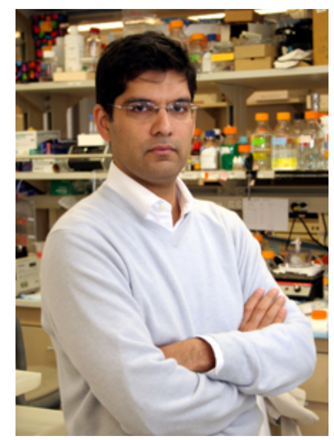
Composing graphical models with neural networks for structured representations and fast inference

Matthew J Johnson (mattjj@google.com)

Deep Learning Summer School

Montreal 2017

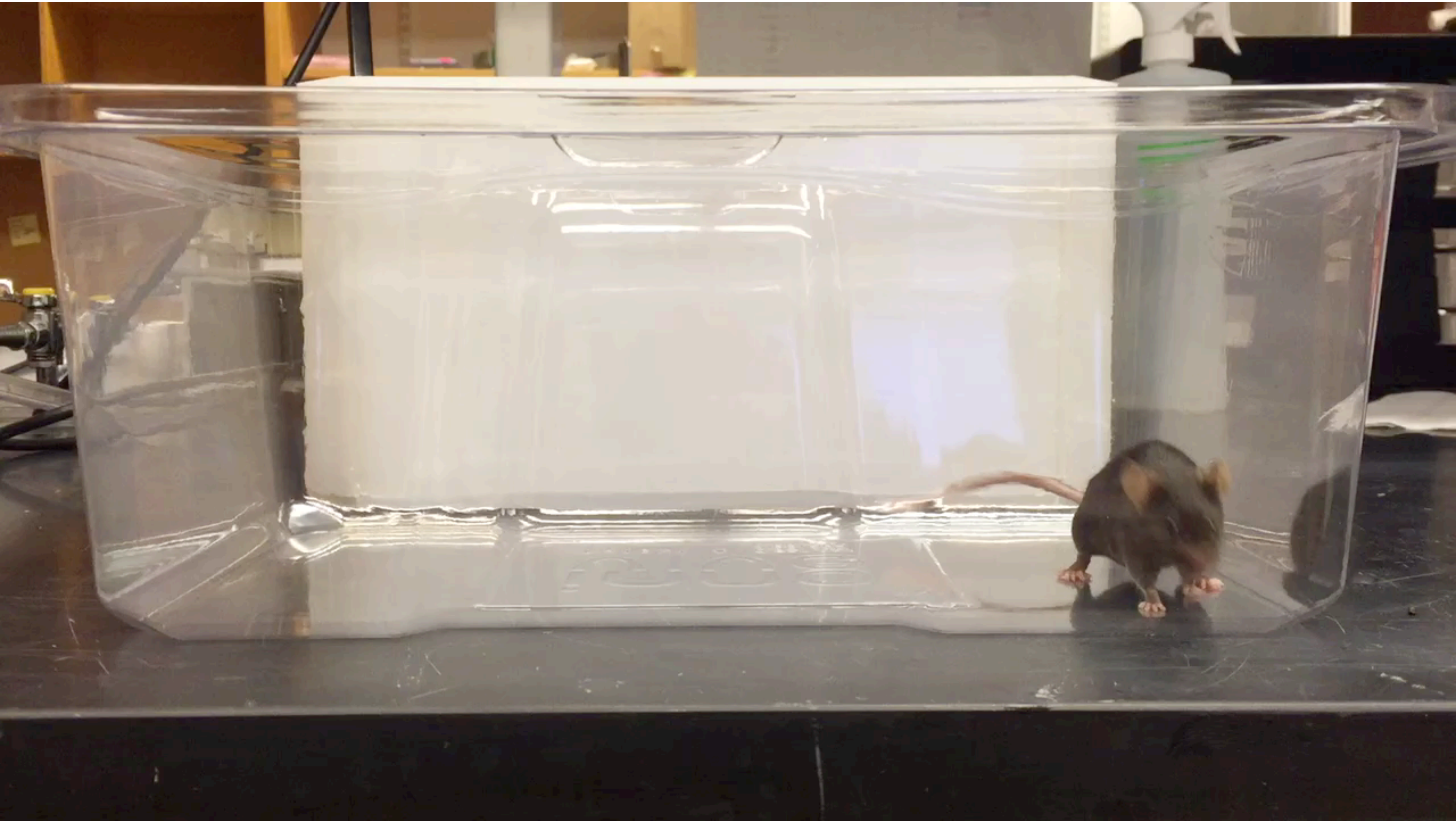


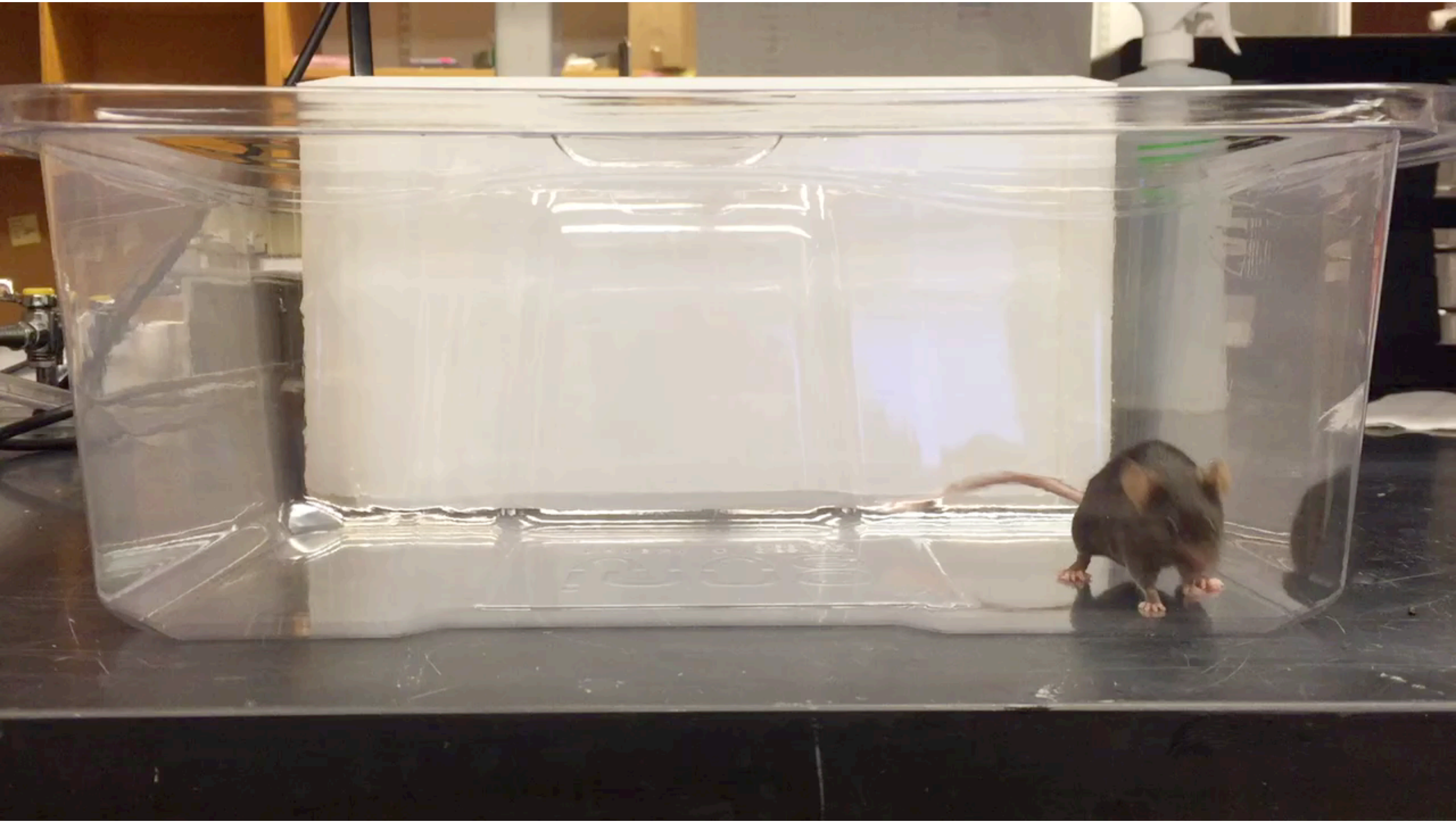


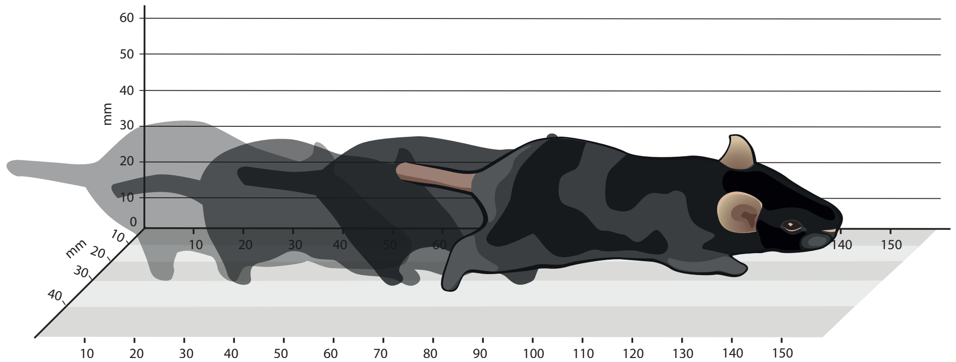



HARVARD INTELLIGENT PROBABILISTIC SYSTEMS

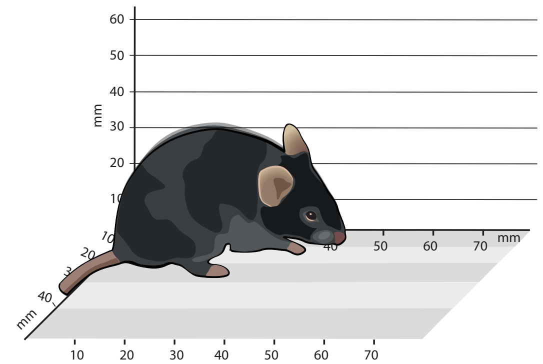
HARVARD MEDICAL SCHOOL



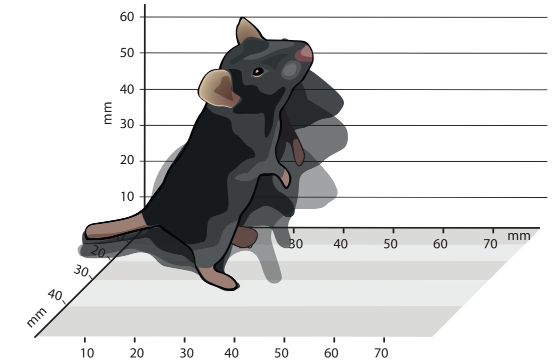




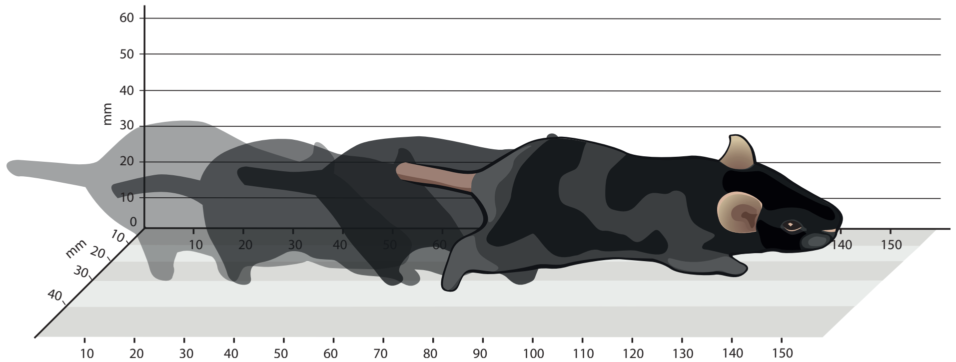
dart



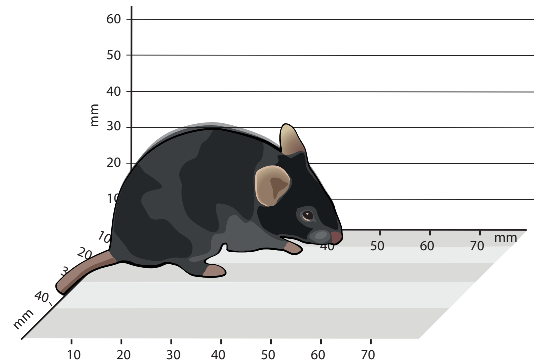
pause



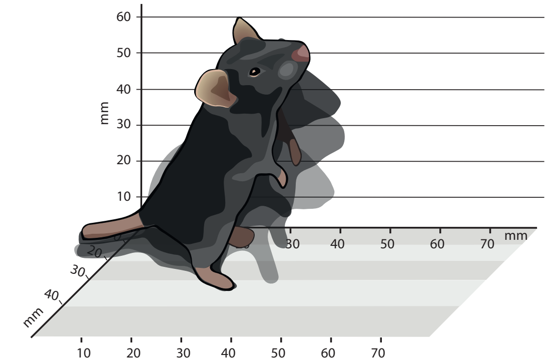
rear



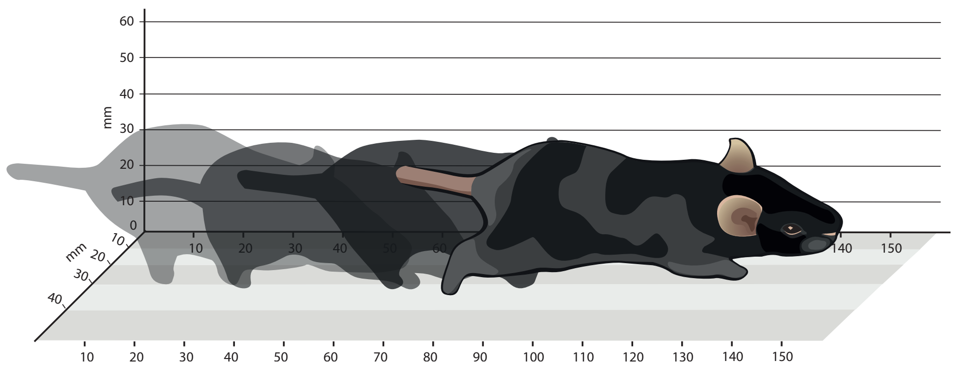
dart



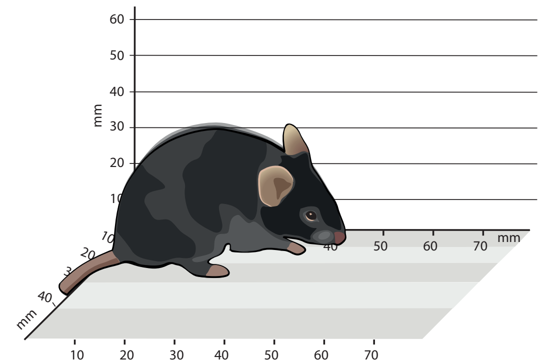
pause



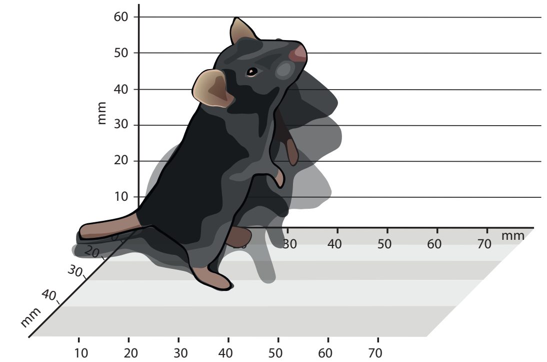
rear



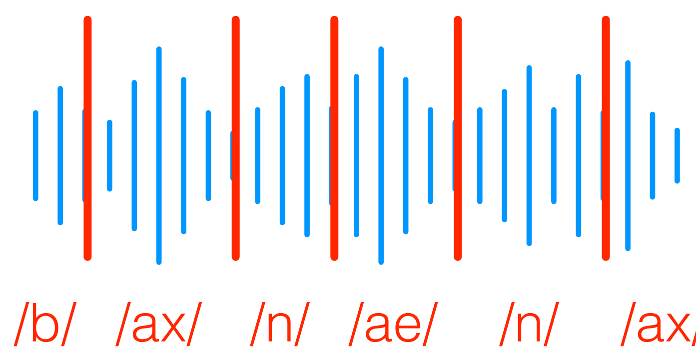
dart



pause

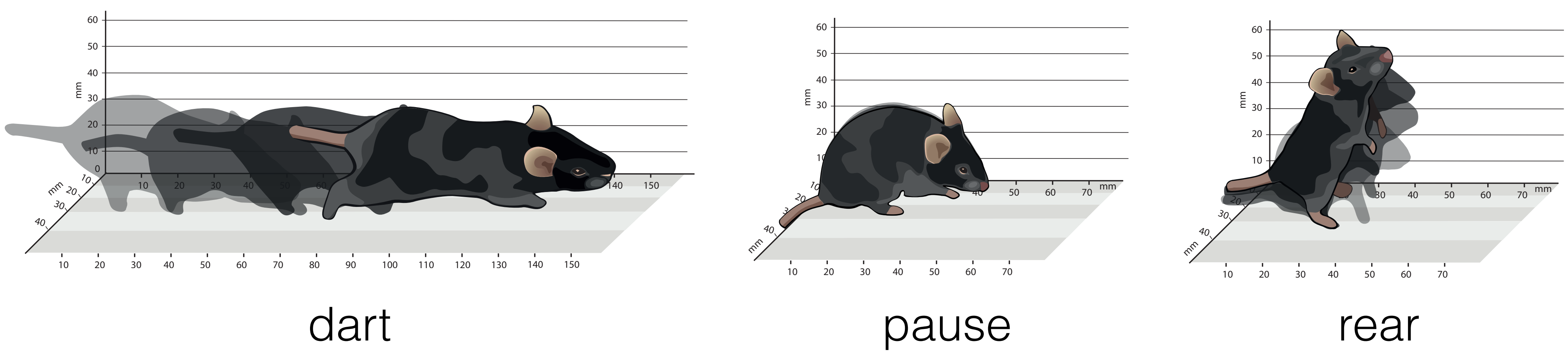


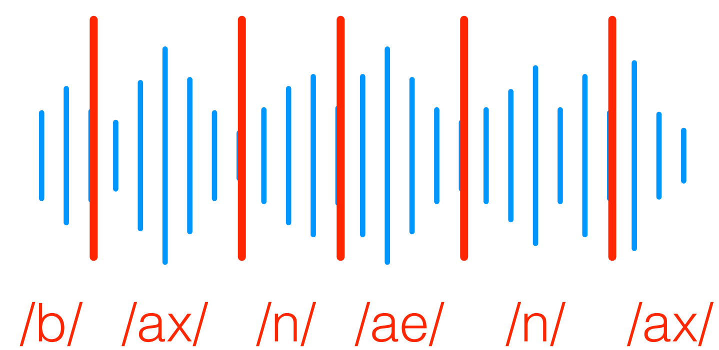
rear



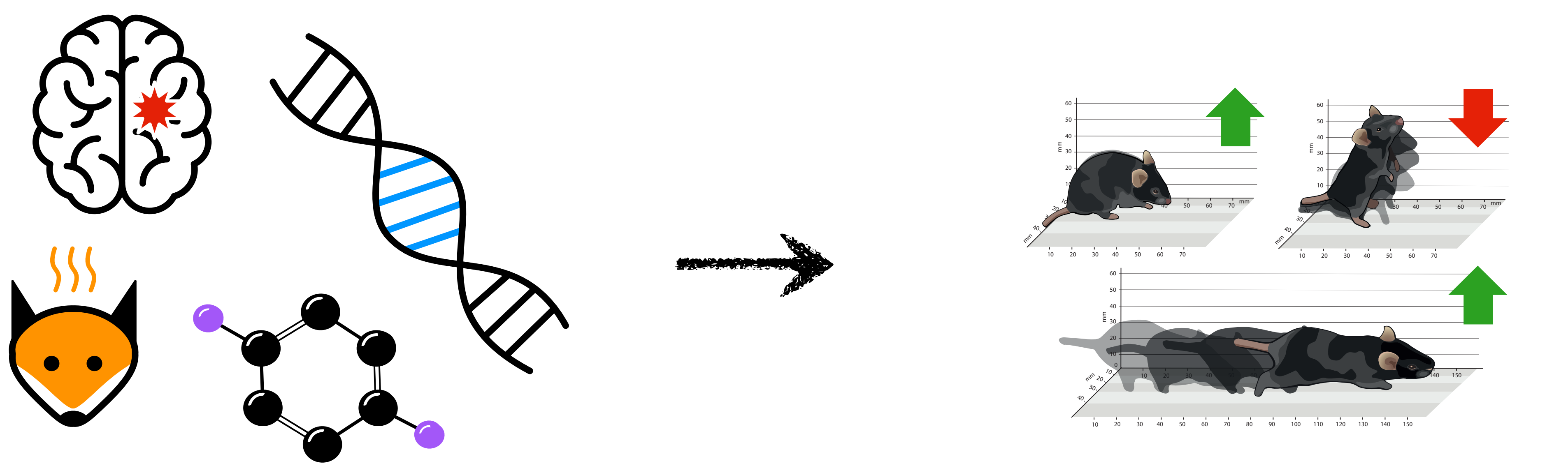
[1,2]

[1] Lee and Glass. A Nonparametric Bayesian Approach to Acoustic Model Discovery. ACL 2012.
[2] Lee. Discovering Linguistic Structures in Speech: Models and Applications. MIT Ph.D. Thesis 2014.

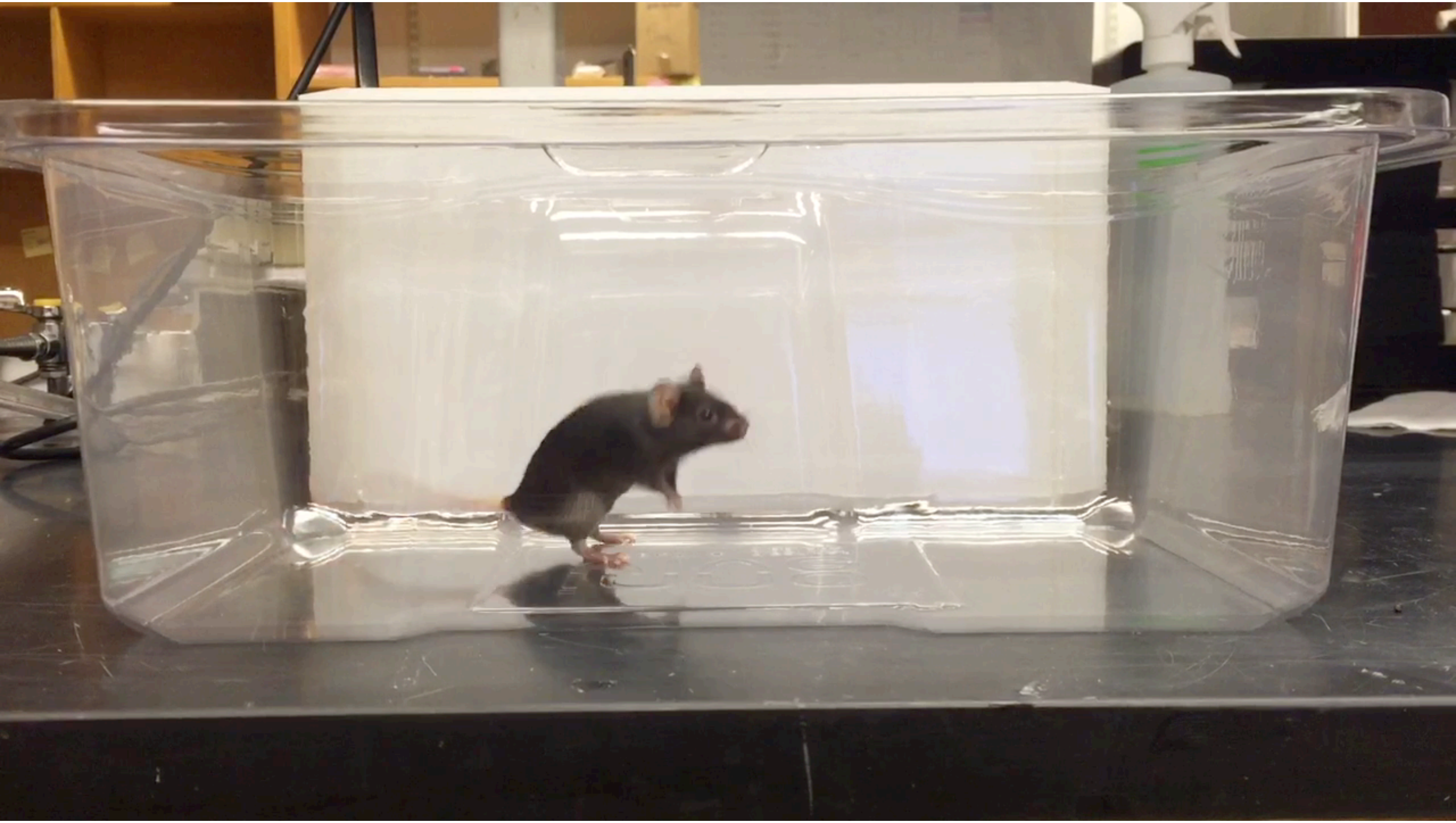


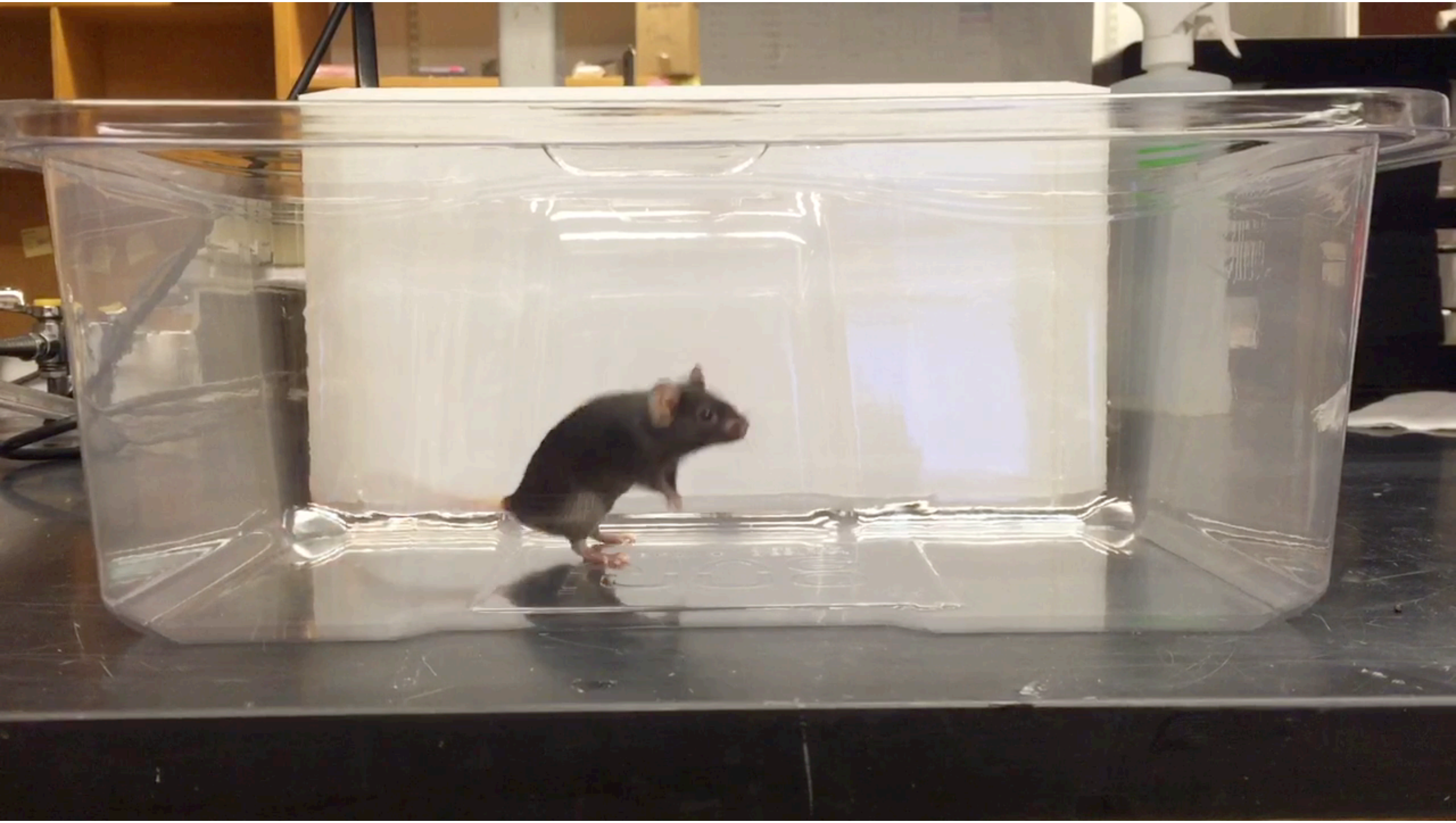


[1,2]



[1] Lee and Glass. A Nonparametric Bayesian Approach to Acoustic Model Discovery. ACL 2012.
[2] Lee. Discovering Linguistic Structures in Speech: Models and Applications. MIT Ph.D. Thesis 2014.





Alexander Wiltschko, **Matthew Johnson**, et al., Neuron 2015.

Alexander Wiltschko, **Matthew Johnson**, et al., Neuron 2015.

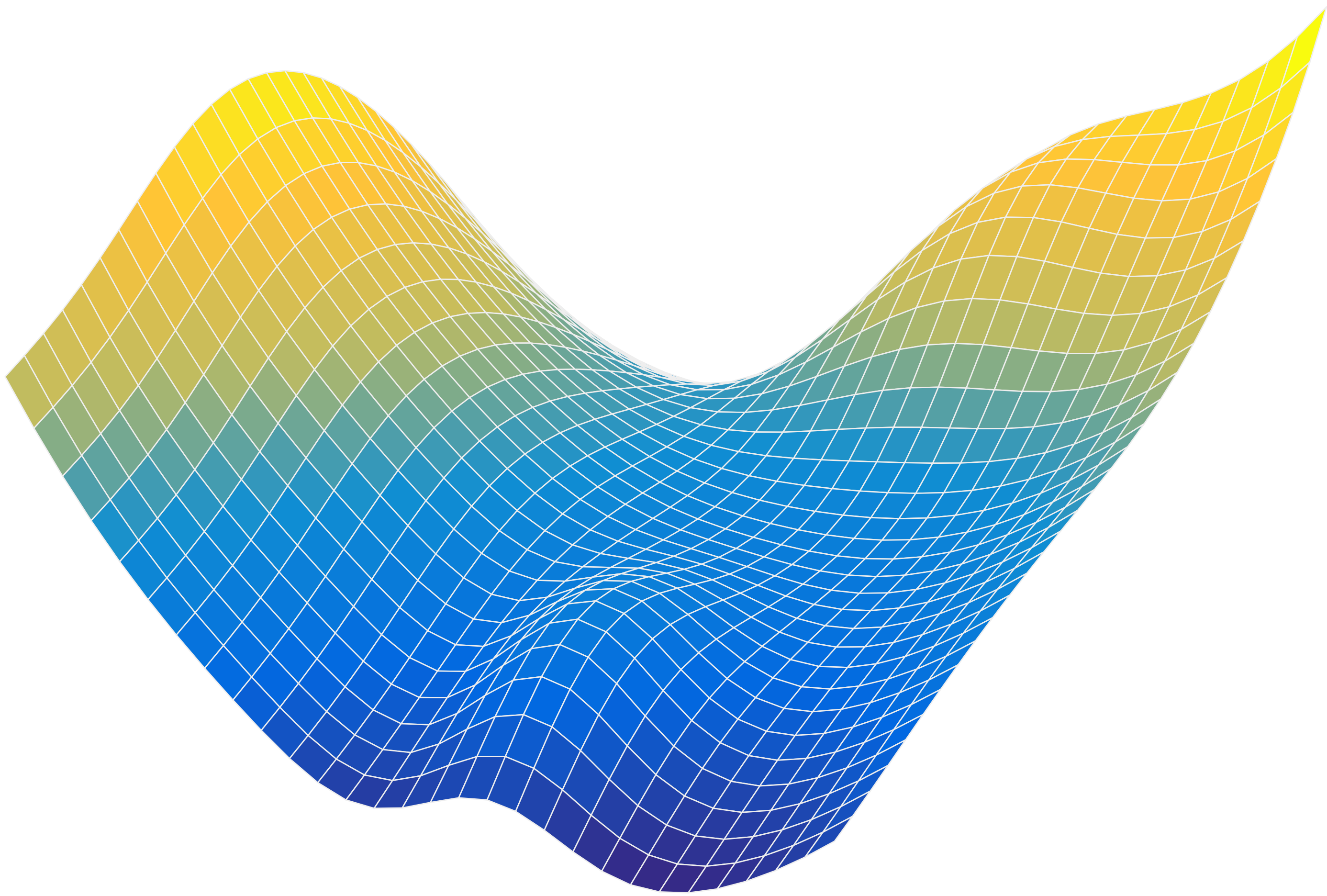
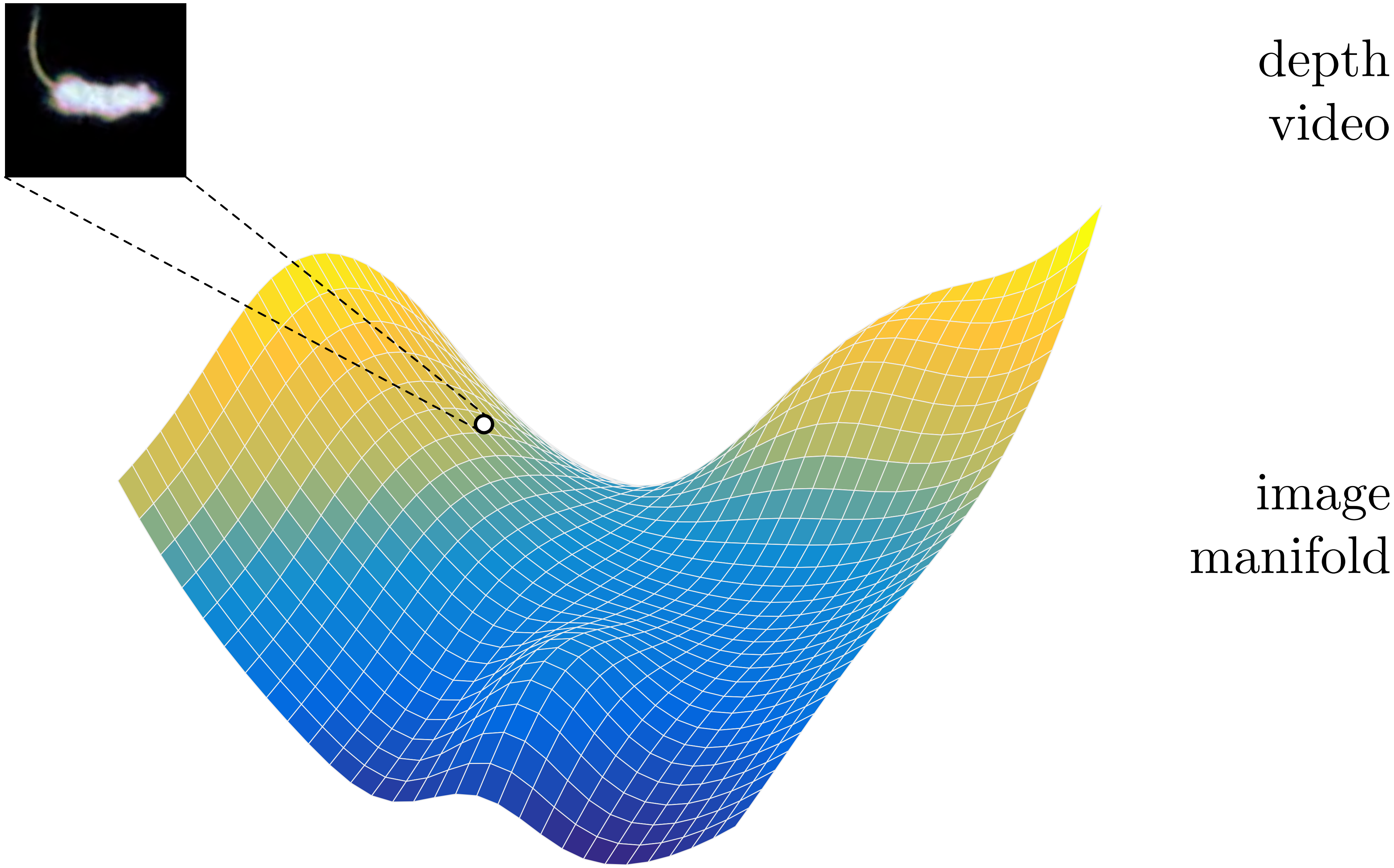
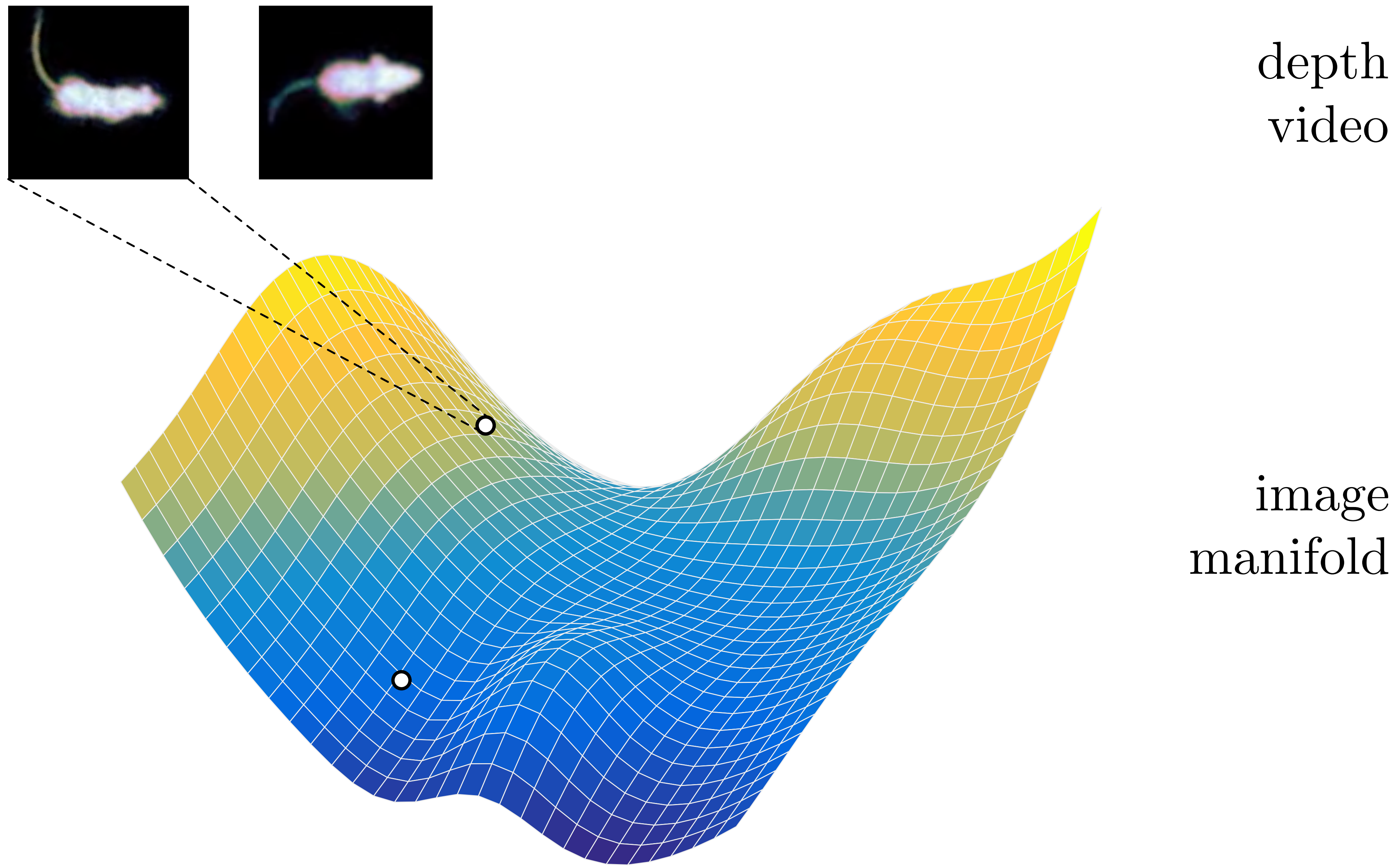
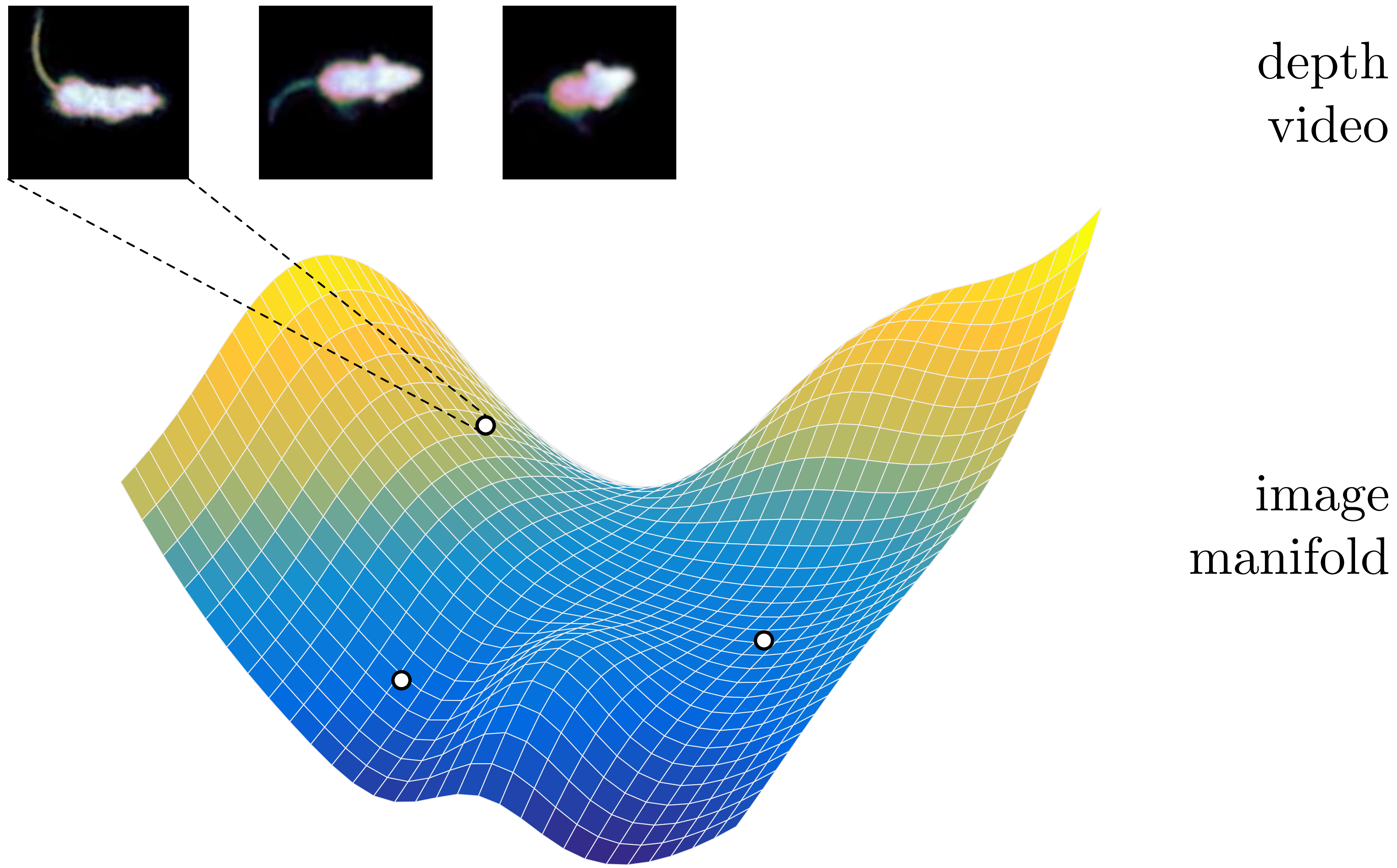
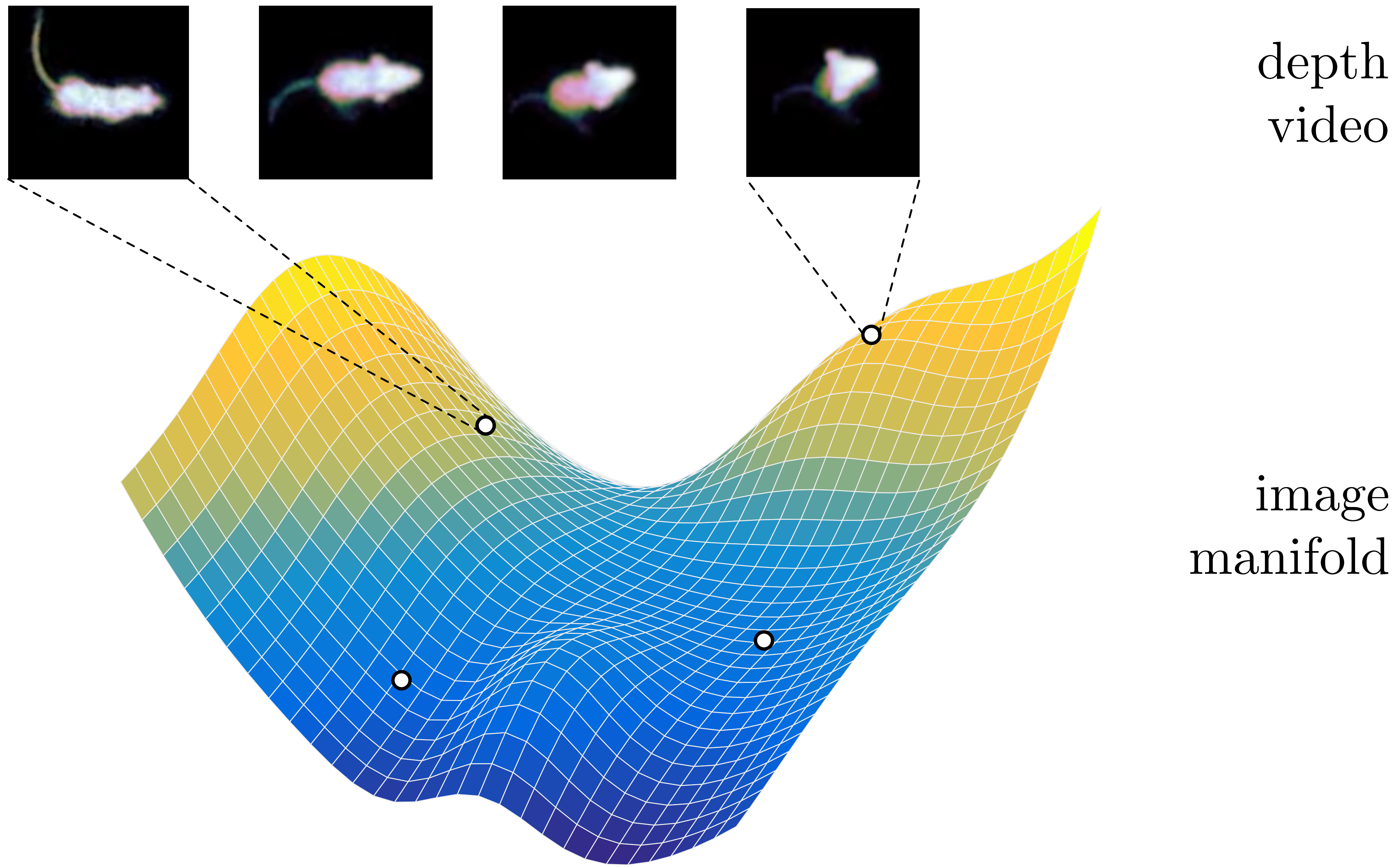


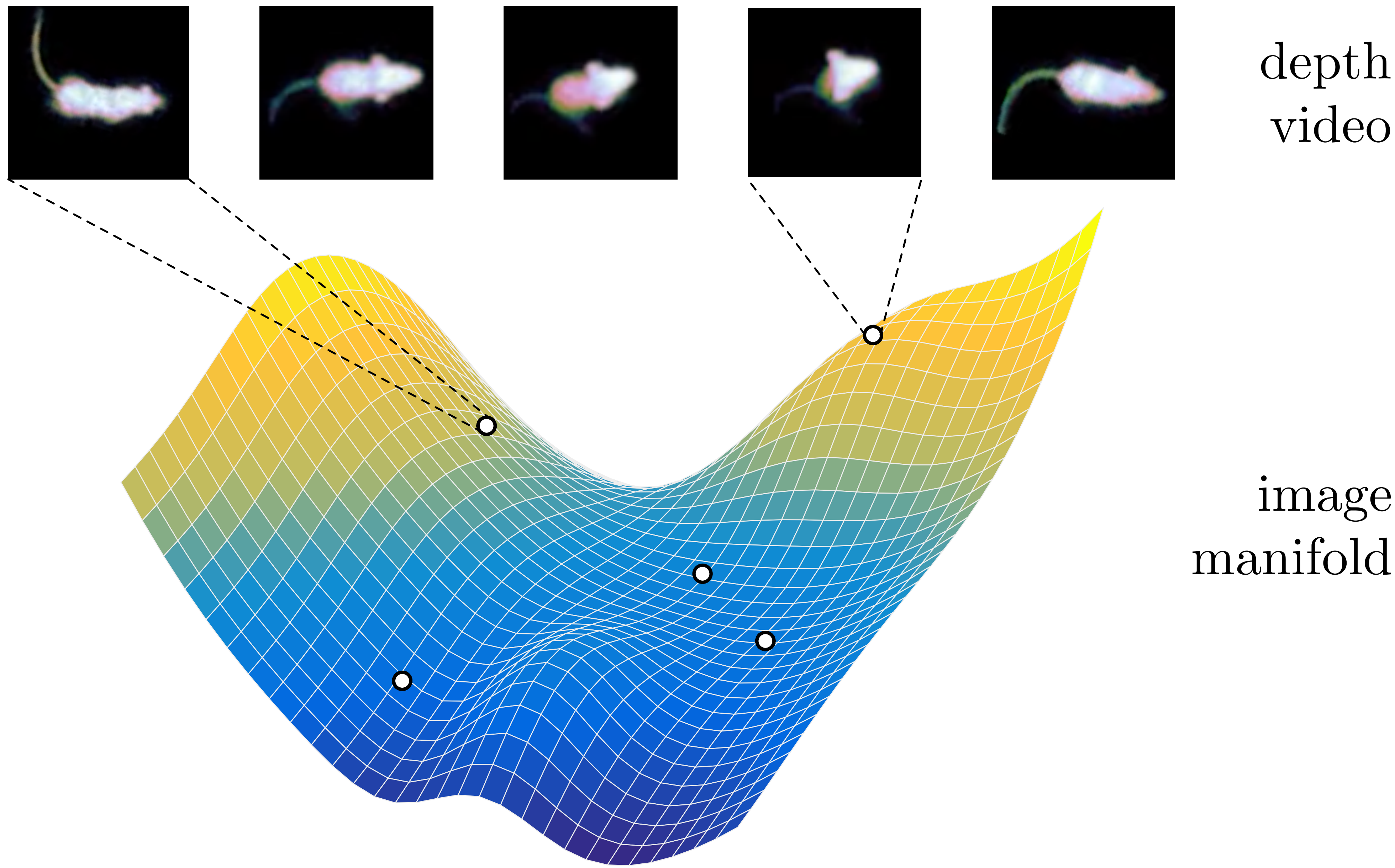
image
manifold

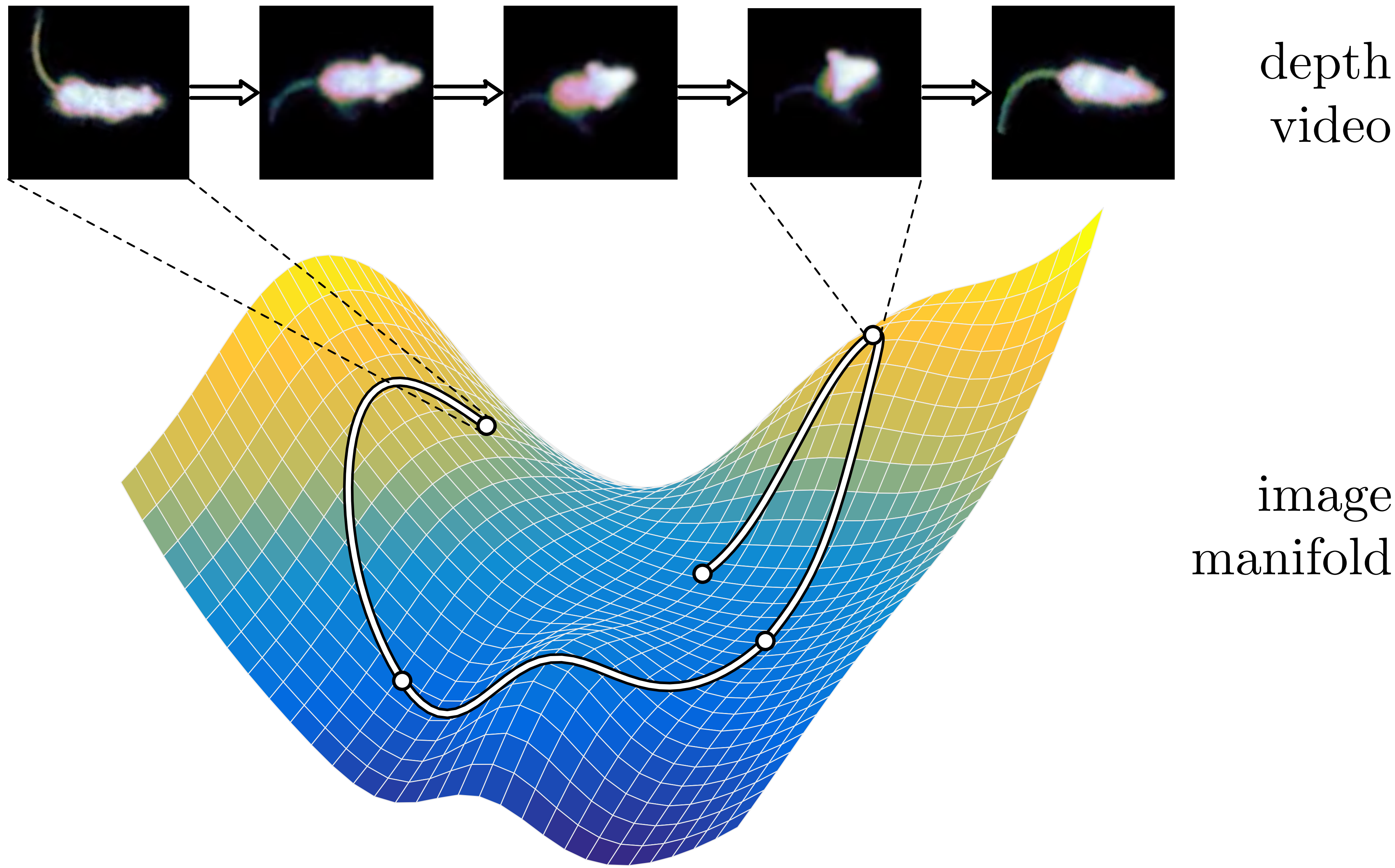

depth
video
image
manifold

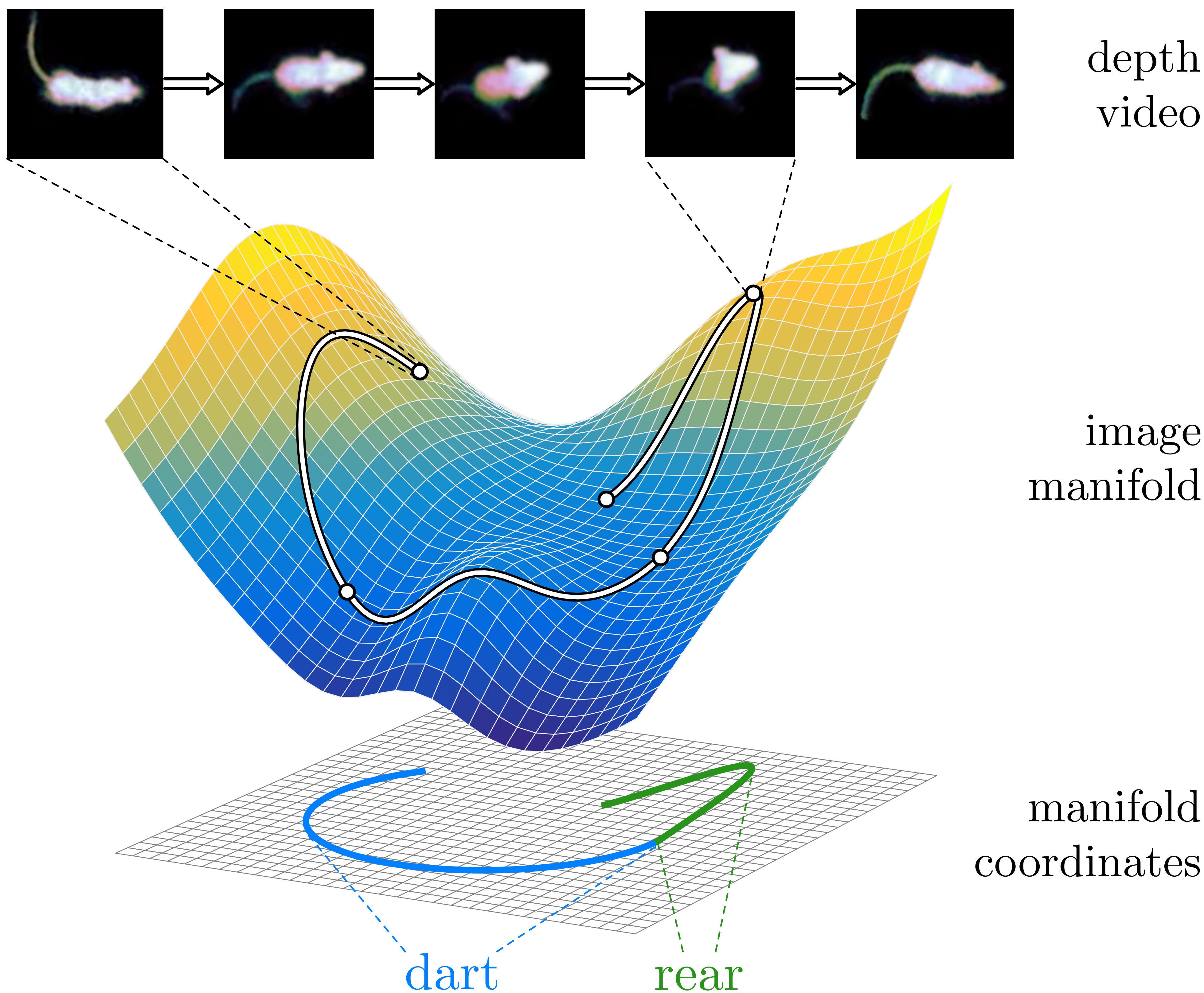

depth
video
image
manifold


depth
video
image
manifold


depth
video
image
manifold


depth
video
image
manifold


depth
video
image
manifold


depth
video
image
manifold
manifold
coordinates
dart
rear

Recurrent neural networks? [1,2,3]

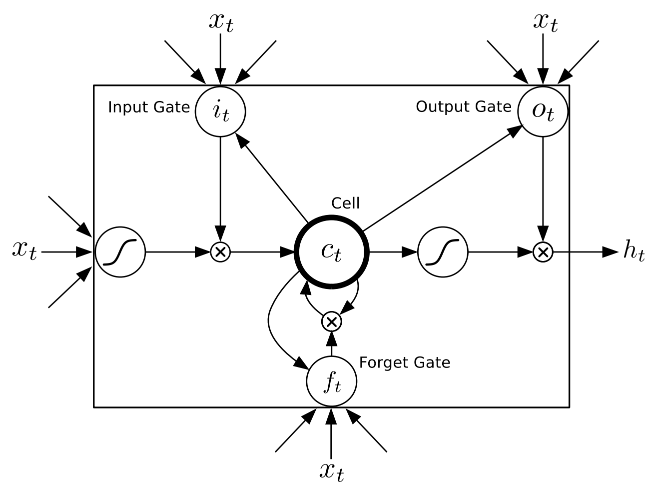


Figure 1. LSTM unit

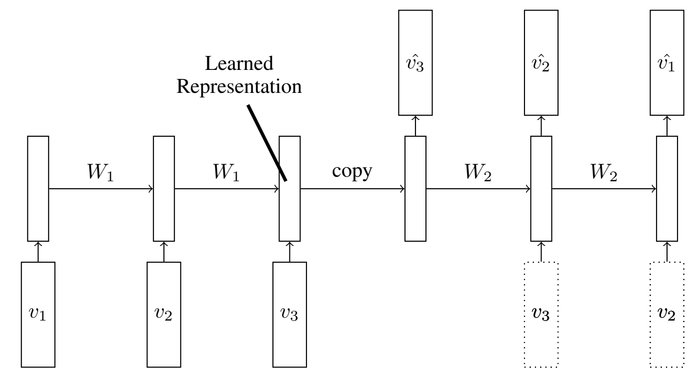


Figure 2. LSTM Autoencoder Model

[1] Srivastava, Mansimov, Salakhutdinov. Unsupervised learning of video representations using LSTMs. ICML 2015.
[2] Ranzato, MarcAurelio, et al. Video (language) modeling: a baseline for generative models of natural videos. Preprint 2015.
[3] Sutskever, Hinton, and Taylor. The Recurrent Temporal Restricted Boltzmann Machine. NIPS 2008.

Recurrent neural networks? [1,2,3]

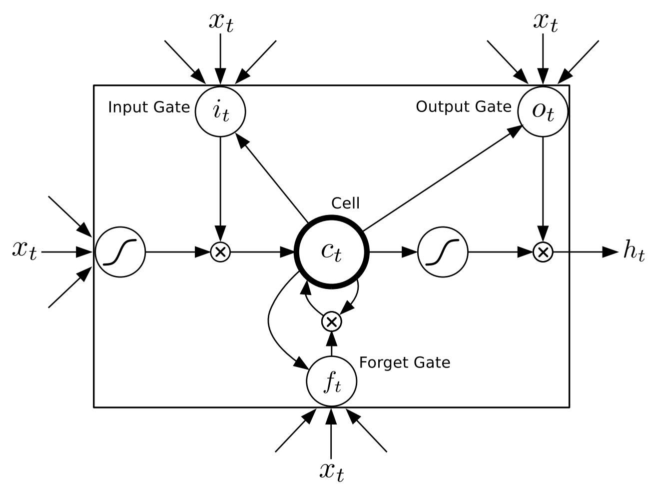


Figure 1. LSTM unit

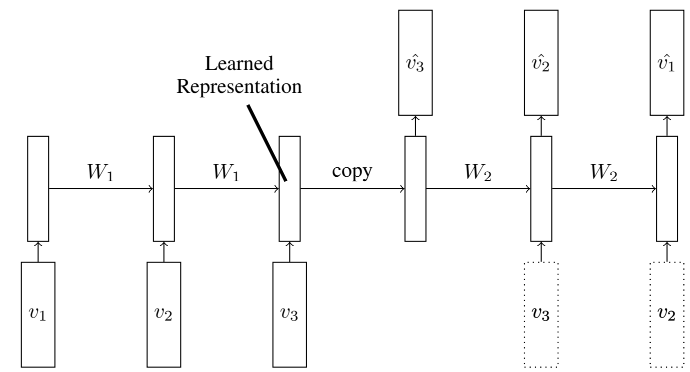
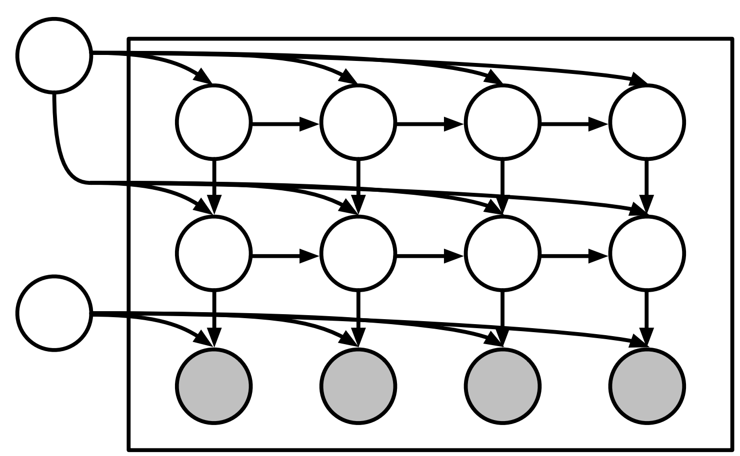
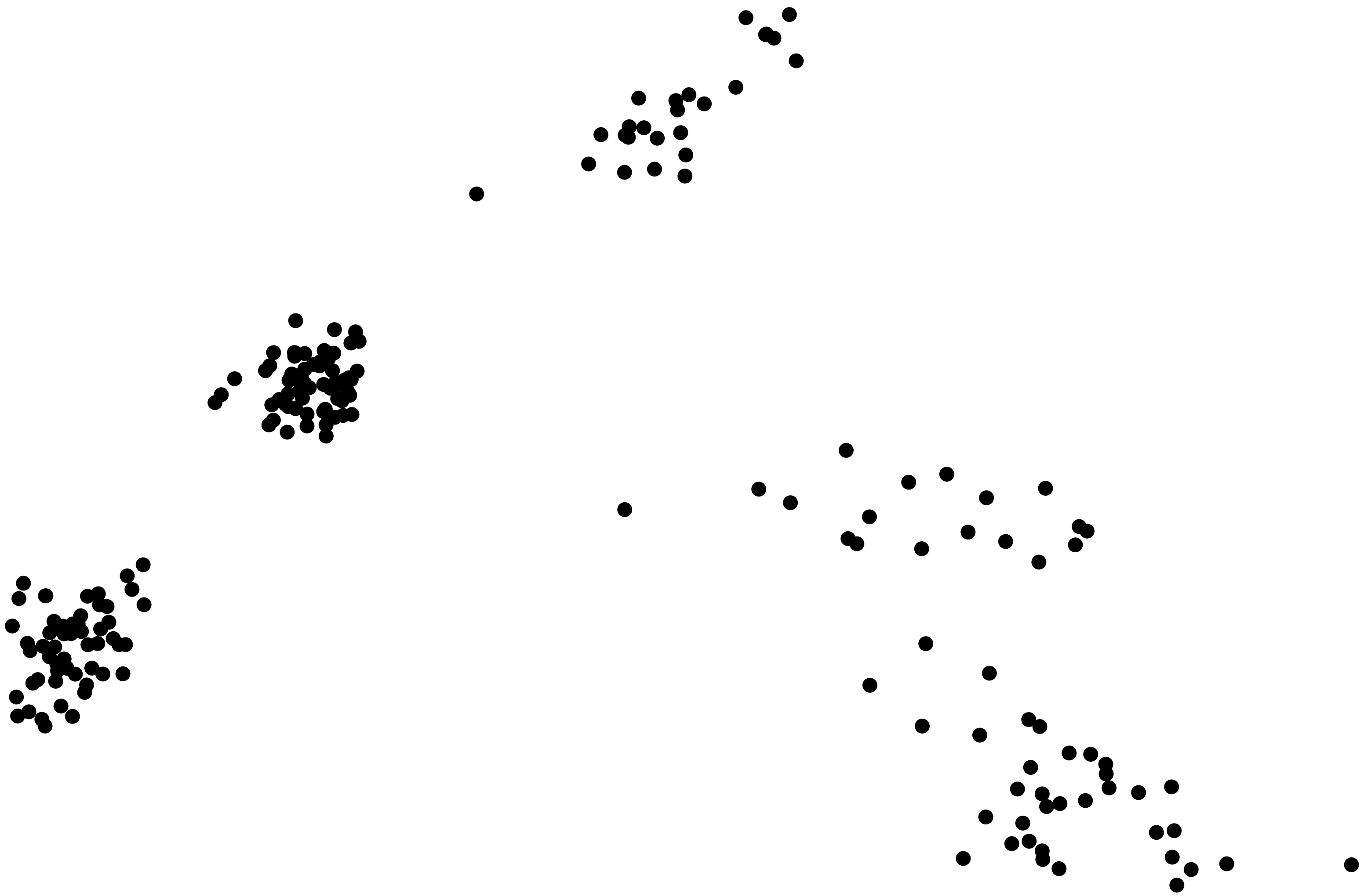


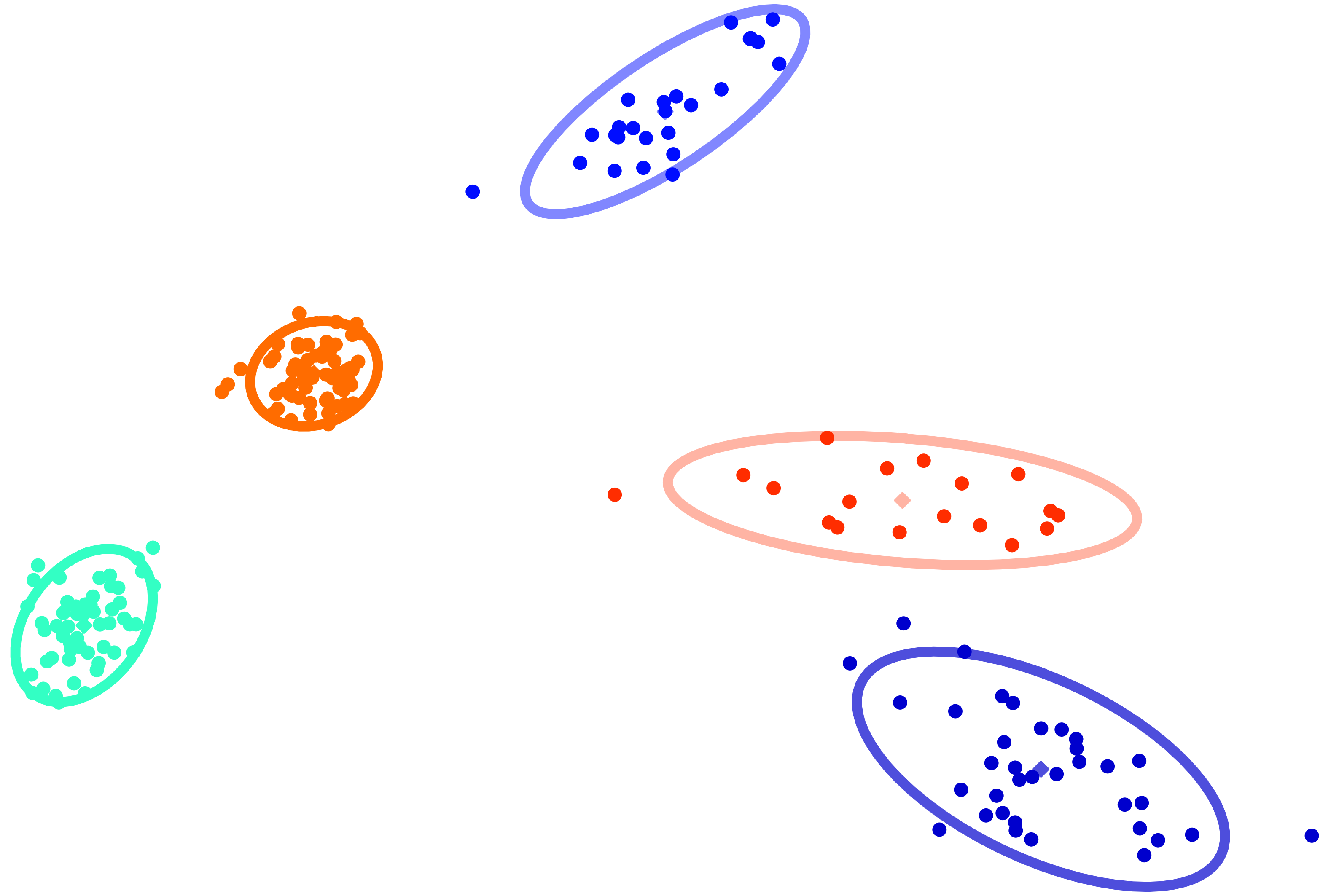
Figure 2. LSTM Autoencoder Model

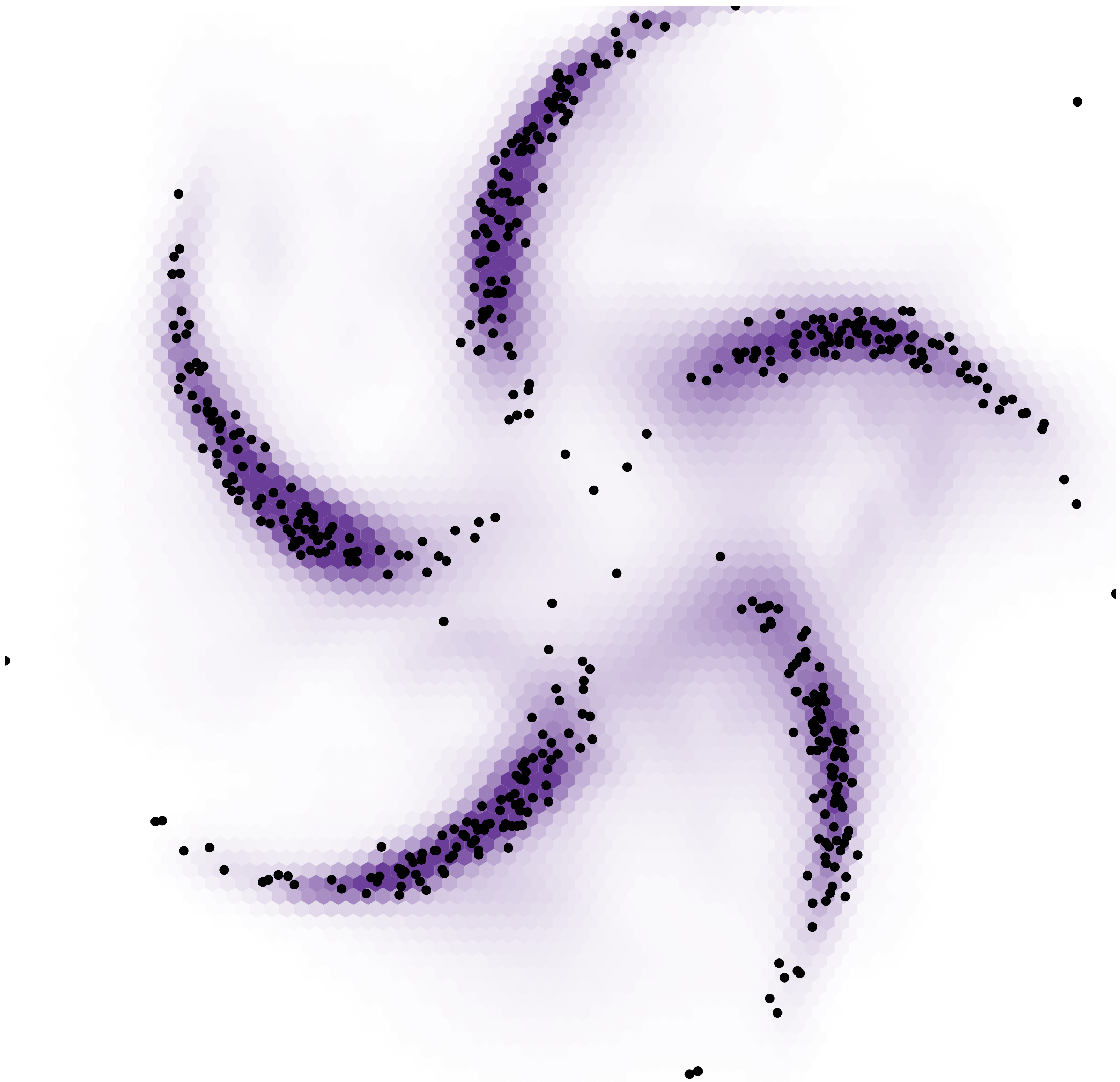
Probabilistic graphical models? [4,5,6]

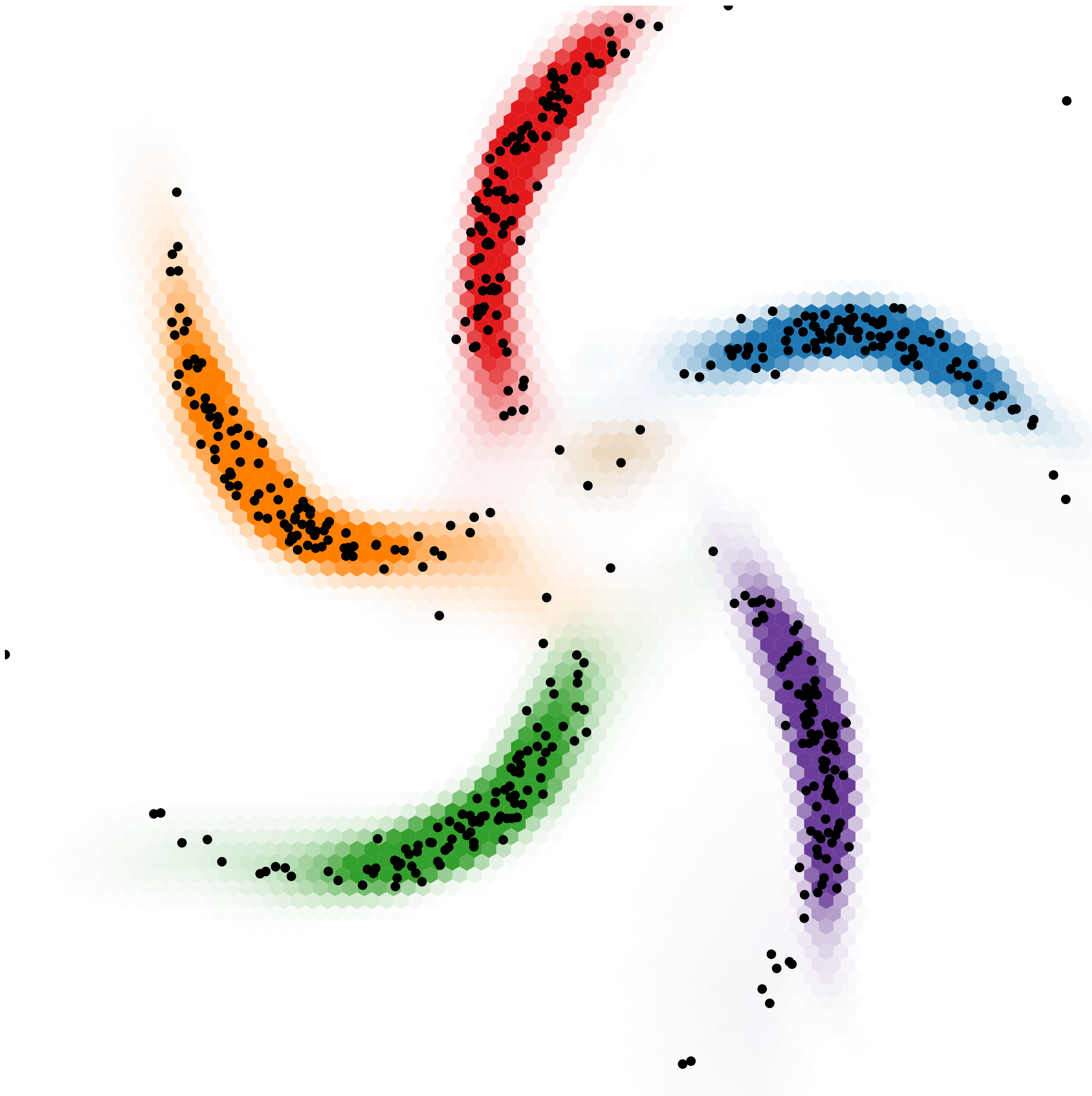


[1] Srivastava, Mansimov, Salakhutdinov. Unsupervised learning of video representations using LSTMs. ICML 2015.
[2] Ranzato, MarcAurelio, et al. Video (language) modeling: a baseline for generative models of natural videos. Preprint 2015.
[3] Sutskever, Hinton, and Taylor. The Recurrent Temporal Restricted Boltzmann Machine. NIPS 2008.
[4] Fox, Sudderth, Jordan, Willsky. Bayesian nonparametric inference of switching dynamic linear models. IEEE TSP 2011.
[5] **Johnson** and Willsky. Bayesian nonparametric hidden semi-Markov models. JMLR 2013.
[6] Murphy. Machine learning: a probabilistic perspective. MIT Press 2012.

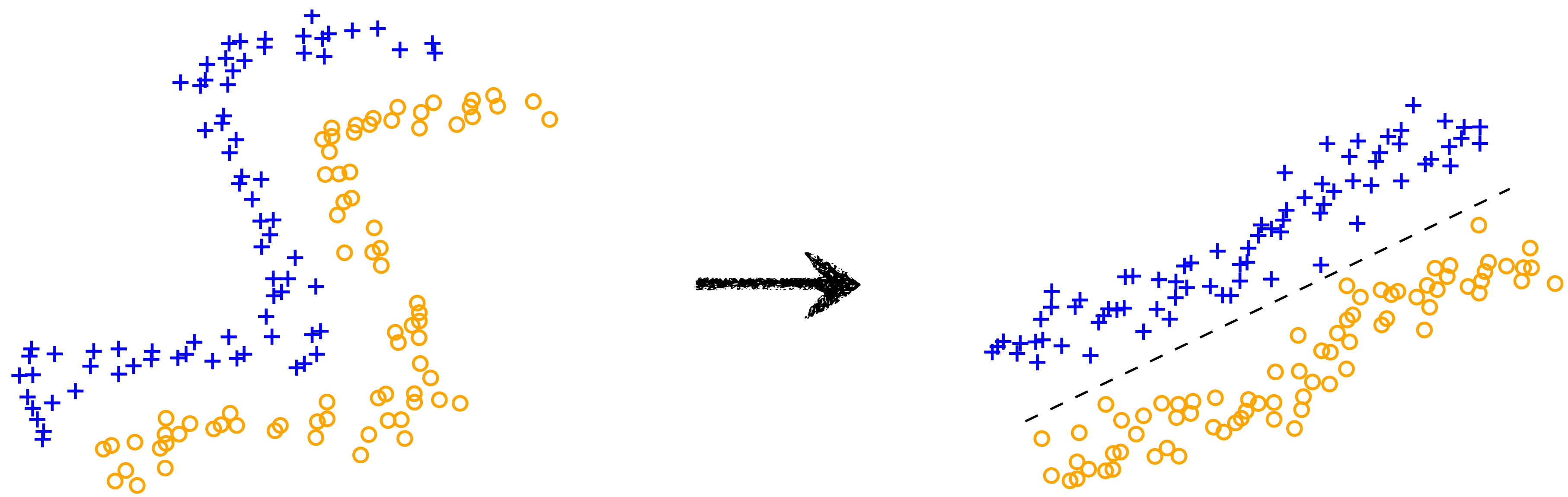




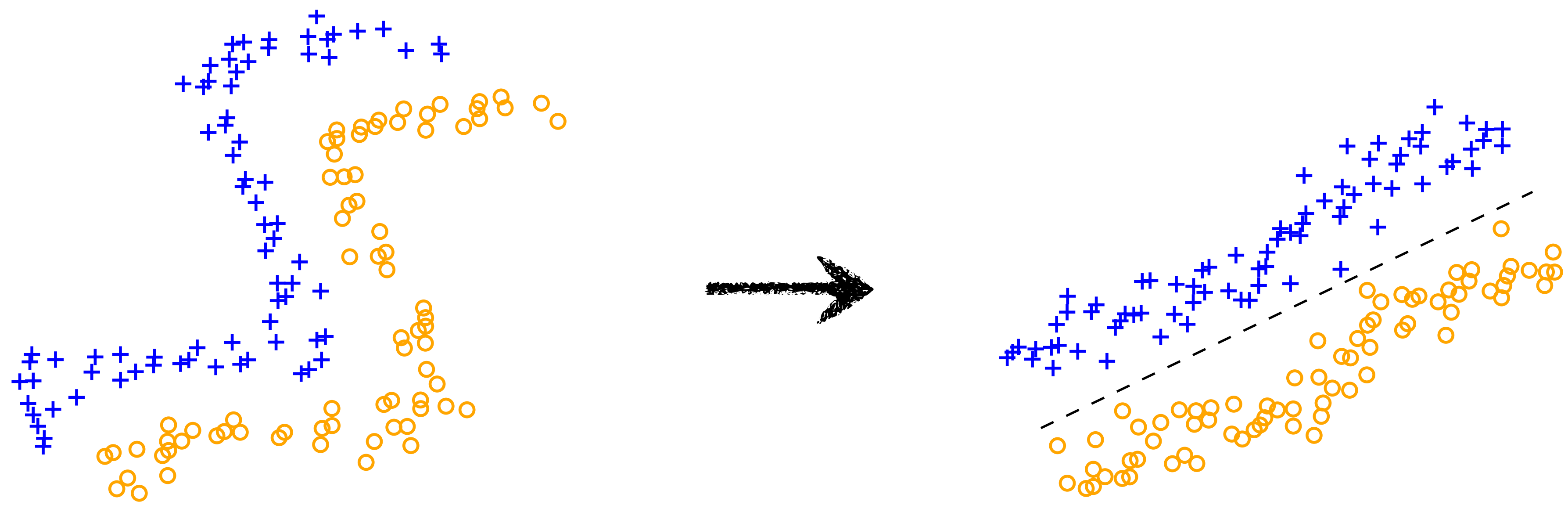




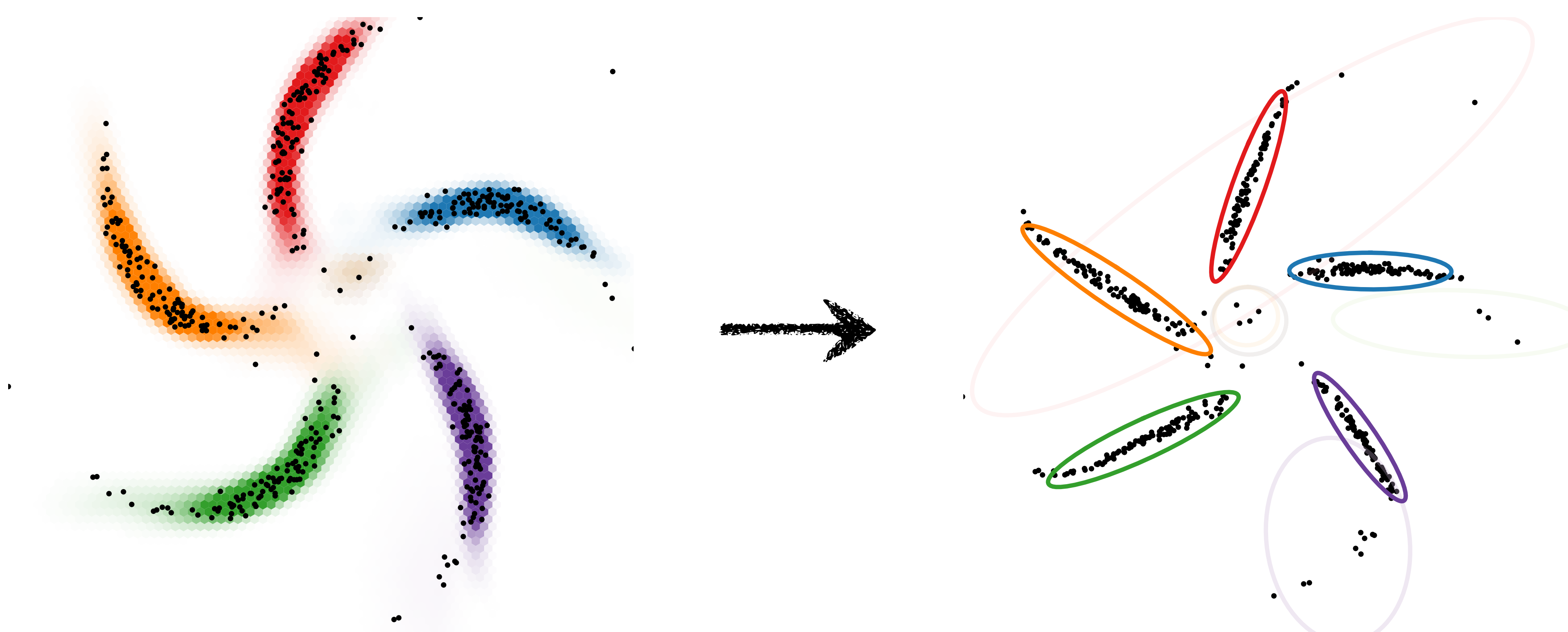
supervised
learning



supervised
learning



unsupervised
learning



Probabilistic graphical models

Deep learning

Probabilistic graphical models

+ structured representations

Deep learning

Probabilistic graphical models

+ structured representations

+ priors and uncertainty

Deep learning

Probabilistic graphical models

Deep learning

+ structured representations

+ priors and uncertainty

– rigid assumptions may not fit

Probabilistic graphical models

+ structured representations

+ priors and uncertainty

– rigid assumptions may not fit

– feature engineering

Deep learning

Probabilistic graphical models	Deep learning
+ structured representations
+ priors and uncertainty
– rigid assumptions may not fit
– feature engineering
+ arbitrary inference queries

Probabilistic graphical models

\+ structured representations

\+ priors and uncertainty

– rigid assumptions may not fit

– feature engineering

\+ arbitrary inference queries

\+ data and computational efficiency
within rigid model classes

Deep learning

Probabilistic graphical models

+ structured representations

+ priors and uncertainty

– rigid assumptions may not fit

– feature engineering

+ arbitrary inference queries

+ data and computational efficiency within rigid model classes

– more flexible models can require slow top-down inference

Deep learning

Probabilistic graphical models

+ structured representations

+ priors and uncertainty

– rigid assumptions may not fit

– feature engineering

+ arbitrary inference queries

+ data and computational efficiency within rigid model classes

– more flexible models can require slow top-down inference

Deep learning

– neural net “goo”

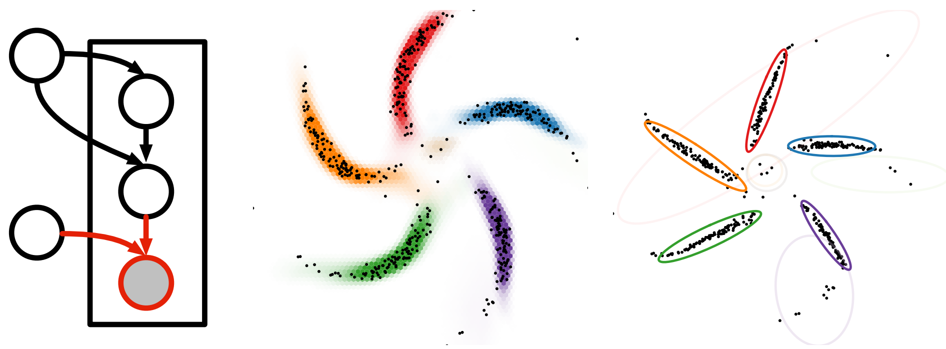
– difficult parameterization

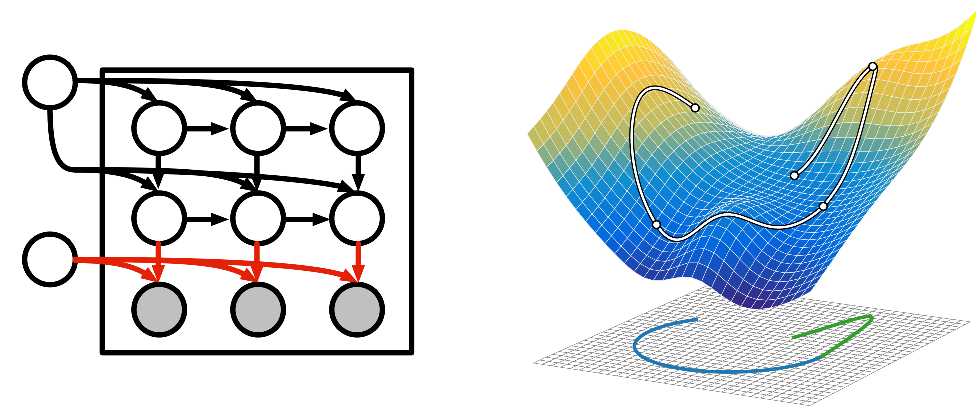
+ flexible, high capacity

+ feature learning

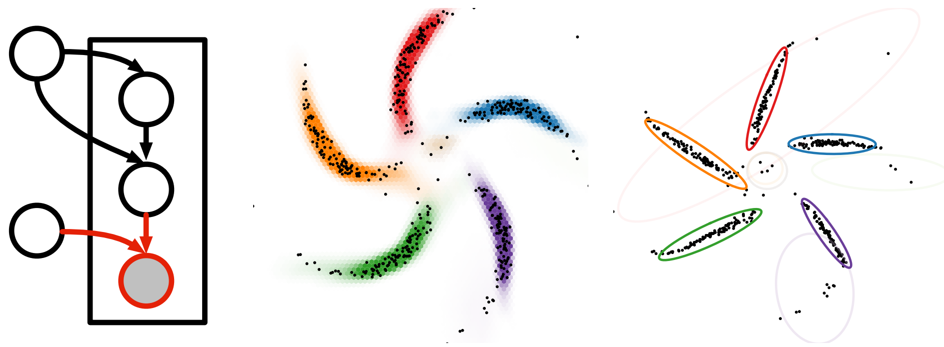
Probabilistic graphical models	Deep learning
+ structured representations	– neural net “goo”
+ priors and uncertainty	– difficult parameterization
– rigid assumptions may not fit	+ flexible, high capacity
– feature engineering	+ feature learning
+ arbitrary inference queries	– limited inference queries
+ data and computational efficiency within rigid model classes	– data- and compute-hungry
– more flexible models can require slow top-down inference	+ recognition networks learn how to do inference

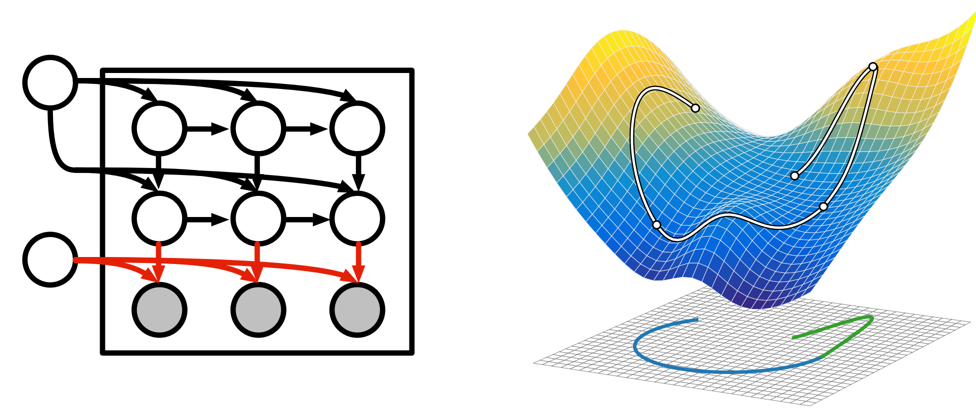
Modeling idea: graphical models on latent variables,
neural network models for observations



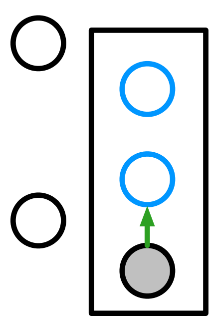


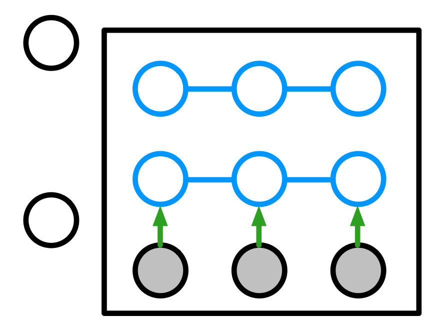
Modeling idea: graphical models on latent variables,
neural network models for observations



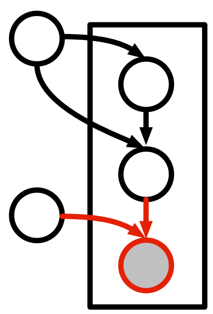


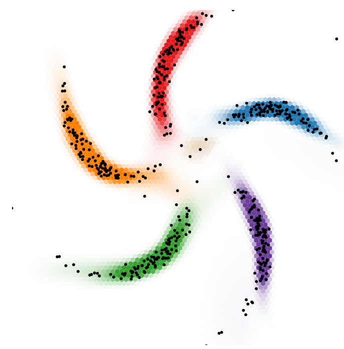
Inference: recognition networks output conjugate potentials,
then apply fast graphical model inference

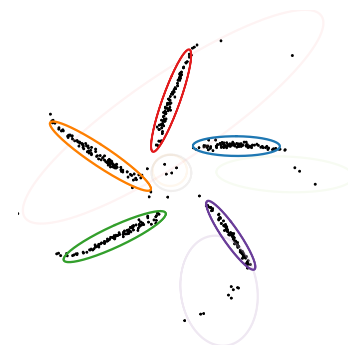


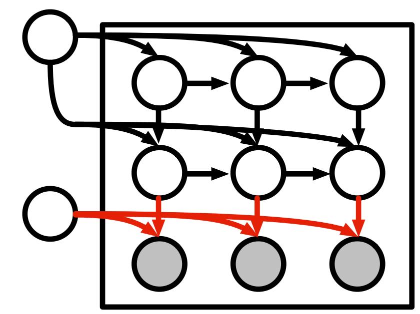


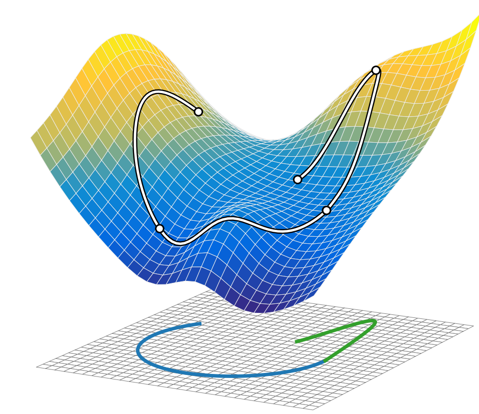
Modeling idea: graphical models on latent variables,
neural network models for observations



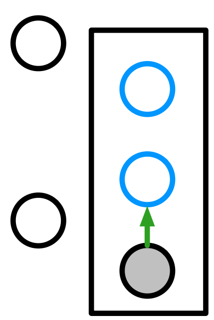


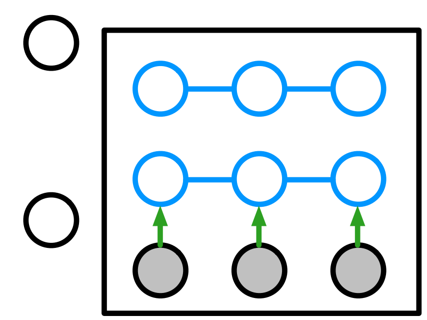




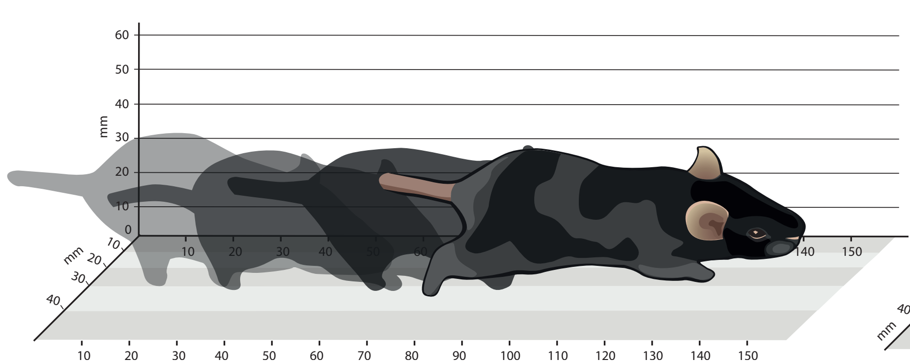


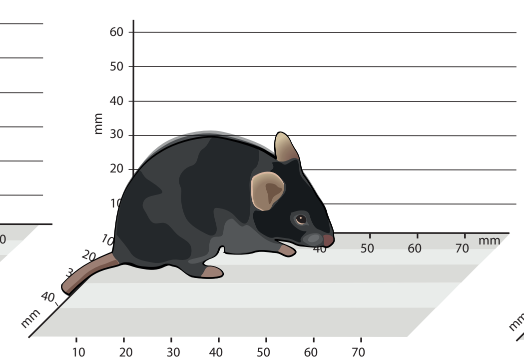
Inference: recognition networks output conjugate potentials,
then apply fast graphical model inference

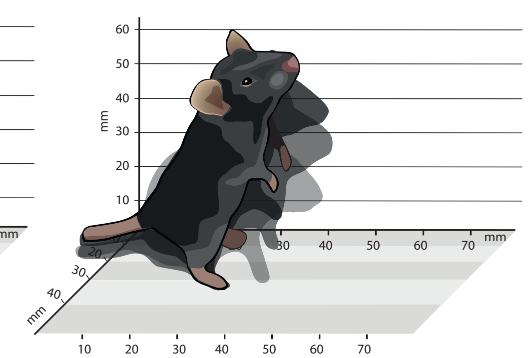




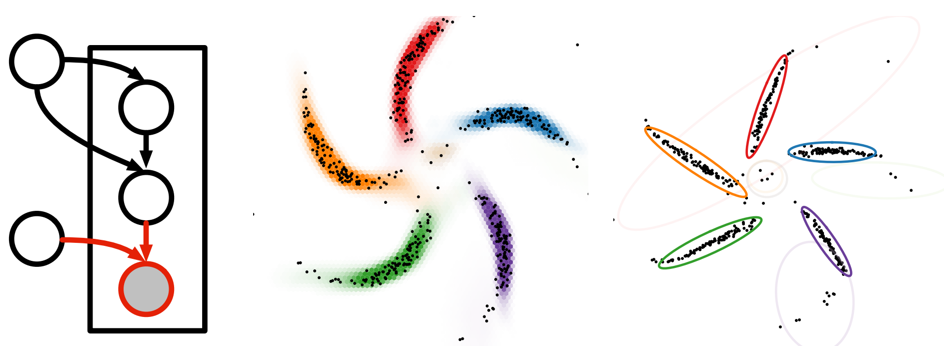
Application: learn syllable representation of behavior from video

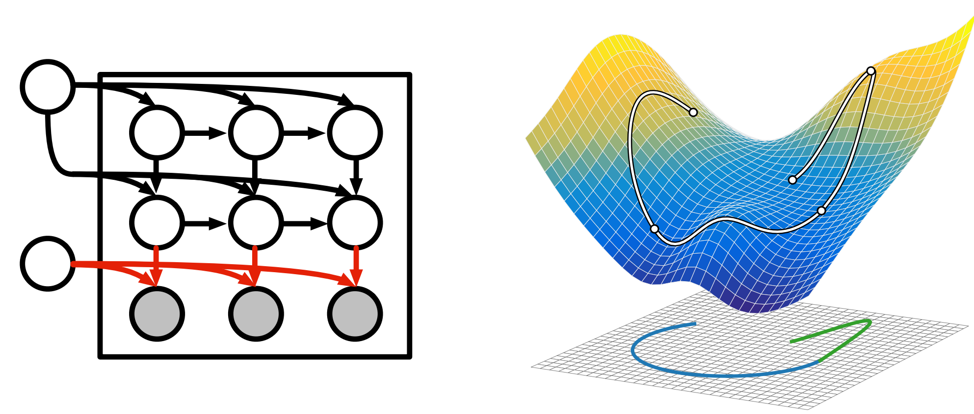


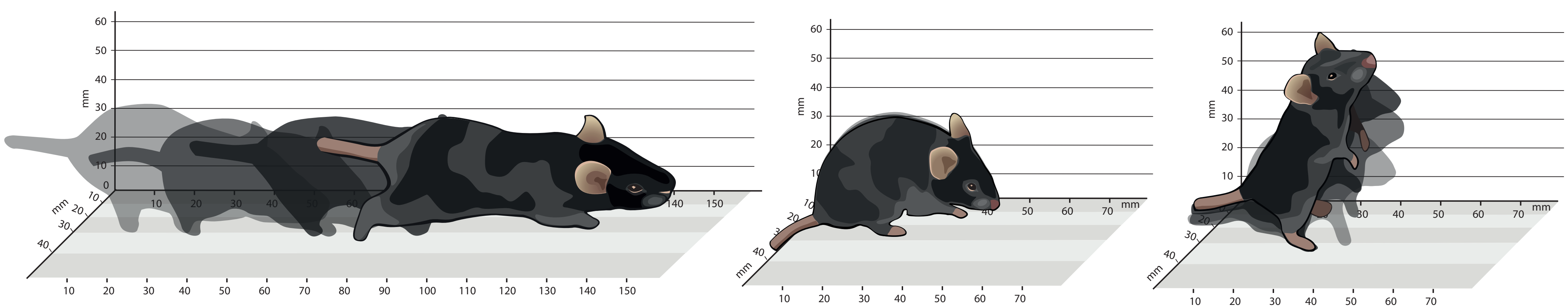




Modeling idea: graphical models on latent variables,
neural network models for observations



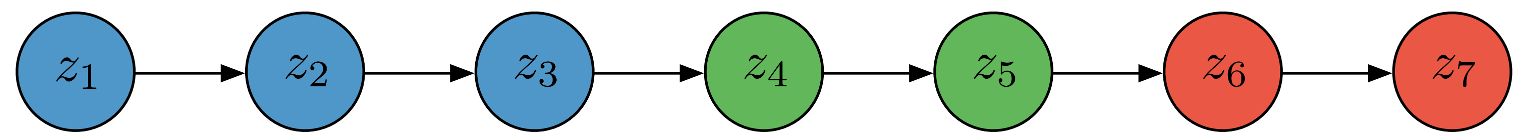


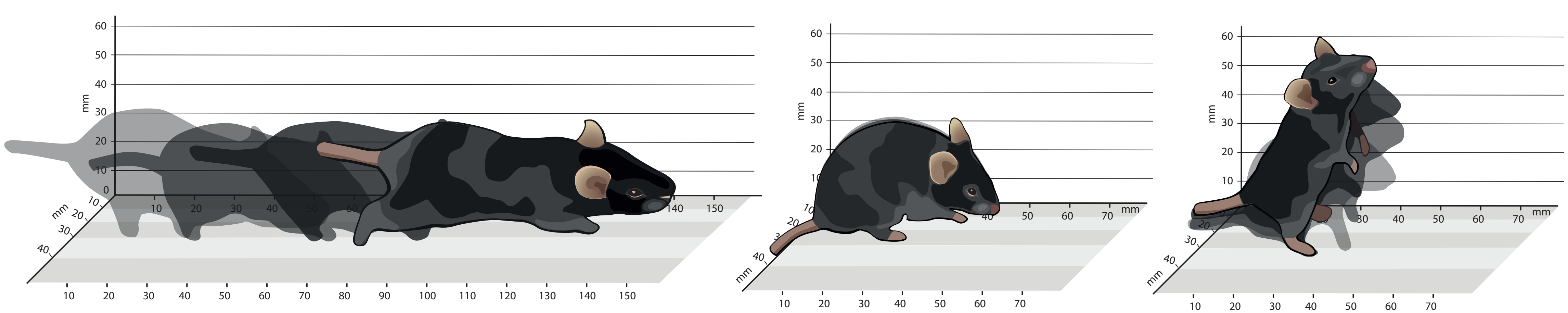


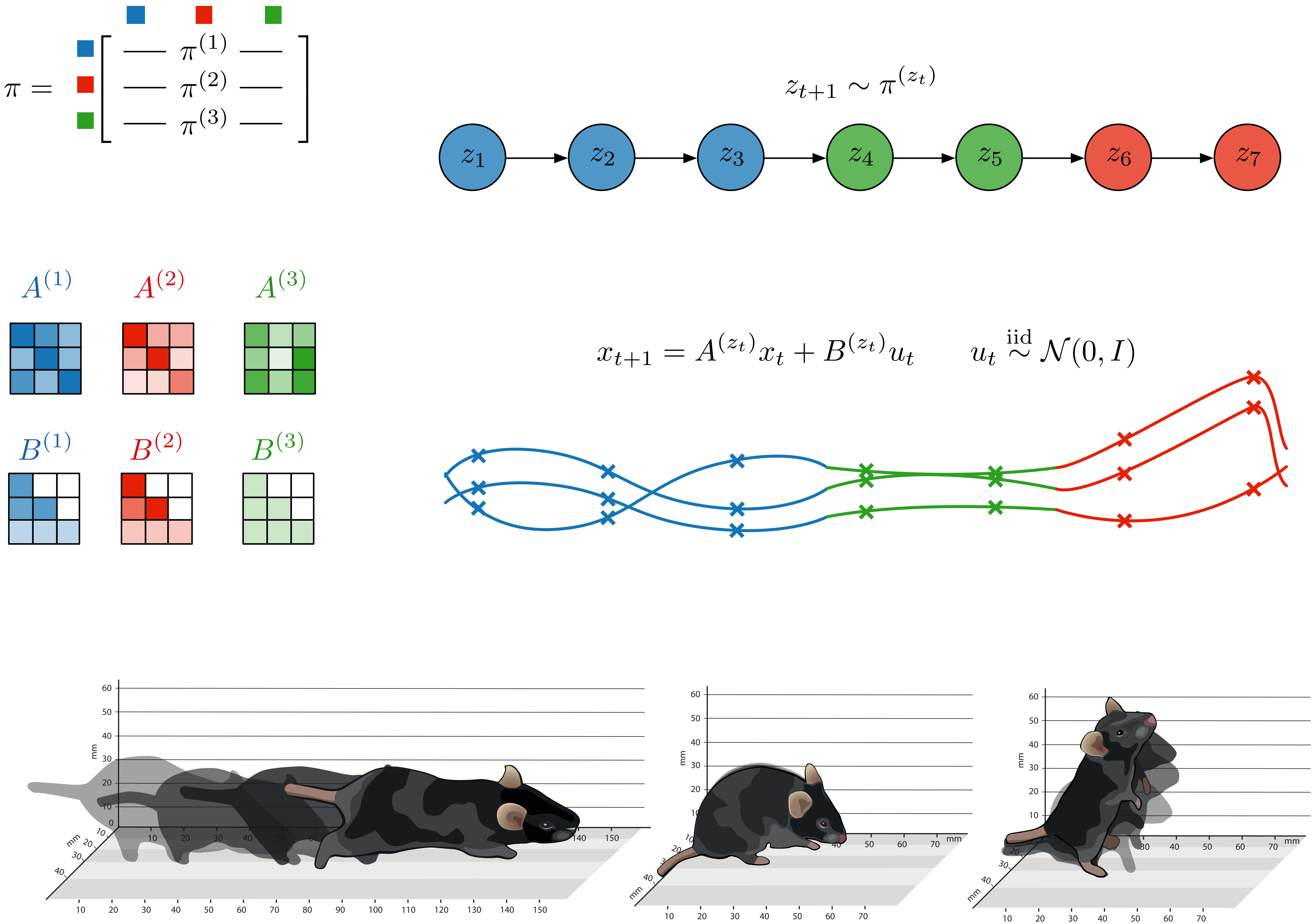
mm
60
50
40
30
20
10
0
10 20 30 40 50 60 140 150
mm 10 20 30 40
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
mm
60
50
40
30
20
40 50 60 70 mm
mm 40
10 20 30 40 50 60 70
mm
60
50
40
30
20
10
30 40 50 60 70 mm
mm 40 30 20
10 20 30 40 50 60 70

$$\pi = \begin{bmatrix} \text{---} & \pi^{(1)} & \text{---} \\ \text{---} & \pi^{(2)} & \text{---} \\ \text{---} & \pi^{(3)} & \text{---} \end{bmatrix}$$

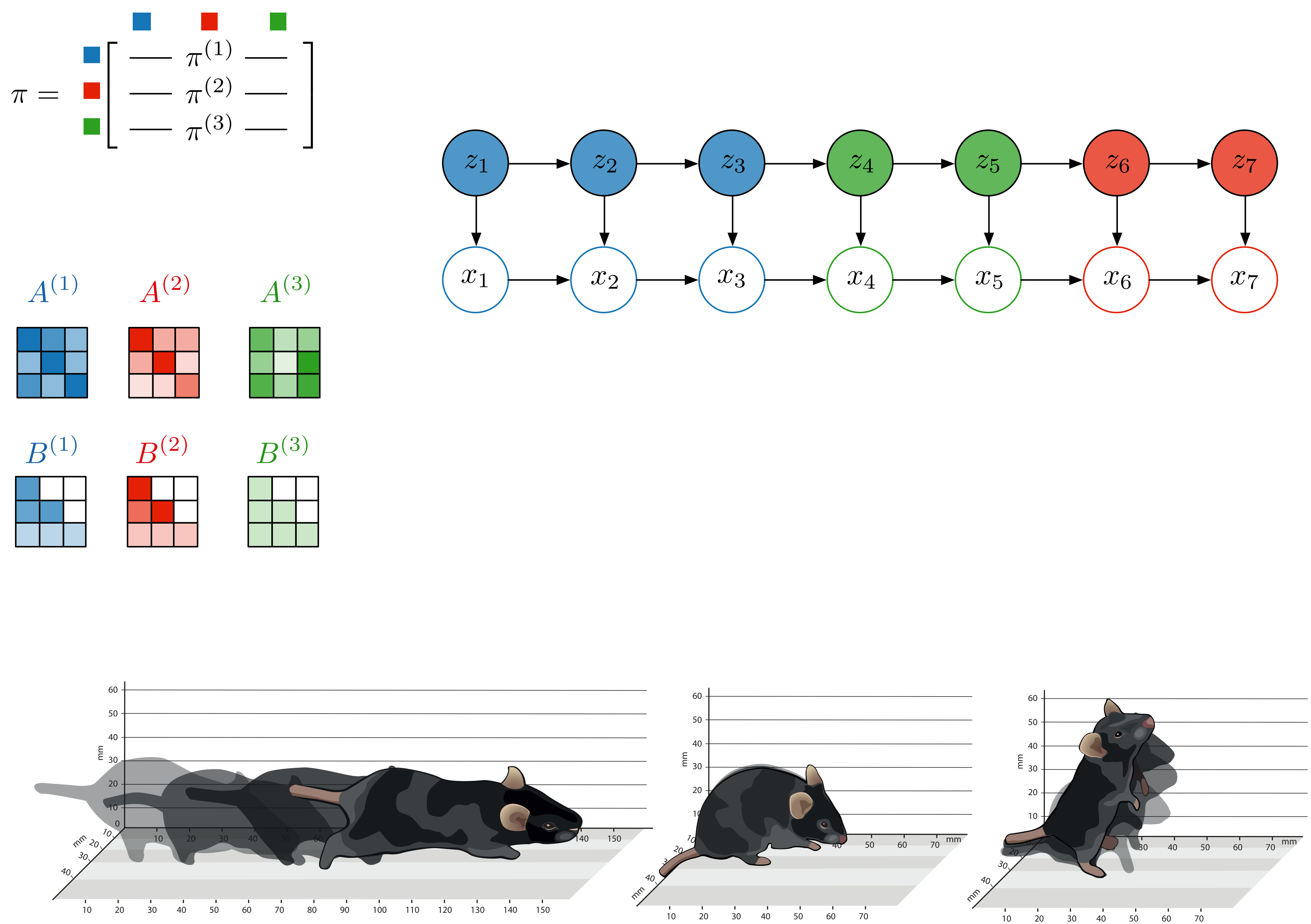
$$z_{t+1} \sim \pi^{(z_t)}$$



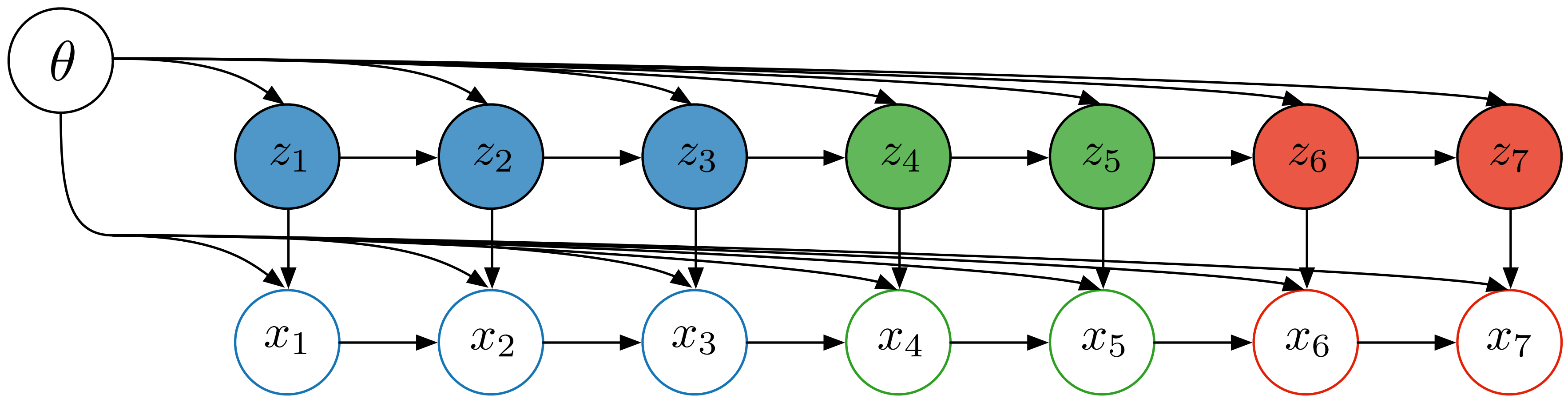


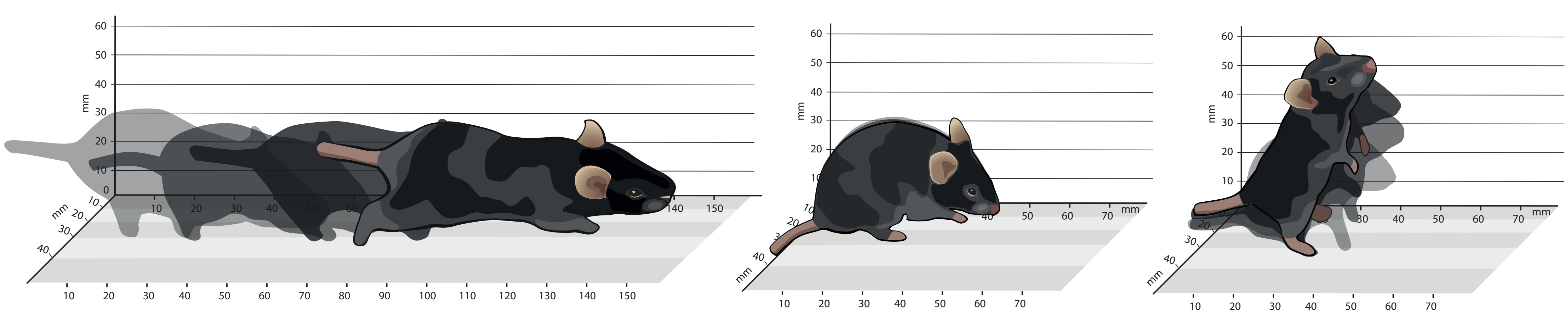


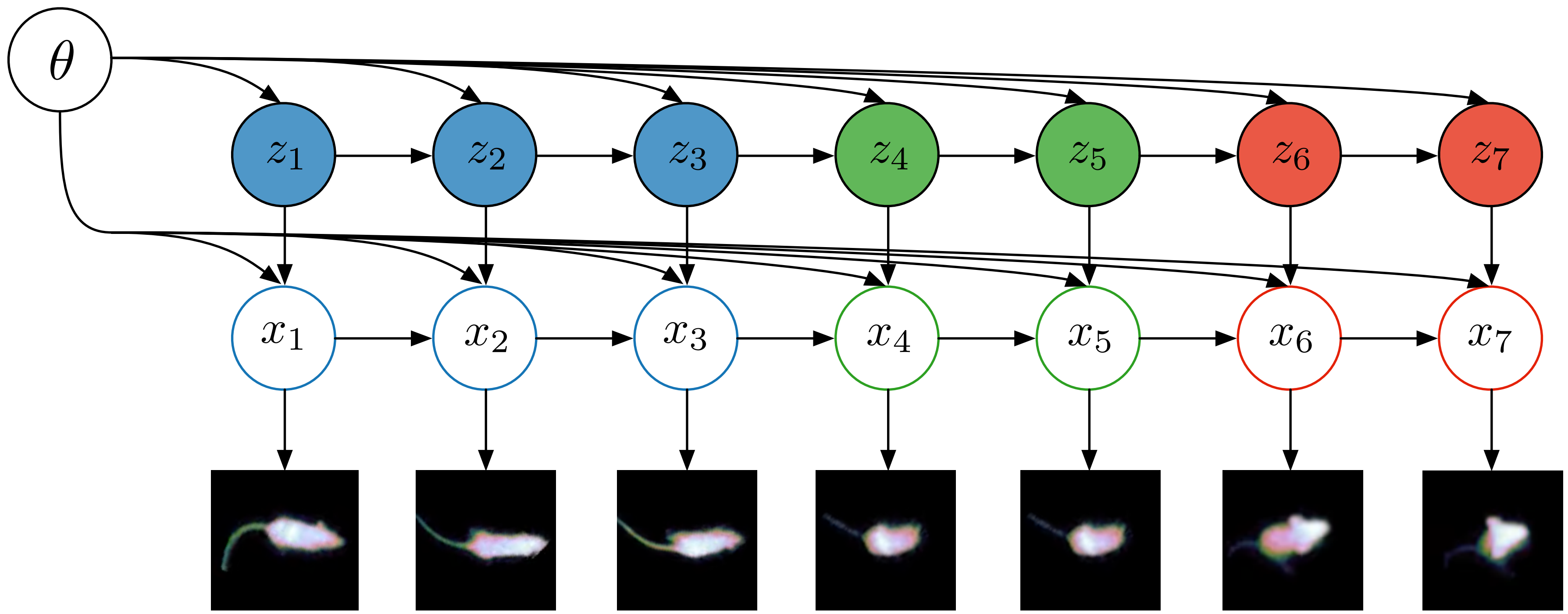
$\pi = \begin{bmatrix} — \pi^{(1)} — \\ — \pi^{(2)} — \\ — \pi^{(3)} — \end{bmatrix}$
$z_{t+1} \sim \pi^{(z_t)}$
z_1 z_2 z_3 z_4 z_5 z_6 z_7
$A^{(1)}$ $A^{(2)}$ $A^{(3)}$
$B^{(1)}$ $B^{(2)}$ $B^{(3)}$
$x_{t+1} = A^{(z_t)} x_t + B^{(z_t)} u_t \qquad u_t \overset{\text{iid}}{\sim} \mathcal{N}(0, I)$
mm

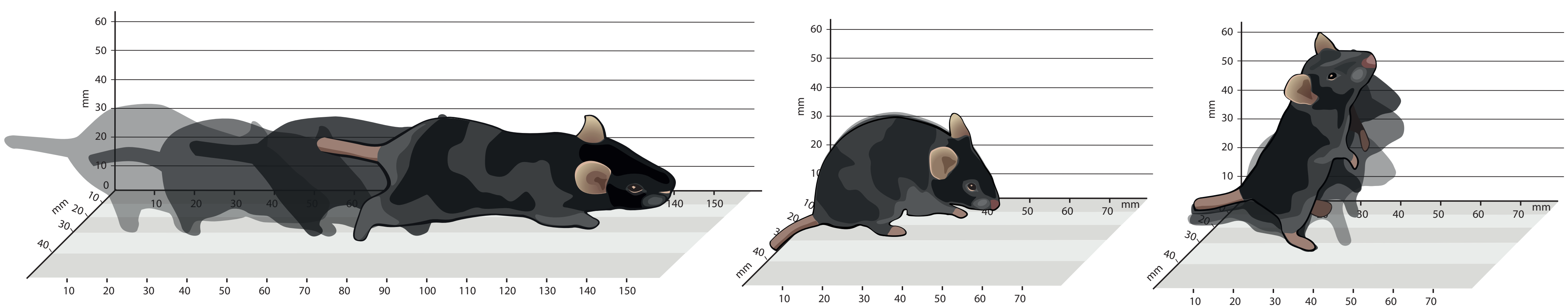


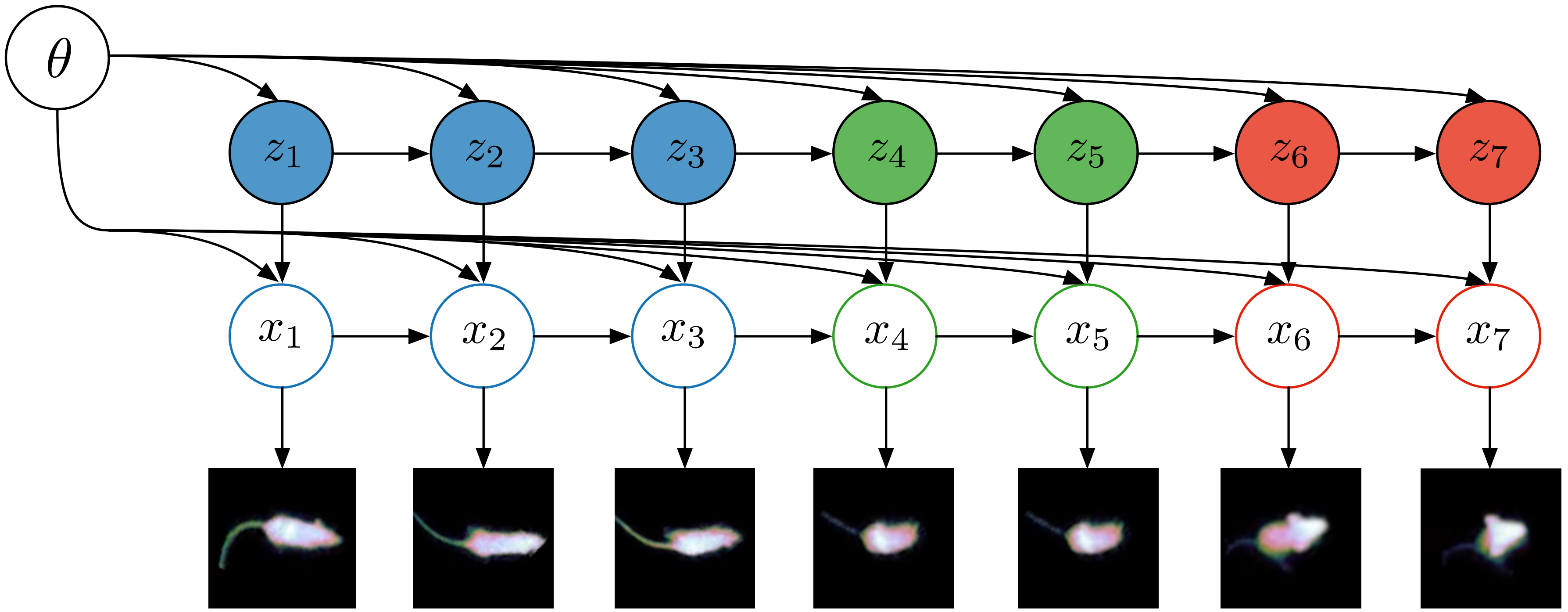
$\pi = \begin{bmatrix} — \pi^{(1)} — \\ — \pi^{(2)} — \\ — \pi^{(3)} — \end{bmatrix}$
$A^{(1)}$ $A^{(2)}$ $A^{(3)}$
$B^{(1)}$ $B^{(2)}$ $B^{(3)}$
z_1 z_2 z_3 z_4 z_5 z_6 z_7
x_1 x_2 x_3 x_4 x_5 x_6 x_7
mm


θ
z_1 z_2 z_3 z_4 z_5 z_6 z_7
x_1 x_2 x_3 x_4 x_5 x_6 x_7

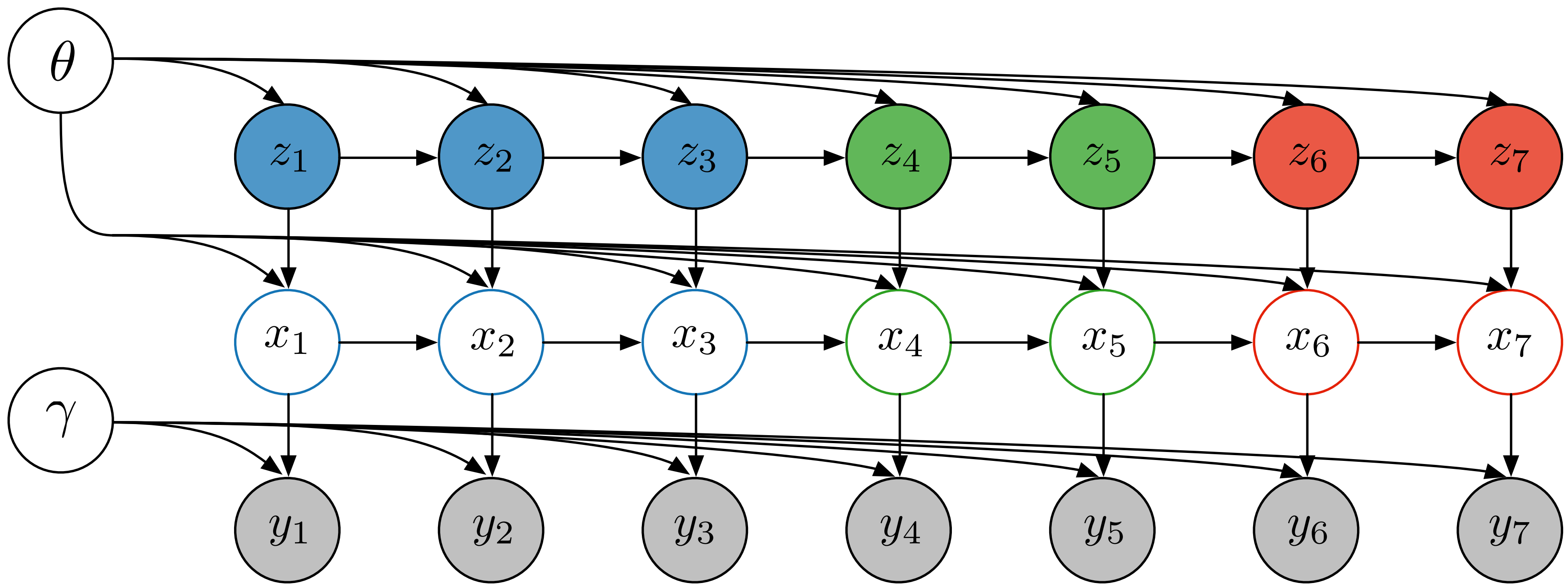

mm
0 10 20 30 40 50 60
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
mm
10 20 30 40 50 60
10 20 30 40 50 60 70
mm
10 20 30 40 50 60
10 20 30 40 50 60 70


θ
z_1 z_2 z_3 z_4 z_5 z_6 z_7
x_1 x_2 x_3 x_4 x_5 x_6 x_7

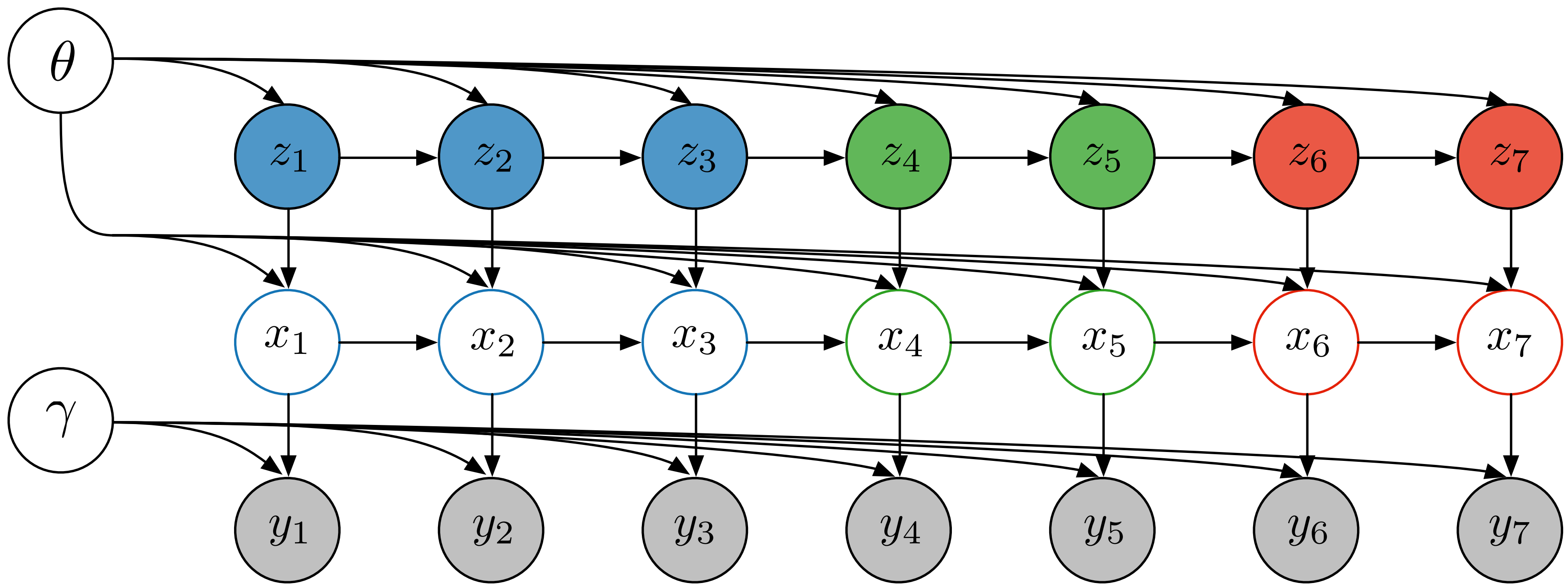

mm



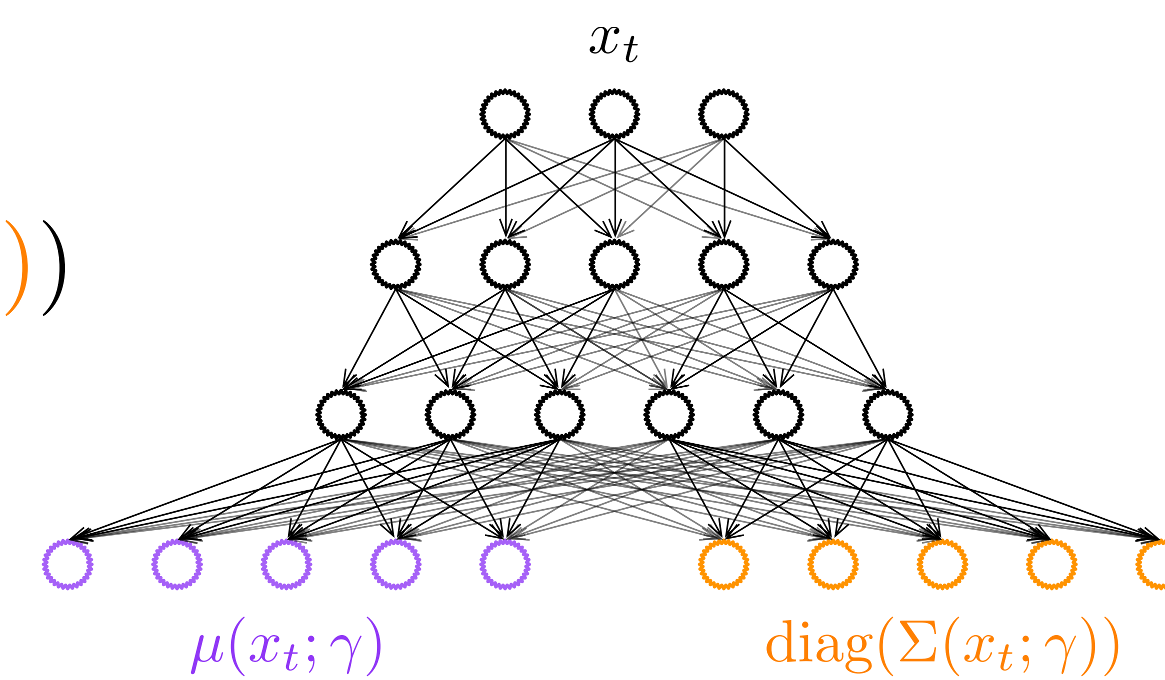
θ
z_1
z_2
z_3
z_4
z_5
z_6
z_7
x_1
x_2
x_3
x_4
x_5
x_6
x_7

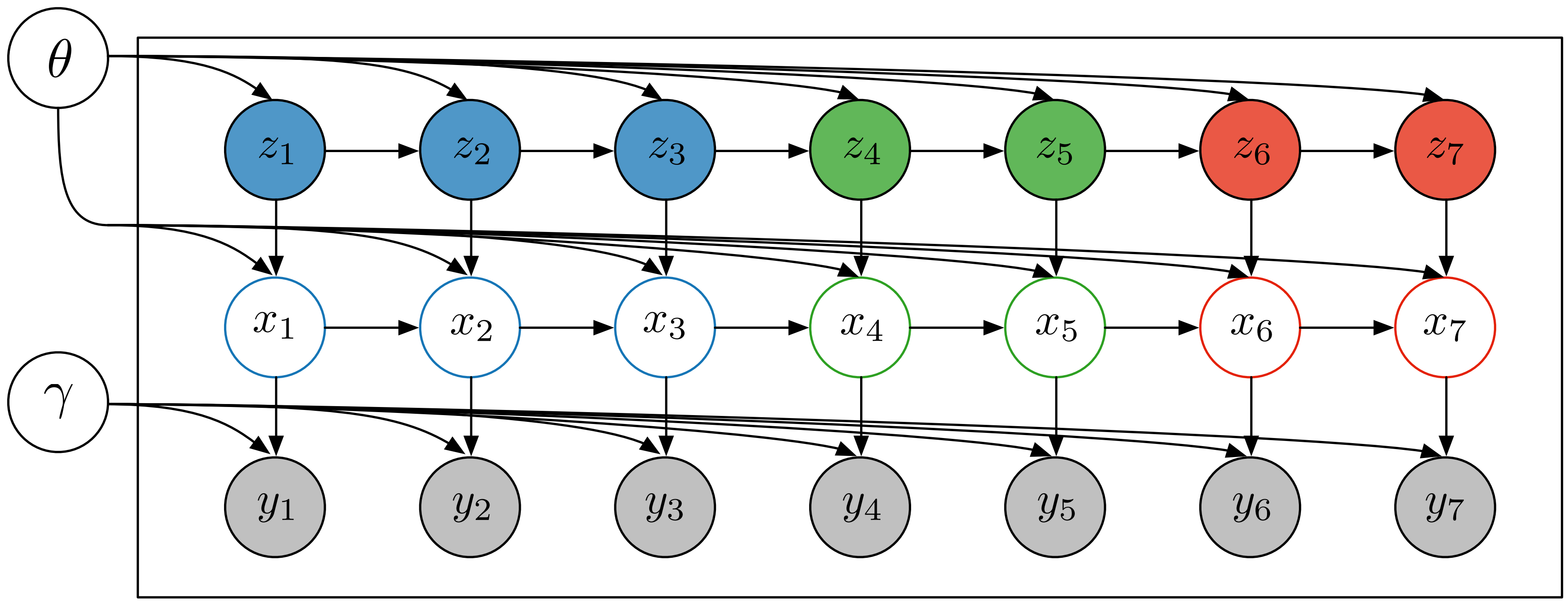


θ
γ
z_1
z_2
z_3
z_4
z_5
z_6
z_7
x_1
x_2
x_3
x_4
x_5
x_6
x_7
y_1
y_2
y_3
y_4
y_5
y_6
y_7

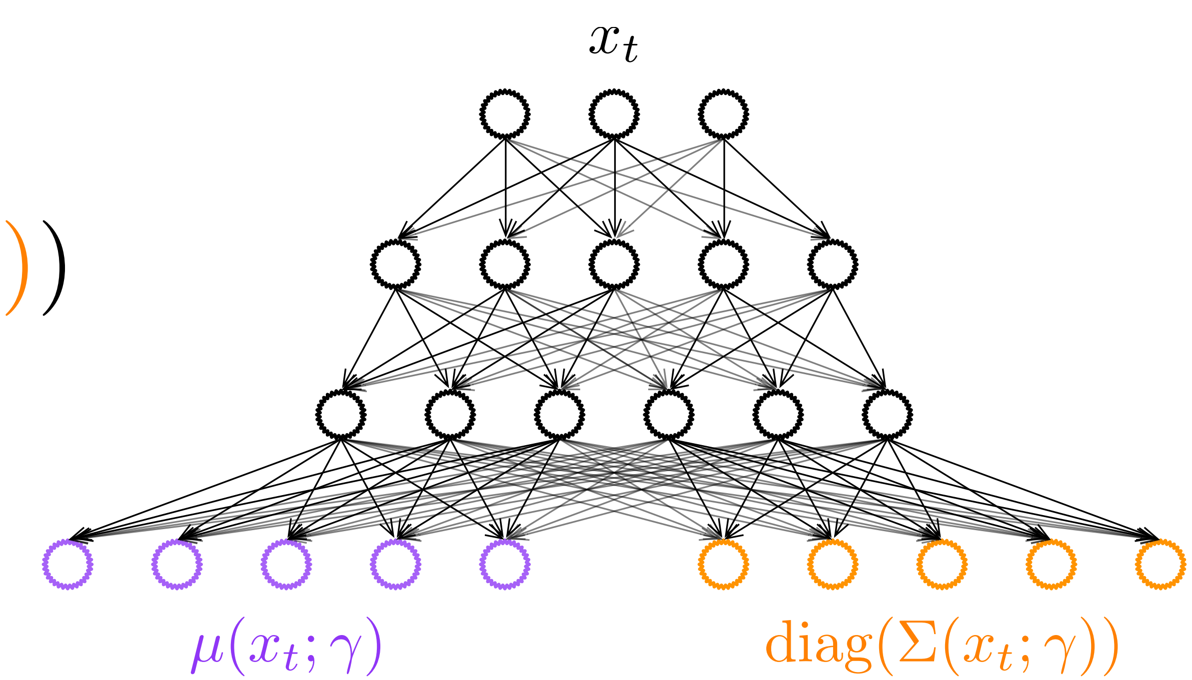


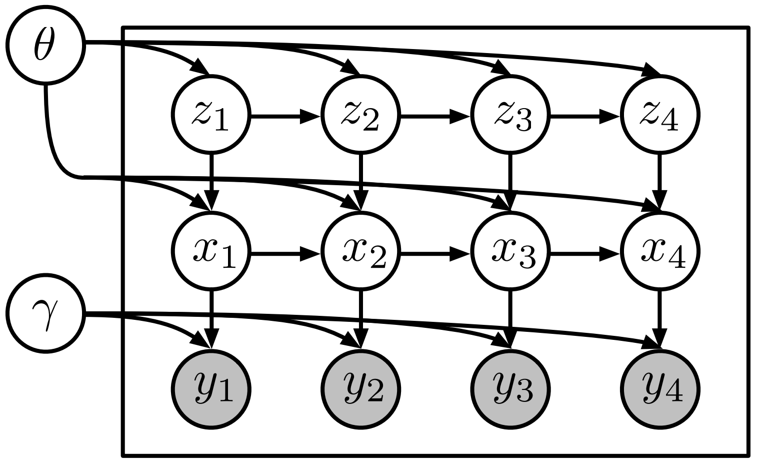
$$y_t \mid x_t, \gamma \;\sim\; \mathcal{N}(\mu(x_t;\gamma), \Sigma(x_t;\gamma))$$



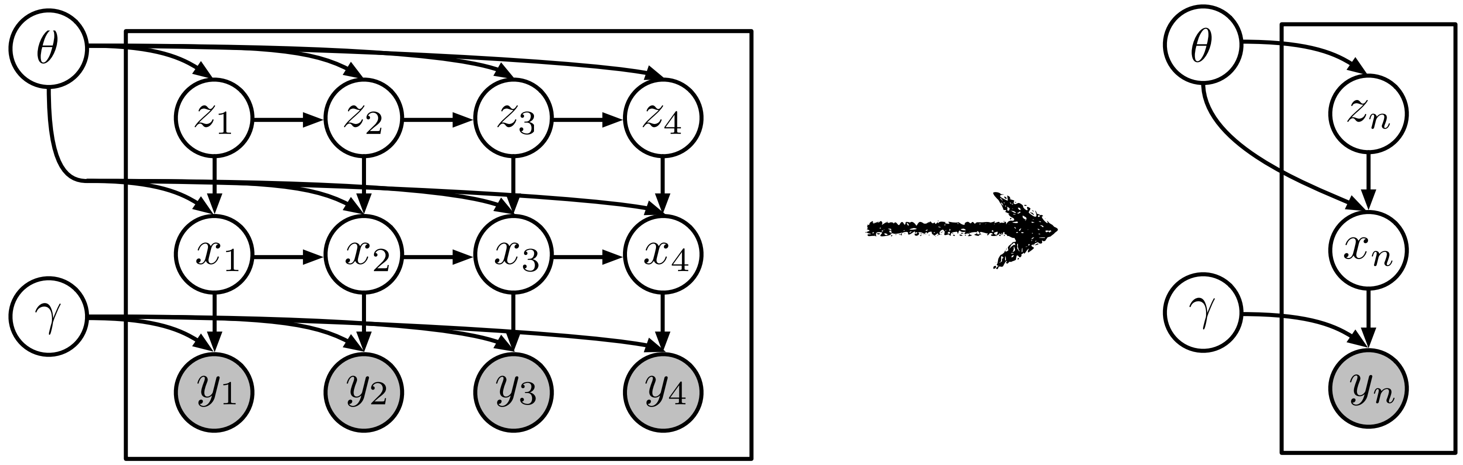


$$y_t \mid x_t, \gamma \sim \mathcal{N}(\mu(x_t;\gamma), \Sigma(x_t;\gamma))$$

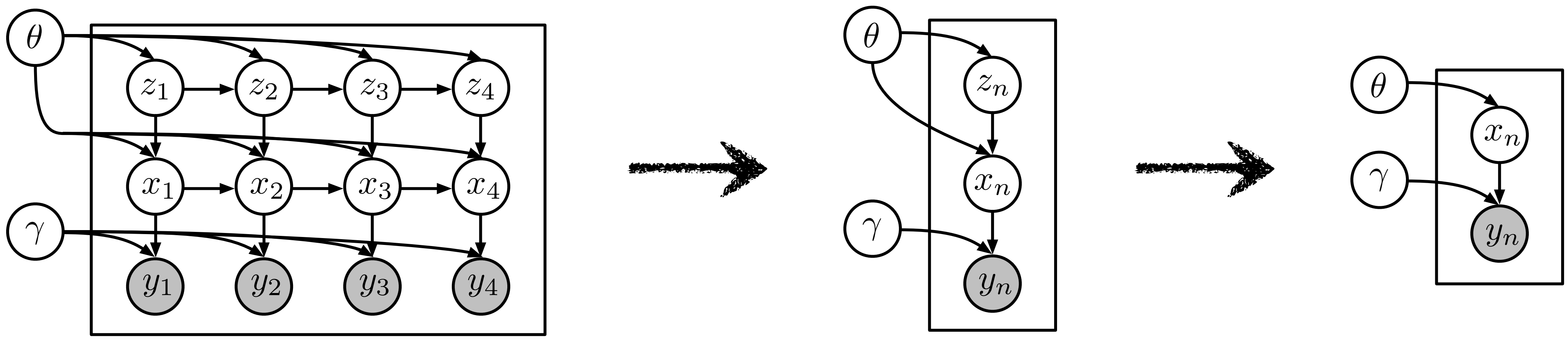




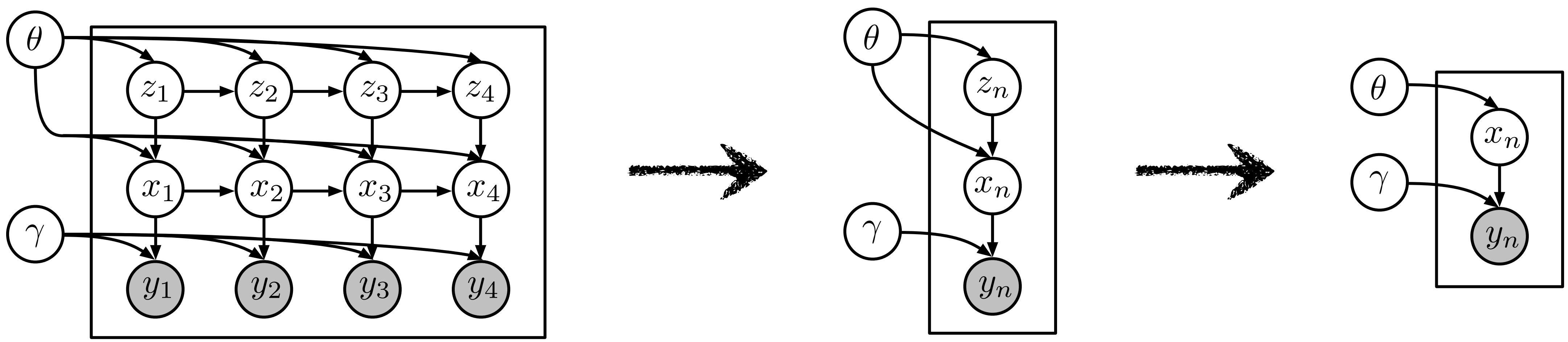
θ
γ
z_1
z_2
z_3
z_4
x_1
x_2
x_3
x_4
y_1
y_2
y_3
y_4



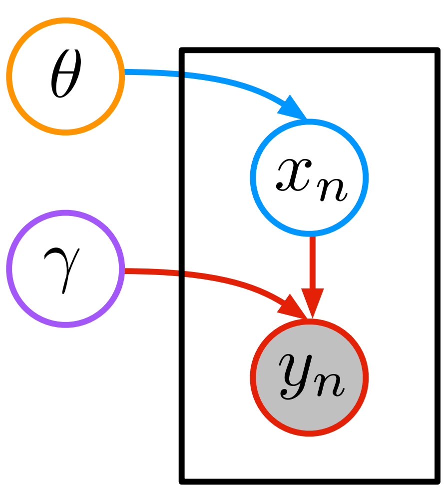
θ
z_1
z_2
z_3
z_4
x_1
x_2
x_3
x_4
γ
y_1
y_2
y_3
y_4
θ
z_n
x_n
γ
y_n



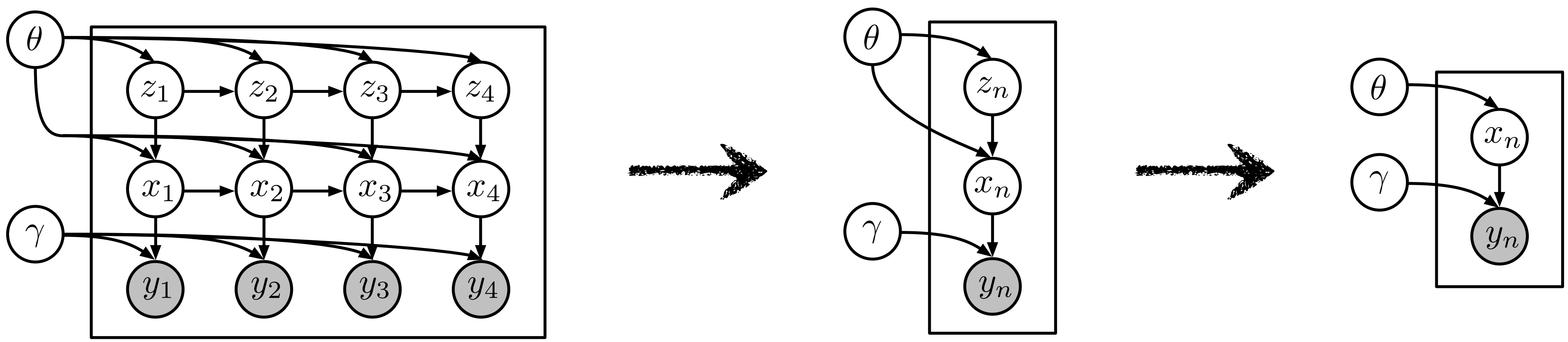
θ
γ
z1 z2 z3 z4
x1 x2 x3 x4
y1 y2 y3 y4
θ
γ
zn
xn
yn
θ
γ
xn
yn

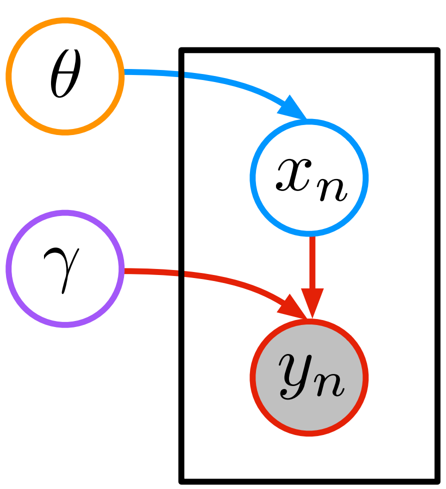


θ
γ
z_1 z_2 z_3 z_4
x_1 x_2 x_3 x_4
y_1 y_2 y_3 y_4
θ
γ
z_n
x_n
y_n
θ
γ
x_n
y_n



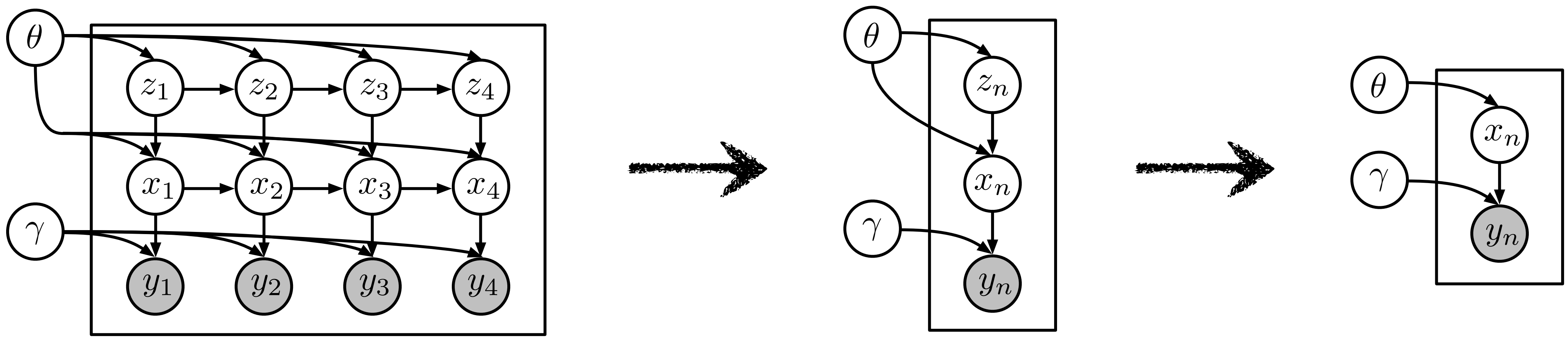
θ
γ
x_n
y_n

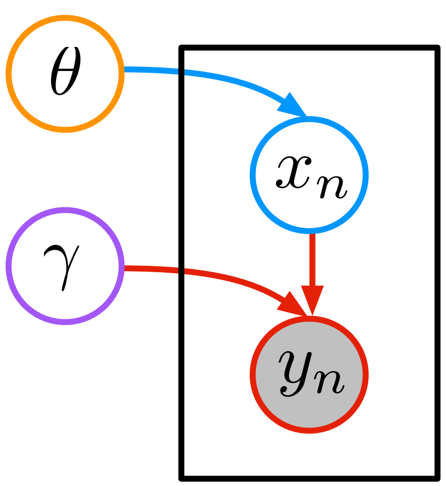




$p(\theta)$ conjugate prior on global variables

$p(x \mid \theta)$ exponential family on local variables



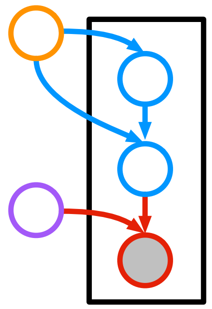


$p(\theta)$ conjugate prior on global variables

$p(x \mid \theta)$ exponential family on local variables

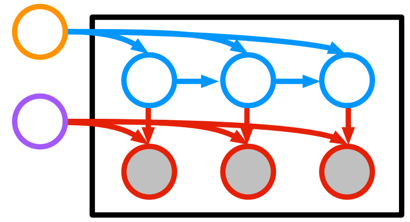
$p(\gamma)$ any prior on observation parameters

$p(y \mid x, \gamma)$ neural network observation model



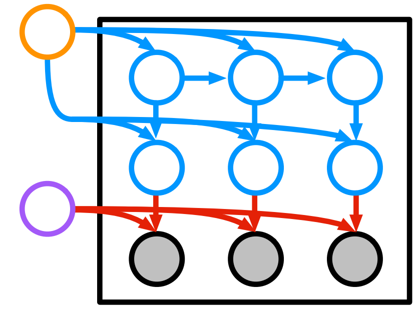
[1]

Gaussian mixture model



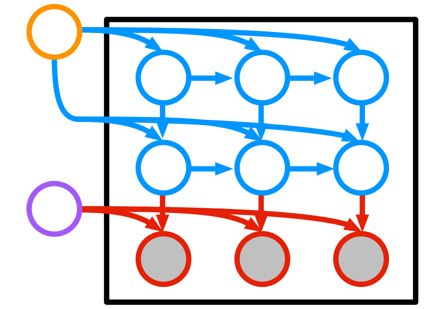
[2]

Linear dynamical system



[3]

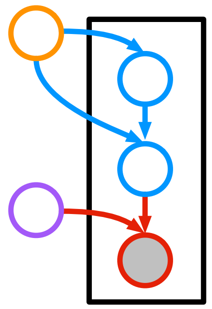
Hidden Markov model



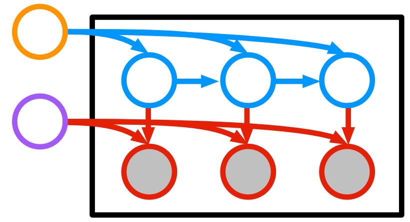
[4]

Switching LDS

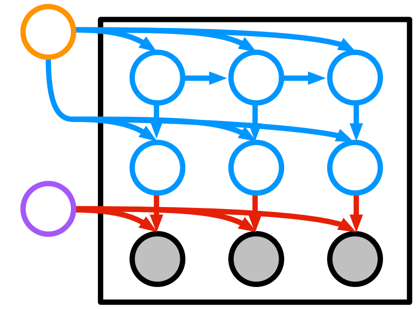
[1] Palmer, Wipf, Kreutz-Delgado, and Rao. Variational EM algorithms for non-Gaussian latent variable models. NIPS 2005.
[2] Ghahramani and Beal. Propagation algorithms for variational Bayesian learning. NIPS 2001.
[3] Beal. Variational algorithms for approximate Bayesian inference, Ch. 3. U of London Ph.D. Thesis 2003.
[4] Ghahramani and Hinton. Variational learning for switching state-space models. Neural Computation 2000.


[1]

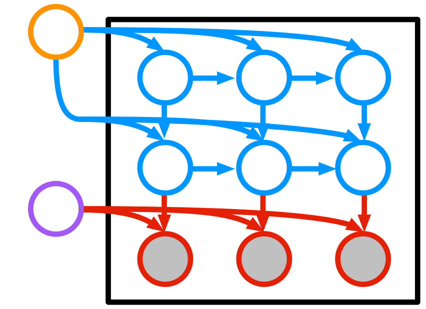
Gaussian mixture model


[2]

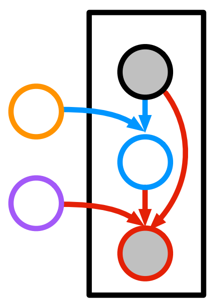
Linear dynamical system


[3]

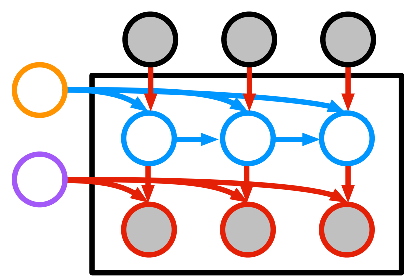
Hidden Markov model


[4]

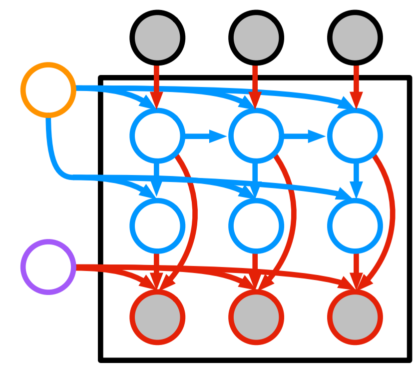
Switching LDS


[5]

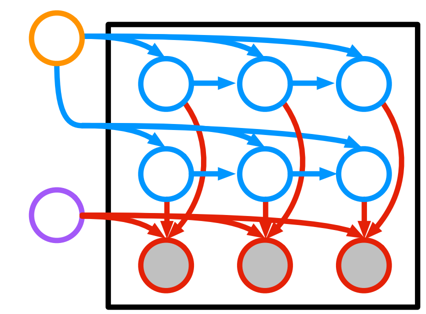
Mixture of Experts


[2]

Driven LDS

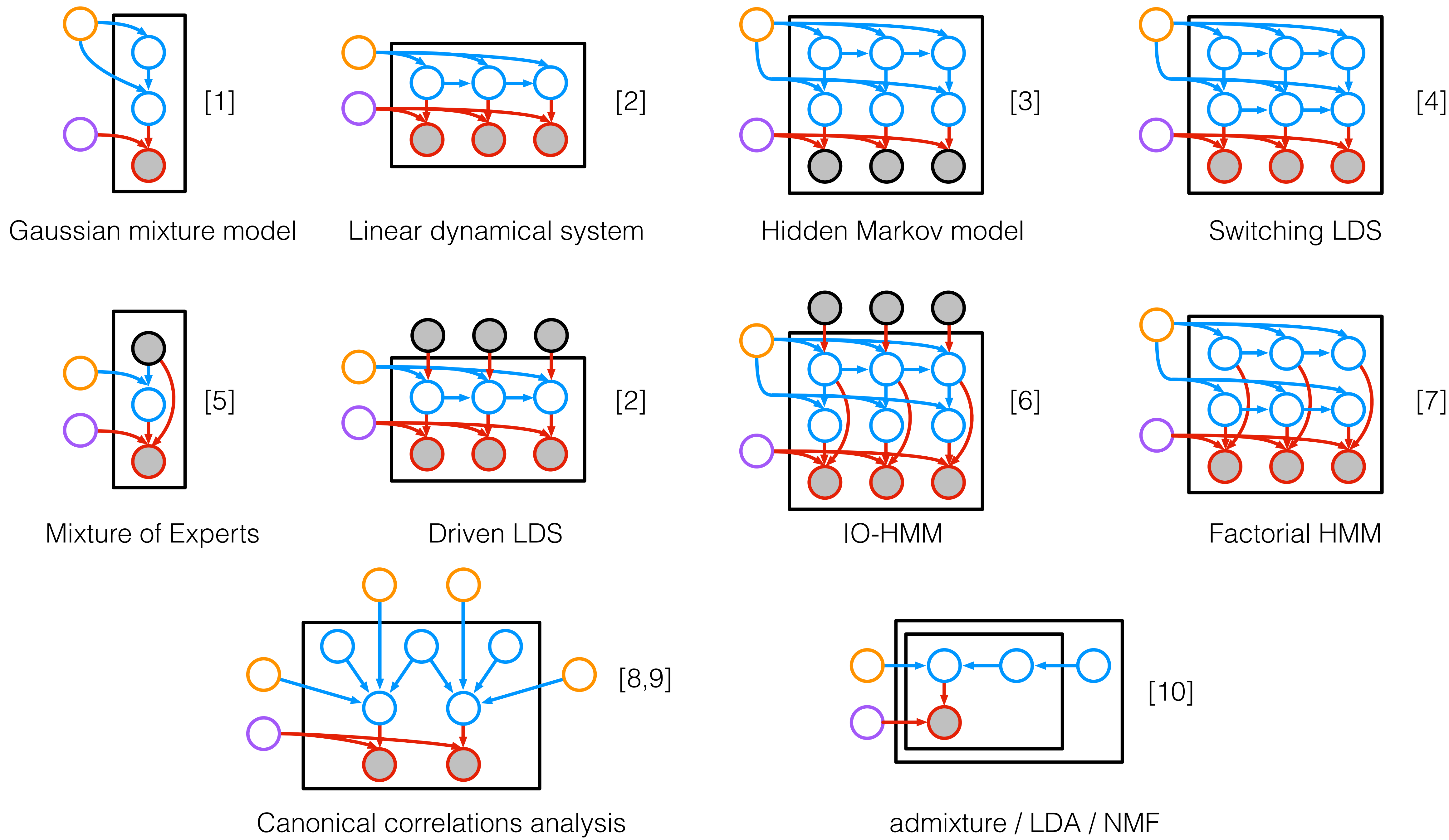

[6]

IO-HMM

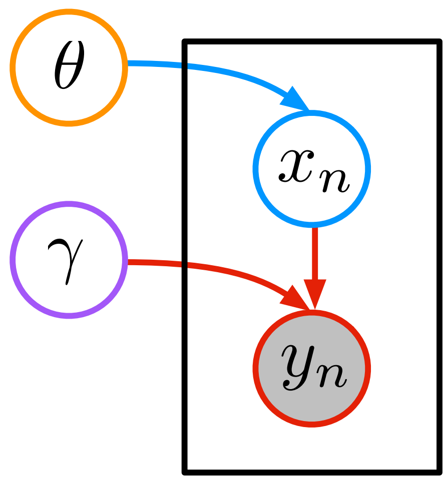

[7]

Factorial HMM

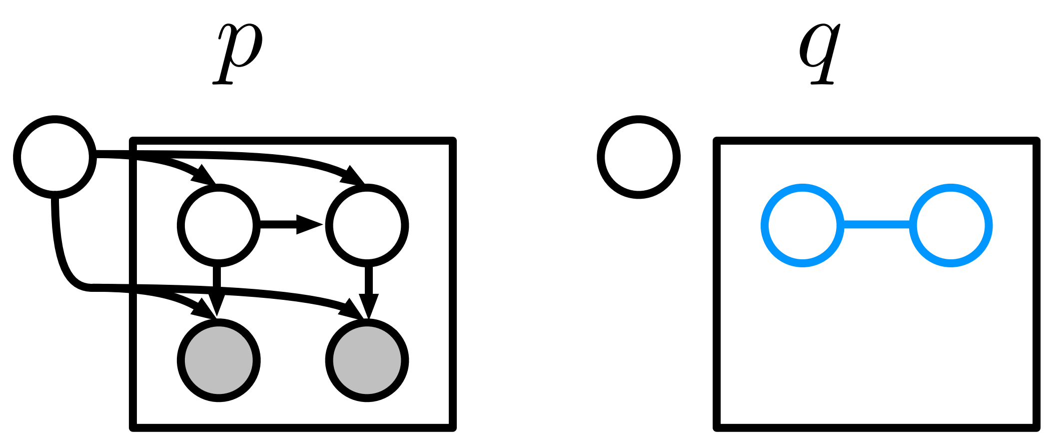
[1] Palmer, Wipf, Kreutz-Delgado, and Rao. Variational EM algorithms for non-Gaussian latent variable models. NIPS 2005.
[2] Ghahramani and Beal. Propagation algorithms for variational Bayesian learning. NIPS 2001.
[3] Beal. Variational algorithms for approximate Bayesian inference, Ch. 3. U of London 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] Palmer, Wipf, Kreutz-Delgado, and Rao. Variational EM algorithms for non-Gaussian latent variable models. NIPS 2005.
[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.



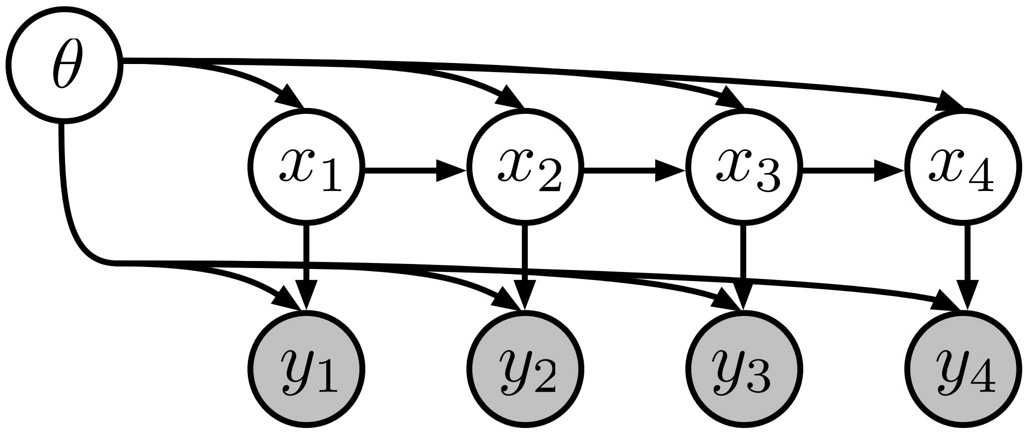
Inference?



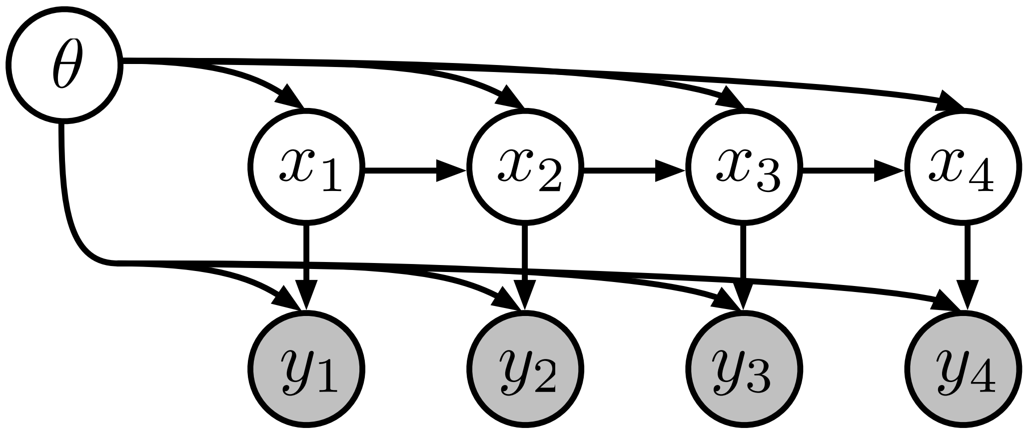
$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\, q(\theta) q(x) \,]$$

Natural gradient SVI
for nice exp. fam. 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.



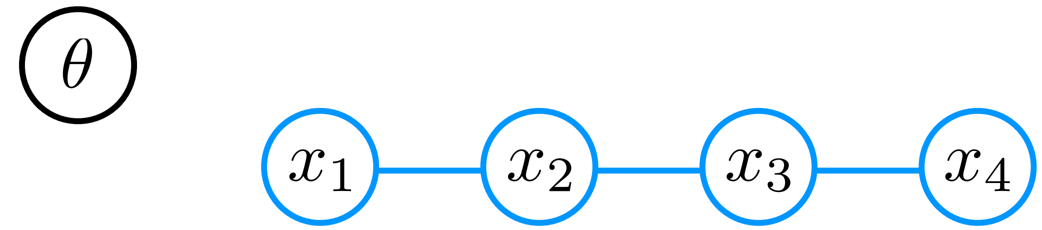
$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \theta)$ is a linear-Gaussian observation
$p(\theta)$ is a conjugate prior



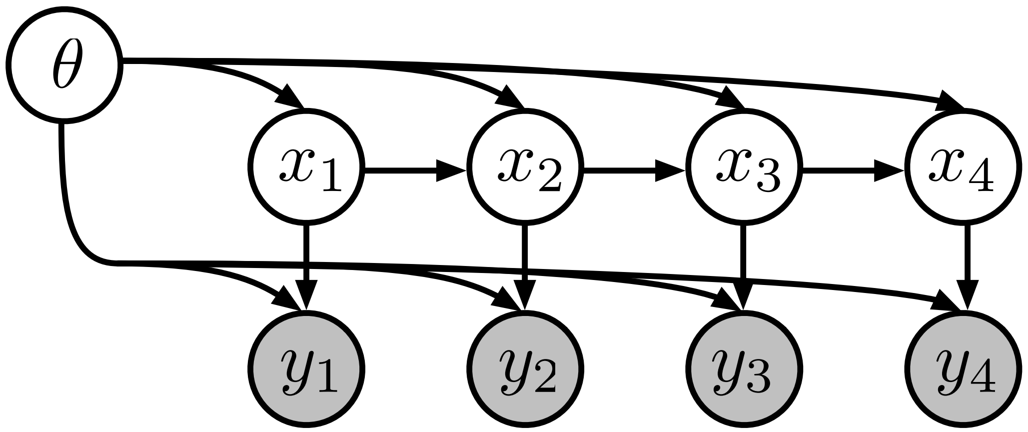
$p(x \mid \theta)$ is a linear dynamical system

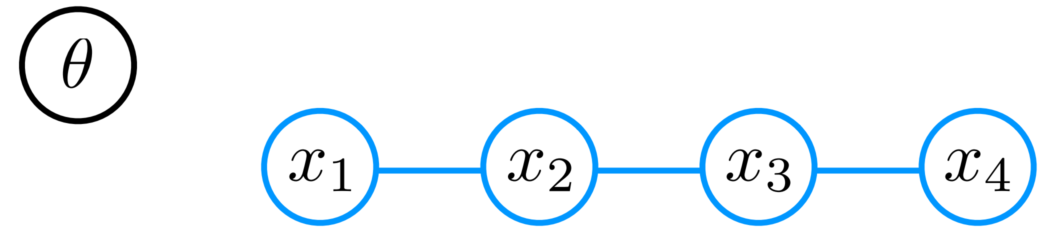
$p(y \mid x, \theta)$ is a linear-Gaussian observation

$p(\theta)$ is a conjugate prior



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

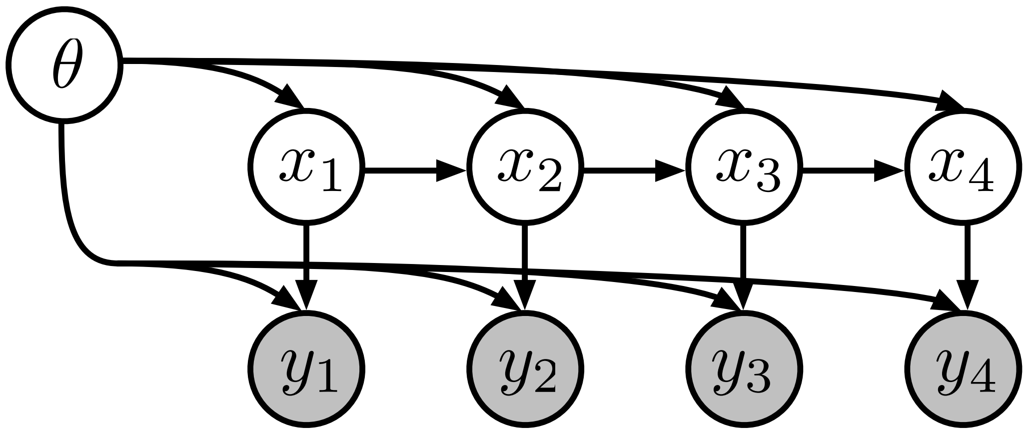


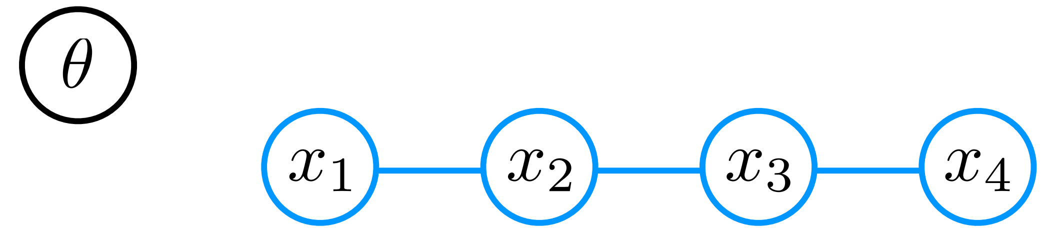


$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \theta)$ is a linear-Gaussian observation
$p(\theta)$ is a conjugate prior

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

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



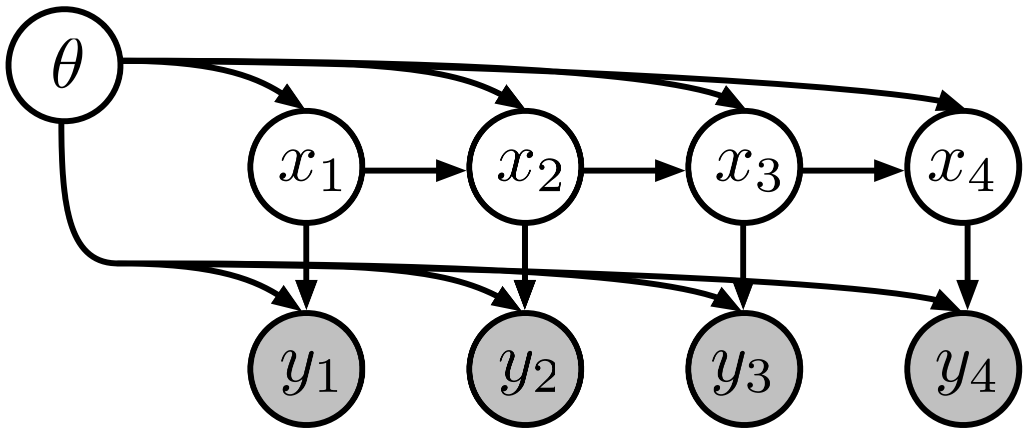


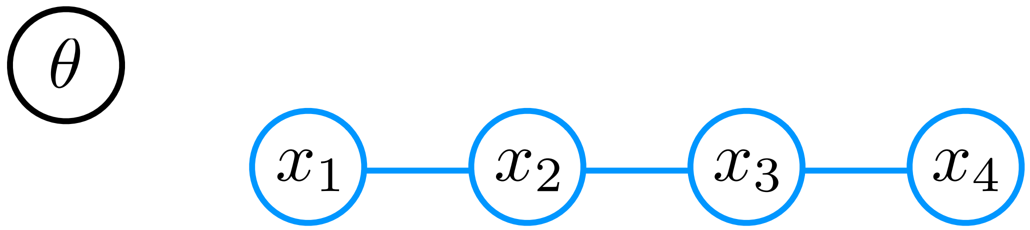
$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \theta)$ is a linear-Gaussian observation
$p(\theta)$ is a conjugate prior

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

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

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





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

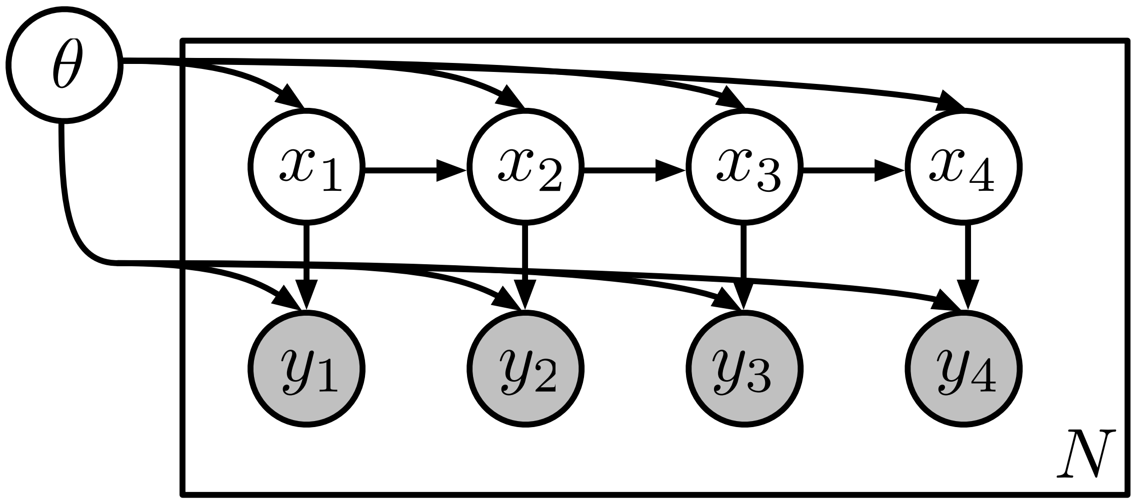
$$q(\theta)q(x) \approx p(\theta, x \,|\, y)$$

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

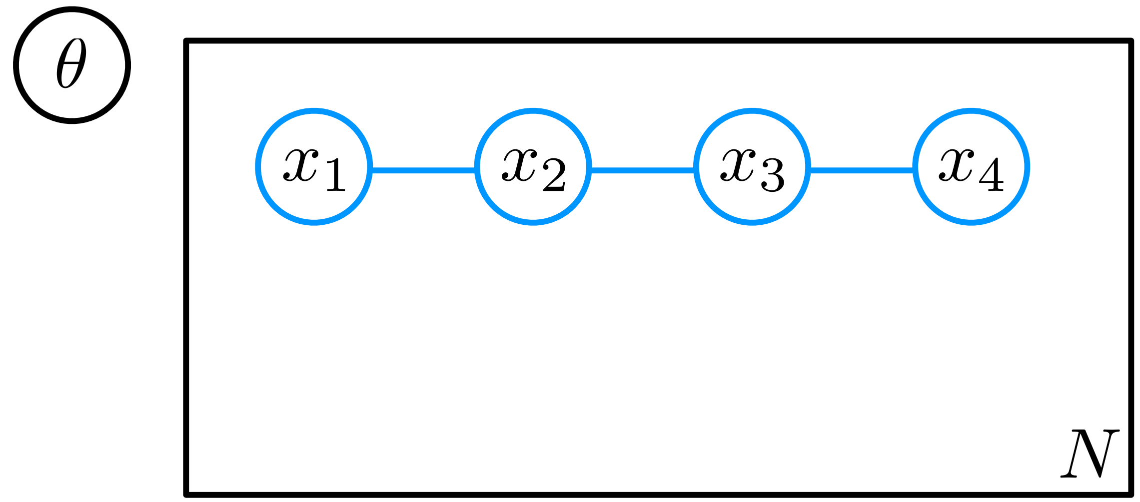
$$\eta_x^*(\eta_\theta) \triangleq \arg\max_{\eta_x} \mathcal{L}(\eta_\theta, \eta_x) \qquad \mathcal{L}_{\text{SVI}}(\eta_\theta) \triangleq \mathcal{L}(\eta_\theta, \eta_x^*(\eta_\theta))$$

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



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

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

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

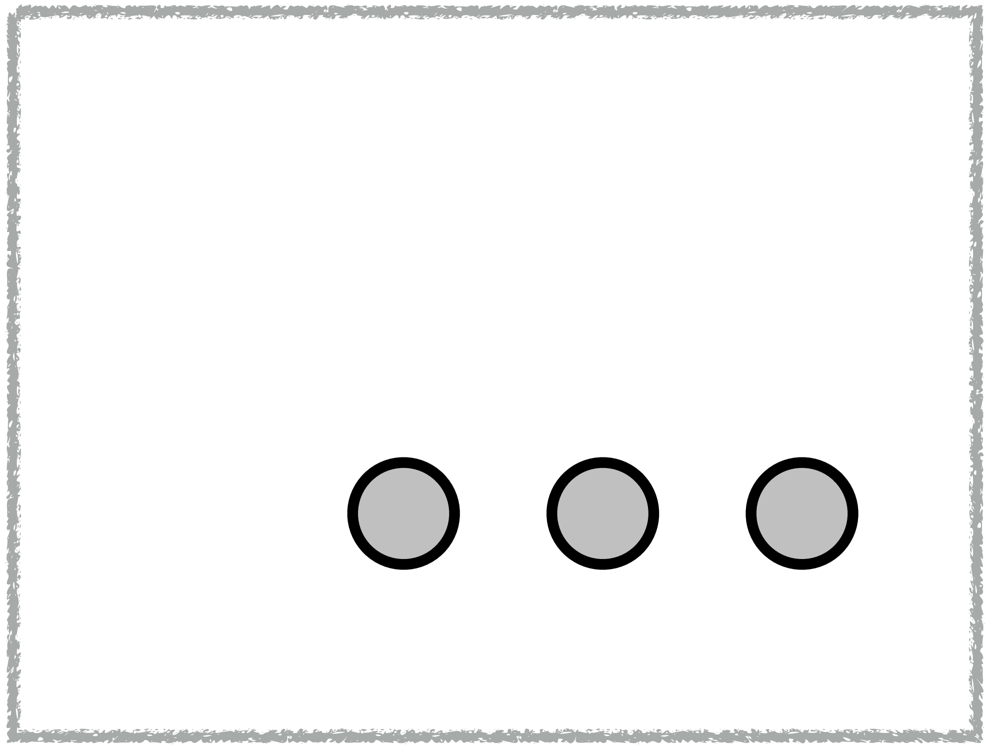
Proposition (natural gradient SVI of Hoffman et al. 2013)

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

Step 1: compute evidence potentials

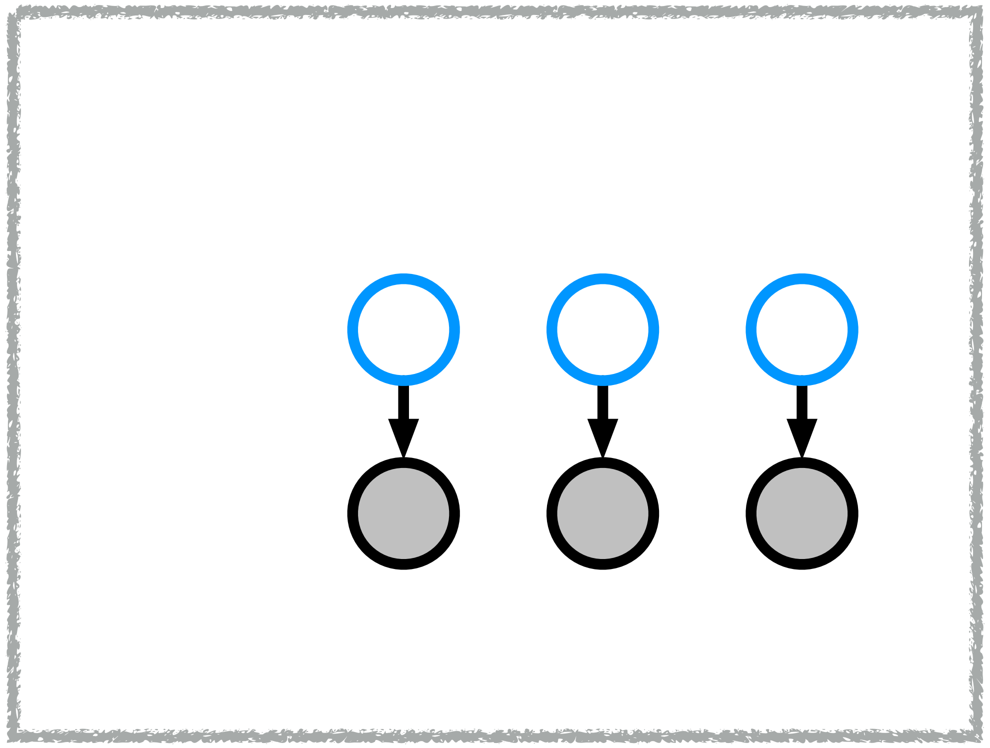
[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

Step 1: compute evidence potentials



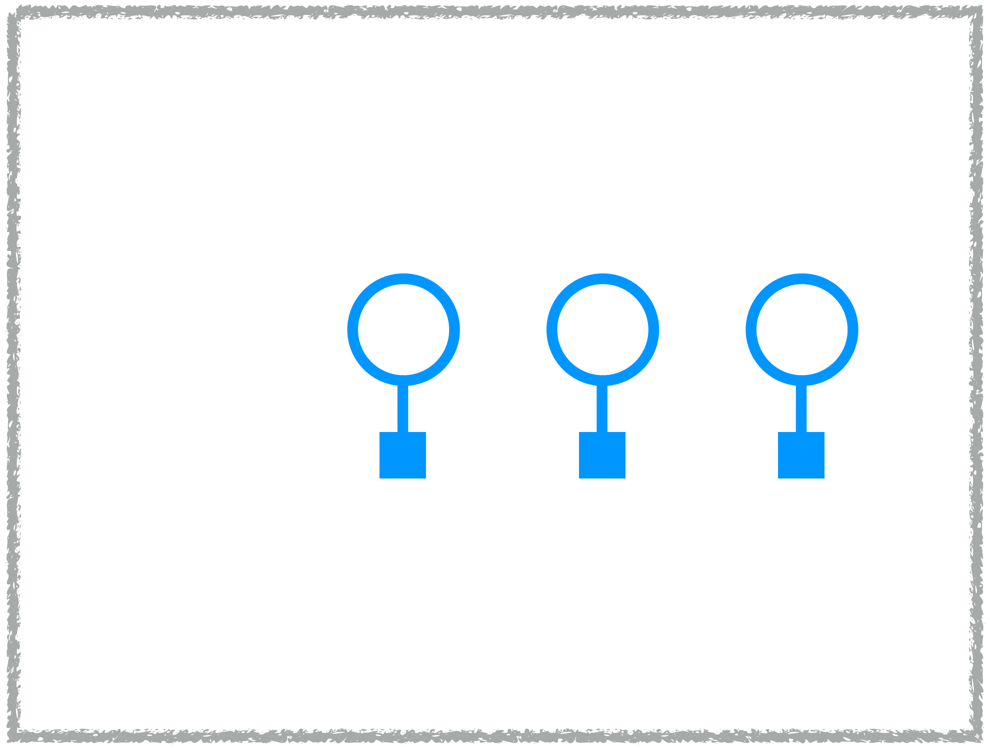
[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

Step 1: compute evidence potentials



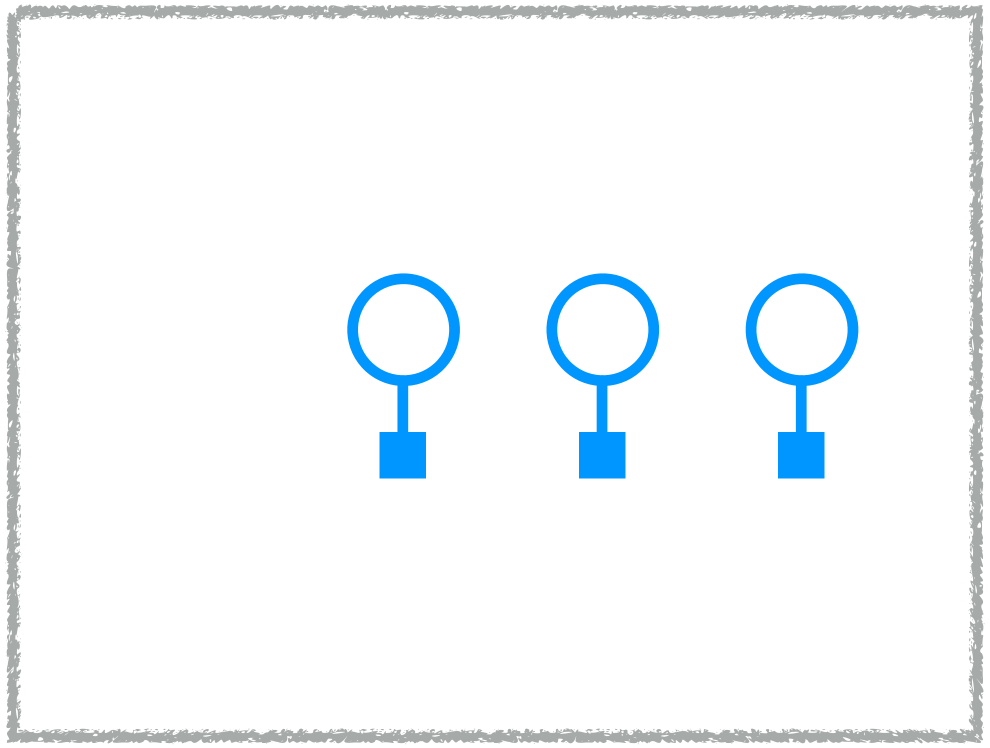
[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

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.

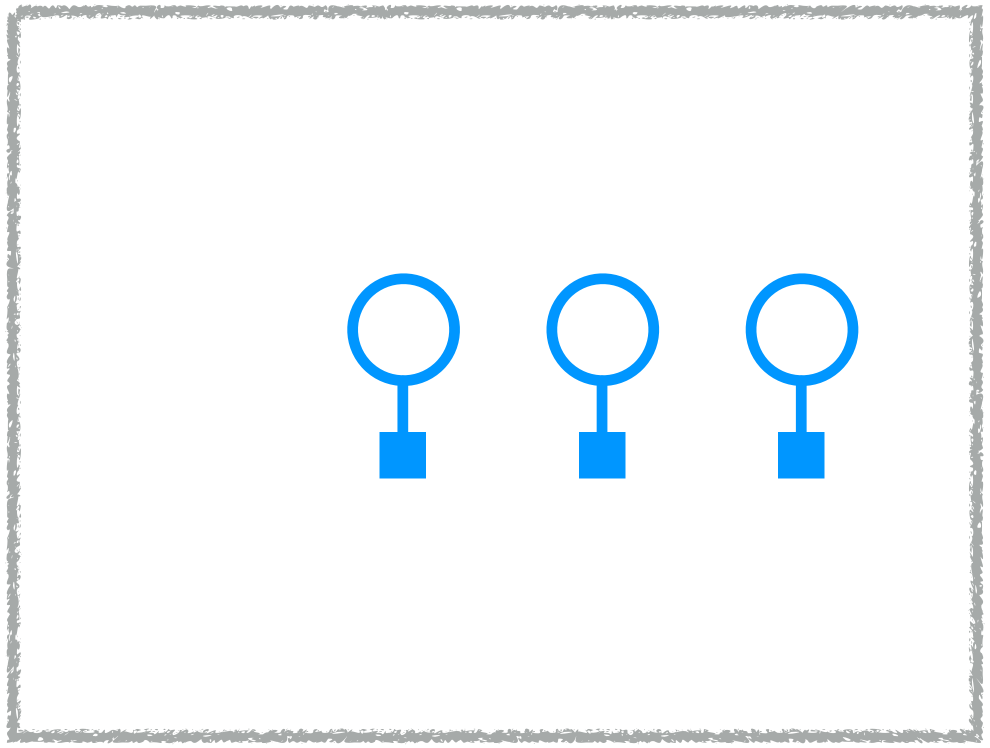
Step 1: compute evidence potentials



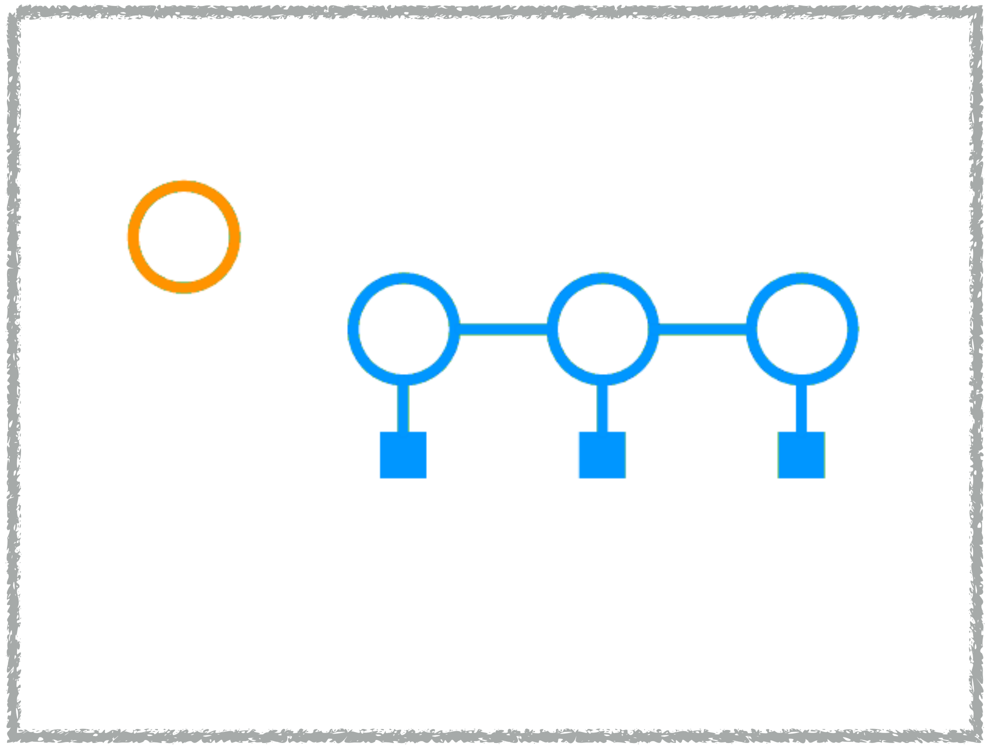
Step 2: run fast message passing

[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

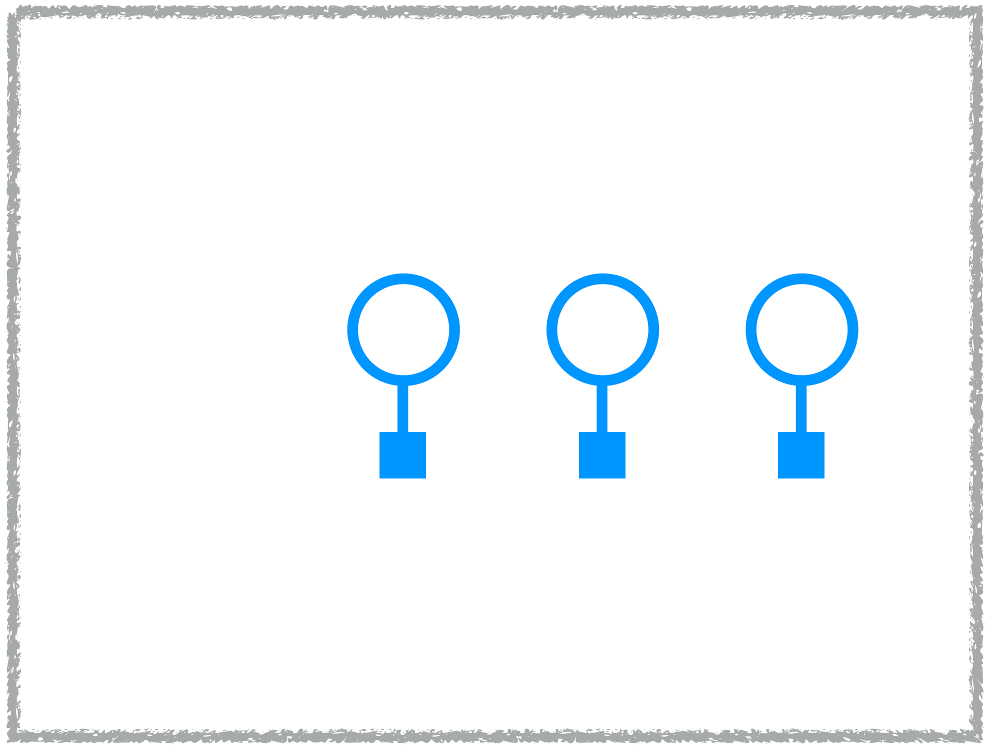


Step 2: run fast message passing

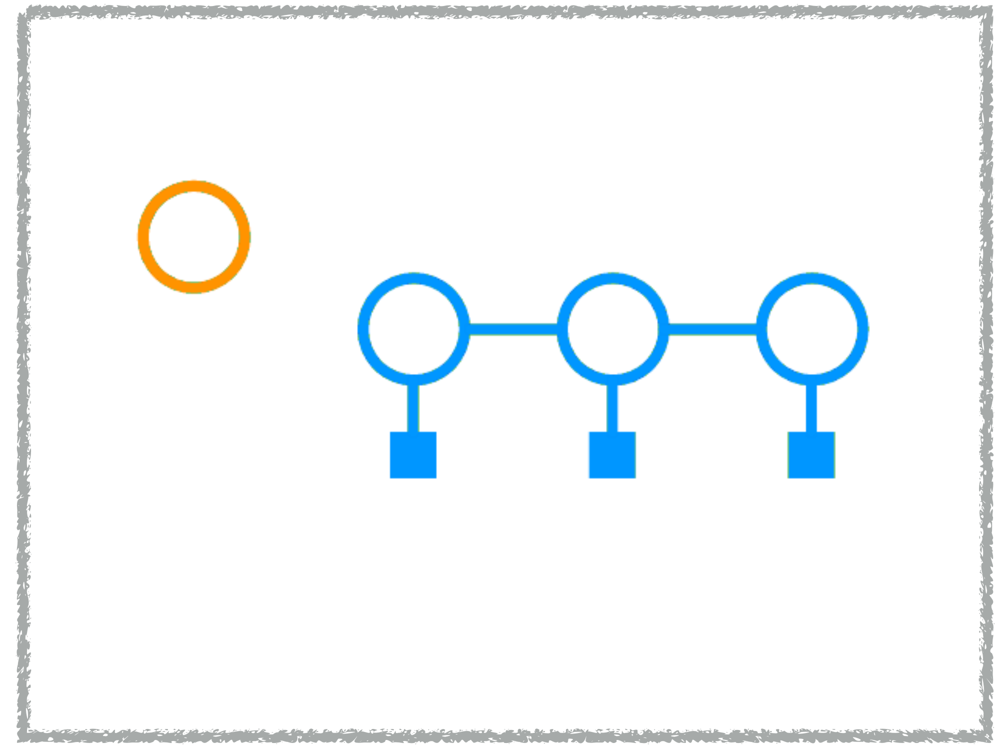


[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

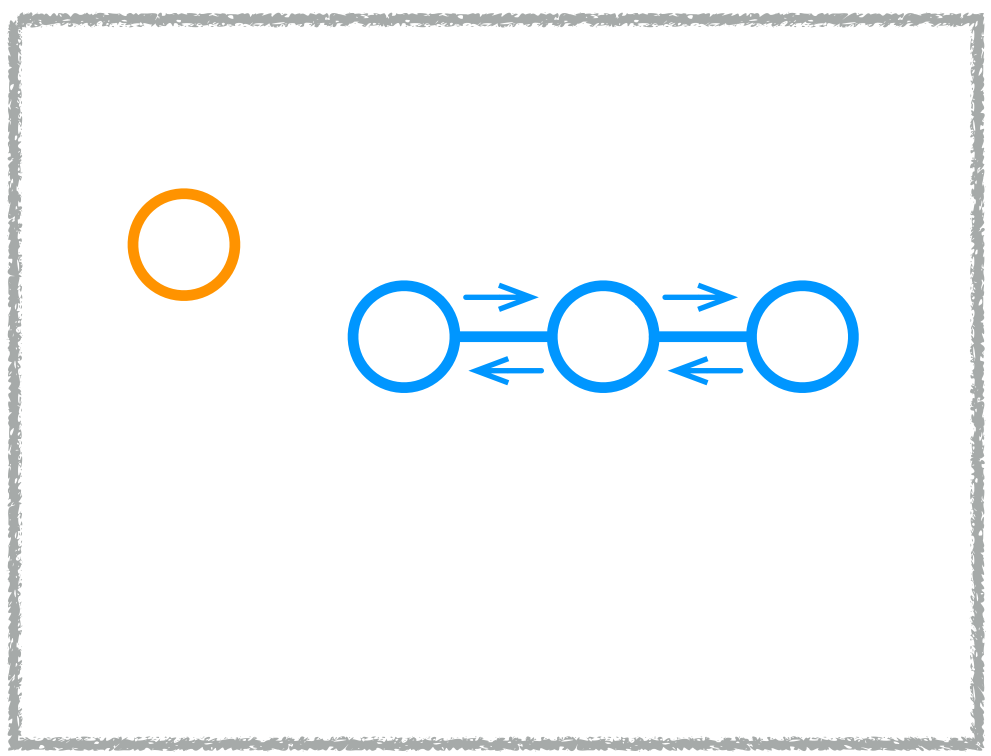
Step 1: compute evidence potentials



Step 2: run fast message passing

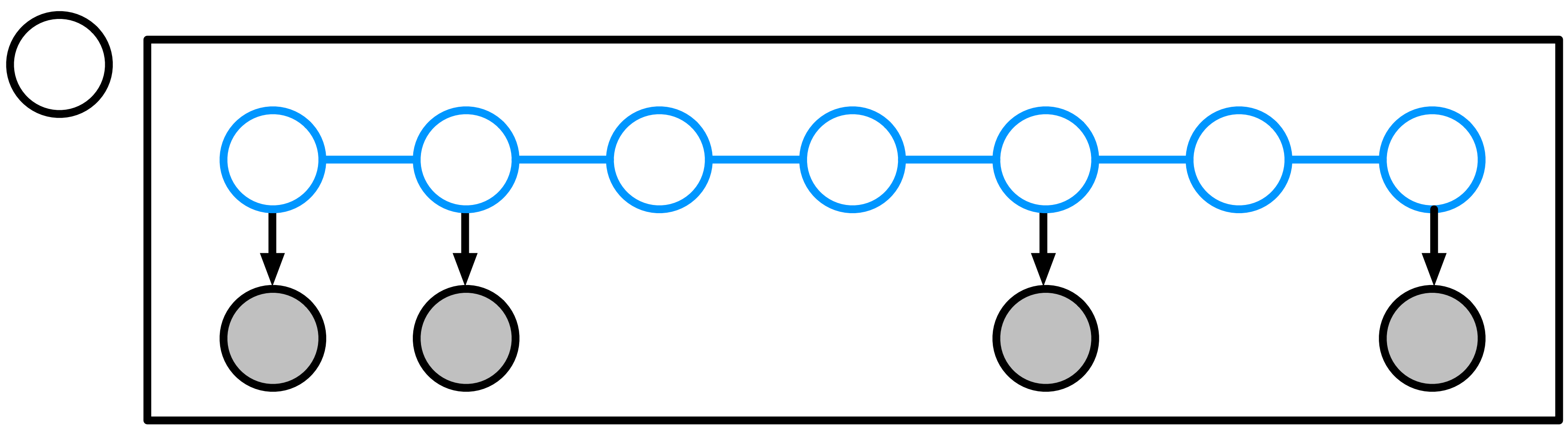


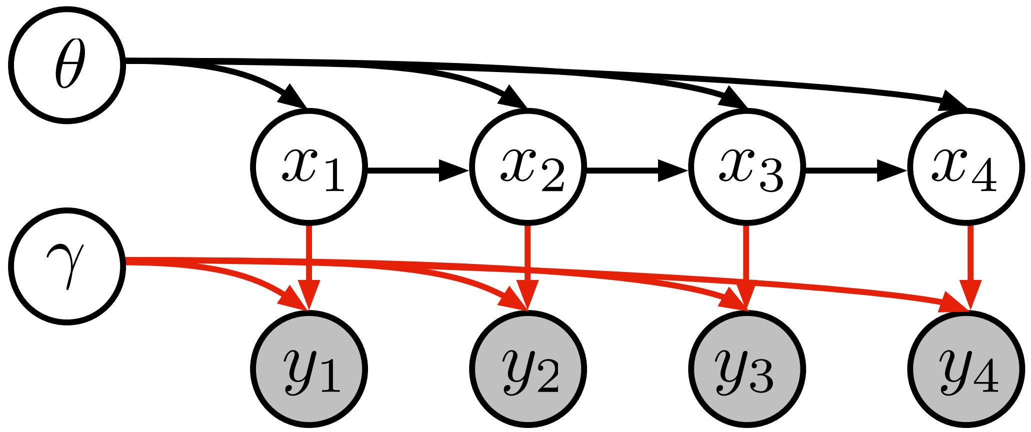
Step 3: compute natural gradient



[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

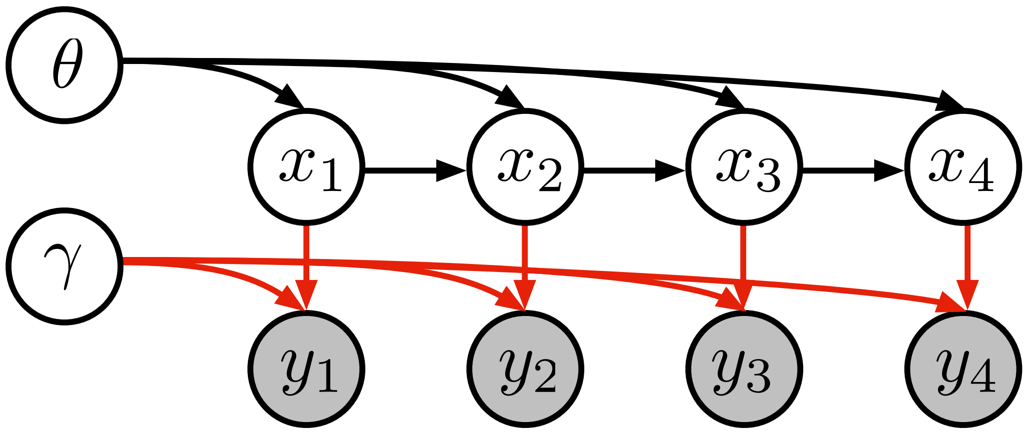




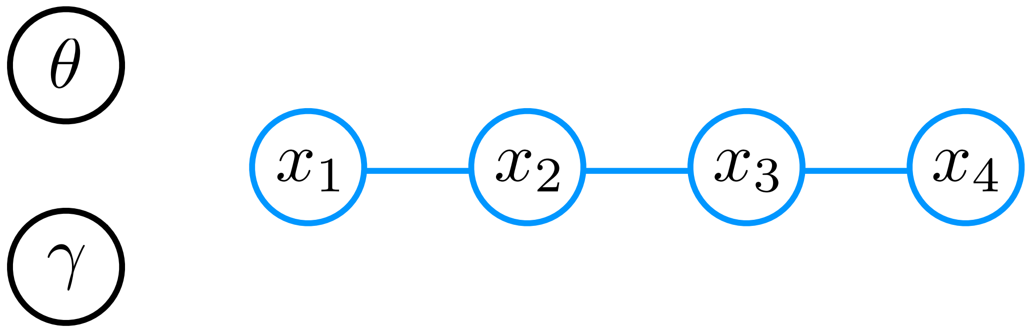
$p(x \mid \theta)$ is a linear dynamical system

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

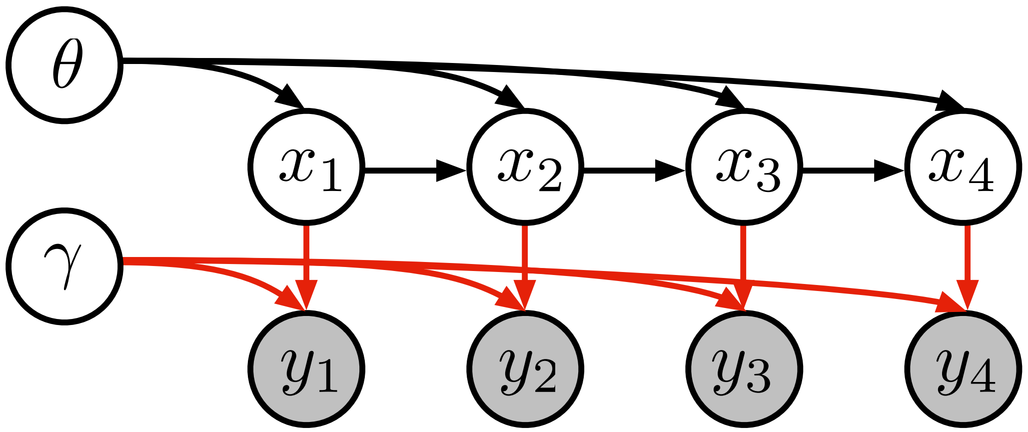
$p(\theta)$ is a conjugate prior, $p(\gamma)$ is generic

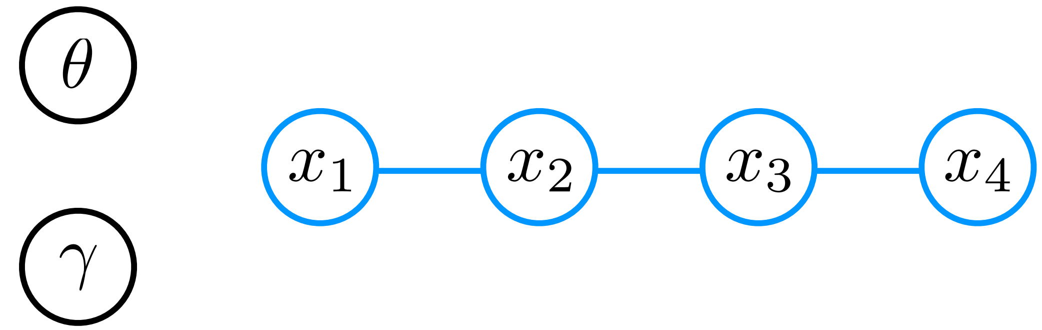


$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \gamma)$ is a neural network decoder
$p(\theta)$ is a conjugate prior, $p(\gamma)$ is generic



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

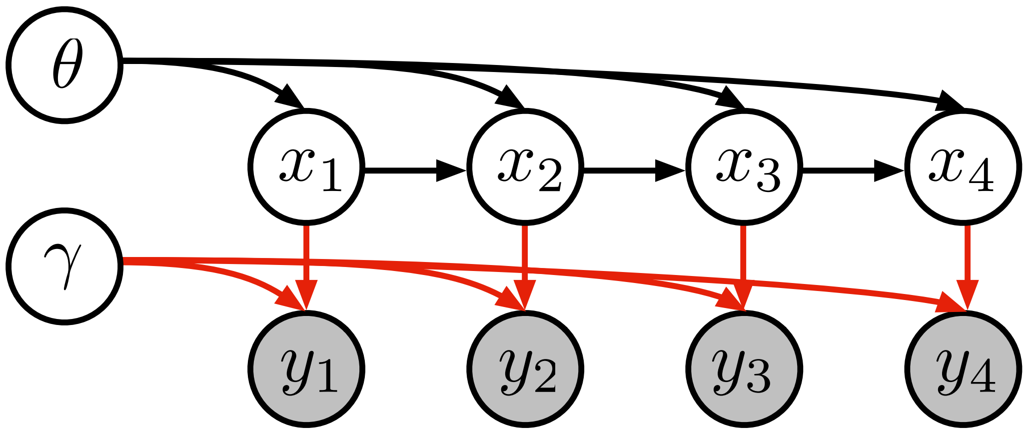


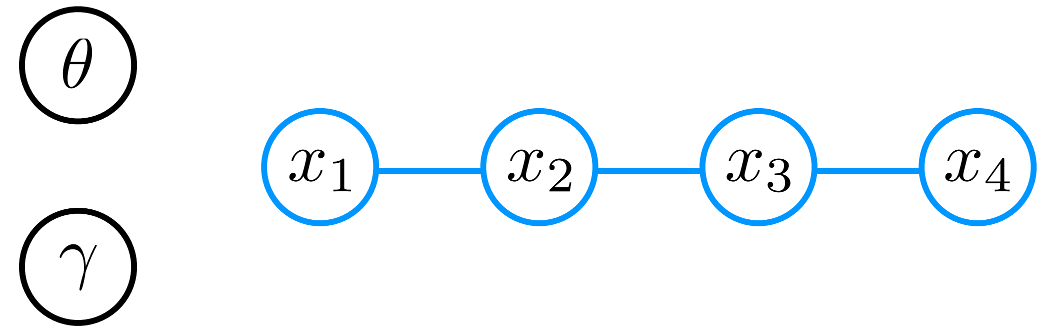


$p(x \mid \theta)$ is a linear dynamical system
$p(y \mid x, \gamma)$ is a neural network decoder
$p(\theta)$ is a conjugate prior, $p(\gamma)$ is generic

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

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





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

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

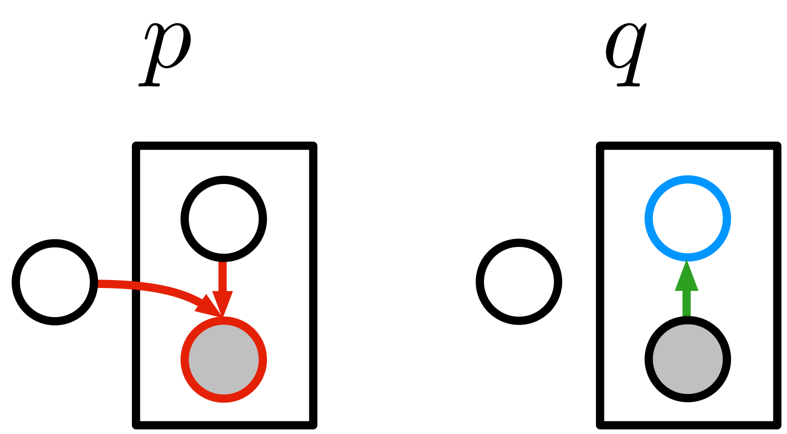
$p(\theta)$ is a conjugate prior, $p(\gamma)$ is generic

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

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

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

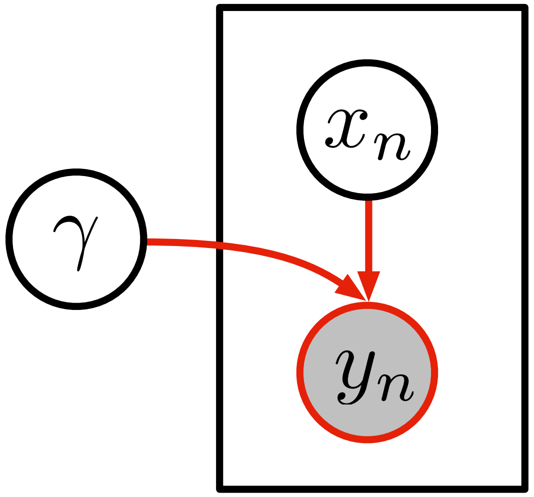
$$\mathcal{L}_{\text{SVI}}(\eta_\theta, \eta_\gamma) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^\star(\eta_\theta, \eta_\gamma))$$



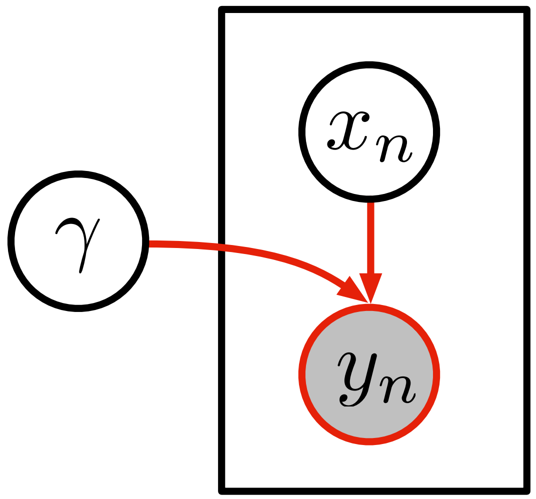
$$q^*(x) \triangleq \mathcal{N}(x \,|\, \mu(y;\phi), \Sigma(y;\phi))$$

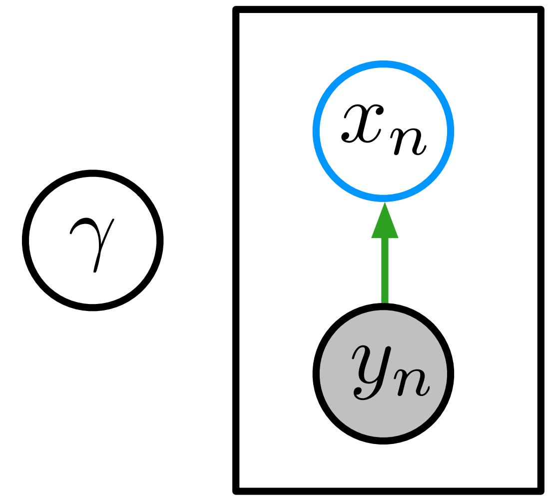
Variational autoencoders and amortized inference [1,2]

[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

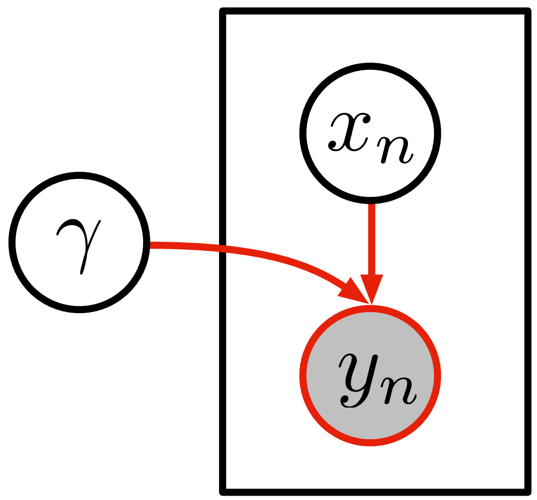


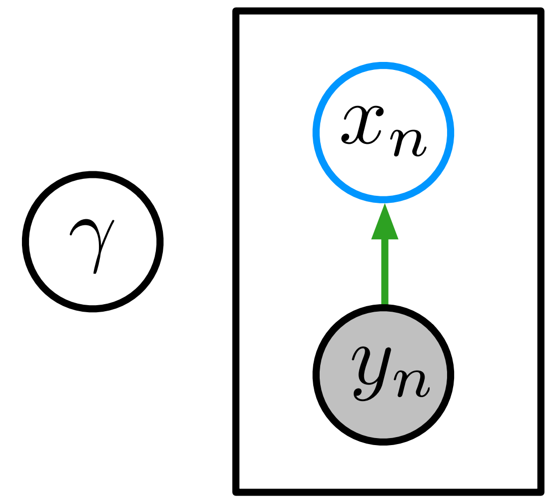
γ
x_n
y_n

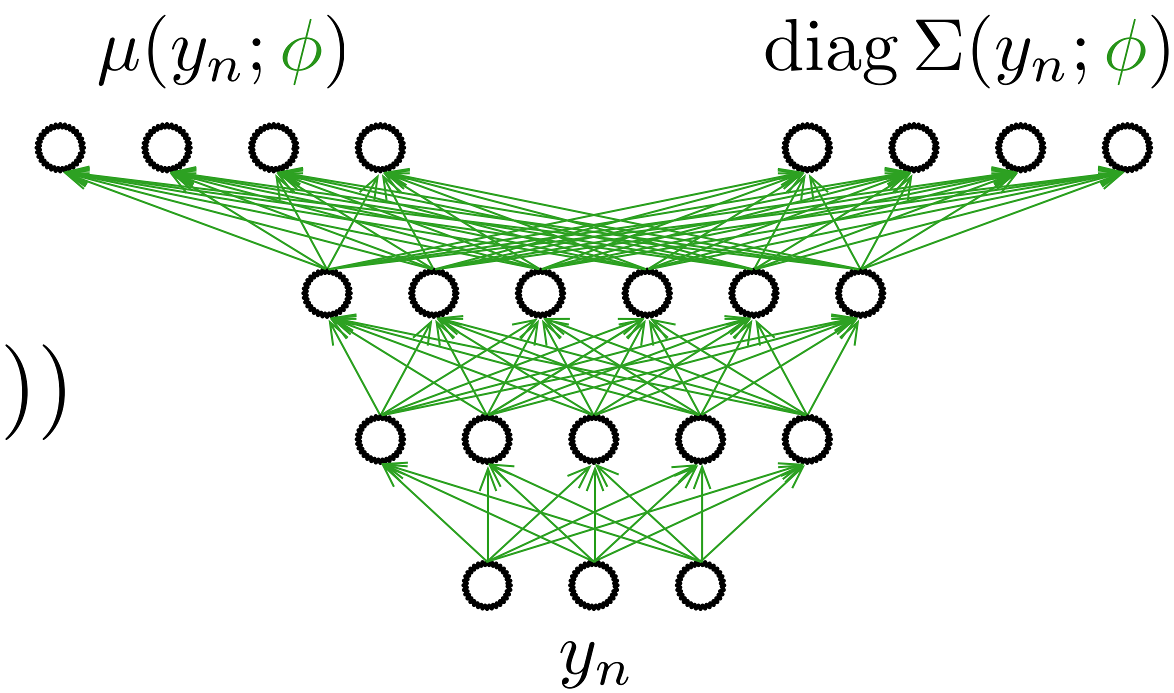




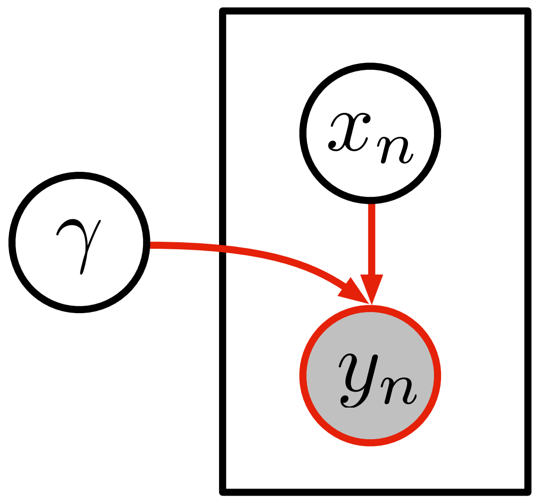
$$q^\star(x_n) \triangleq \mathcal{N}(x_n \mid \mu(y_n; \phi), \, \Sigma(y_n; \phi))$$

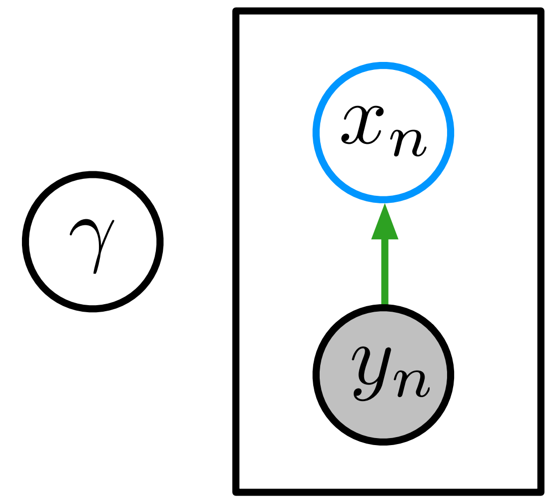


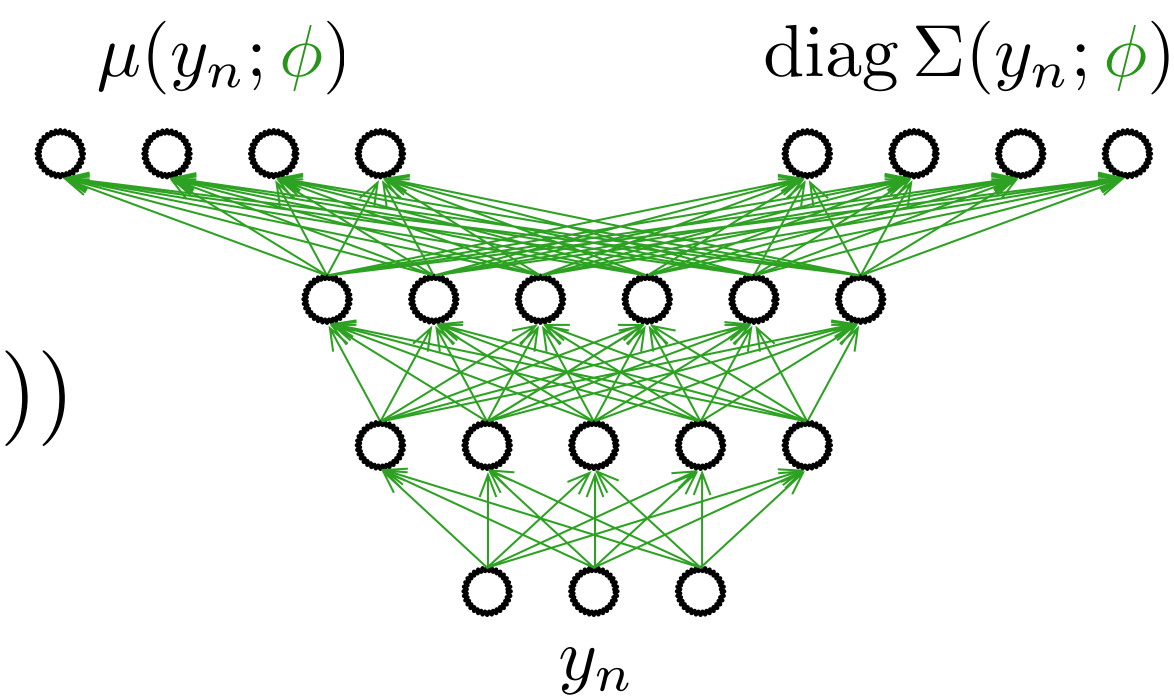




$$q^{\star}(x_n) \triangleq \mathcal{N}(x_n \mid \mu(y_n; \phi),\ \Sigma(y_n; \phi))$$







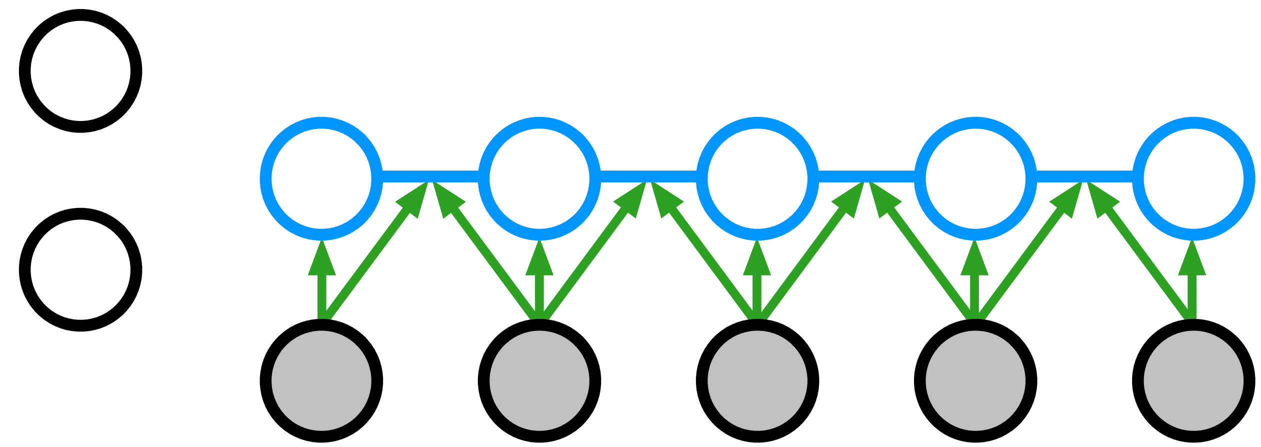
$$q^\star(x_n) \triangleq \mathcal{N}(x_n \mid \mu(y_n; \phi), \ \Sigma(y_n; \phi))$$

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

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

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

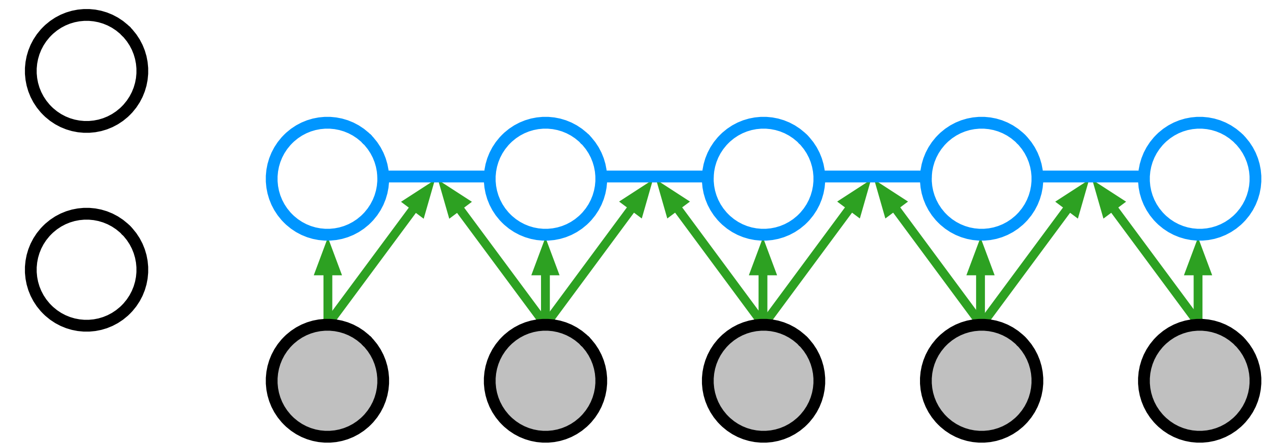
$$J_{t,t+1}(y_t, y_{t+1}; \phi_B)$$



[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.

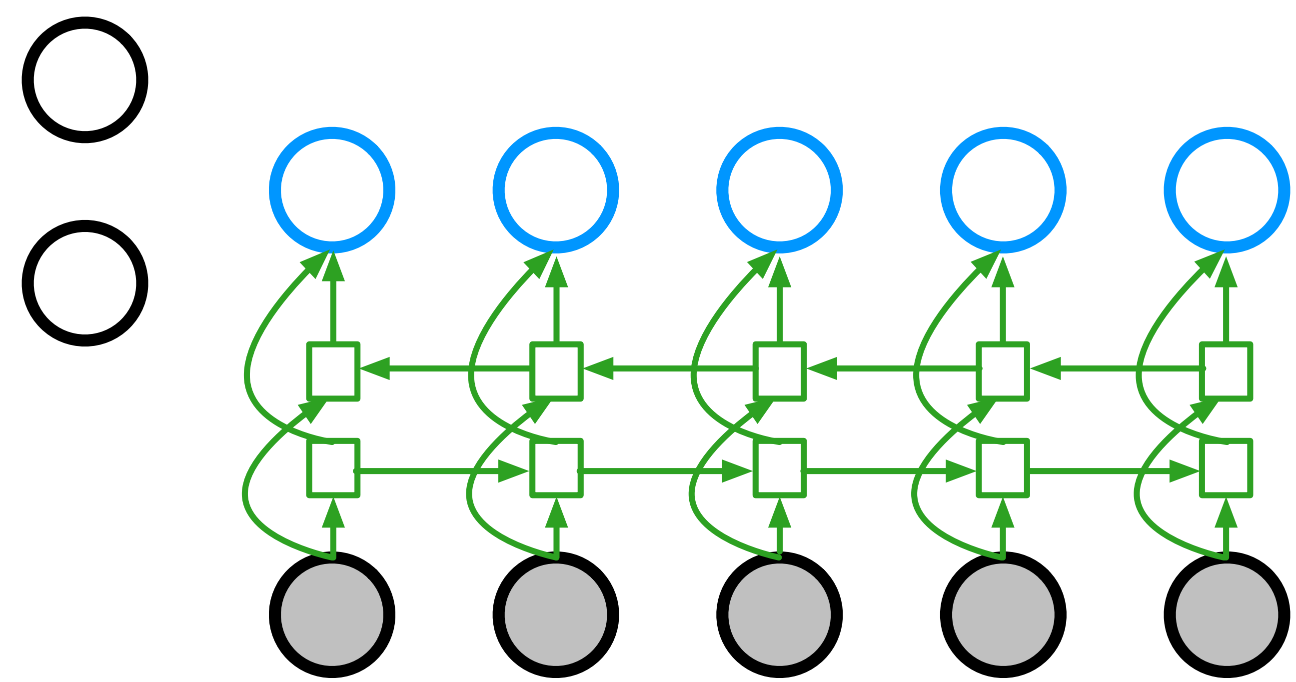
[1,2]

$$\mu_t(y_t; \phi_\mu)$$
$$J_{t,t}(y_t; \phi_D)$$
$$J_{t,t+1}(y_t, y_{t+1}; \phi_B)$$

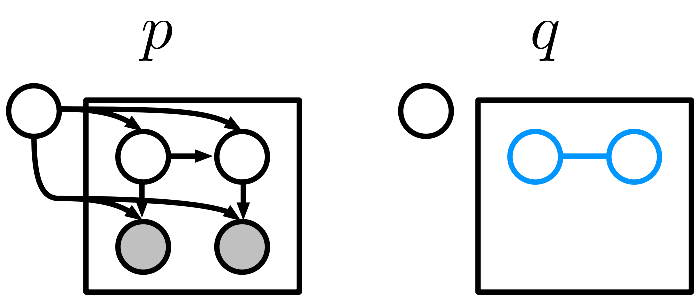


[3]

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

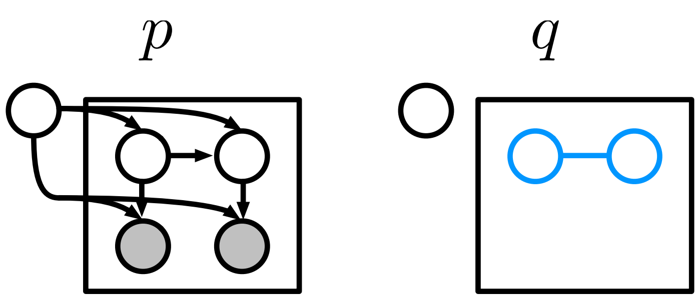


[1] Archer, Park, Buesing, Cunningham, Paninski. Black box variational inference for state space models. ICLR 2016 Workshops.
[2] Gao*, Archer*, Paninski, Cunningham. Linear dynamical neural population models through nonlinear embeddings. NIPS 2016.
[3] Krishnan, Shalit, Sontag. Structured inference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



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

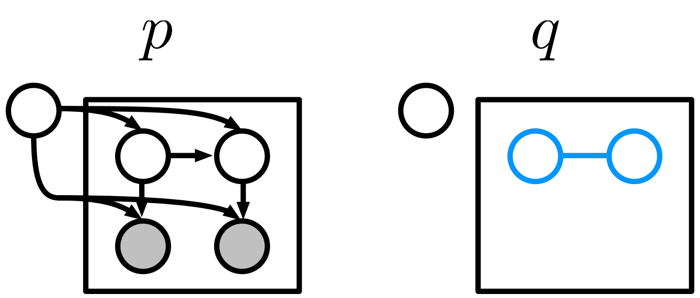
Natural gradient SVI



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

Natural gradient SVI

– expensive for general obs.

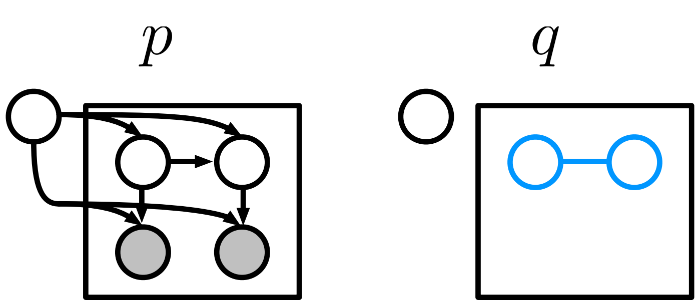


$$q^*(x) \triangleq \arg\max_{q(x)} \mathcal{L}[\,q(\theta)q(x)\,]$$

Natural gradient SVI

− expensive for general obs.

+ optimal local factor



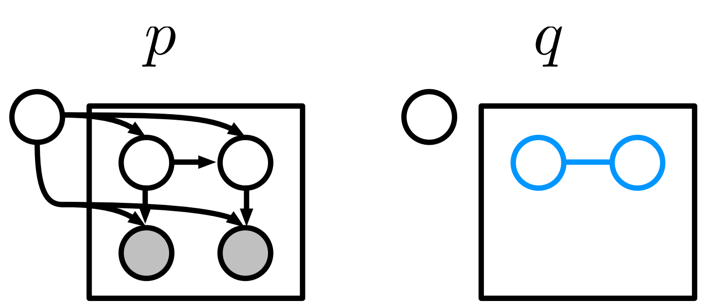
$$q^*(x) \triangleq \underset{q(x)}{\arg\max}\, \mathcal{L}[\, q(\theta)q(x)\,]$$

Natural gradient SVI

– expensive for general obs.

+ optimal local factor

+ exploits conj. graph structure



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

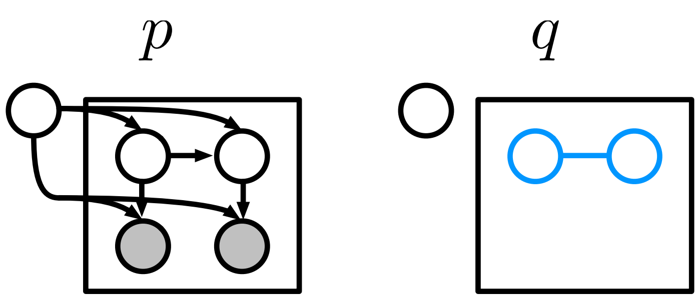
Natural gradient SVI

– expensive for general obs.

+ optimal local factor

+ exploits conj. graph structure

+ arbitrary inference queries



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

Natural gradient SVI

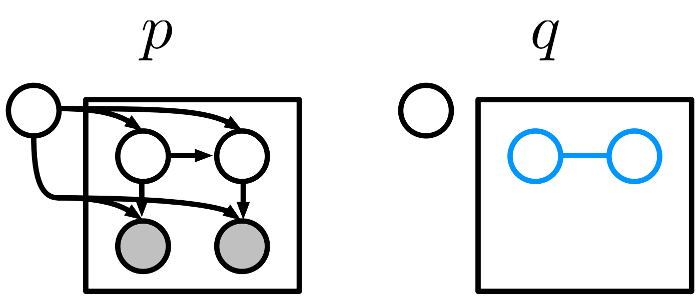
– expensive for general obs.

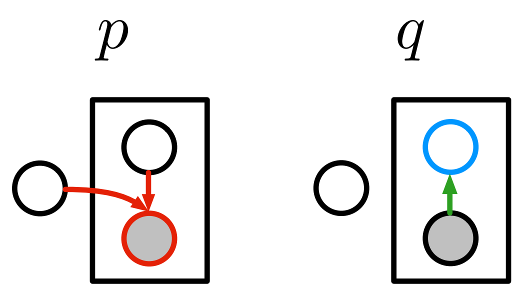
+ optimal local factor

+ 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

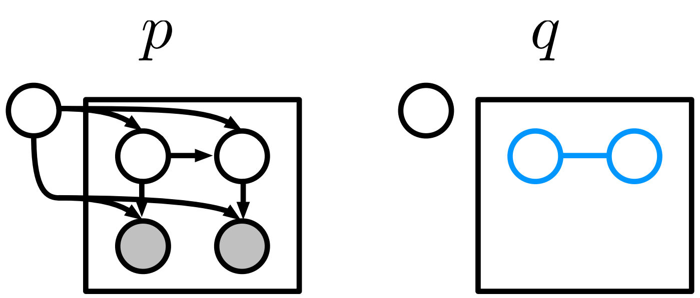
– expensive for general obs.

\+ optimal local factor

\+ exploits conj. graph structure

\+ arbitrary inference queries

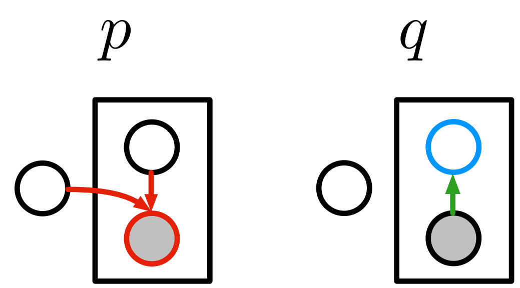
\+ natural gradients



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

Natural gradient SVI

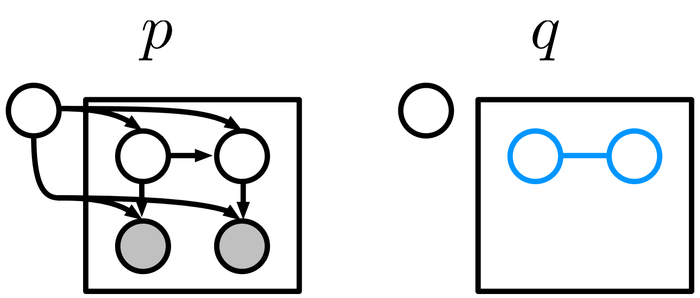
- – expensive for general obs.
- + optimal local factor
- + exploits conj. graph structure
- + arbitrary inference queries
- + natural gradients



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

Variational autoencoders

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



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

Natural gradient SVI

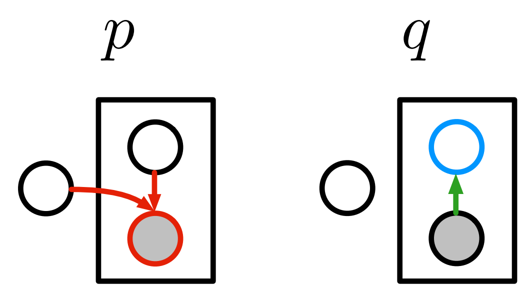
– expensive for general obs.

+ 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

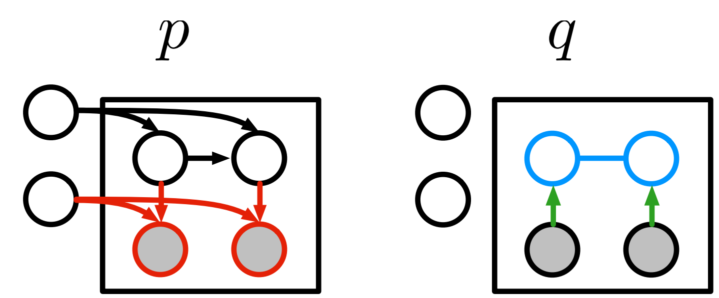
+ fast for general obs.

– suboptimal local inference

– ϕ does all local inference

– limited inference queries

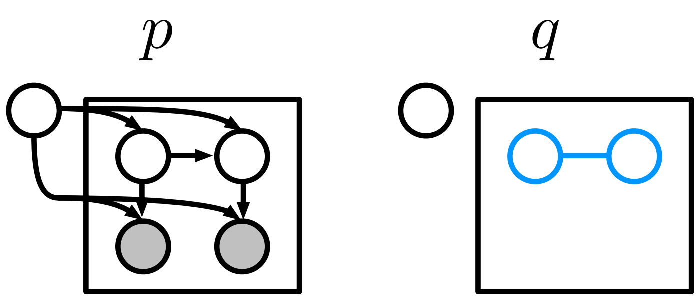
– no cheap natural gradients



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

Structured VAEs [1]

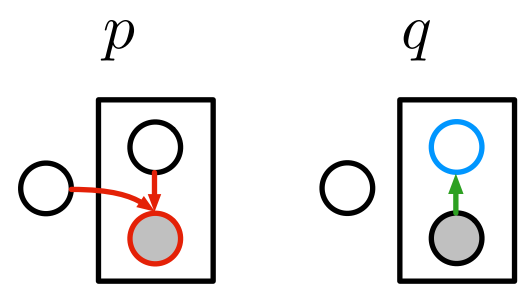
[1] **Johnson**, Duvenaud, Wiltschko, Datta, and Adams. Composing graphical models and neural networks. NIPS 2016.



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

Natural gradient SVI

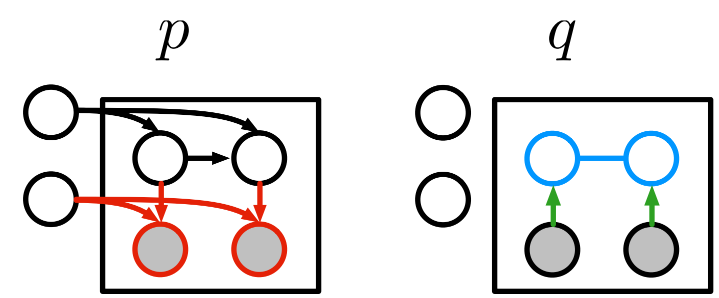
- – expensive for general obs.
- + optimal local factor
- + exploits conj. graph structure
- + arbitrary inference queries
- + natural gradients



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

Variational autoencoders

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



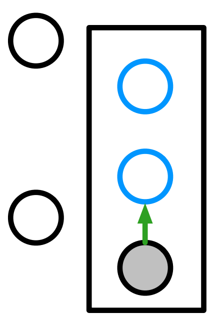
$$q^*(x) \triangleq \;?$$

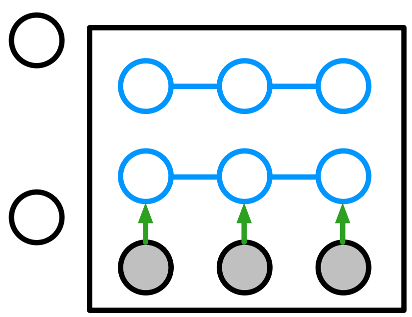
Structured VAEs [1]

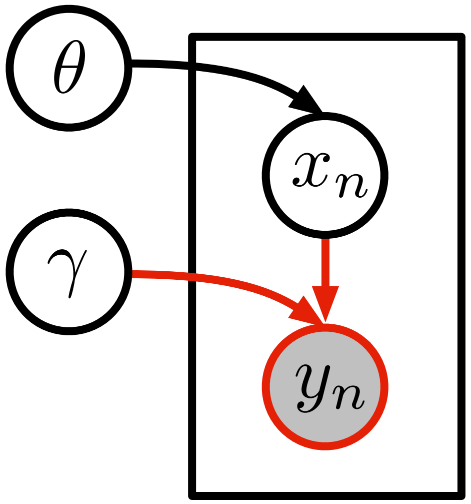
- + fast for general obs.
- ± optimal given conj. evidence
- + exploits conj. graph structure
- + arbitrary inference queries
- + natural gradients on η_θ

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

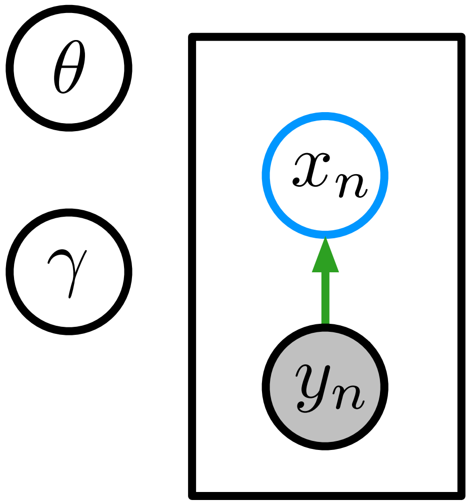
Inference: recognition networks output conjugate potentials, then apply fast graphical model inference





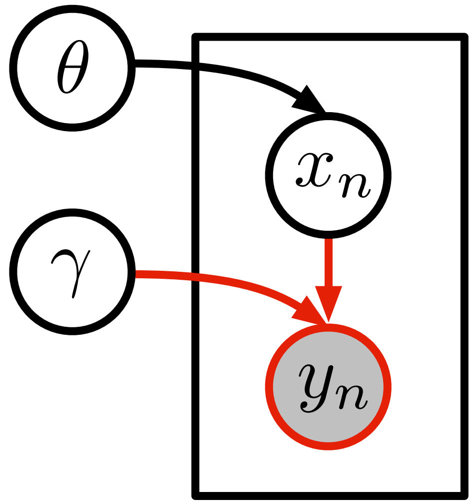


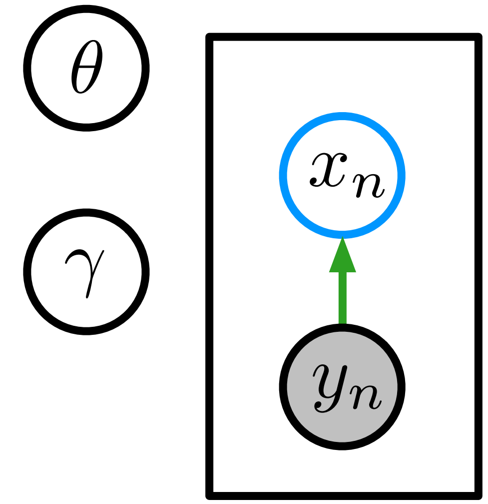
θ
x_n
γ
y_n



θ
x_n
γ
y_n

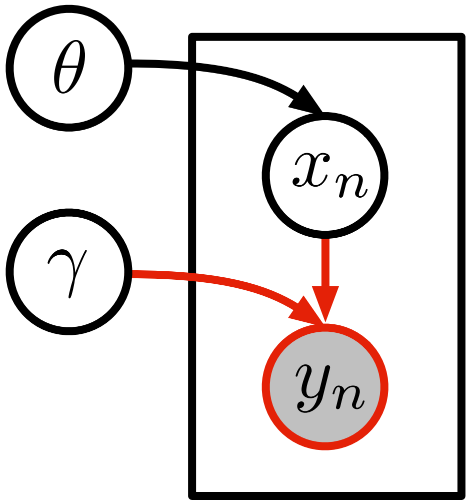
$$\mathcal{L}[\,q(\theta)q(\gamma)q(x)\,] \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y\mid x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$

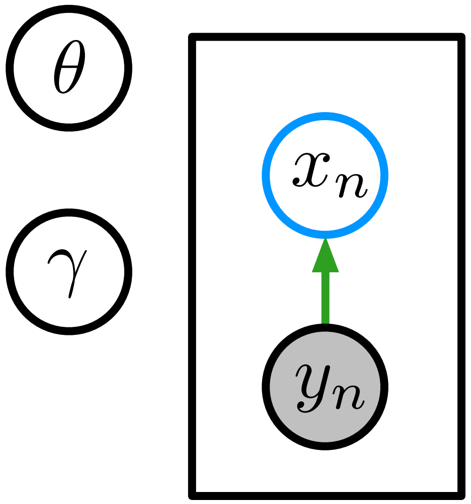




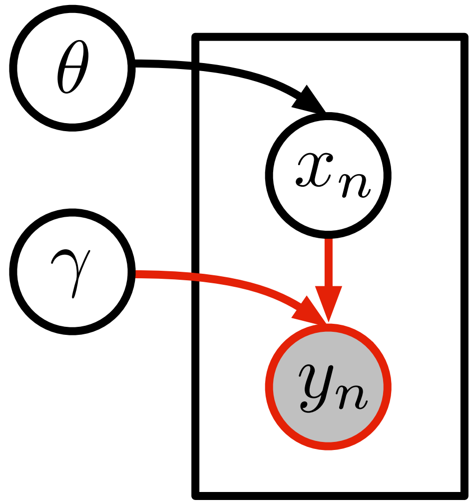
$$\mathcal{L}[\,q(\theta)q(\gamma){\color{blue}q(x)}\,] \triangleq \mathbb{E}_{q(\theta)q(\gamma){\color{blue}q(x)}}\left[\log \frac{p(\theta,\gamma,x){\color{red}p(y\,|\,x,\gamma)}}{q(\theta)q(\gamma){\color{blue}q(x)}}\right]$$

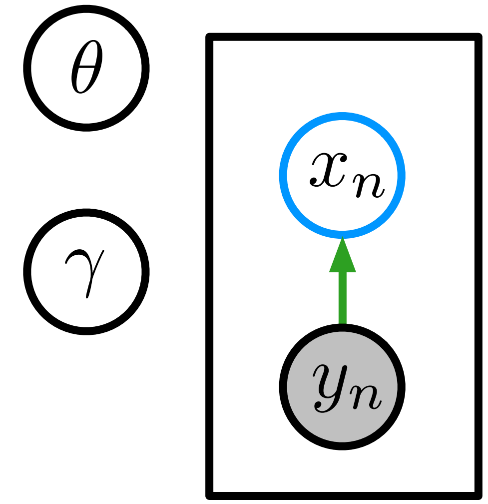
$$q(\theta) \leftrightarrow \eta_\theta \qquad q(\gamma) \leftrightarrow \eta_\gamma \qquad {\color{blue}q(x)} \leftrightarrow {\color{blue}\eta_x}$$





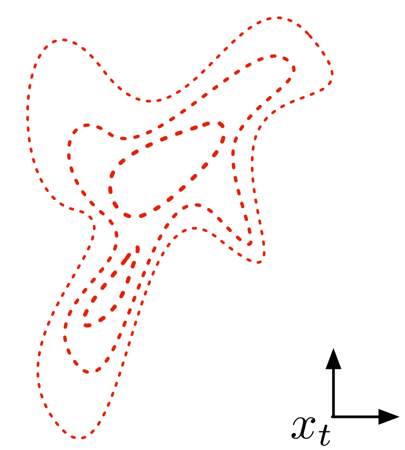
$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y \mid x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$

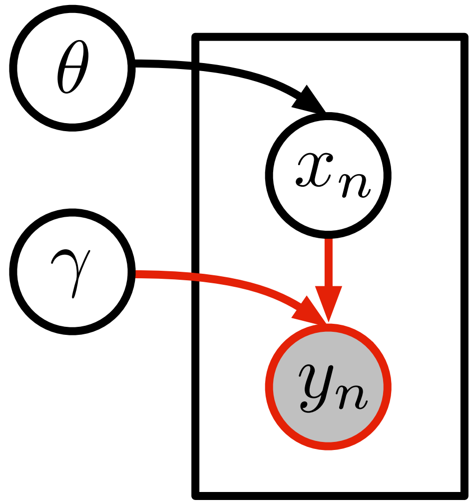


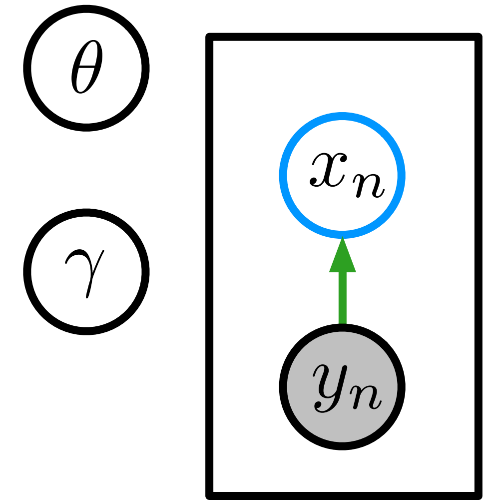


$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y \mid x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$

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

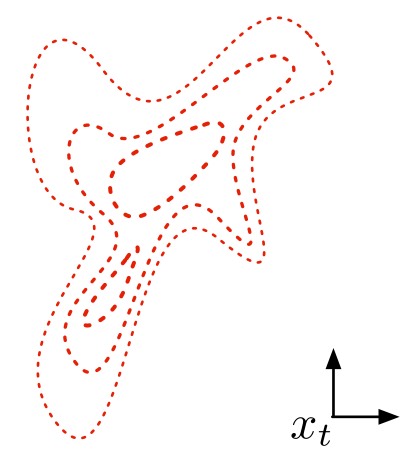






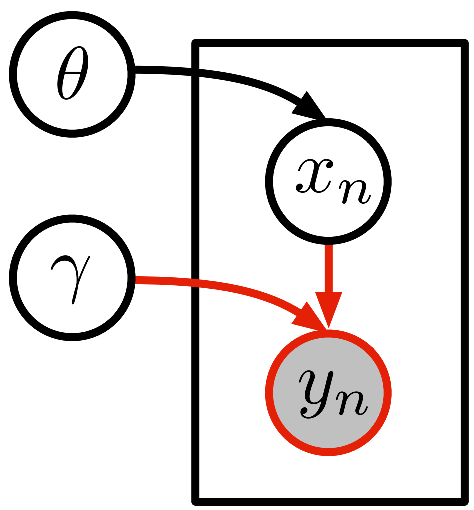
$$\mathbb{E}_{q(\gamma)} \log p(y_t \mid x_t, \gamma)$$

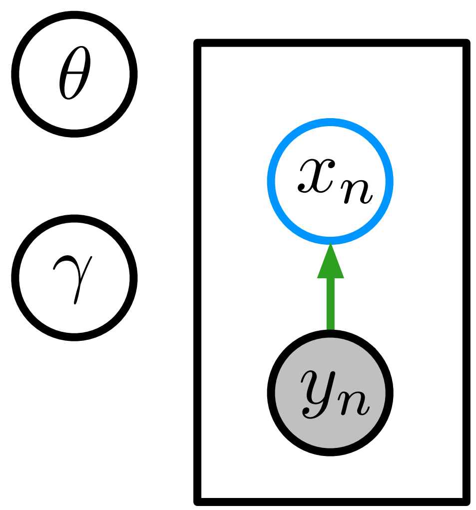
$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y \mid x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$



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

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

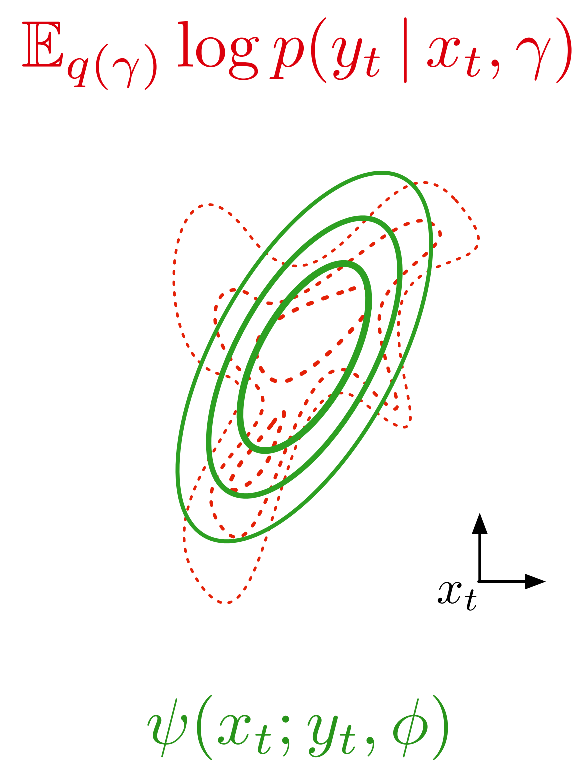


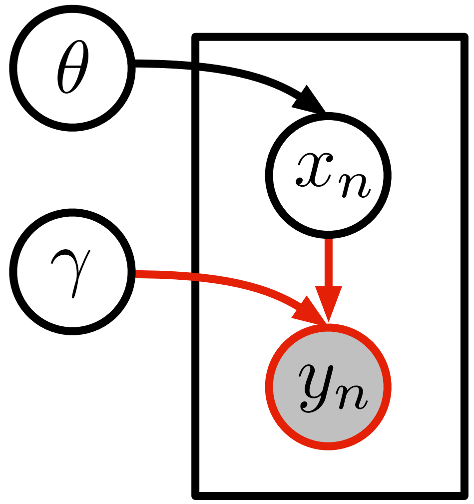


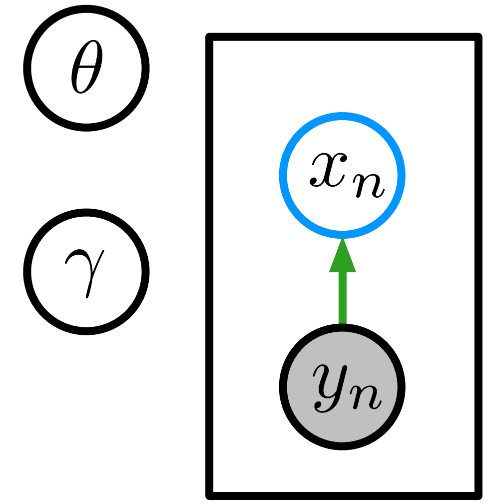
$$\mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y \mid x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$

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

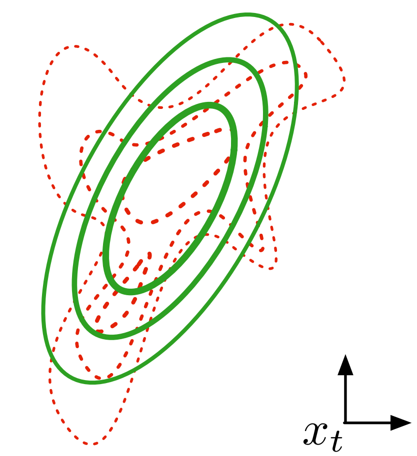
where $\psi(x; y, \phi)$ is a conjugate potential for $p(x \mid \theta)$







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



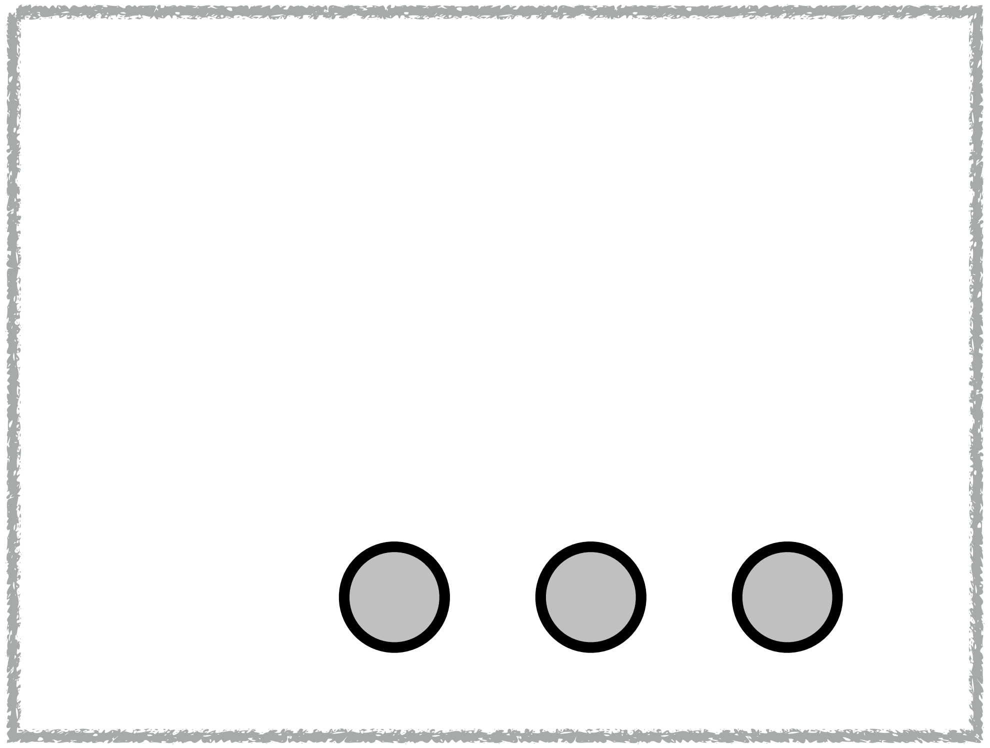
$$\widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)\exp\{\psi(x;y,\phi)\}}{q(\theta)q(\gamma)q(x)}\right]$$

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

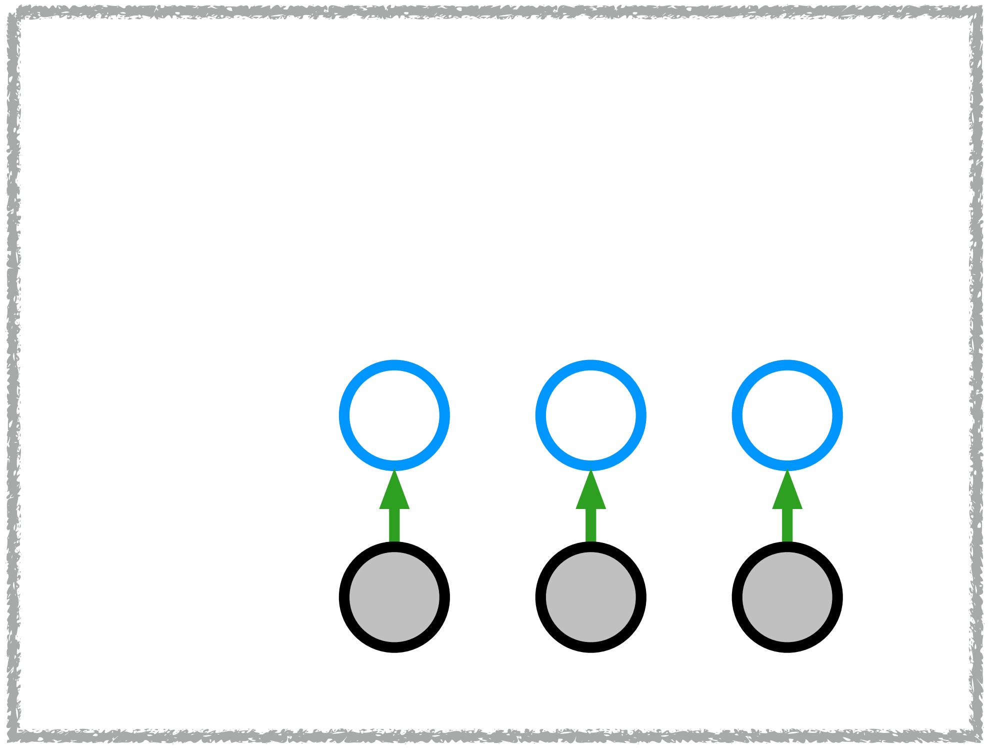
$$\eta_x^*(\eta_\theta, \phi) \triangleq \arg\max_{\eta_x} \widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \qquad \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi))$$

Step 1: apply recognition network

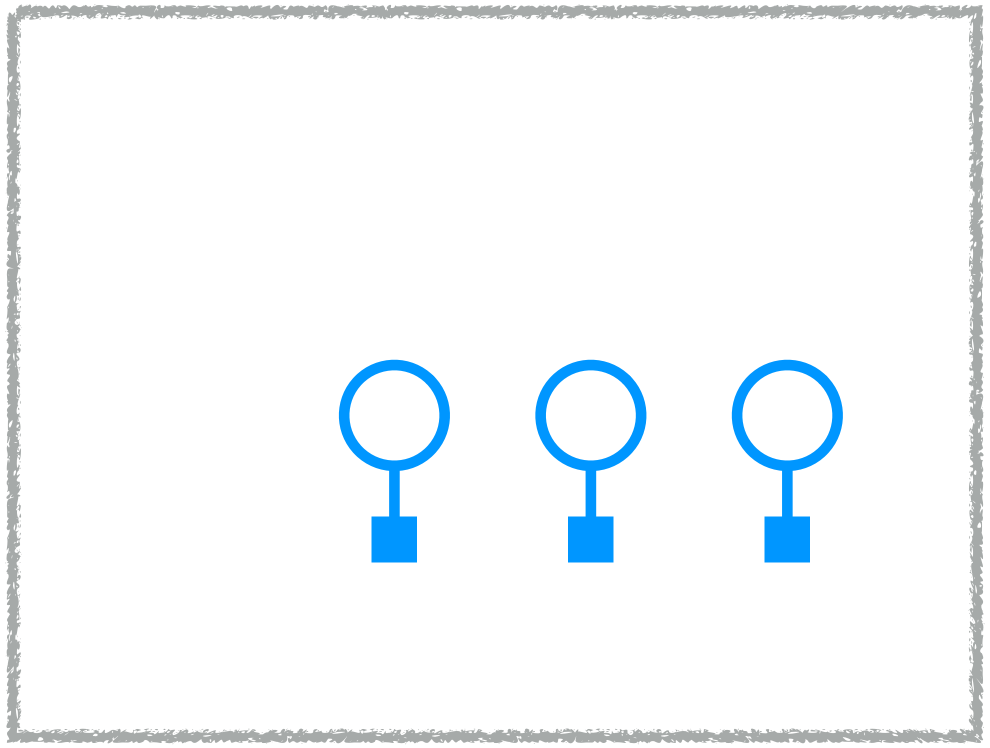
Step 1: apply recognition network



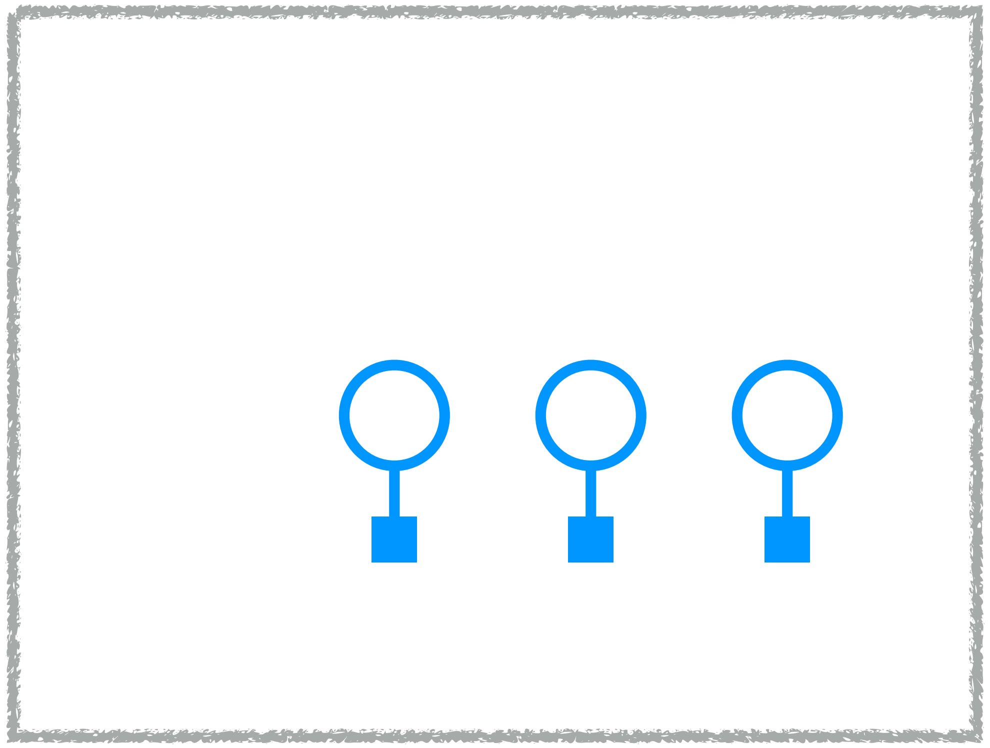
Step 1: apply recognition network



Step 1: apply recognition network

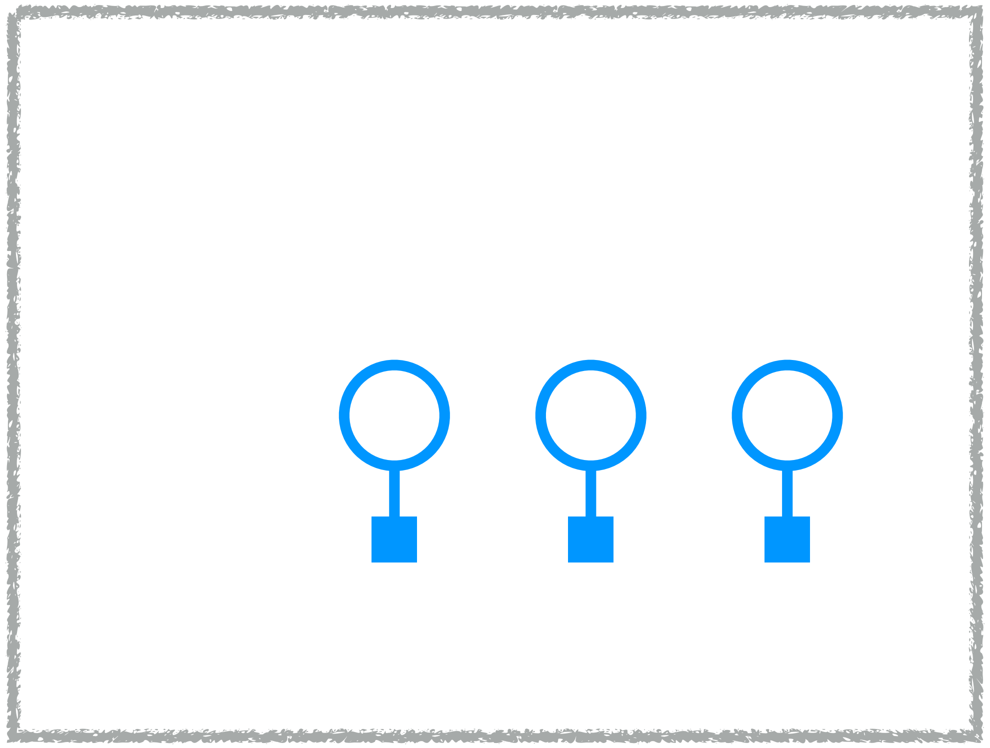


Step 1: apply recognition network

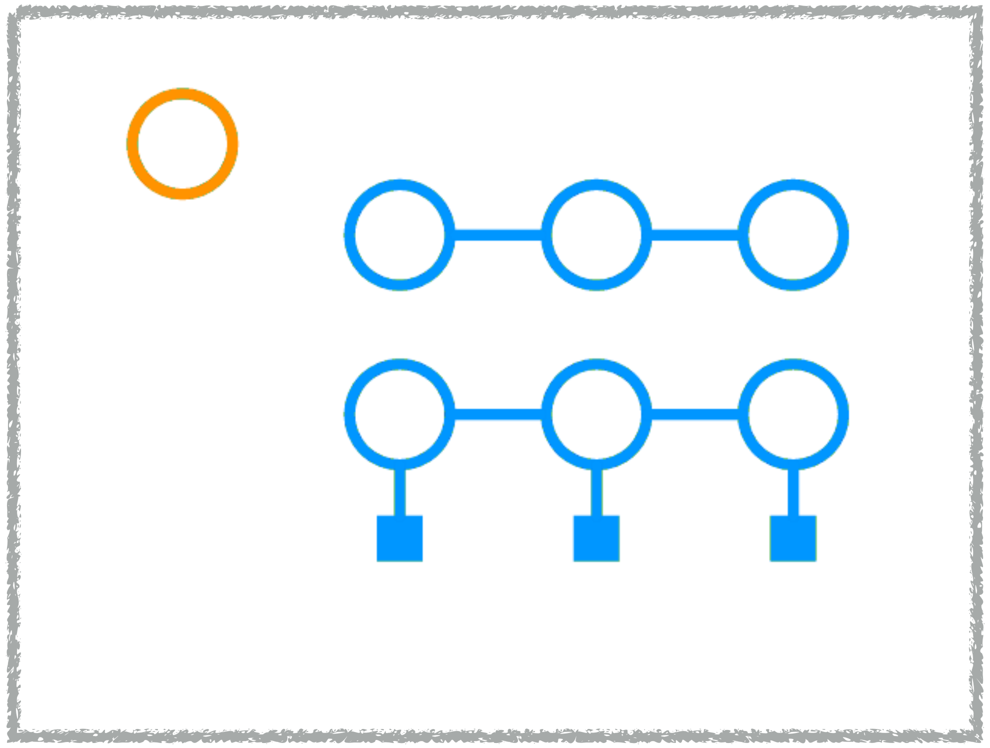


Step 2: run fast PGM algorithms

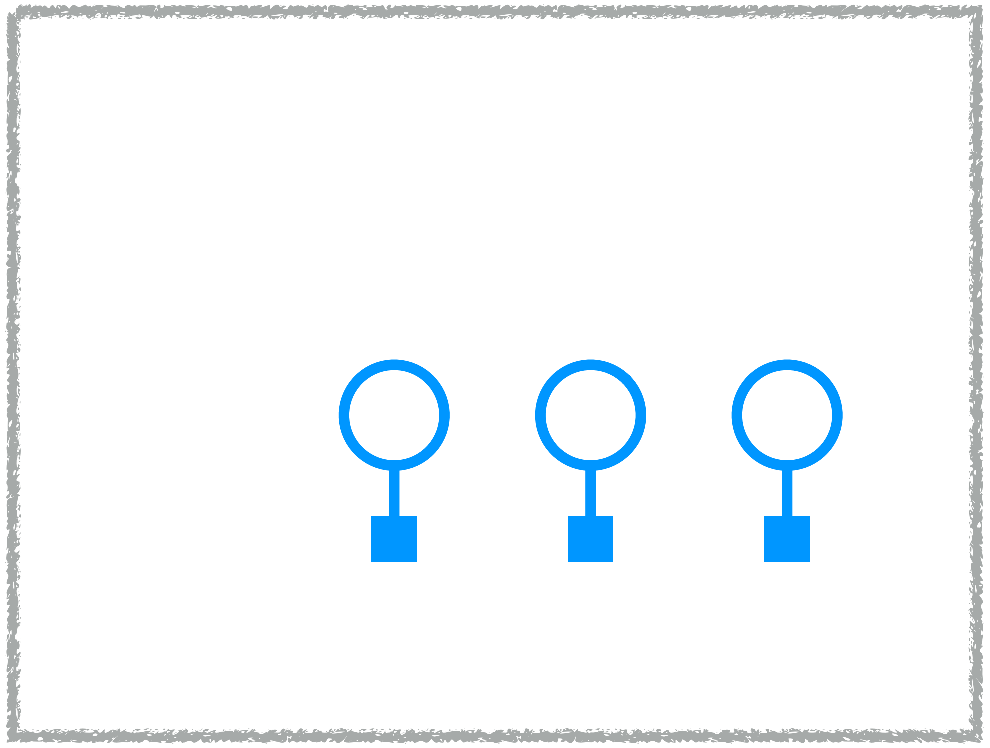
Step 1: apply recognition network



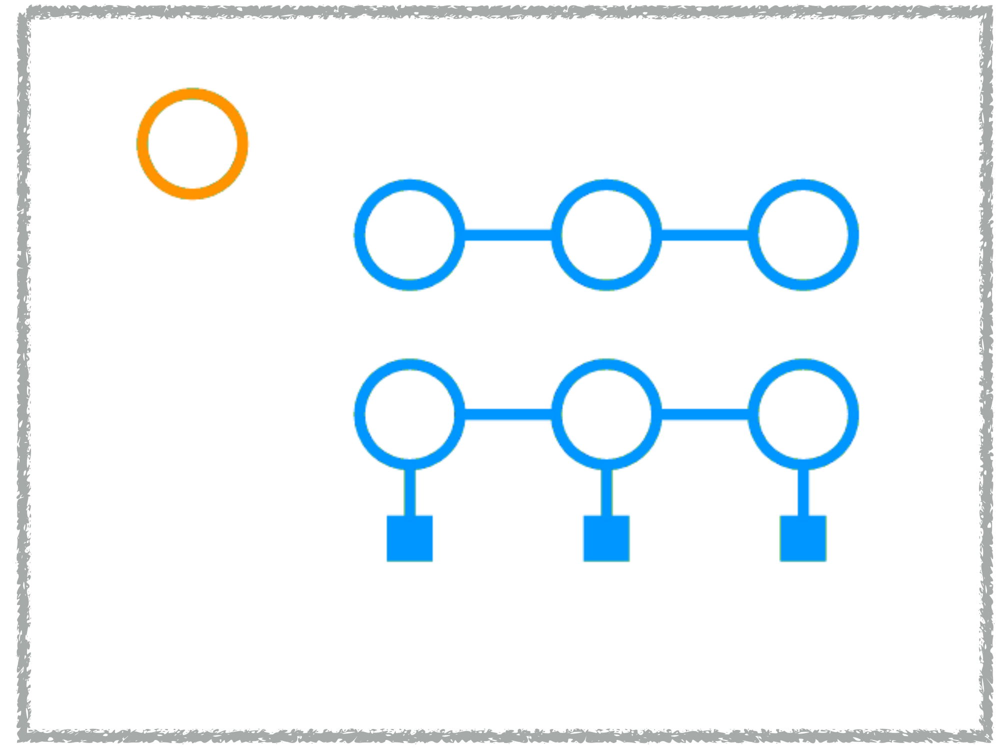
Step 2: run fast PGM algorithms



Step 1: apply recognition network

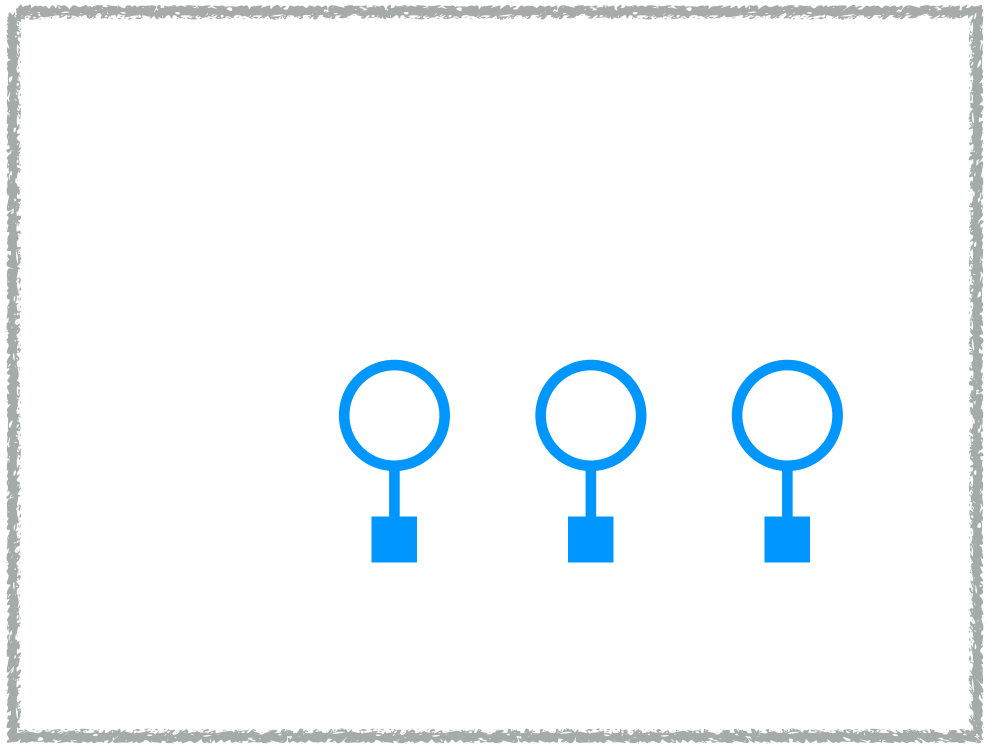


Step 2: run fast PGM algorithms

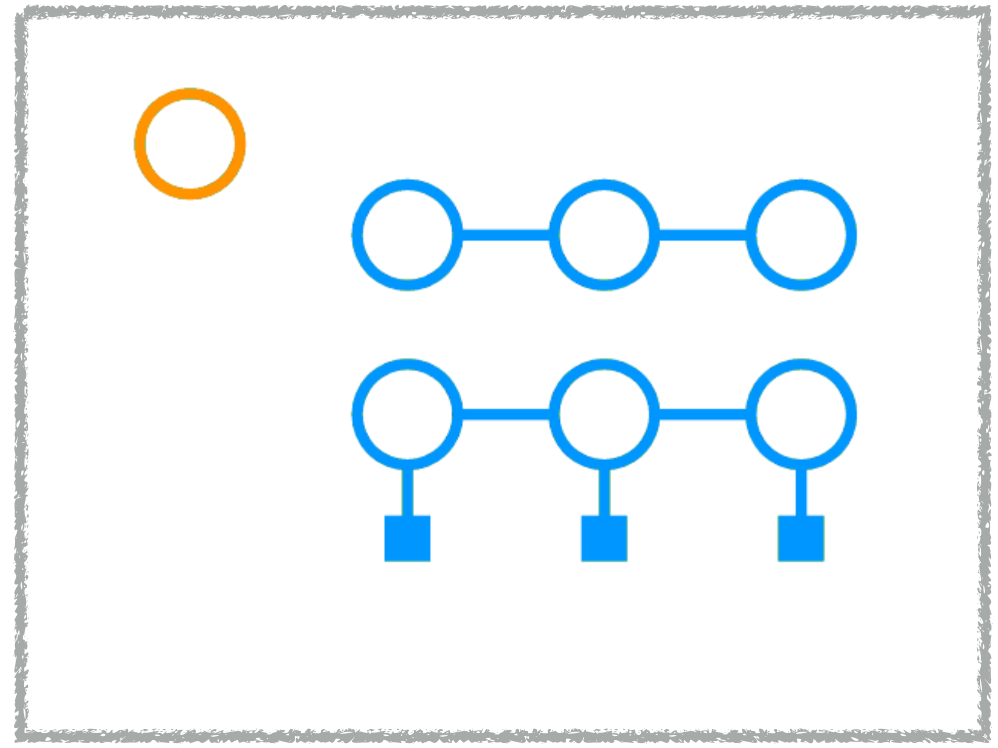


Step 3: sample, compute flat grads

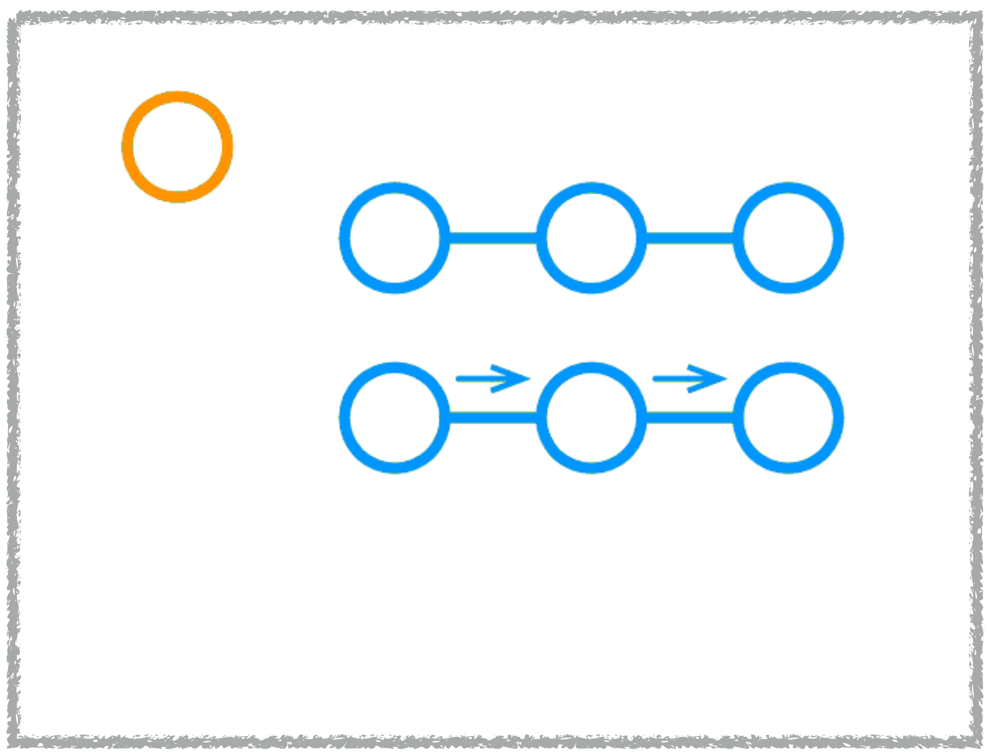
Step 1: apply recognition network



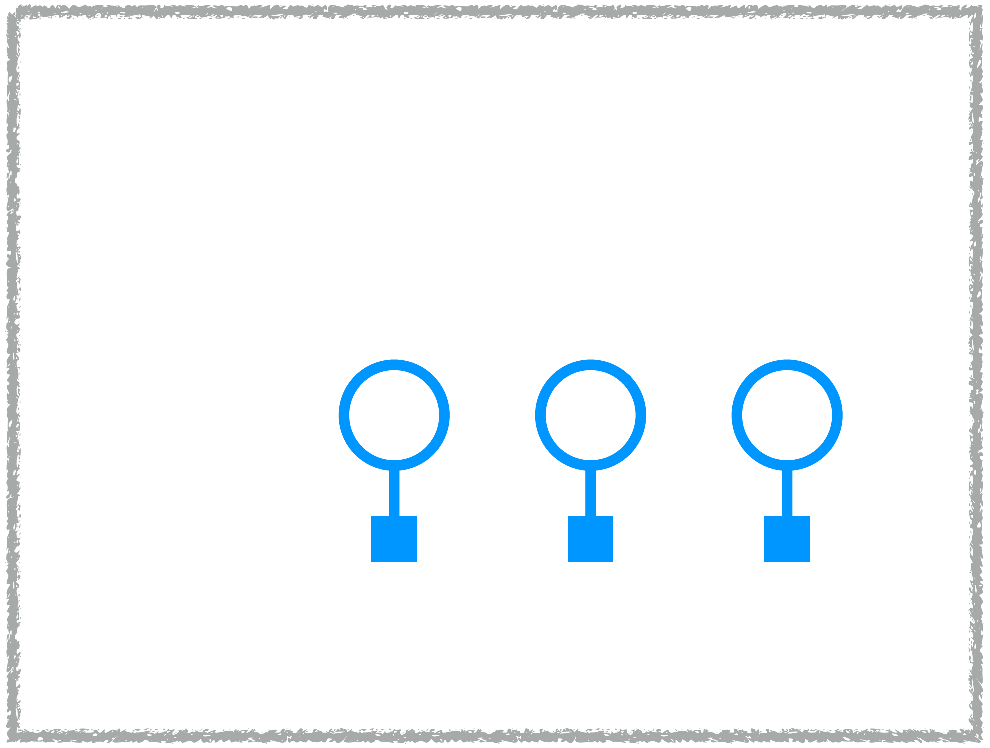
Step 2: run fast PGM algorithms



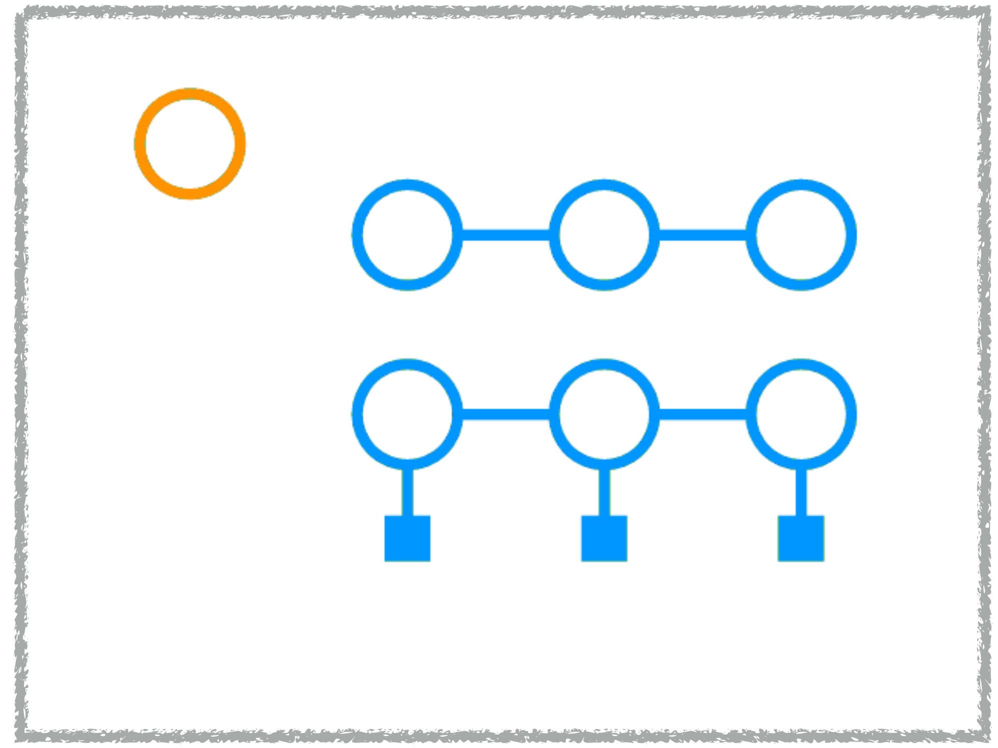
Step 3: sample, compute flat grads



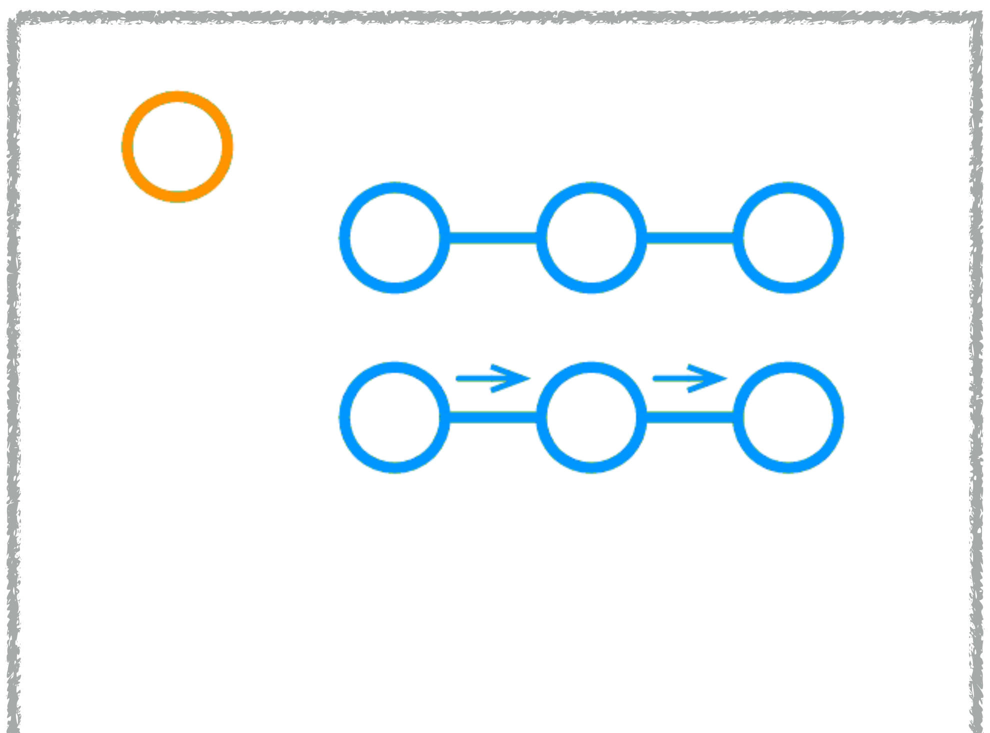
Step 1: apply recognition network



Step 2: run fast PGM algorithms

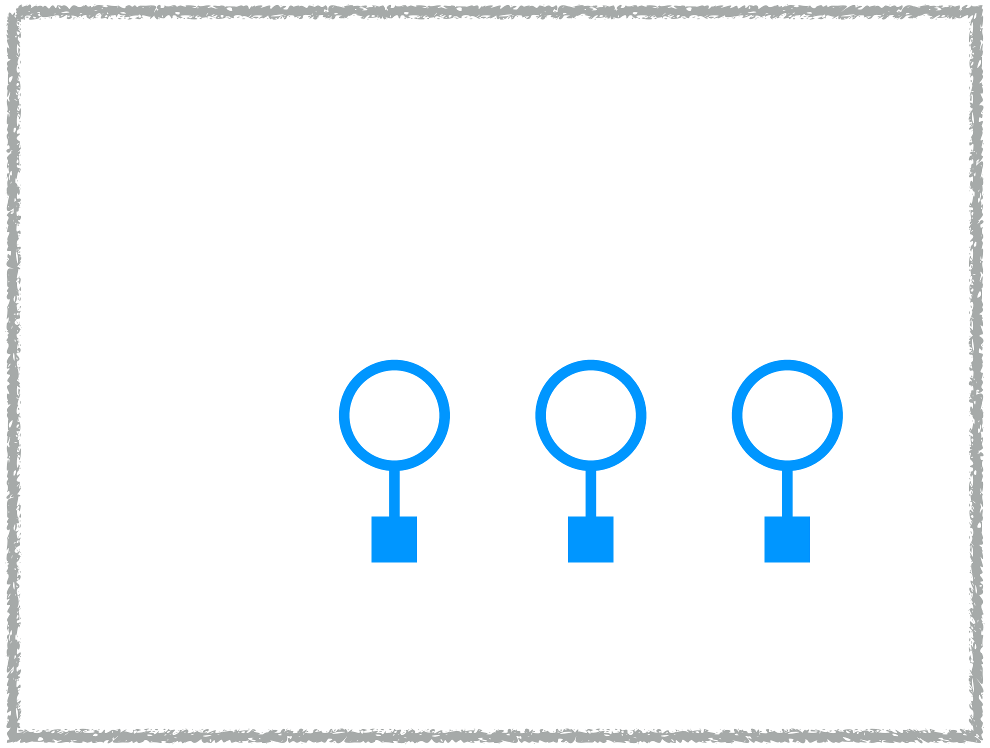


Step 3: sample, compute flat grads

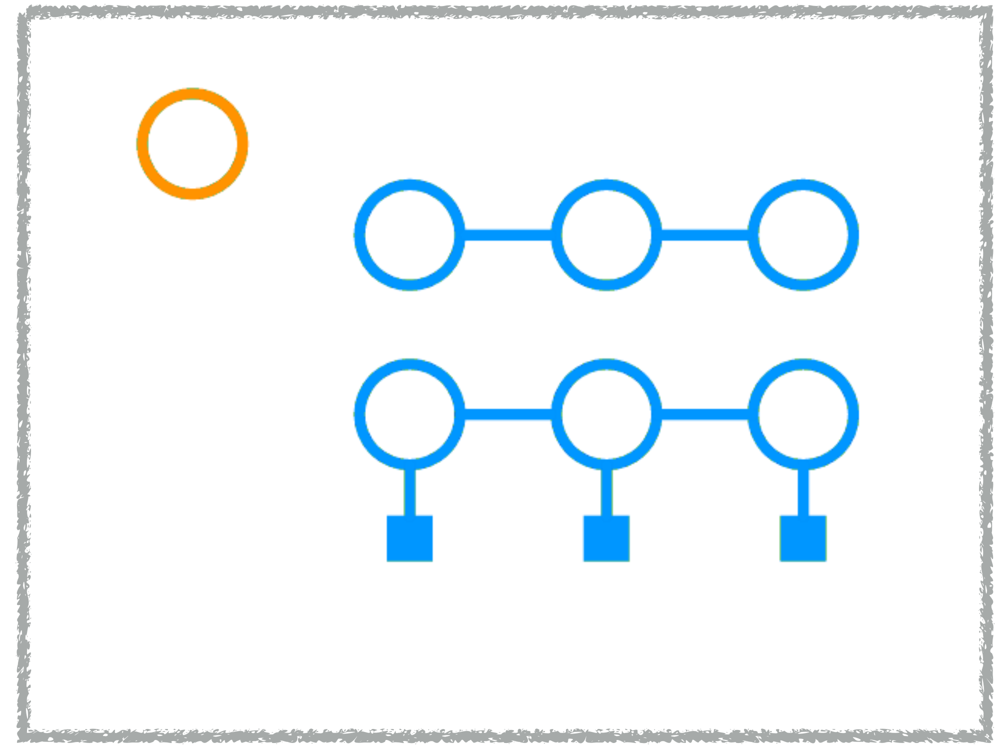


Step 4: compute natural gradient

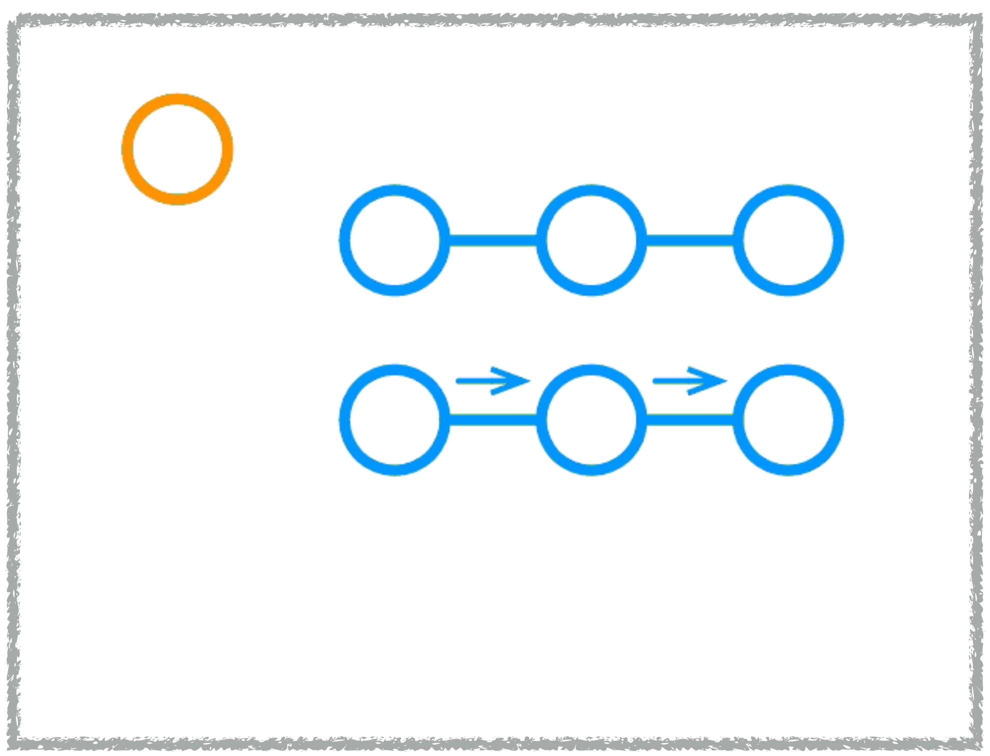
Step 1: apply recognition network



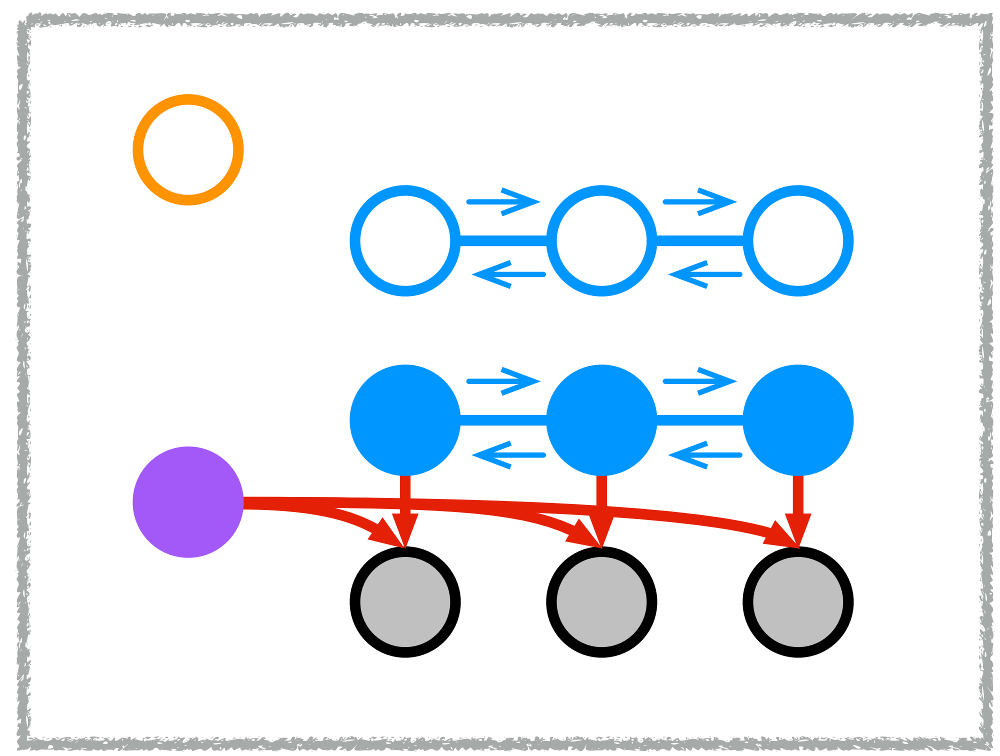
Step 2: run fast PGM algorithms

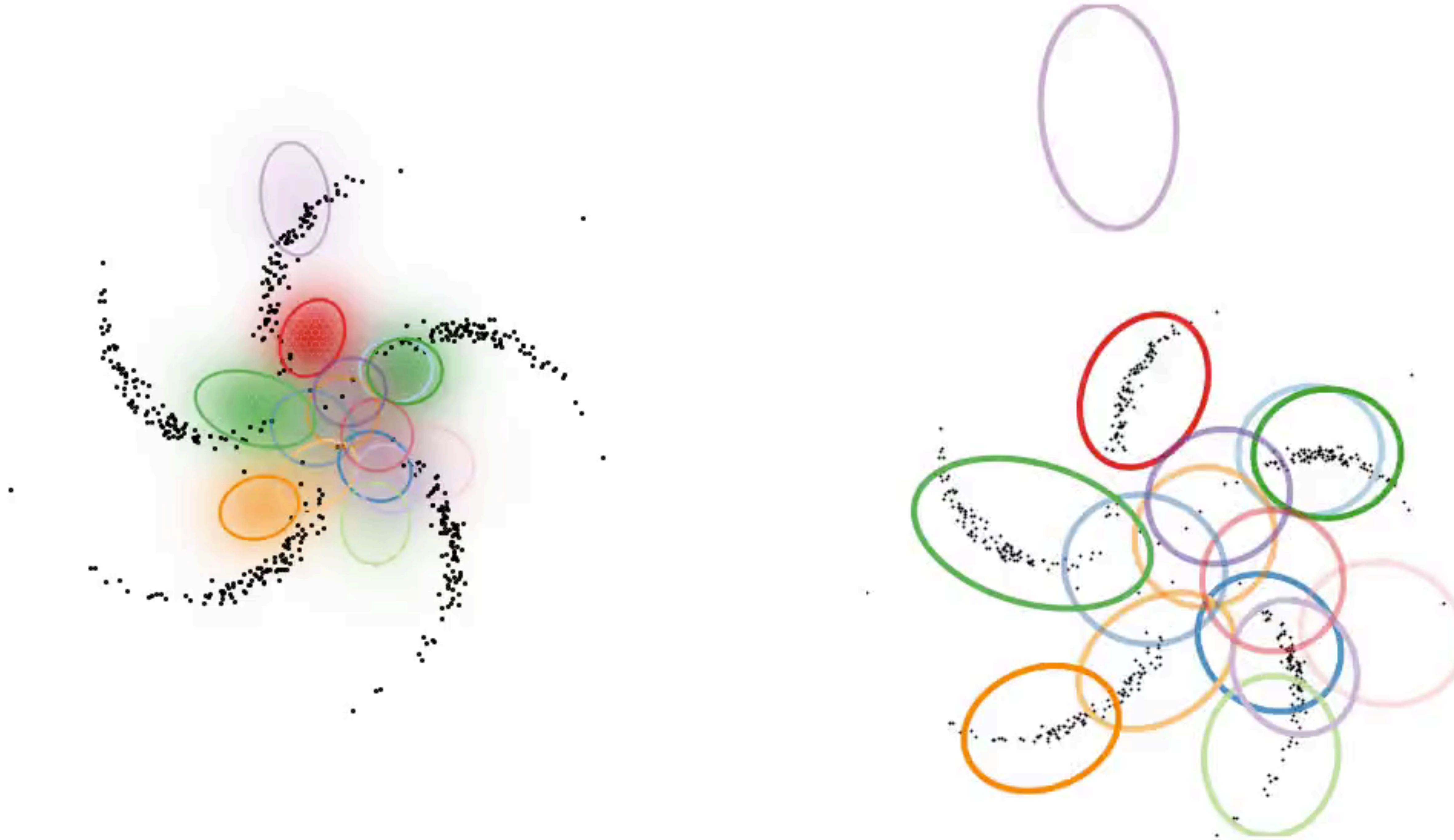


Step 3: sample, compute flat grads

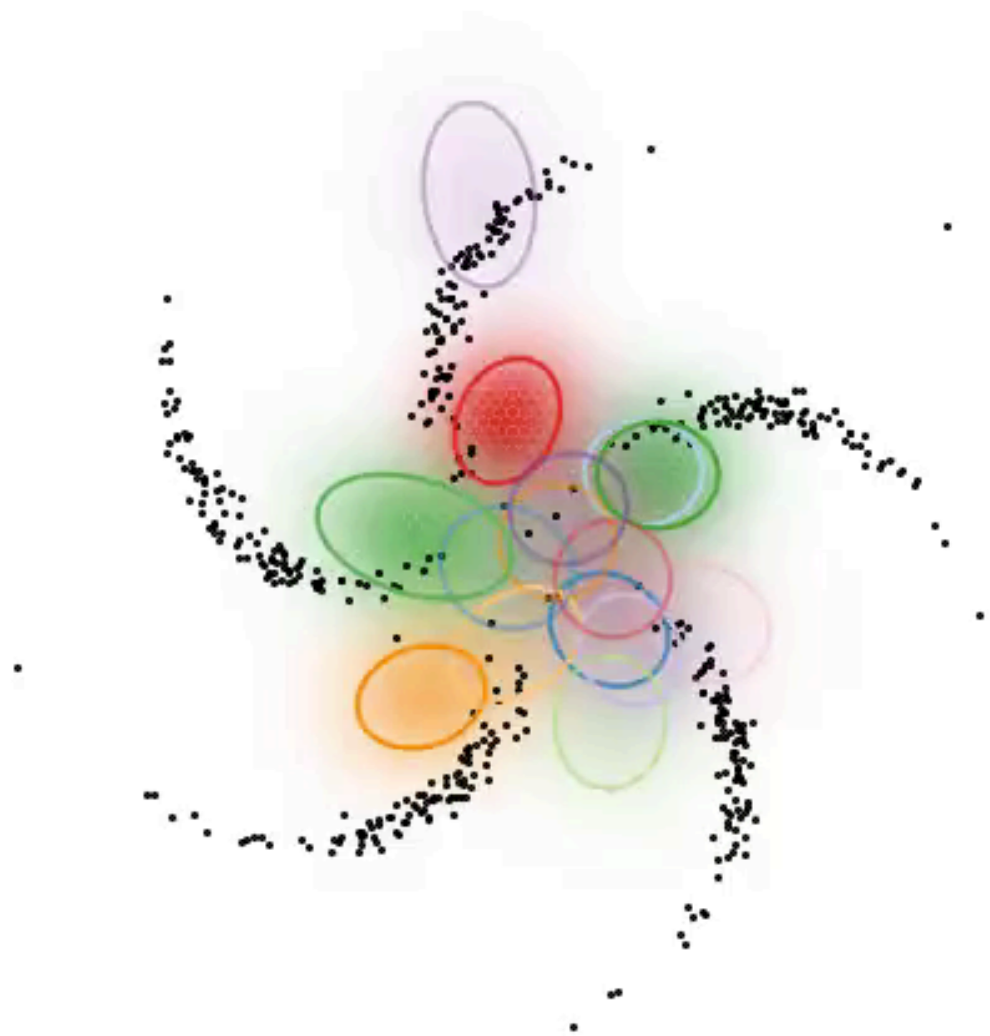


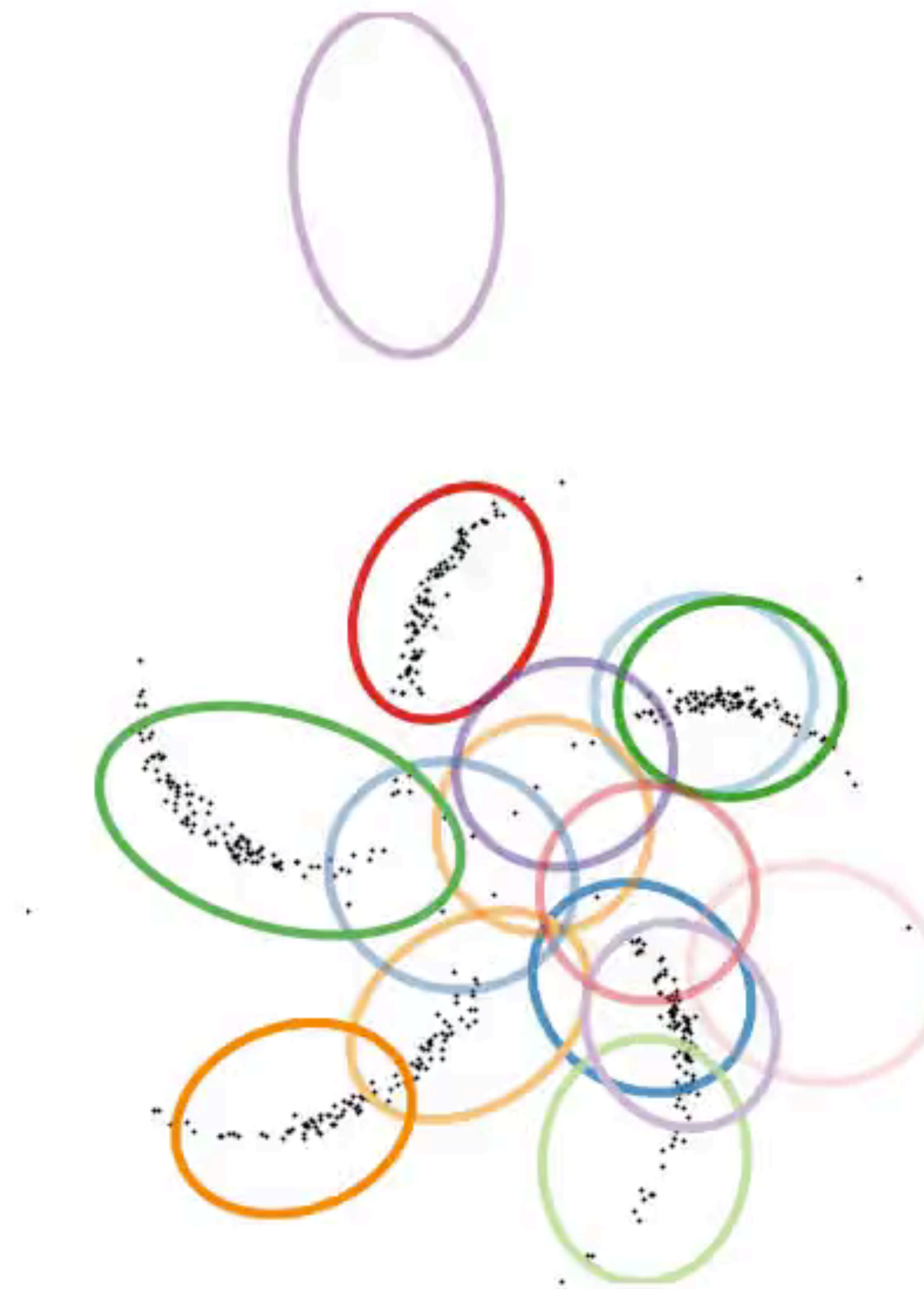
Step 4: compute natural gradient

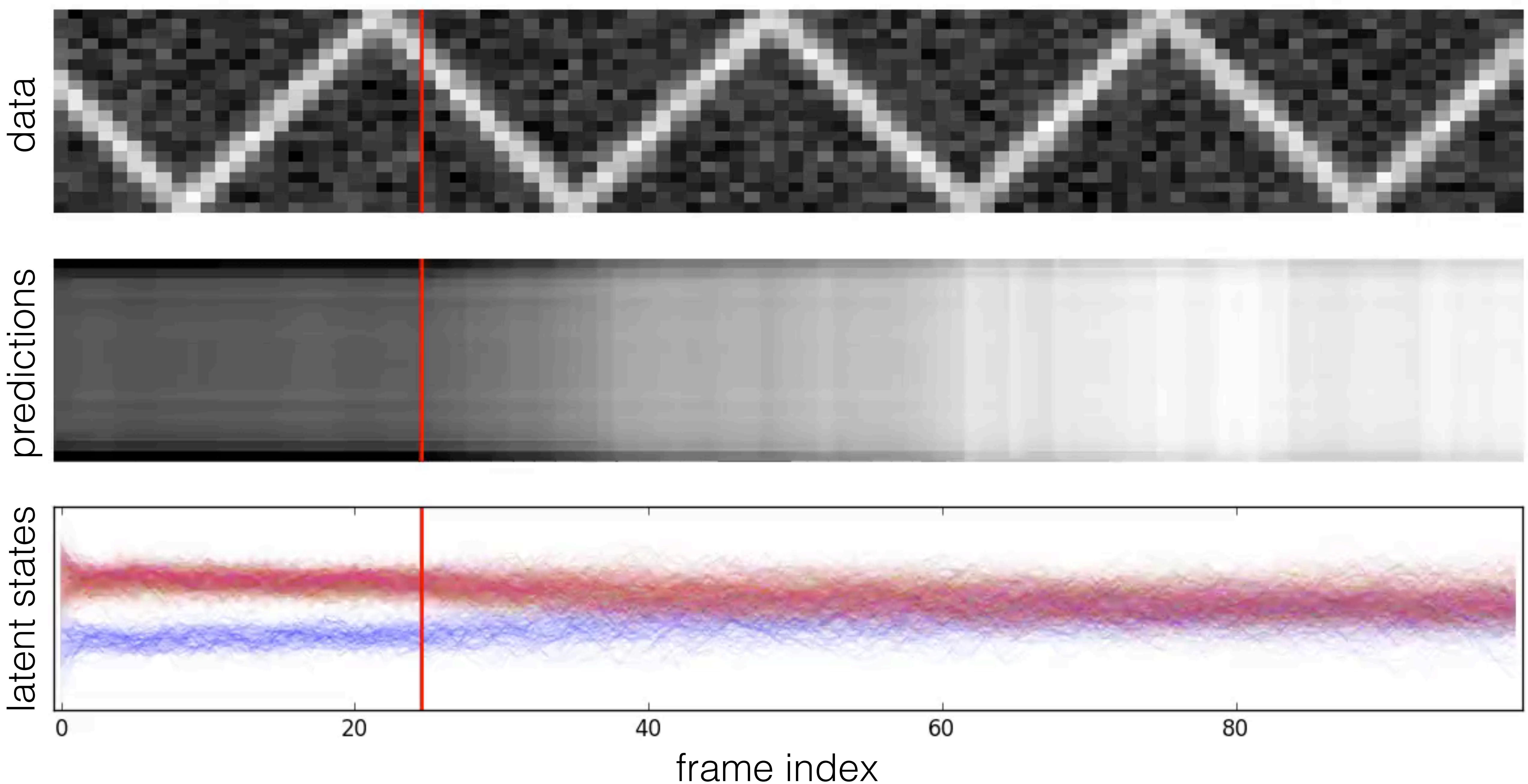




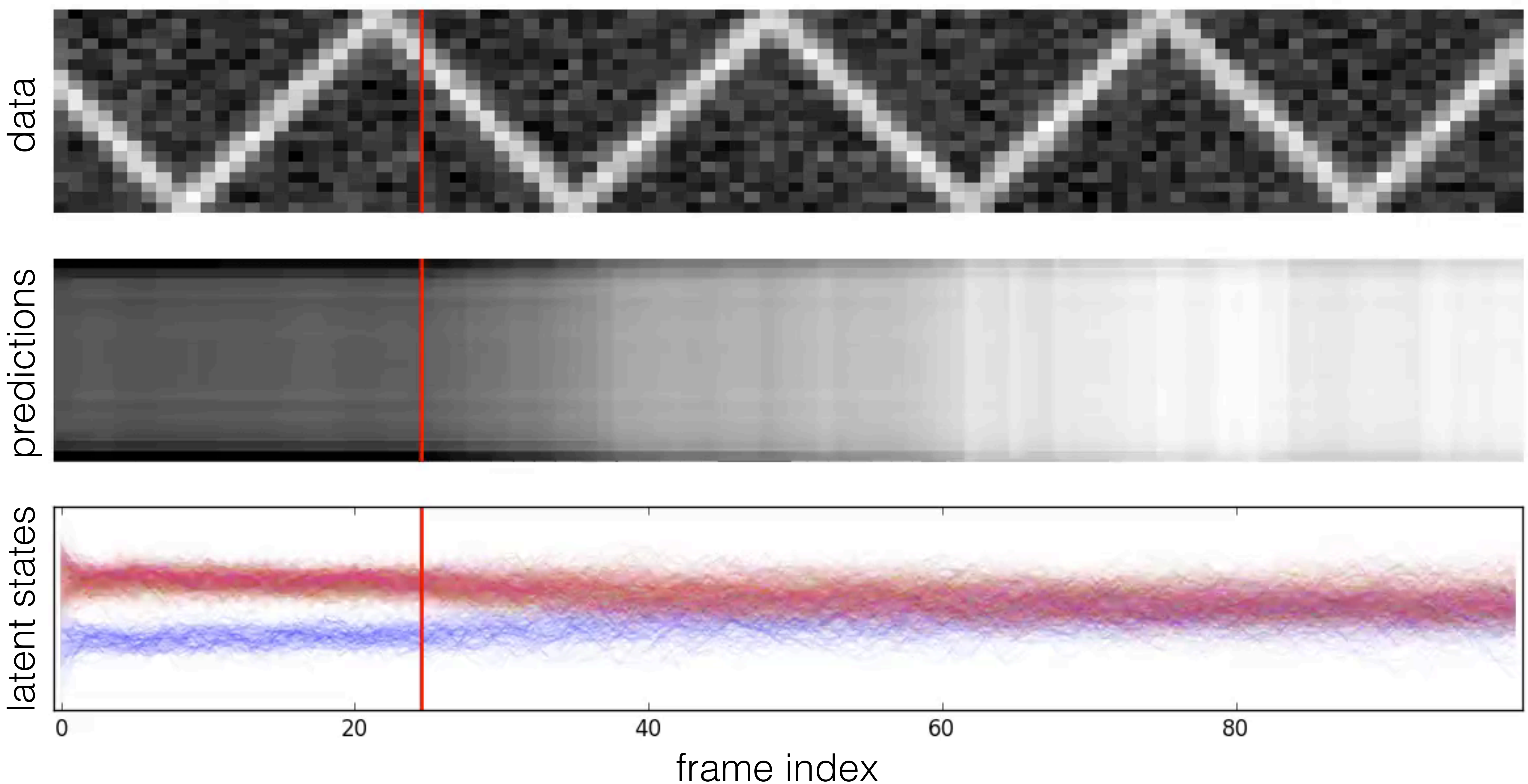
data space latent space

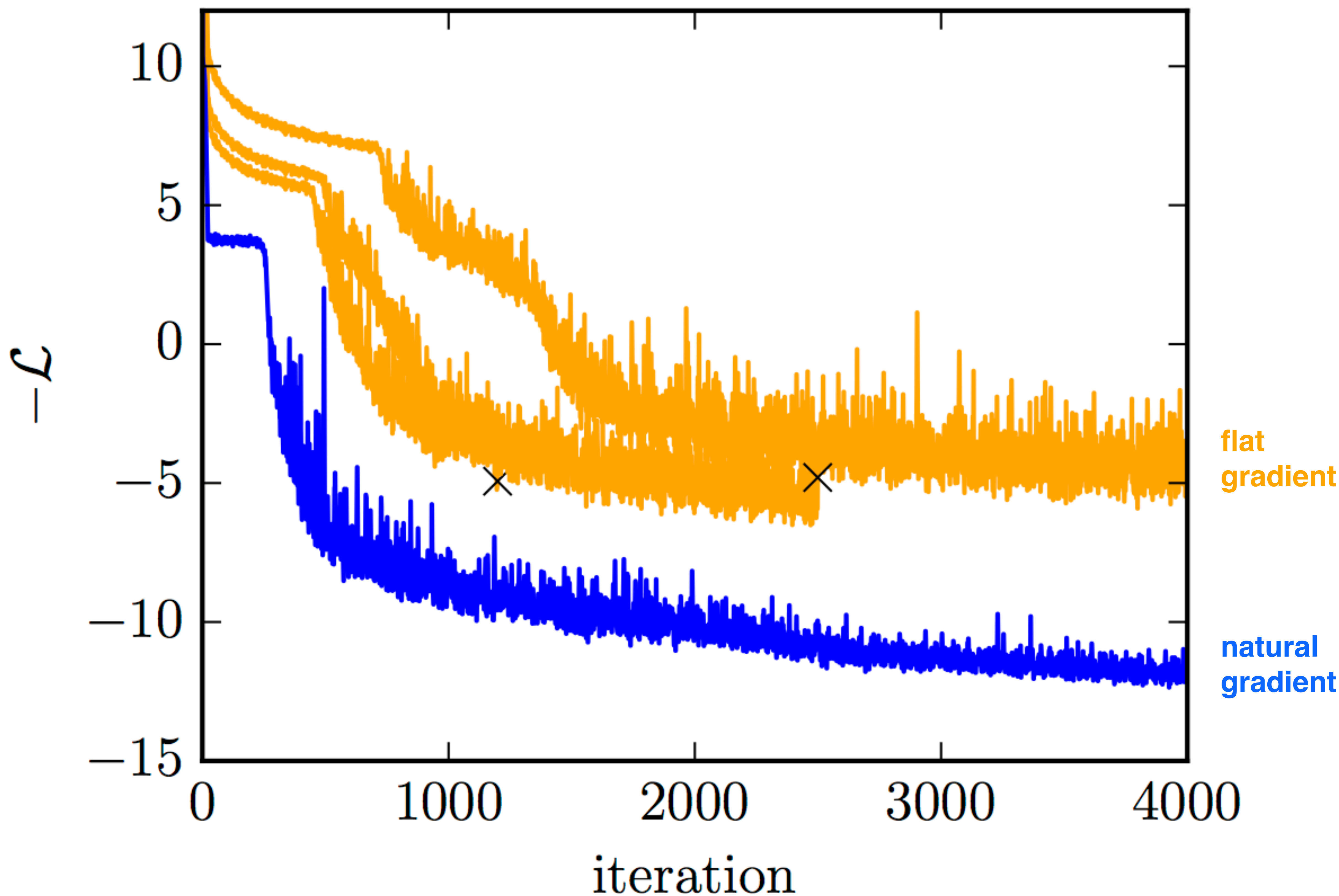

data space


latent space



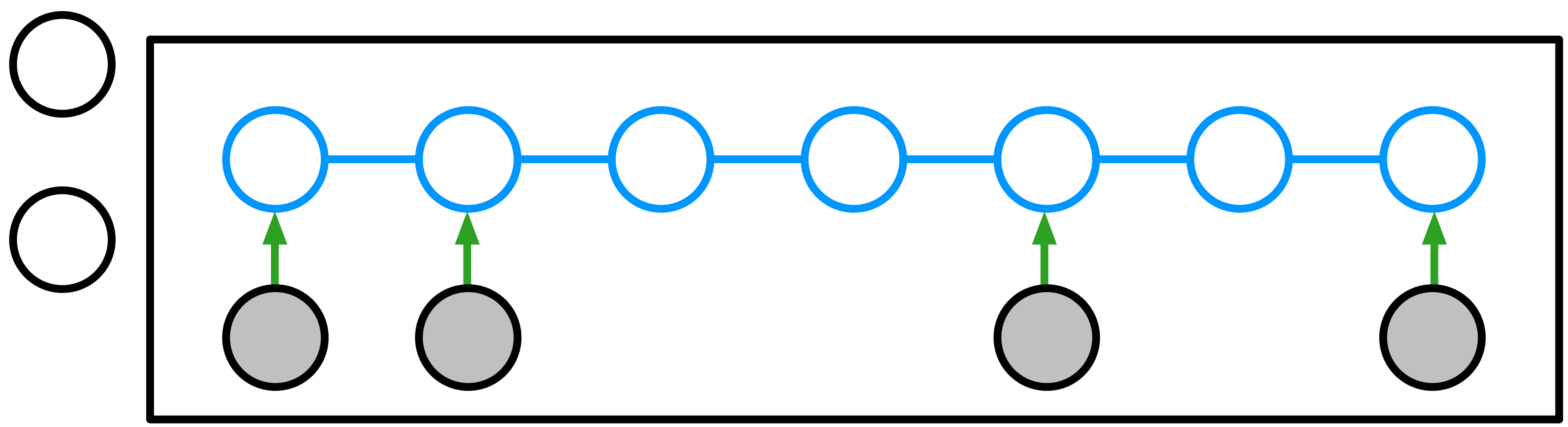
data
predictions
latent states
0
20
40
60
80
frame index


data
predictions
latent states
0
20
40
60
80
frame index



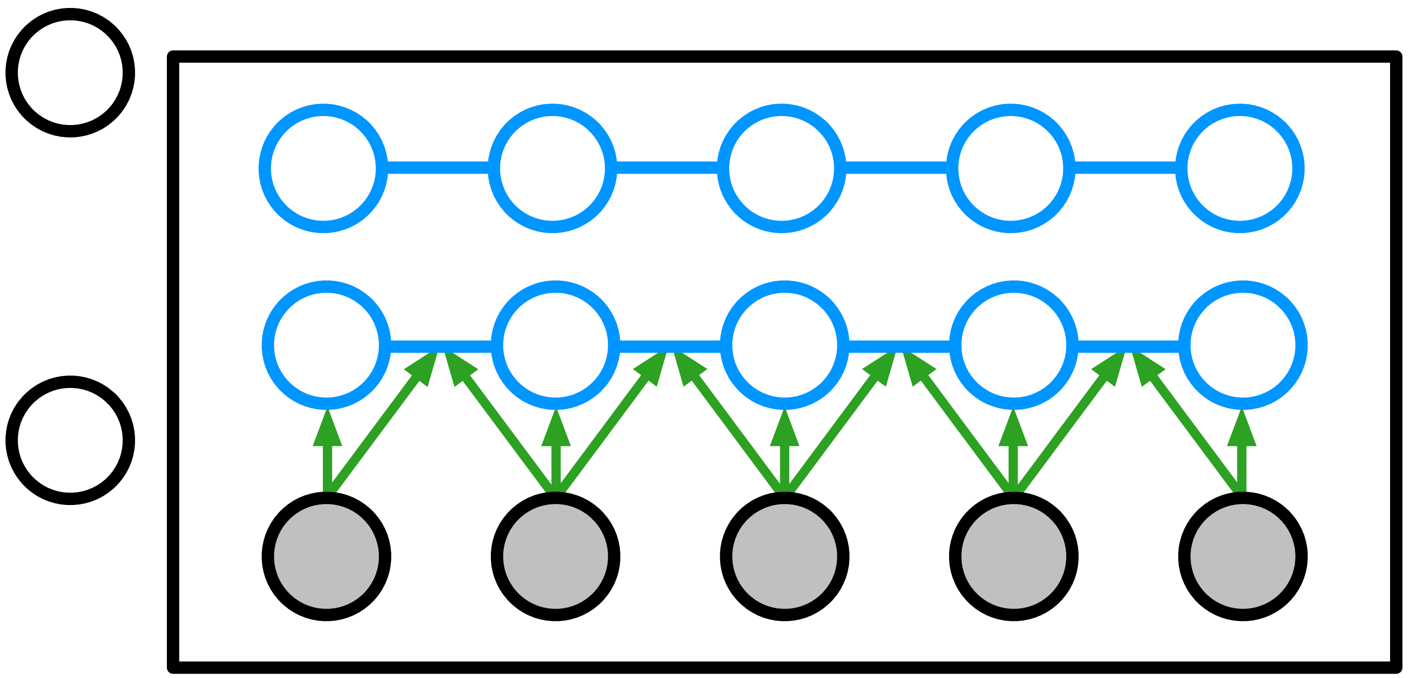
10
5
0
−5
−10
−15
0
1000
2000
3000
4000
iteration
−ℒ
flat gradient
natural gradient

arbitrary inference queries*

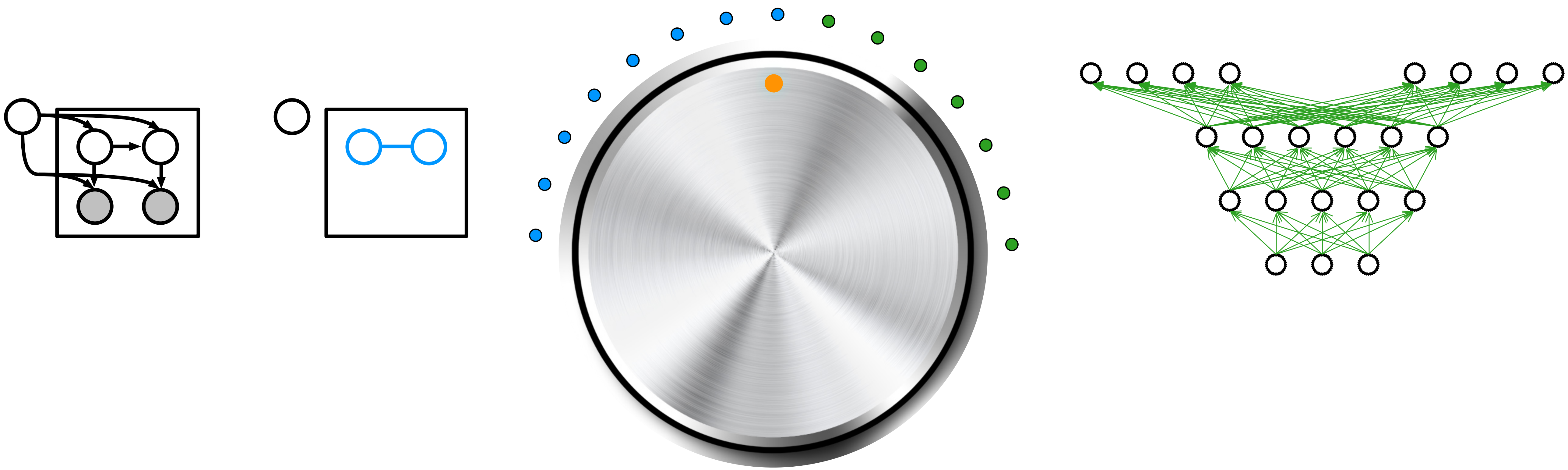


*see next slide

SVAEs can use any inference network architectures

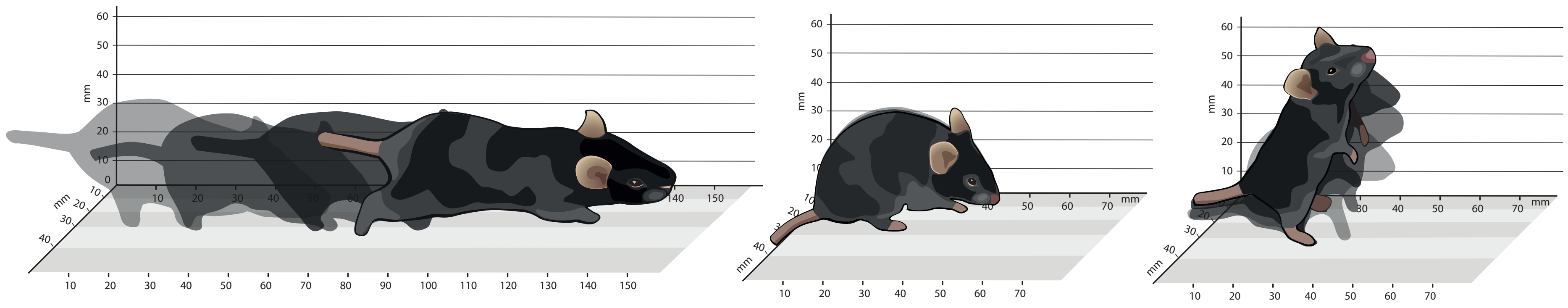


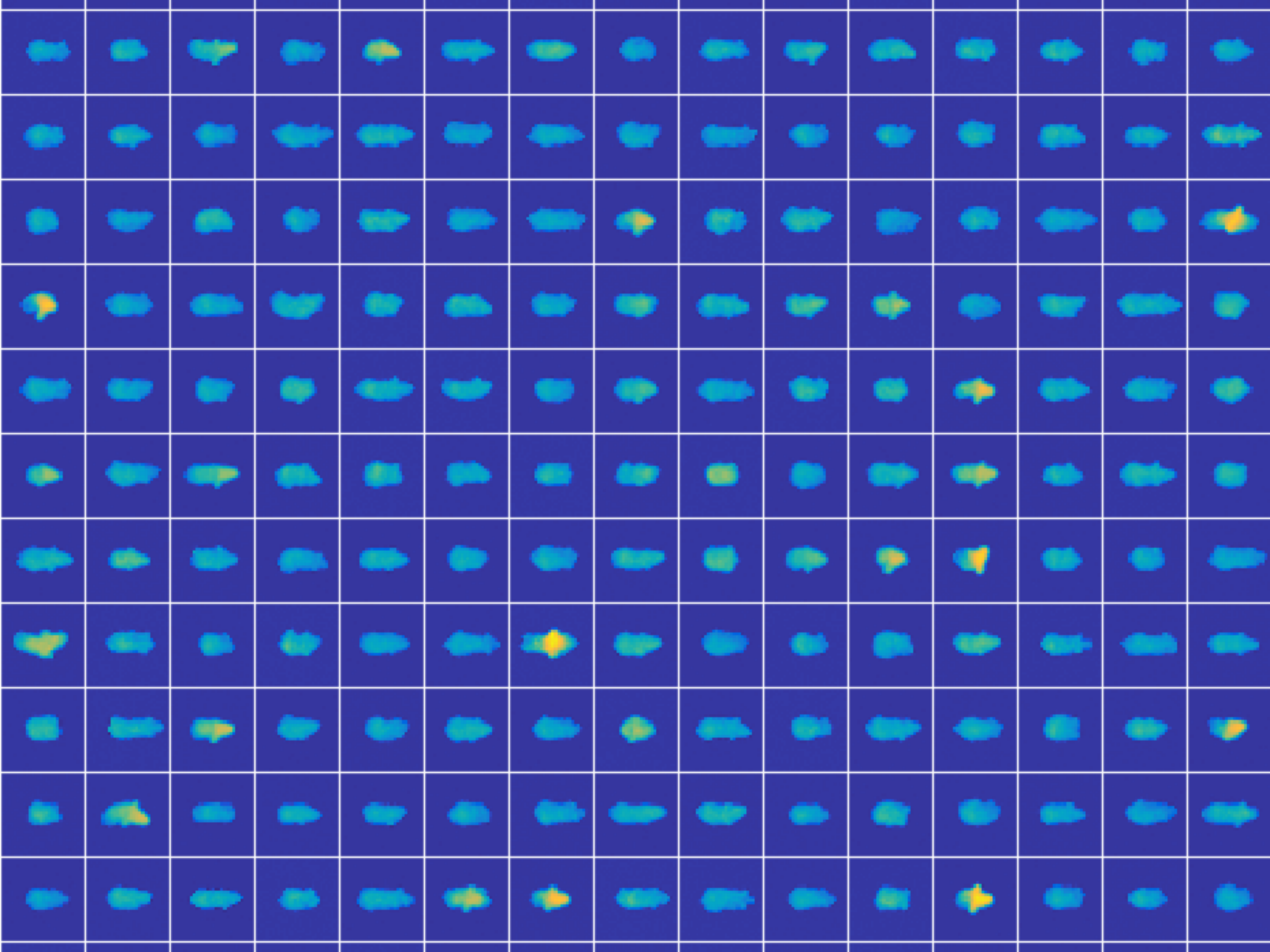
[1] Archer, Park, Buesing, Cunningham, Paninski. Black box variational inference for state space models. ICLR 2016 Workshops.
[2] Gao*, Archer*, Paninski, Cunningham. Linear dynamical neural population models through nonlinear embeddings. NIPS 2016.

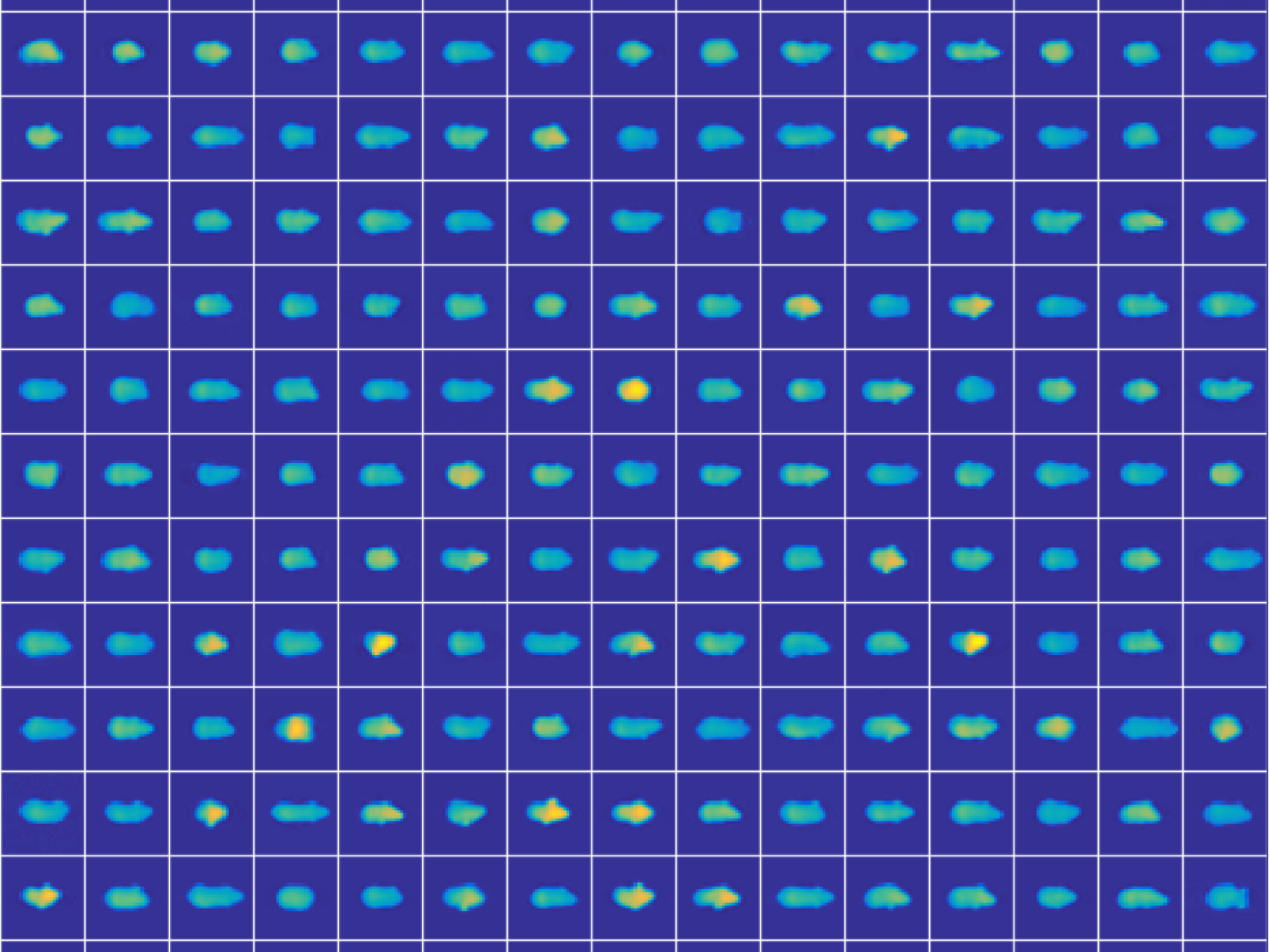


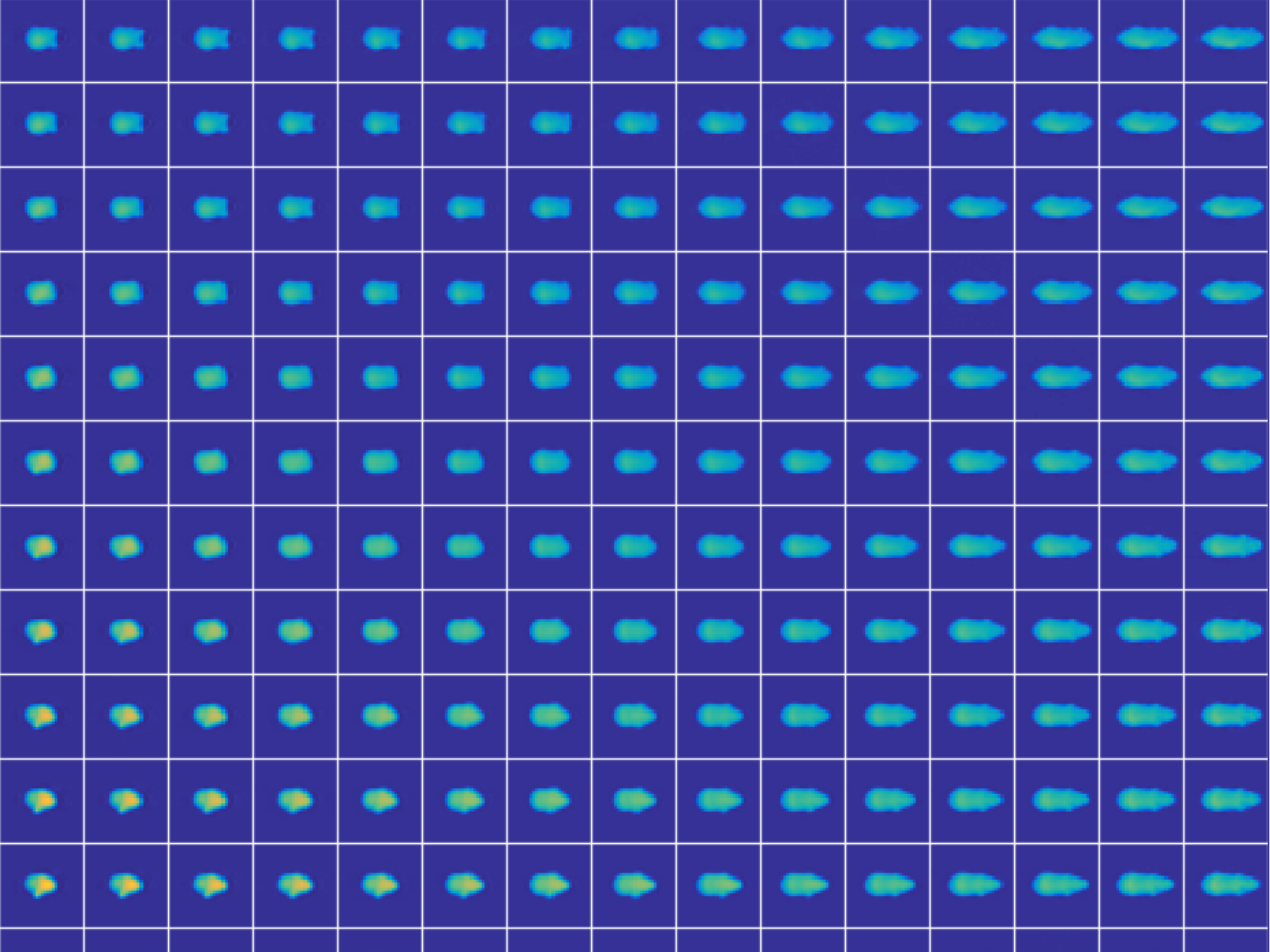
SVAEs

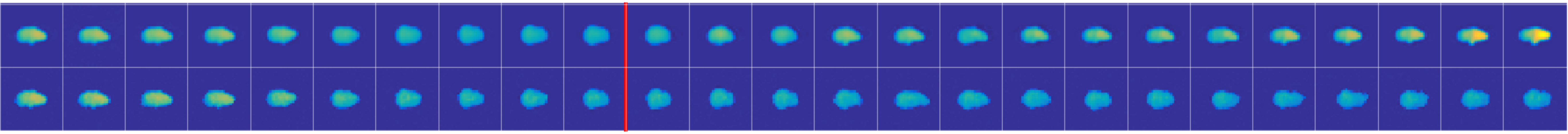
Application: learn syllable representation of behavior from video

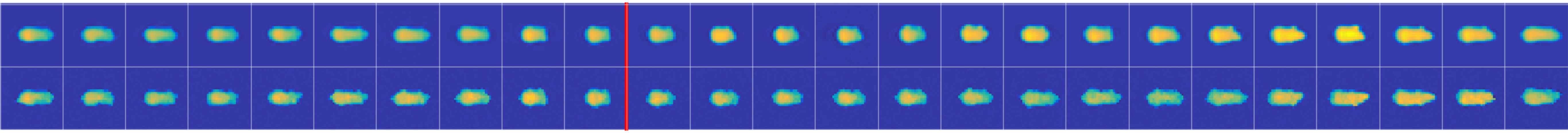


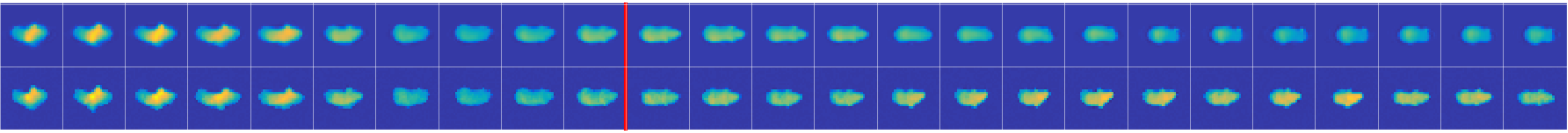


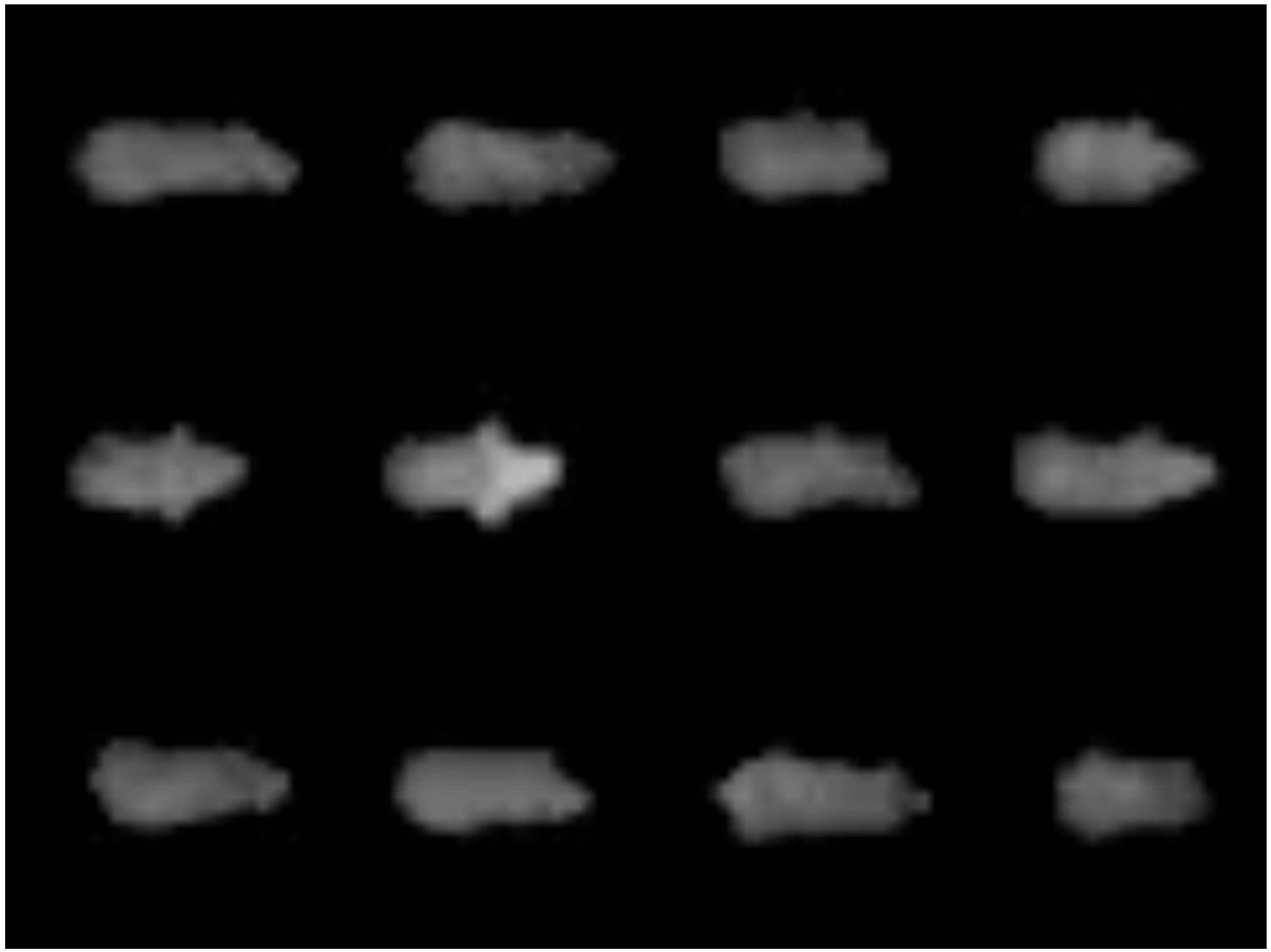




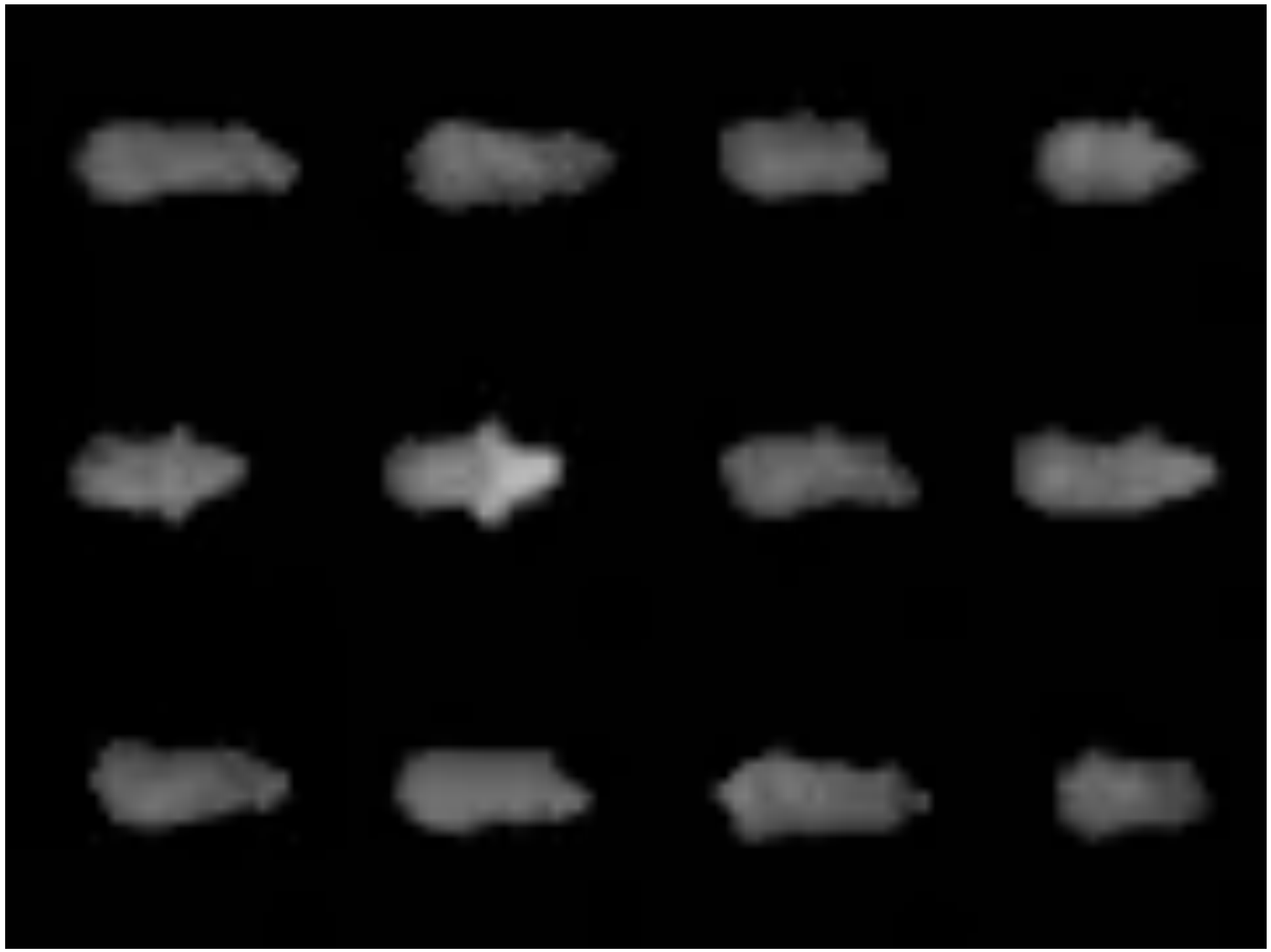




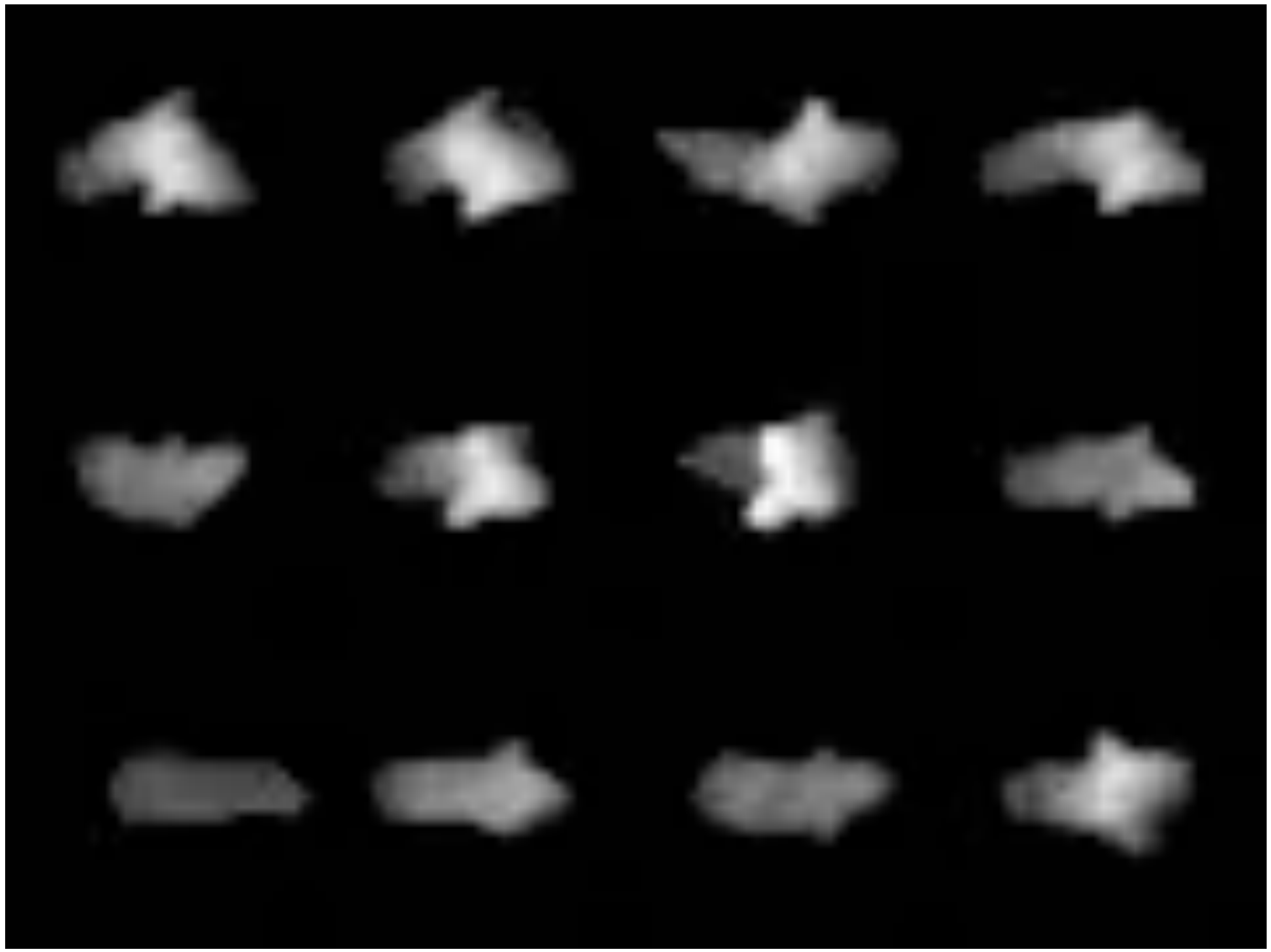




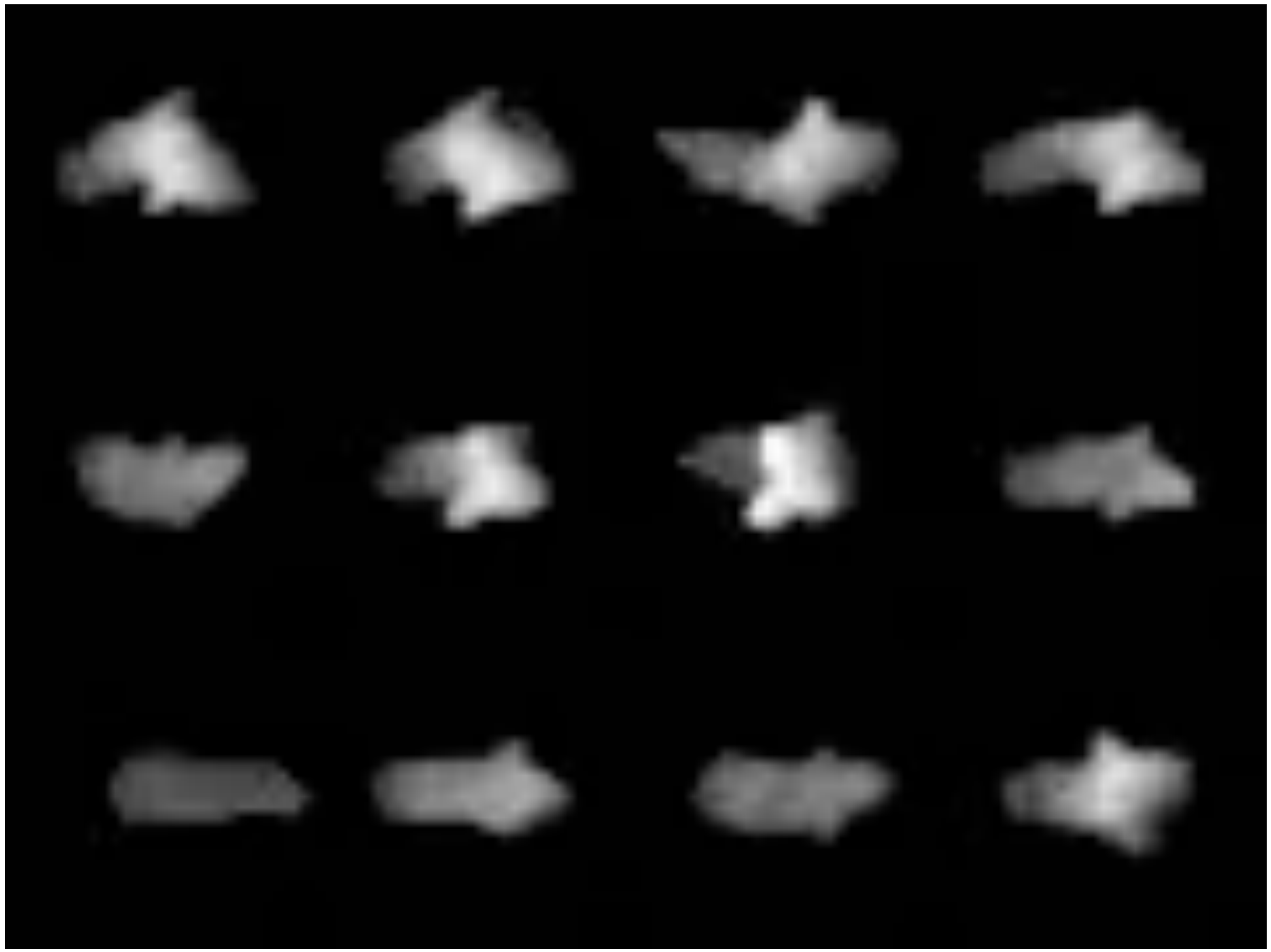
start rear



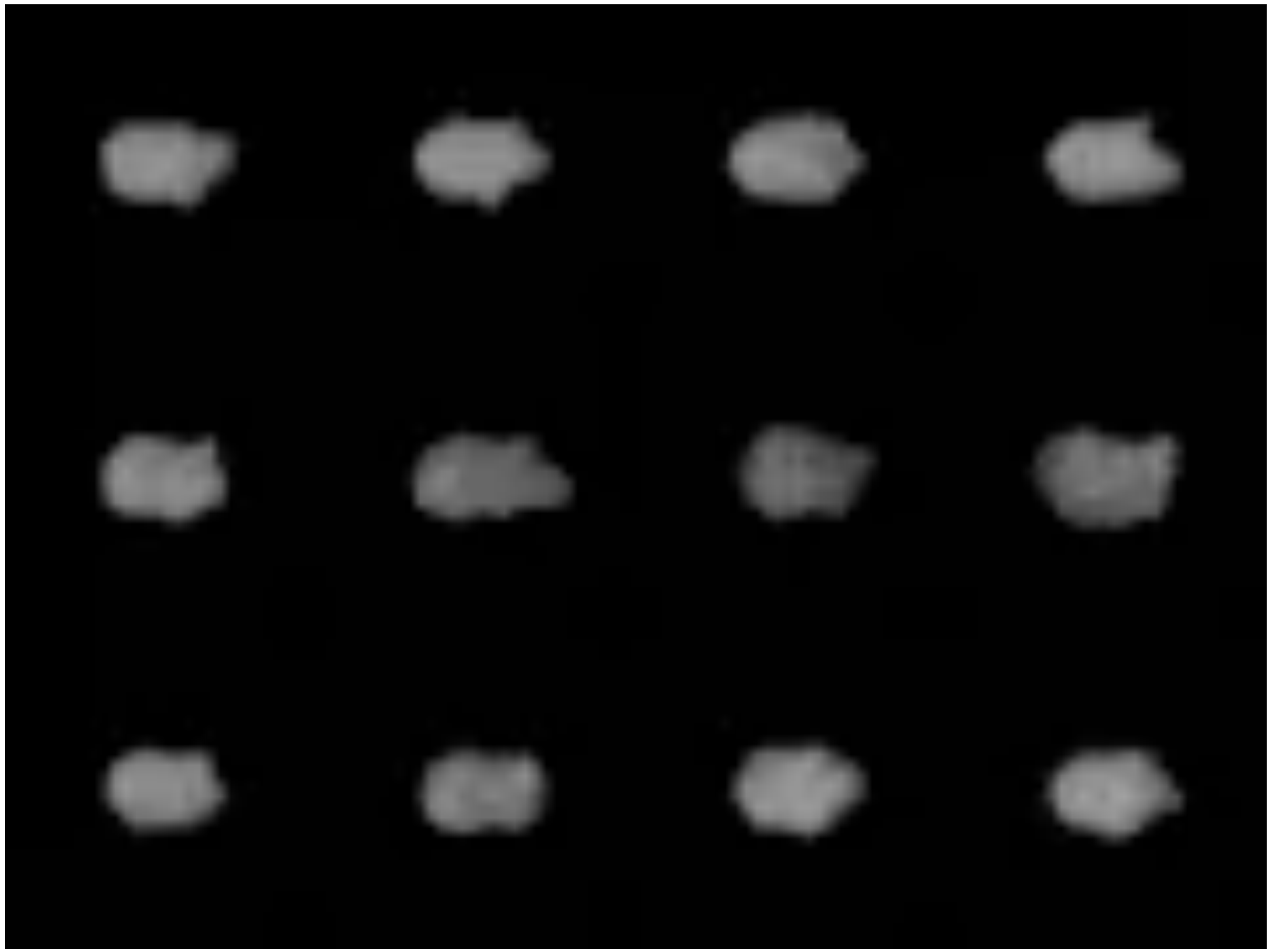
start rear



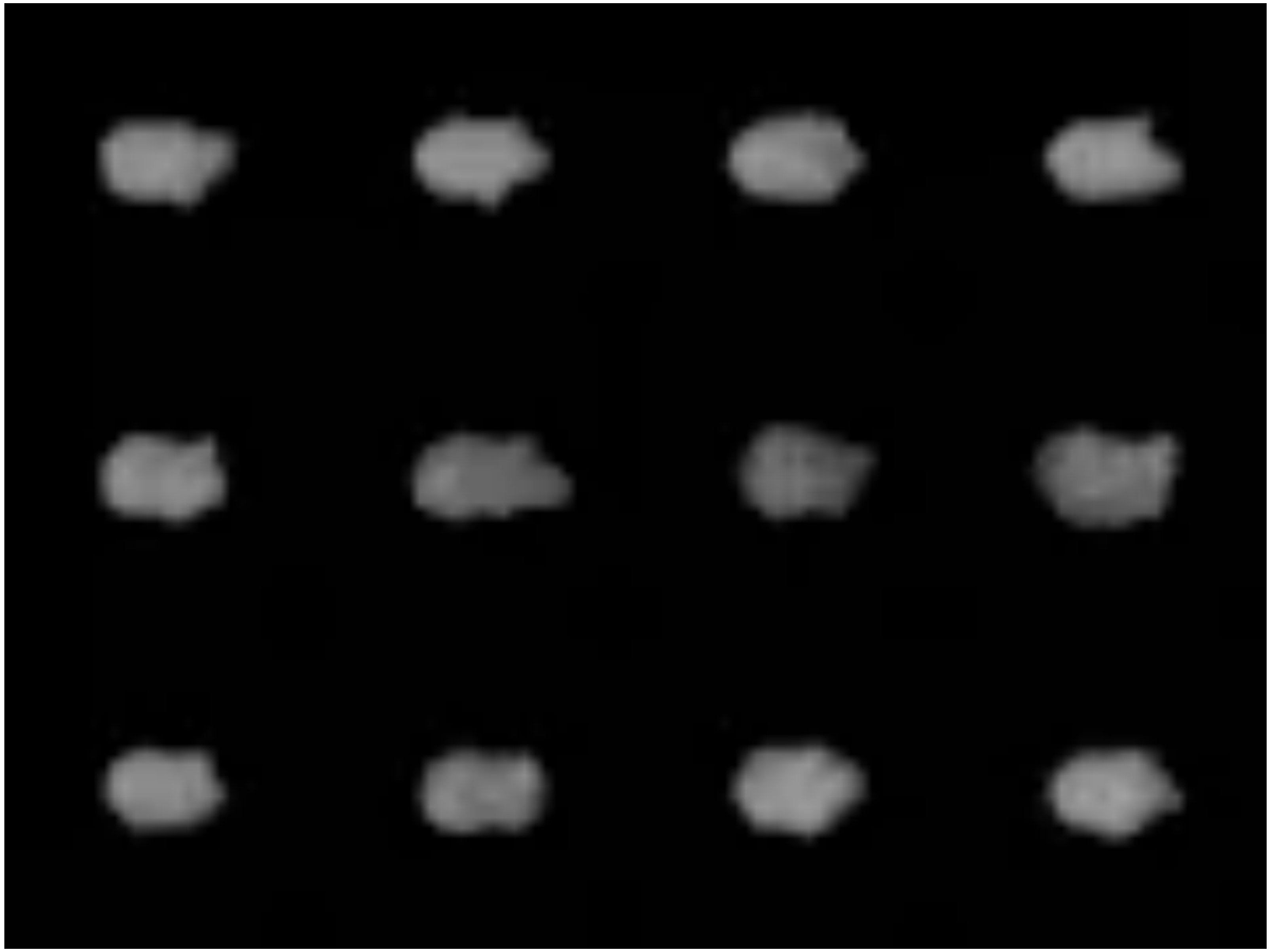
fall from rear



fall from rear

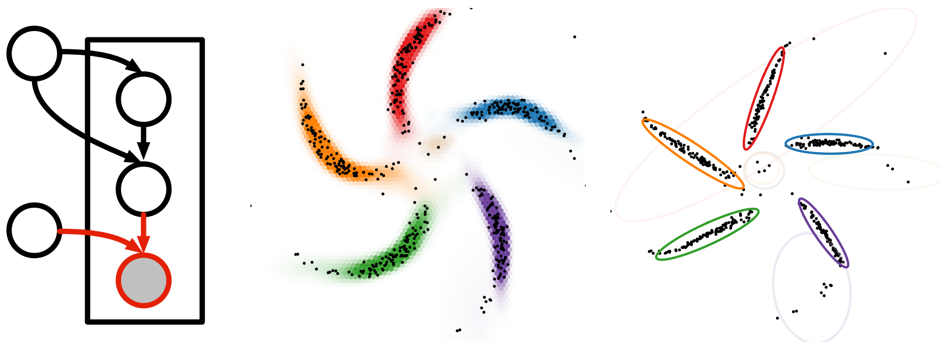


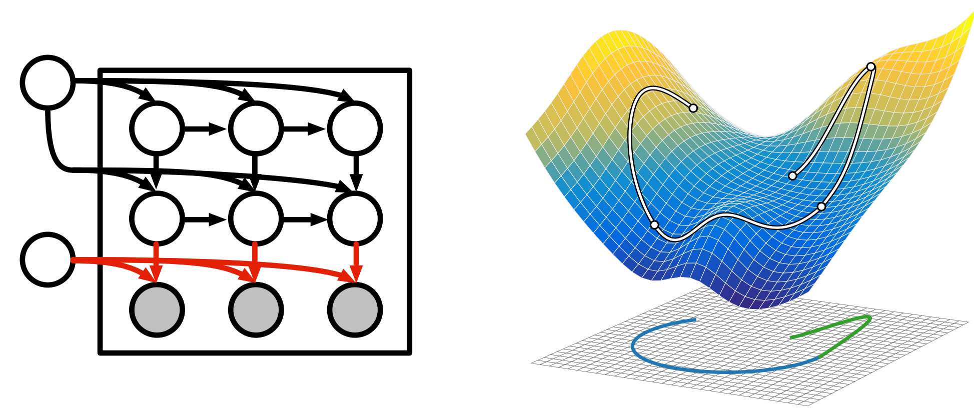
grooming



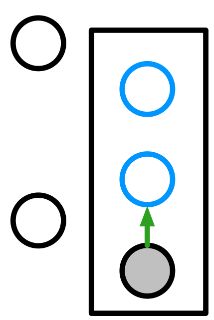
grooming

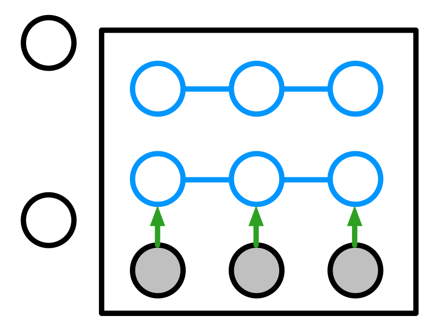
Modeling idea: graphical models on latent variables,
neural network models for observations



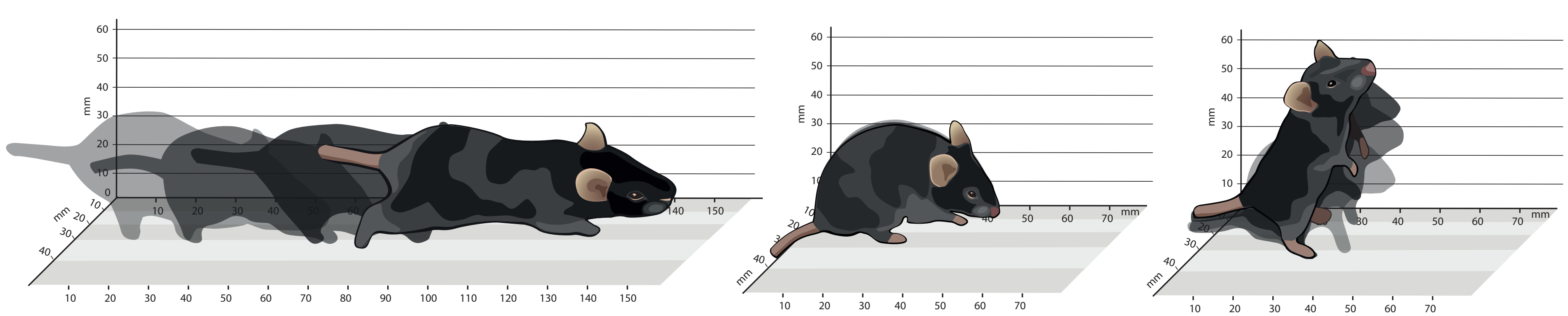


Inference: recognition networks output conjugate potentials,
then apply fast graphical model inference



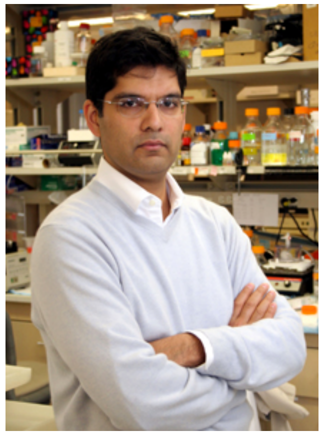


Application: learn syllable representation of behavior from video



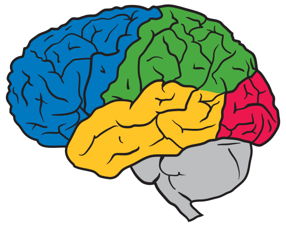
Thanks!








HARVARD
INTELLIGENT
PROBABILISTIC
SYSTEMS



HARVARD
MEDICAL SCHOOL