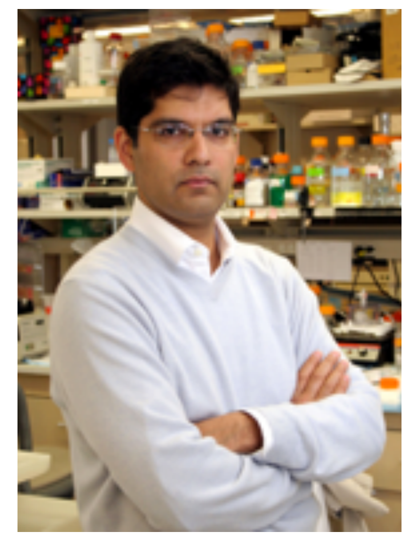
Composing graphical models with neural networks for structured representations and fast inference

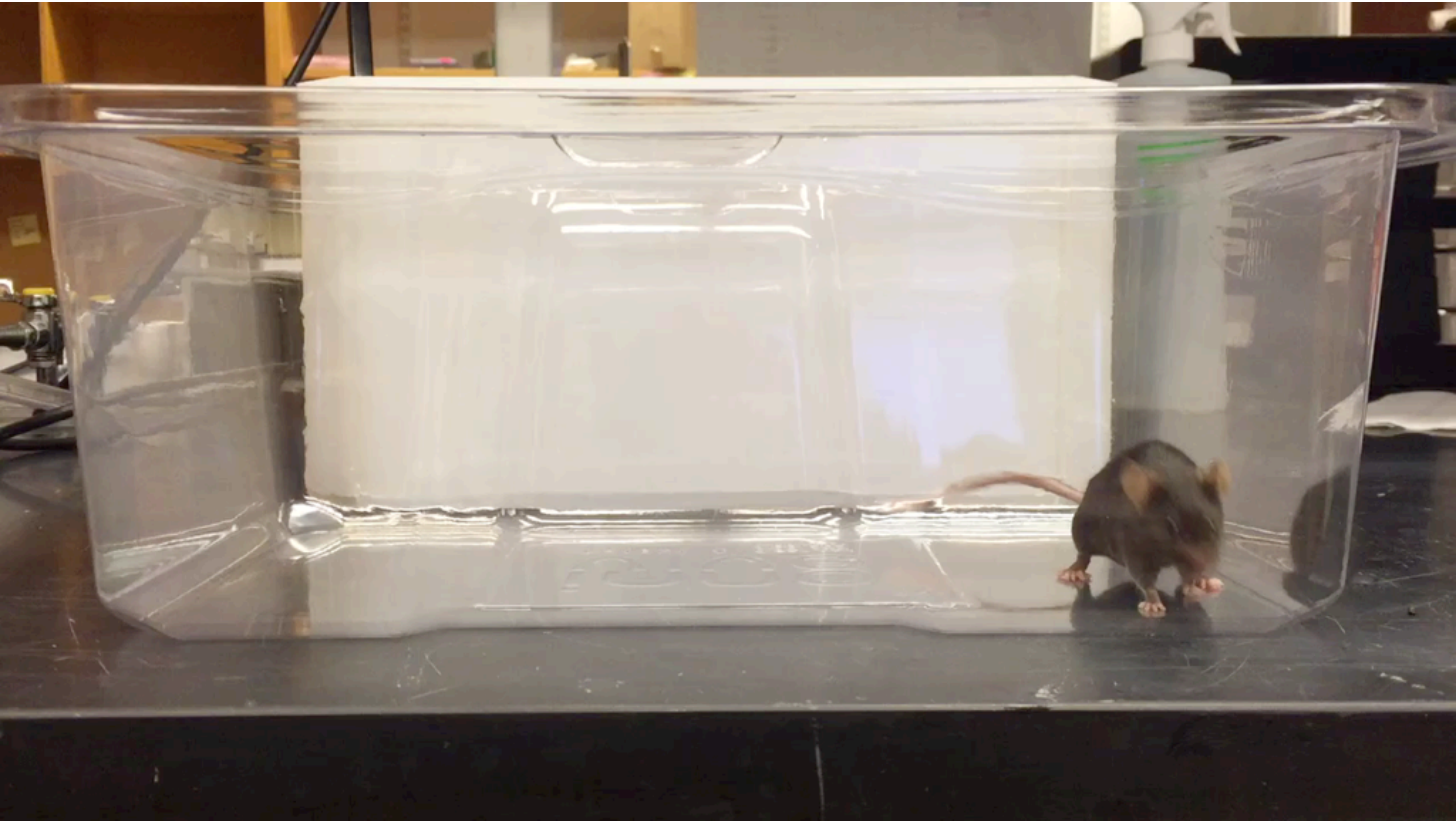
Matt Johnson, David Duvenaud, Alex Wiltschko, Bob Datta, Ryan Adams

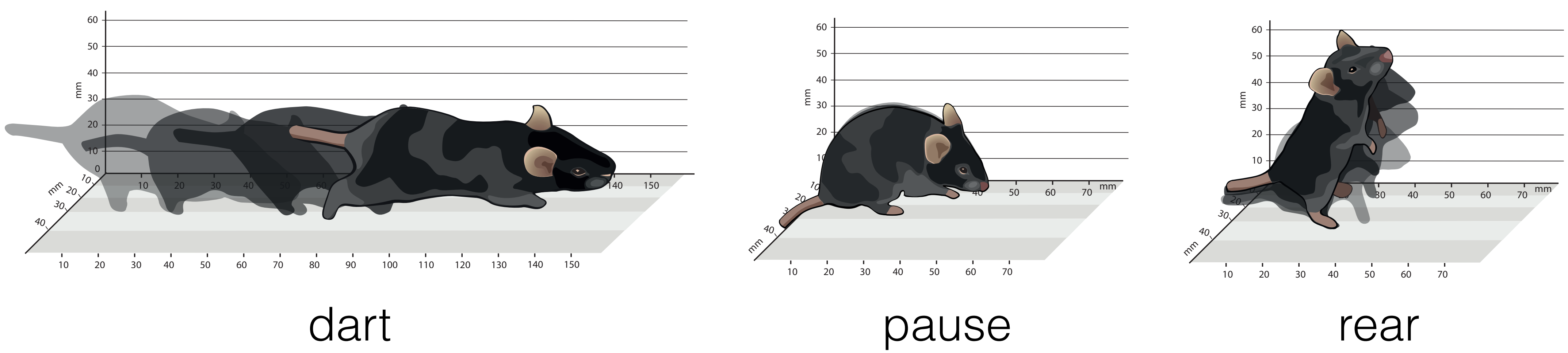


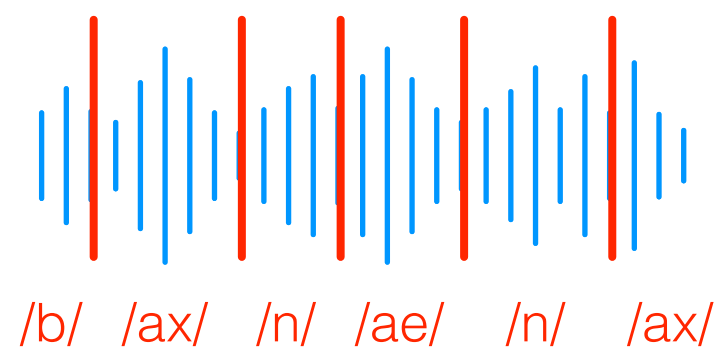




HARVARD INTELLIGENT PROBABILISTIC SYSTEMS

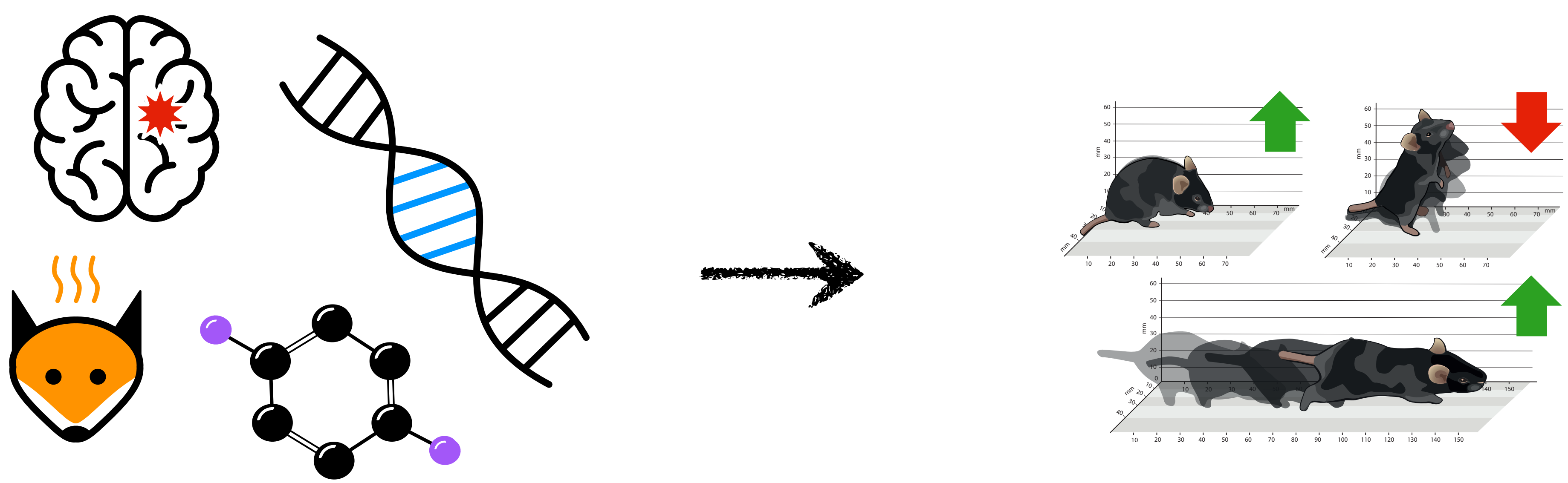
HARVARD MEDICAL SCHOOL



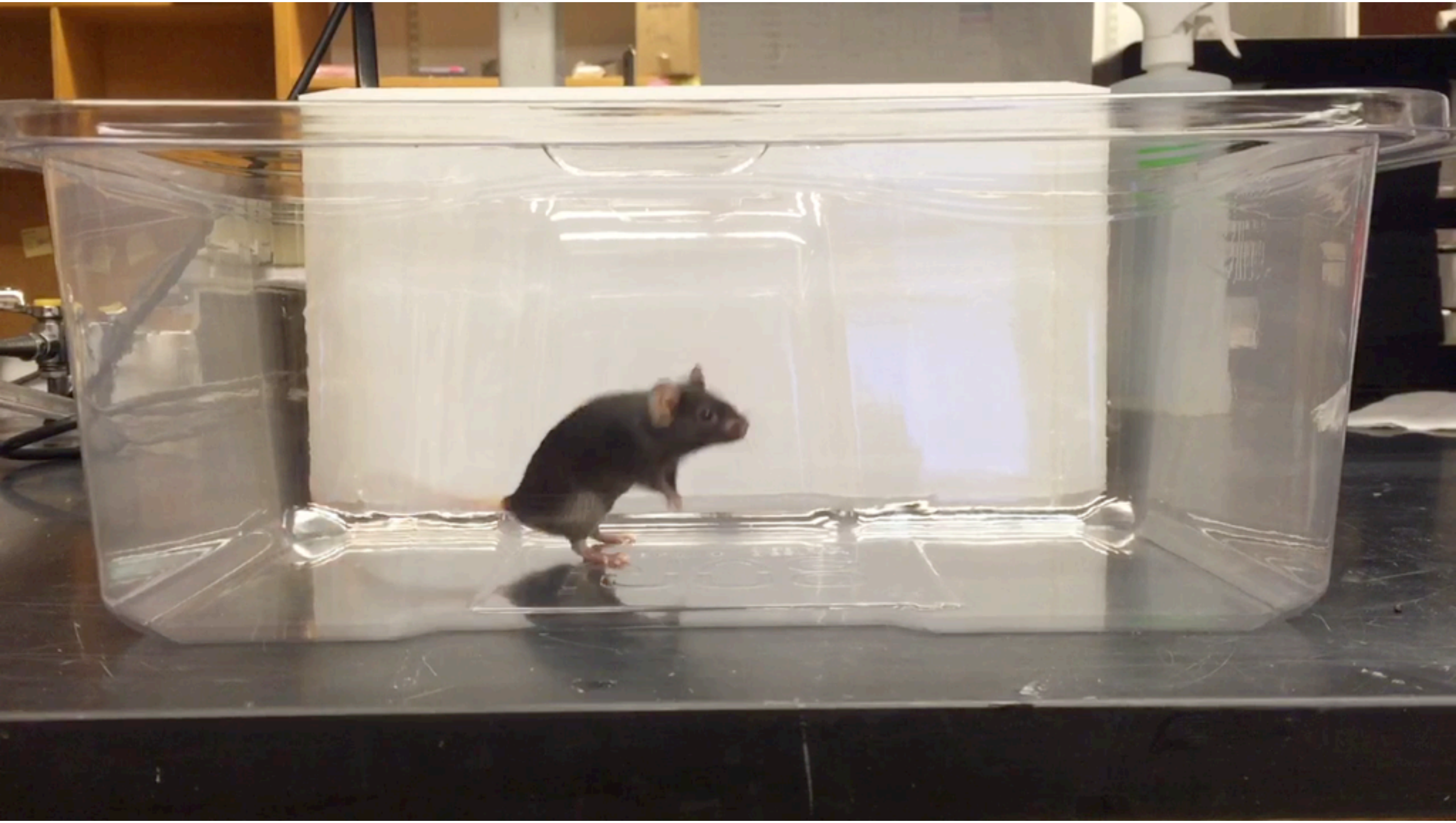


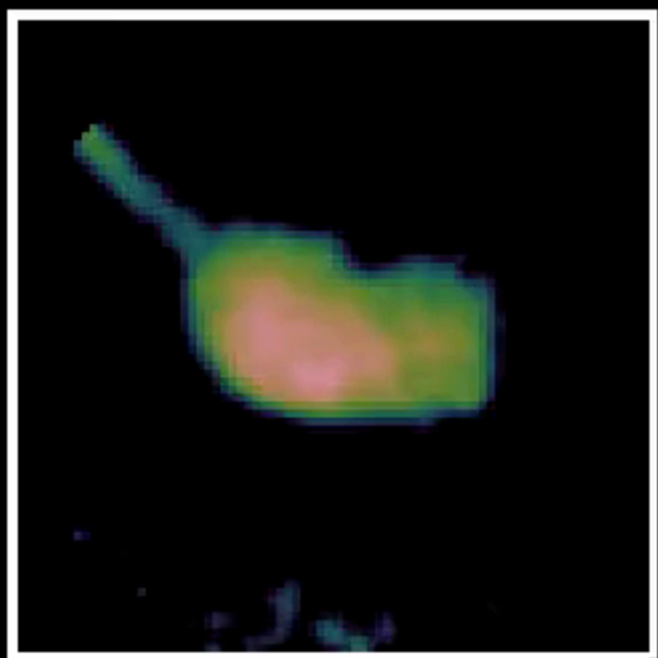


[1,2]

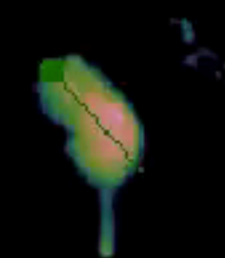


[1] Lee and Glass. A Nonparametric Bayesian Approach to Acoustic Model Discovery. ACL 2012.
[2] Lee. Discovering Linguistic Structures in Speech: Models and Applications. MIT Ph.D. Thesis 2014.





Frame 0



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

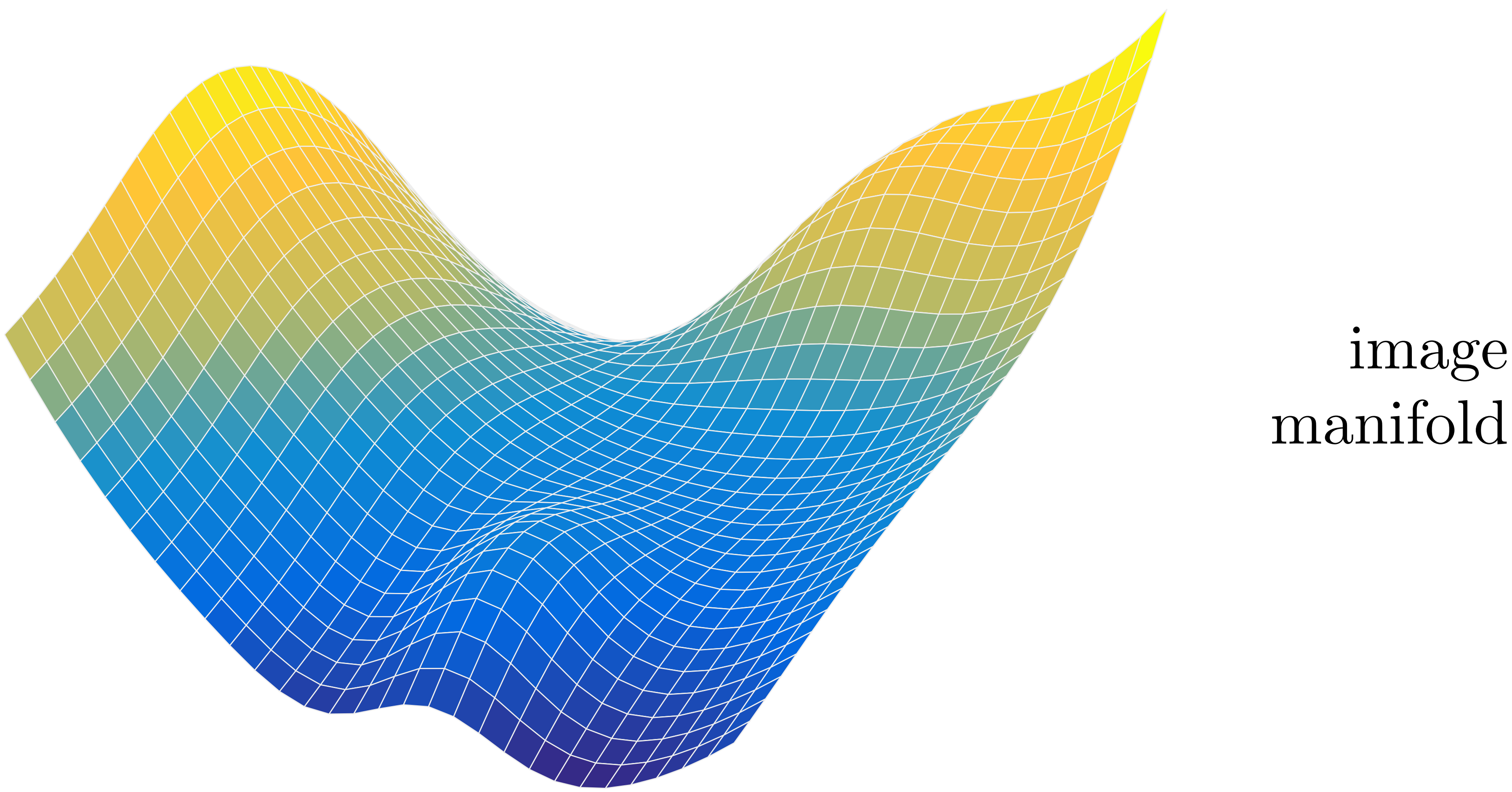
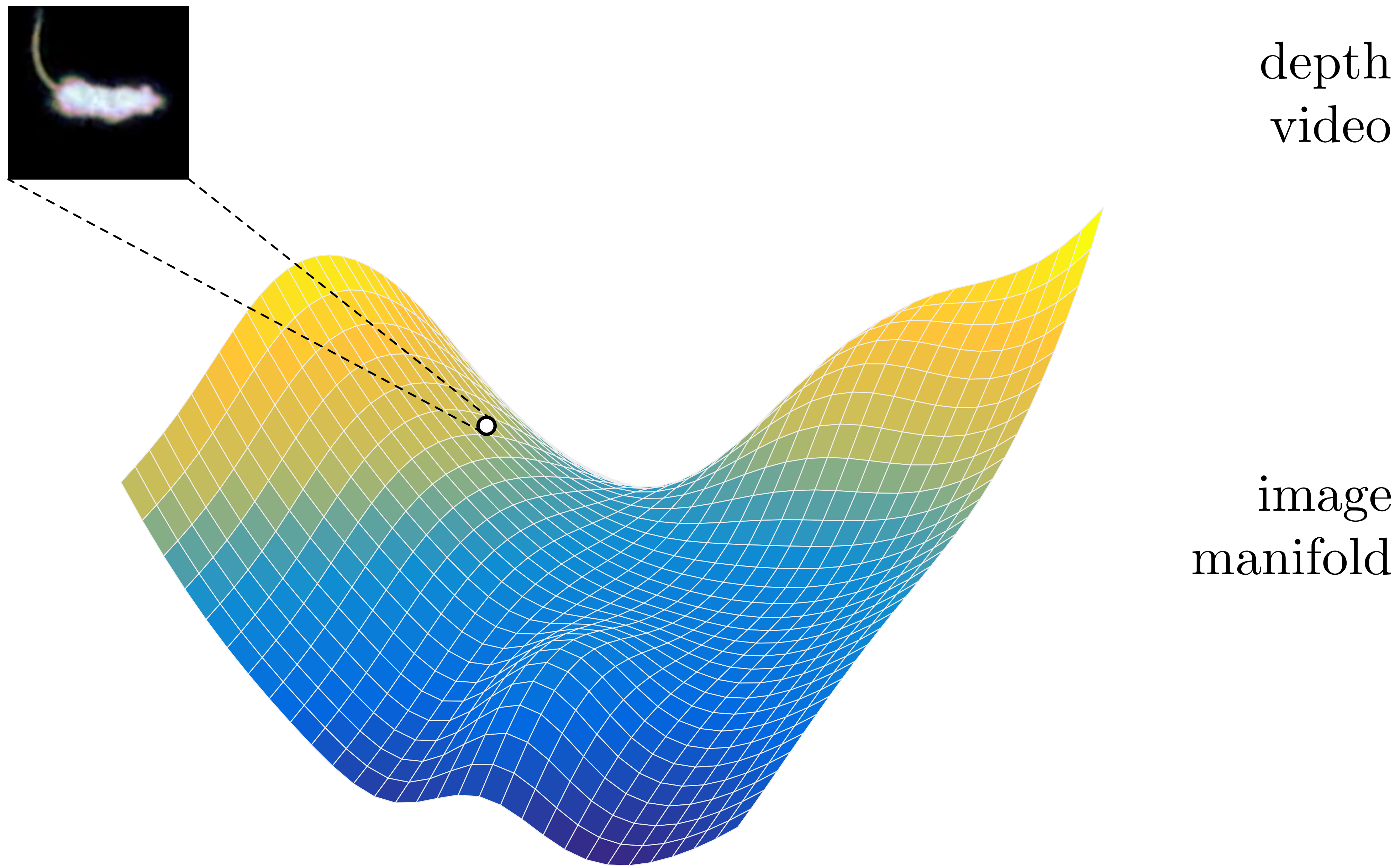
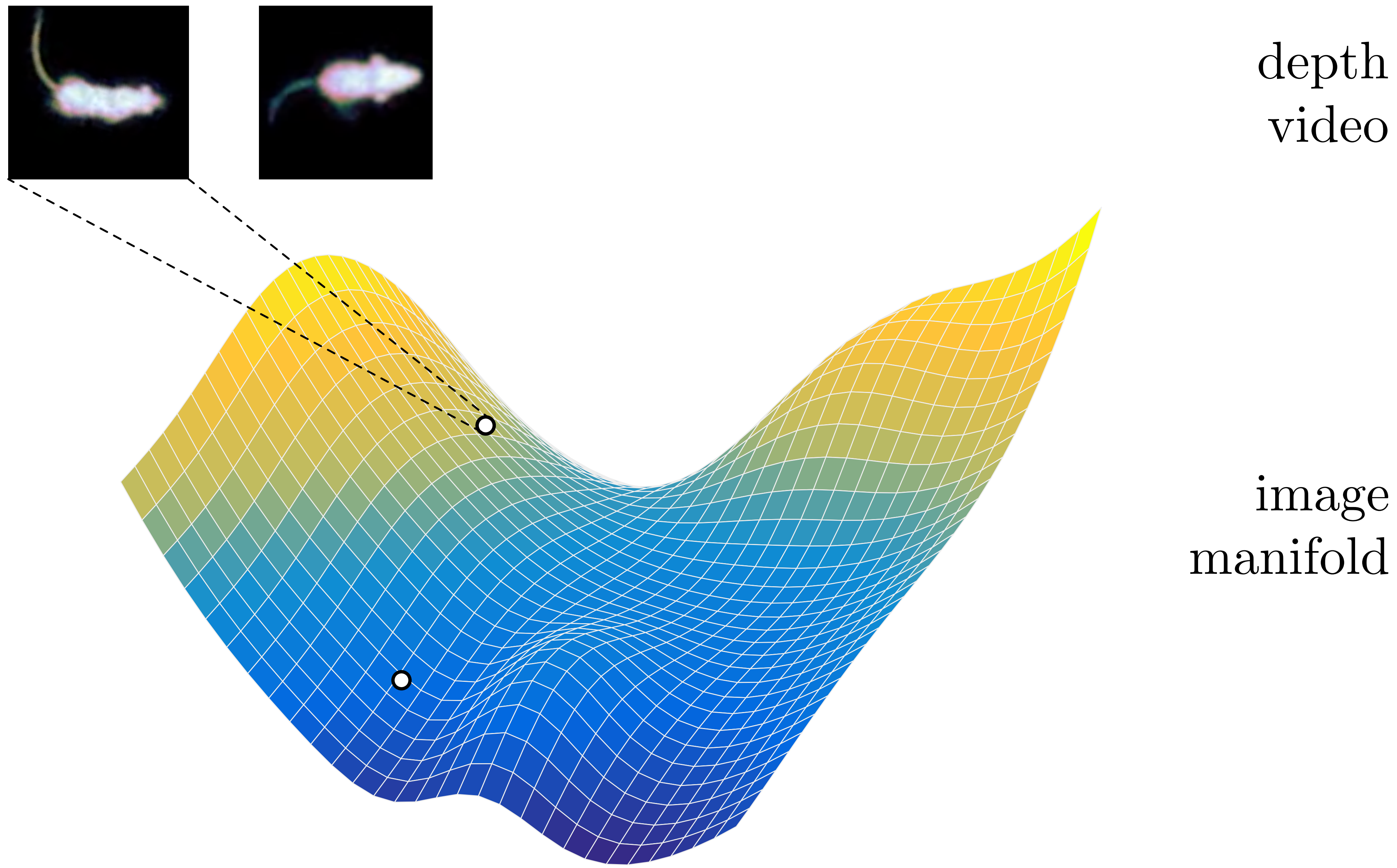
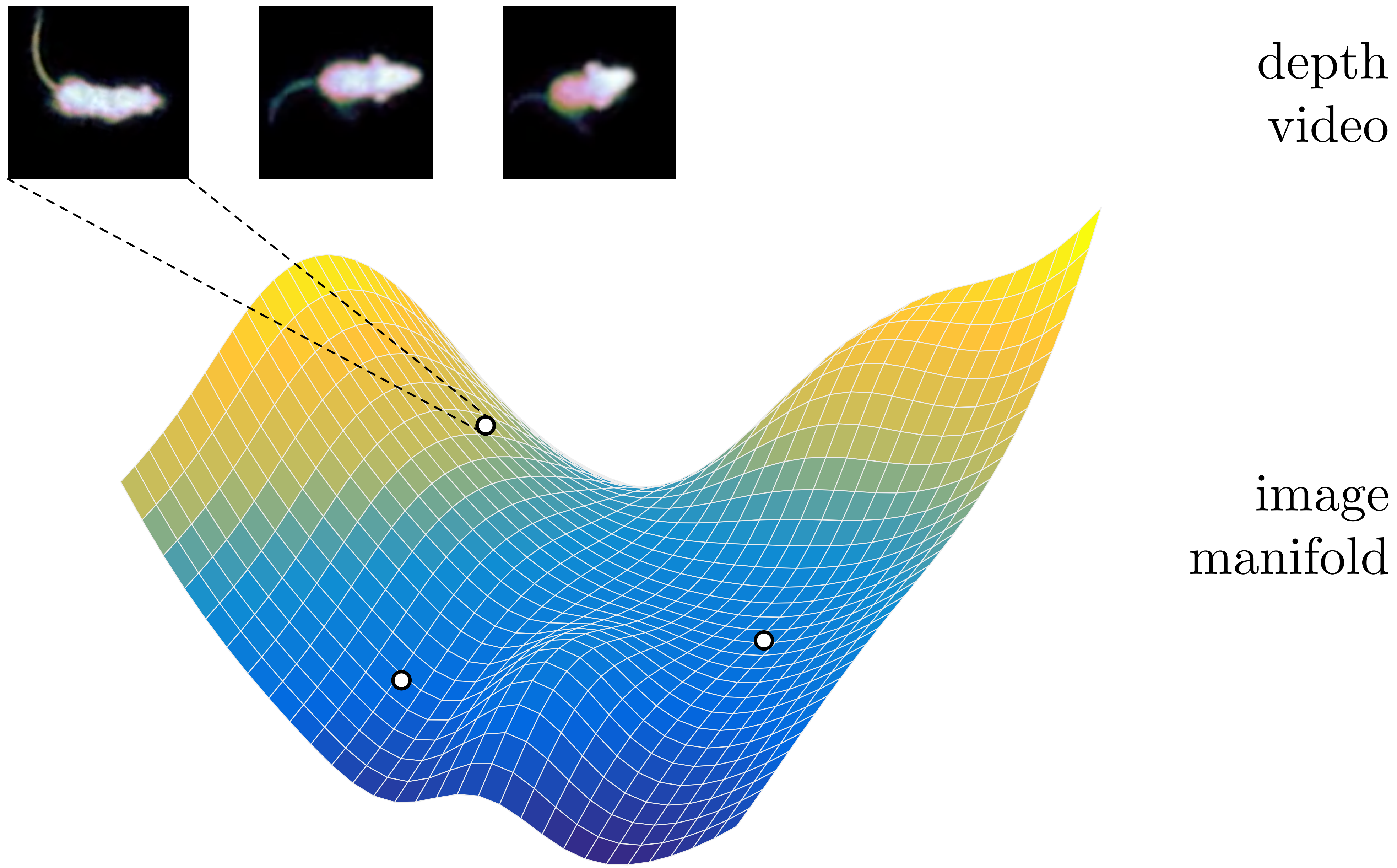
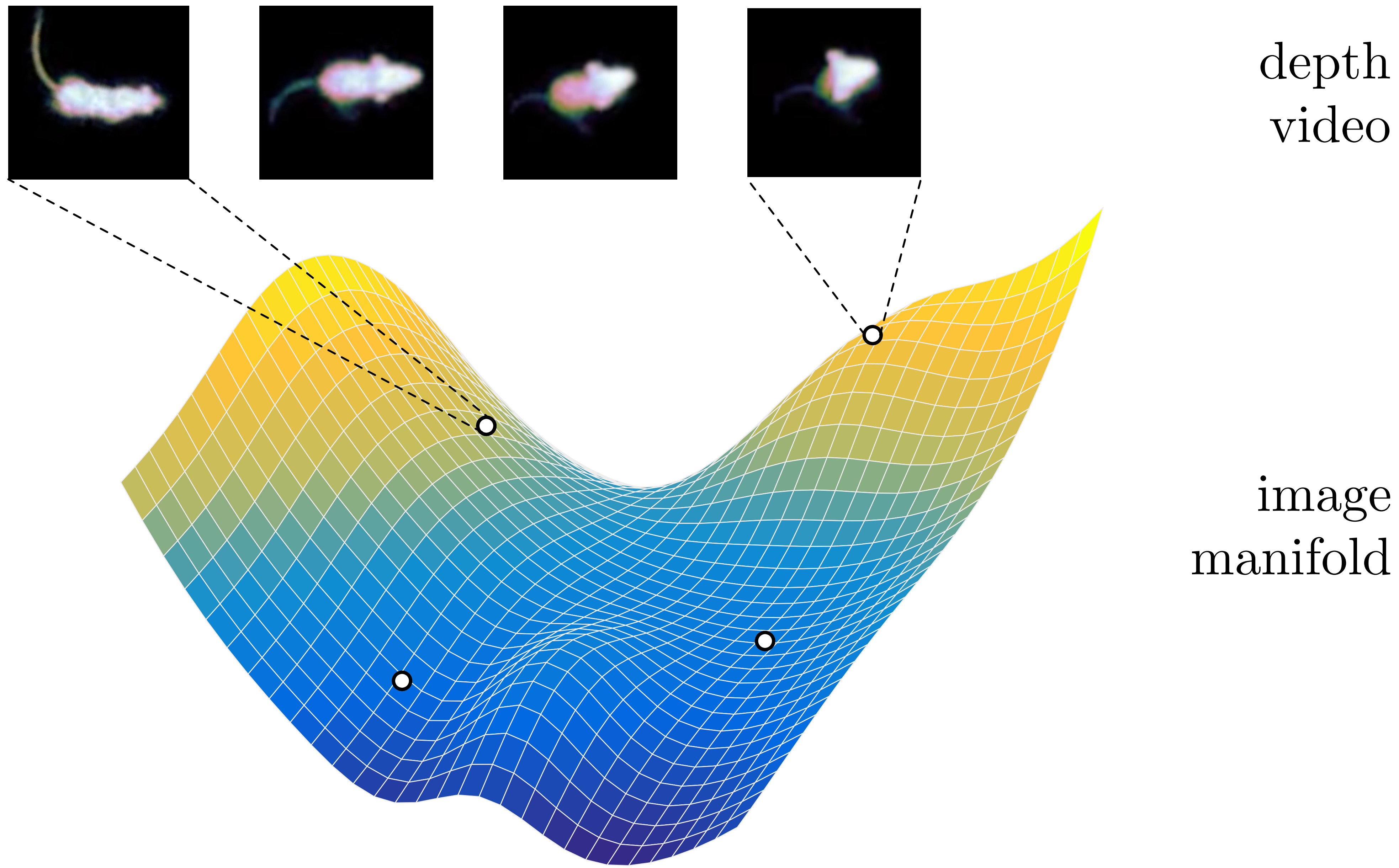


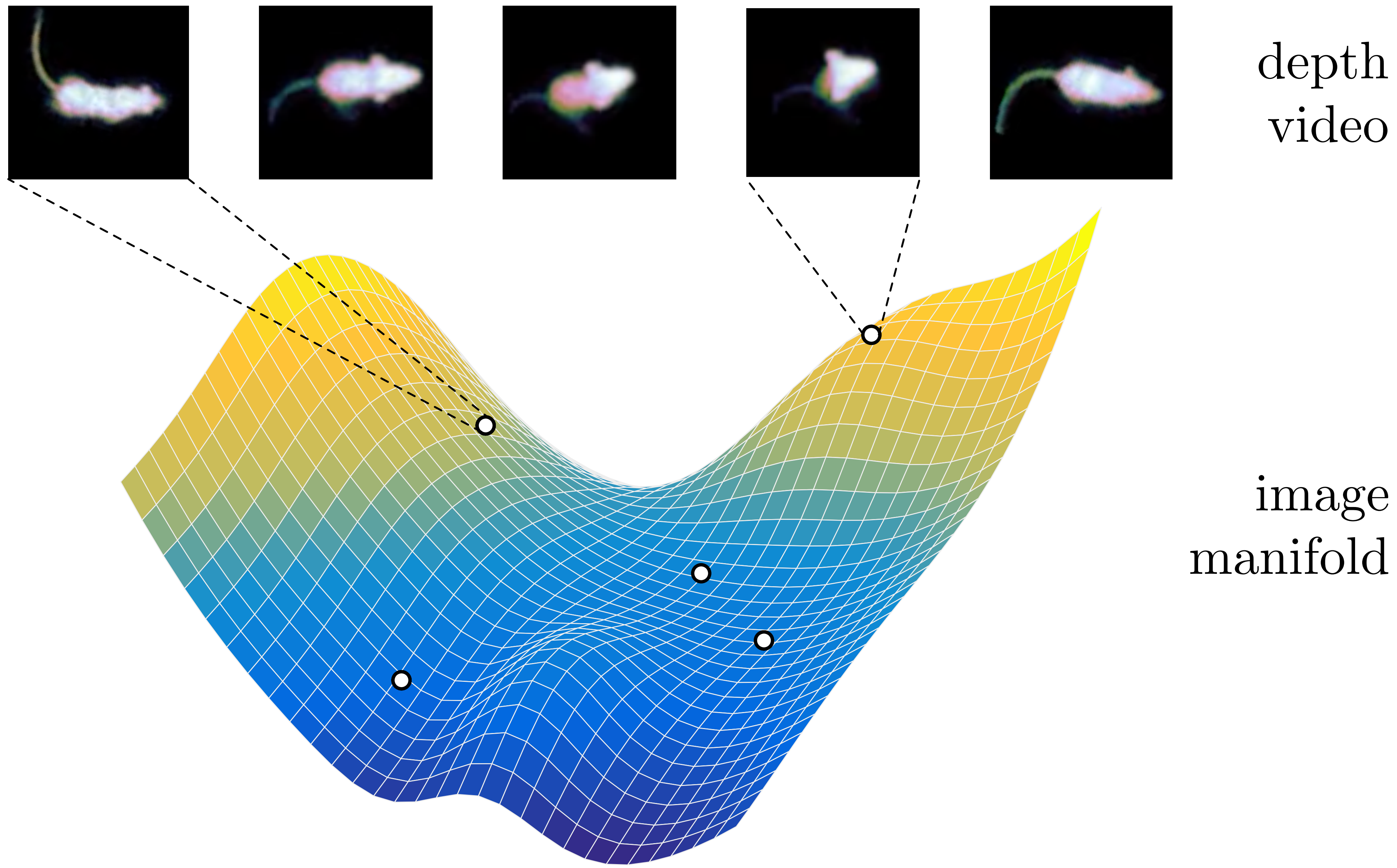
image
manifold

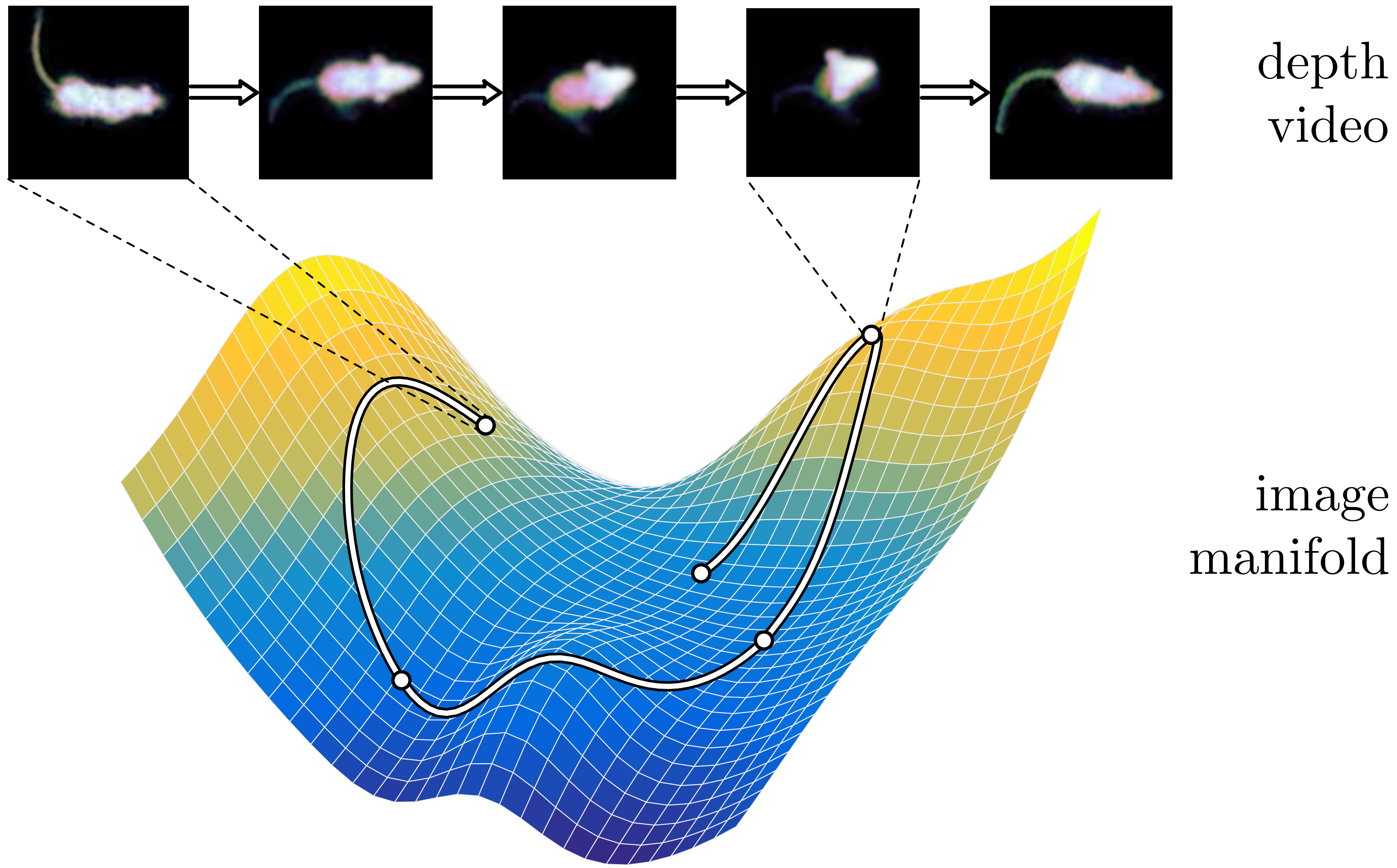

depth
video
image
manifold

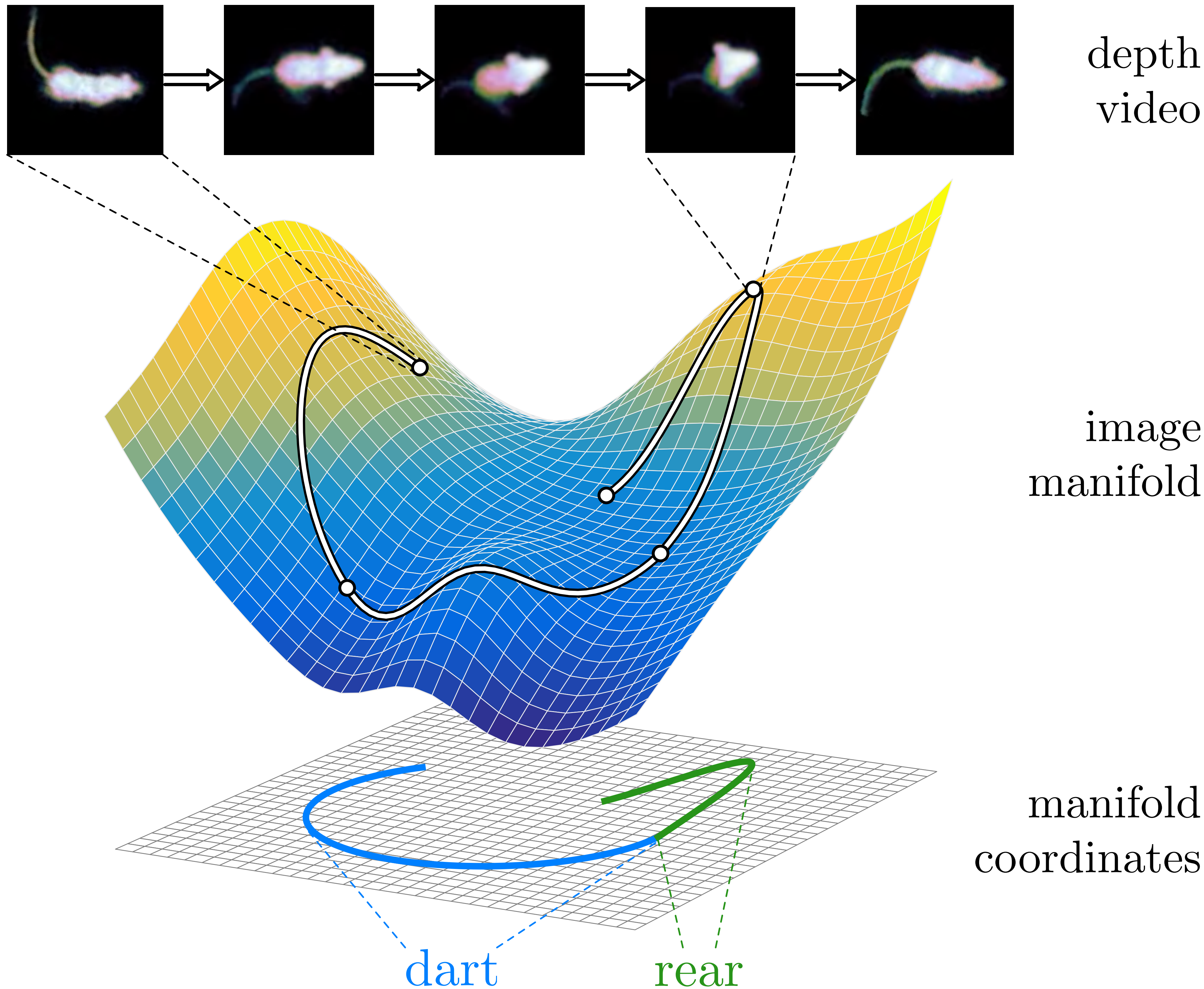

depth
video
image
manifold


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]

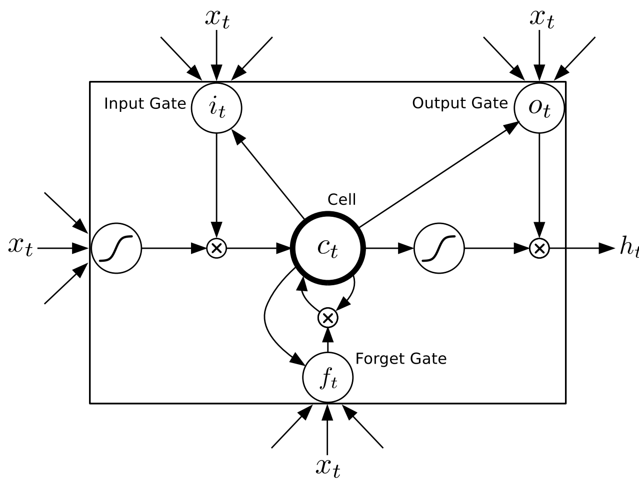


Figure 1. LSTM unit

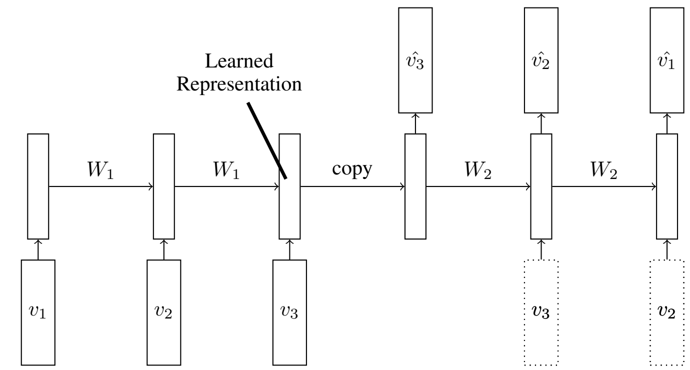
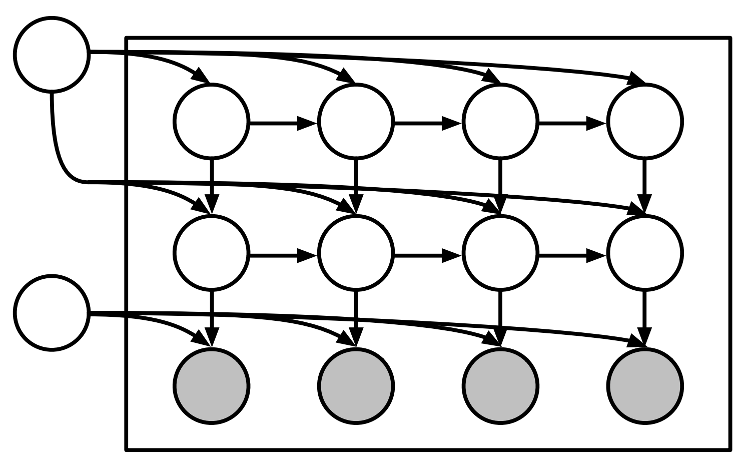
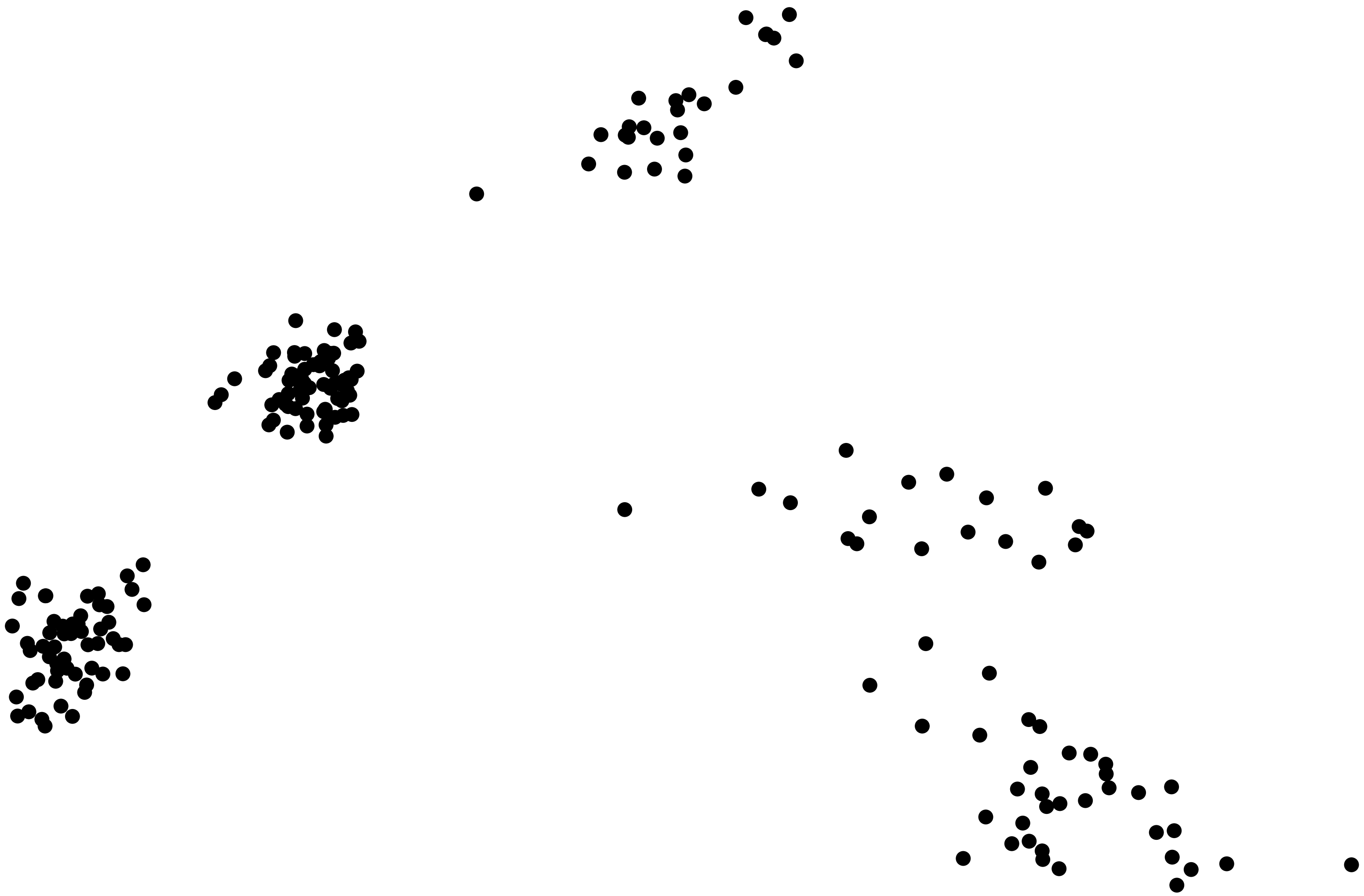


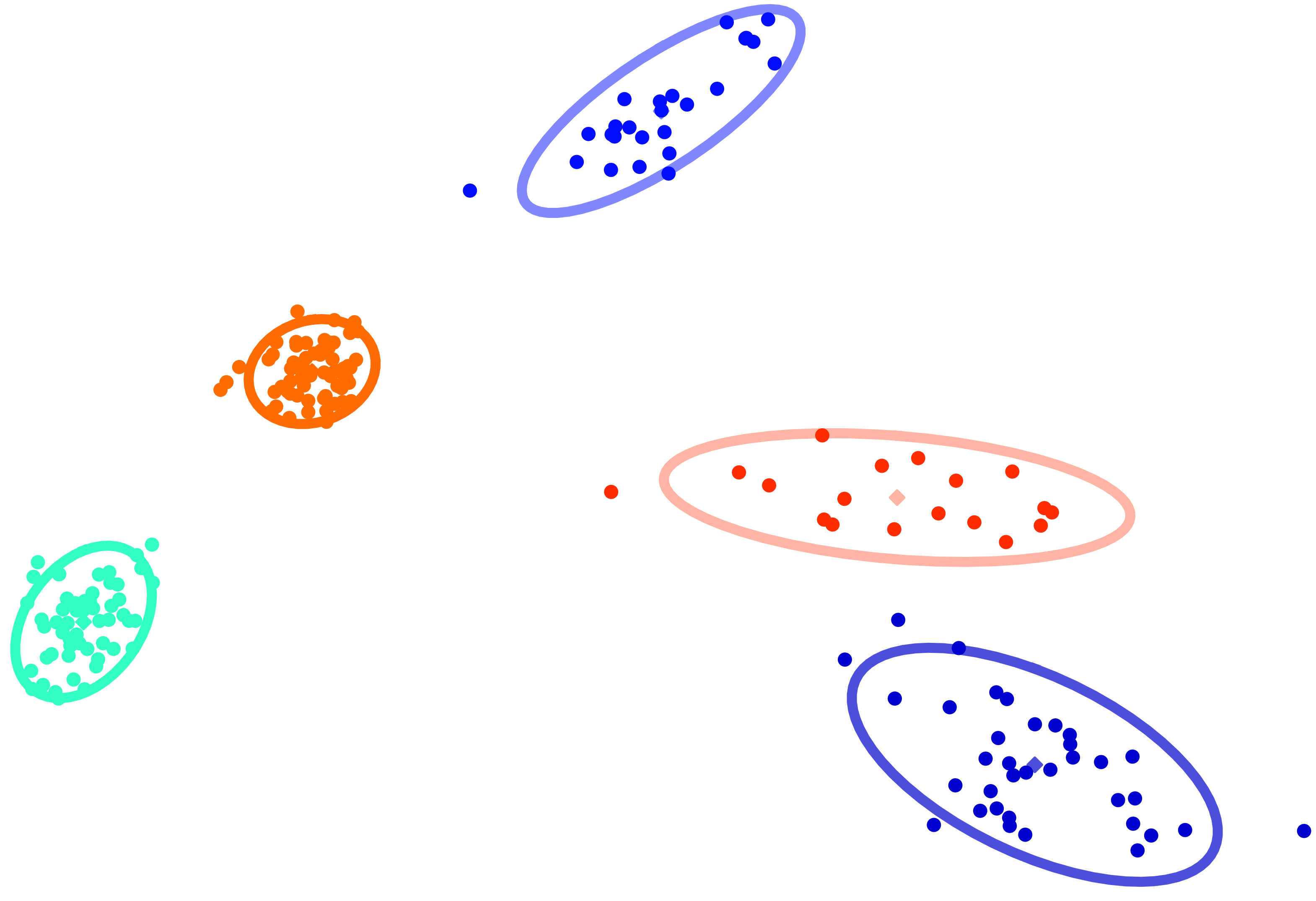
Figure 2. LSTM Autoencoder Model

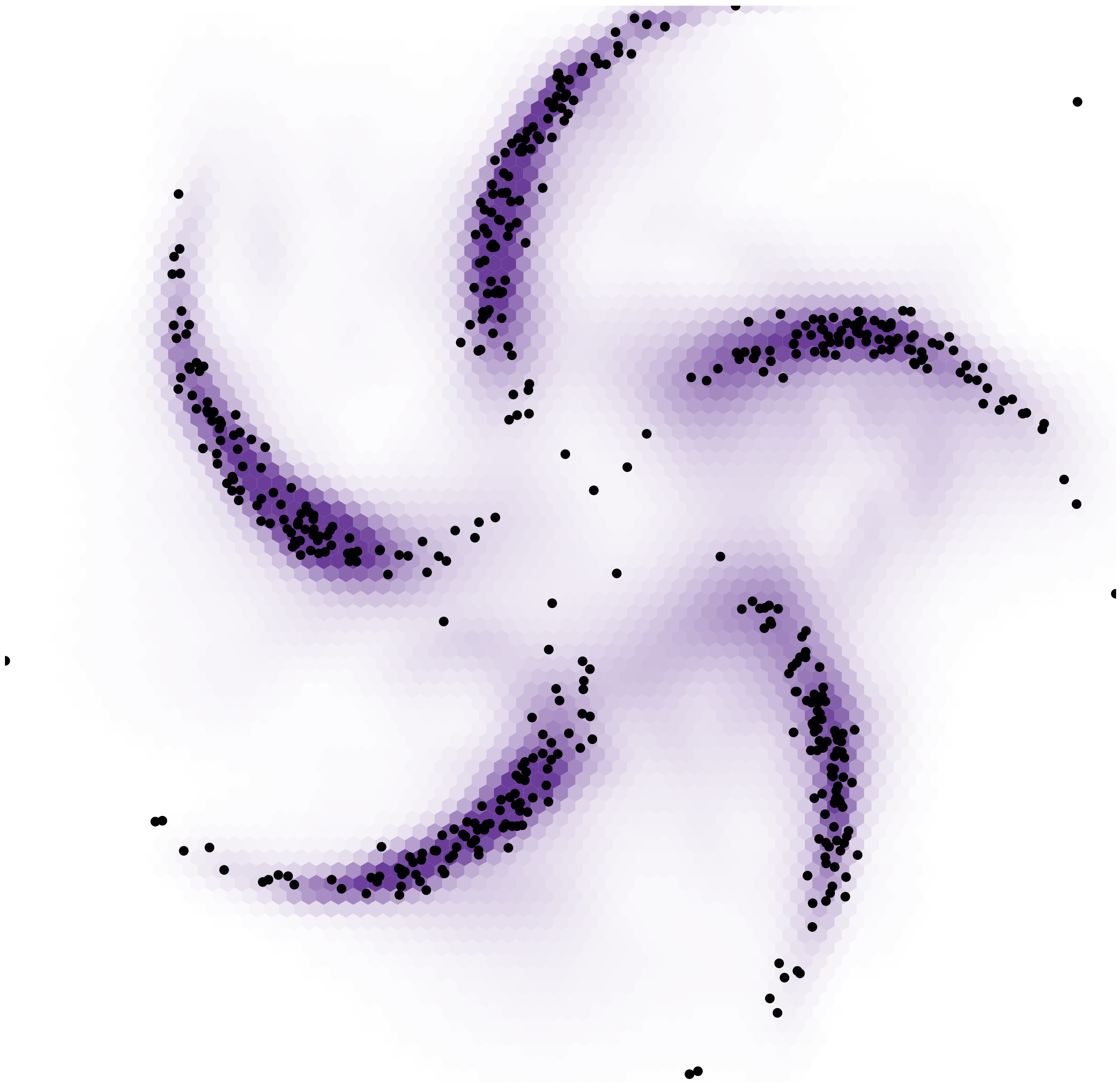
Probabilistic graphical models? [4,5,6]

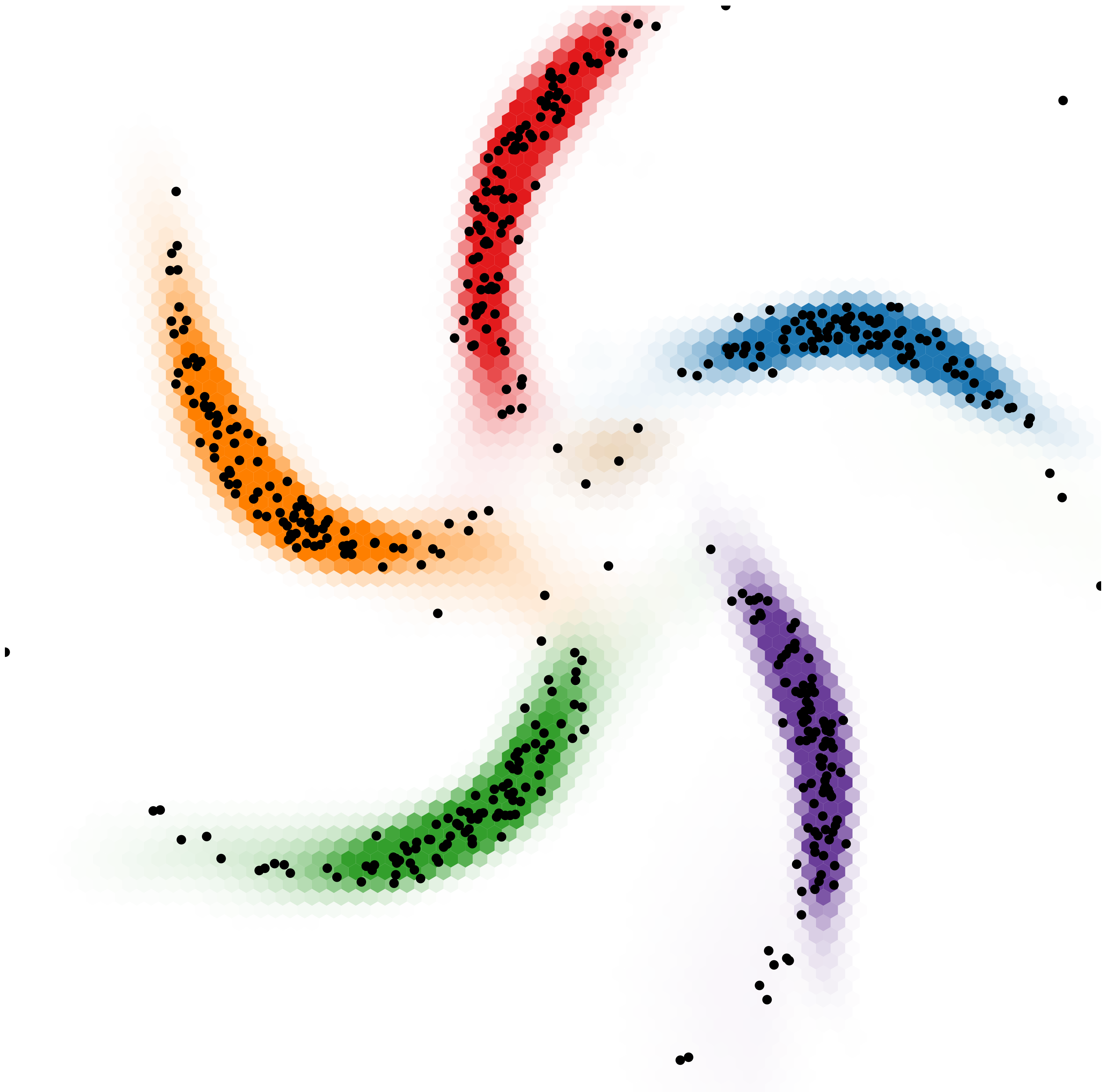


[1] Srivastava, Mansimov, Salakhutdinov. Unsupervised learning of video representations using LSTMs. ICML 2015.
[2] Ranzato, MarcAurelio, et al. Video (language) modeling: a baseline for generative models of natural videos. Preprint 2015.
[3] Sutskever, Hinton, and Taylor. The Recurrent Temporal Restricted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

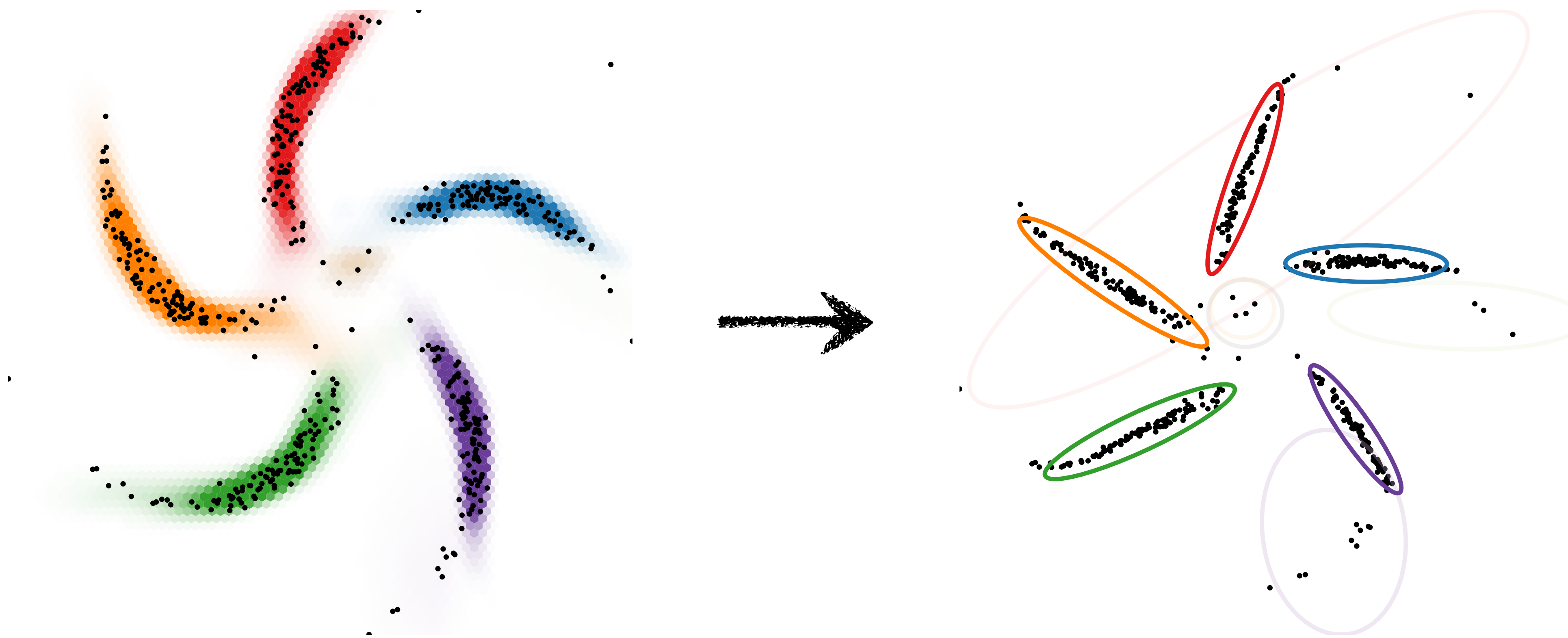




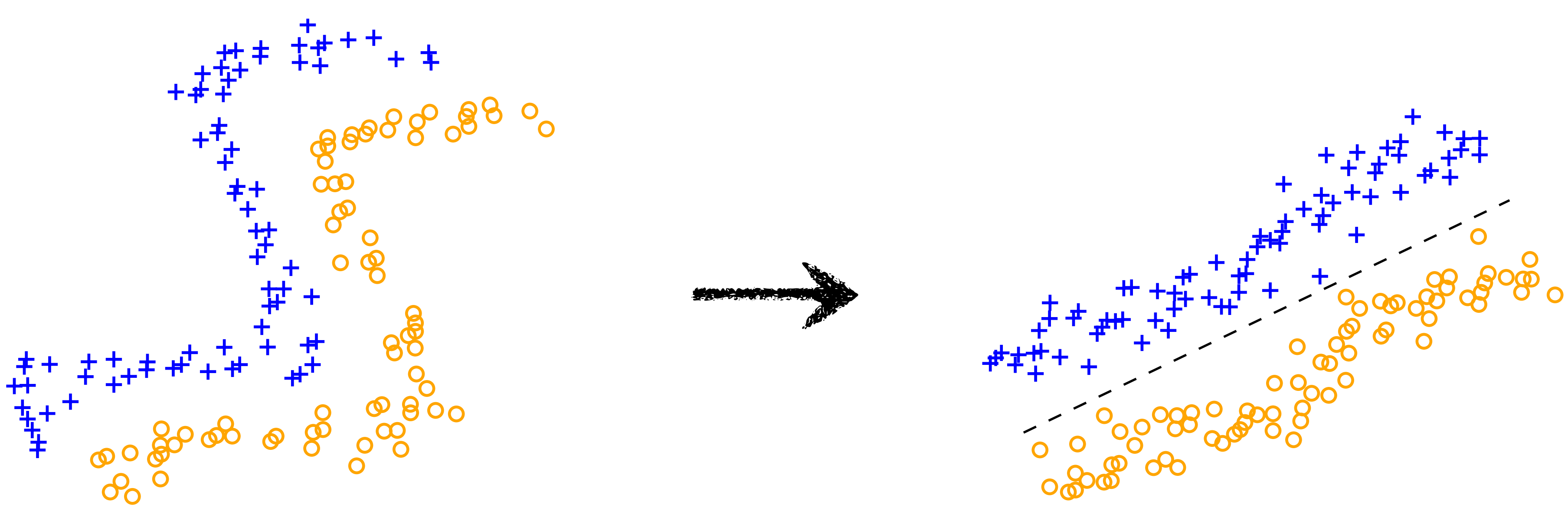




unsupervised learning



supervised learning



Probabilistic graphical models

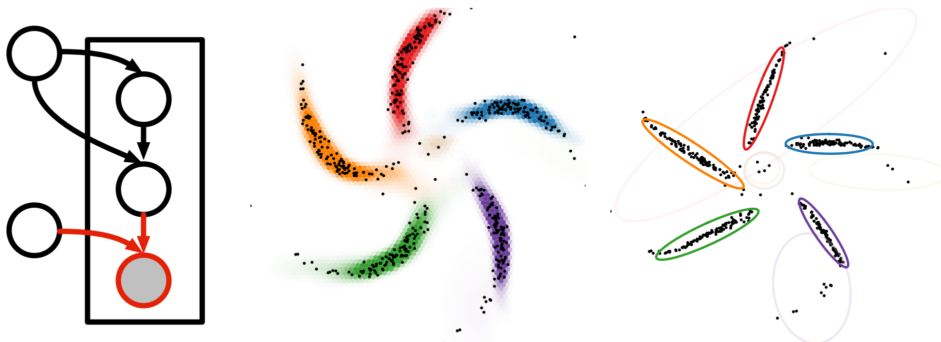
+ structured representations
+ priors and uncertainty
+ data and computational efficiency
– rigid assumptions may not fit
– feature engineering
– top-down inference

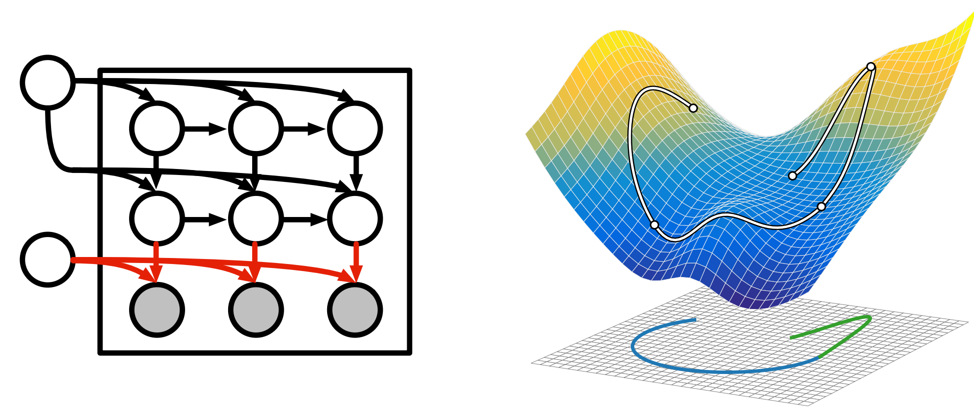
Deep learning

– neural net “goo”
– difficult parameterization
– can require lots of data
+ flexible
+ feature learning
+ recognition networks

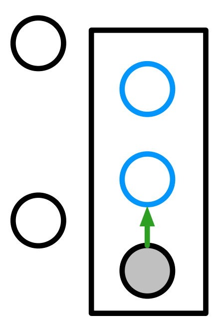

MAKE PGMS
GREAT AGAIN

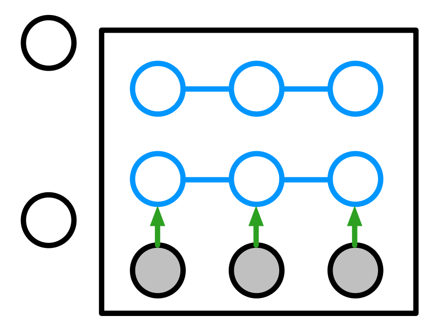
Modeling idea: graphical models on latent variables, neural network models for observations



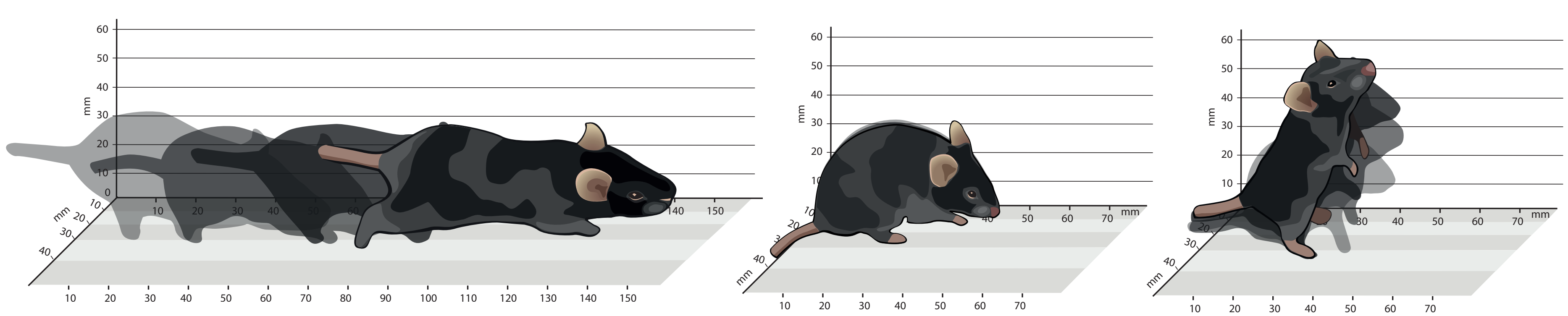


Inference: recognition networks output conjugate potentials, then apply fast graphical model inference

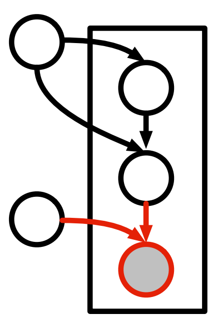


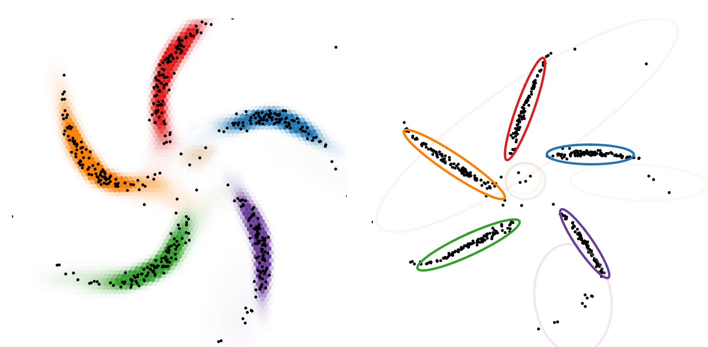


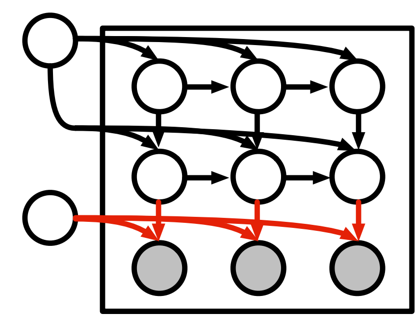
Application: learn syllable representation of behavior from video

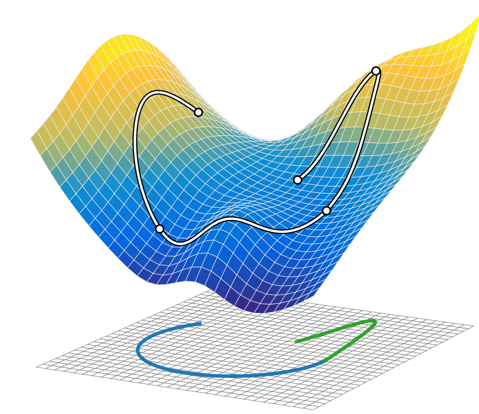


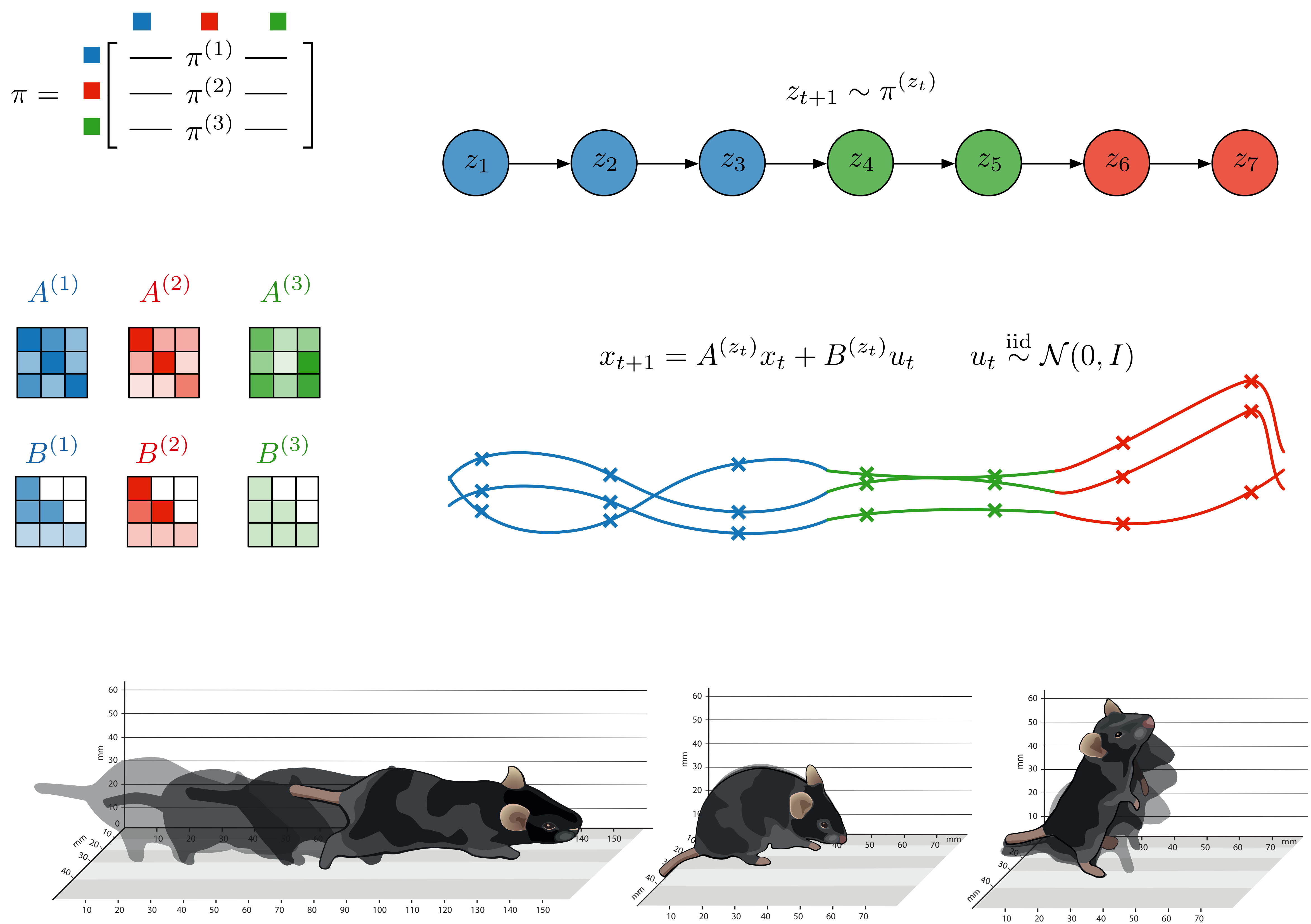
Modeling idea: graphical models on latent variables,
neural network models for observations



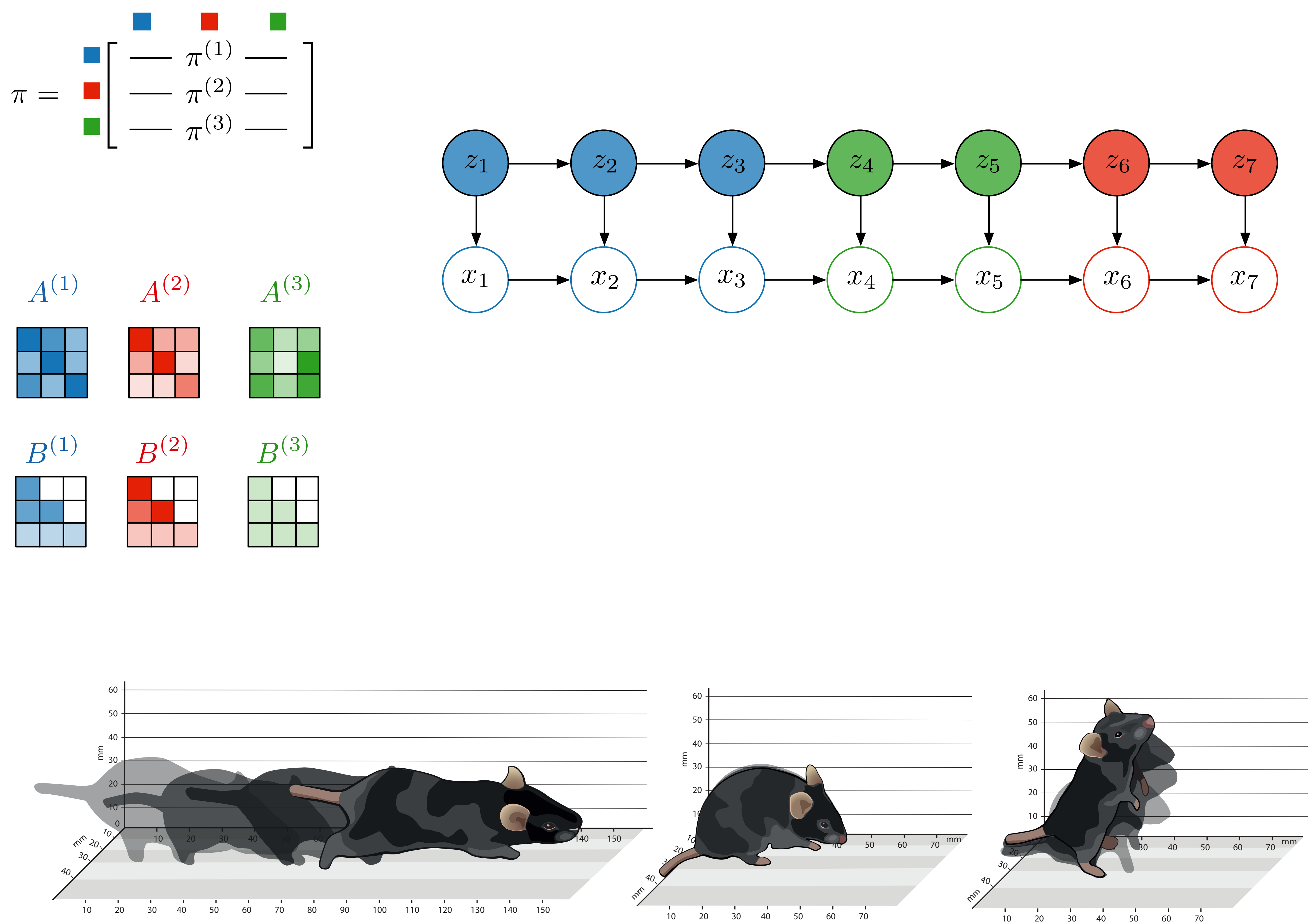




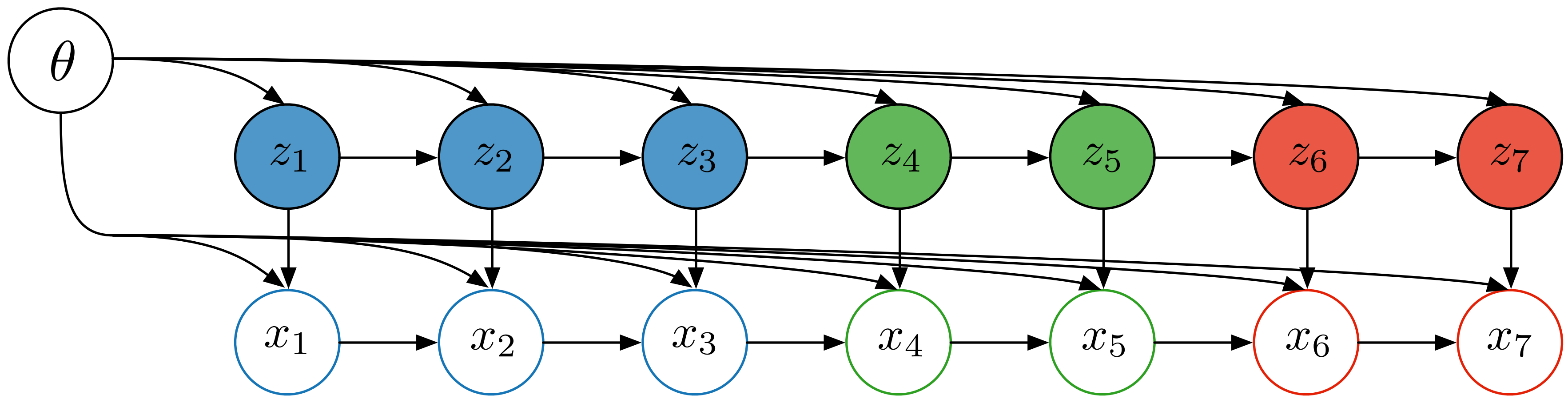


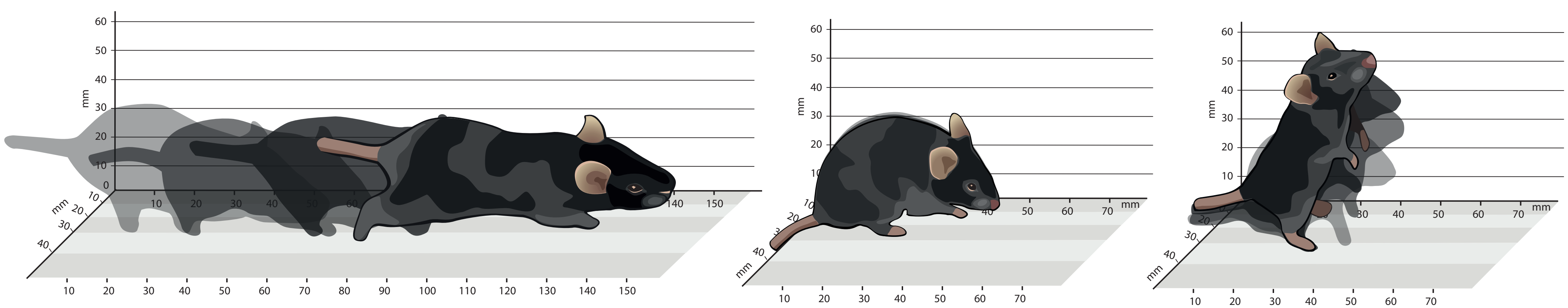


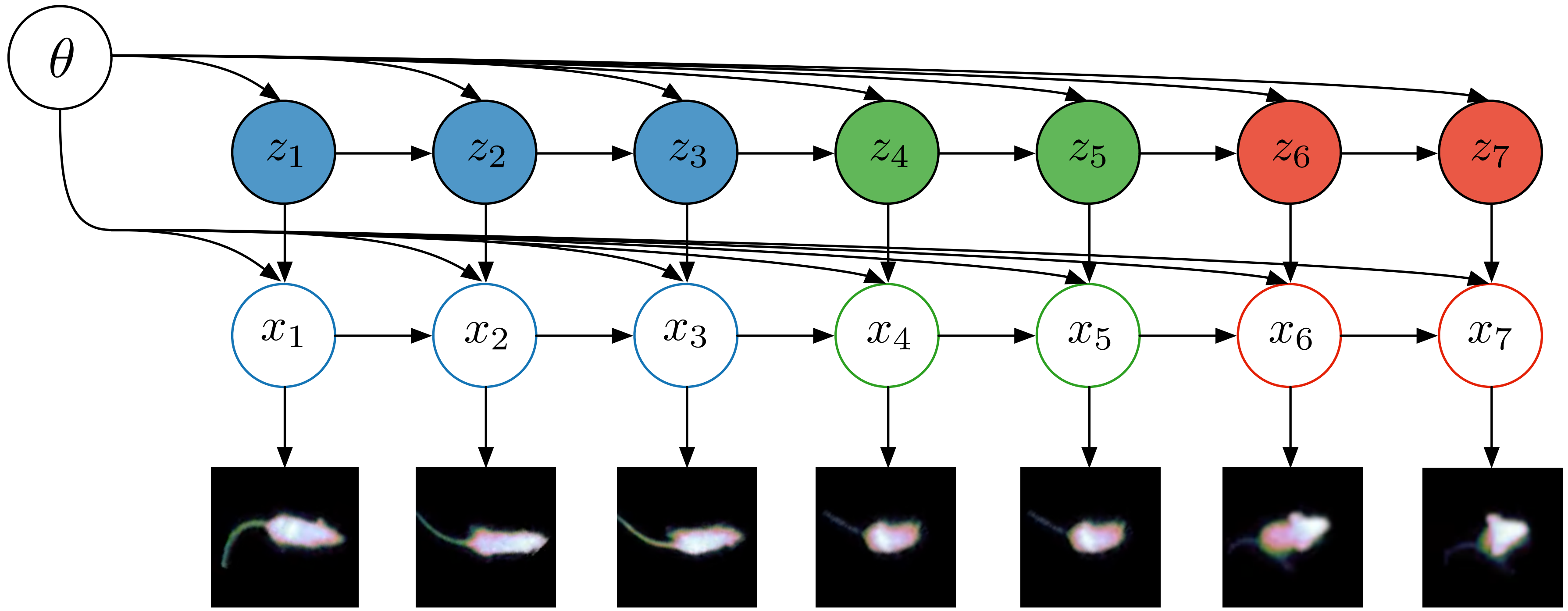
$\pi = \begin{bmatrix} — \pi^{(1)} — \\ — \pi^{(2)} — \\ — \pi^{(3)} — \end{bmatrix}$
$z_{t+1} \sim \pi^{(z_t)}$
z_1 z_2 z_3 z_4 z_5 z_6 z_7
$A^{(1)}$ $A^{(2)}$ $A^{(3)}$
$B^{(1)}$ $B^{(2)}$ $B^{(3)}$
$x_{t+1} = A^{(z_t)} x_t + B^{(z_t)} u_t \qquad u_t \overset{\text{iid}}{\sim} \mathcal{N}(0, I)$
mm



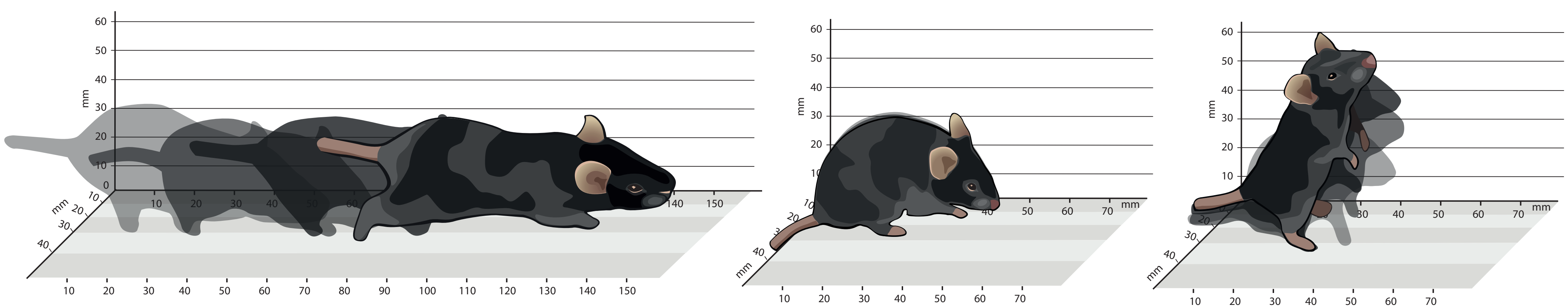
$\pi =$
$\pi^{(1)}$
$\pi^{(2)}$
$\pi^{(3)}$
z_1 z_2 z_3 z_4 z_5 z_6 z_7
x_1 x_2 x_3 x_4 x_5 x_6 x_7
$A^{(1)}$ $A^{(2)}$ $A^{(3)}$
$B^{(1)}$ $B^{(2)}$ $B^{(3)}$
mm


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

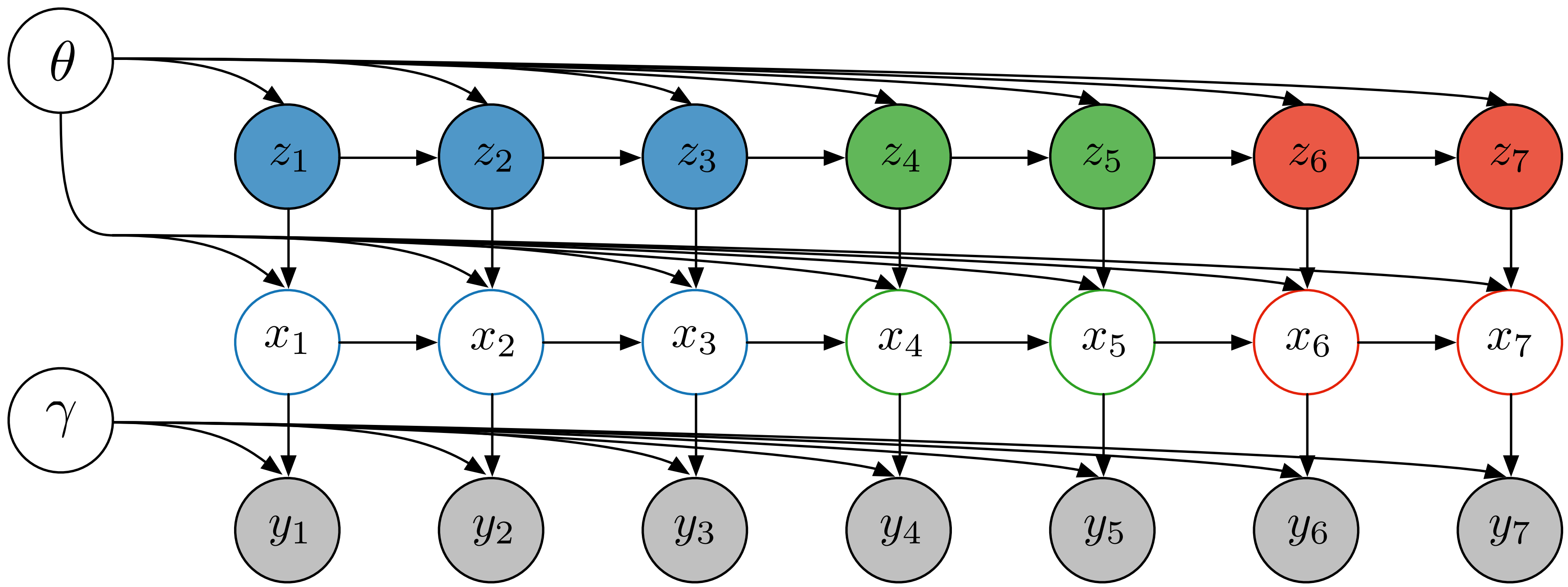

mm



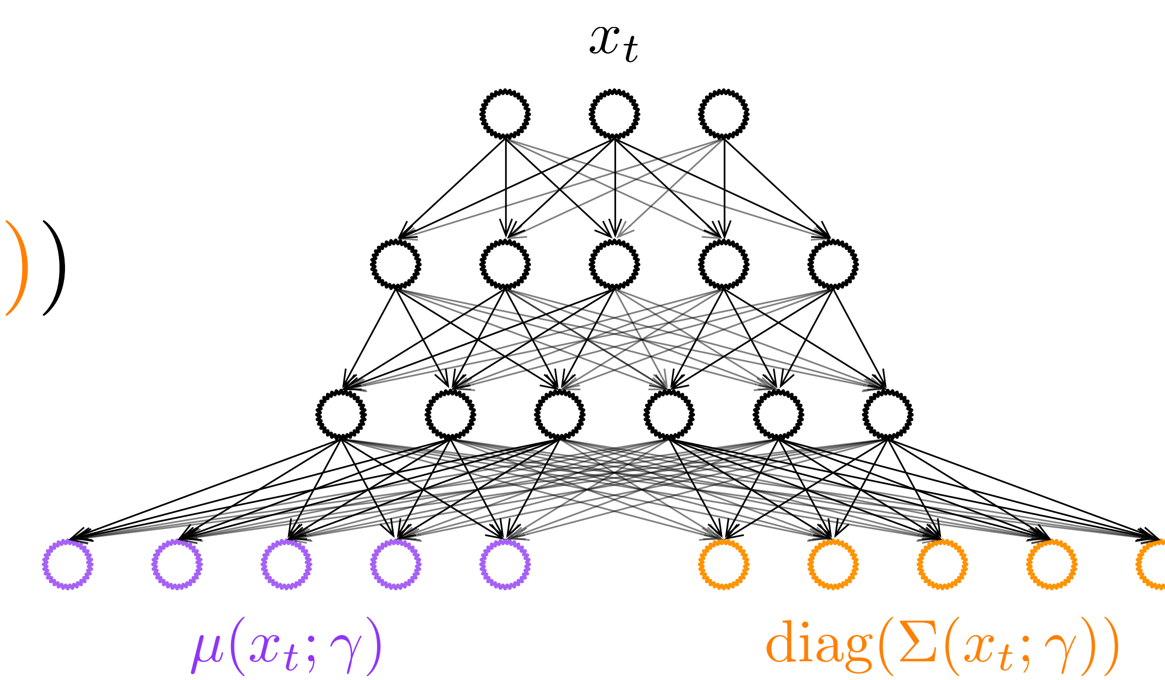
θ
z_1
z_2
z_3
z_4
z_5
z_6
z_7
x_1
x_2
x_3
x_4
x_5
x_6
x_7

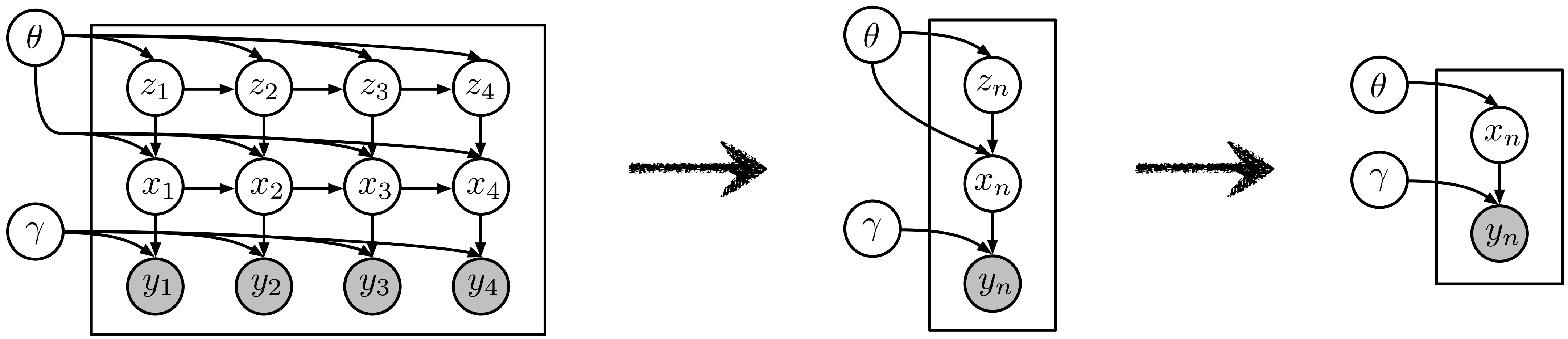


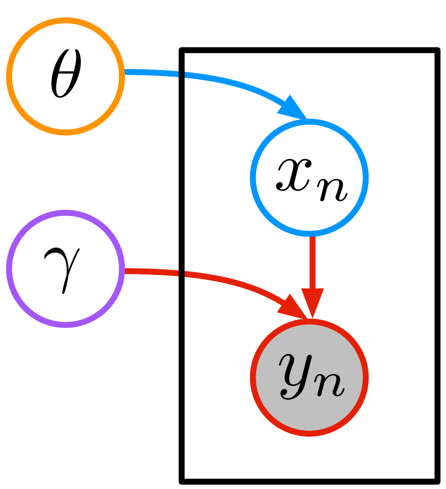
mm



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





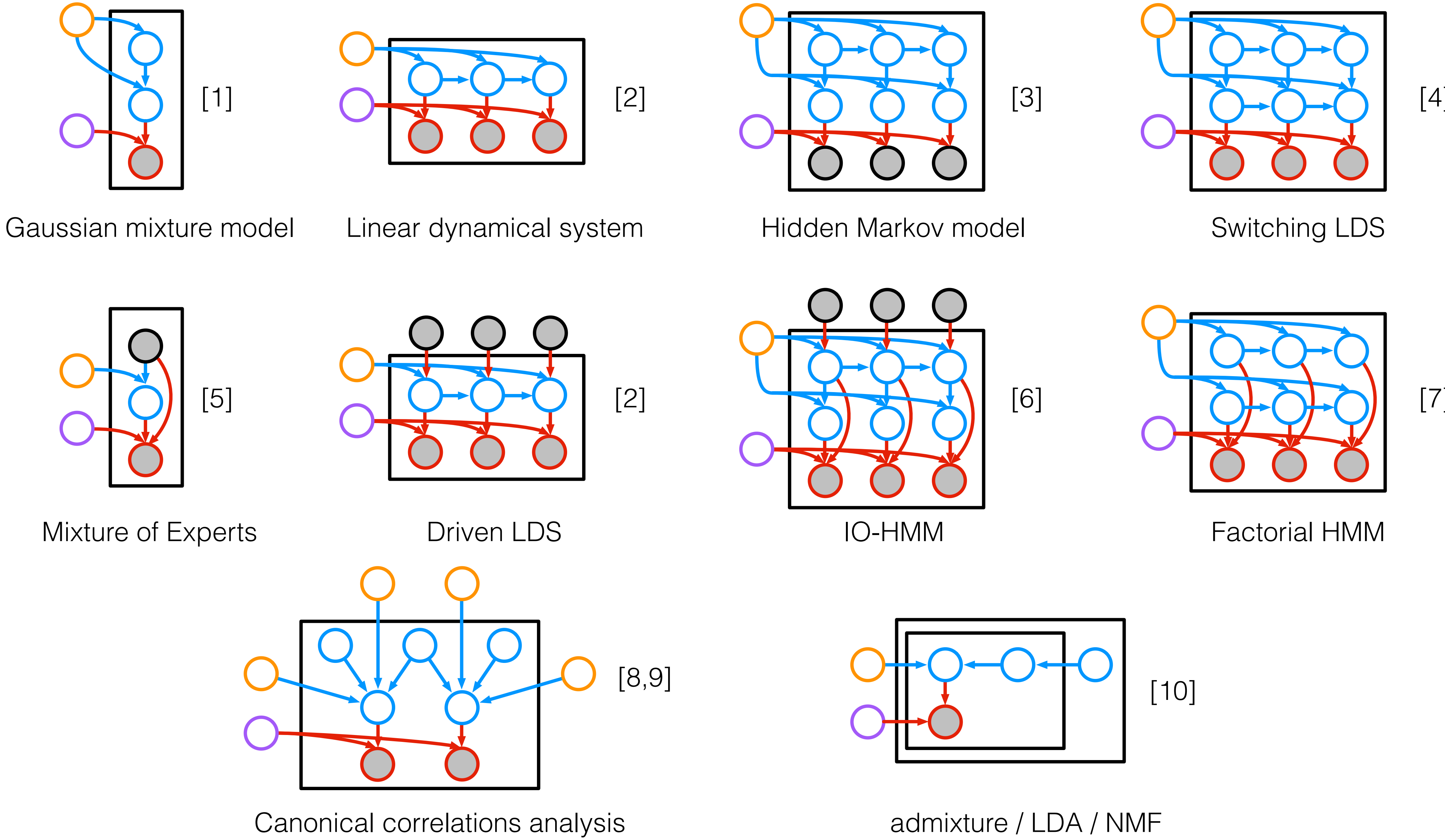


$p(\theta)$ conjugate prior on global variables

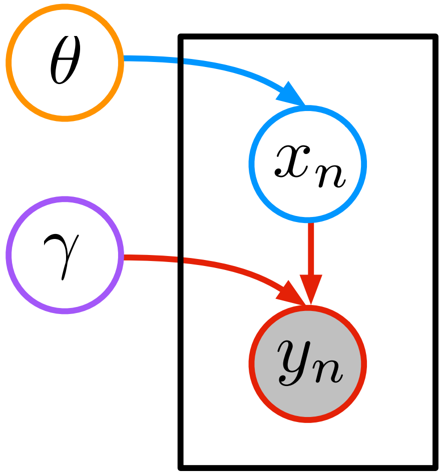
$p(x \mid \theta)$ exponential family on local variables

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

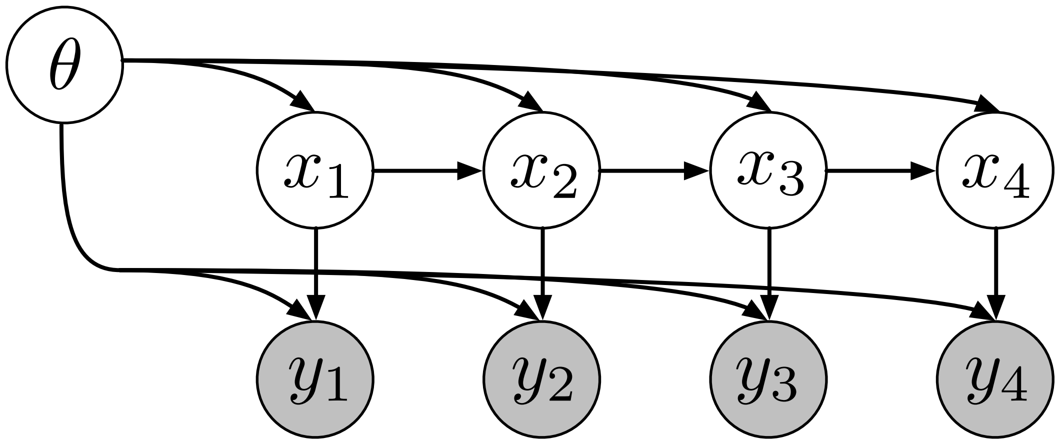
$p(y \mid x, \gamma)$ neural network observation model

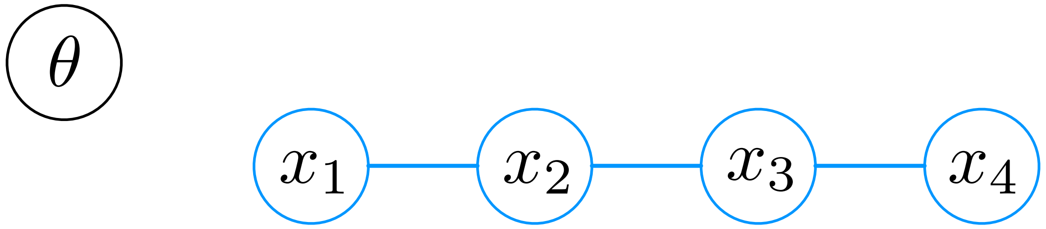


[1] Palmer, Wipf, Kreutz-Delgado, and Rao. Variational EM algorithms for non-Gaussian latent variable models. NIPS 2005.
[2] Ghahramani and Beal. Propagation algorithms for variational Bayesian learning. NIPS 2001.
[3] Beal. Variational algorithms for approximate Bayesian inference, Ch. 3. U of London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



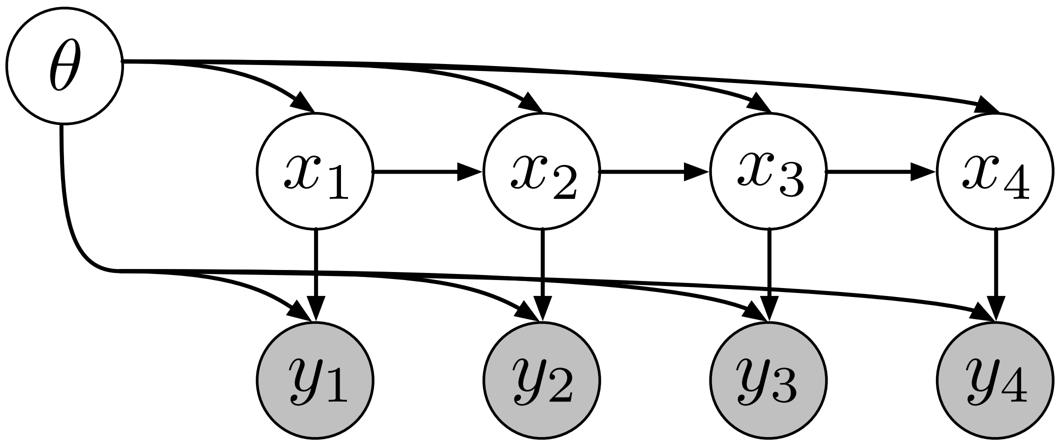


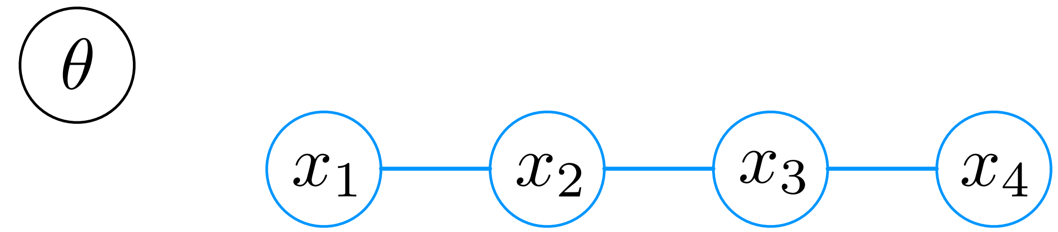
$p(x \,|\, \theta)$ is linear dynamical system
$p(y \,|\, x, \theta)$ is linear-Gaussian
$p(\theta)$ is conjugate prior

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

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

$$q(\theta) \leftrightarrow \eta_\theta \qquad q(x) \leftrightarrow \eta_x$$





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

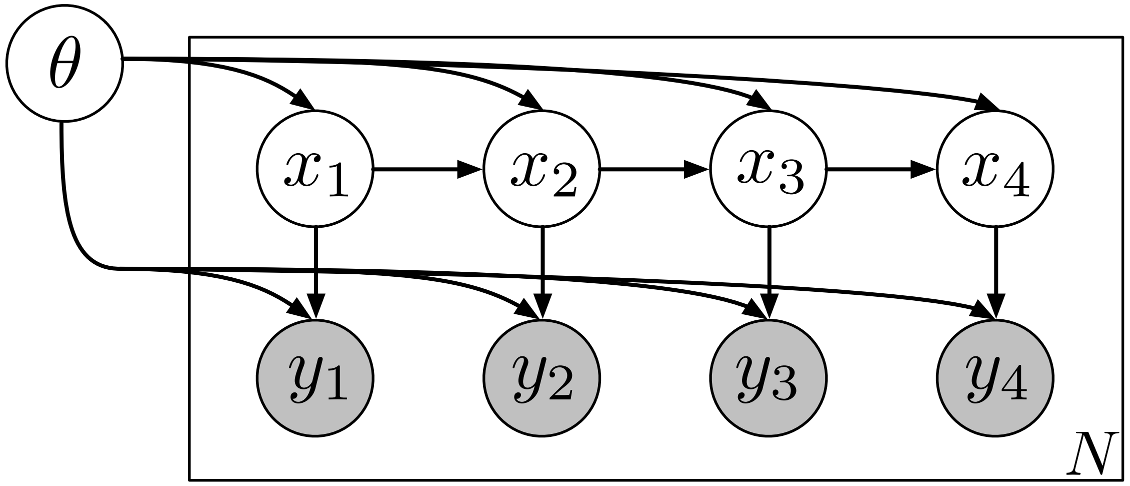
$$q(\theta)q(x) \approx p(\theta, x \mid y)$$

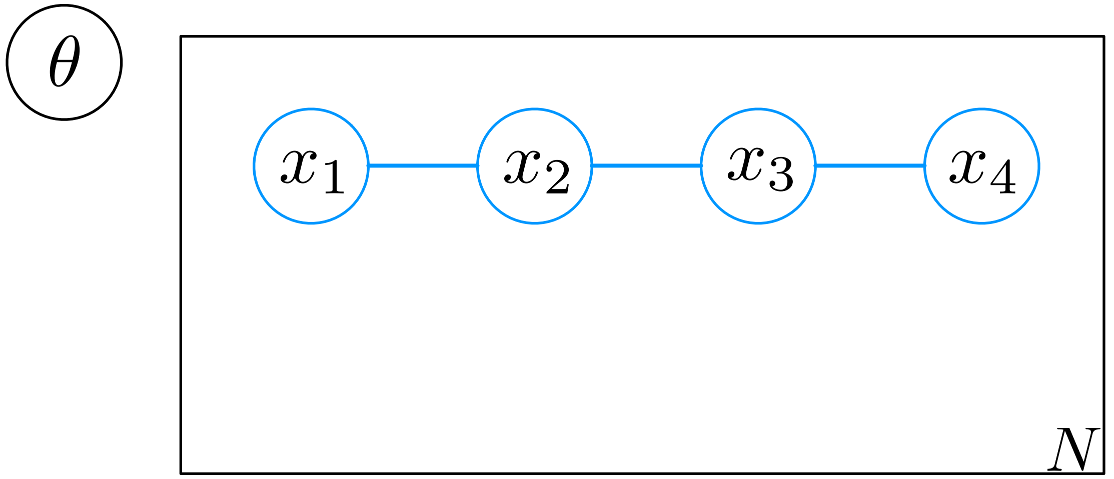
$$\mathcal{L}(\eta_\theta, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(x)}\left[\log \frac{p(\theta, x, y)}{q(\theta)q(x)}\right]$$

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

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

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

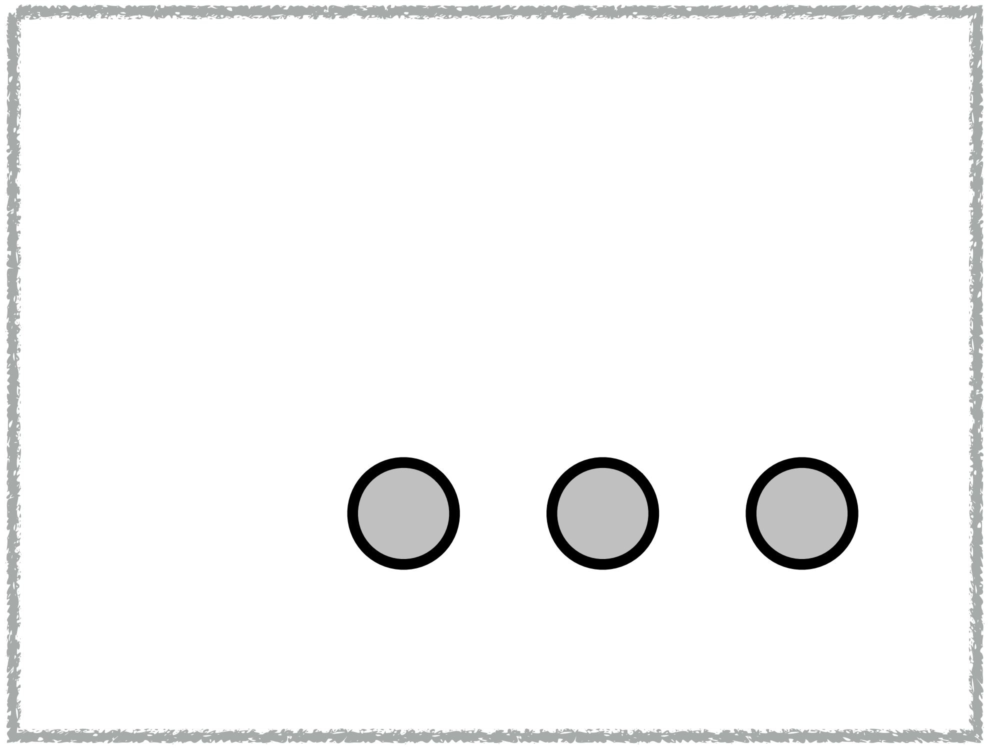
$$\mathcal{L}(\eta_\theta, \eta_x) \triangleq \mathbb{E}_{q(\theta)q(x)}\left[\log \frac{p(\theta,x,y)}{q(\theta)q(x)}\right]$$

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

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

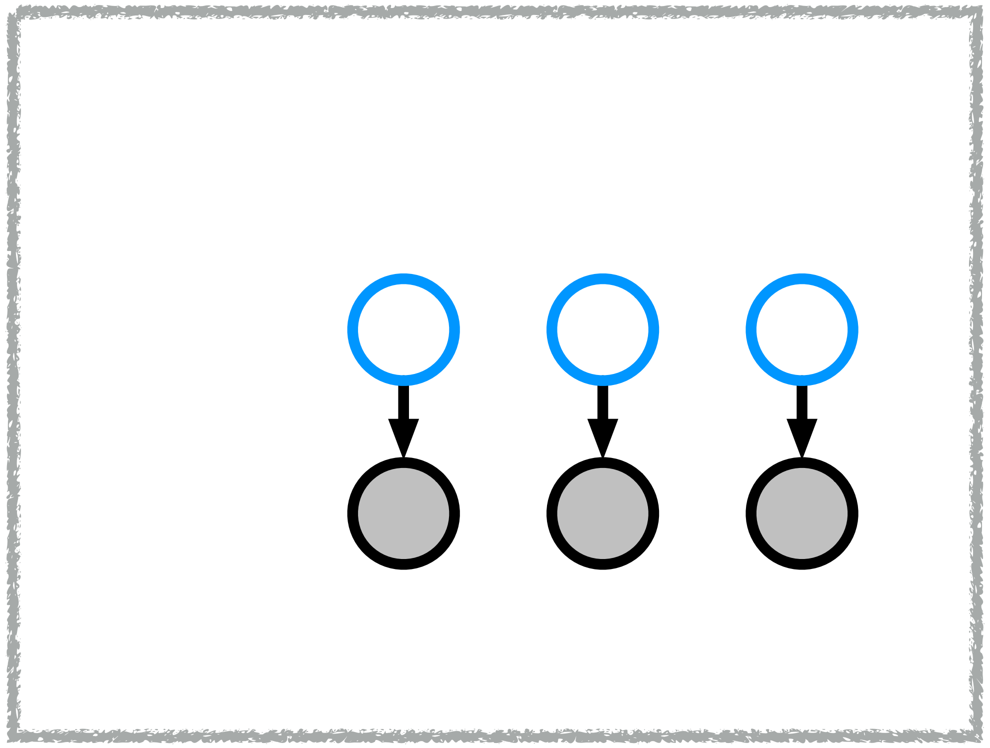
$$\widetilde{\nabla}\mathcal{L}_{\text{SVI}}(\eta_\theta) = \eta_\theta^0 + \sum_{n=1}^{N} \mathbb{E}_{q^*(x_n)}(t_{xy}(x_n, y_n), 1) - \eta_\theta$$

Step 1: compute evidence potentials



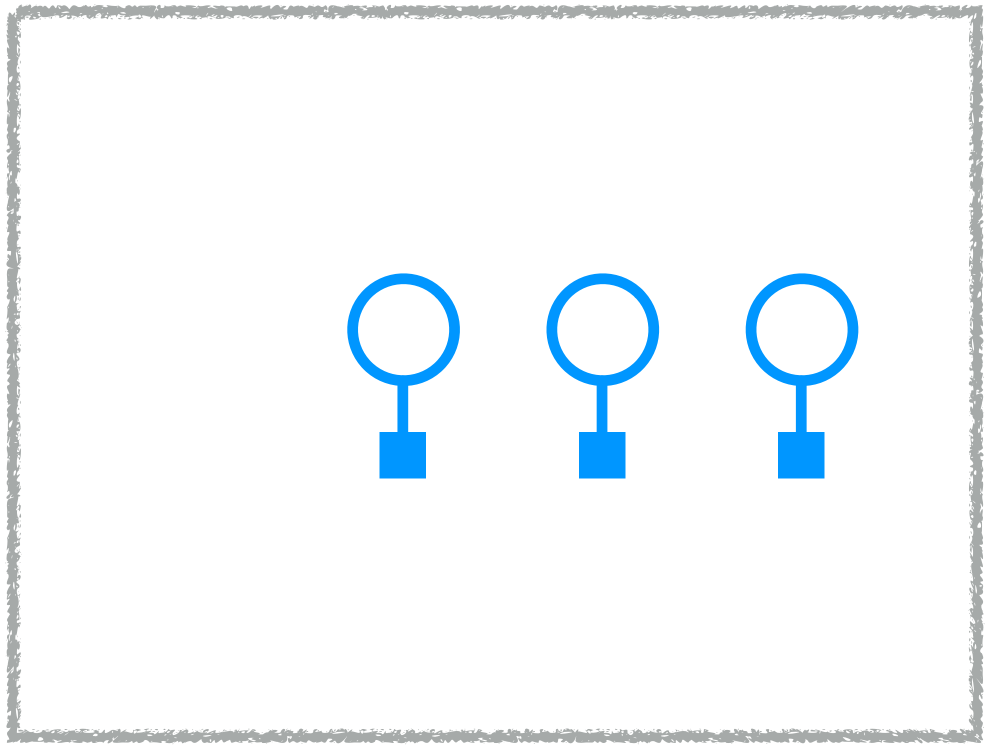
[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

Step 1: compute evidence potentials

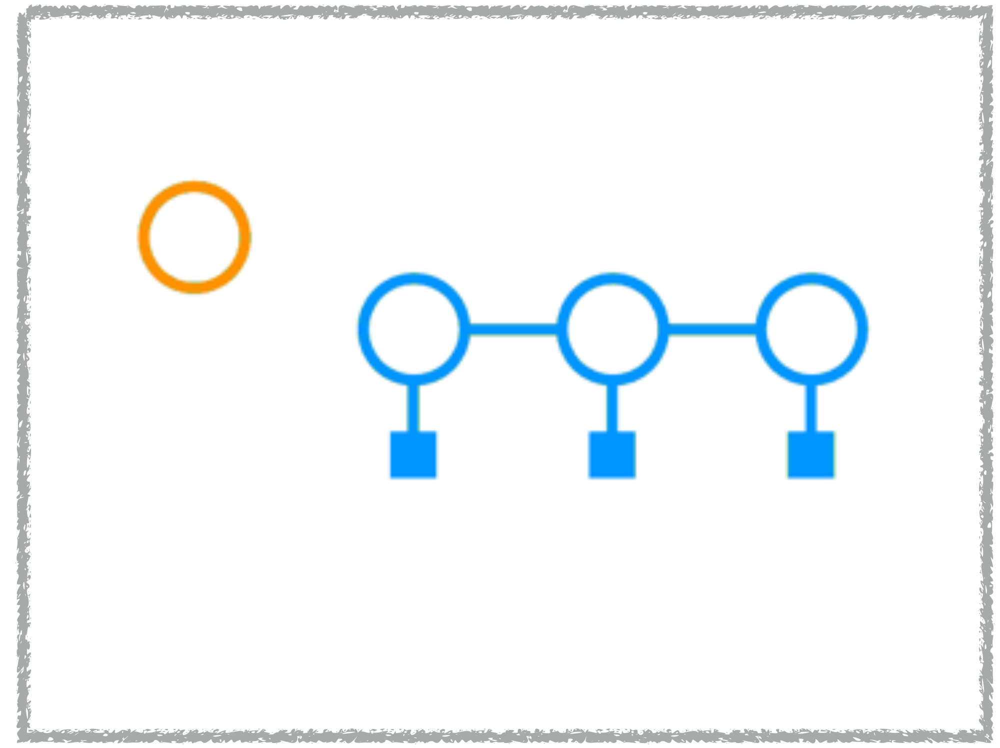


[1] **Johnson** and Willsky. Stochastic variational inference for Bayesian time series models. ICML 2014.
[2] Foti, Xu, Laird, and Fox. Stochastic variational inference for hidden Markov models. NIPS 2014.

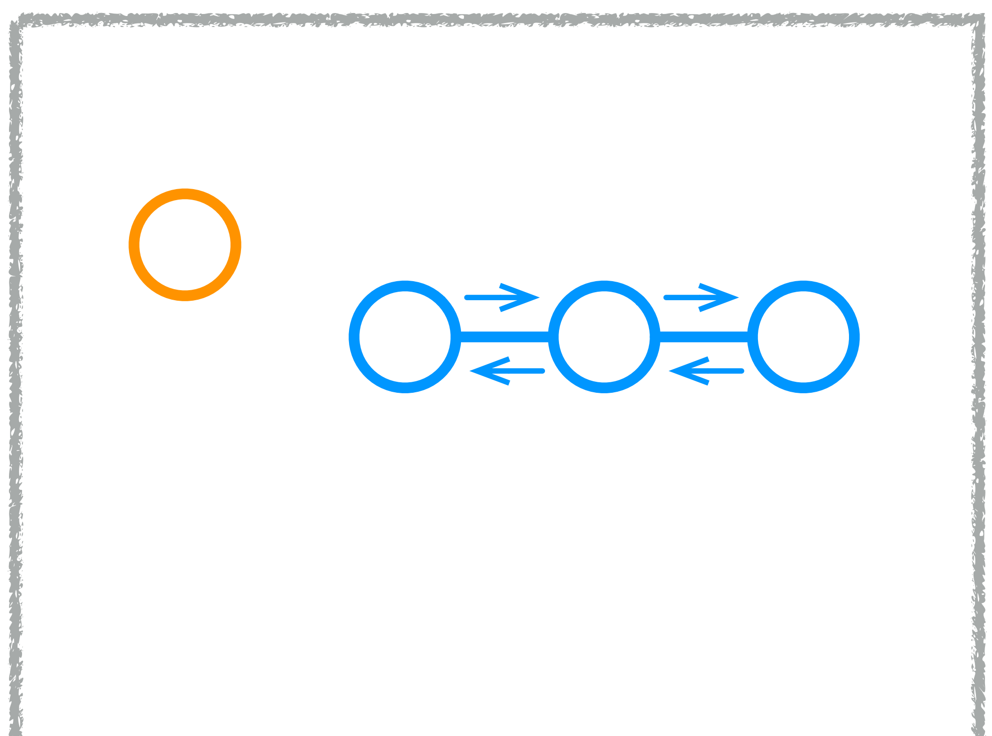
Step 1: compute evidence potentials



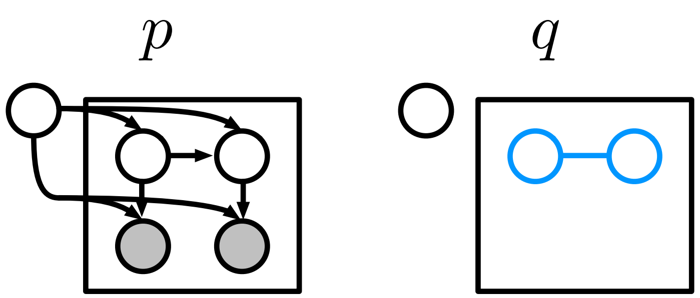
Step 2: run fast message passing

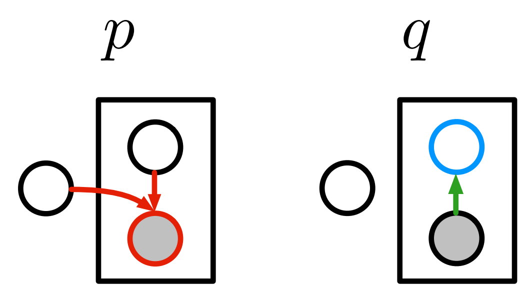


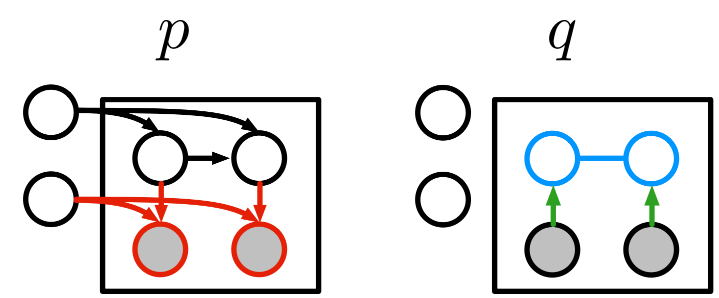
Step 3: compute natural gradient



[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.







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

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

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

Natural gradient SVI

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

Variational autoencoders [1,2]

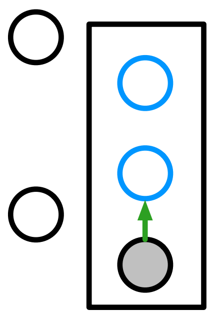
– suboptimal local factor
+ fast for general obs.
– ϕ does all local inference
– no natural gradients

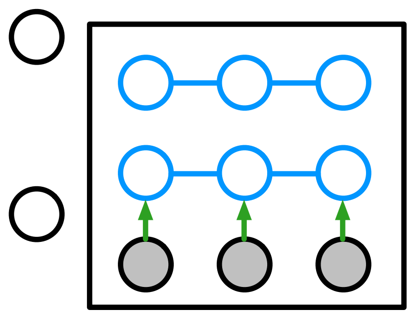
Structured VAEs

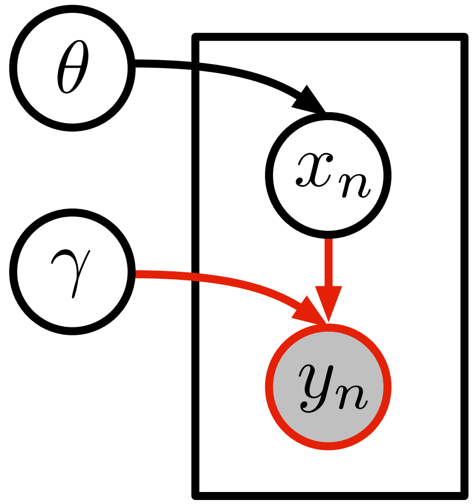
± optimal given conj. evidence
+ fast for general obs.
+ exploits conj. graph structure
+ natural gradients on η_θ

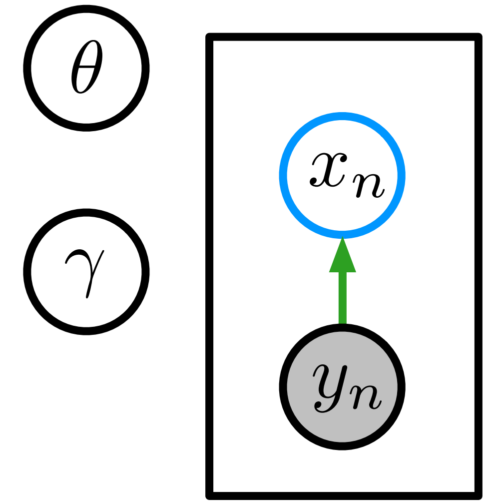
[1] Kingma and Welling. Auto-encoding variational Bayes. ICLR 2014.
[2] Rezende, Mohamed, and Wierstra. Stochastic backpropagation and approximate inference in deep generative models. ICML 2014

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



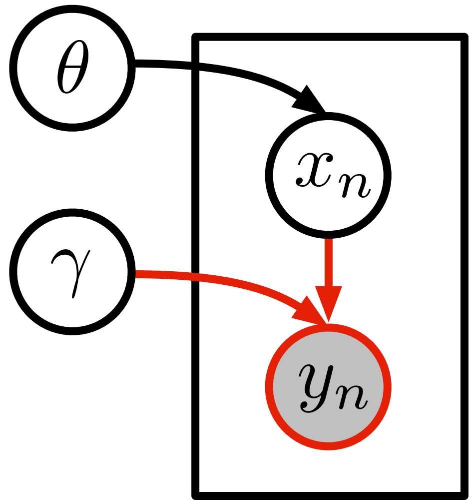


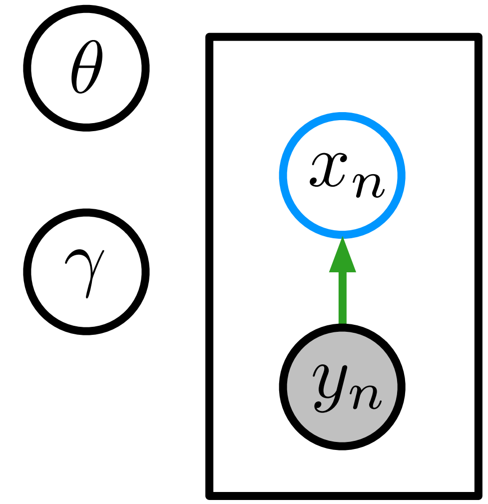




$$\mathcal{L}[\,q(\theta)q(\gamma)q(x)\,] \triangleq \mathbb{E}_{q(\theta)q(\gamma)q(x)}\left[\log \frac{p(\theta,\gamma,x)p(y\mid x,\gamma)}{q(\theta)q(\gamma)q(x)}\right]$$

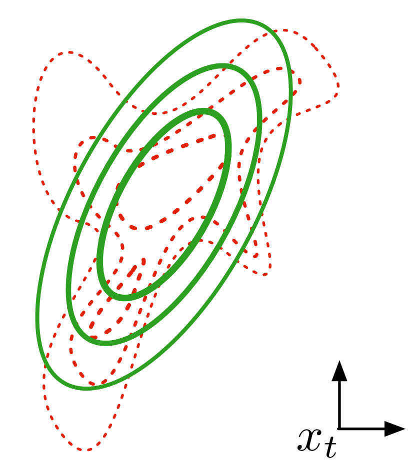
$$q(\theta) \leftrightarrow \eta_\theta \qquad q(\gamma) \leftrightarrow \eta_\gamma \qquad q(x) \leftrightarrow \eta_x$$





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

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



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

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

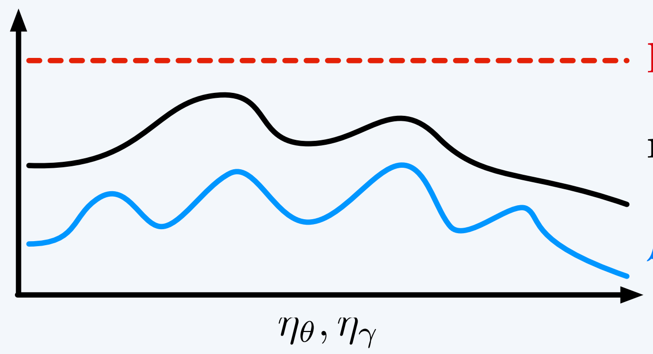
where $\psi(x; y, \phi)$ is a conjugate potential for $p(x \mid \theta)$

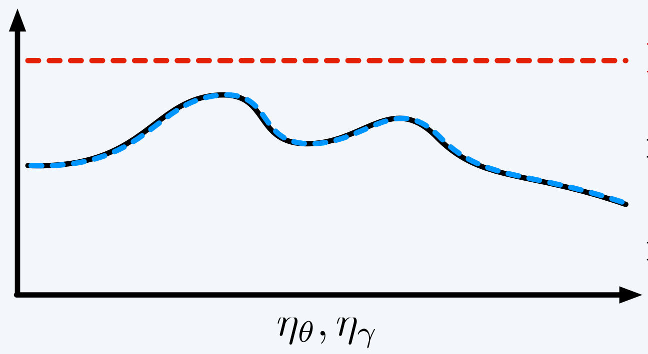
$$\eta_x^*(\eta_\theta, \phi) \triangleq \arg\max_{\eta_x} \widehat{\mathcal{L}}(\eta_\theta, \eta_x, \phi) \qquad \mathcal{L}_{\text{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \triangleq \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi))$$

Fact (conjugate graphical models are easy)

The local variational parameter $\eta_x^*(\eta_\theta, \phi)$ is easy to compute.

Proposition (log evidence lower bound)





if $\exists\, \phi \in \mathbb{R}^m$ with $\psi(x; y, \phi) = \mathbb{E}_{q(\gamma)} \log p(y \,|\, x, \gamma)$

Proposition (reparameterization trick)

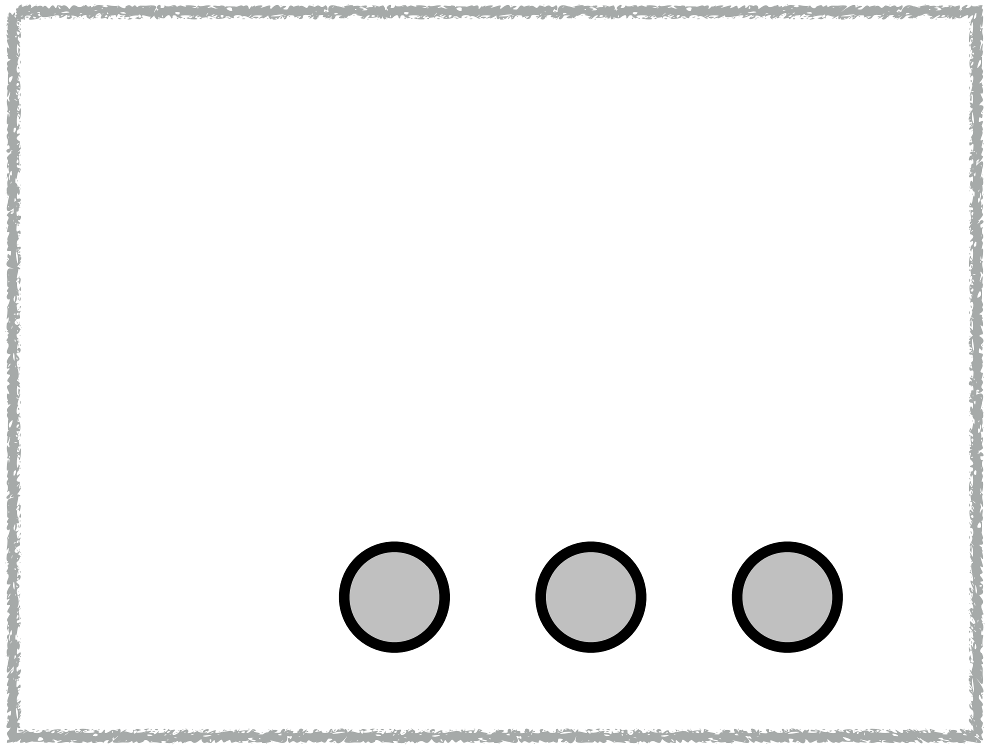
Estimate $\nabla_{\eta_\gamma, \phi} \mathcal{L}_{\mathrm{SVAE}}(\eta_\theta, \eta_\gamma, \phi)$ with samples $\hat{\gamma} \sim q(\gamma)$ and $\hat{x} \sim q^*(x \,|\, \phi)$ via

$$\mathcal{L}_{\mathrm{SVAE}}(\eta_\theta, \eta_\gamma, \phi) \approx \log p(y \,|\, \hat{x}, \hat{\gamma}) - \mathrm{KL}(q(\theta) q(\gamma) q^*(x \,|\, \phi) \,\|\, p(\theta, \gamma, x))$$

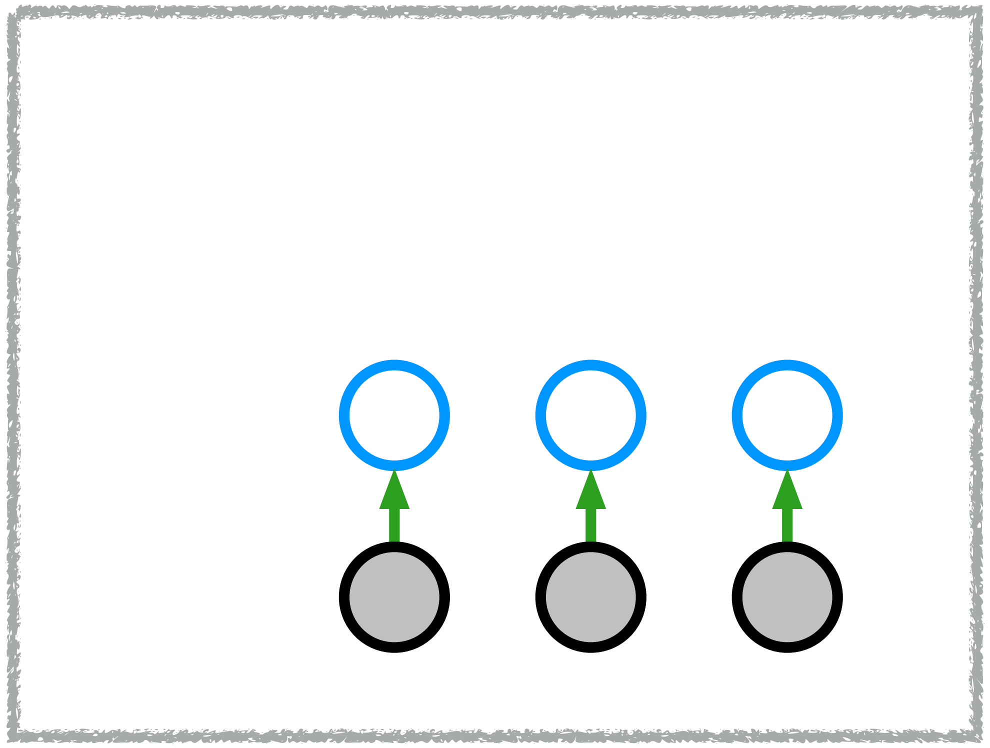
Proposition (easy natural gradient)

$$\widetilde{\nabla}_{\eta_\theta} \mathcal{L}_{\mathrm{SVAE}}(\eta_\theta, \eta_\gamma, \phi) = (\eta_\theta^0 + \mathbb{E}_{q^*(x \,|\, \phi)}(t_x(x), 1) - \eta_\theta) + (\nabla_{\eta_x} \mathcal{L}(\eta_\theta, \eta_\gamma, \eta_x^*(\eta_\theta, \phi)), 0)$$

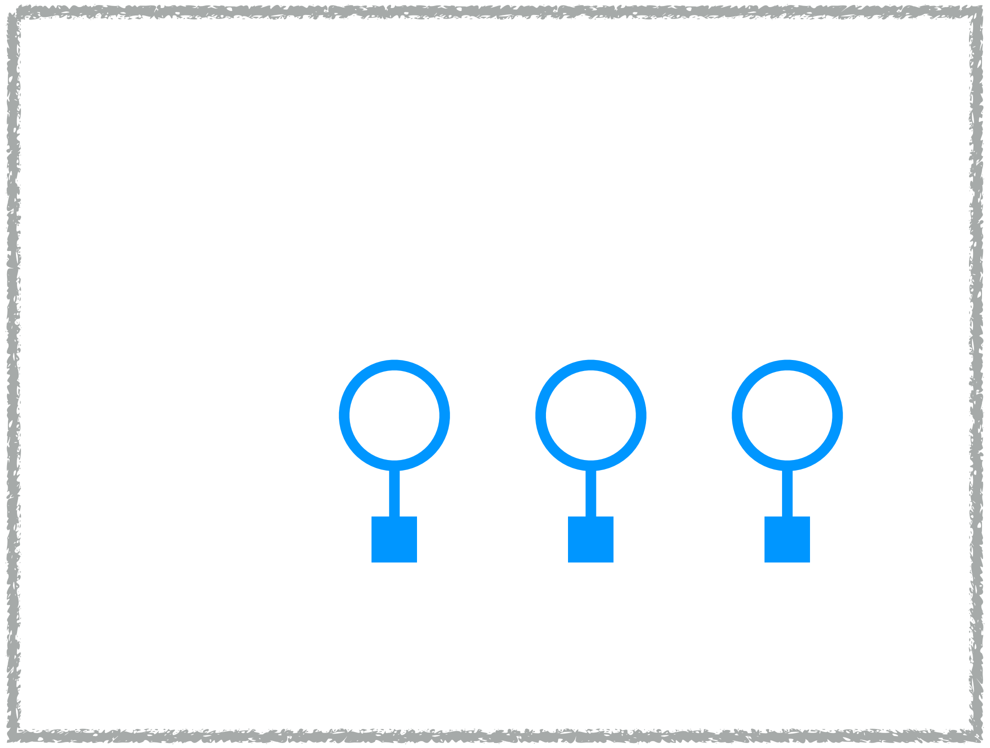
Step 1: apply recognition network



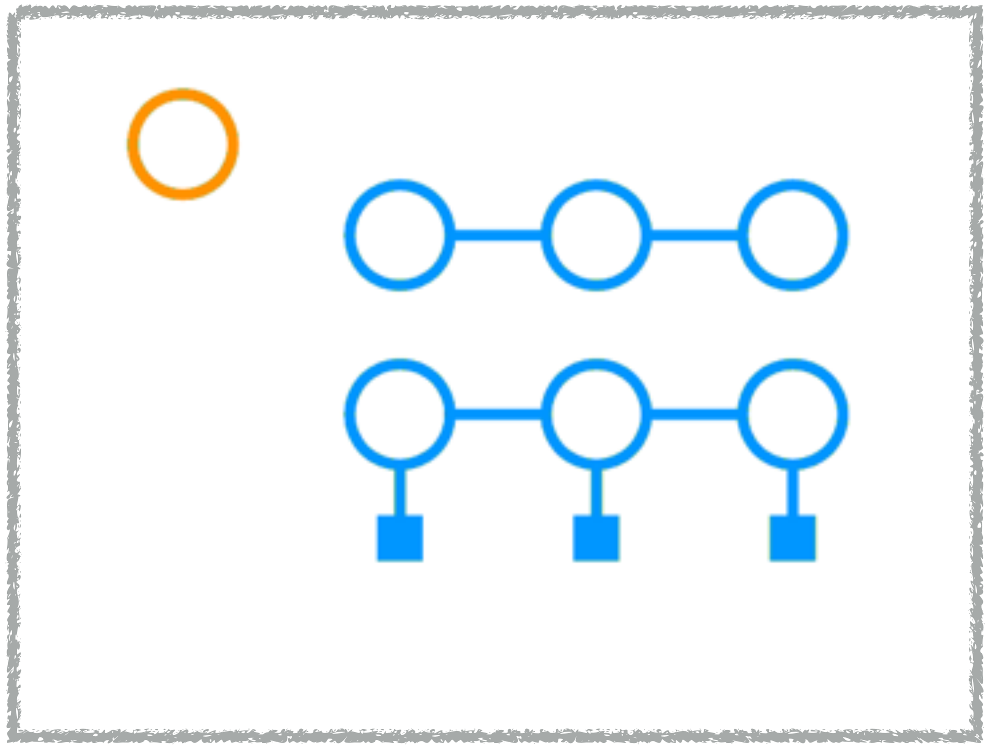
Step 1: apply recognition network



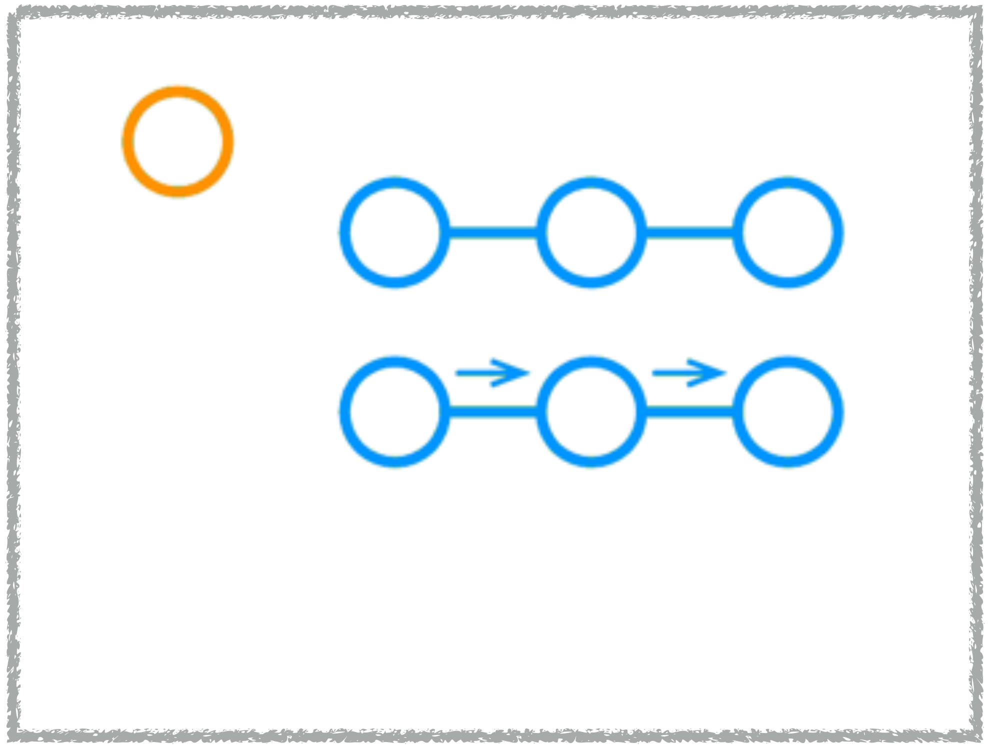
Step 1: apply recognition network



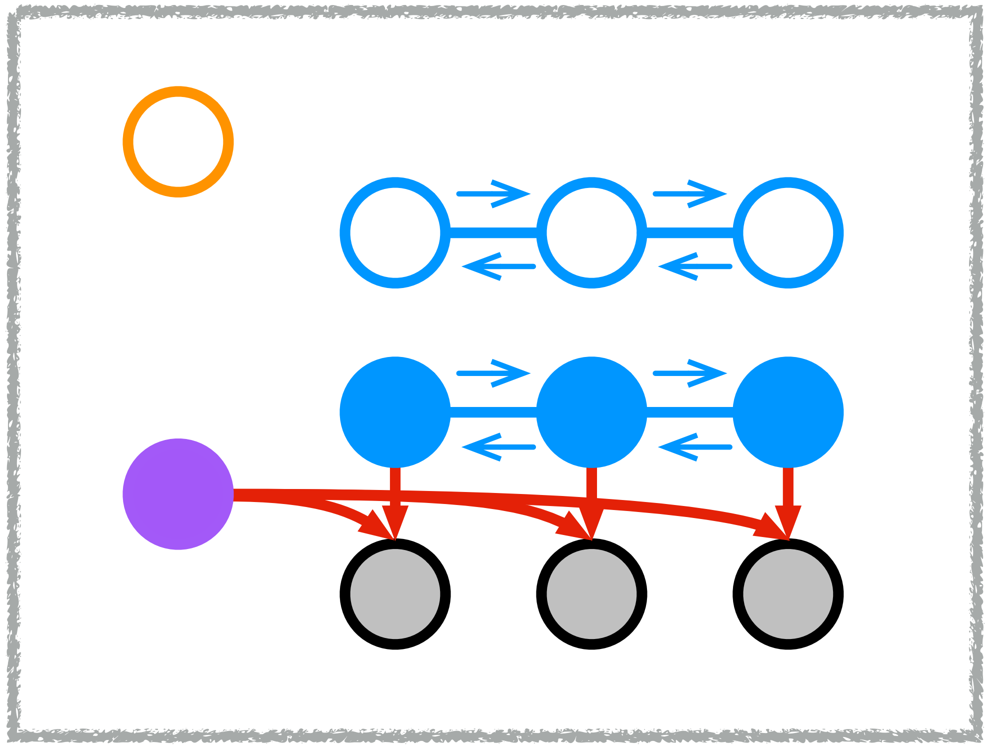
Step 2: run fast PGM algorithms



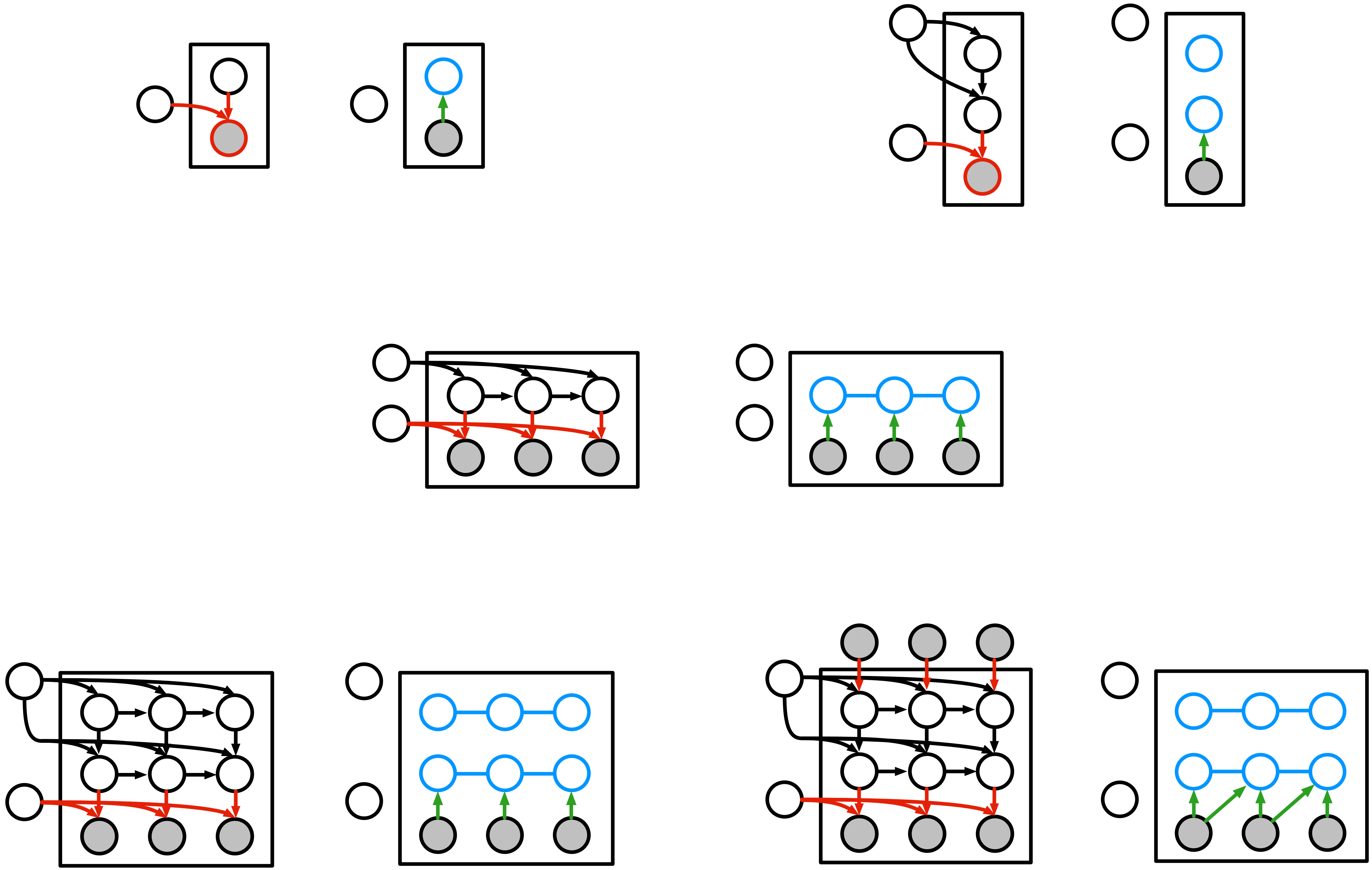
Step 3: sample, compute flat grads

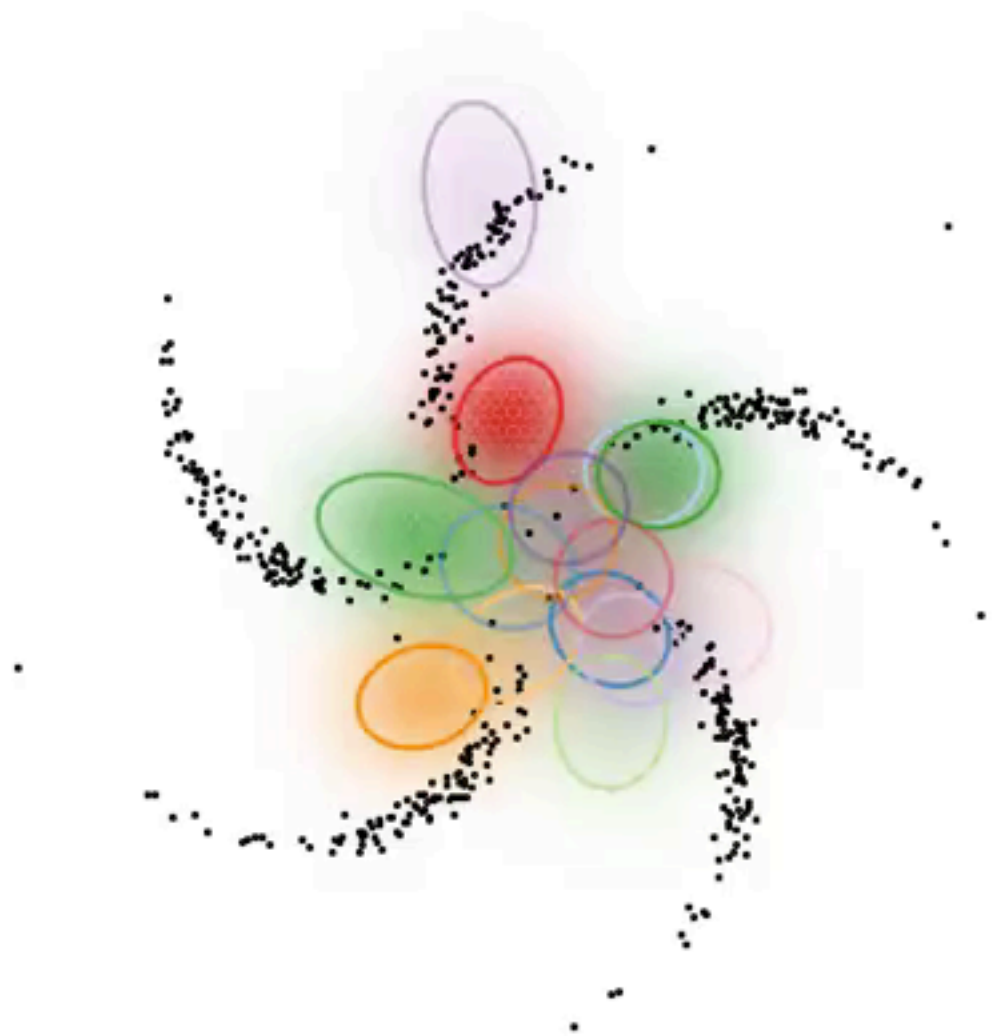


Step 4: compute natural gradient

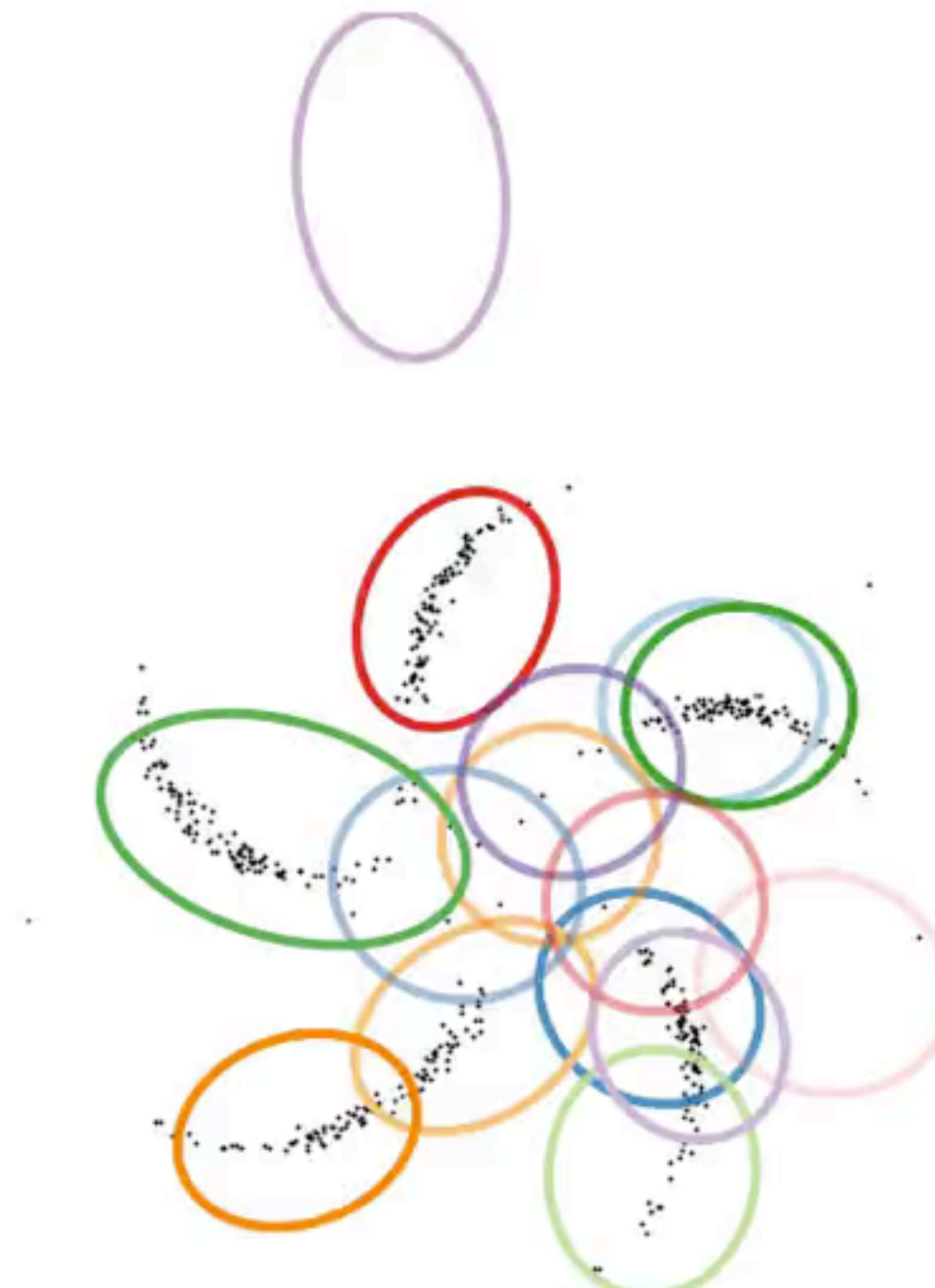


```python
from autograd import value_and_grad as vgrad
from util import add, sub, scale, unbox


def make_gradfun(run_inference, recognize, loglike, eta_prior, return_flat=False):
    saved = lambda: None

    def mc_vlb(eta, gamma, phi, y_n, N, L):
        T = y_n.shape[0]
        nn_potentials = recognize(y_n, phi)
        samples, stats, global_kl, local_kl = run_inference(
            eta_prior, eta, nn_potentials, num_samples=L)
        saved.stats = scale(N, unbox(stats))
        return -global_kl + N * (-local_kl + loglike(y_n, samples, gamma))

    def gradfun(y_n, N, L, eta, gamma, phi):
        objective = lambda (gamma, phi): mc_vlb(eta, gamma, phi, y_n, N, L)
        vlb, (gamma_grad, phi_grad) = vgrad(objective)((gamma, phi))
        eta_natgrad = sub(add(eta_prior, saved.stats), eta)
        return vlb, (eta_natgrad, gamma_grad, phi_grad)

    return gradfun
```

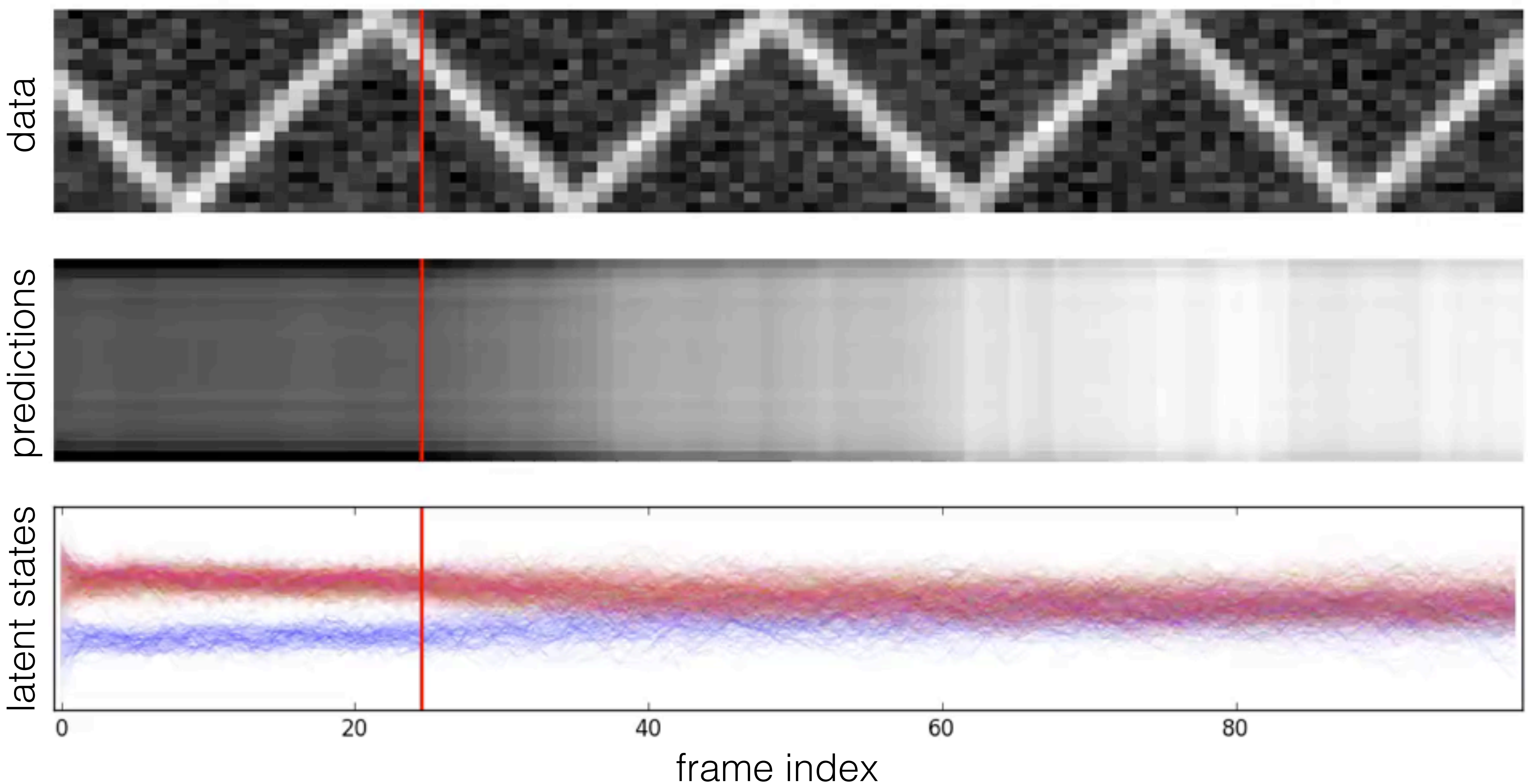




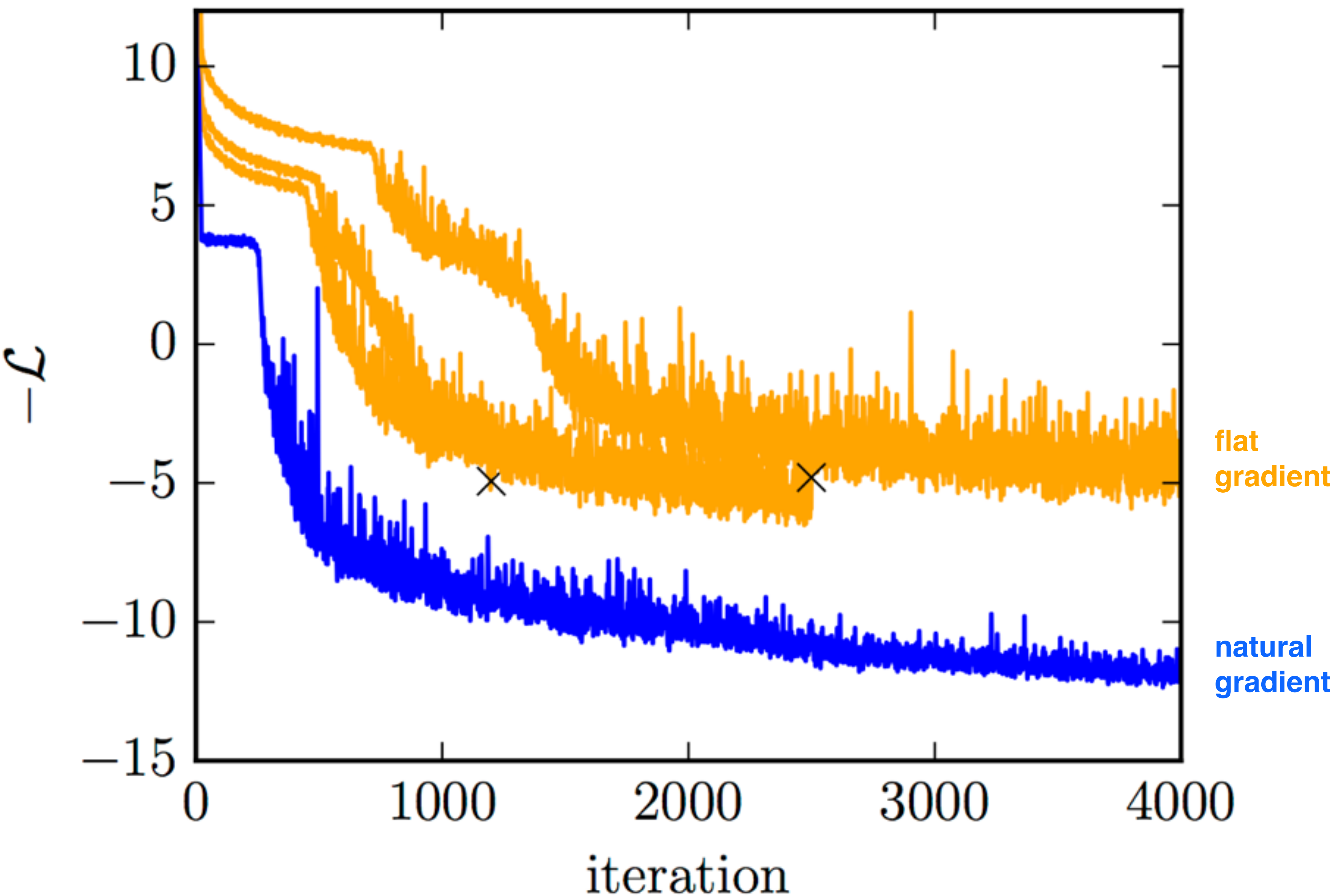
data space



latent space

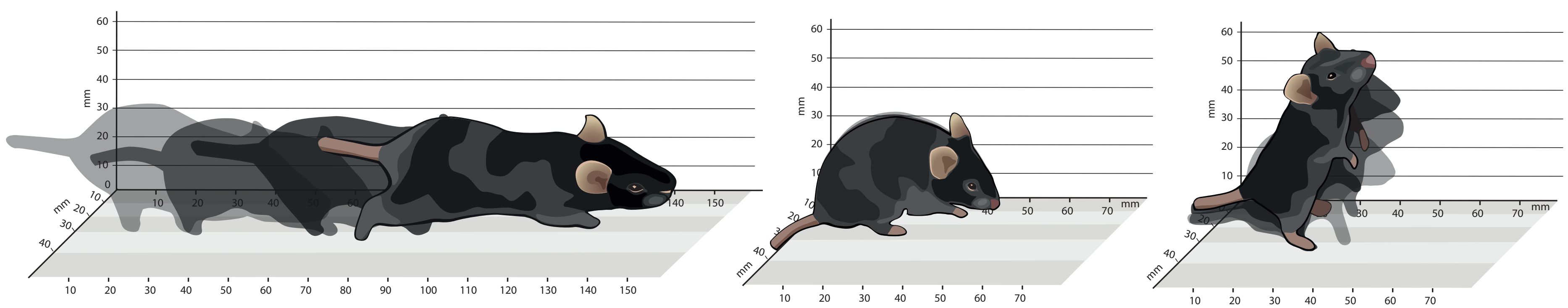


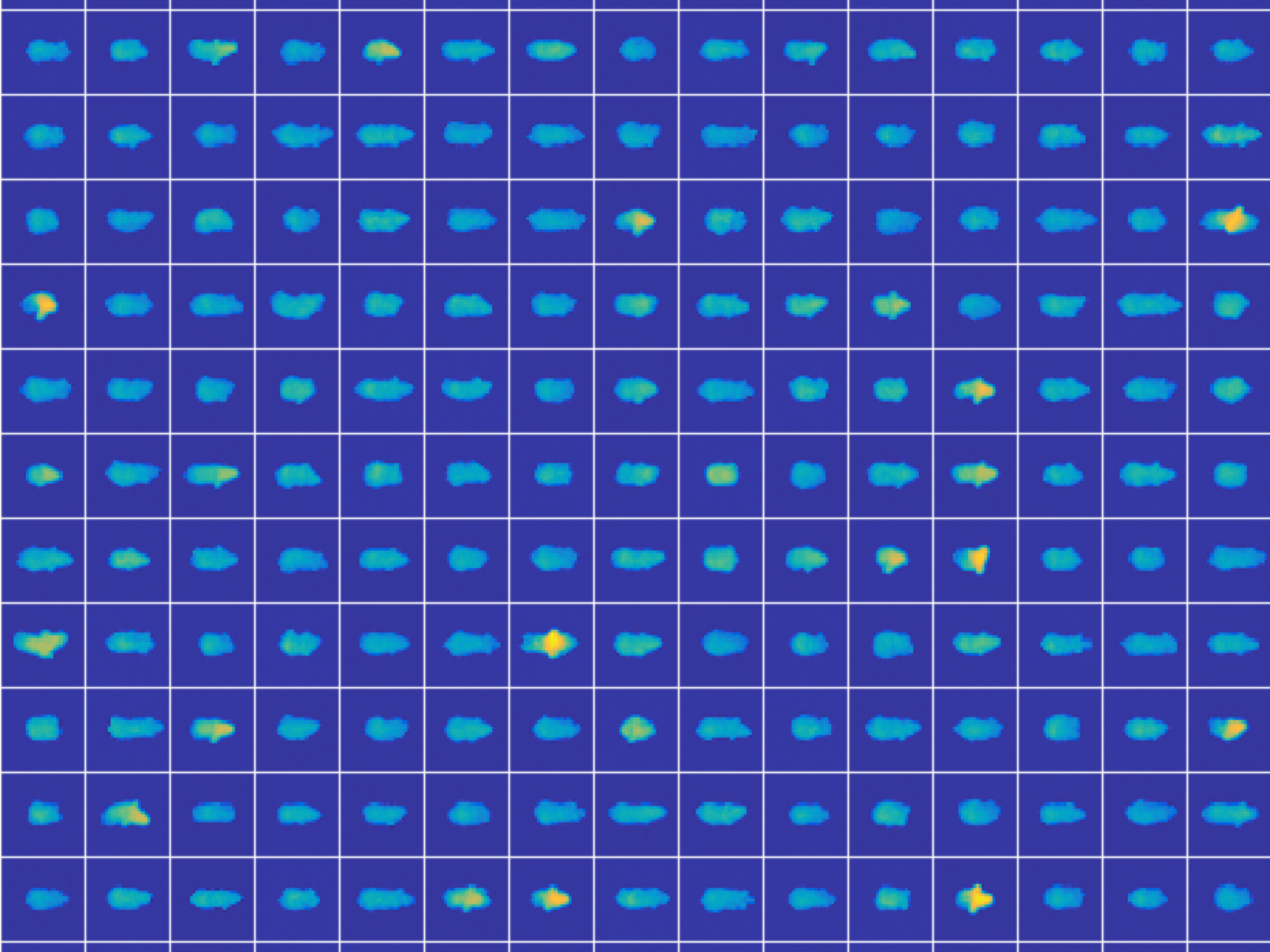
data
predictions
latent states
0
20
40
60
80
frame index

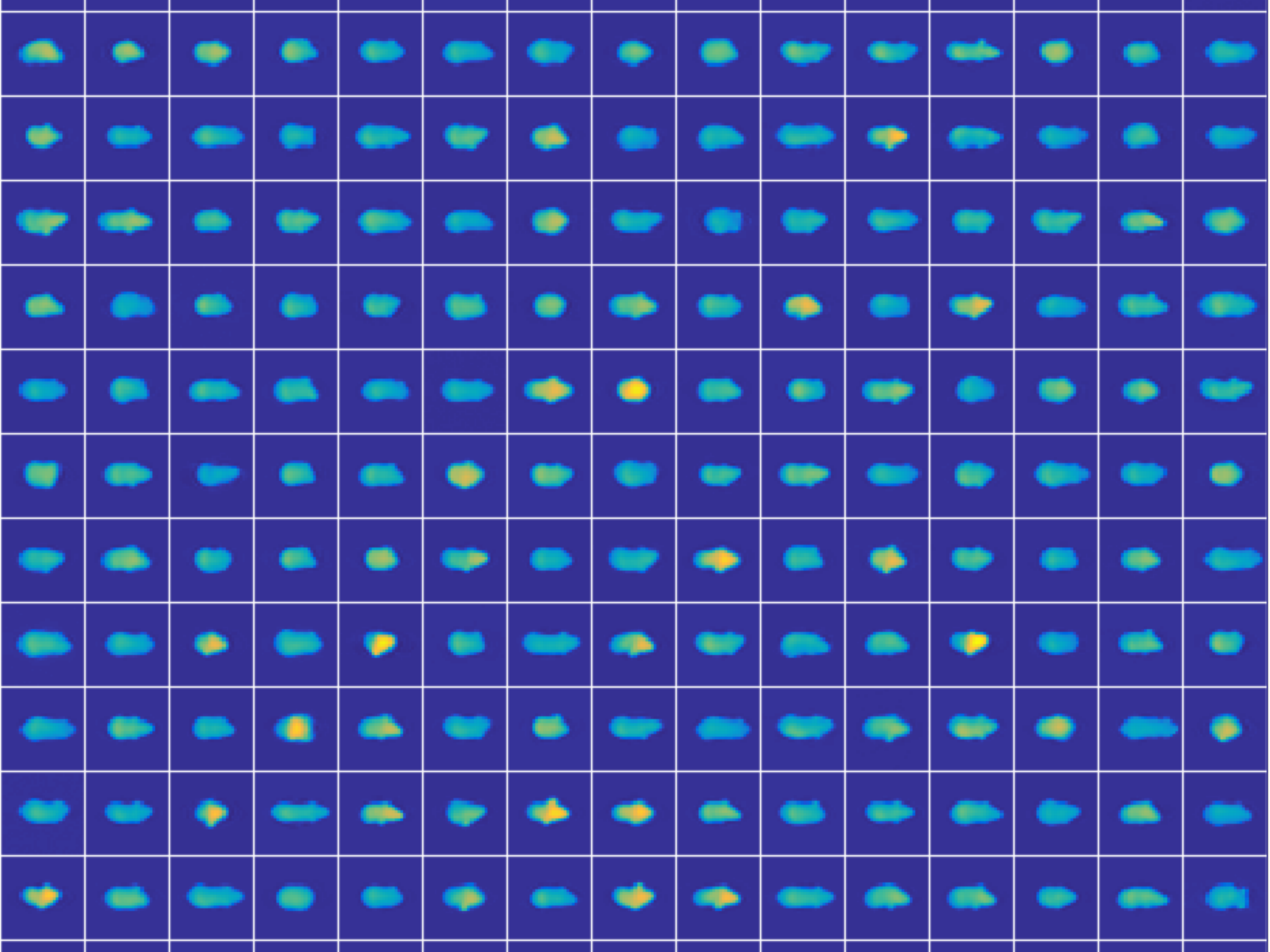


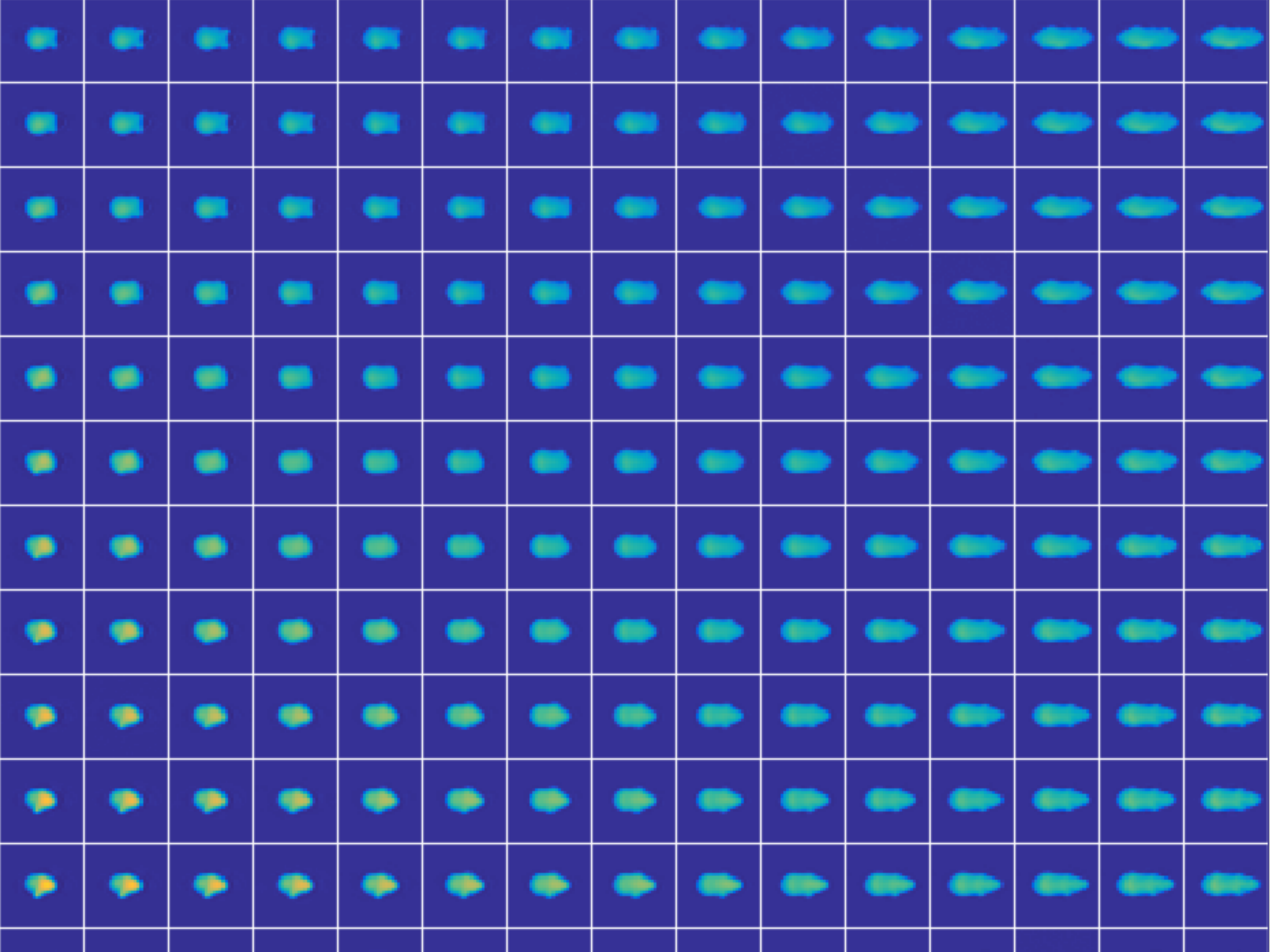
10
5
0
−5
−10
−15
−ℒ
0
1000
2000
3000
4000
iteration
flat
gradient
natural
gradient

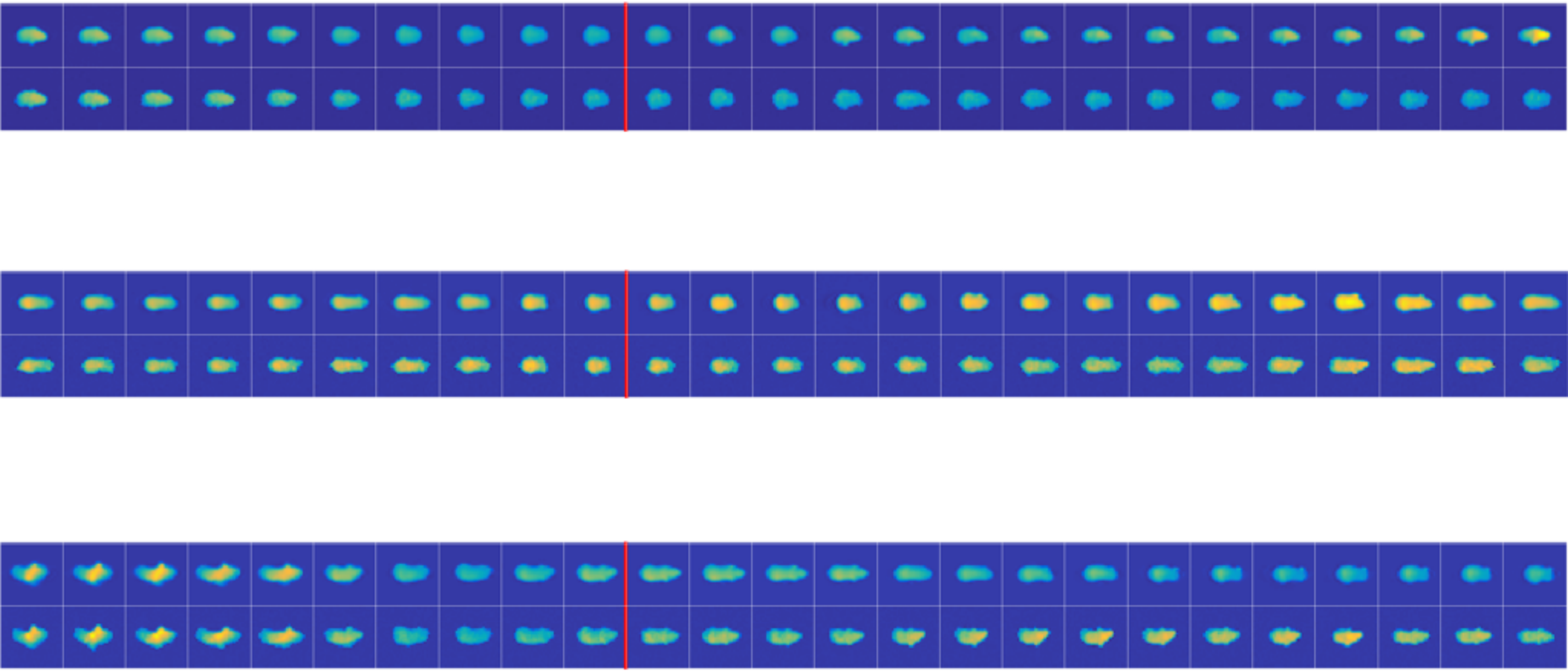
Application: learn syllable representation of behavior from video

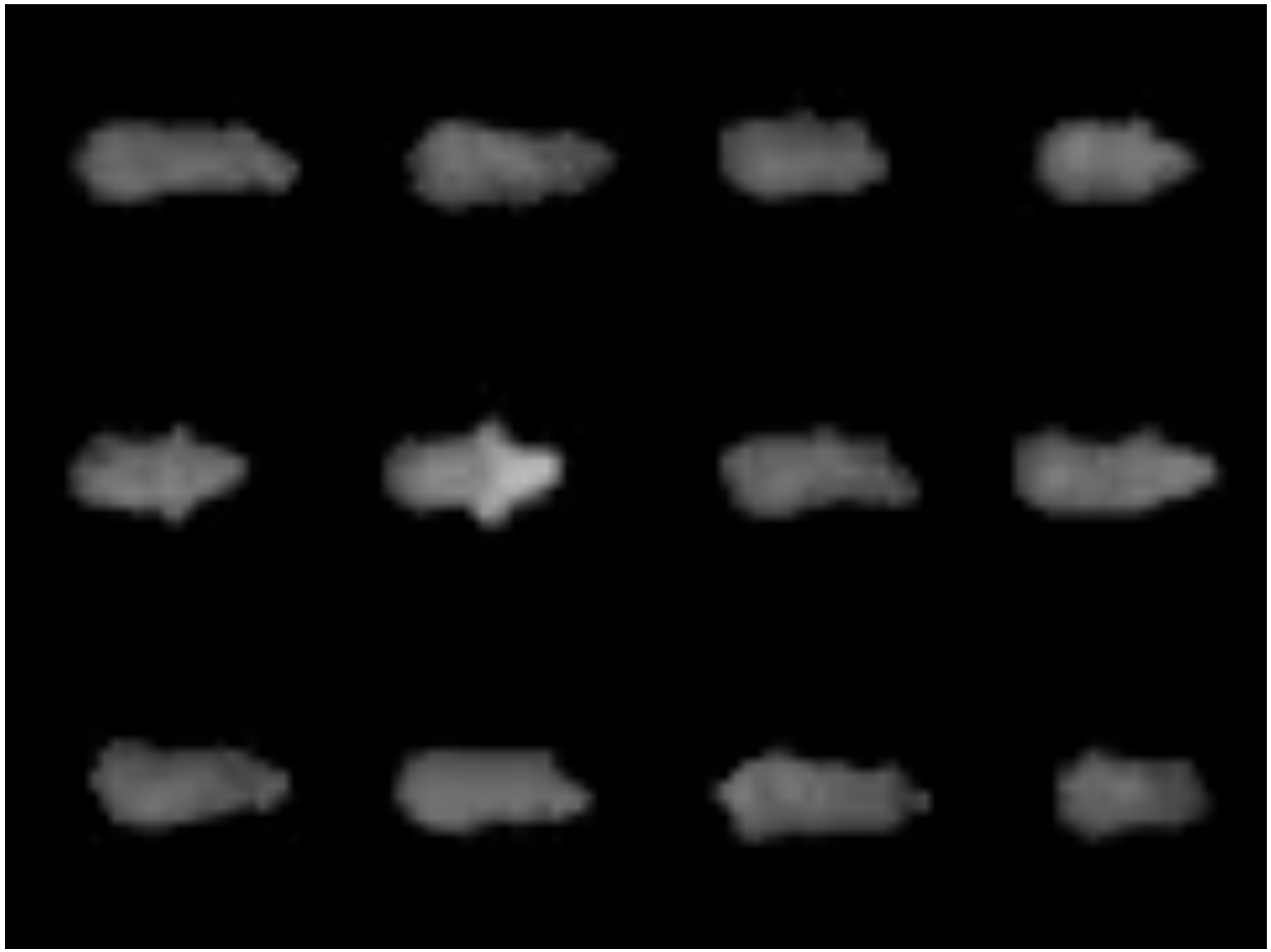




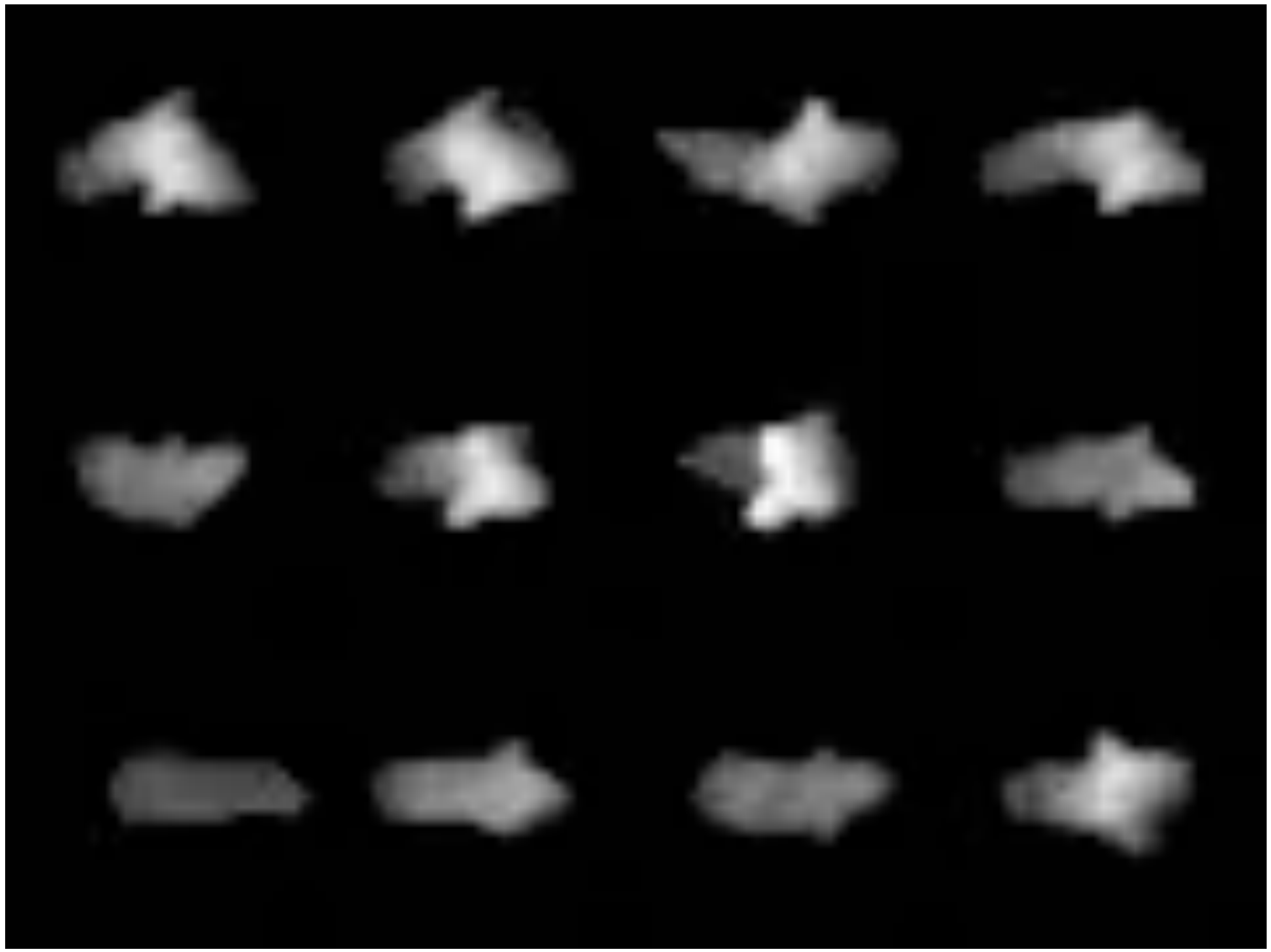




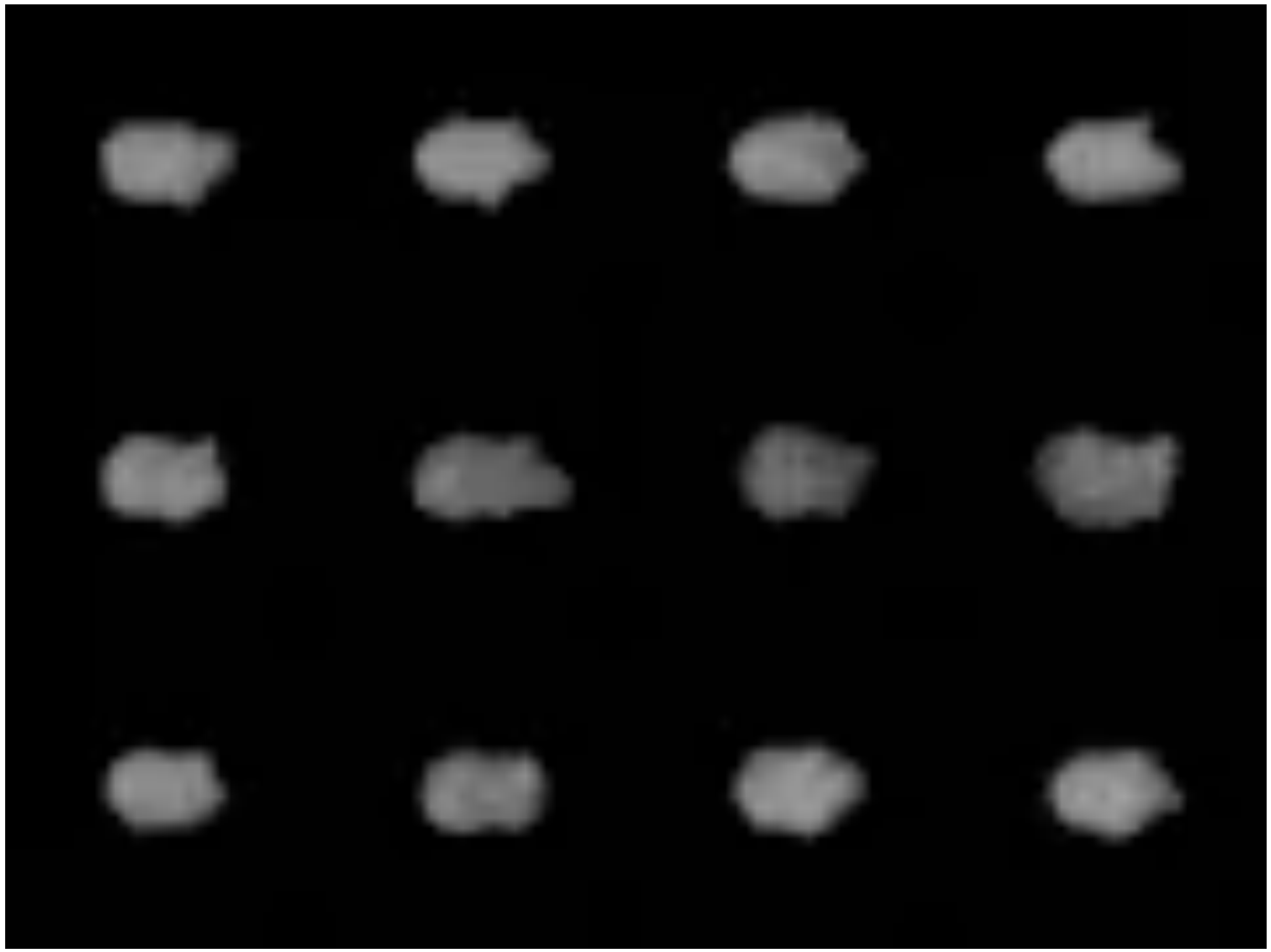




start rear

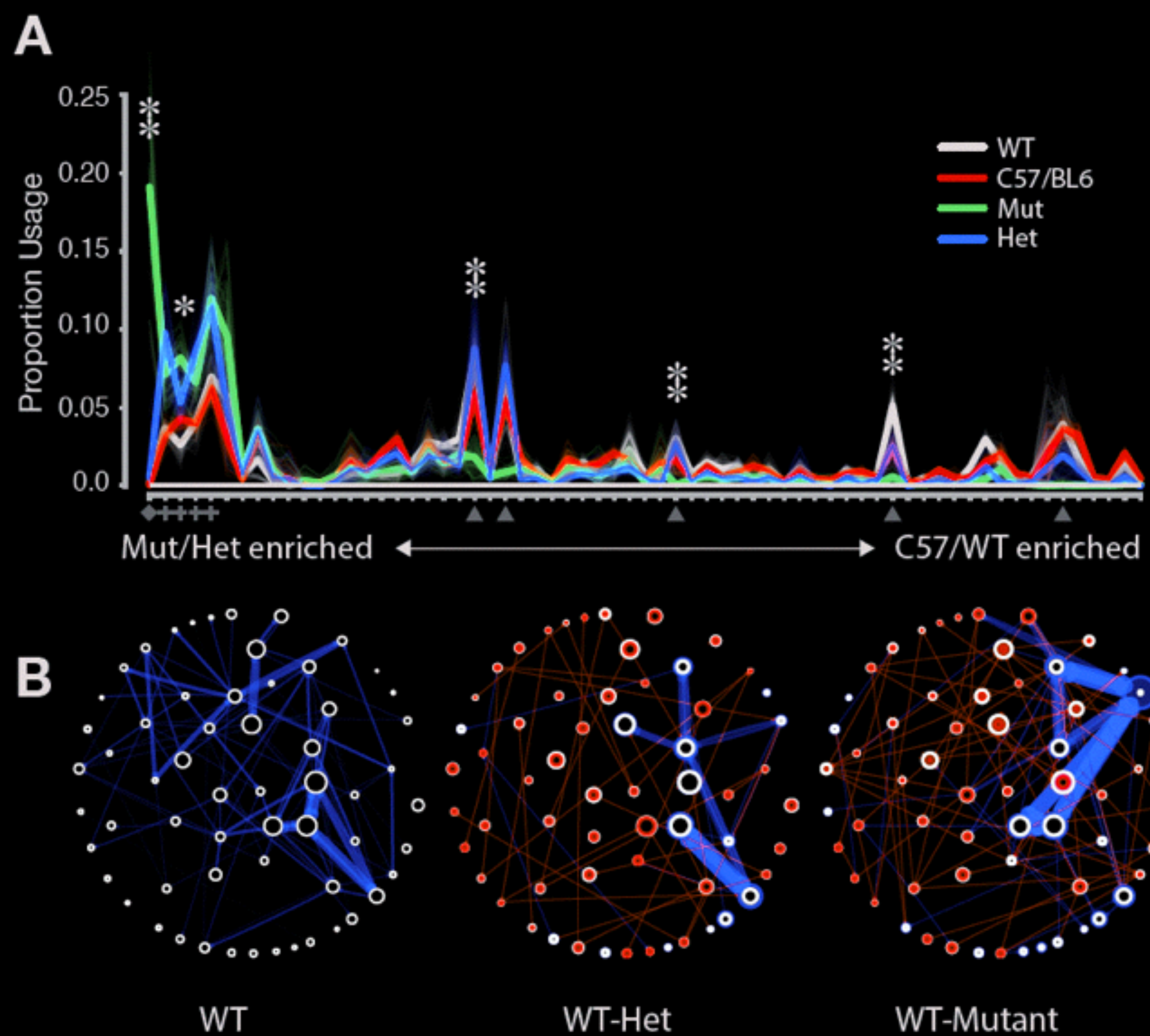


fall from rear



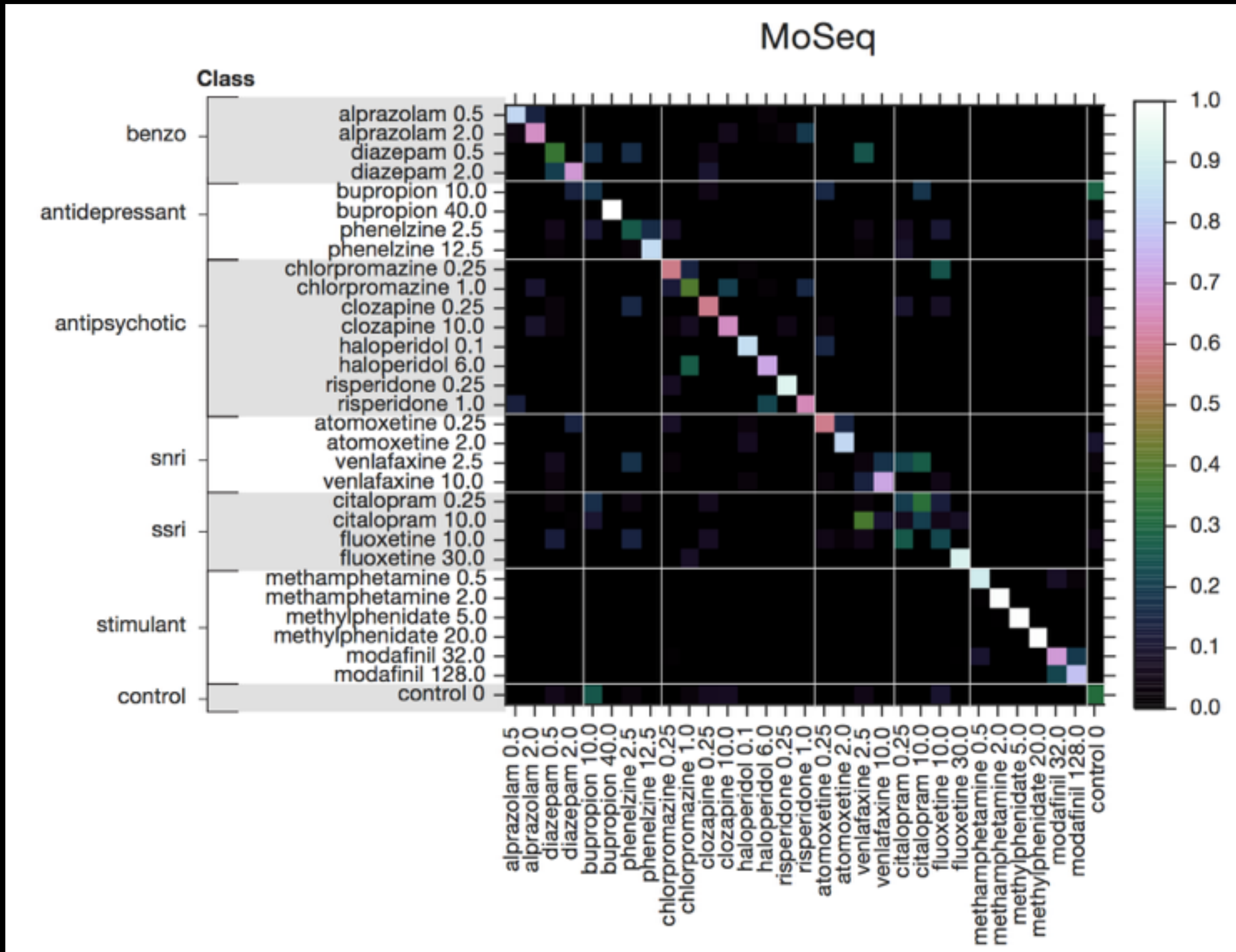
grooming

Discovery of Heterozygous Phenotypes in Ror1b Mice



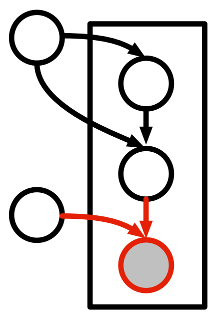
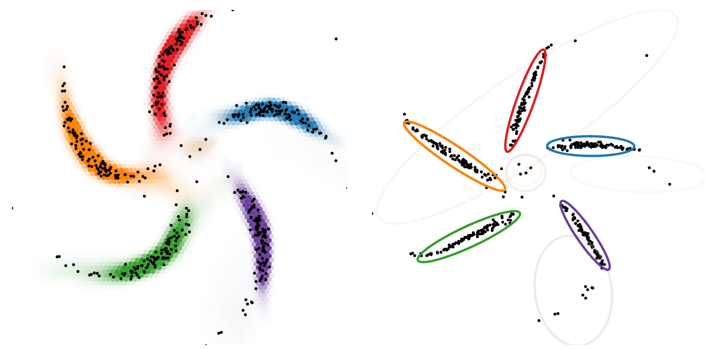
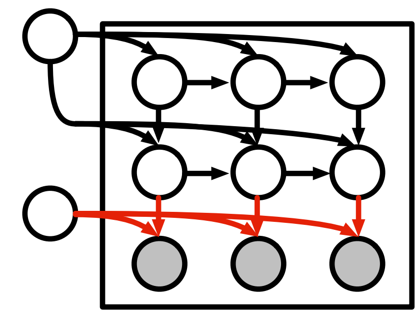
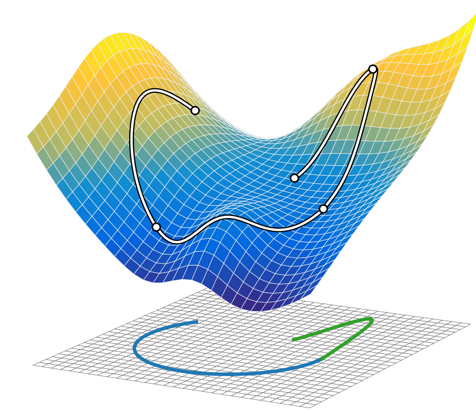
Alexander Wiltschko, **Matthew Johnson**, et al., Neuron 2015.

… and high and low doses of each drug

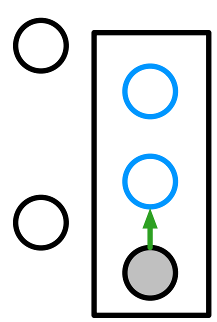
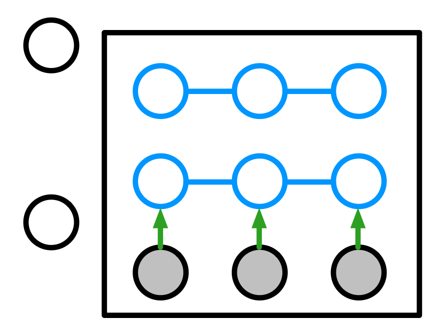


from Alex Wiltschko preprint

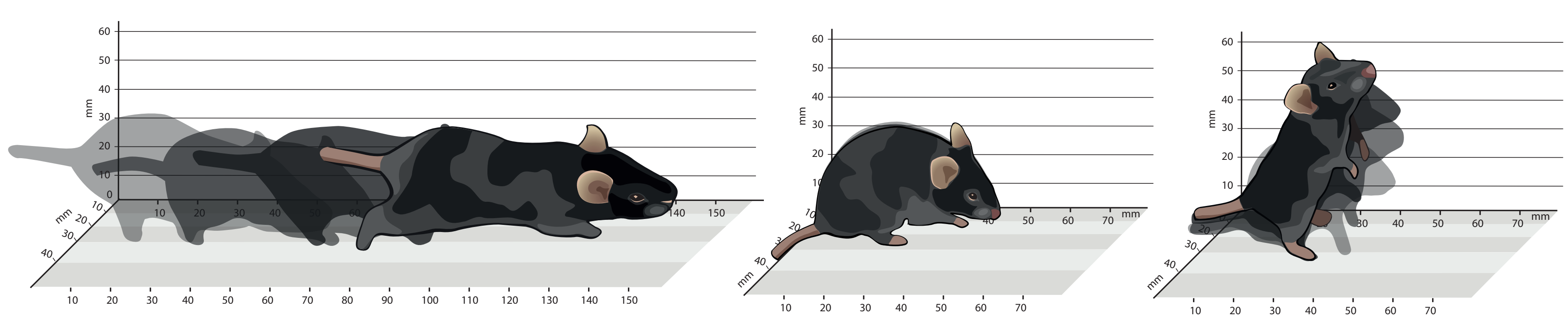
Modeling idea: graphical models on latent variables,
neural network models for observations

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

Application: learn syllable representation of behavior from video



Limitations and future work

capacity

- How expressive is latent linear structure?
  - word embeddings [1], analogical reasoning in image models
  - SVAE can use nonlinear latent structure

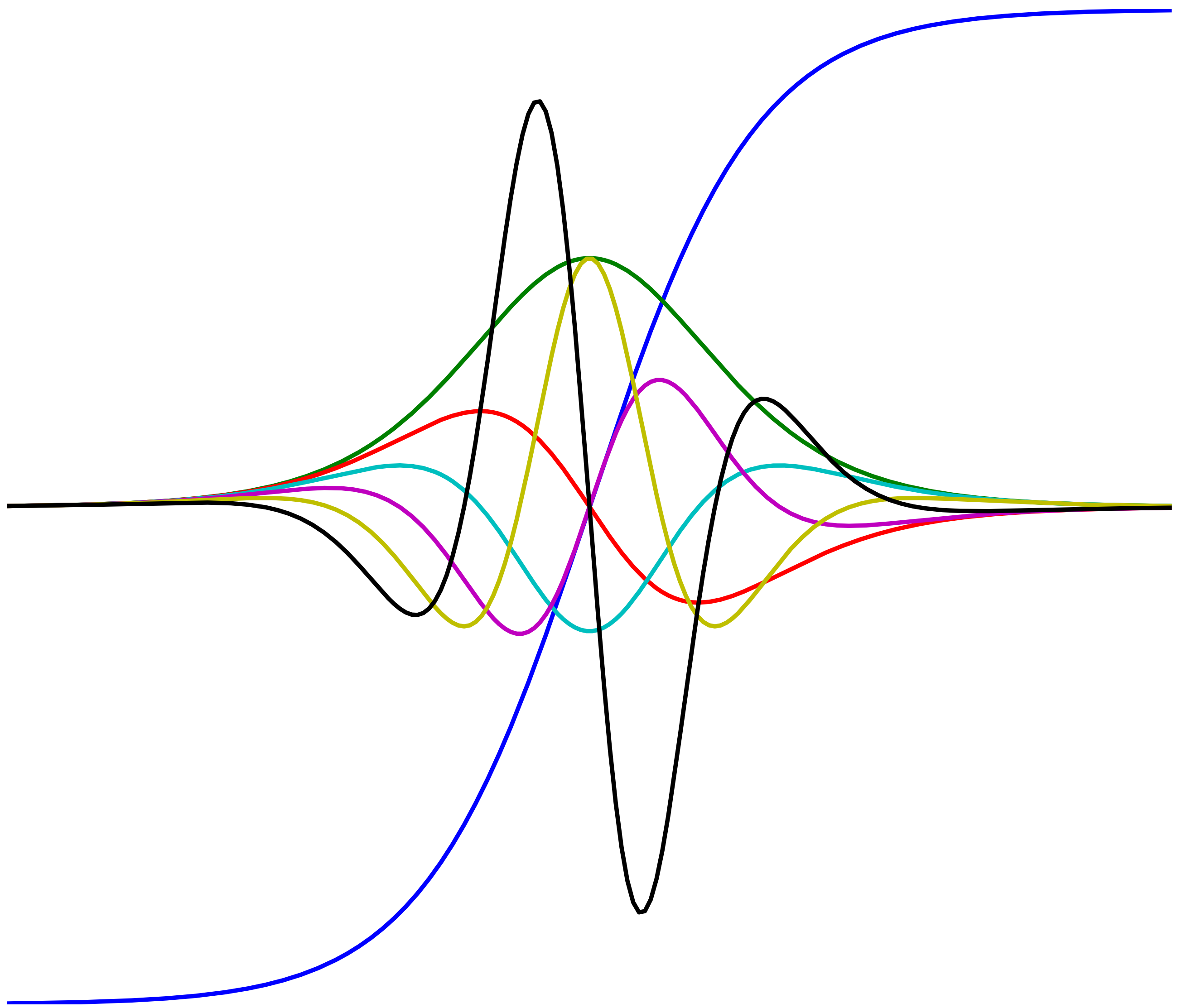
complexity

- PGMs get complicated
  - SVAE keeps complexity modular

future work

- model-based reinforcement learning
- automatic structure search [2,3]
- semi-supervised applications

[1] Hashimoto, Alvarez-Melis, and Jaakkola, Word, graph and manifold embedding from Markov processes, Preprint 2015.
[2] Grosse et al., Exploiting compositionality to explore a large space of model structures, UAI 2012.
[3] Duvenaud et al., Structure discovery in nonparametric regression through compositional kernel search, ICML 2013.



github.com/hips/autograd

Thanks!