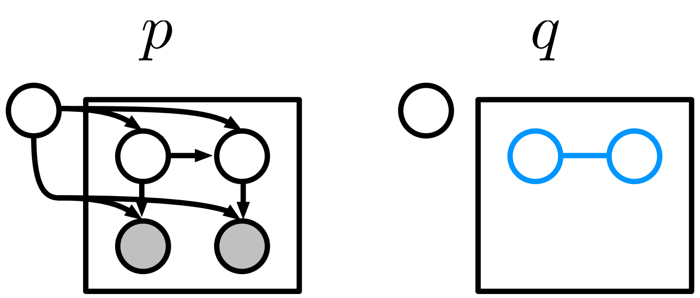
Natural gradient SVI

− expensive for general obs.

+ optimal local factor

+ exploits conj. graph structure

+ arbitrary inference queries



$$\color{blue}{q^*(x)} \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta)q(x) \,]$$

Natural gradient SVI

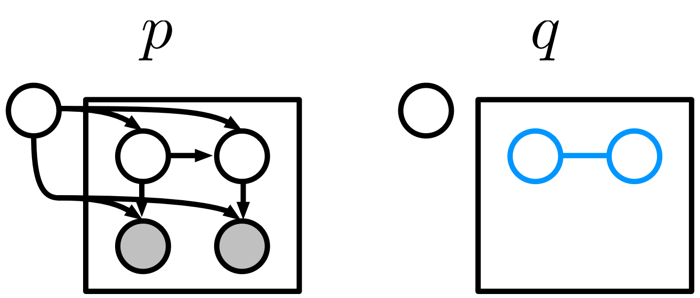
− expensive for general obs.

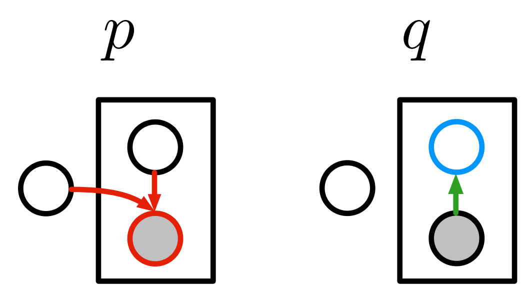
\+ optimal local factor

\+ exploits conj. graph structure

\+ arbitrary inference queries

\+ natural gradients





$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta)q(x)\,]$$

$$q^*(x) \triangleq \mathcal{N}(x \mid \mu(y;\phi), \Sigma(y;\phi))$$

Natural gradient SVI

Variational autoencoders

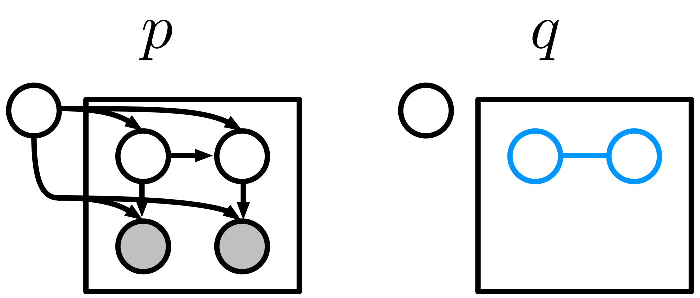
– expensive for general obs.

+ optimal local factor

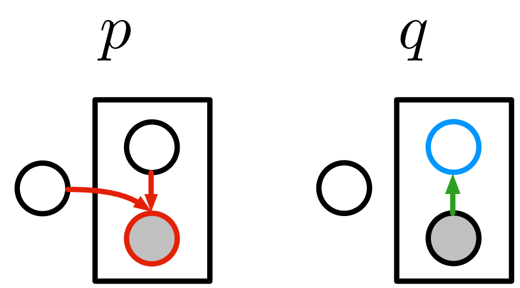
+ exploits conj. graph structure

+ arbitrary inference queries

+ natural gradients



$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta)q(x)\,]$$



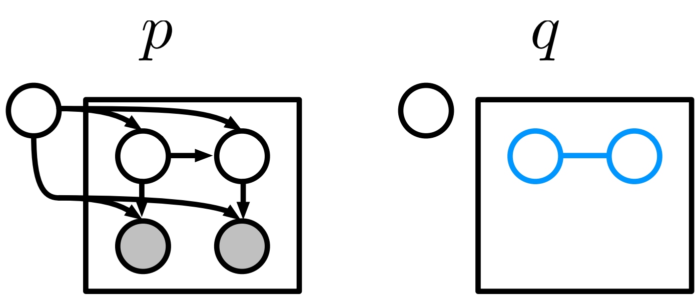
$$q^*(x) \triangleq \mathcal{N}(x \,|\, \mu(y;\phi), \Sigma(y;\phi))$$

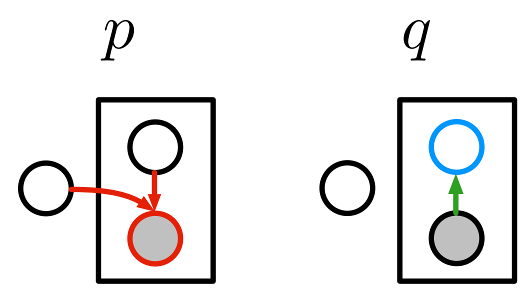
Natural gradient SVI

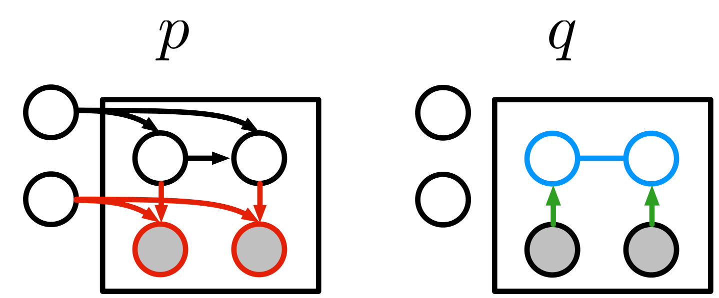
- – expensive for general obs.
- + optimal local factor
- + exploits conj. graph structure
- + arbitrary inference queries
- + natural gradients

Variational autoencoders

- + fast for general obs.
- – suboptimal local factor
- – ϕ does all local inference
- – limited inference queries
- – no natural gradients







$q^*(x) \triangleq \underset{q(x)}{\arg\max}\, \mathcal{L}[\, q(\theta)q(x)\,]$

$q^*(x) \triangleq \mathcal{N}(x \,|\, \mu(y;\phi), \Sigma(y;\phi))$

$q^*(x) \triangleq \; ?$

Natural gradient SVI

– expensive for general obs.

+ optimal local factor

+ exploits conj. graph structure

+ arbitrary inference queries

+ natural gradients

Variational autoencoders

+ fast for general obs.

– suboptimal local factor

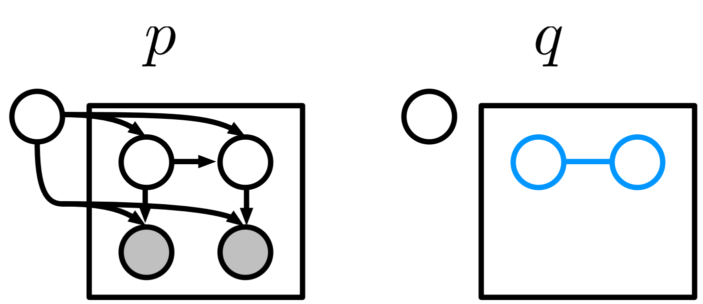
– ϕ does all local inference

– limited inference queries

– no natural gradients

Structured VAEs [1]

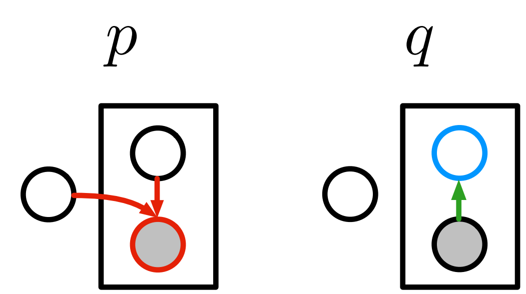
[1] **Johnson**, Duvenaud, Wiltschko, Datta, and Adams. Composing graphical models and neural networks. NIPS 2016.



$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta) q(x) \,]$$

Natural gradient SVI

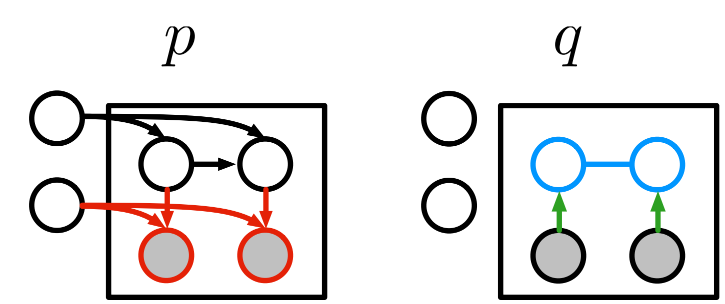
- – expensive for general obs.
- + optimal local factor
- + exploits conj. graph structure
- + arbitrary inference queries
- + natural gradients



$$q^*(x) \triangleq \mathcal{N}(x \,|\, \mu(y;\phi), \Sigma(y;\phi))$$

Variational autoencoders

- + fast for general obs.
- – suboptimal local factor
- – ϕ does all local inference
- – limited inference queries
- – no natural gradients



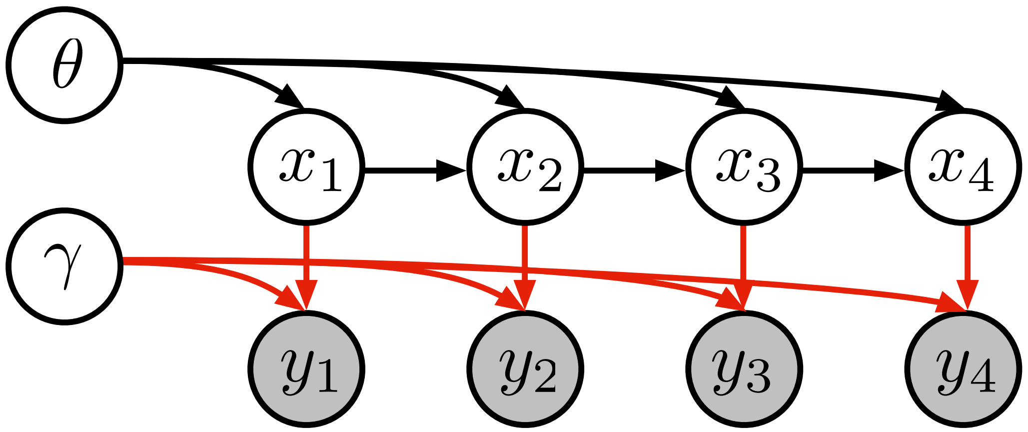
$$q^*(x) \triangleq \; ?$$

Structured VAEs [1]

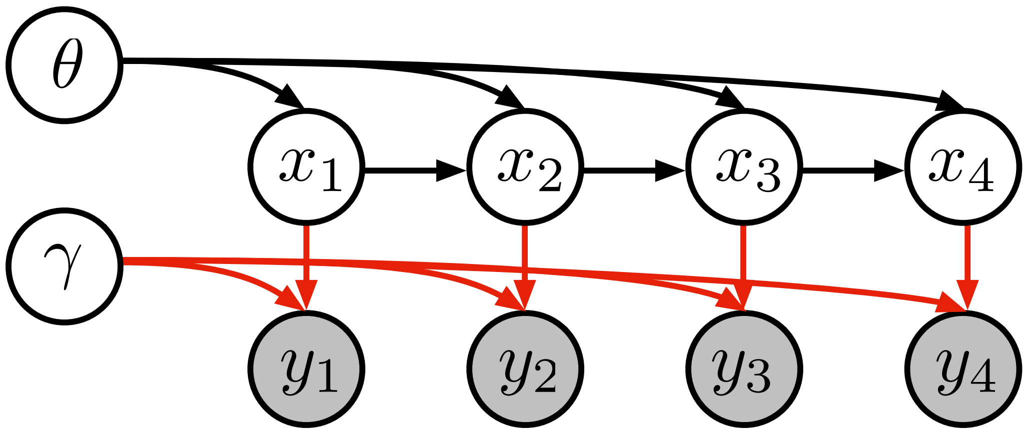
- + fast for general obs.
- ± optimal given conj. evidence
- + exploits conj. graph structure
- + arbitrary inference queries
- + natural gradients on η_θ

[1] **Johnson**, Duvenaud, Wiltschko, Datta, and Adams. Composing graphical models and neural networks. NIPS 2016.

SVAEs: recognition networks output conjugate potentials, then apply fast graphical model algorithms

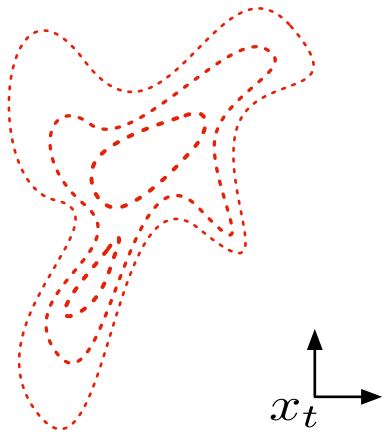


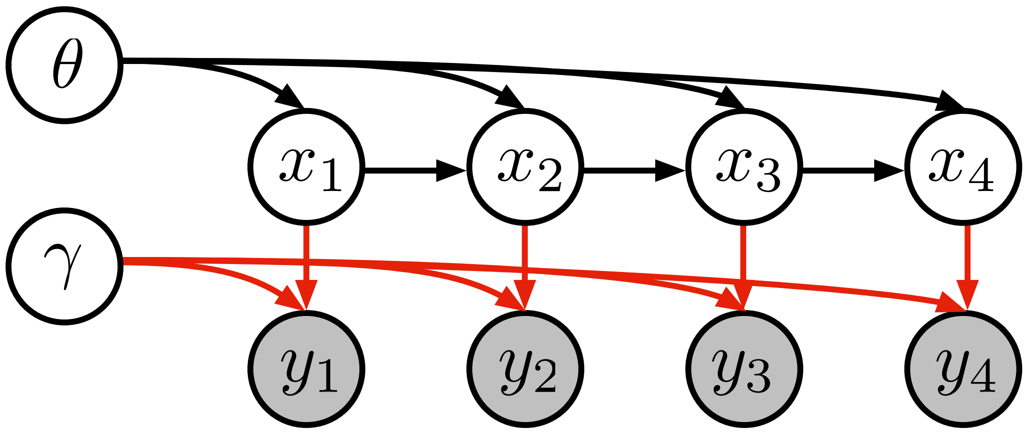
$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y \mid x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$



$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y \mid x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$

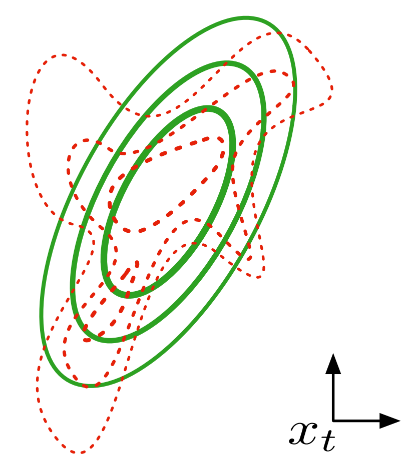
$$\mathbb{E}_{q(\gamma)} \log p(y_t \mid x_t, \gamma)$$



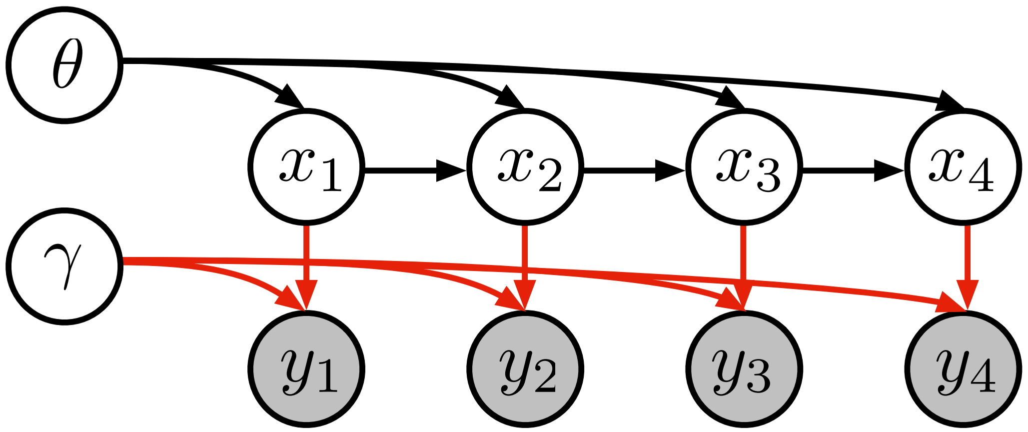


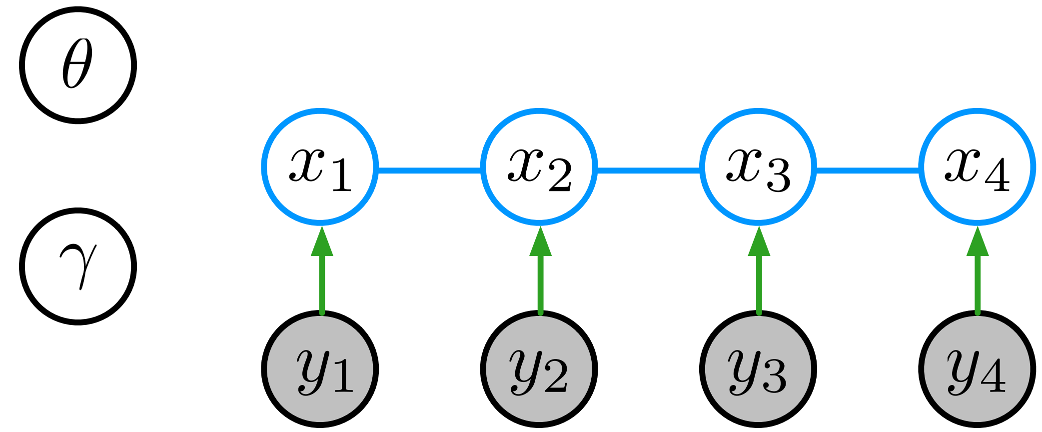
$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y \mid x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$

$\mathbb{E}_{q(\gamma)} \log p(y_t \mid x_t, \gamma)$



$\psi(x_t; y_t, \phi)$



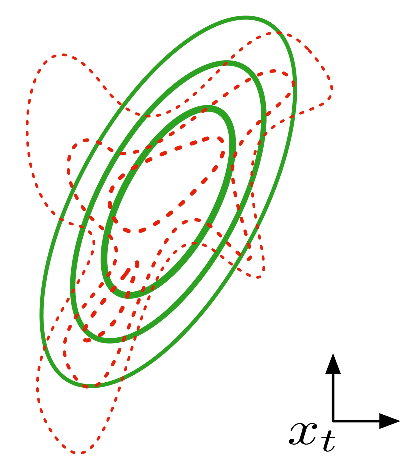


$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y\,|\,x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$

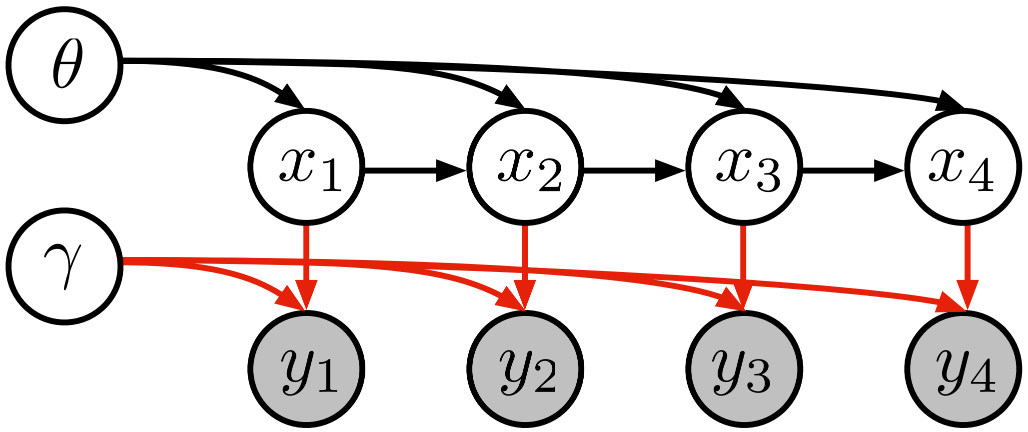
$$\widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)\exp\{\psi(x;y,\phi)\}}{q(\theta)q(\gamma)q(x)}\right]$$

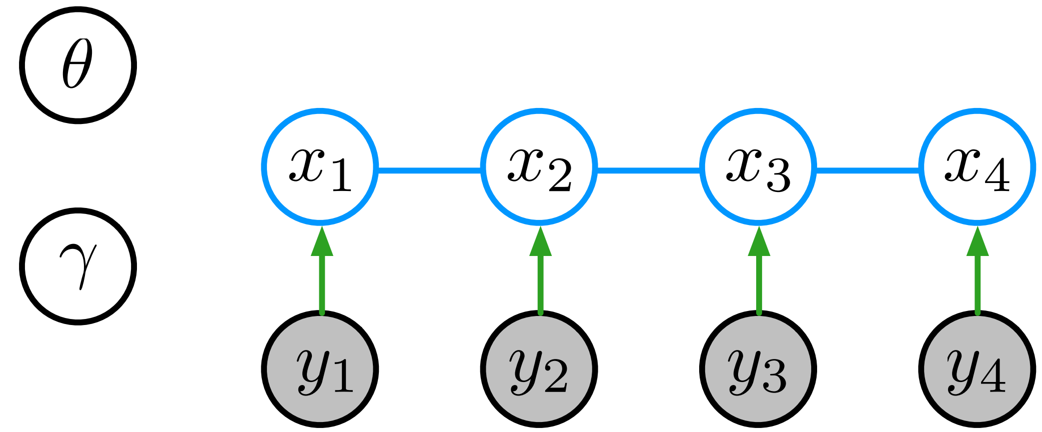
where $\psi(x; y, \phi)$ is a conjugate potential for $p(x\,|\,\theta)$

$\mathbb{E}_{q(\gamma)} \log p(y_t\,|\,x_t, \gamma)$



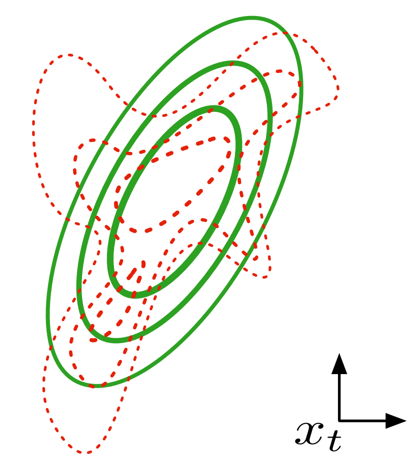
$\psi(x_t; y_t, \phi)$





$\mathbb{E}_{q(\gamma)} \log p(y_t \,|\, x_t, \gamma)$

$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y\,|\,x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$



$$\widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)\exp\{\psi(x;y,\phi)\}}{q(\theta)q(\gamma)q(x)}\right]$$

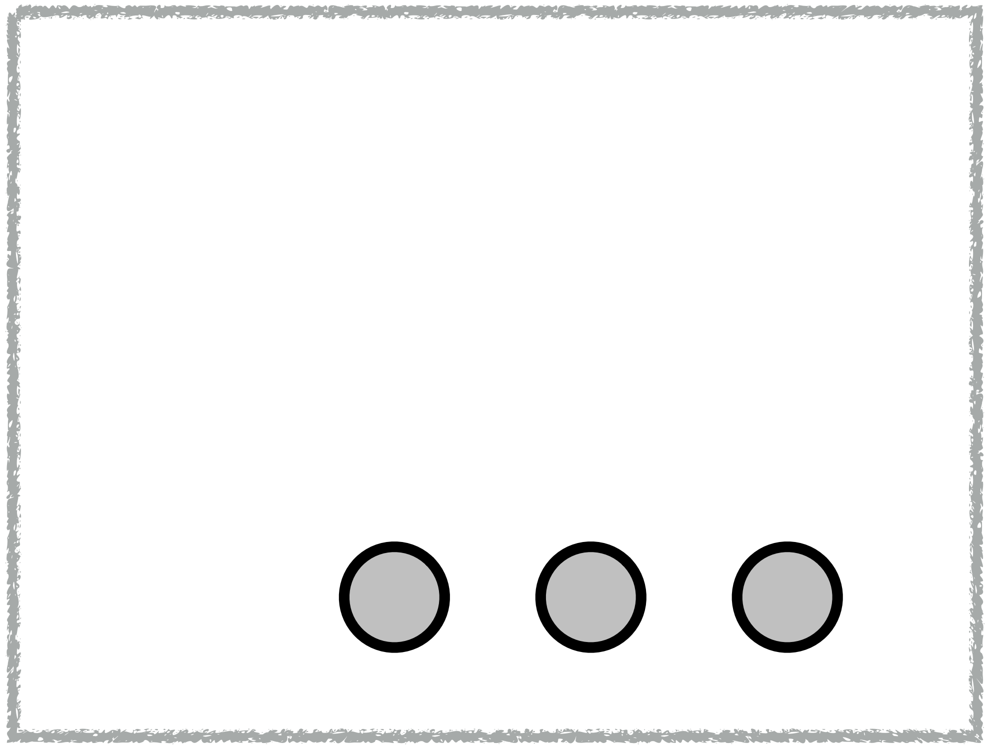
where $\psi(x; y, \phi)$ is a conjugate potential for $p(x \,|\, \theta)$

$\psi(x_t; y_t, \phi)$

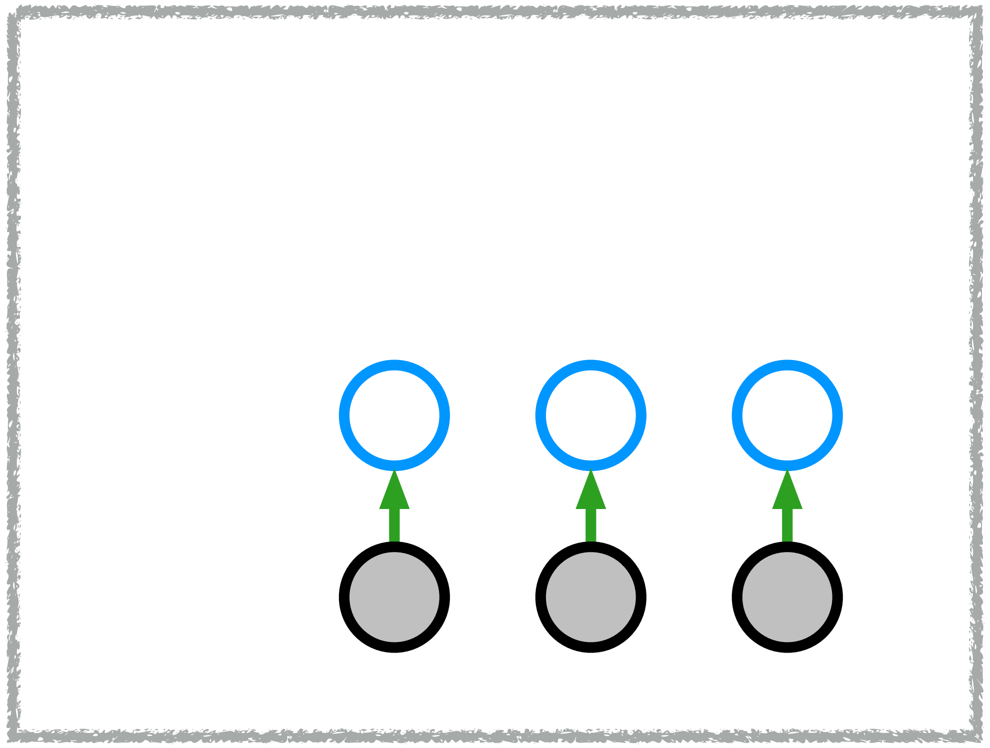
$$\eta_x^*(\eta_\theta, \phi) \triangleq \underset{\eta_x}{\arg\max}\, \widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \qquad \mathcal{L}_{\mathrm{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi))$$

Step 1: apply recognition network

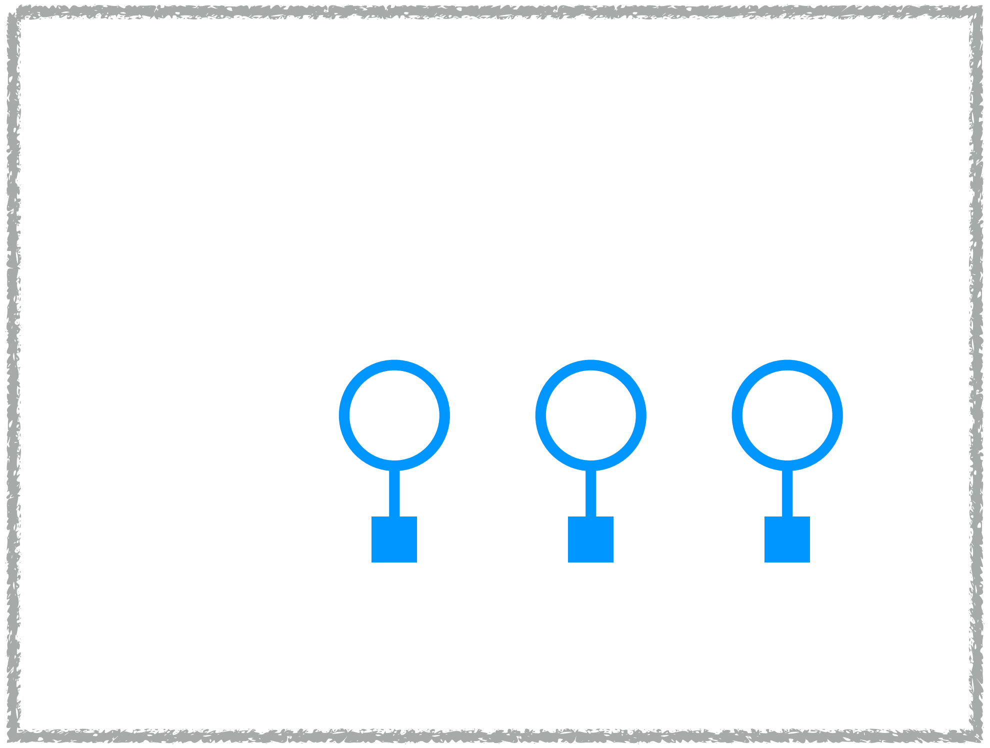
Step 1: apply recognition network



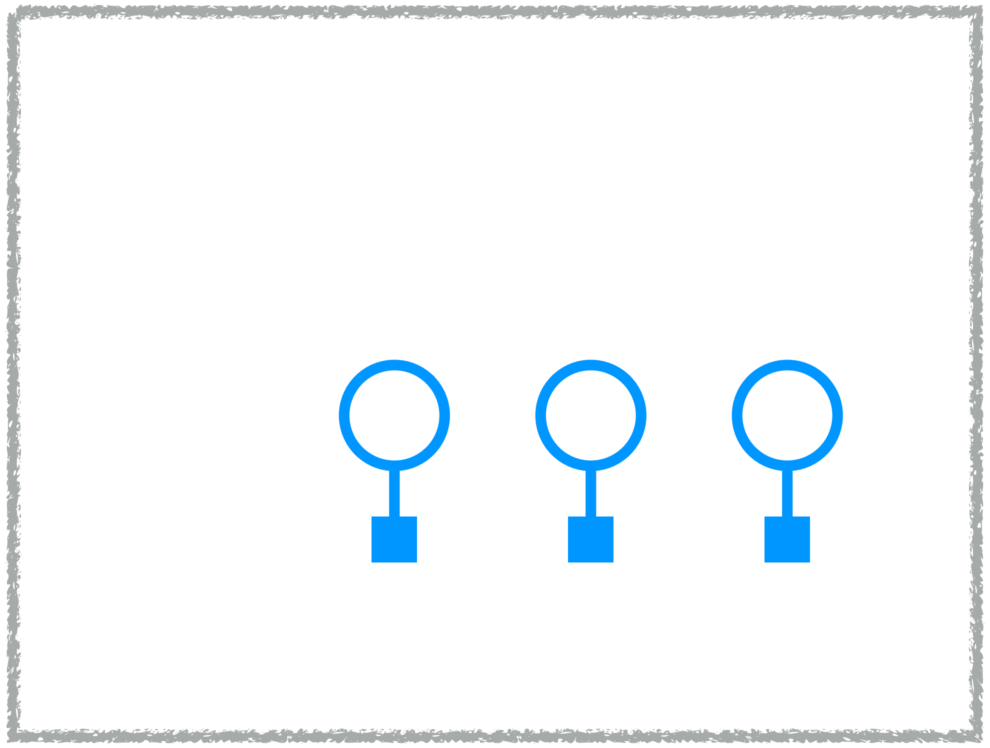
Step 1: apply recognition network



Step 1: apply recognition network

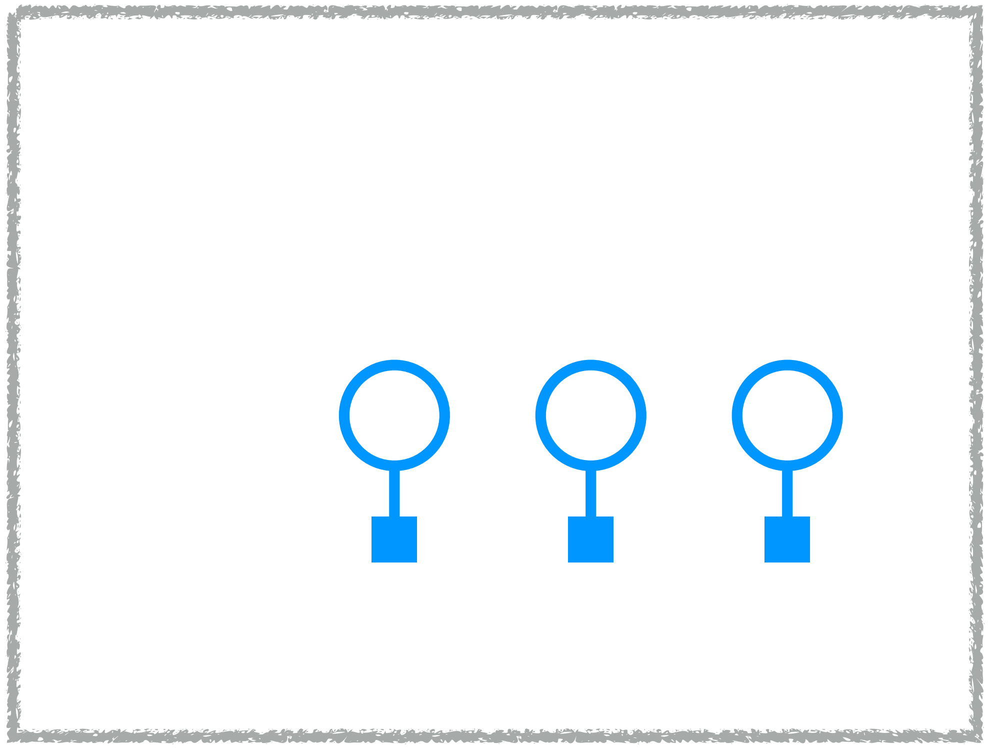


Step 1: apply recognition network

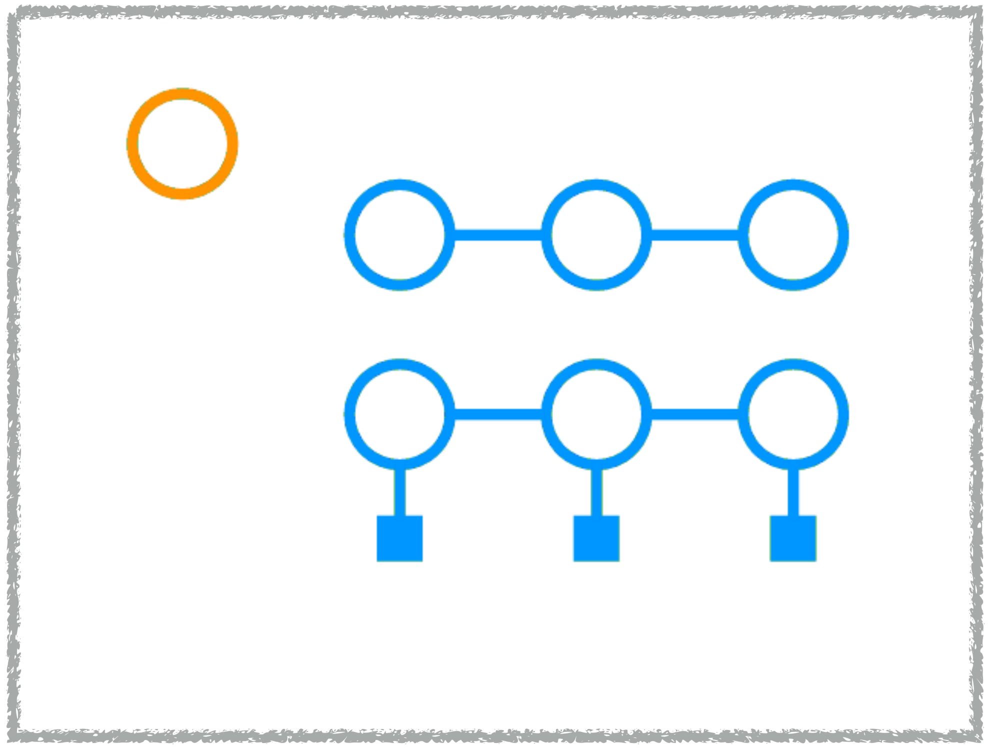


Step 2: run fast PGM algorithms

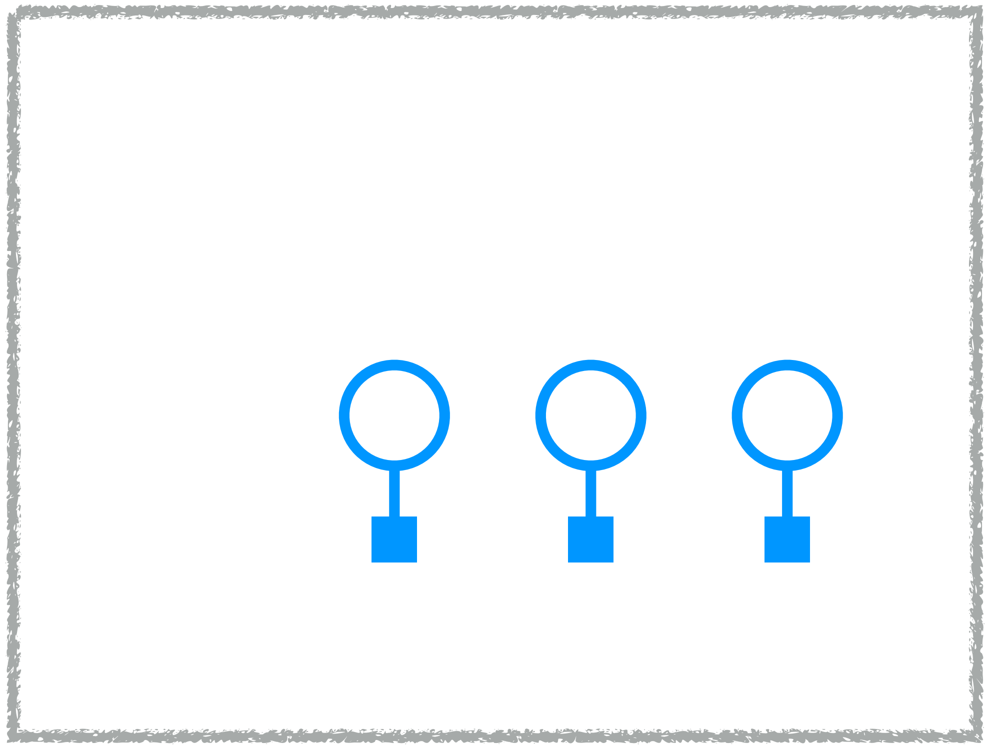
Step 1: apply recognition network



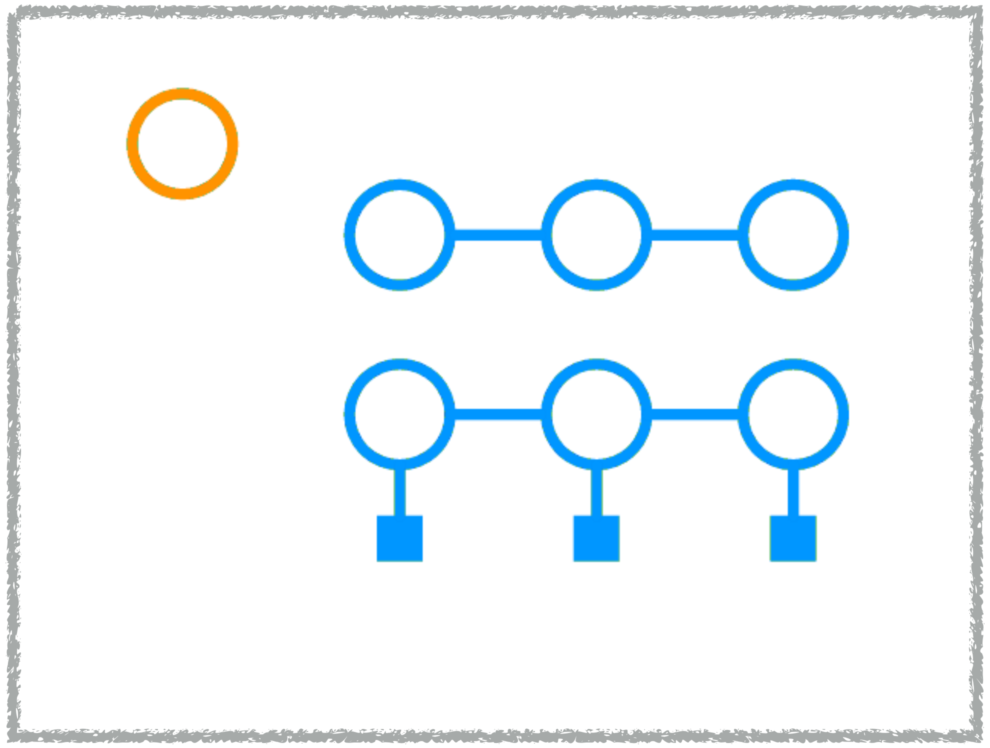
Step 2: run fast PGM algorithms



Step 1: apply recognition network

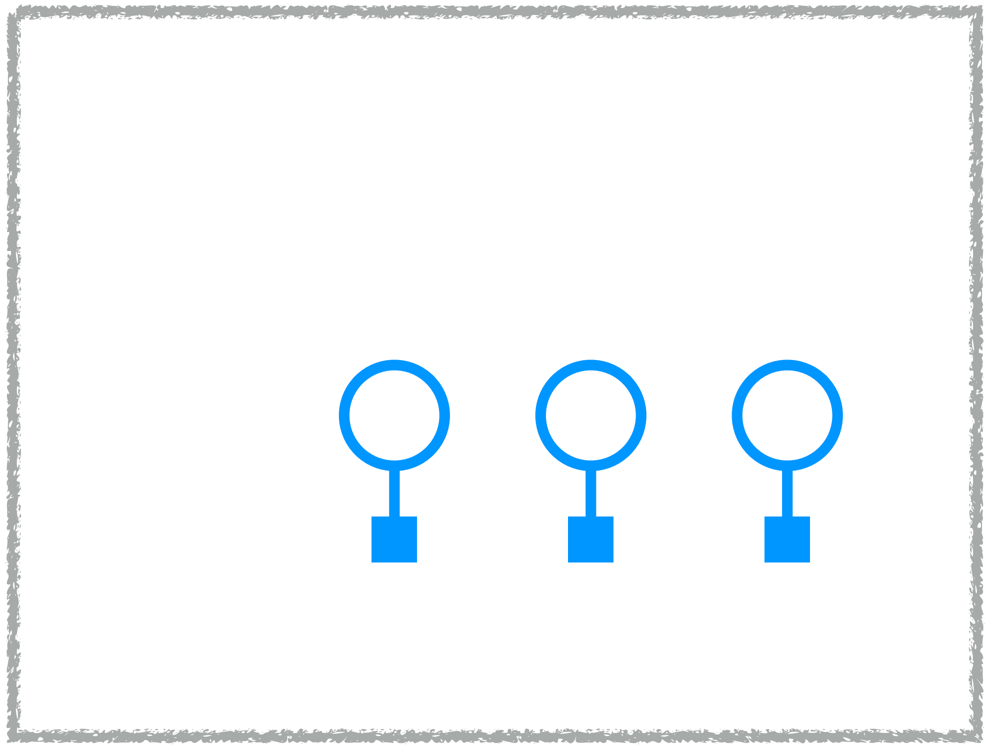


Step 2: run fast PGM algorithms

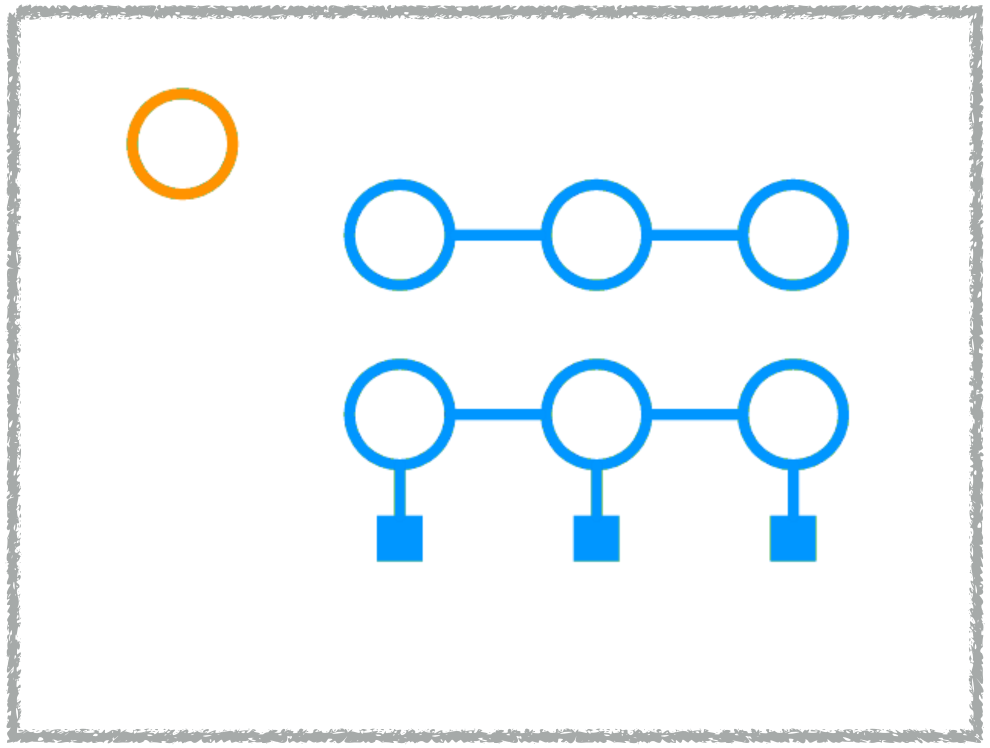


Step 3: sample, compute flat grads

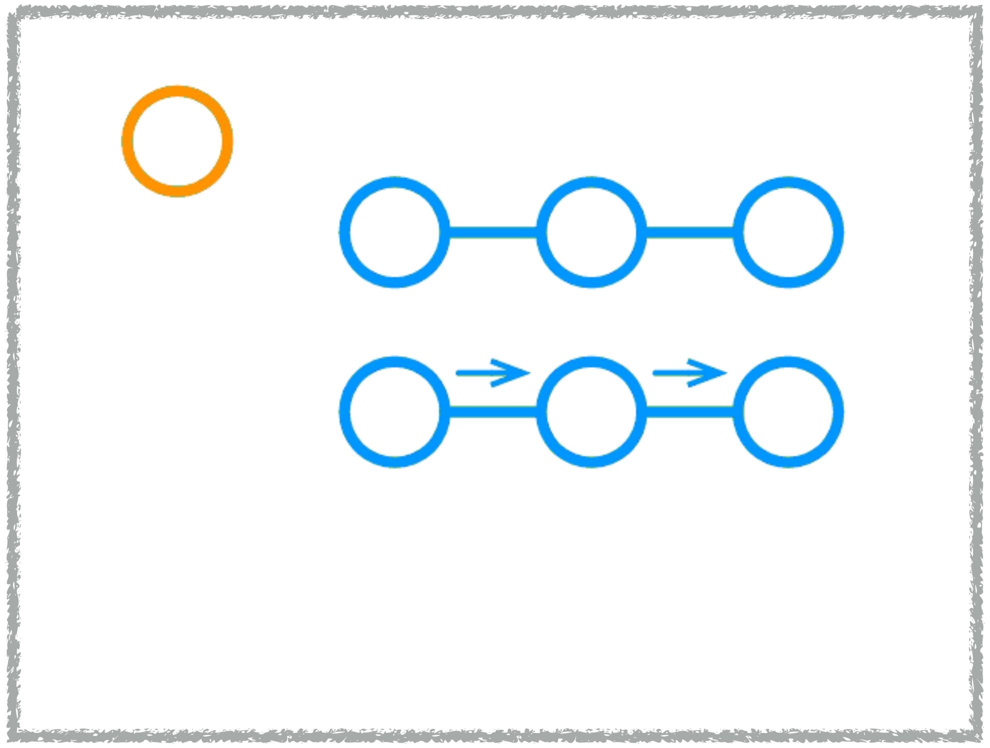
Step 1: apply recognition network



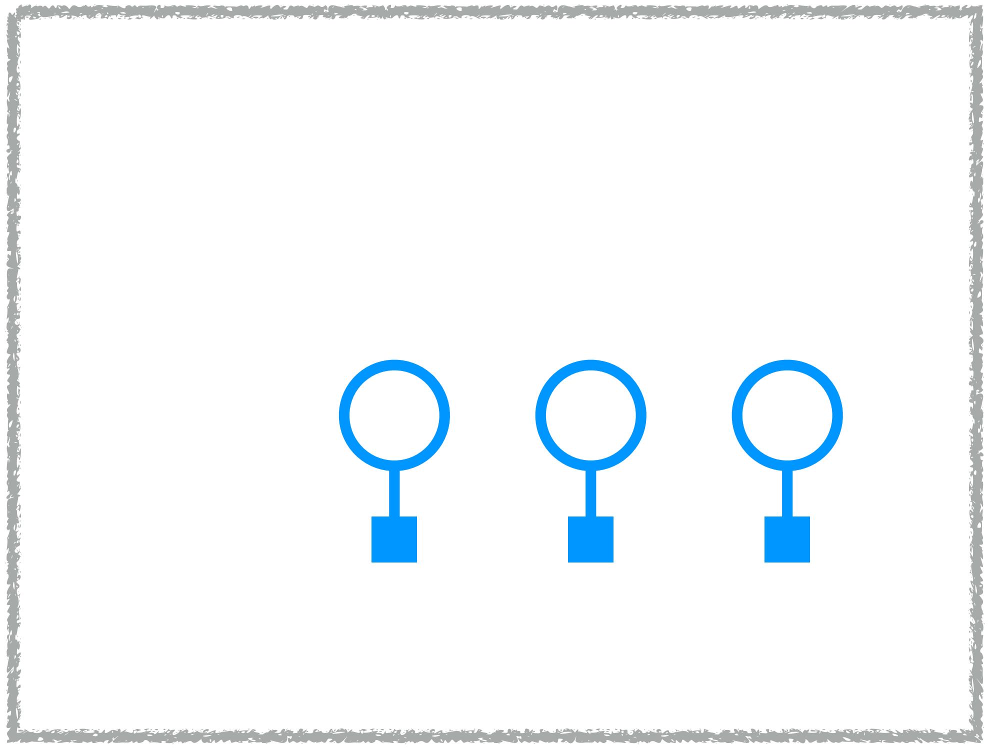
Step 2: run fast PGM algorithms



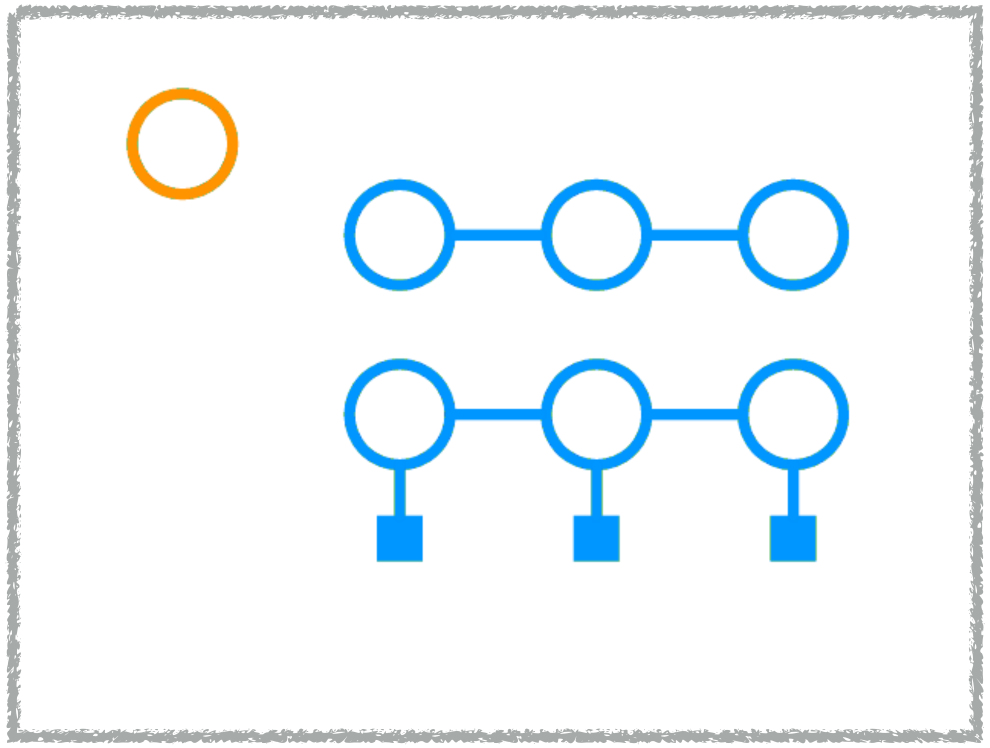
Step 3: sample, compute flat grads



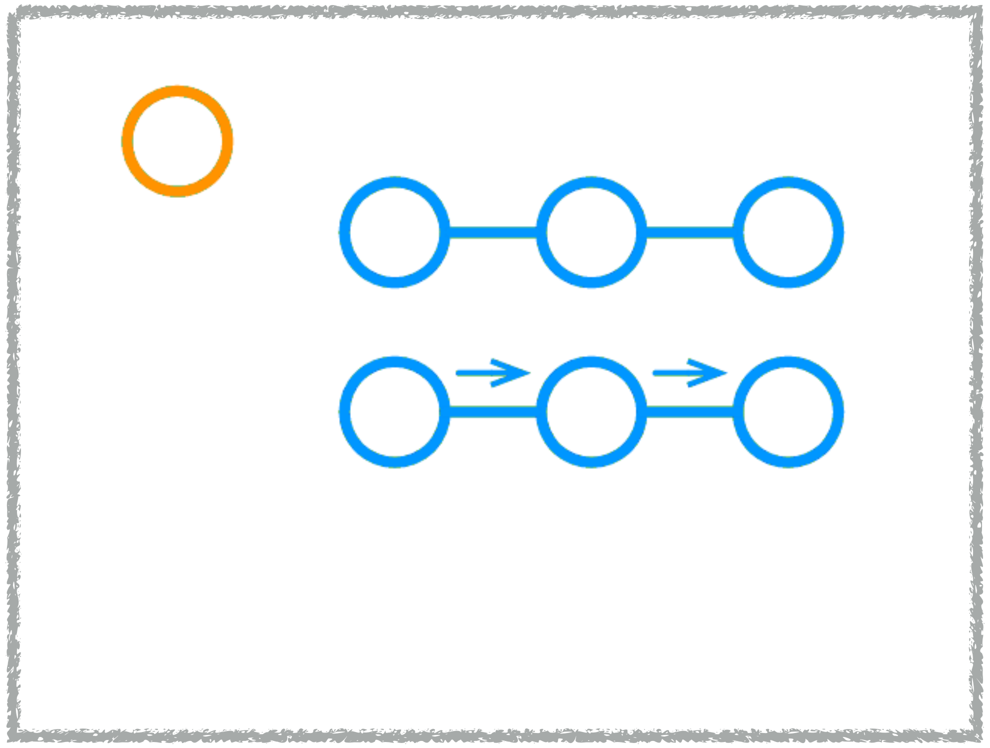
Step 1: apply recognition network



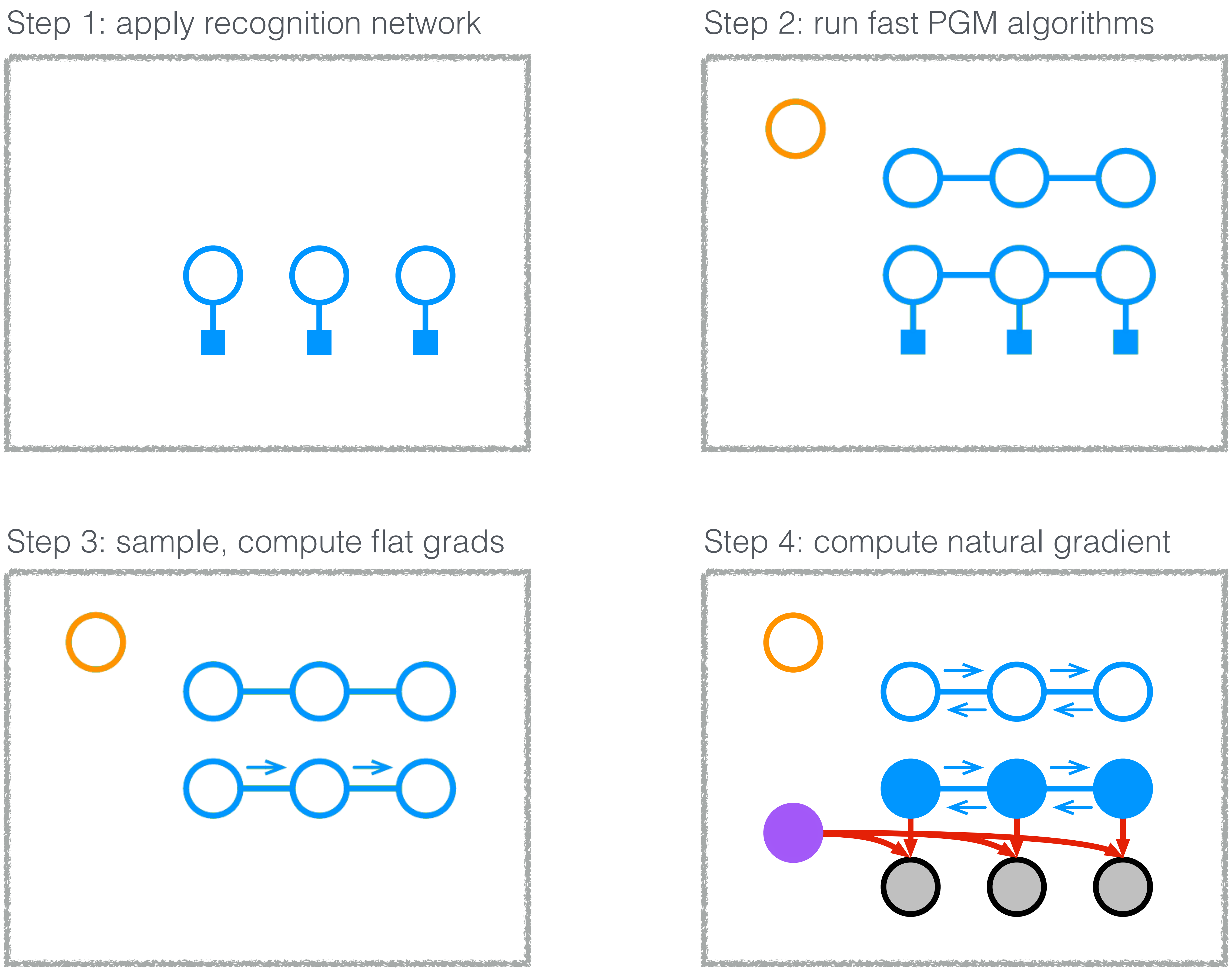
Step 2: run fast PGM algorithms



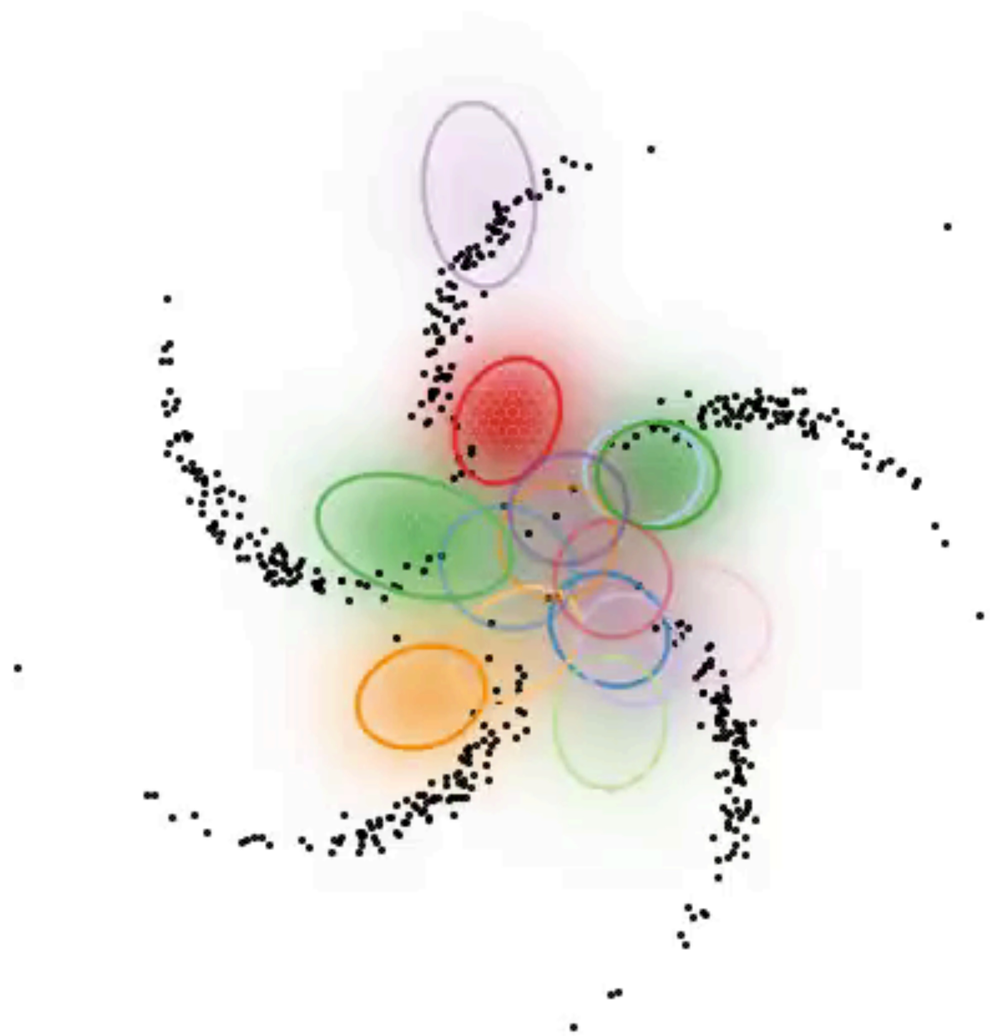
Step 3: sample, compute flat grads



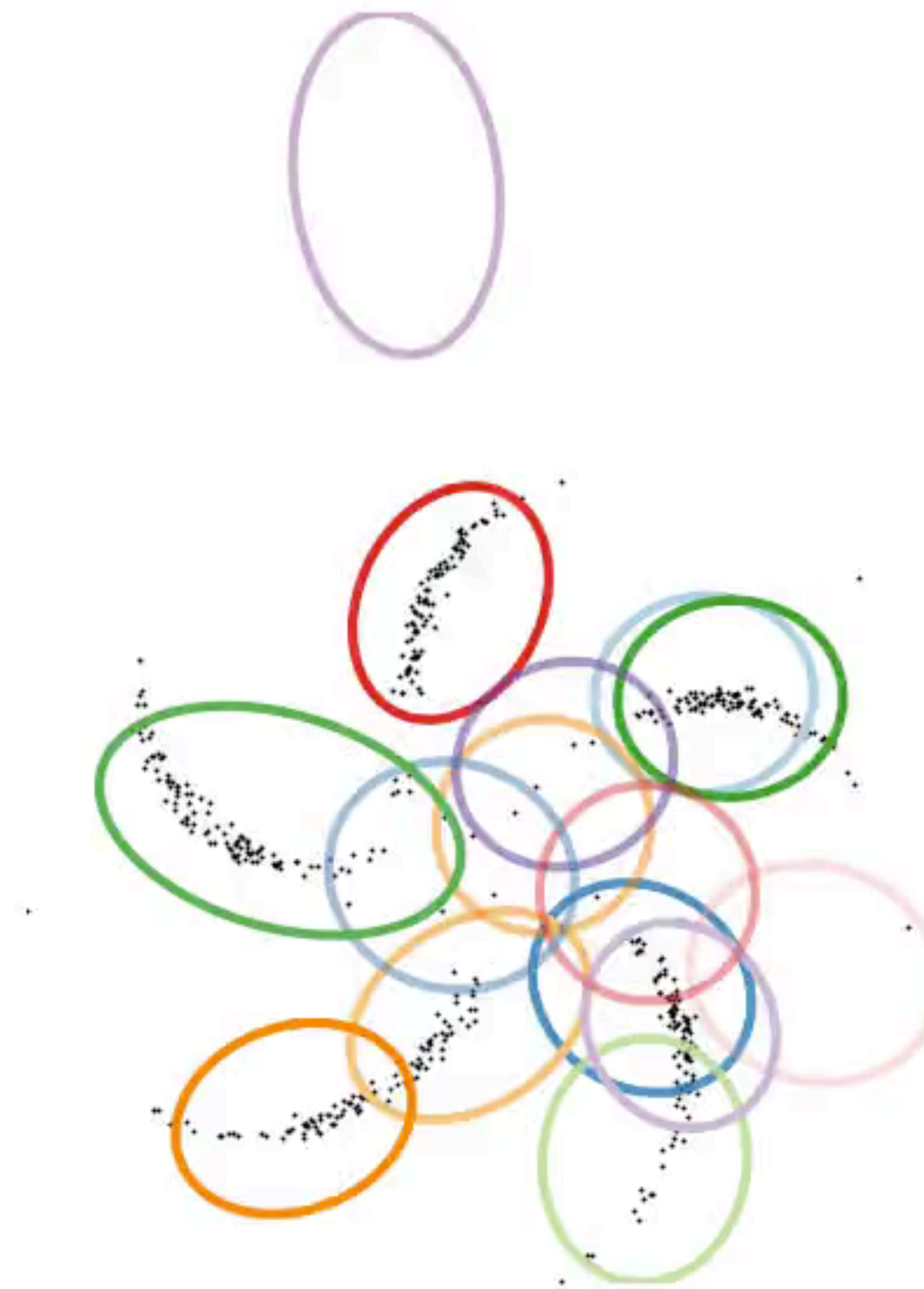
Step 4: compute natural gradient



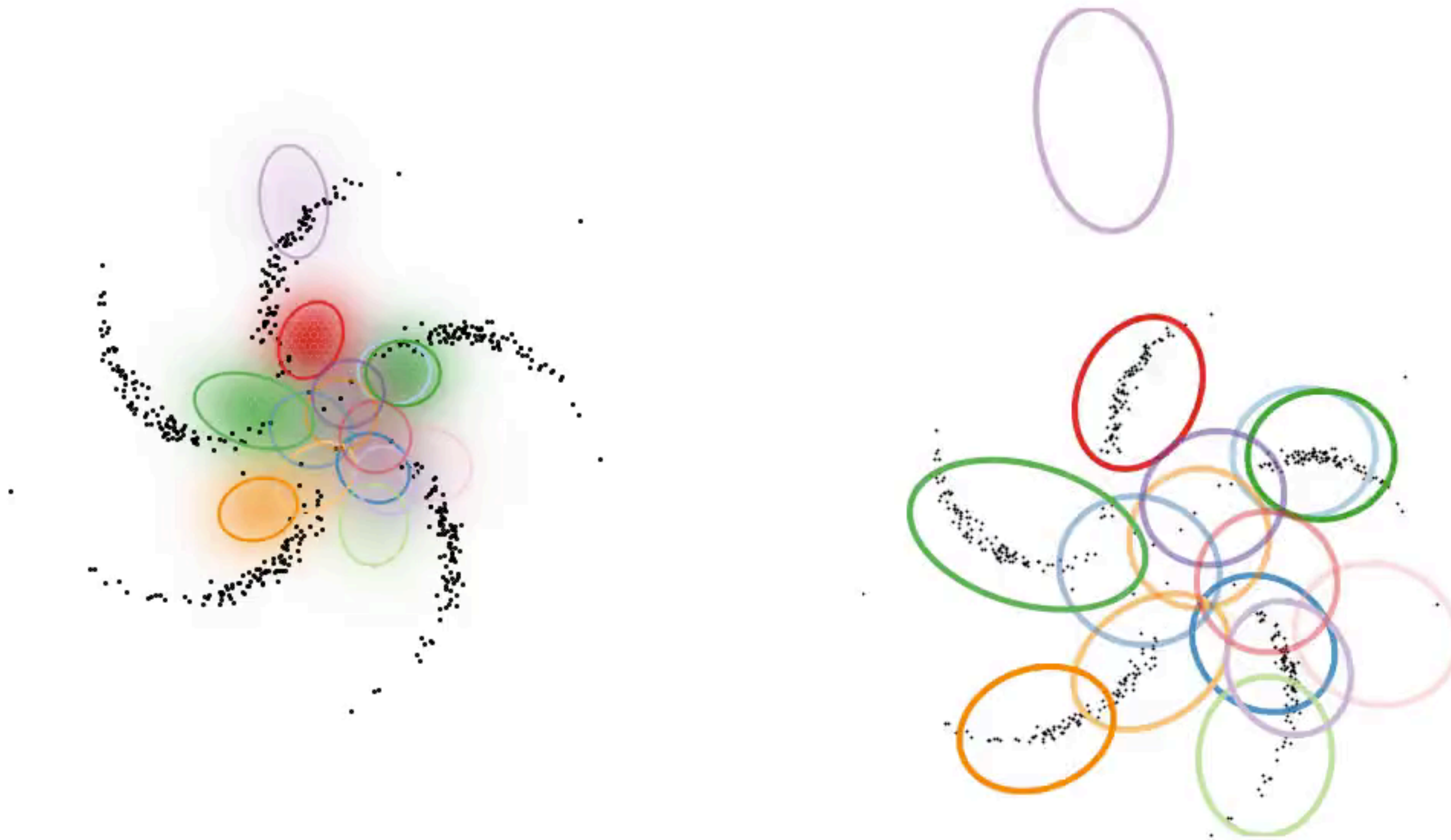
Step 1: apply recognition network
Step 2: run fast PGM algorithms
Step 3: sample, compute flat grads
Step 4: compute natural gradient



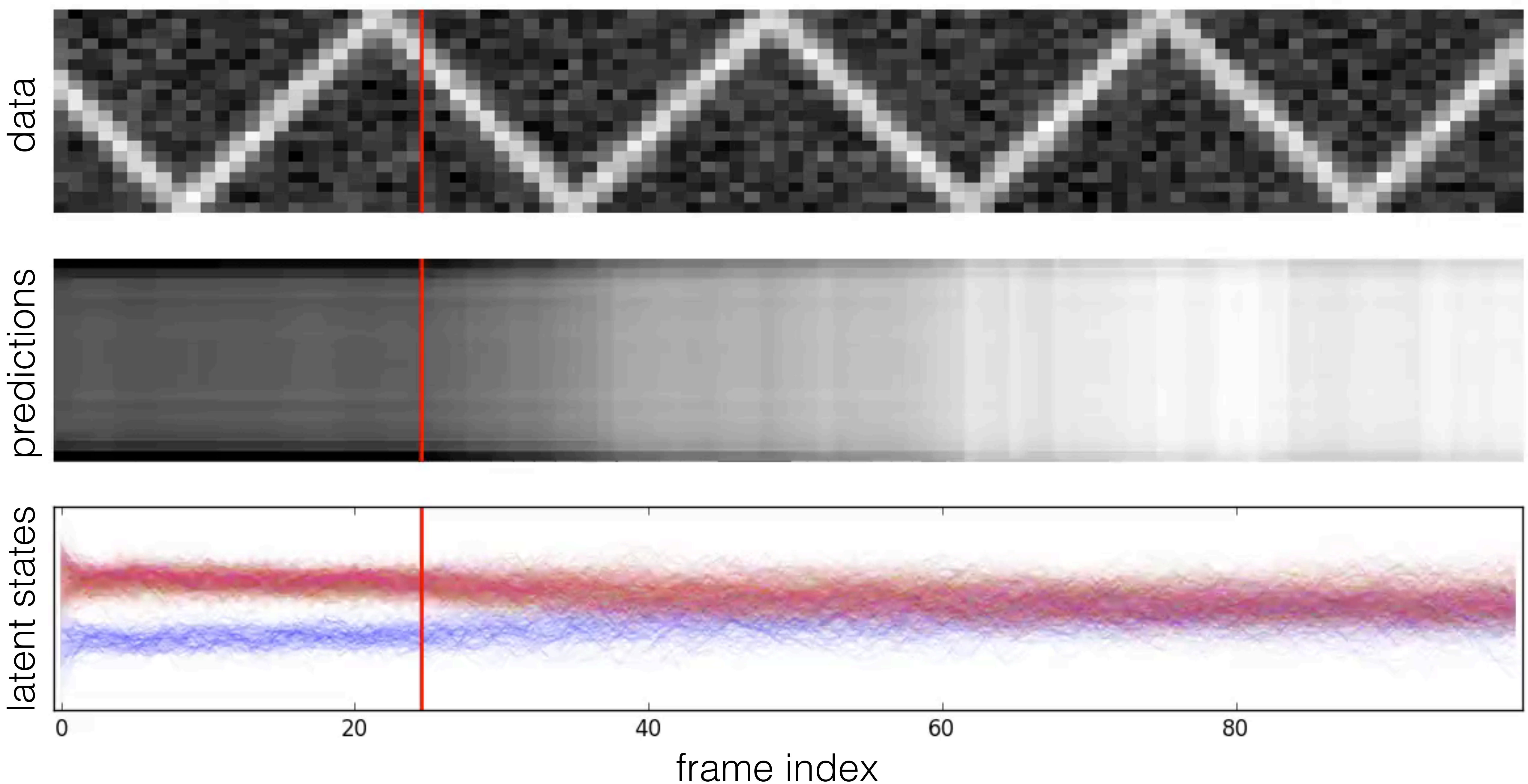
data space

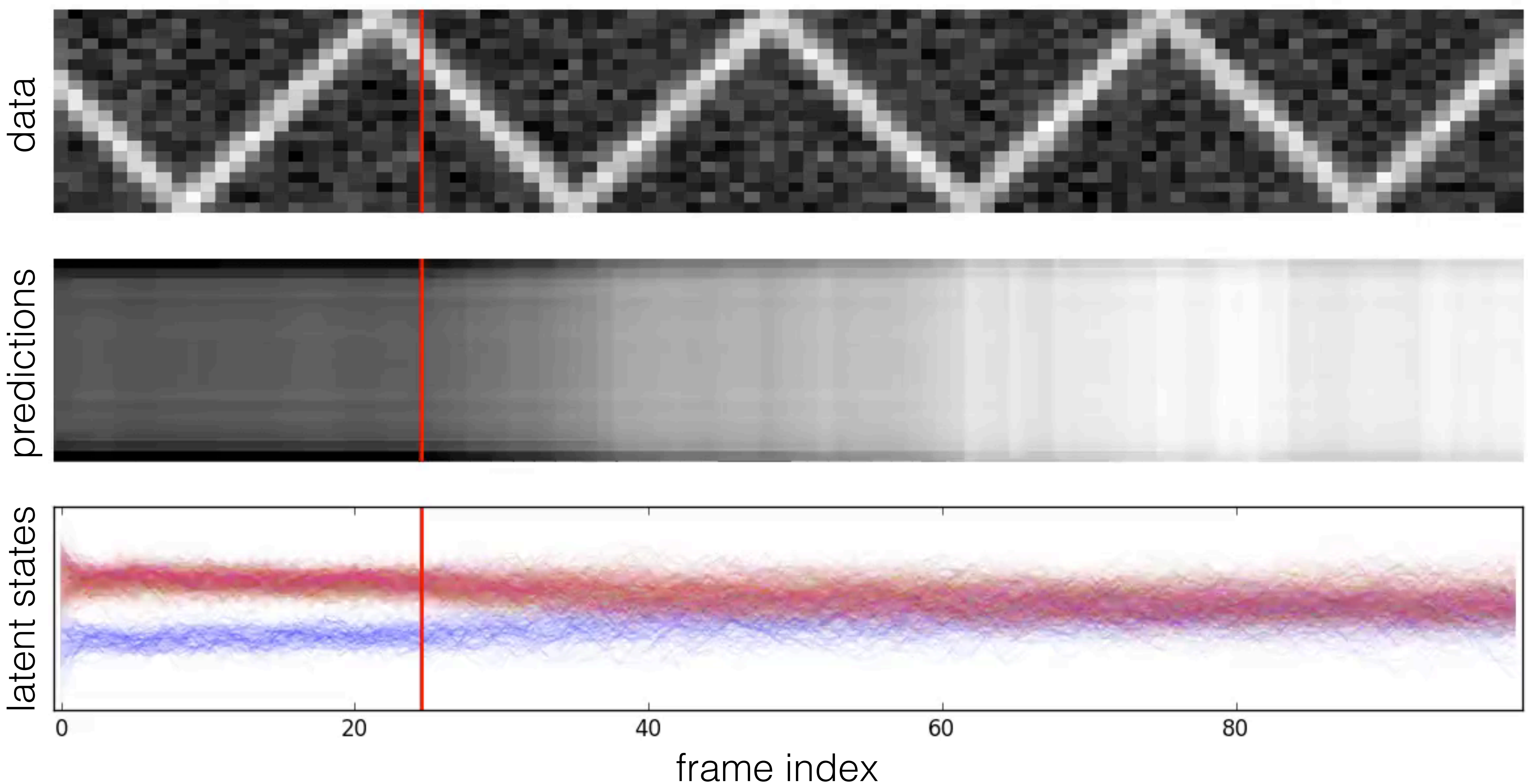


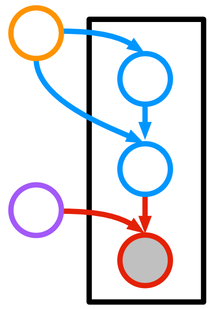
latent space

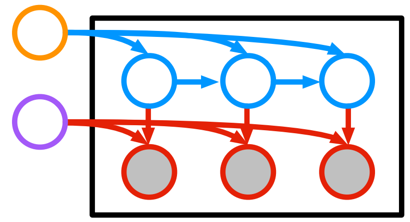


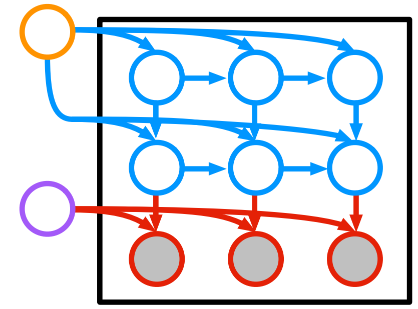
data space　　　　　　　　　　　　　　latent space

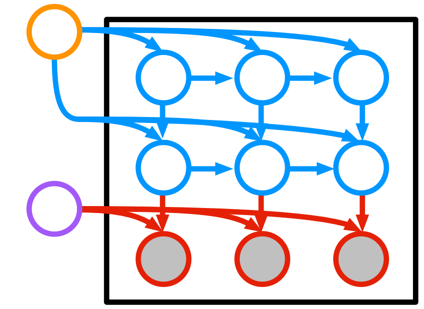

data
predictions
latent states
0
20
40
60
80
frame index


data
predictions
latent states
0
20
40
60
80
frame index

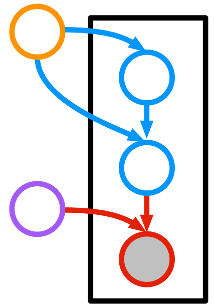

[1]
Gaussian mixture model


[2]
Linear dynamical system


[3]
Hidden Markov model

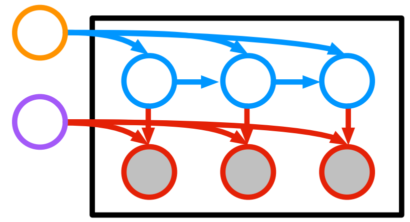

[4]
Switching LDS

[1] Corduneanu and Bishop. Variational Bayesian Model Selection for Mixture Distributions. AISTATS 2001.
[2] Ghahramani and Beal. Propagation algorithms for variational Bayesian learning. NIPS 2001.
[3] Beal. Variational algorithms for approximate Bayesian inference, Ch. 3. U of London Ph.D. Thesis 2003.
[4] Ghahramani and Hinton. Variational learning for switching state-space models. Neural Computation 2000.



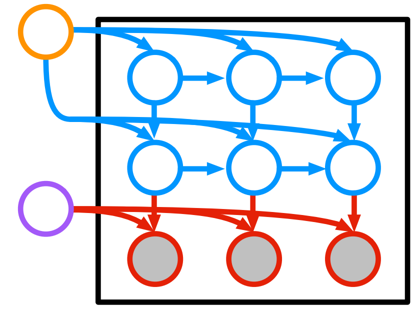
[1]

Gaussian mixture model



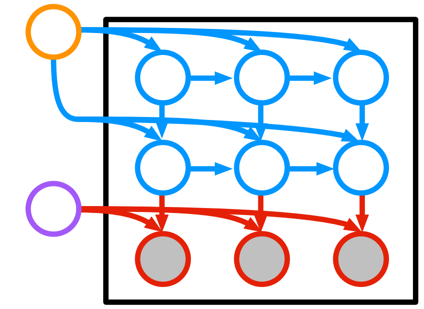
[2]

Linear dynamical system



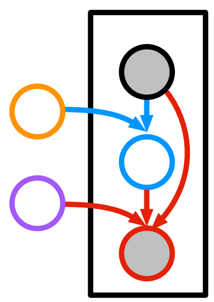
[3]

Hidden Markov model



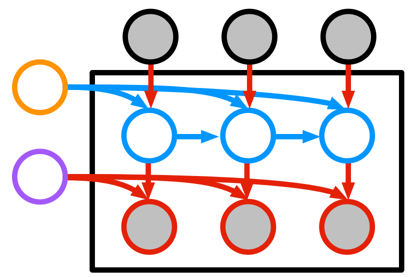
[4]

Switching LDS



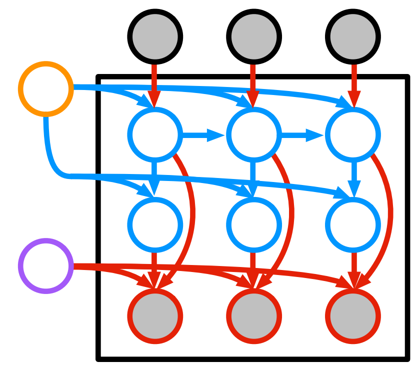
[5]

Mixture of Experts



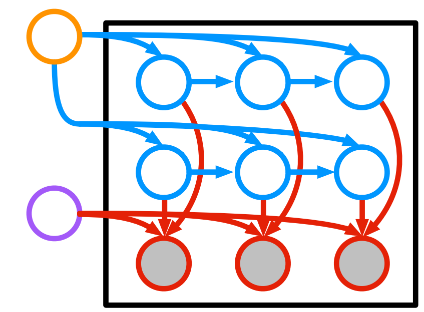
[2]

Driven LDS



[6]

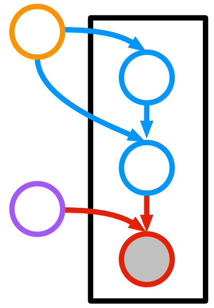
IO-HMM



[7]

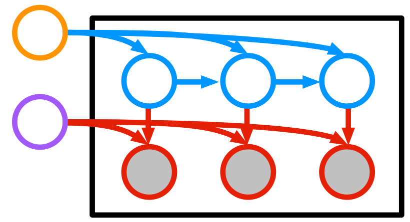
Factorial HMM

[1] Corduneanu and Bishop. Variational Bayesian Model Selection for Mixture Distributions. AISTATS 2001.
[2] Ghahramani and Beal. Propagation algorithms for variational Bayesian learning. NIPS 2001.
[3] Beal. Variational algorithms for approximate Bayesian inference, Ch. 3. U of London Ph.D. Thesis 2003.
[4] Ghahramani and Hinton. Variational learning for switching state-space models. Neural Computation 2000.
[5] Jordan and Jacobs. Hierarchical Mixtures of Experts and the EM algorithm. Neural Computation 1994.
[6] Bengio and Frasconi. An Input Output HMM Architecture. NIPS 1995.
[7] Ghahramani and Jordan. Factorial Hidden Markov Models. Machine Learning 1997.



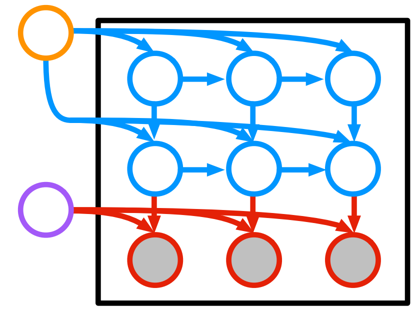
[1]

Gaussian mixture model



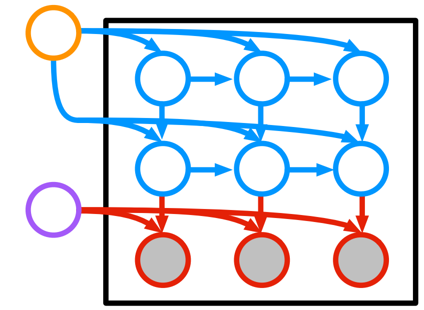
[2]

Linear dynamical system



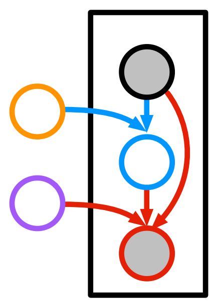
[3]

Hidden Markov model



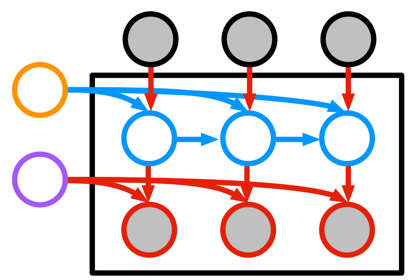
[4]

Switching LDS



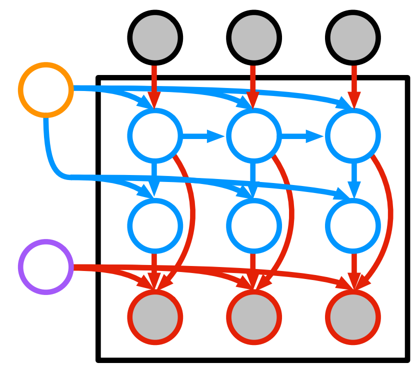
[5]

Mixture of Experts



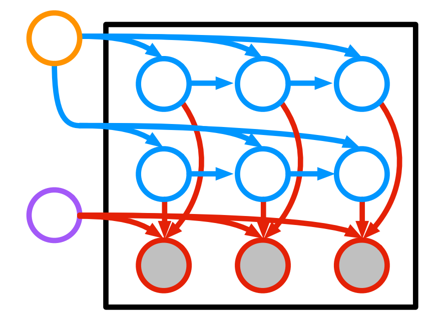
[2]

Driven LDS



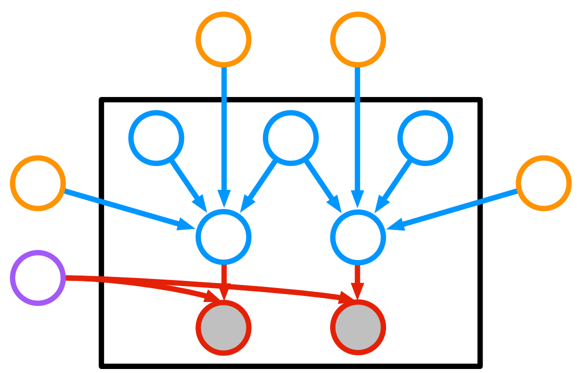
[6]

IO-HMM



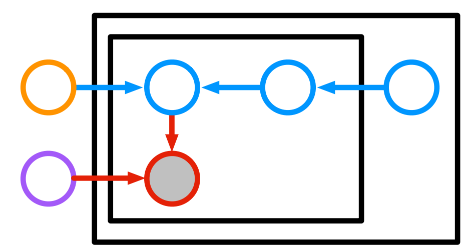
[7]

Factorial HMM



[8,9]

Canonical correlations analysis

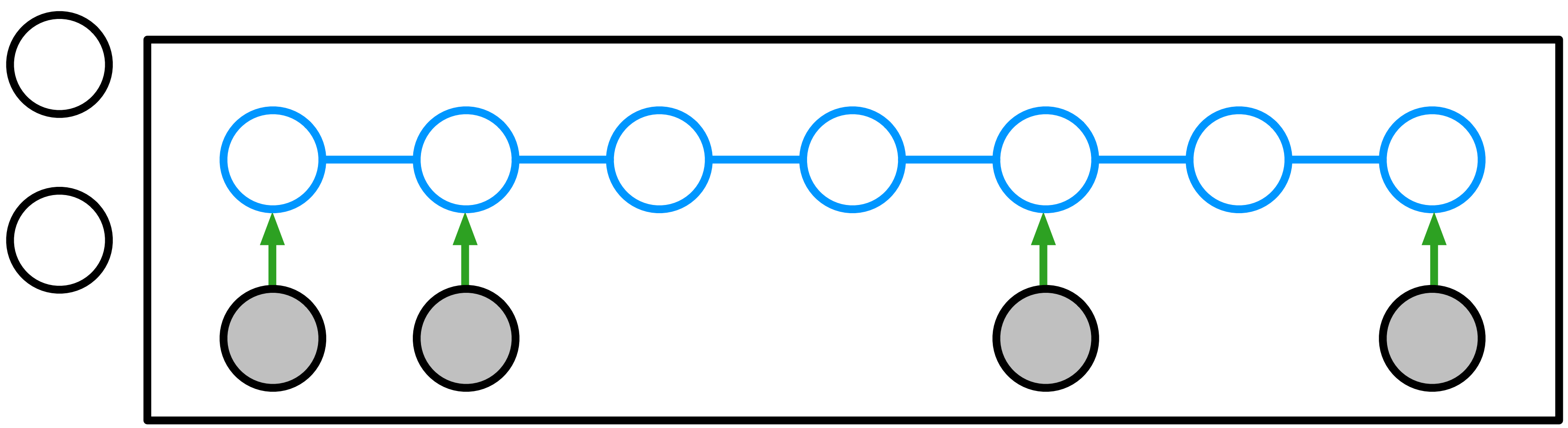


[10]

admixture / LDA / NMF

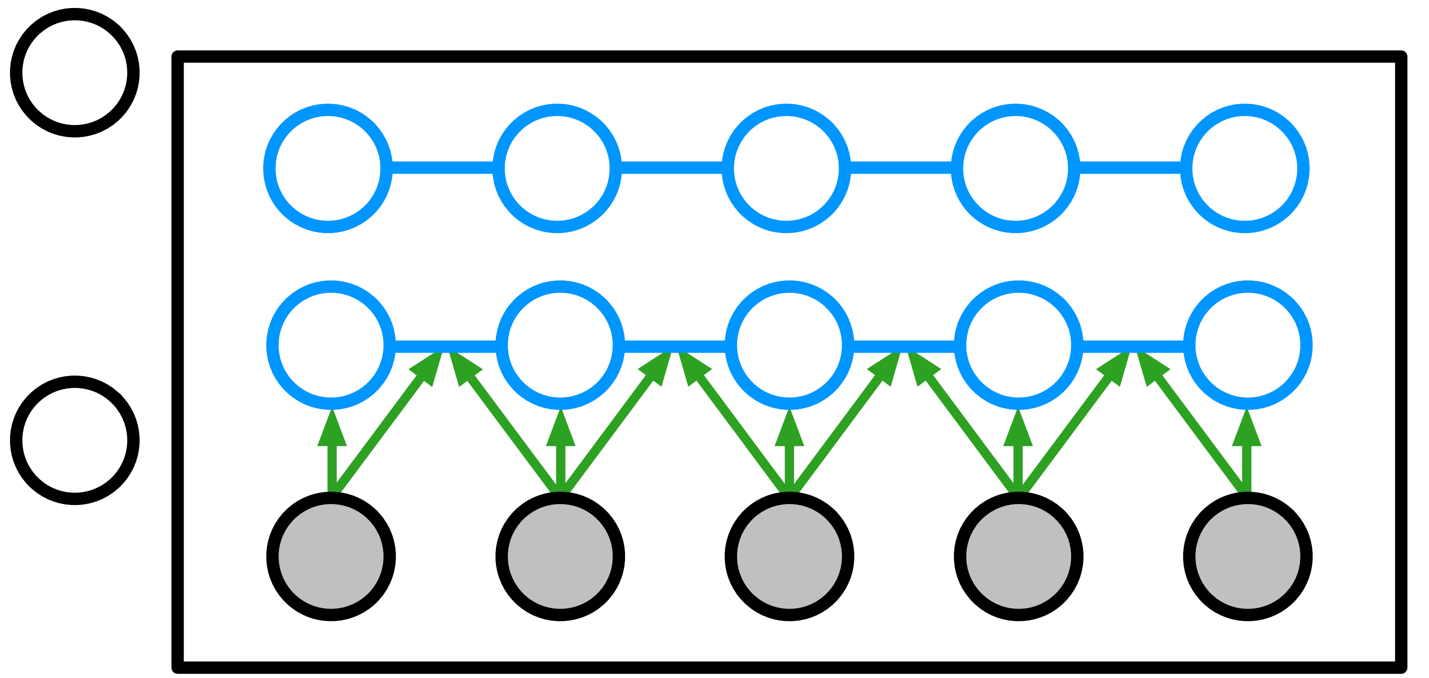
[1] Corduneanu and Bishop. Variational Bayesian Model Selection for Mixture Distributions. AISTATS 2001.
[2] Ghahramani and Beal. Propagation algorithms for variational Bayesian learning. NIPS 2001.
[3] Beal. Variational algorithms for approximate Bayesian inference, Ch. 3. U of London Ph.D. Thesis 2003.
[4] Ghahramani and Hinton. Variational learning for switching state-space models. Neural Computation 2000.
[5] Jordan and Jacobs. Hierarchical Mixtures of Experts and the EM algorithm. Neural Computation 1994.
[6] Bengio and Frasconi. An Input Output HMM Architecture. NIPS 1995.
[7] Ghahramani and Jordan. Factorial Hidden Markov Models. Machine Learning 1997.
[8] Bach and Jordan. A probabilistic interpretation of Canonical Correlation Analysis. Tech. Report 2005.
[9] Archambeau and Bach. Sparse probabilistic projections. NIPS 2008.
[10] Hoffman, Bach, Blei. Online learning for Latent Dirichlet Allocation. NIPS 2010.

arbitrary inference queries*

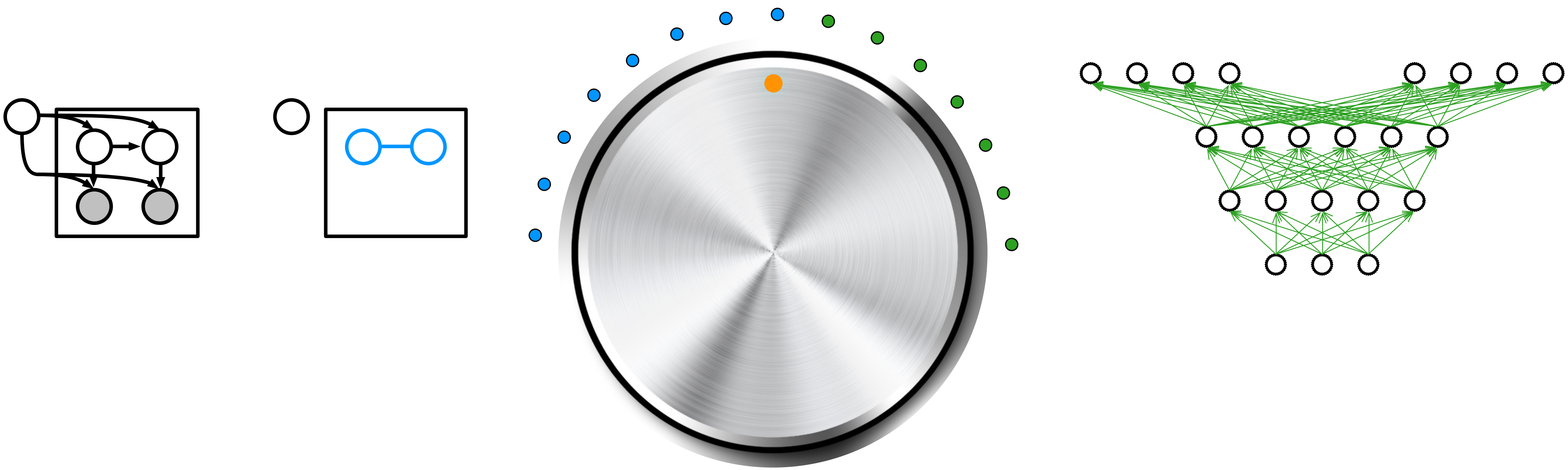


*see next slide

SVAEs can use any inference network architectures

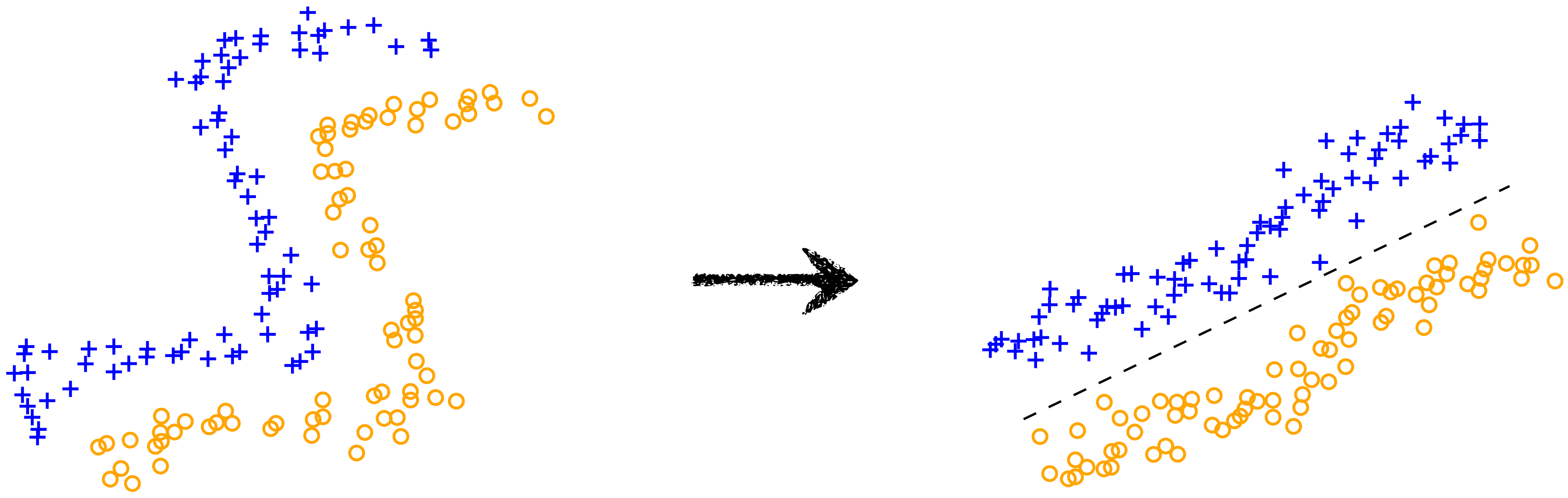


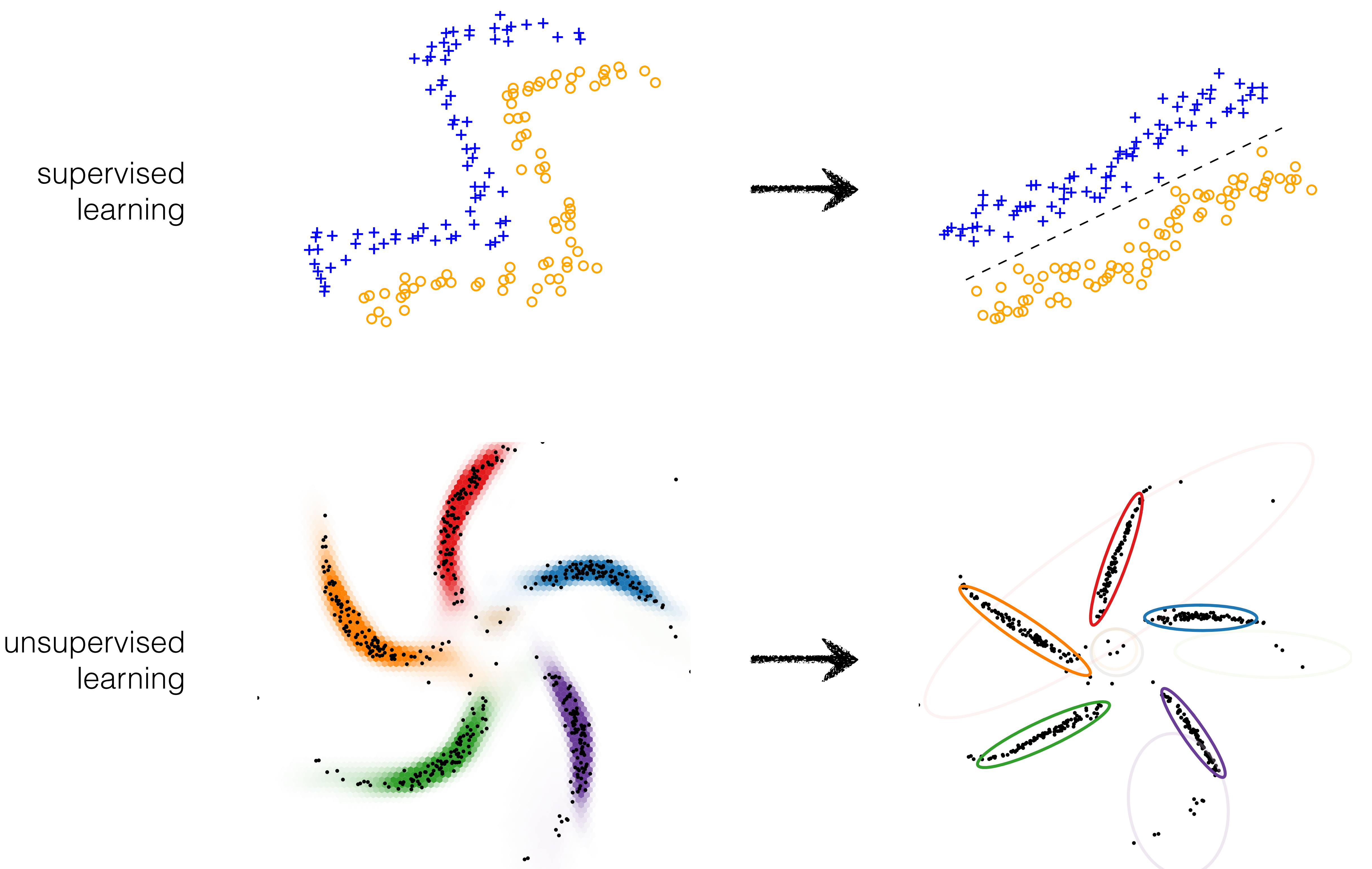
[1] Archer, Park, Buesing, Cunningham, Paninski. Black box variational inference for state space models. ICLR 2016 Workshops.
[2] Gao*, Archer*, Paninski, Cunningham. Linear dynamical neural population models through nonlinear embeddings. NIPS 2016.

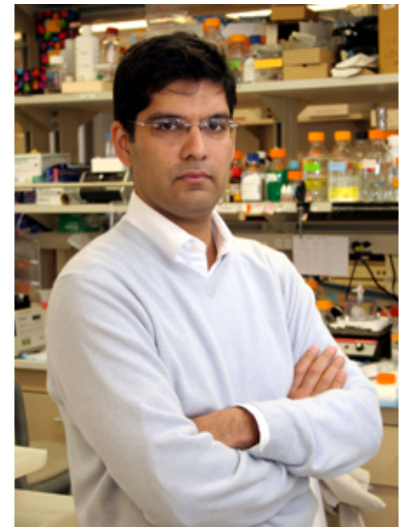


SVAEs

supervised
learning




supervised
learning
unsupervised
learning

github.com/mattjj/svae

HARVARD INTELLIGENT PROBABILISTIC SYSTEMS

HARVARD MEDICAL SCHOOL