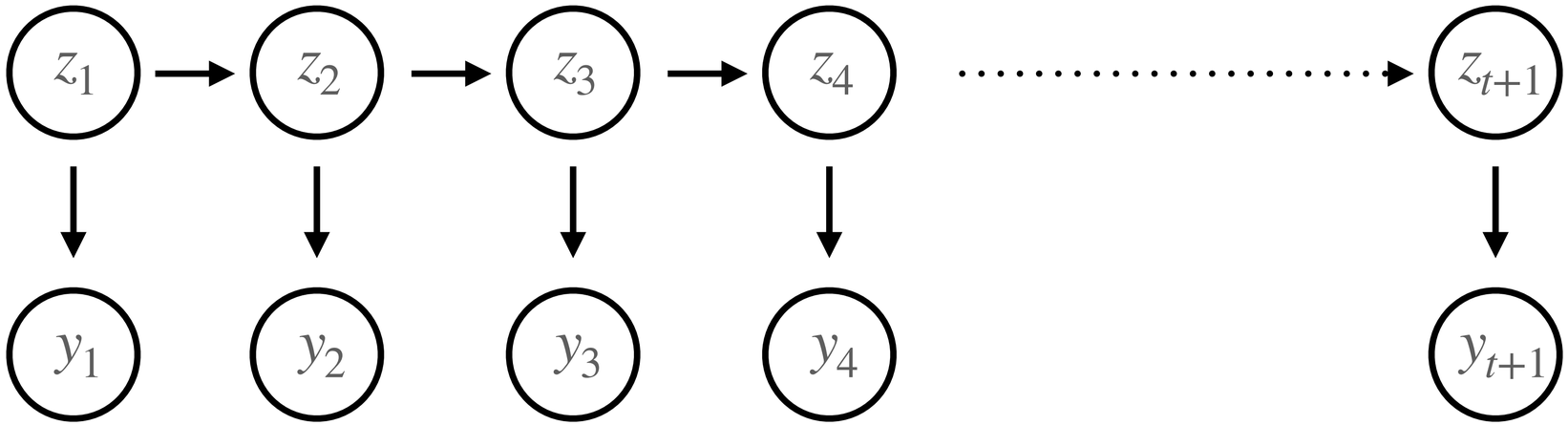
LDS + Extensions, VAEs, SCA, Bi-cross-validation

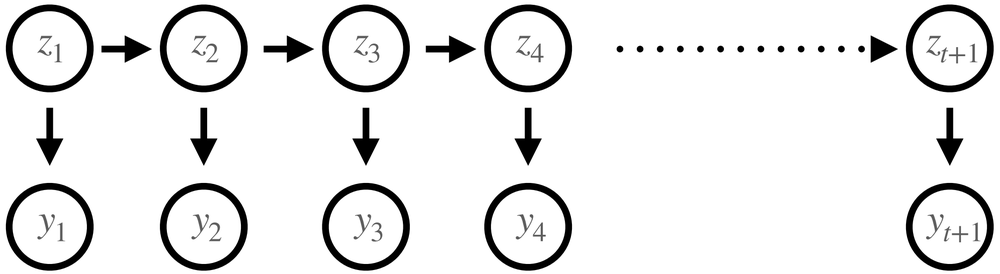
Linear Dynamical Systems



$$P(z_{1:T}, y_{1:T}) = P(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \prod_{t=1}^{T} p(y_t \mid z_t)$$

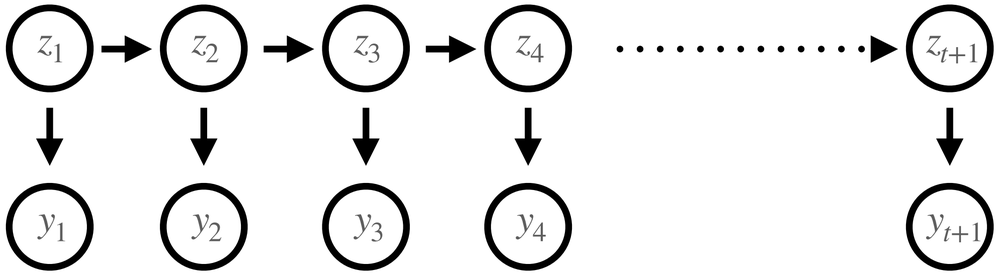
- Same graphical model as an HMM, but now z is continuous!
- Note the change in notation from *x* -> *y*

Linear Dynamical Systems



$$P(z_{1:T}, y_{1:T}) = P(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \prod_{t=1}^{T} p(y_t \mid z_t)$$

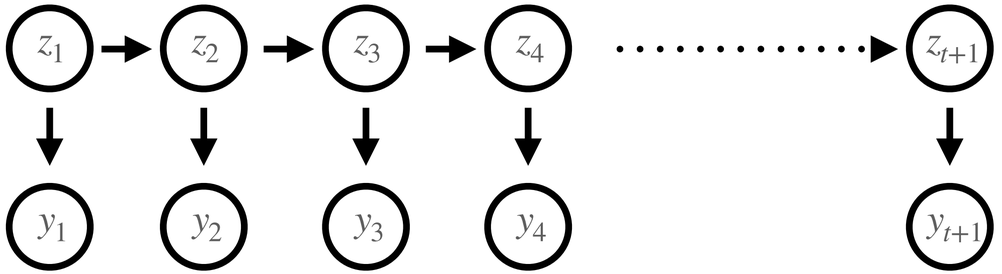
Linear Dynamical Systems



$$P(z_{1:T}, y_{1:T}) = P(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \prod_{t=1}^{T} p(y_t \mid z_t)$$

$$\mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \mathbf{I}) \qquad \mathcal{N}(\boldsymbol{z}_t \mid \mathbf{F}_t \boldsymbol{z}_{t-1}, \mathbf{Q}_t) \qquad \mathcal{N}(\boldsymbol{y}_t \mid \mathbf{H}_t \boldsymbol{z}_t, \mathbf{R}_t)$$

Linear Dynamical Systems



$$P(z_{1:T}, y_{1:T}) = P(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \prod_{t=1}^{T} p(y_t \mid z_t)$$

$$\mathcal{N}(\boldsymbol{z} | \mathbf{0}, \mathbf{I})$$

$$\mathcal{N}(\boldsymbol{z}_t | \mathbf{F}_t \boldsymbol{z}_{t-1}, \mathbf{Q}_t)$$

Or more fully:

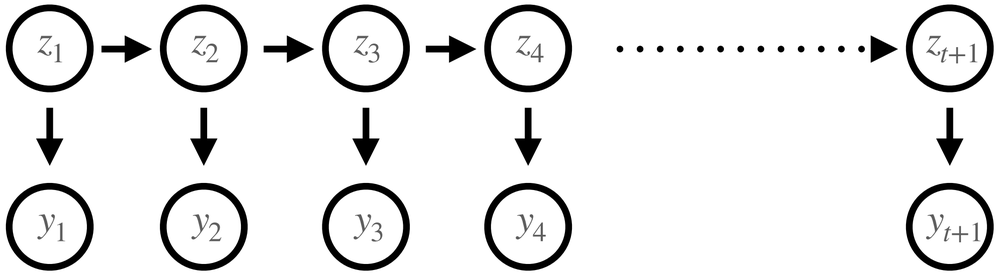
$$\mathcal{N}(\boldsymbol{z}_t | \mathbf{F}_t \boldsymbol{z}_{t-1} + \mathbf{B}_t \boldsymbol{u}_t + \boldsymbol{b}_t, \mathbf{Q}_t)$$

$$\mathcal{N}(\boldsymbol{y}_t | \mathbf{H}_t \boldsymbol{z}_t, \mathbf{R}_t)$$

Or more fully:

$$\mathcal{N}(\boldsymbol{y}_t | \mathbf{H}_t \boldsymbol{z}_t + \mathbf{D}_t \boldsymbol{u}_t + \boldsymbol{d}_t, \mathbf{R}_t)$$

Linear Dynamical Systems



$$P(z_{1:T}, y_{1:T}) = P(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \prod_{t=1}^{T} p(y_t \mid z_t)$$

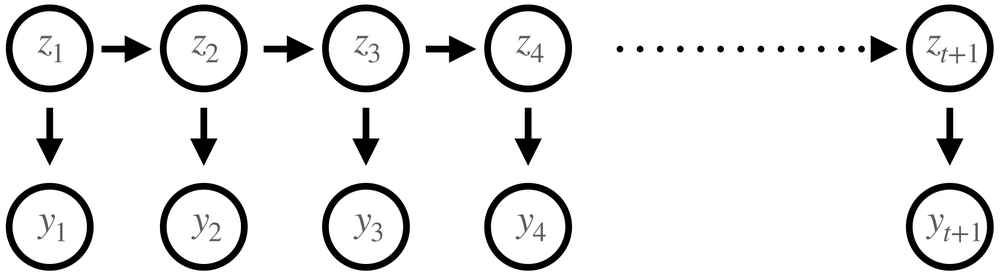
$$\mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \mathbf{I})$$

$$\mathcal{N}(\boldsymbol{z}_t \mid \mathbf{F}_t \boldsymbol{z}_{t-1}, \mathbf{Q}_t)$$

$$\mathcal{N}(\boldsymbol{y}_t \mid \mathbf{H}_t \boldsymbol{z}_t, \mathbf{R}_t)$$

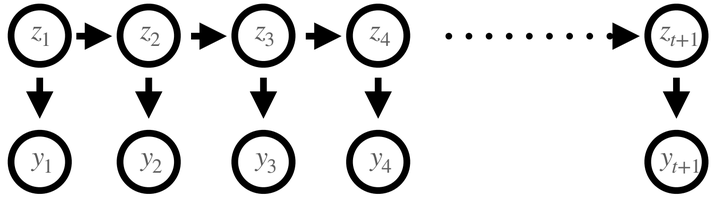
Just like the Factor Analysis Model, but now with a dynamics term!

Simulating from an ~~HMM~~ LDS

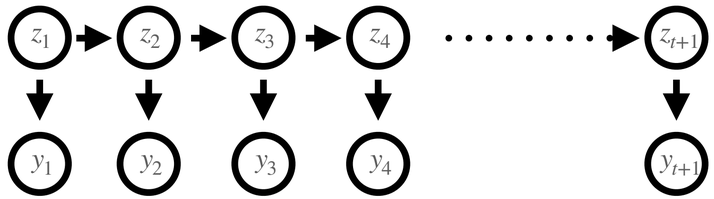


- Sample from $P(z_1)$
- Sample from $P(y_1 \mid z_1)$
- For all future time steps:
  - Sample $P(z_{t+1} \mid z_t)$
  - Sample $P(y_{t+1} \mid z_{t+1})$

LDS: Inferring the Latent States with a Kalman Filter



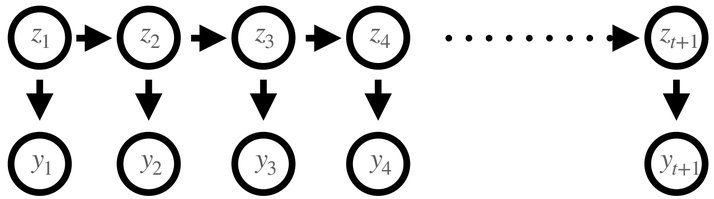
LDS: Inferring the Latent States with a Kalman Filter



- Predict Step:
  - Predict forward in time with dynamics model

- Update Step:
  - Use observations to modify dynamics-only prediction from above

LDS: Inferring the Latent States with a Kalman Filter



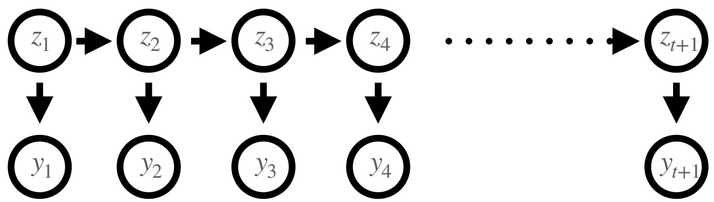
$$\mathcal{N}(\boldsymbol{z}_t|\mathbf{F}_t\boldsymbol{z}_{t-1}, \mathbf{Q}_t)$$

$$\mathcal{N}(\boldsymbol{y}_t|\mathbf{H}_t\boldsymbol{z}_t, \mathbf{R}_t)$$

- Predict Step:
  - Predict forward in time with dynamics model

- Update Step:
  - Use observations to modify dynamics-only prediction from above

LDS: Inferring the Latent States with a Kalman Filter



$$\mathcal{N}(\boldsymbol{z}_t|\mathbf{F}_t\boldsymbol{z}_{t-1}, \mathbf{Q}_t)$$

$$\mathcal{N}(\boldsymbol{y}_t|\mathbf{H}_t\boldsymbol{z}_t, \mathbf{R}_t)$$

- Predict Step:
  - Predict forward in time with dynamics model

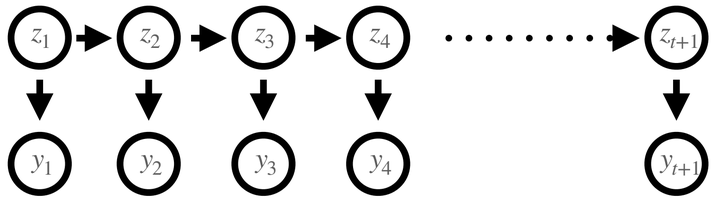
$$p(\boldsymbol{z}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{u}_{1:t}) = \mathcal{N}(\boldsymbol{z}_t|\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$$

$$\boldsymbol{\mu}_{t|t-1} = \mathbf{F}_t\boldsymbol{\mu}_{t-1|t-1} + \mathbf{B}_t\boldsymbol{u}_t + \boldsymbol{b}_t$$

$$\boldsymbol{\Sigma}_{t|t-1} = \mathbf{F}_t\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{F}_t^{\mathsf{T}} + \mathbf{Q}_t$$

- Update Step:
  - Use observations to modify dynamics-only prediction from above

LDS: Inferring the Latent States with a Kalman Filter



$$\mathcal{N}(\boldsymbol{z}_t|\mathbf{F}_t\boldsymbol{z}_{t-1}, \mathbf{Q}_t)$$

$$\mathcal{N}(\boldsymbol{y}_t|\mathbf{H}_t\boldsymbol{z}_t, \mathbf{R}_t)$$

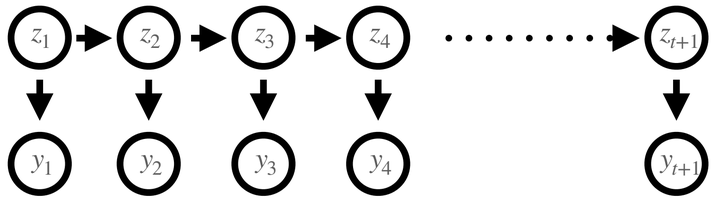
- Predict Step:
  - Predict forward in time with dynamics model

$$p(\boldsymbol{z}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{u}_{1:t}) = \mathcal{N}(\boldsymbol{z}_t|\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$$
$$\boldsymbol{\mu}_{t|t-1} = \mathbf{F}_t\boldsymbol{\mu}_{t-1|t-1} + \mathbf{B}_t\boldsymbol{u}_t + \boldsymbol{b}_t$$
$$\boldsymbol{\Sigma}_{t|t-1} = \mathbf{F}_t\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{F}_t^\mathsf{T} + \mathbf{Q}_t$$

- Update Step:
  - Use observations to modify dynamics-only prediction from above

$$p(\boldsymbol{z}_t|\boldsymbol{y}_{1:t}, \boldsymbol{u}_{1:t}) = \mathcal{N}(\boldsymbol{z}_t|\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t})$$
$$\hat{\boldsymbol{y}}_t = \mathbf{H}_t\boldsymbol{\mu}_{t|t-1} + \mathbf{D}_t\boldsymbol{u}_t + \boldsymbol{d}_t$$
$$\mathbf{S}_t = \mathbf{H}_t\boldsymbol{\Sigma}_{t|t-1}\mathbf{H}_t^\mathsf{T} + \mathbf{R}_t$$
$$\mathbf{K}_t = \boldsymbol{\Sigma}_{t|t-1}\mathbf{H}_t^\mathsf{T}\mathbf{S}_t^{-1}$$
$$\boldsymbol{\mu}_{t|t} = \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)$$

LDS: Inferring the Latent States with a Kalman Filter



$$\mathcal{N}(\boldsymbol{z}_t|\mathbf{F}_t\boldsymbol{z}_{t-1}, \mathbf{Q}_t)$$

$$\mathcal{N}(\boldsymbol{y}_t|\mathbf{H}_t\boldsymbol{z}_t, \mathbf{R}_t)$$

- Predict Step:
  - Predict forward in time with dynamics model

$$p(\boldsymbol{z}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{u}_{1:t}) = \mathcal{N}(\boldsymbol{z}_t|\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$$
$$\boldsymbol{\mu}_{t|t-1} = \mathbf{F}_t\boldsymbol{\mu}_{t-1|t-1} + \mathbf{B}_t\boldsymbol{u}_t + \boldsymbol{b}_t$$
$$\boldsymbol{\Sigma}_{t|t-1} = \mathbf{F}_t\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{F}_t^\mathsf{T} + \mathbf{Q}_t$$

- Update Step:
  - Use observations to modify dynamics-only prediction from above

$$p(\boldsymbol{z}_t|\boldsymbol{y}_{1:t}, \boldsymbol{u}_{1:t}) = \mathcal{N}(\boldsymbol{z}_t|\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t})$$
$$\hat{\boldsymbol{y}}_t = \mathbf{H}_t\boldsymbol{\mu}_{t|t-1} + \mathbf{D}_t\boldsymbol{u}_t + \boldsymbol{d}_t$$
$$\mathbf{S}_t = \mathbf{H}_t\boldsymbol{\Sigma}_{t|t-1}\mathbf{H}_t^\mathsf{T} + \mathbf{R}_t$$
$$\mathbf{K}_t = \boldsymbol{\Sigma}_{t|t-1}\mathbf{H}_t^\mathsf{T}\mathbf{S}_t^{-1}$$
$$\boldsymbol{\mu}_{t|t} = \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t(\boldsymbol{y}_t - \hat{\boldsymbol{y}}_t)$$

Dynamics Noise

Observations Noise

LDS: Inferring the Latent States with a Kalman Smoother

- The Kalman Filter finds $p(z_t \mid y_{1:t}, u_{1:t})$
  - Comparable to the “forward” algorithm for HMMs

LDS: Inferring the Latent States with a Kalman Smoother

- The Kalman Filter finds $p(z_t \mid y_{1:t}, u_{1:t})$
  - Comparable to the “forward” algorithm for HMMs

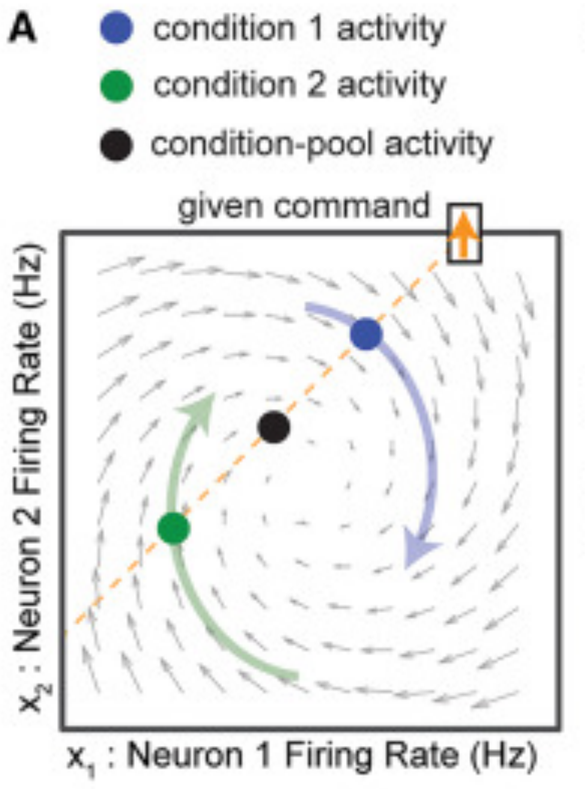
- The Kalman Smoother finds $p(z_t \mid y_{1:T}, u_{1:T})$
  - Also includes a calculation going backwards in time, and is comparable to the “forward-backward” algorithm for HMMs

LDS: Model Fitting

- Most standard is EM
  - E-step is Kalman Smoother
  - M-step optimizes parameters
    - Can learn interesting things about the dynamics too!

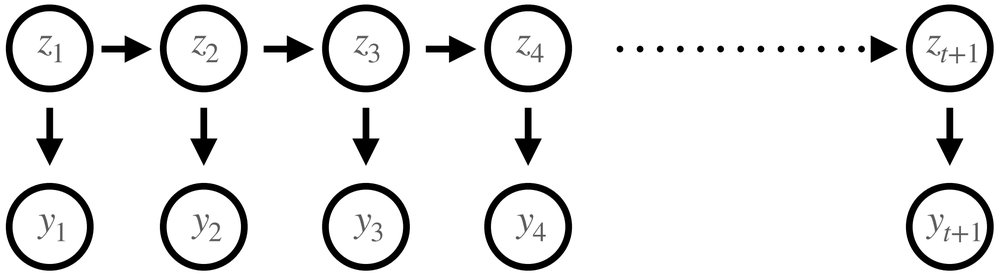
- Subspace Identification (SSID) is also used

Invariant neural dynamics drive commands to control different movements



Extensions

Poisson Linear Dynamical Systems



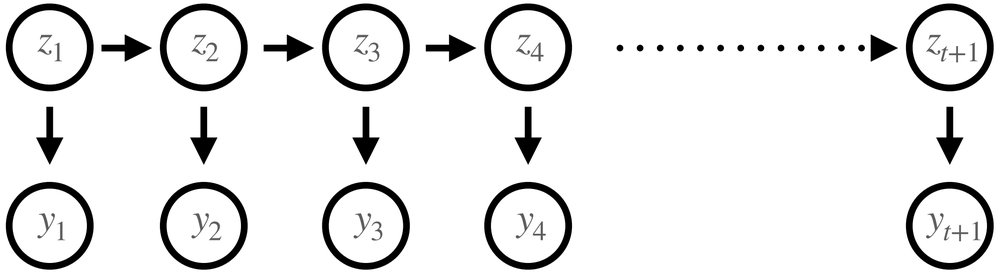
$$P(z_{1:T}, y_{1:T}) = P(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \prod_{t=1}^{T} p(y_t \mid z_t)$$

$$\mathcal{N}(\boldsymbol{z} \mid \mathbf{0}, \mathbf{I}) \qquad \mathcal{N}(\boldsymbol{z}_t \mid \mathbf{F}_t \boldsymbol{z}_{t-1}, \mathbf{Q}_t)$$

Or more fully:

$$\mathcal{N}(\boldsymbol{z}_t \mid \mathbf{F}_t \boldsymbol{z}_{t-1} + \mathbf{B}_t \boldsymbol{u}_t + \boldsymbol{b}_t, \mathbf{Q}_t)$$

Poisson Linear Dynamical Systems



$$P(z_{1:T}, y_{1:T}) = P(z_1)\prod_{t=2}^{T} p(z_t \mid z_{t-1})\prod_{t=1}^{T} p(y_t \mid z_t)$$

$$\mathcal{N}(z|\mathbf{0}, \mathbf{I})$$

$$\mathcal{N}(\boldsymbol{z}_t|\mathbf{F}_t\boldsymbol{z}_{t-1}, \mathbf{Q}_t)$$

Or more fully:

$$\mathcal{N}(\boldsymbol{z}_t|\mathbf{F}_t\boldsymbol{z}_{t-1} + \mathbf{B}_t\boldsymbol{u}_t + \boldsymbol{b}_t, \mathbf{Q}_t)$$

f=exp() or softplus()

$$Poisson\left(y_t \mid f(\mathbf{H}_t z_t)\right)$$

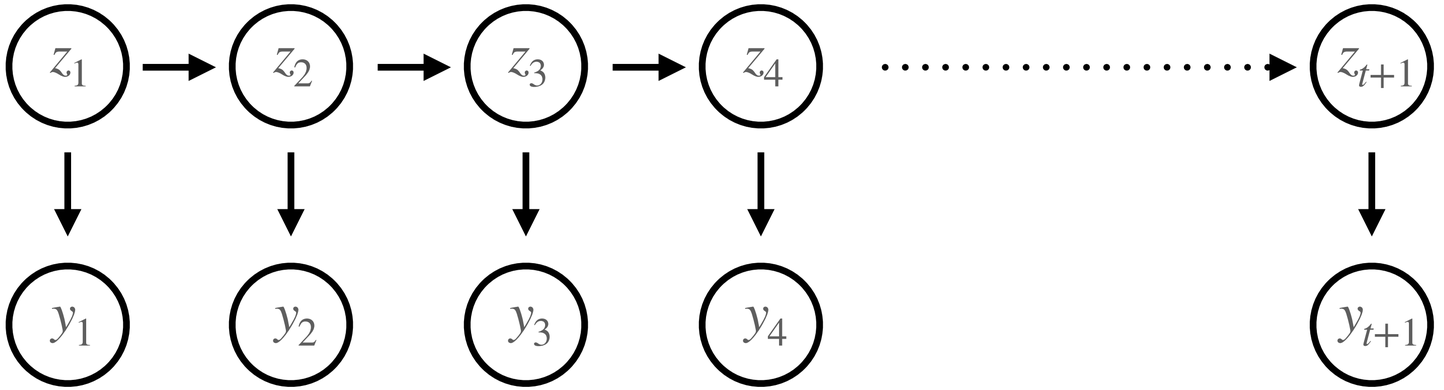
Or more fully:

$$Poisson\left(y_t \mid f(\mathbf{H}_t z_t + \mathbf{D}_t u_t + d_t)\right)$$

Switching Linear Dynamical Systems

Continuous Latents

Observations
(Neural Data)

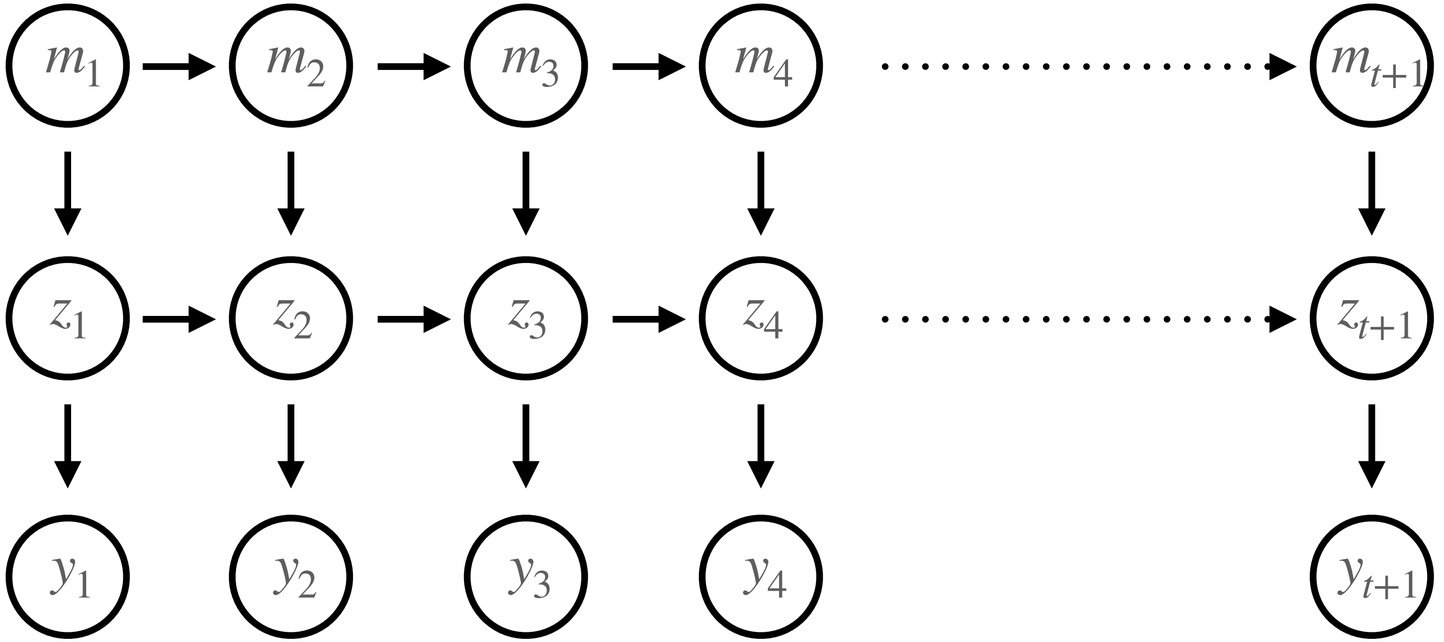


Switching Linear Dynamical Systems

Discrete Latents

Continuous Latents

Observations
(Neural Data)

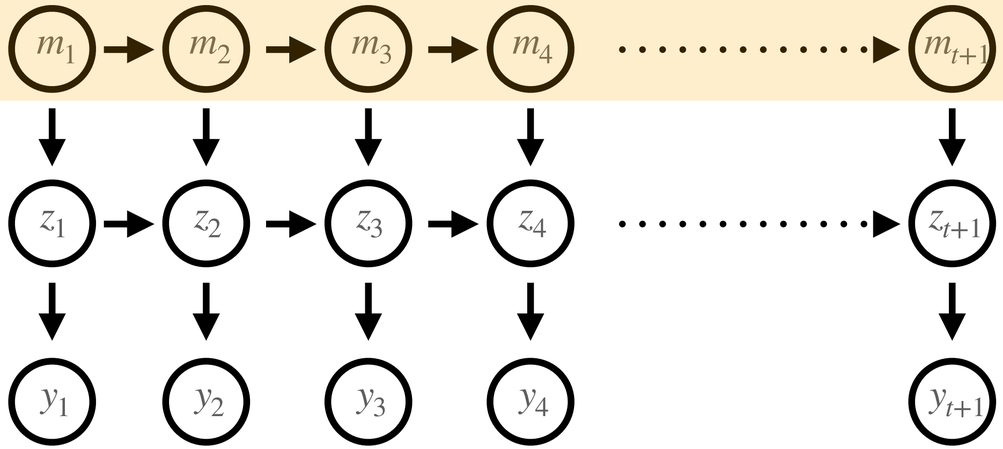


Switching Linear Dynamical Systems

Discrete Latents

Continuous Latents

Observations
(Neural Data)



Discrete Latents $p(m_t = k|m_{t-1} = j) = A_{jk}$

Continuous Latents $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, m_t = k, \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{z}_t|\mathbf{F}_k\boldsymbol{z}_{t-1} + \mathbf{B}_k\boldsymbol{u}_t + \boldsymbol{b}_k, \mathbf{Q}_k)$

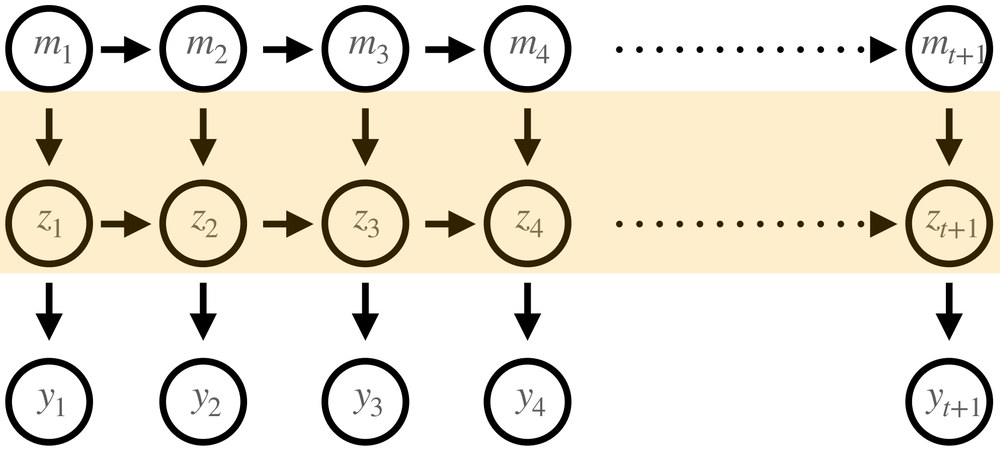
Observations
(Neural Data) $p(\boldsymbol{y}_t|\boldsymbol{z}_t, m_t = k, \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{y}_t|\mathbf{H}_k\boldsymbol{z}_t + \mathbf{D}_k\boldsymbol{u}_t + \boldsymbol{d}_k, \mathbf{R}_k)$

Switching Linear Dynamical Systems

Discrete Latents

Continuous Latents

Observations
(Neural Data)



Discrete Latents $$p(m_t = k | m_{t-1} = j) = A_{jk}$$

Continuous Latents $$p(\boldsymbol{z}_t | \boldsymbol{z}_{t-1}, m_t = k, \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{z}_t | \mathbf{F}_k \boldsymbol{z}_{t-1} + \mathbf{B}_k \boldsymbol{u}_t + \boldsymbol{b}_k, \mathbf{Q}_k)$$

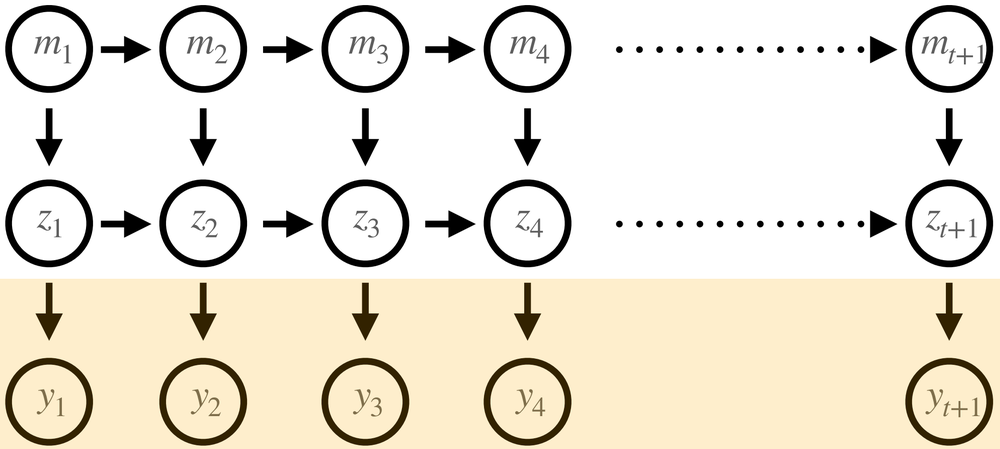
Observations
(Neural Data) $$p(\boldsymbol{y}_t | \boldsymbol{z}_t, m_t = k, \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{y}_t | \mathbf{H}_k \boldsymbol{z}_t + \mathbf{D}_k \boldsymbol{u}_t + \boldsymbol{d}_k, \mathbf{R}_k)$$

Switching Linear Dynamical Systems

Discrete Latents

Continuous Latents

Observations
(Neural Data)



Discrete Latents $p(m_t = k|m_{t-1} = j) = A_{jk}$

Continuous Latents $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, m_t = k, \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{z}_t|\mathbf{F}_k\boldsymbol{z}_{t-1} + \mathbf{B}_k\boldsymbol{u}_t + \boldsymbol{b}_k, \mathbf{Q}_k)$

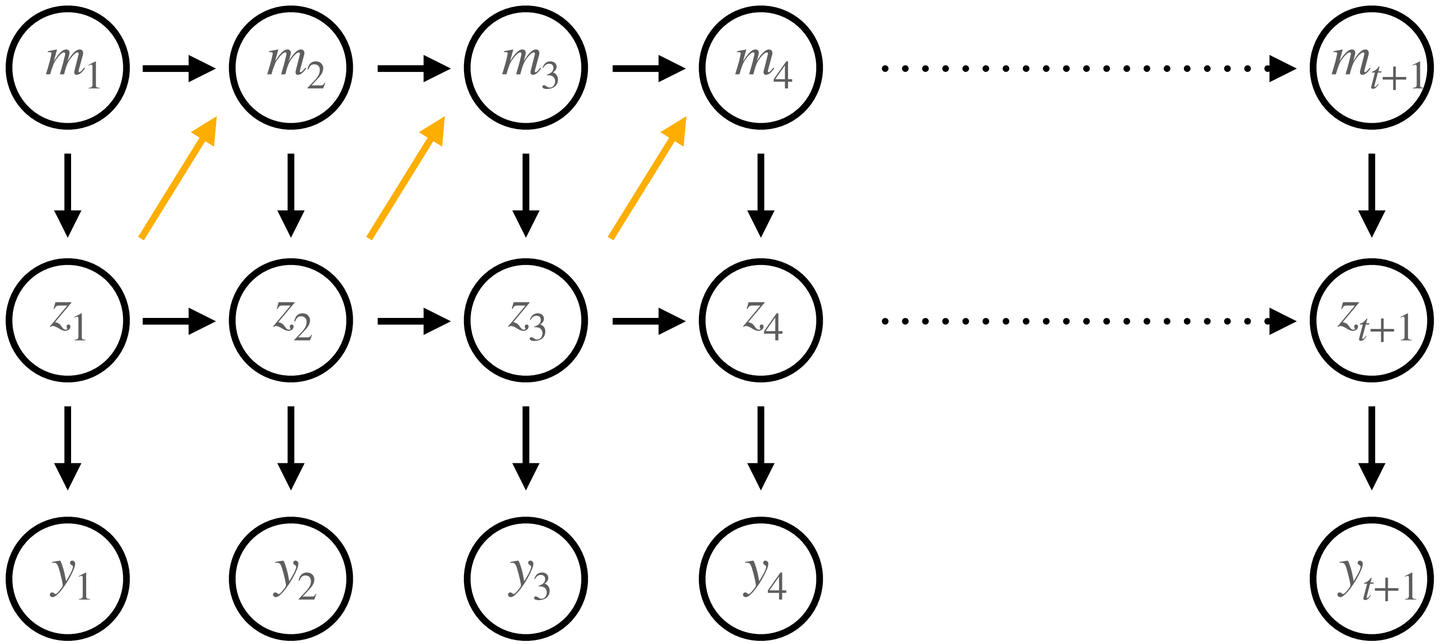
Observations (Neural Data) $p(\boldsymbol{y}_t|\boldsymbol{z}_t, m_t = k, \boldsymbol{u}_t) = \mathcal{N}(\boldsymbol{y}_t|\mathbf{H}_k\boldsymbol{z}_t + \mathbf{D}_k\boldsymbol{u}_t + \boldsymbol{d}_k, \mathbf{R}_k)$

Recurrent Switching Linear Dynamical Systems

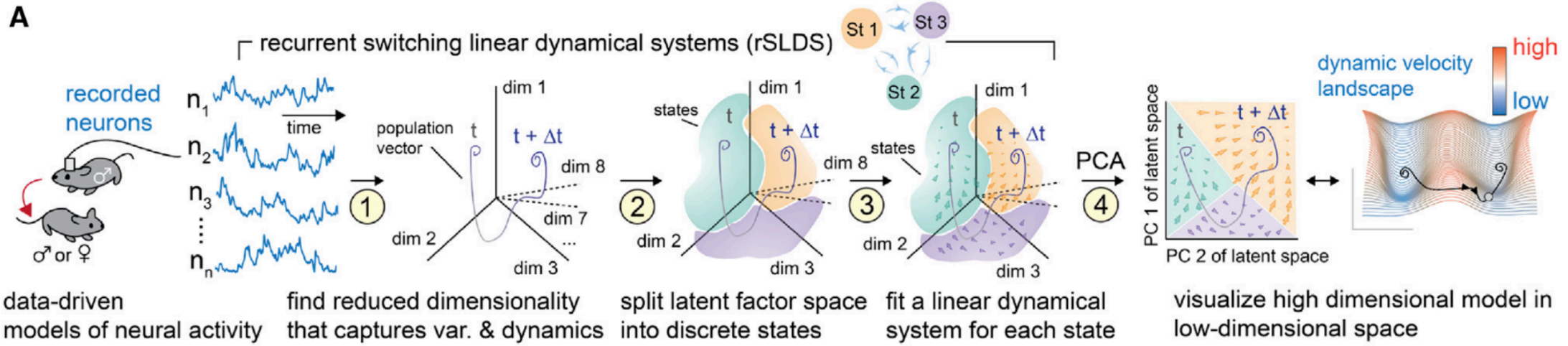
Discrete Latents

Continuous Latents

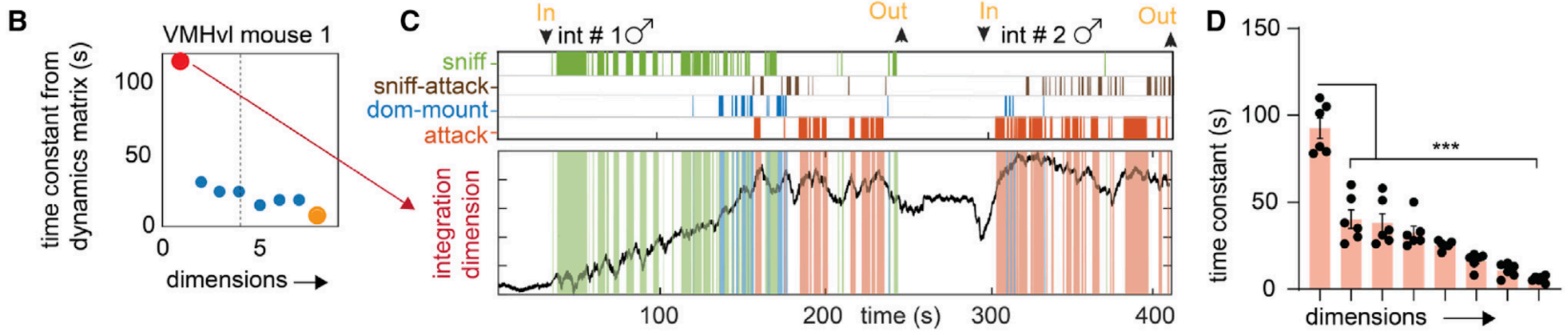
Observations
(Neural Data)



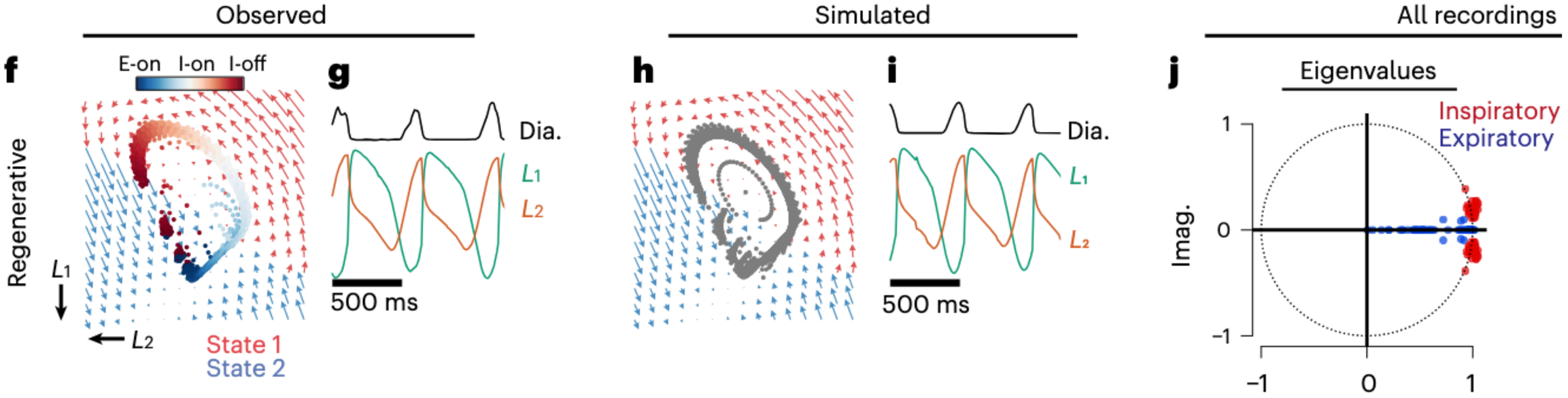
An approximate line attractor in the hypothalamus encodes an aggressive state



VMHvl activity exhibits an integration dimension with ramping activity and persistence during interactions with males



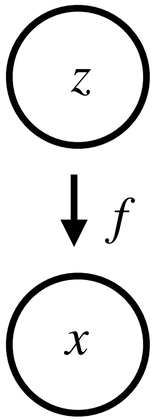
Latent neural population dynamics underlying breathing, opioid-induced respiratory depression and gasping



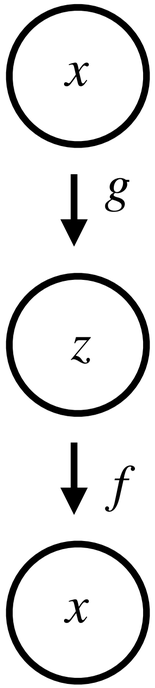
Variational Autoencoder (VAE)

Variational Autoencoder (VAE)

- Classically in latent variable models, we directly optimize ***z***
  - E.g., let's say ***z*** is a 1d Gaussian and ***x*** has *T* time points, then we would optimize a mean and stdev. for all *T* time points.



Variational Autoencoder (VAE)



- Classically in latent variable models, we directly optimize ***z***
  - E.g., let's say ***z*** is a 1d Gaussian and ***x*** has *T* time points, then we would optimize a mean and stdev. for all *T* time points.

- VAEs use "amortized inference", where ***z*** is learned as a function of ***x***, so only the parameters of *g* (neural network parameters) are learned
  - In the above example, there would be separate neural networks trained to predict ***z***'s means and stdevs: $\mu_t = g_\mu(x_t)$ and $\sigma_t = g_\sigma(x_t)$

VAEs vs. Standard Autoencoders

- Often perform similarly

VAEs vs. Standard Autoencoders

- Often perform similarly
- VAEs are better generative models

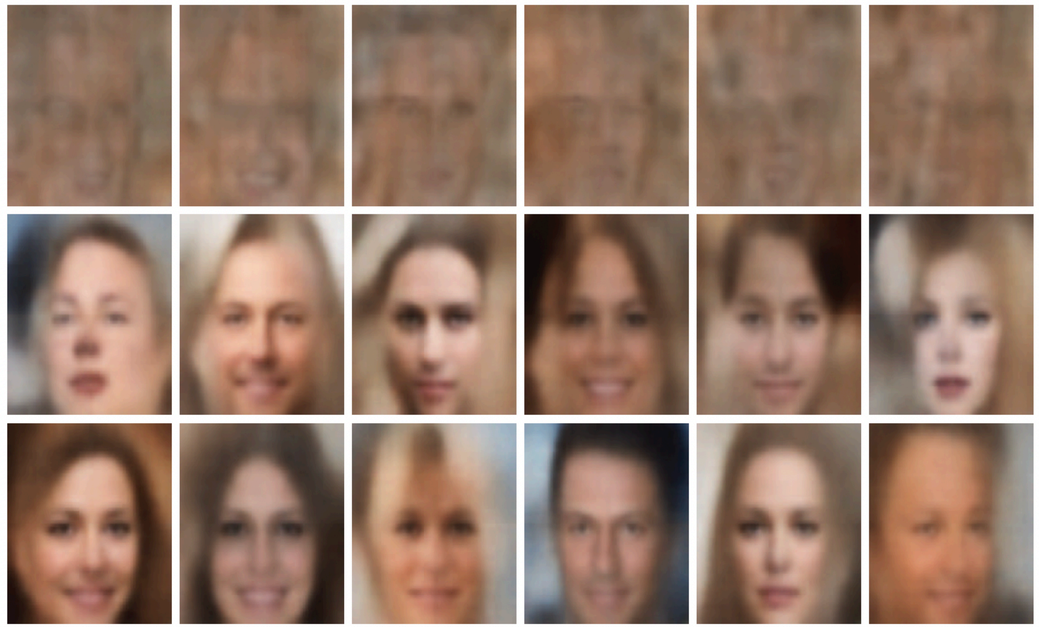


Figure 21.2: Illustration of unconditional image generation using (V)AEs trained on CelebA. Row 1: deterministic autoencoder. Row 2: β-VAE with $\beta = 0.5$. Row 3: VAE (with $\beta = 1$). Generated by celeba_vae_ae_comparison.ipynb.

VAEs vs. Standard Autoencoders

- Often perform similarly
- VAEs are better generative models

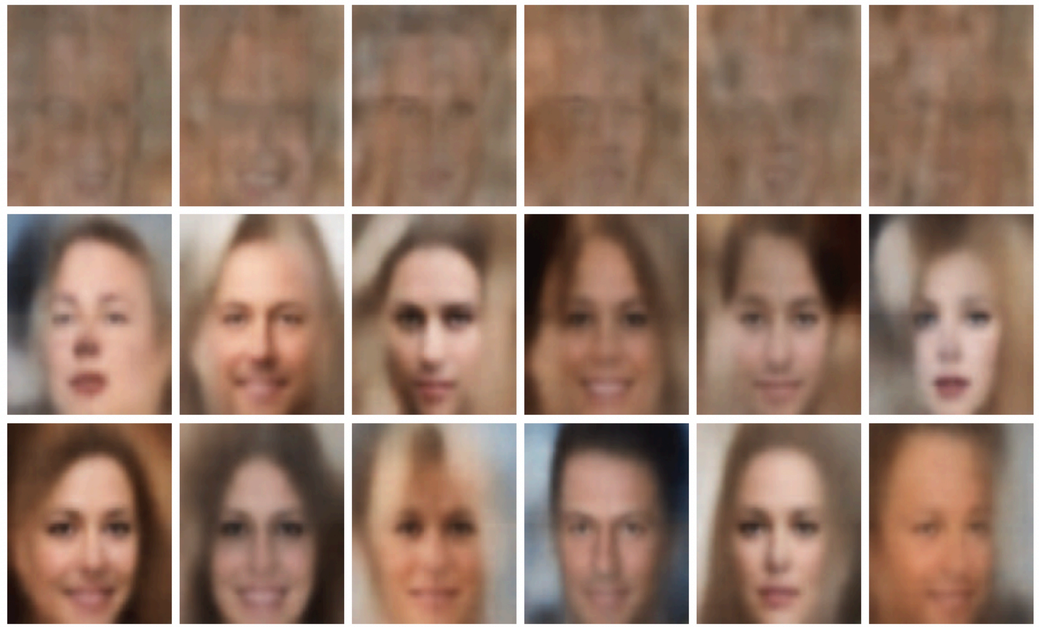
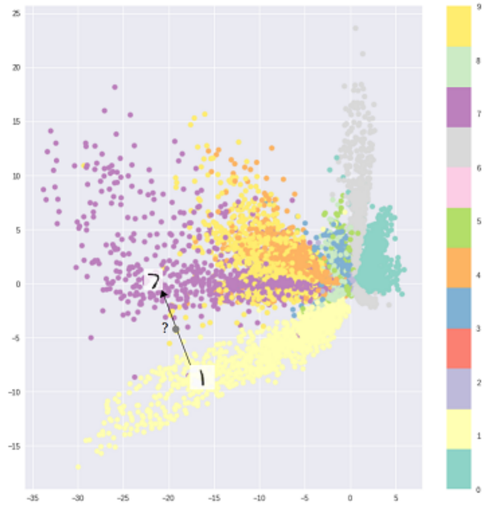


Figure 21.2: Illustration of unconditional image generation using (V)AEs trained on CelebA. Row 1: deterministic autoencoder. Row 2: β-VAE with $\beta = 0.5$. Row 3: VAE (with $\beta = 1$). Generated by celeba_vae_ae_comparison.ipynb.

AE Latents on MNIST



VAE Latents on MNIST

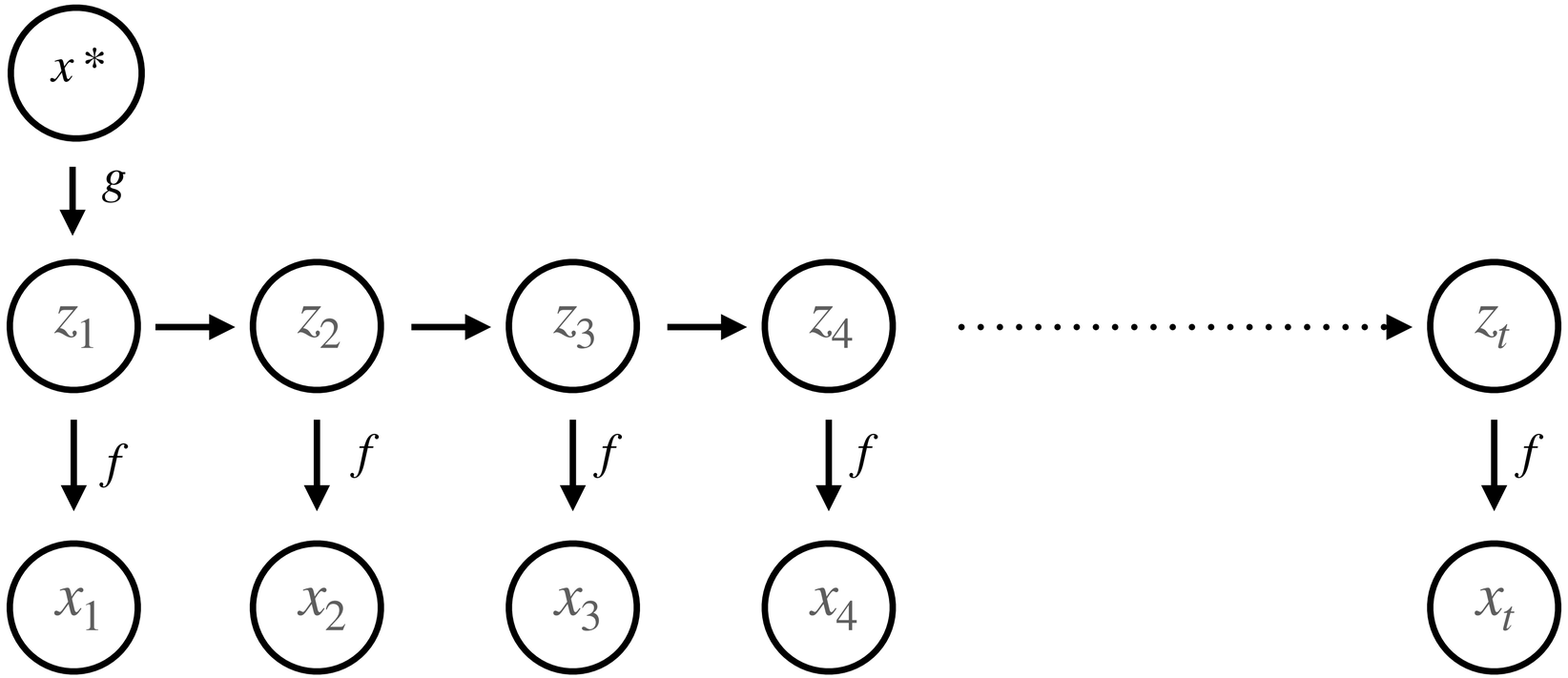
https://stats.stackexchange.com/questions/324340/when-should-i-use-a-variational-autoencoder-as-opposed-to-an-autoencoder

VAEs vs. Standard Autoencoders

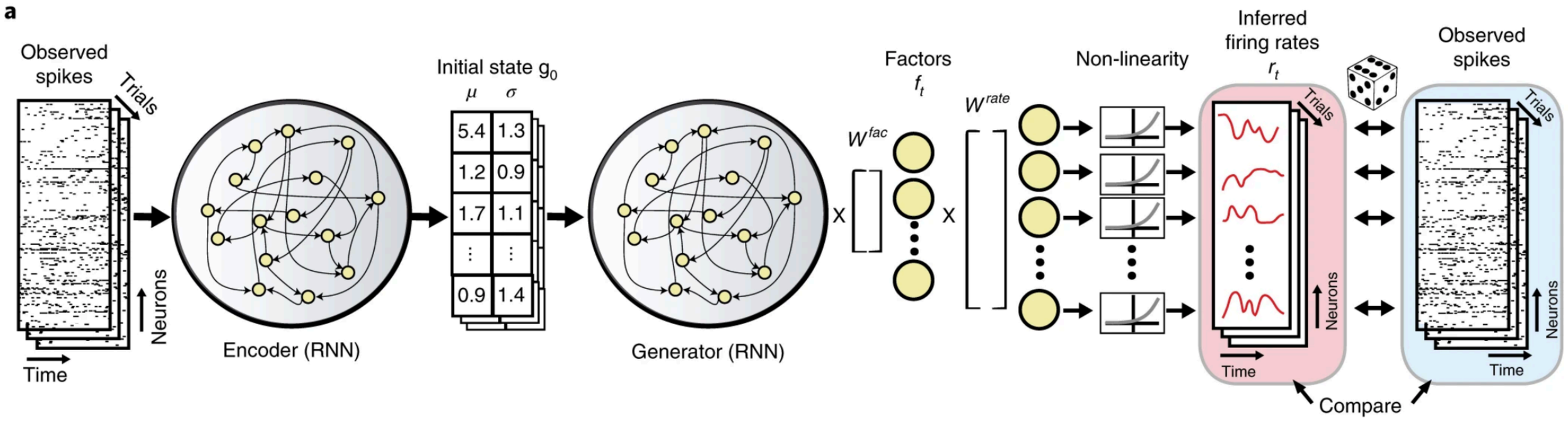
- Often perform similarly
- VAEs are better generative models
- VAEs are flexible probabilistic models (can change prior, noise distributions, etc).

Sequential Variational Autoencoder (sVAE)

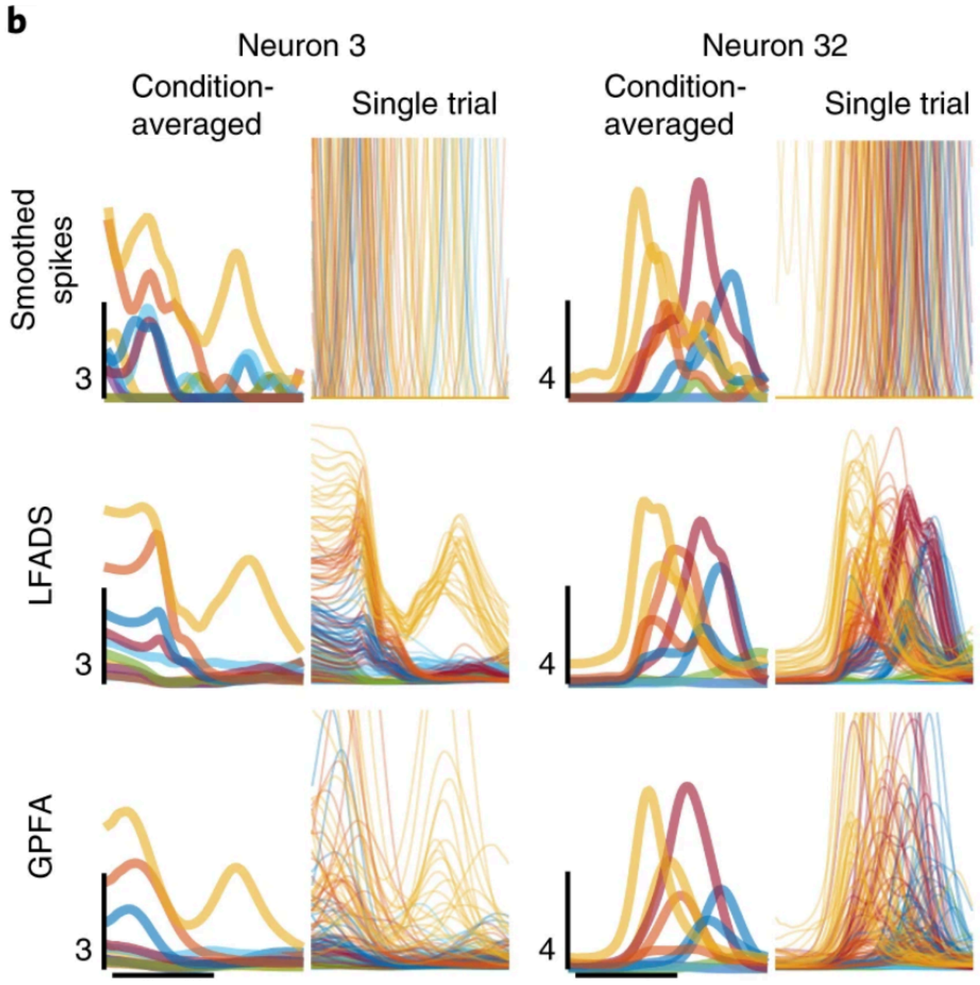
- VAE where the latents have a recurrent structure (e.g. are an RNN)

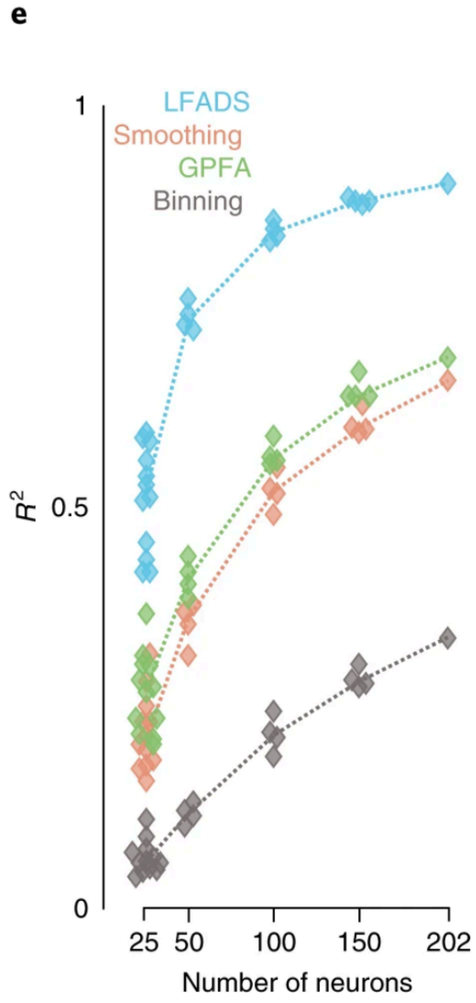


LFADS


a
Observed spikes
Trials
Neurons
Time
Encoder (RNN)
Initial state g_0
μ σ
5.4 1.3
1.2 0.9
1.7 1.1
0.9 1.4
Generator (RNN)
Factors f_t
W^{fac}
W^{rate}
Non-linearity
Inferred firing rates r_t
Observed spikes
Compare

LFADS


b
Neuron 3
Neuron 32
Condition-averaged
Single trial
Condition-averaged
Single trial
Smoothed spikes
LFADS
GPFA
3
4
3
4
3
4


e
LFADS
Smoothing
GPFA
Binning
1
0.5
0
R^2
25
50
100
150
202
Number of neurons

Dimensionality in Linear and Nonlinear Models

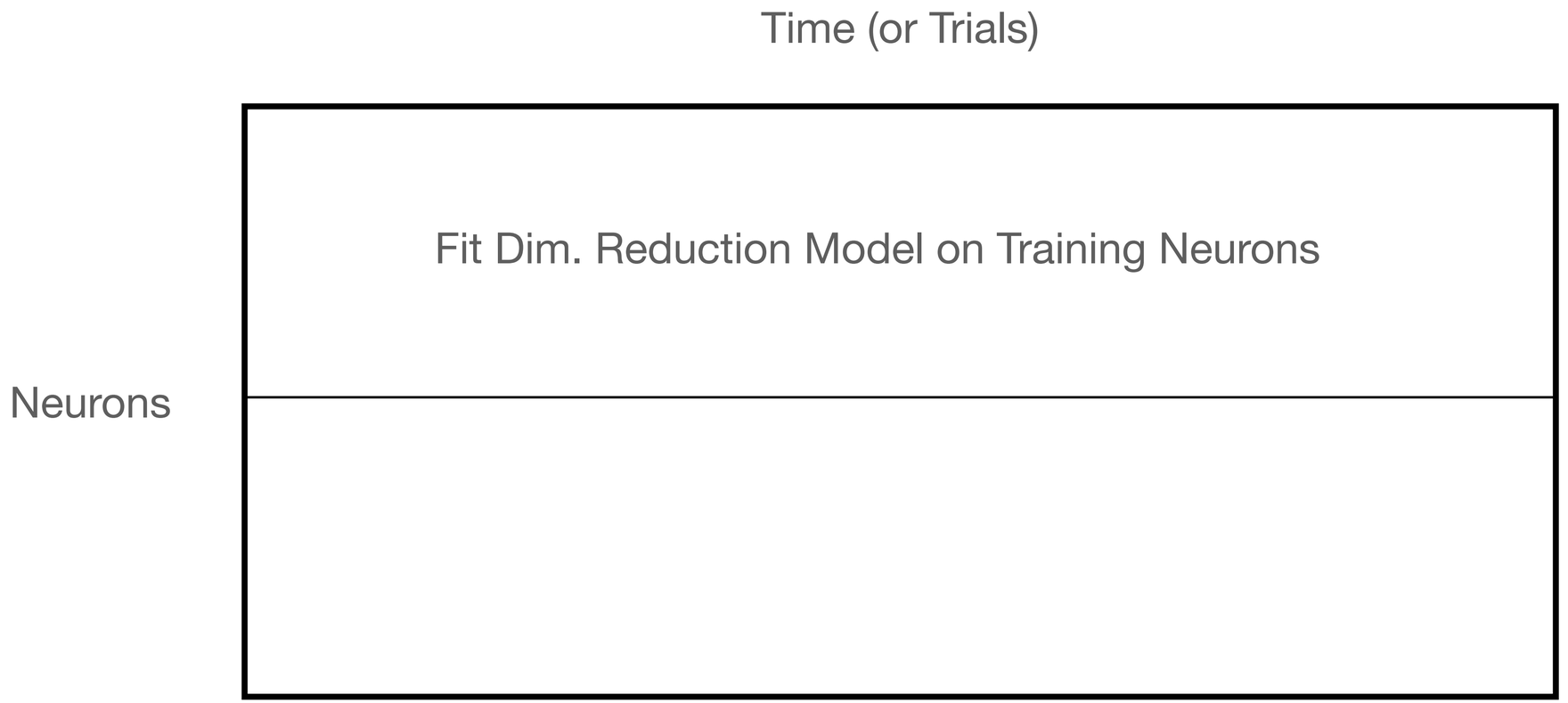
- *See Notebook*

Bi-cross-validation with Latent Variable Models + Dim. Reduction

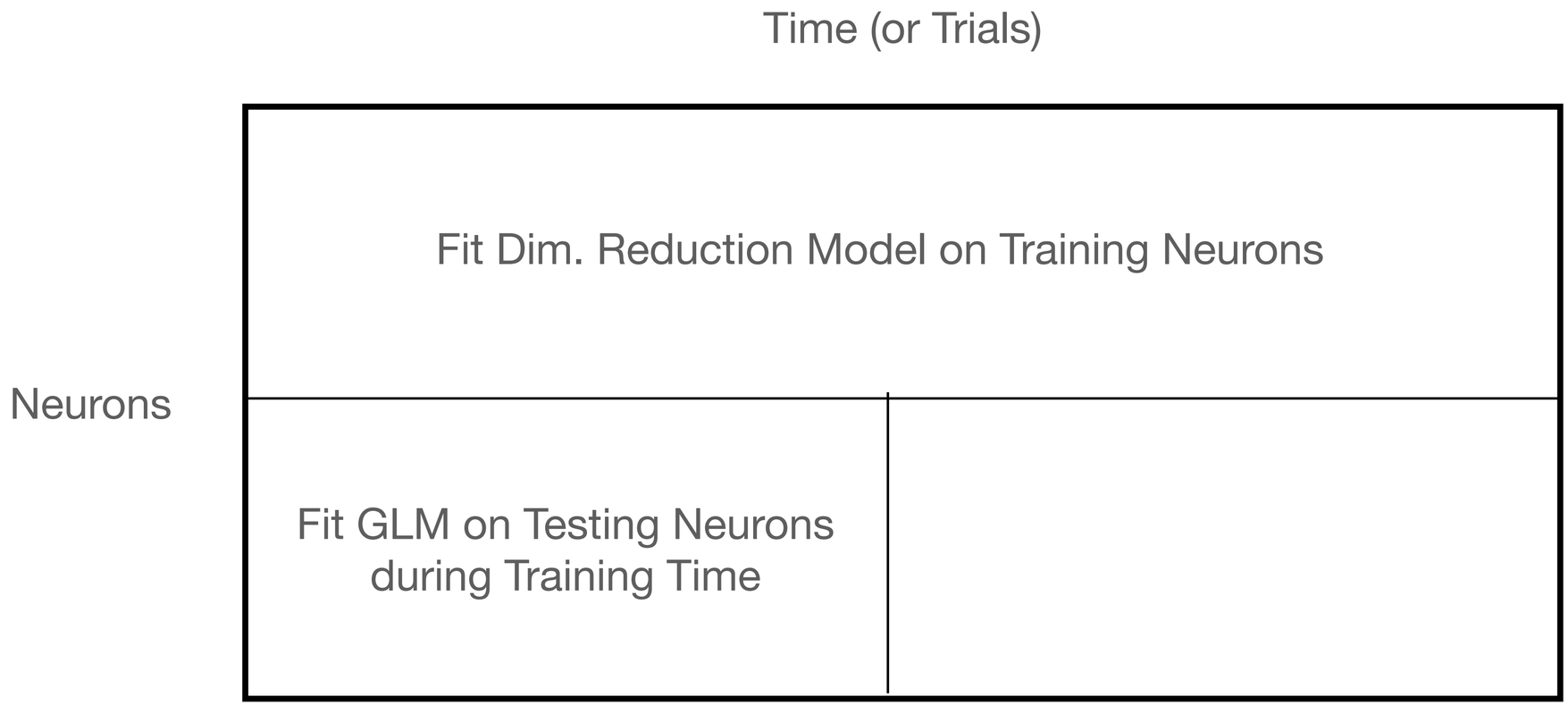
Time (or Trials)

Neurons

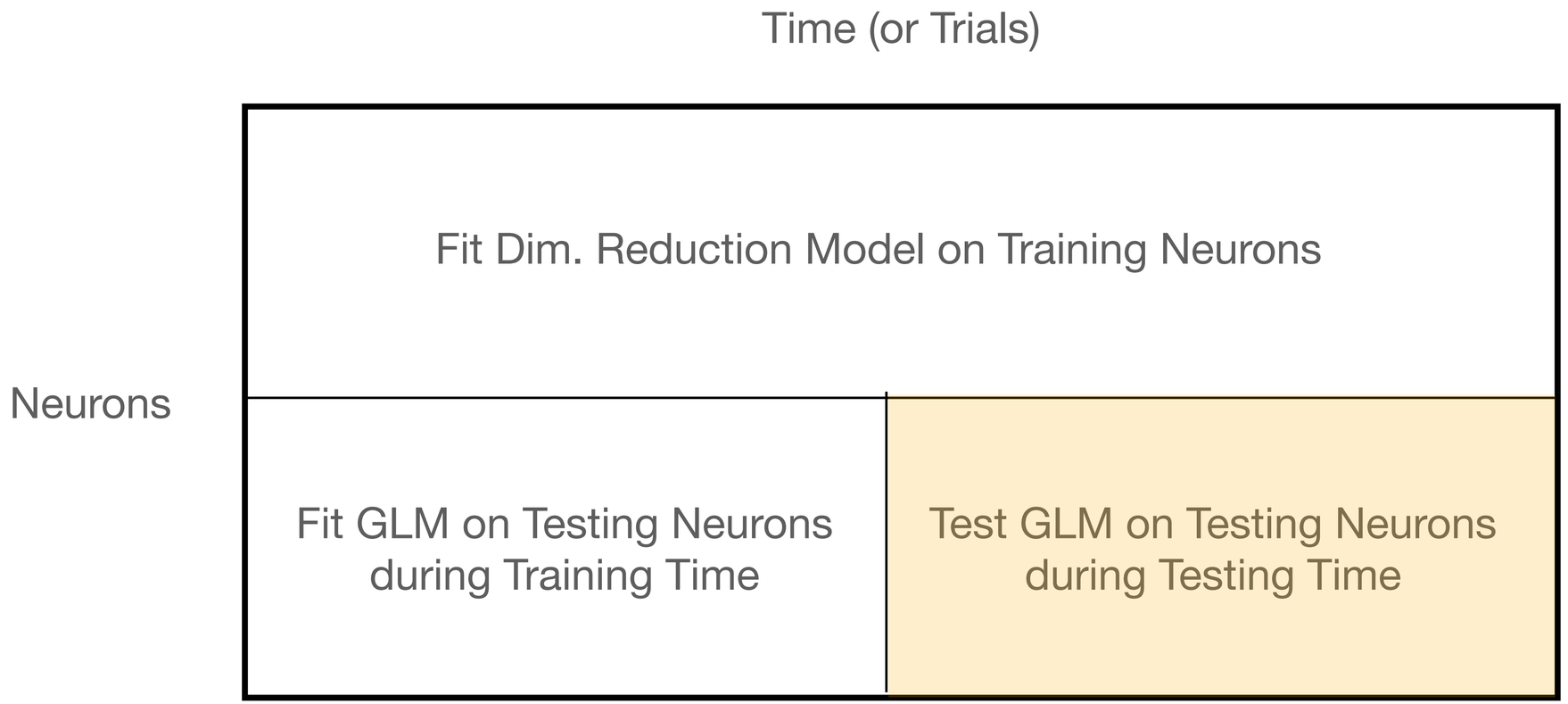
Bi-cross-validation with Latent Variable Models + Dim. Reduction



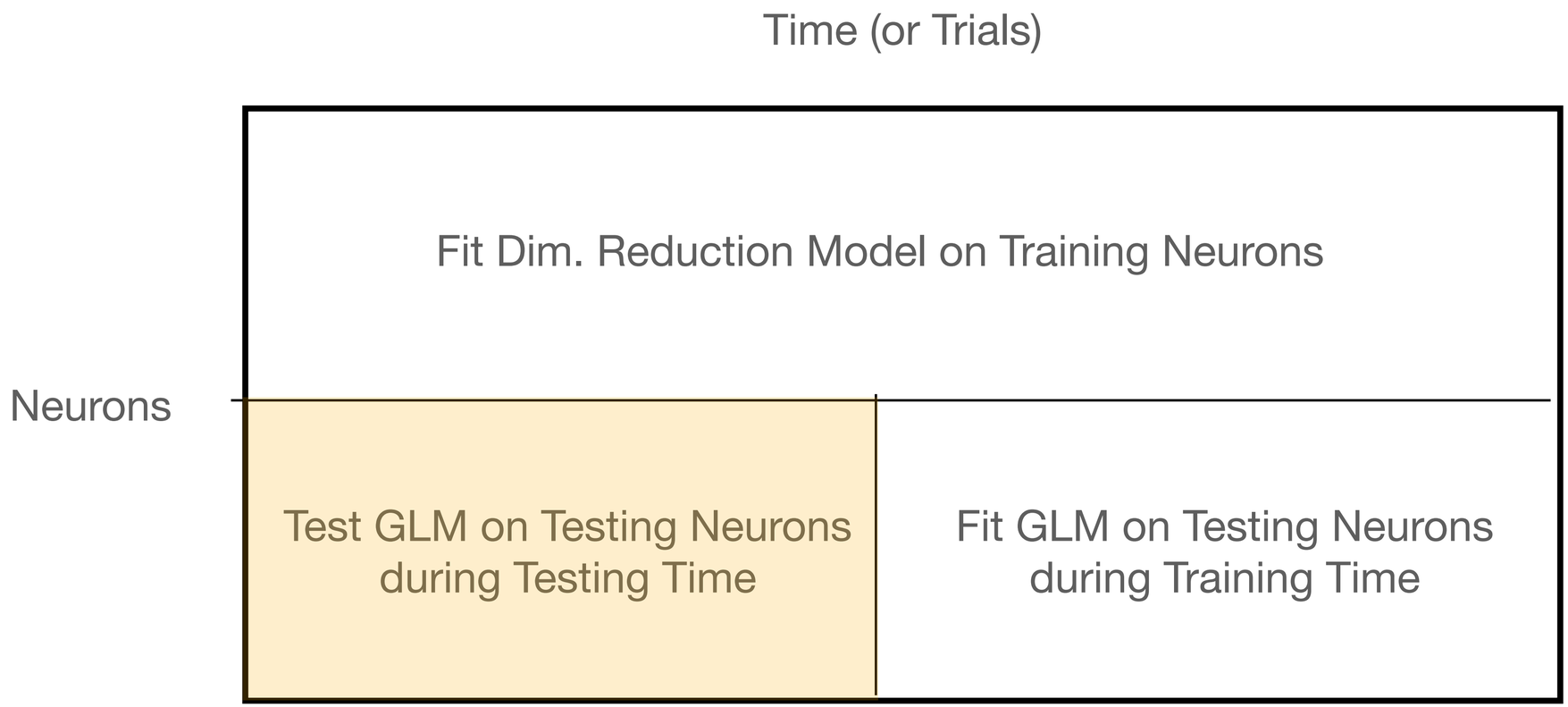
Bi-cross-validation with Latent Variable Models + Dim. Reduction



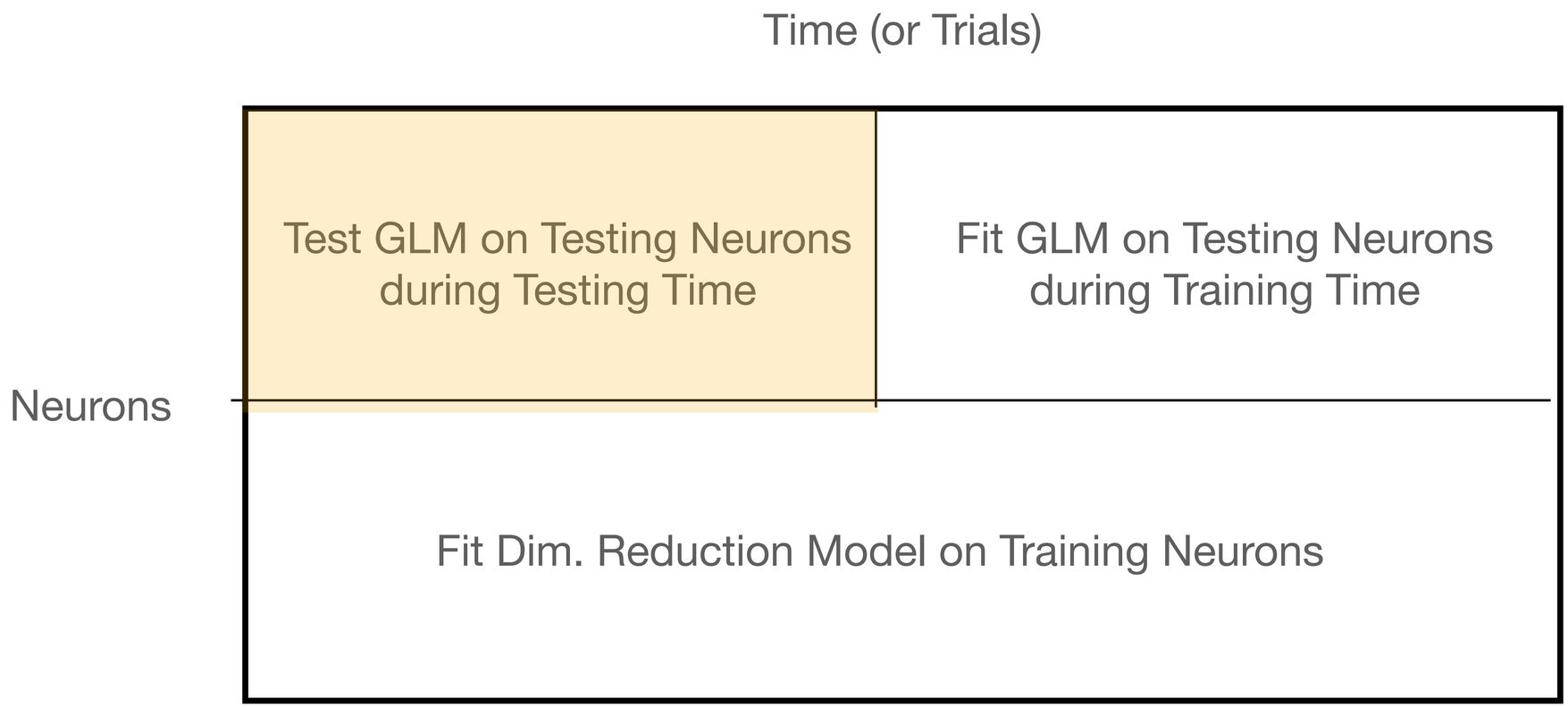
Bi-cross-validation with Latent Variable Models + Dim. Reduction



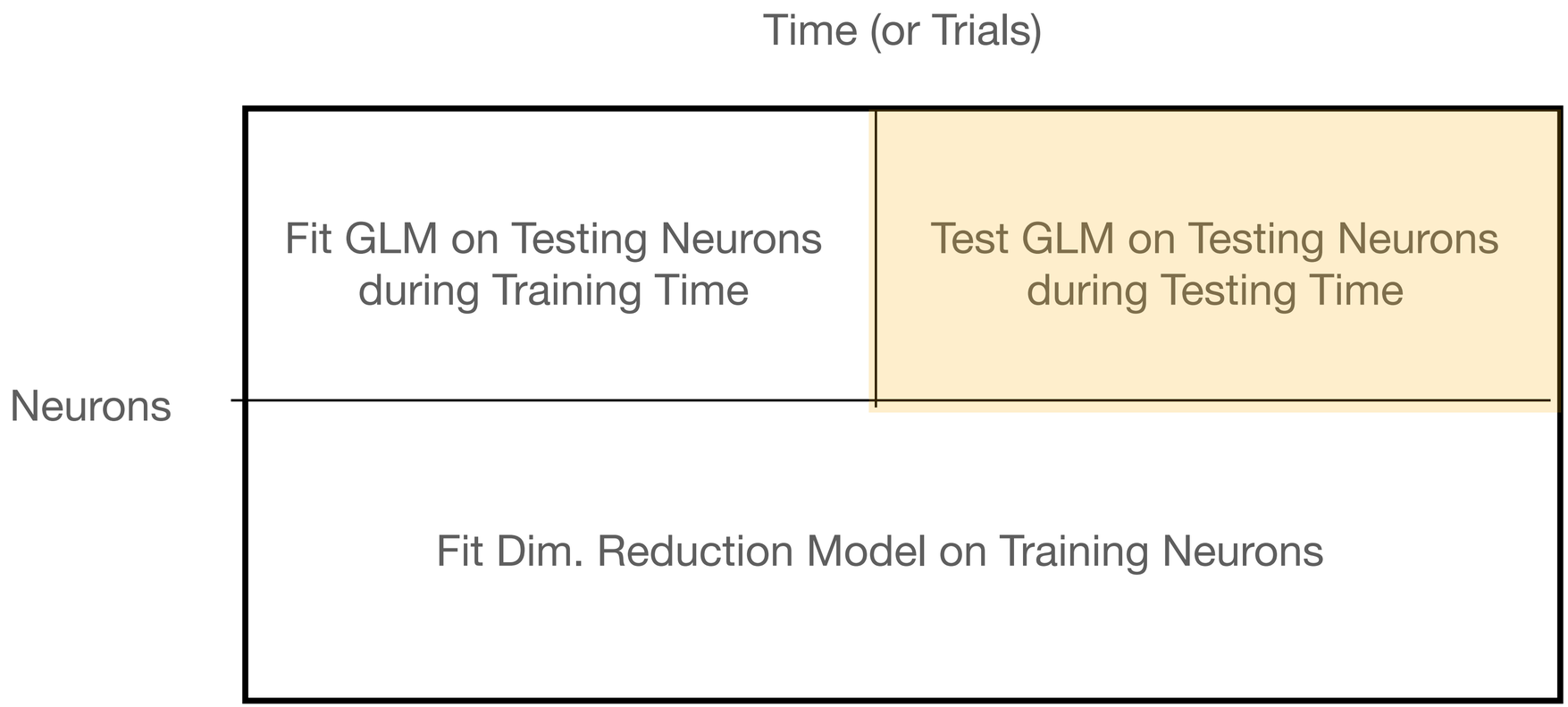
Bi-cross-validation with Latent Variable Models + Dim. Reduction



Bi-cross-validation with Latent Variable Models + Dim. Reduction

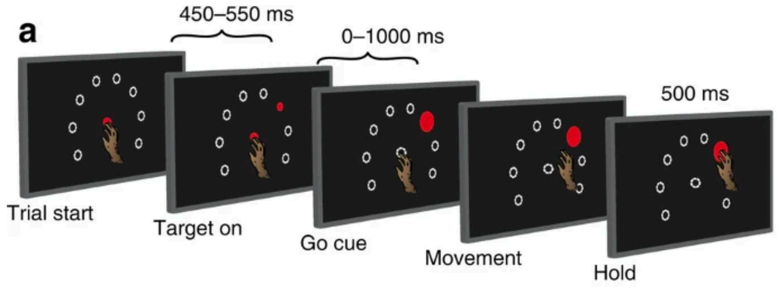


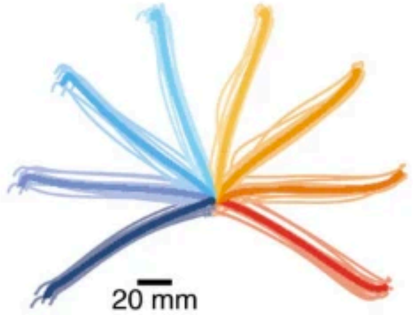
Bi-cross-validation with Latent Variable Models + Dim. Reduction



Sparse Component Analysis

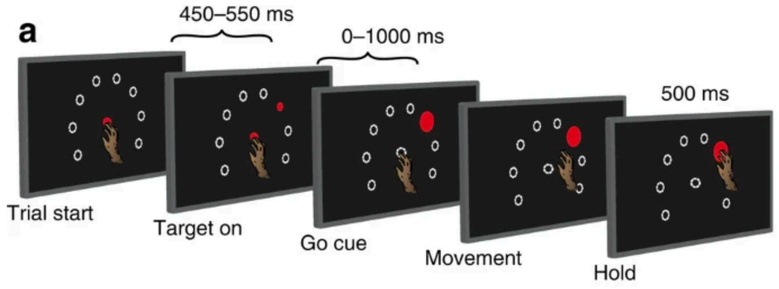
How can we find interpretable structure when neurons' responses are so mixed?

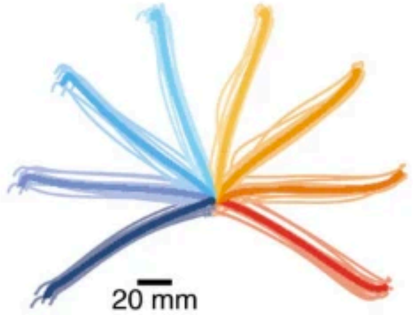




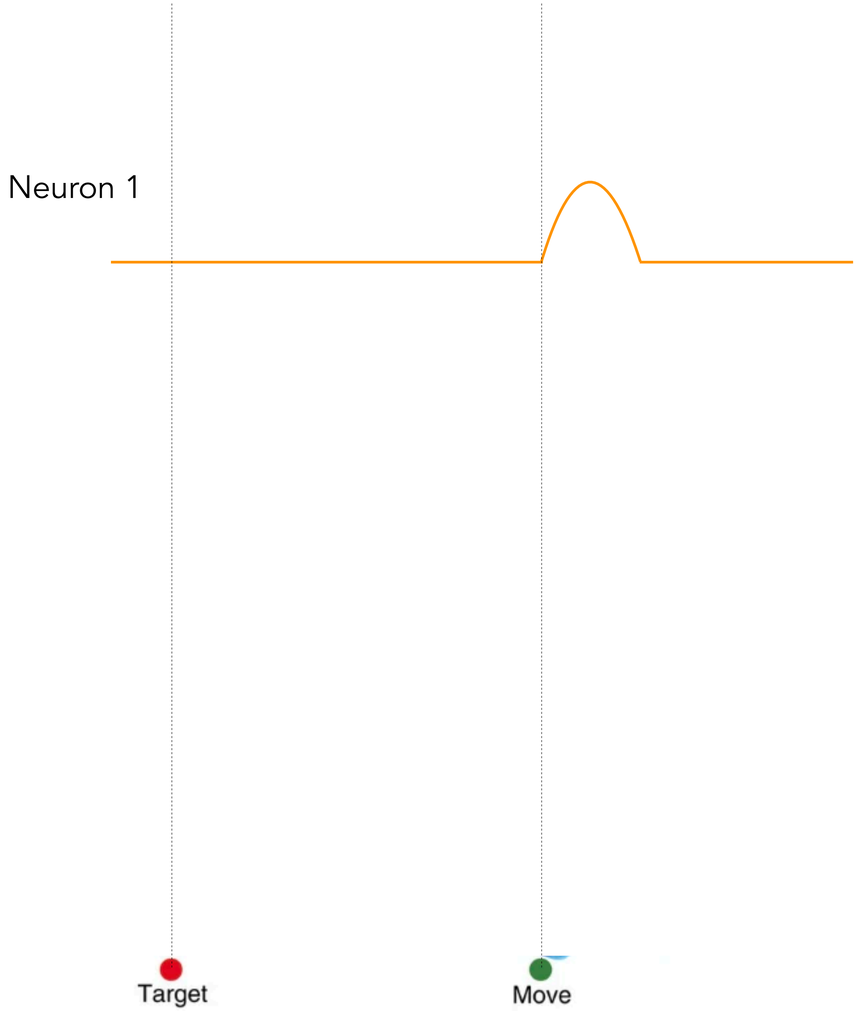
Elsayed et al 2016

How can we find interpretable structure when neurons' responses are so mixed?

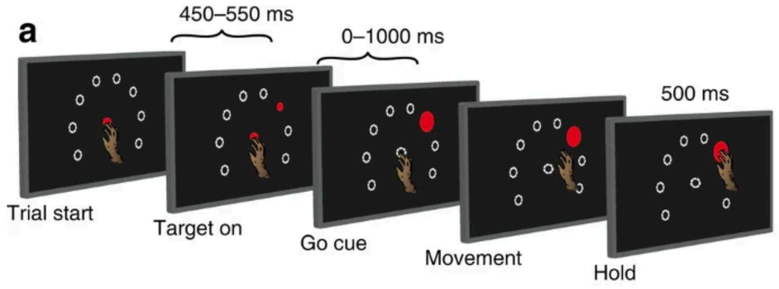


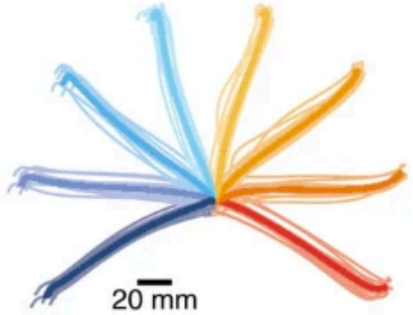


Elsayed et al 2016

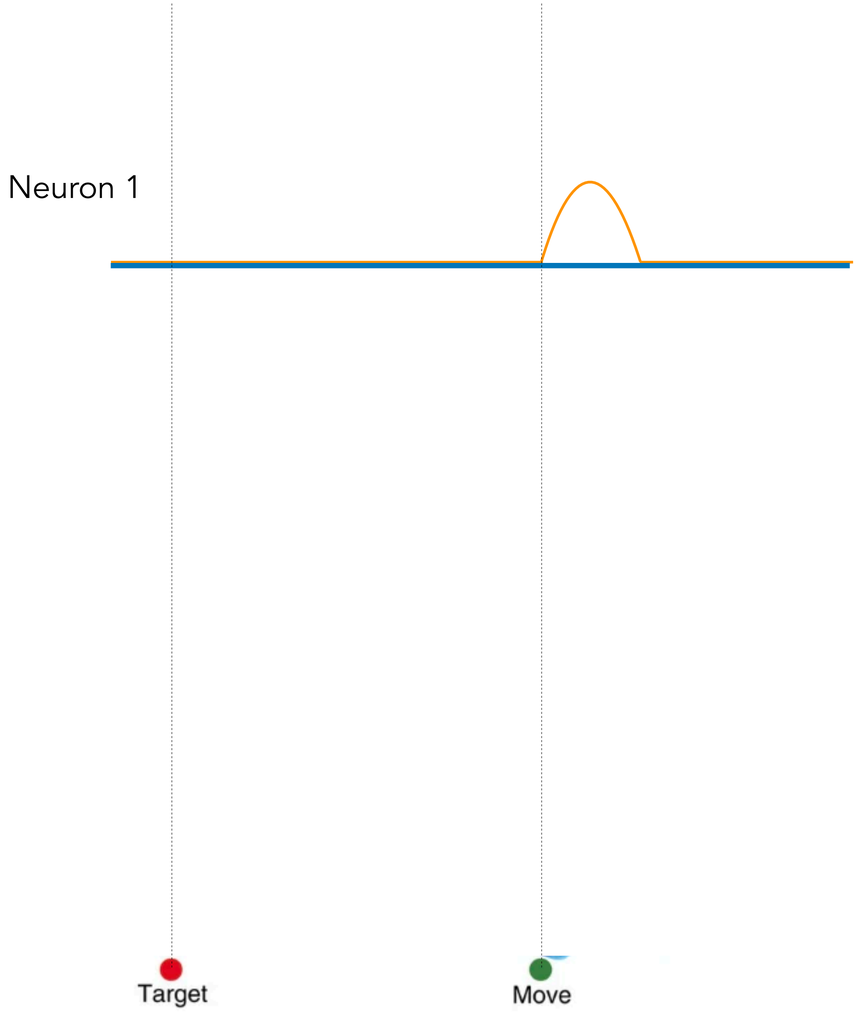


How can we find interpretable structure when neurons' responses are so mixed?

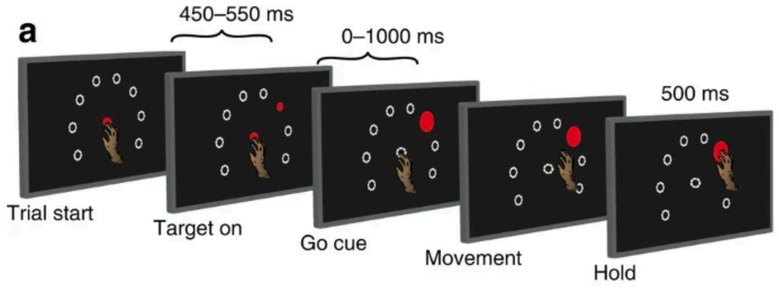


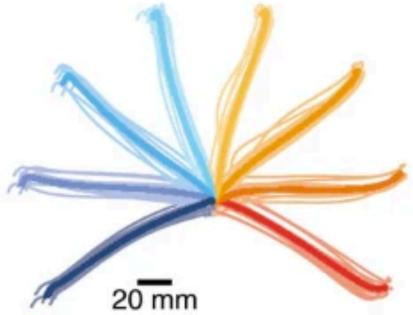


Elsayed et al 2016

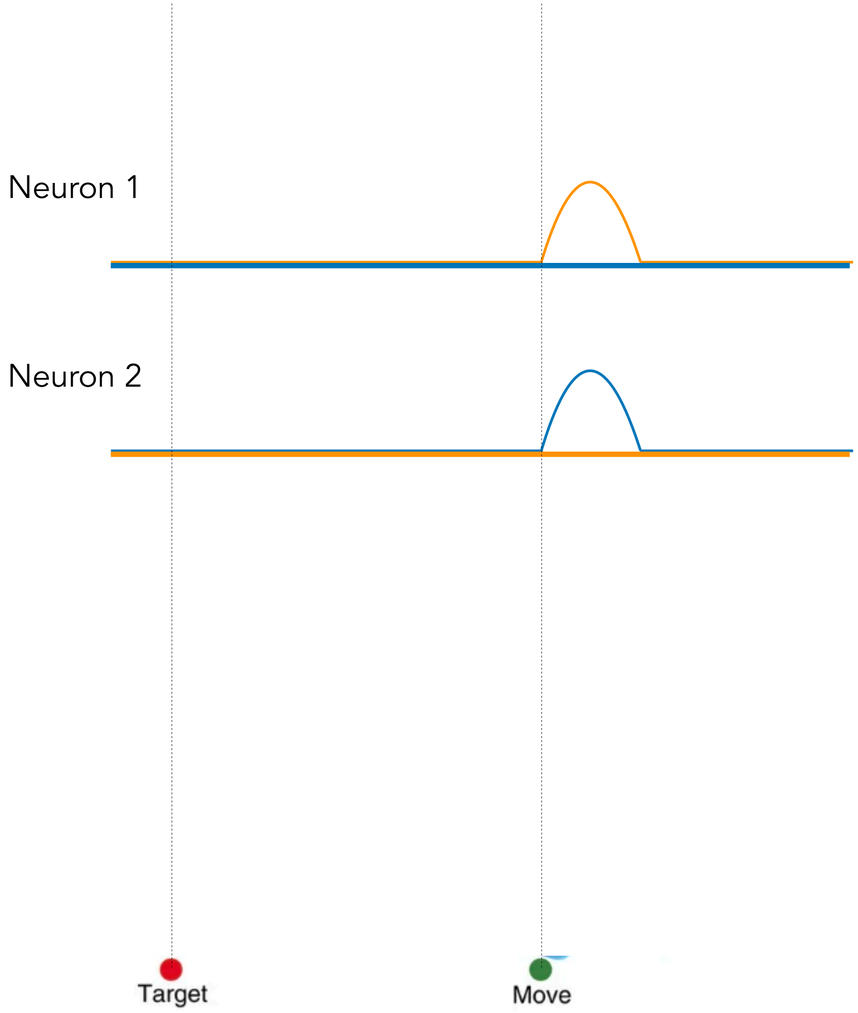


How can we find interpretable structure when neurons' responses are so mixed?

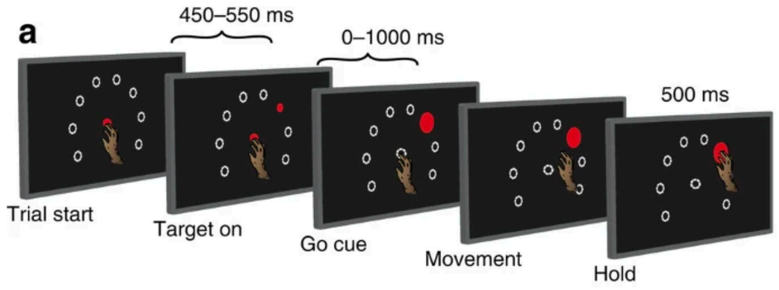


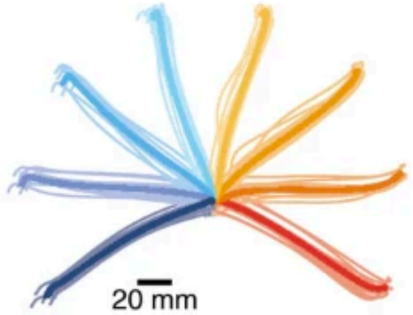


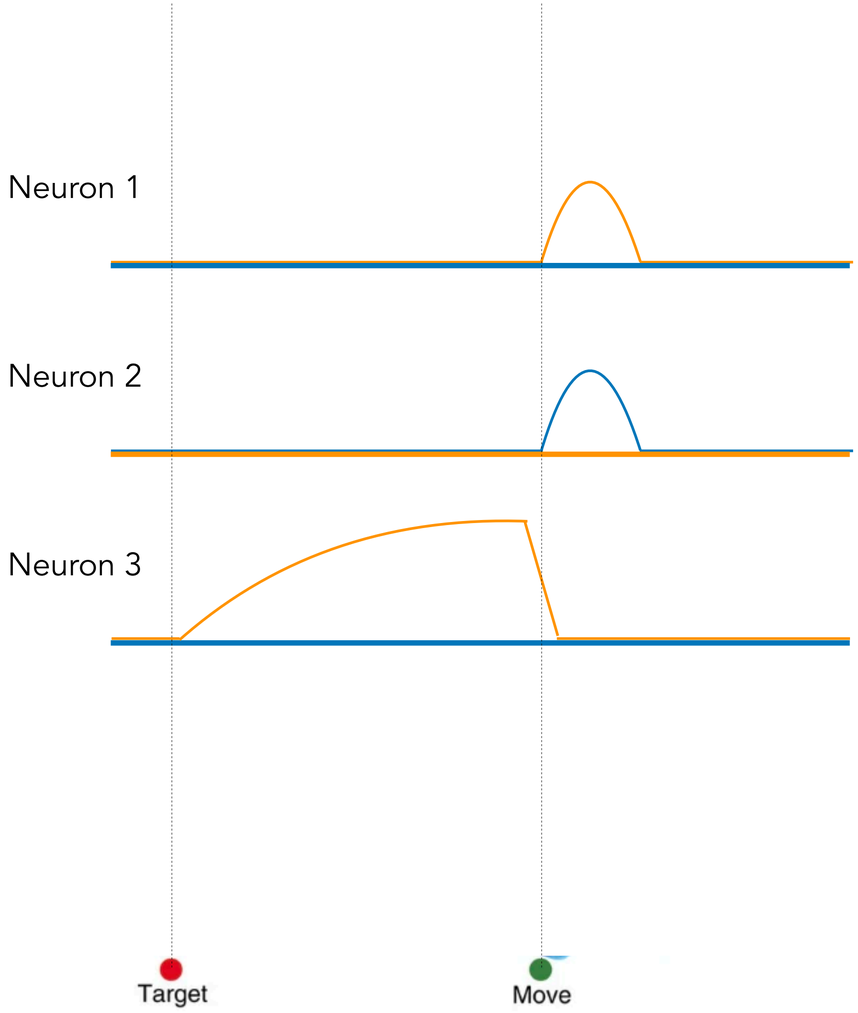
Elsayed et al 2016



How can we find interpretable structure when neurons' responses are so mixed?

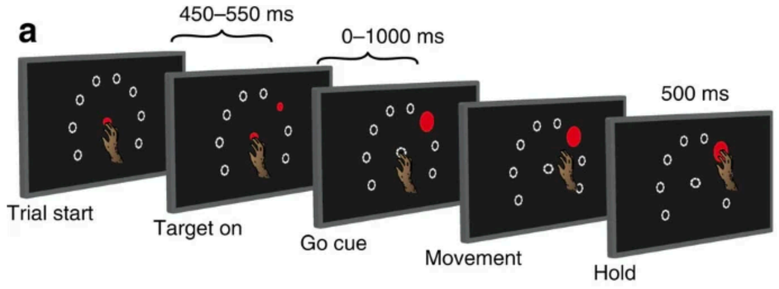


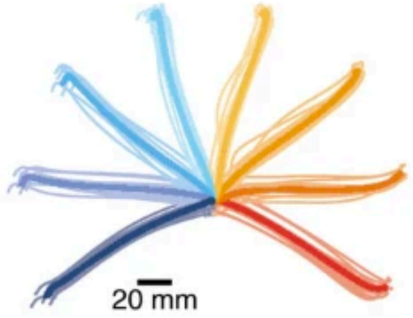


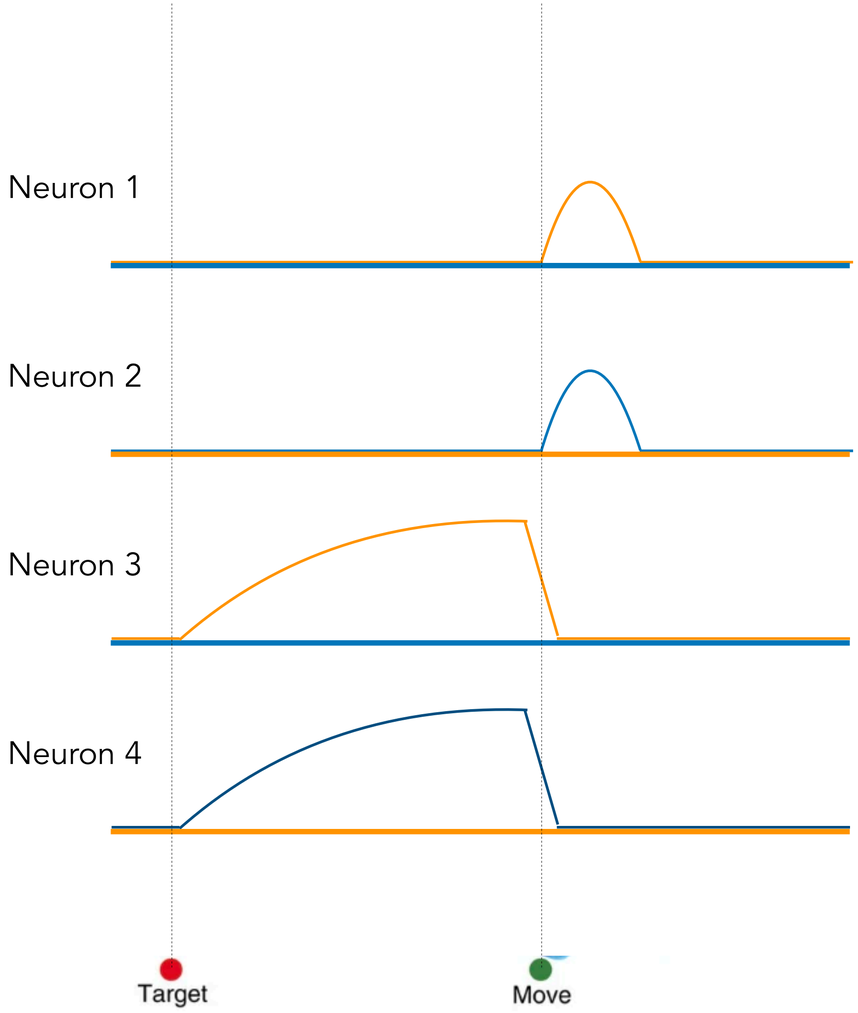


Elsayed et al 2016

How can we find interpretable structure when neurons' responses are so mixed?

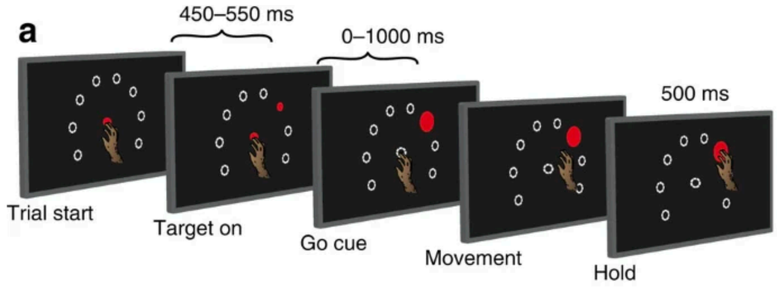


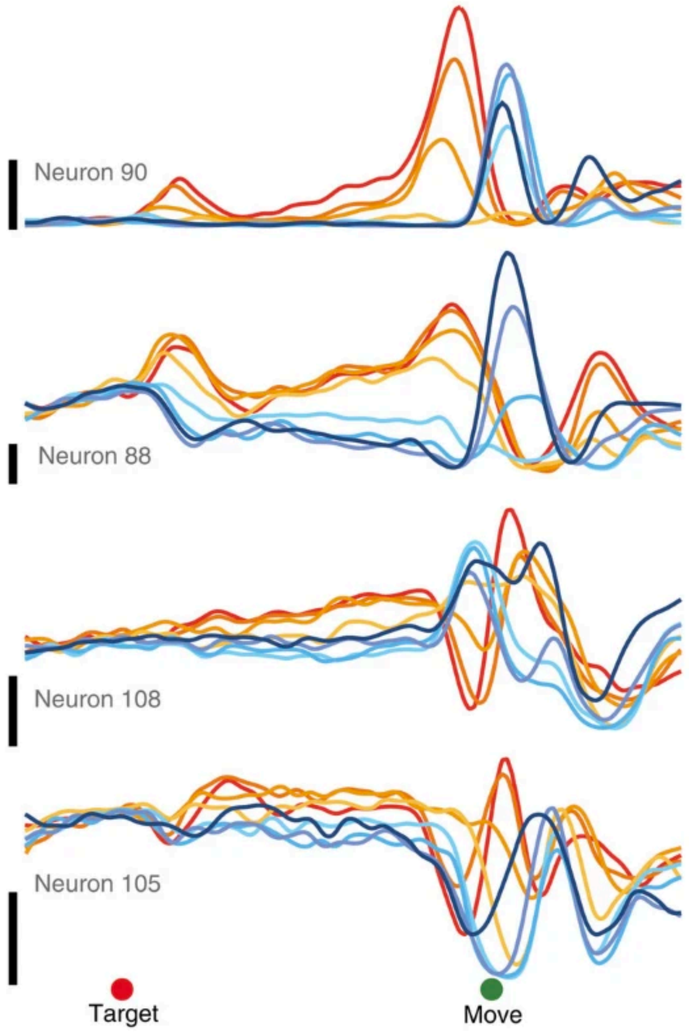




Elsayed et al 2016

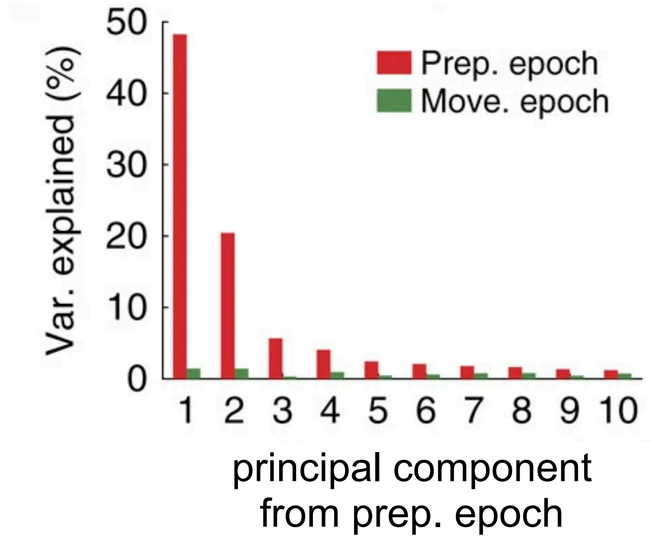
How can we find interpretable structure when neurons' responses are so mixed?

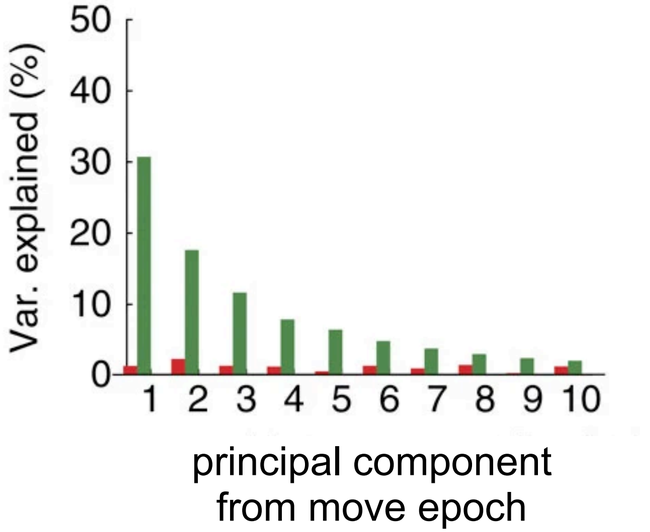




Elsayed et al 2016

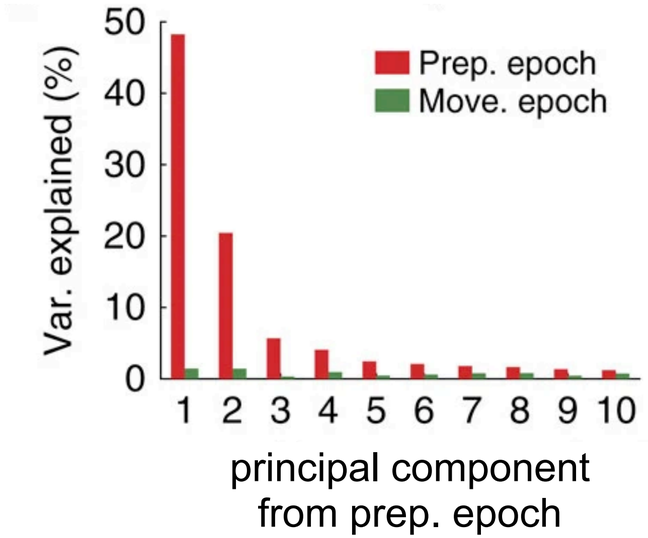
Different computations occur in different dimensions

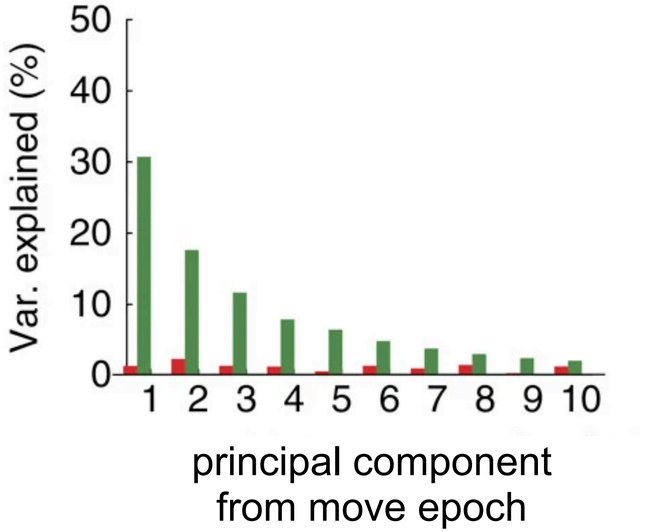




Elsayed et al 2016

Different computations occur in different dimensions

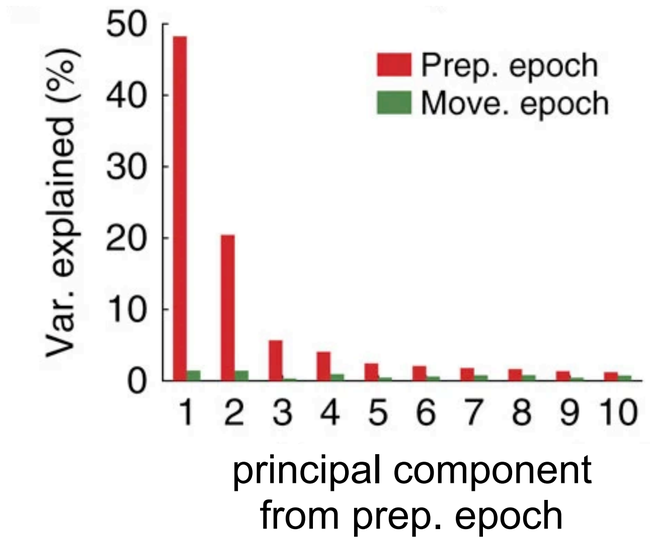


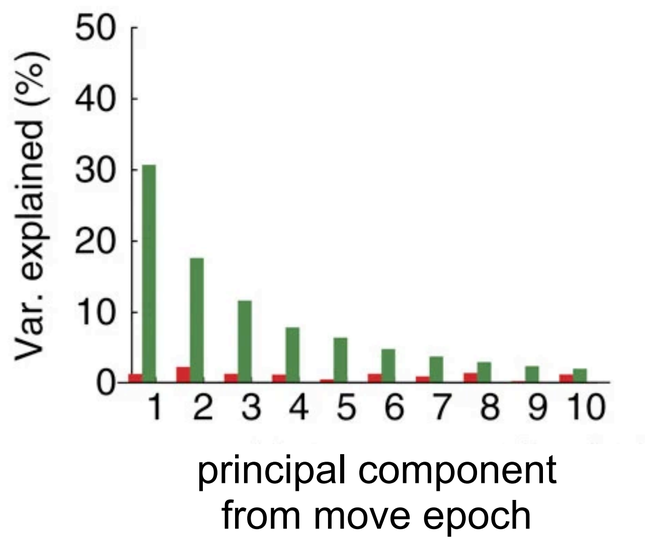


Elsayed et al 2016

- Working memory and motor preparation in prefrontal cortex (Tang et al 2020)
- Evaluation and selection in orbitofrontal cortex (Yoo et al 2020)

Different computations occur in different dimensions

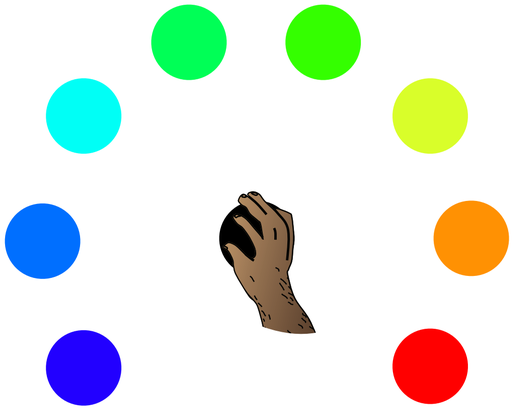


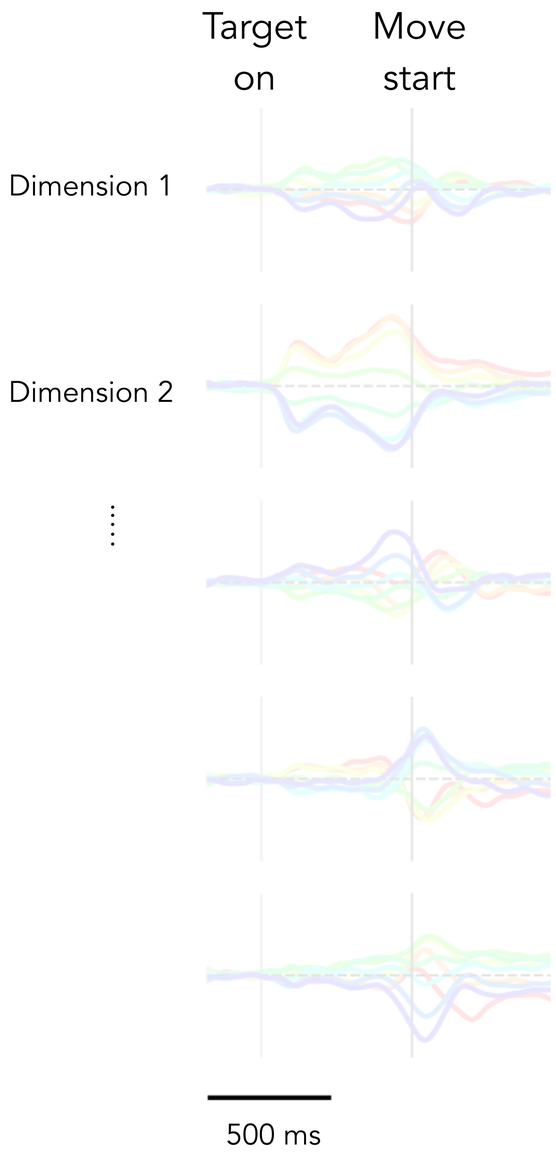


Elsayed et al 2016

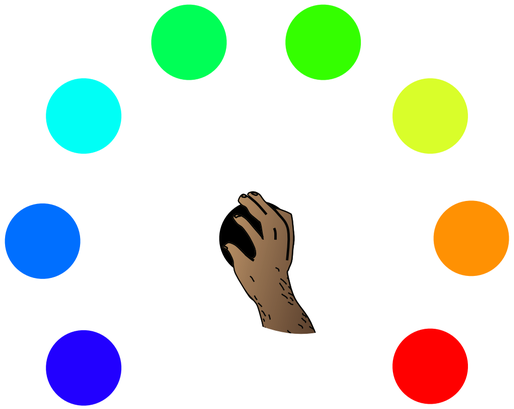
- *Need to know specific times of interest*

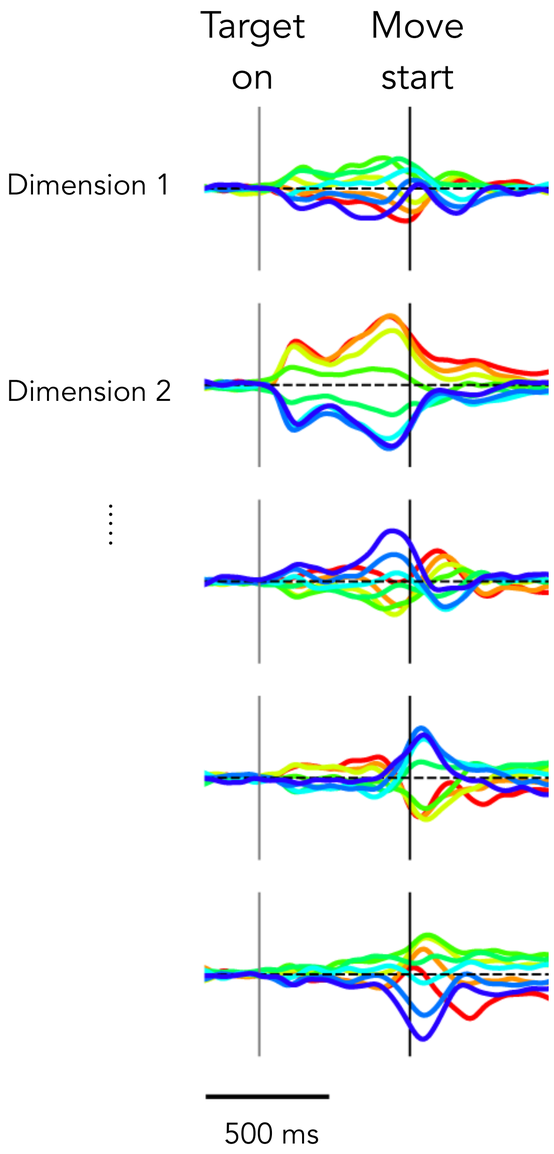
Mixing in the dimensions of PCA





Mixing in the dimensions of PCA



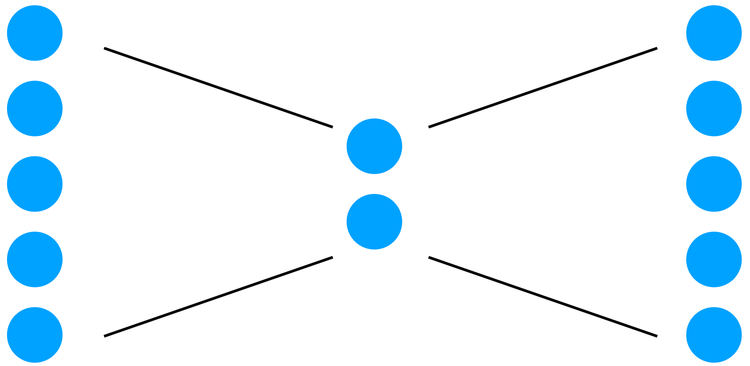


Sparse Component Analysis (SCA)

- **Goal:** find low-D representation whose dimensions are only used at some times (i.e. used sparsely)

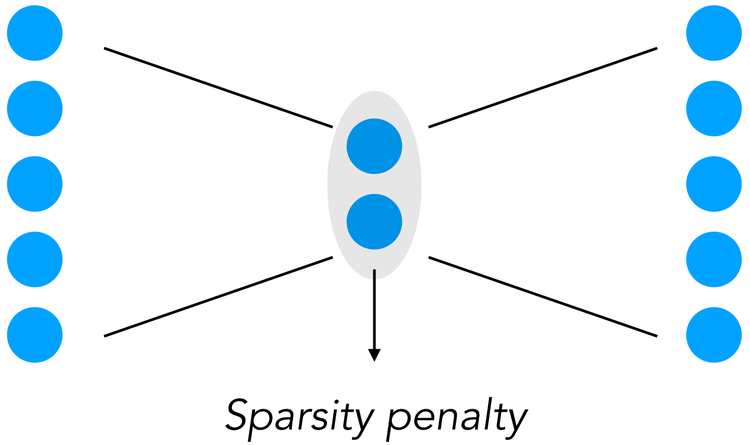
Sparse Component Analysis (SCA)

- **Goal:** find low-D representation whose dimensions are only used at some times (i.e. used sparsely)



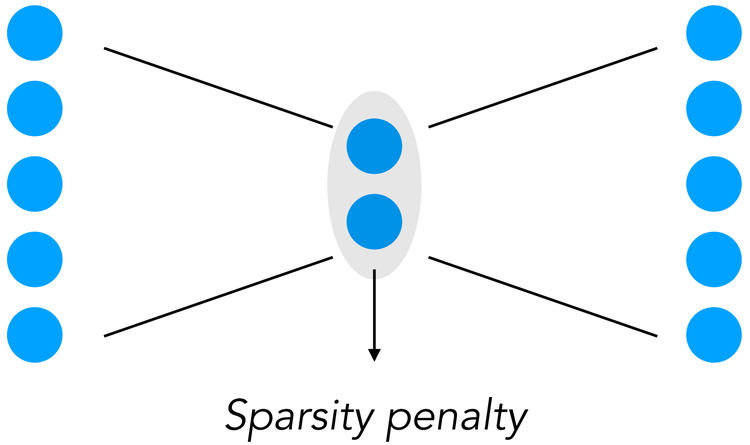
Sparse Component Analysis (SCA)

- **Goal:** find low-D representation whose dimensions are only used at some times (i.e. used sparsely)



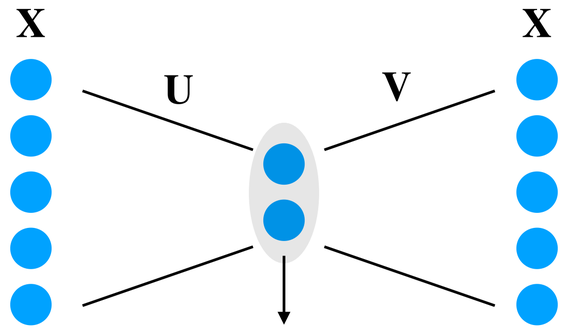
Sparse Component Analysis (SCA)

- **Goal:** find low-D representation whose dimensions are only used at some times (i.e. used sparsely)



- Unsupervised (time epochs and behavior are not input into model)

Sparse Component Analysis (SCA)

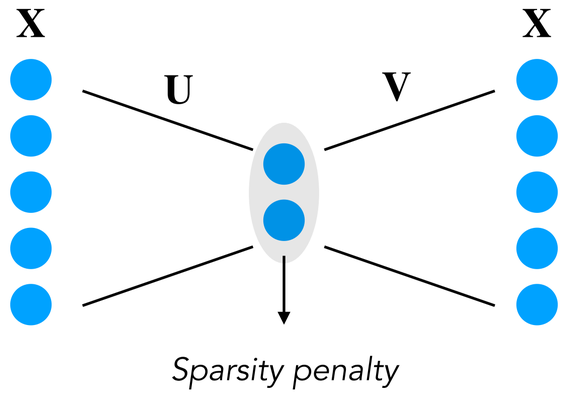


Optimization:

$$\arg\min_{U,V} \left(\| (\mathbf{X} - \mathbf{XUV}) \|^2 \right)$$

Reconstruction

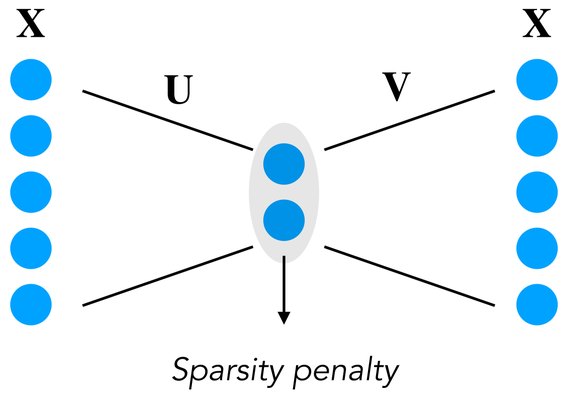
Sparse Component Analysis (SCA)



Optimization:

$$\arg\min_{U,V} \left(\underbrace{\| (\mathbf{X} - \mathbf{XUV}) \|^2}_{\text{Reconstruction}} + \underbrace{\lambda \| \mathbf{XU} \|_1}_{\text{Sparsity}} \right)$$

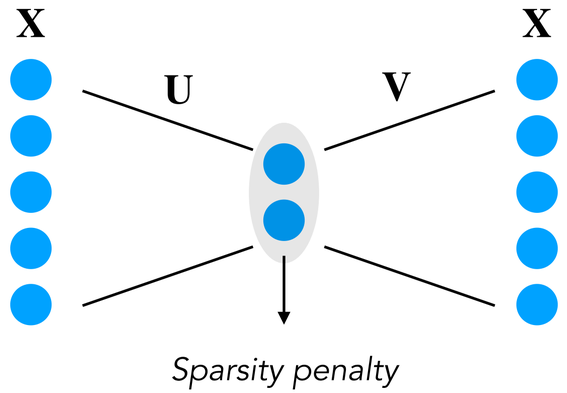
Sparse Component Analysis (SCA)



Optimization:

$$\arg\min_{U,V} \left(\underbrace{\| (\mathbf{X} - \mathbf{XUV}) \|^2}_{\text{Reconstruction}} + \underbrace{\lambda \| \mathbf{XU} \|_1}_{\text{Sparsity}} + \underbrace{\gamma \| \mathbf{V^T V} - \mathbf{I} \|}_{\text{Orthogonality}} \right)$$

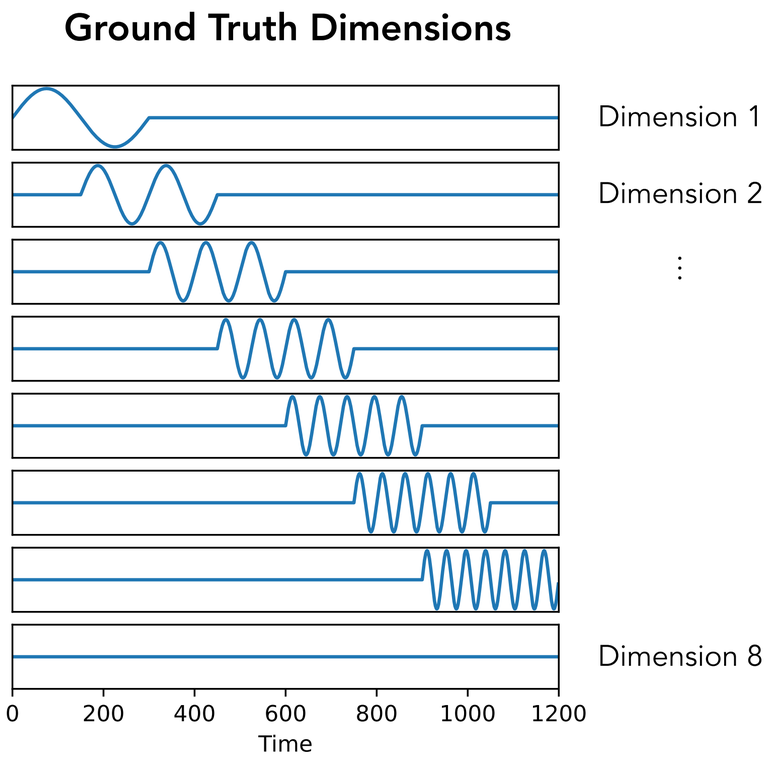
Sparse Component Analysis (SCA)



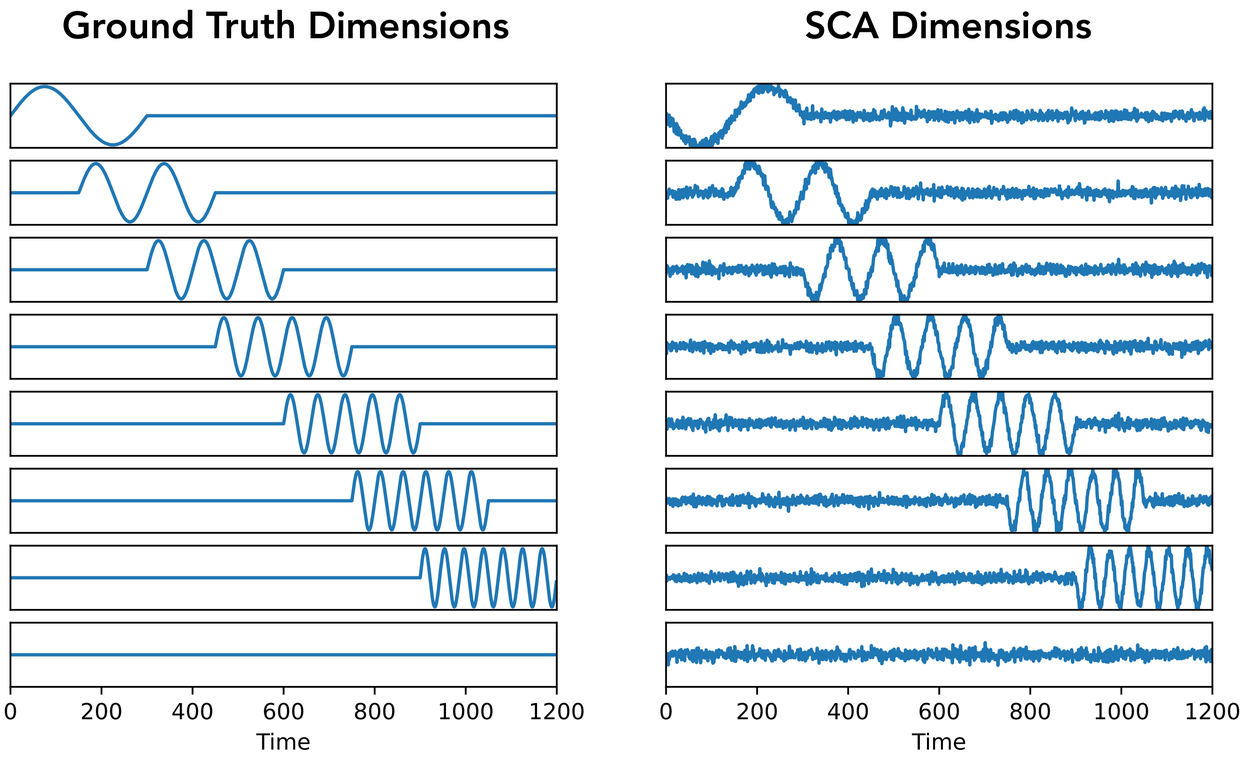
Optimization:

$$\arg\min_{U,V} \left(\underbrace{\| \mathbf{W}(\mathbf{X} - \mathbf{XUV}) \|^2}_{\text{Reconstruction}} + \underbrace{\lambda \| \mathbf{XU} \|_1}_{\text{Sparsity}} + \underbrace{\gamma \| \mathbf{V^T V} - \mathbf{I} \|}_{\text{Orthogonality}} \right)$$

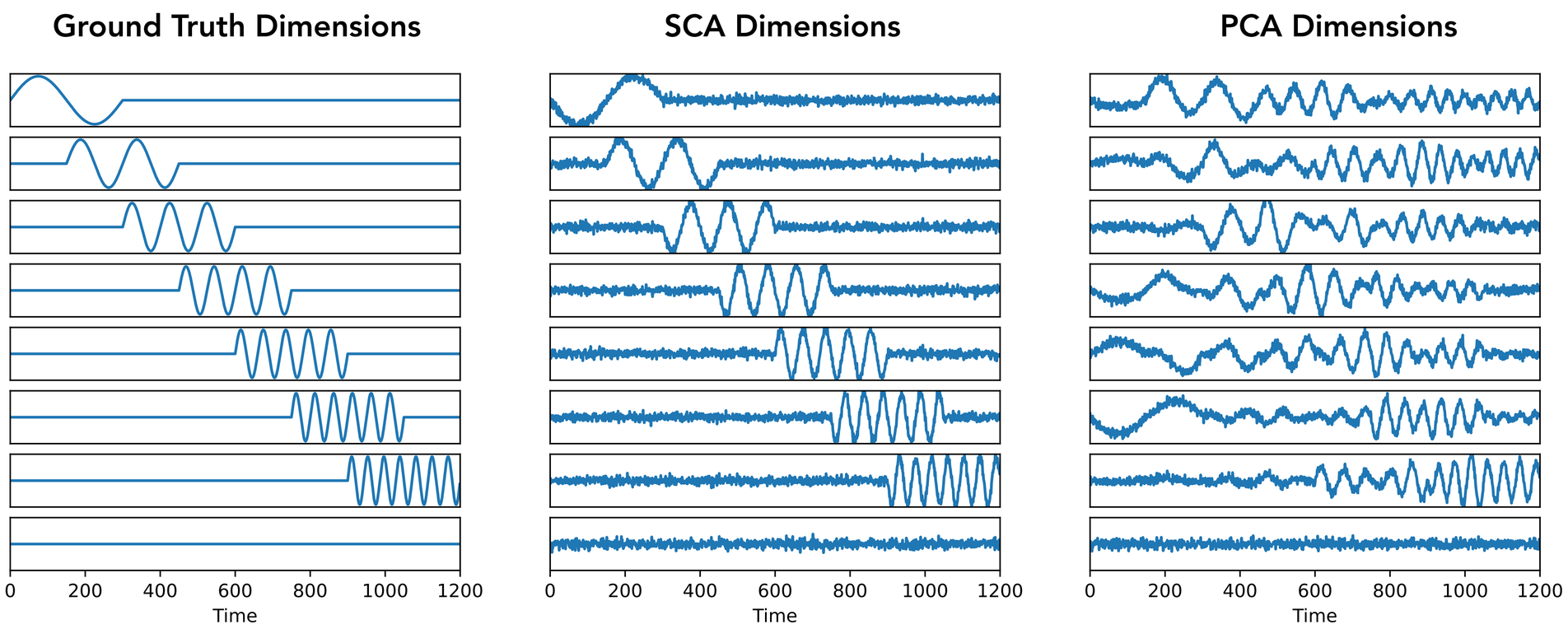
Simulation example



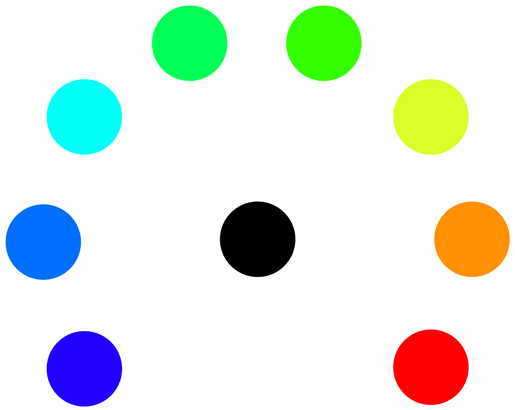
Simulation example



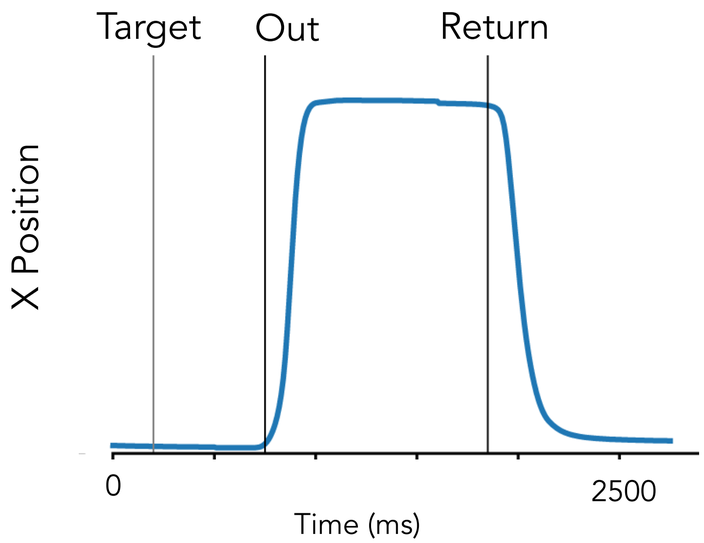
Simulation example

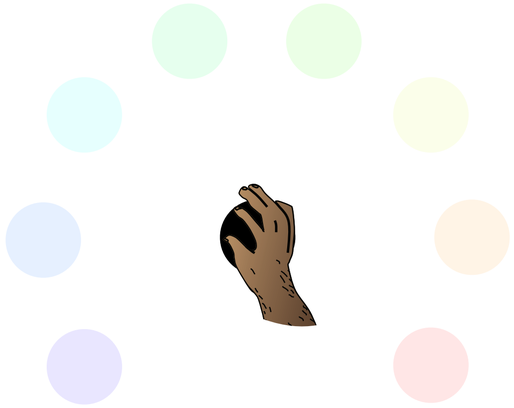


Center-out and self-paced return task

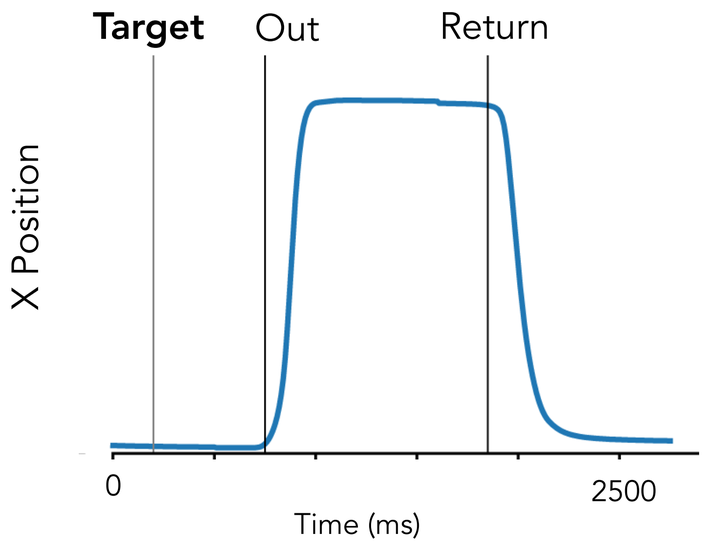


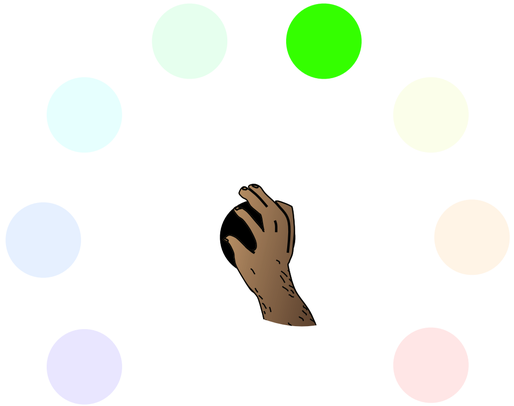
Center-out and self-paced return task



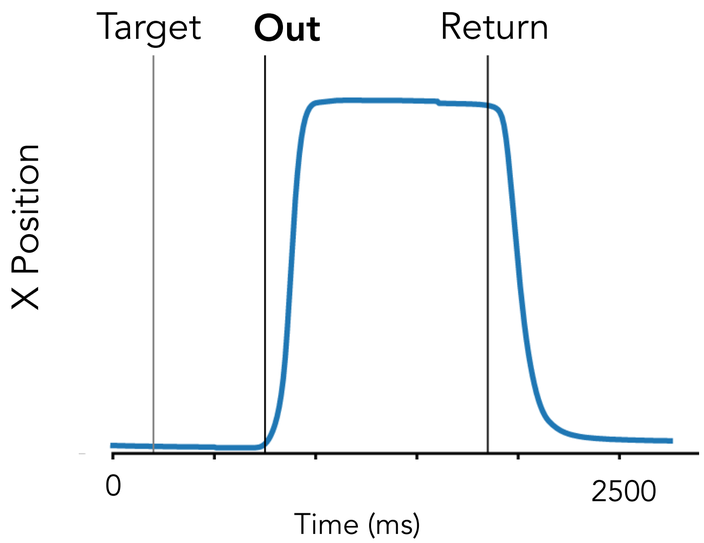


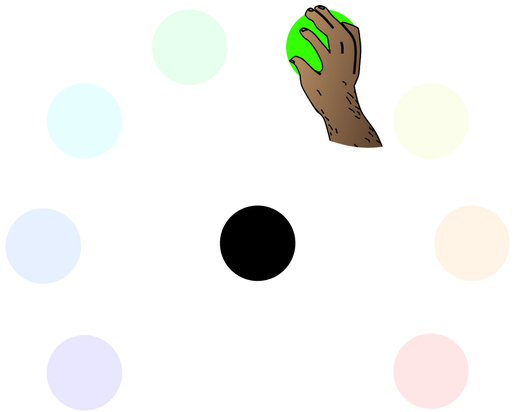
Center-out and self-paced return task



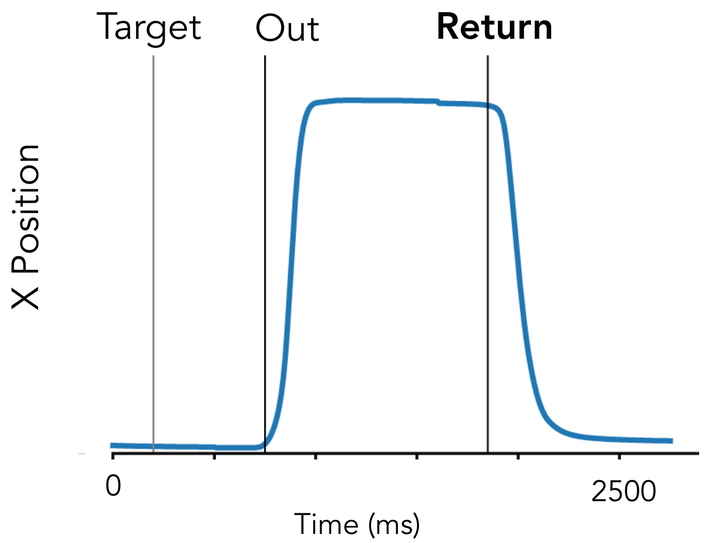


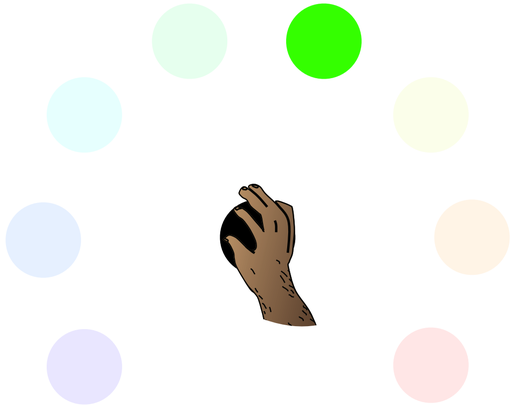
Center-out and self-paced return task



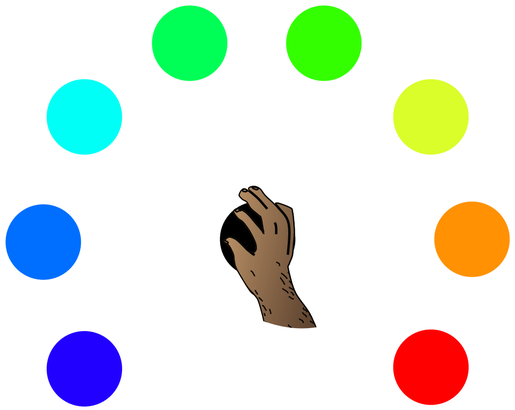


Center-out and self-paced return task



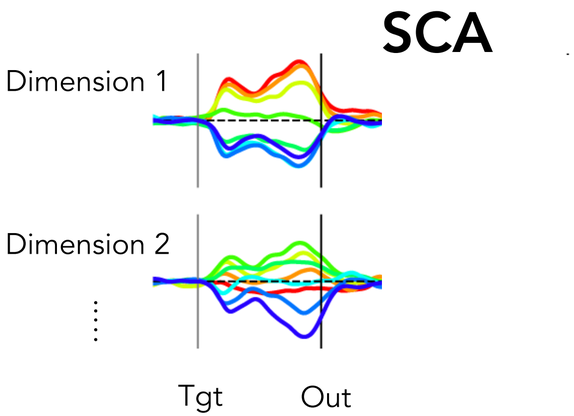


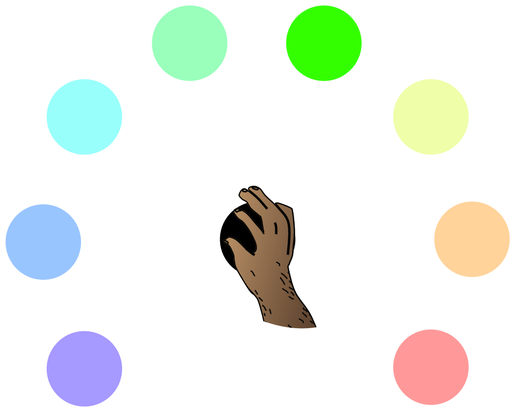
SCA finds more sparse/interpretable dimensions



500 ms

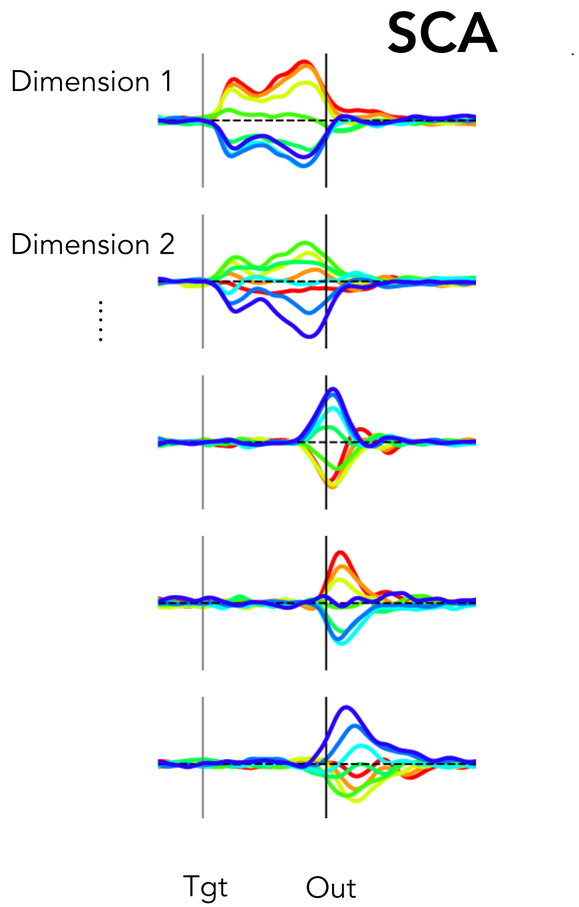
SCA finds more sparse/interpretable dimensions

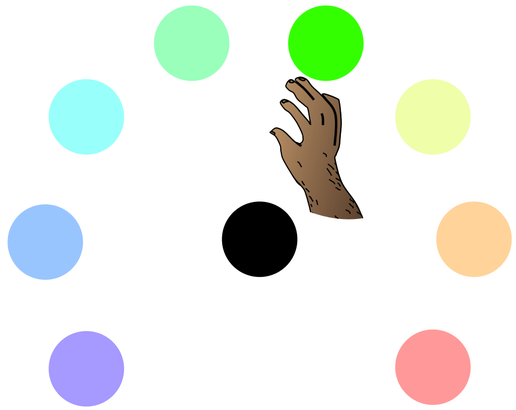




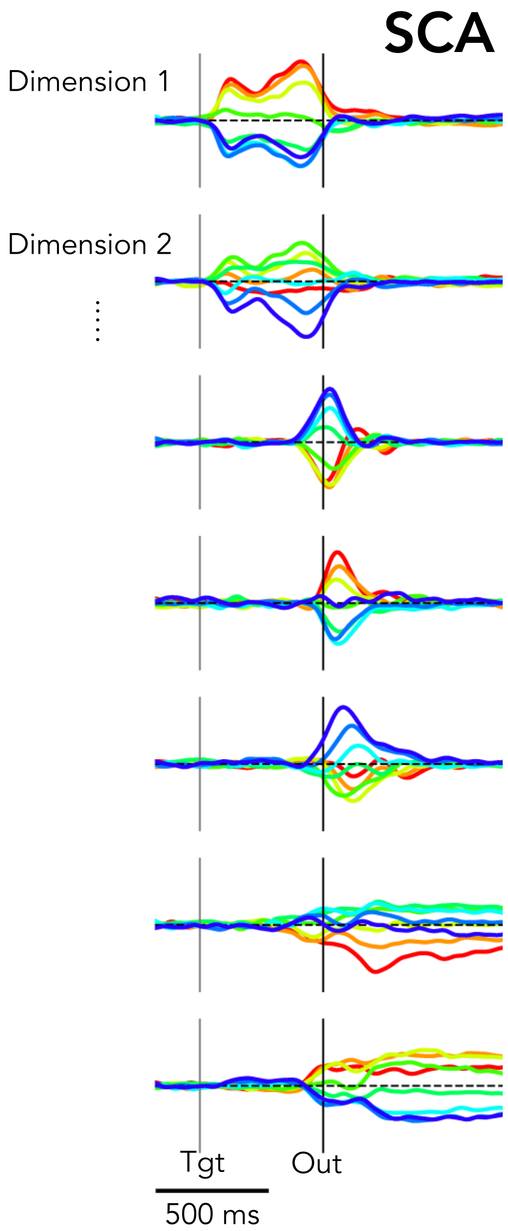
500 ms

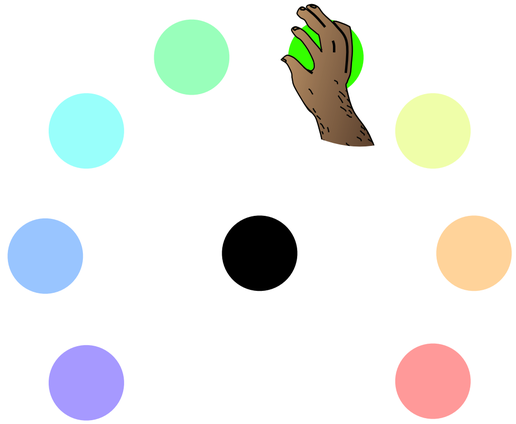
SCA finds more sparse/interpretable dimensions



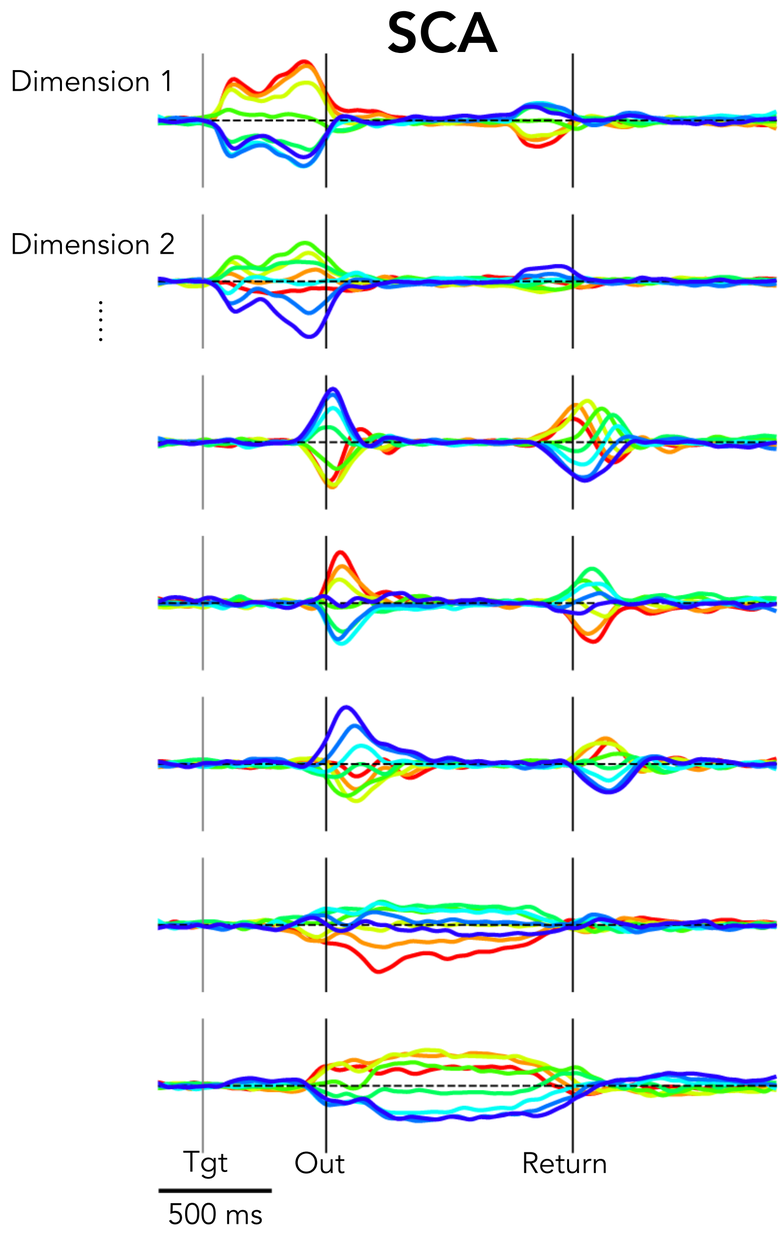


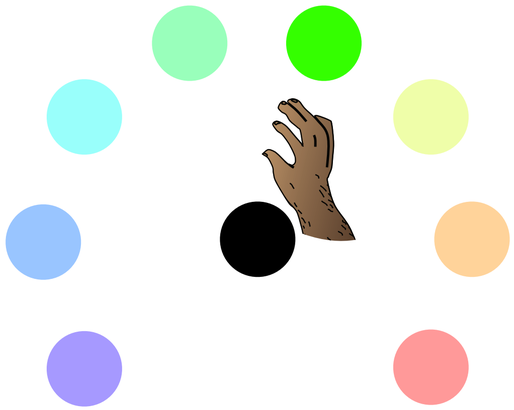
SCA finds more sparse/interpretable dimensions



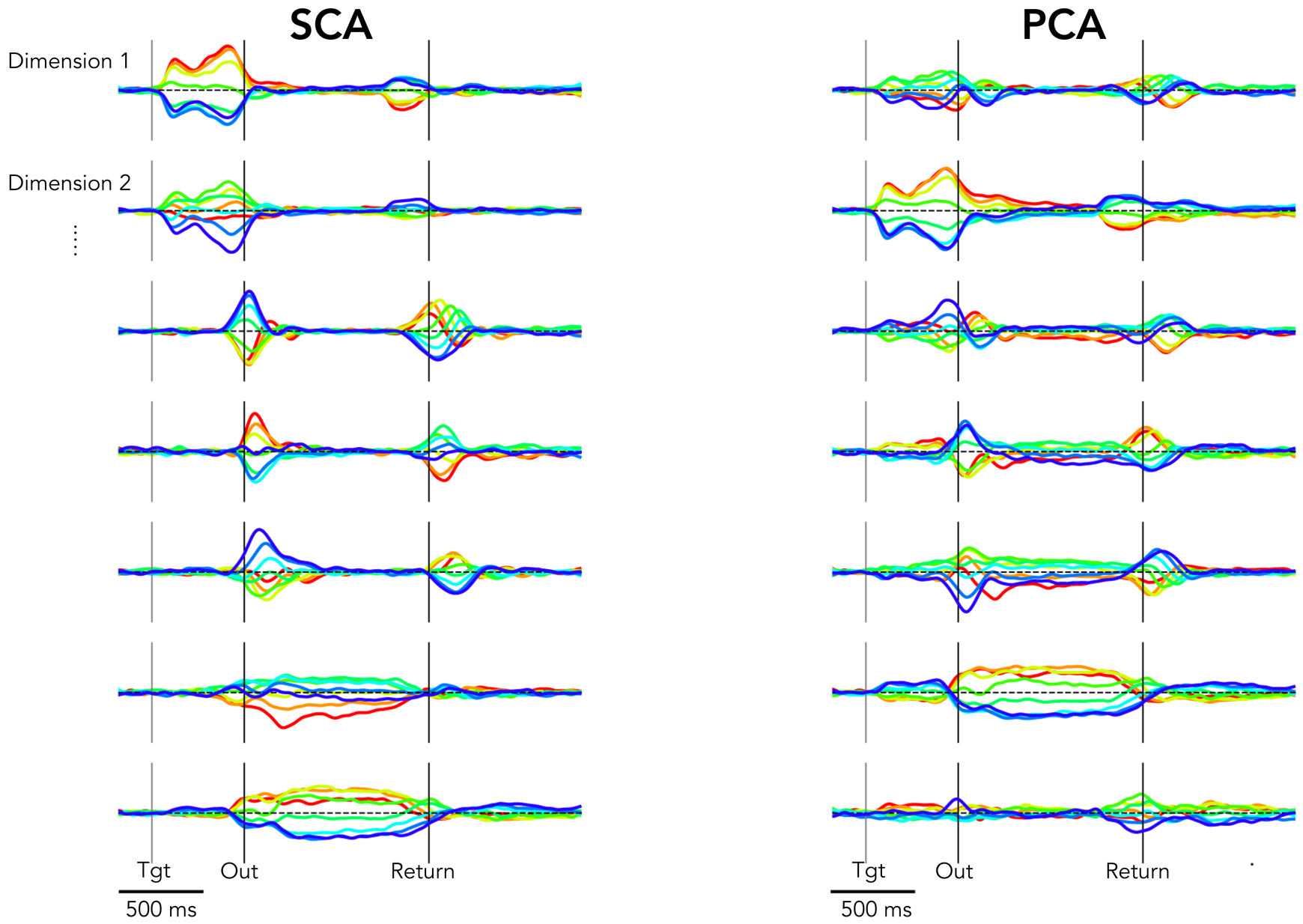


SCA finds more sparse/interpretable dimensions





SCA finds more sparse/interpretable dimensions



Summary

- Find more interpretable dimensions in an unsupervised way!
- Code available at github.com/glaserlab/sca
- Preprint available on *bioRxiv*

Resources

- Probabilistic Machine Learning Book 2, Chaps. 7,10, 29
  - https://probml.github.io/pml-book/book2.html
- Good blog post on cross-validation:
  - https://alexhwilliams.info/itsneuronalblog/2018/02/26/crossval/