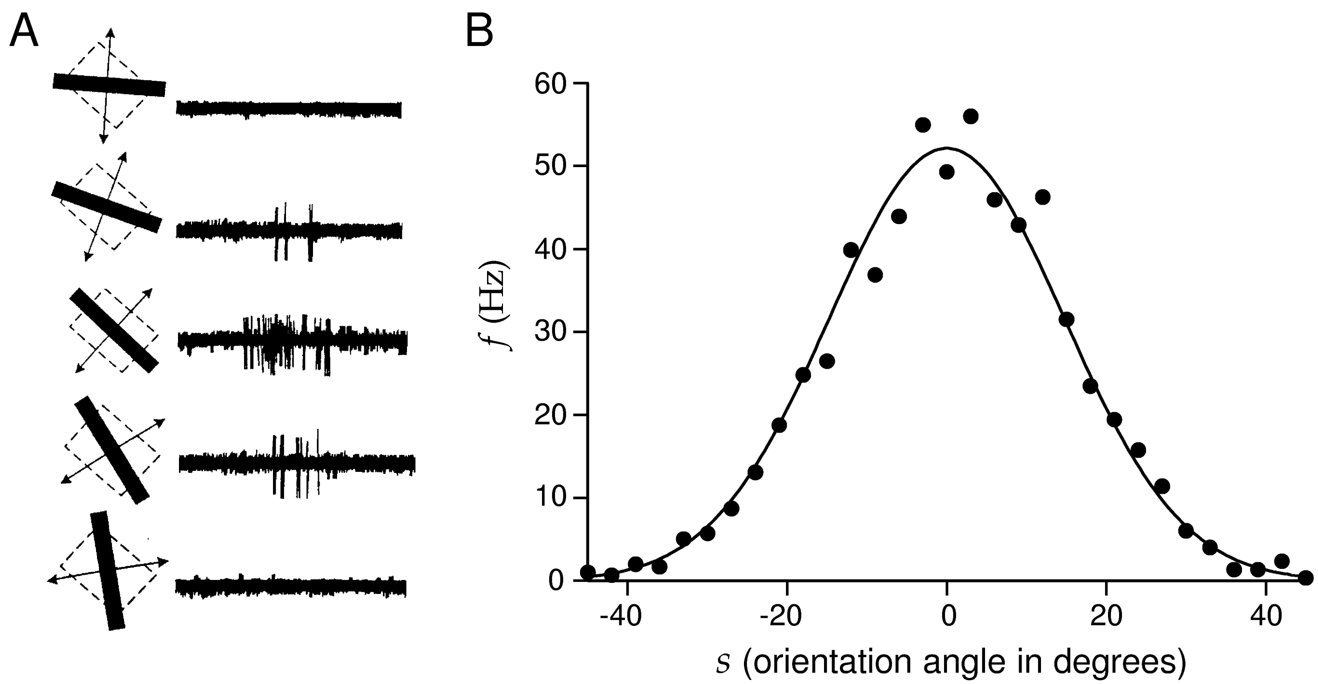
Keeping one's eye on the ball: Inferring latent dynamical state from ensemble neuronal activity

Maneesh Sahani

Gatsby Computational Neuroscience Unit, UCL

Tuning



A
B
60
50
40
30
20
10
0
-40
-20
0
20
40
f (Hz)
s (orientation angle in degrees)

Time-varying stimuli

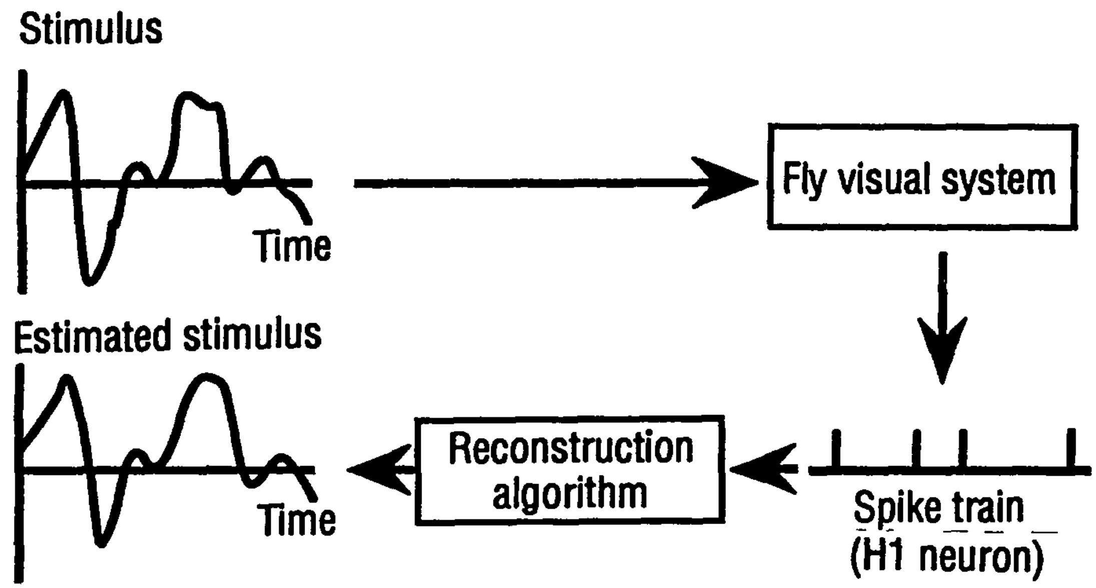
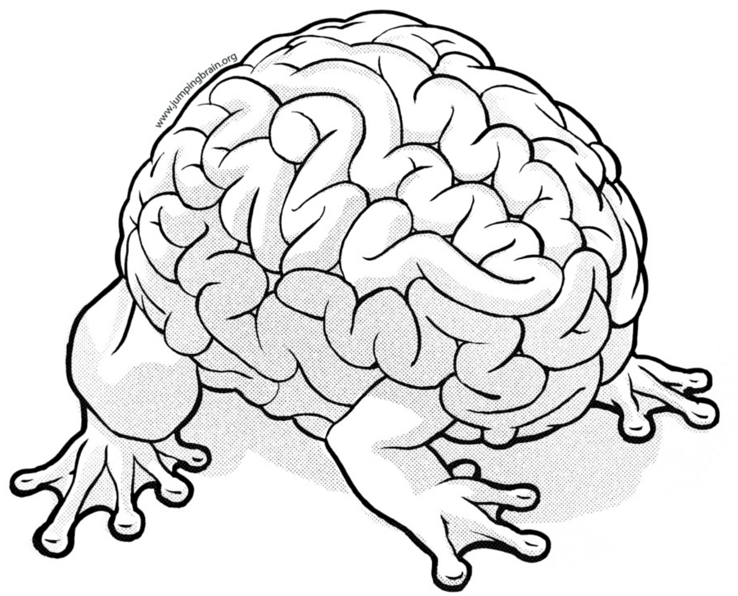


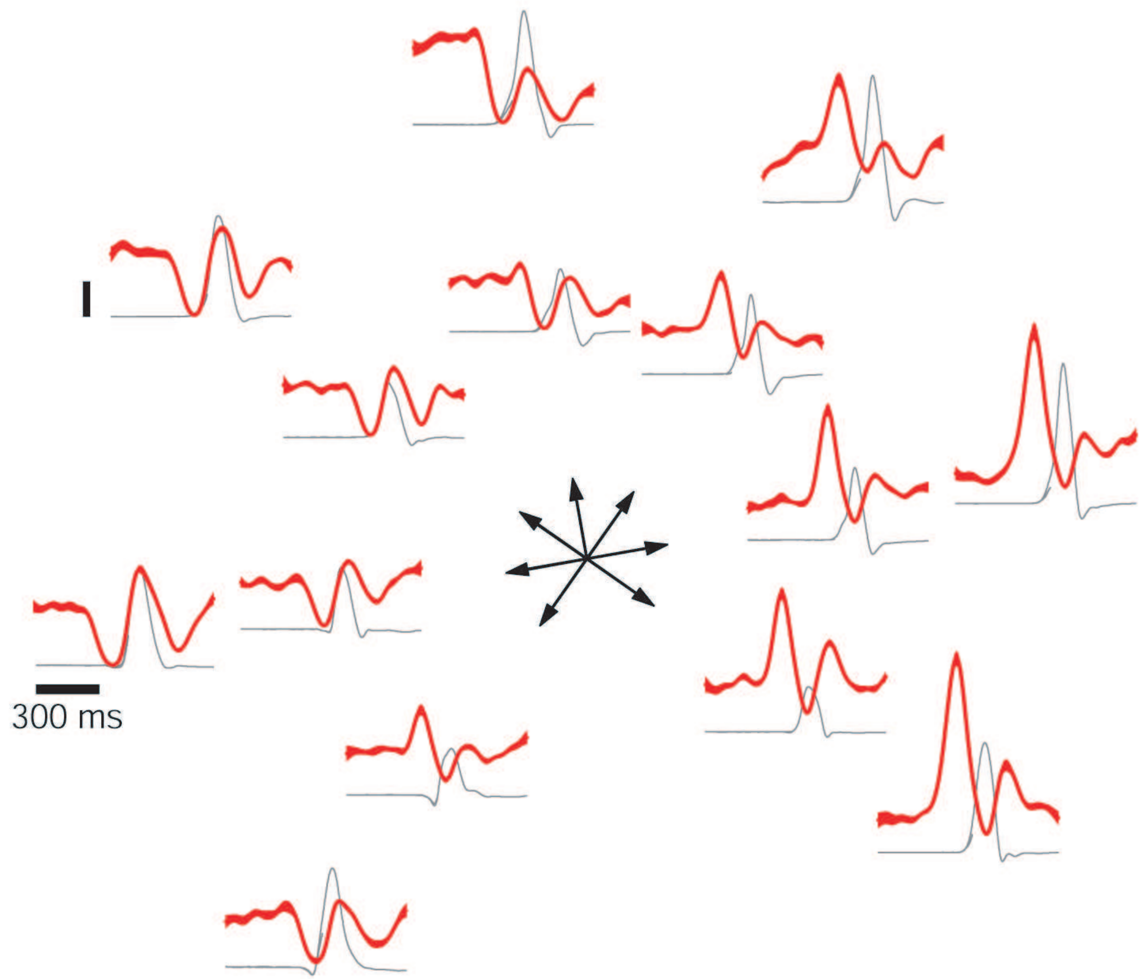
Fig. 1. Schematic view of the decoding process. The "black box" filters the spike train input $\{t_i\}$ to produce an estimate $s_{est}(\tau)$ of the stimulus.

Bialek *et al. Science* 1991

But this ignores the dynamic brain

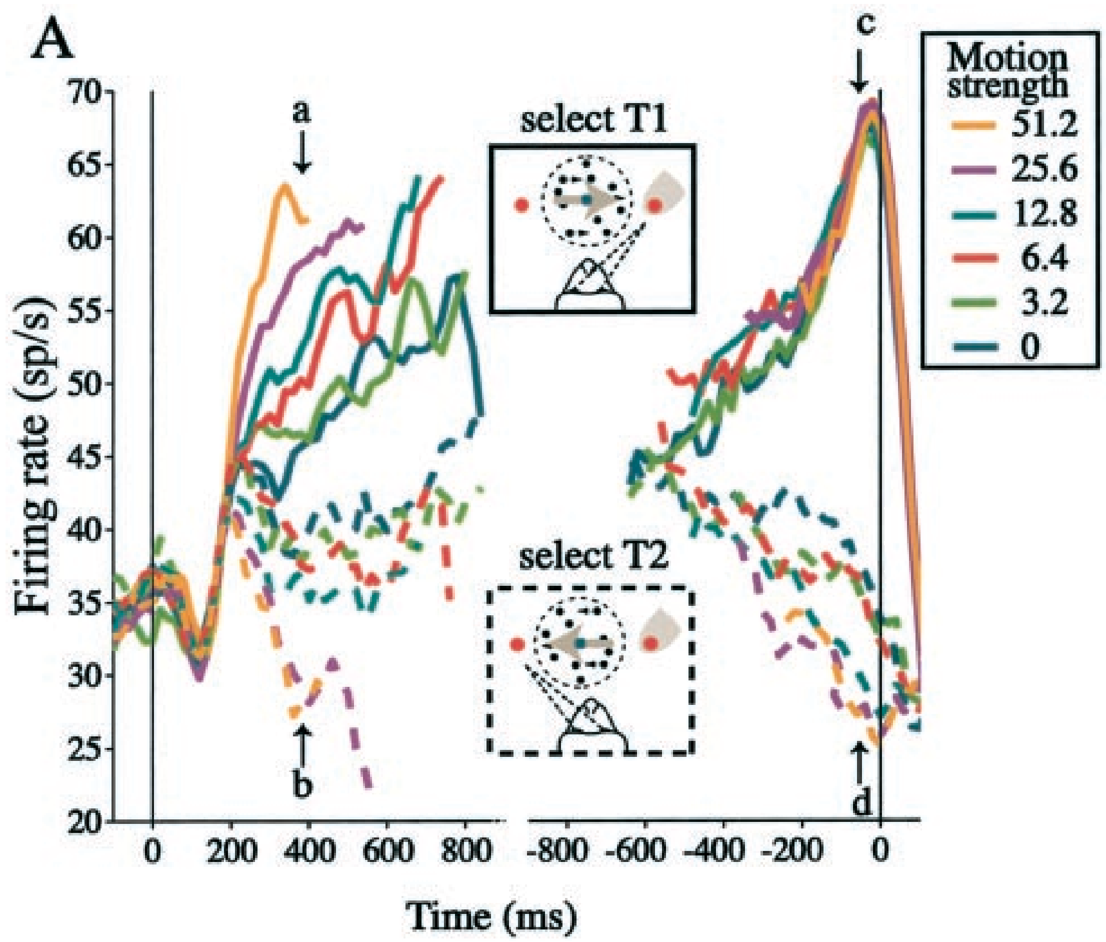


Motor dynamics



- Motor control signal must be internally generated
- Activity extends for longer than the movement

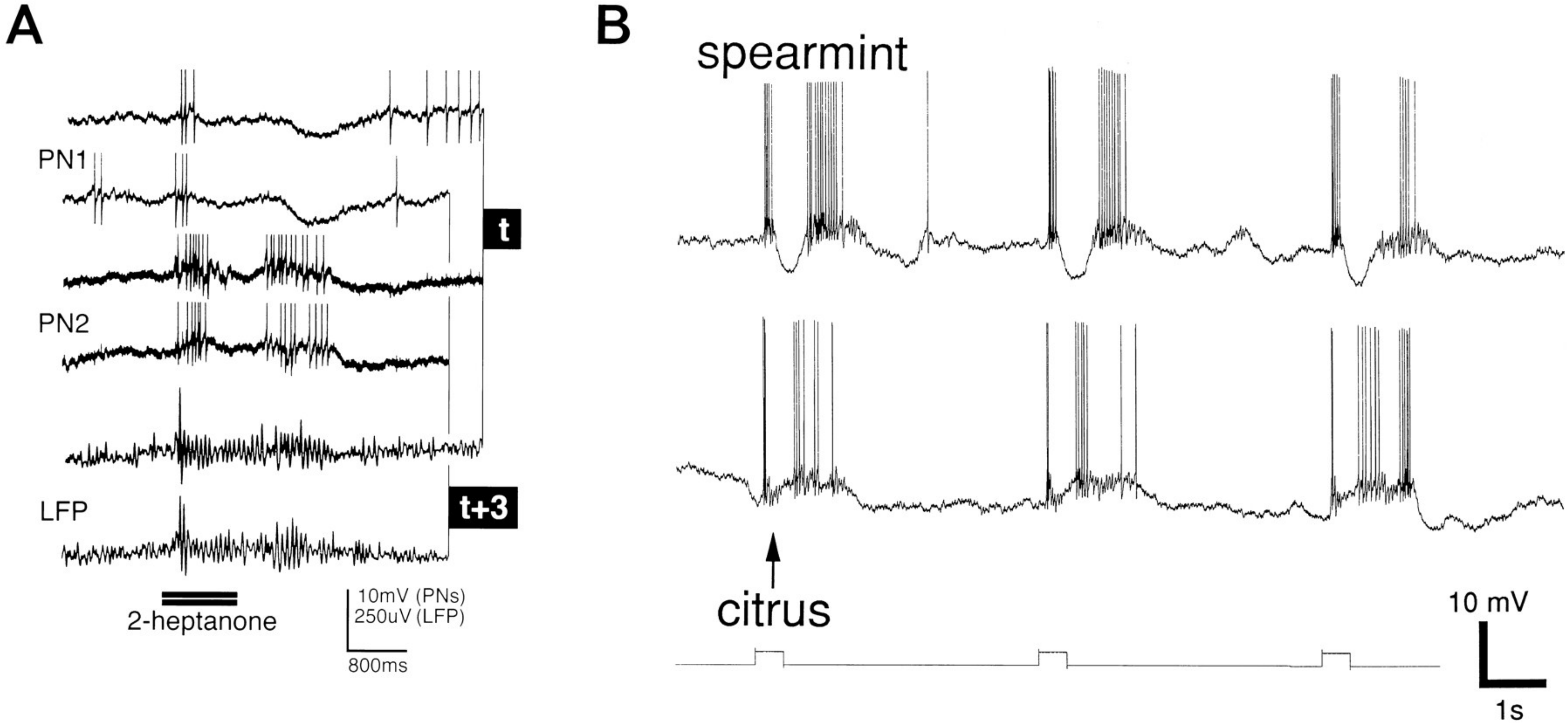
Cognitive dynamics



Roitman & Shadlen *J Neurosci* 2002

- decision making
- action sequencing
- motor planning

Sensory dynamics



Laurent *et al. J Neurosci* 1996

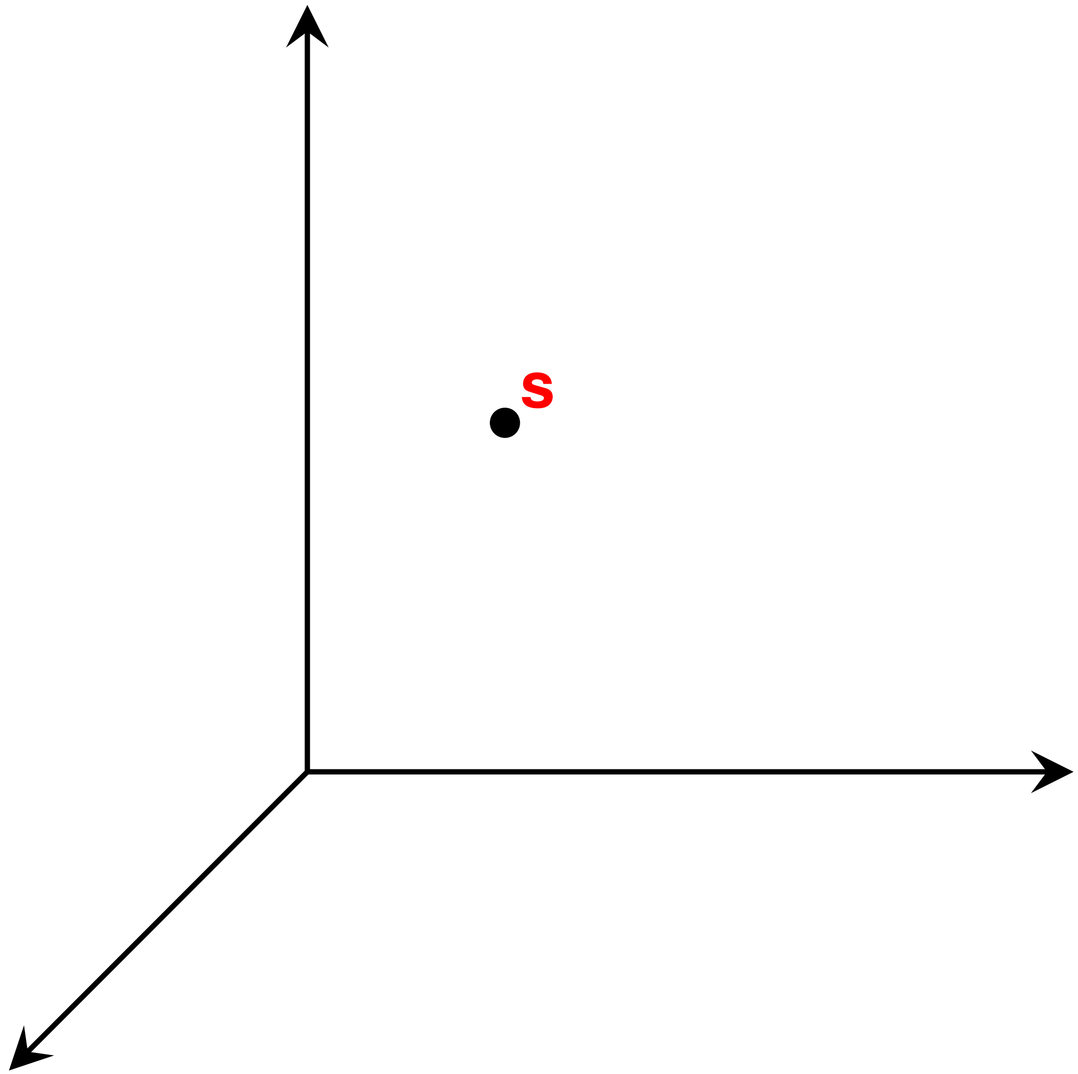
- coding (reservoir computing)
- inferential or perceptual computation?

Can we access these dynamics empirically?

Characterising a dynamical system

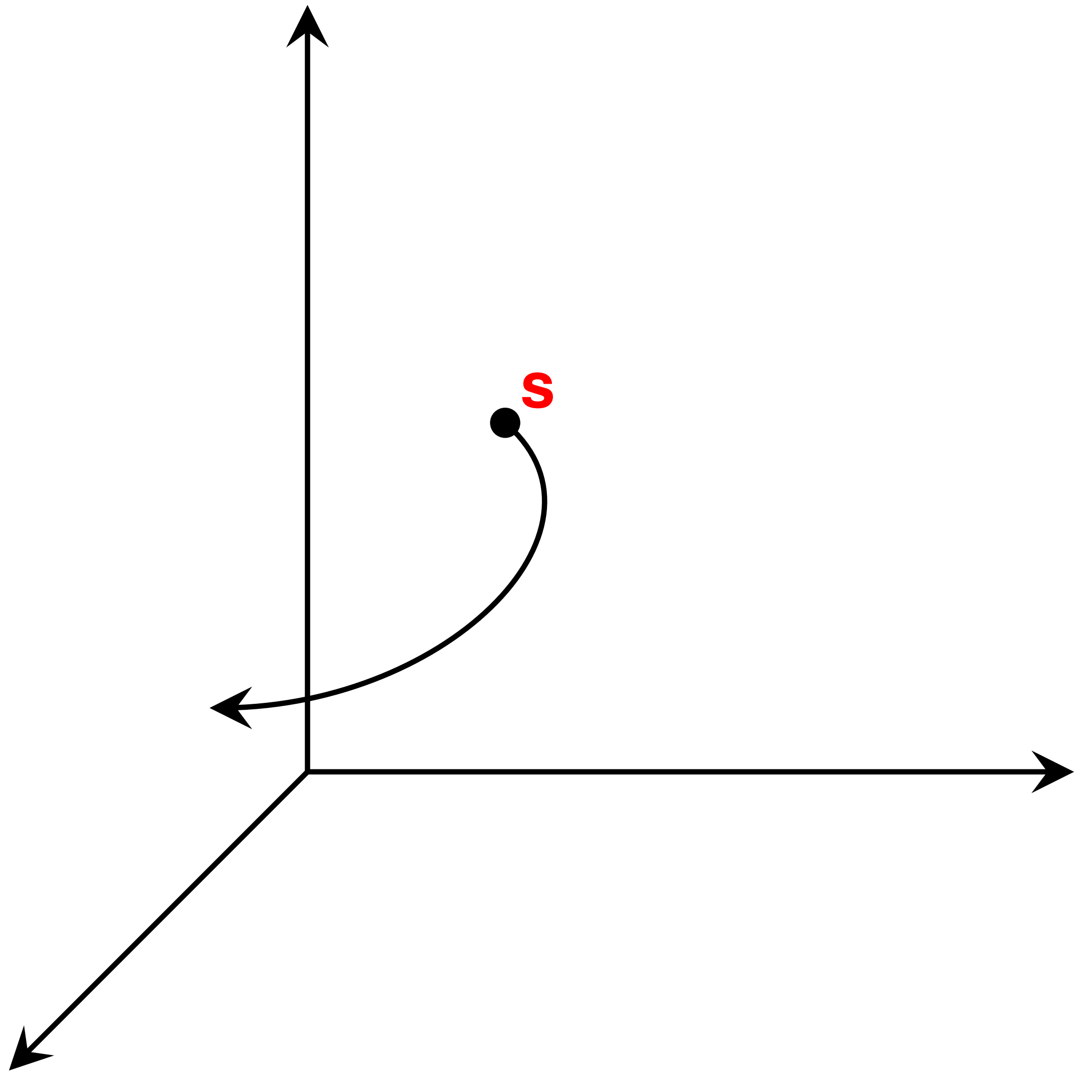
Characterising a dynamical system

- dynamical state



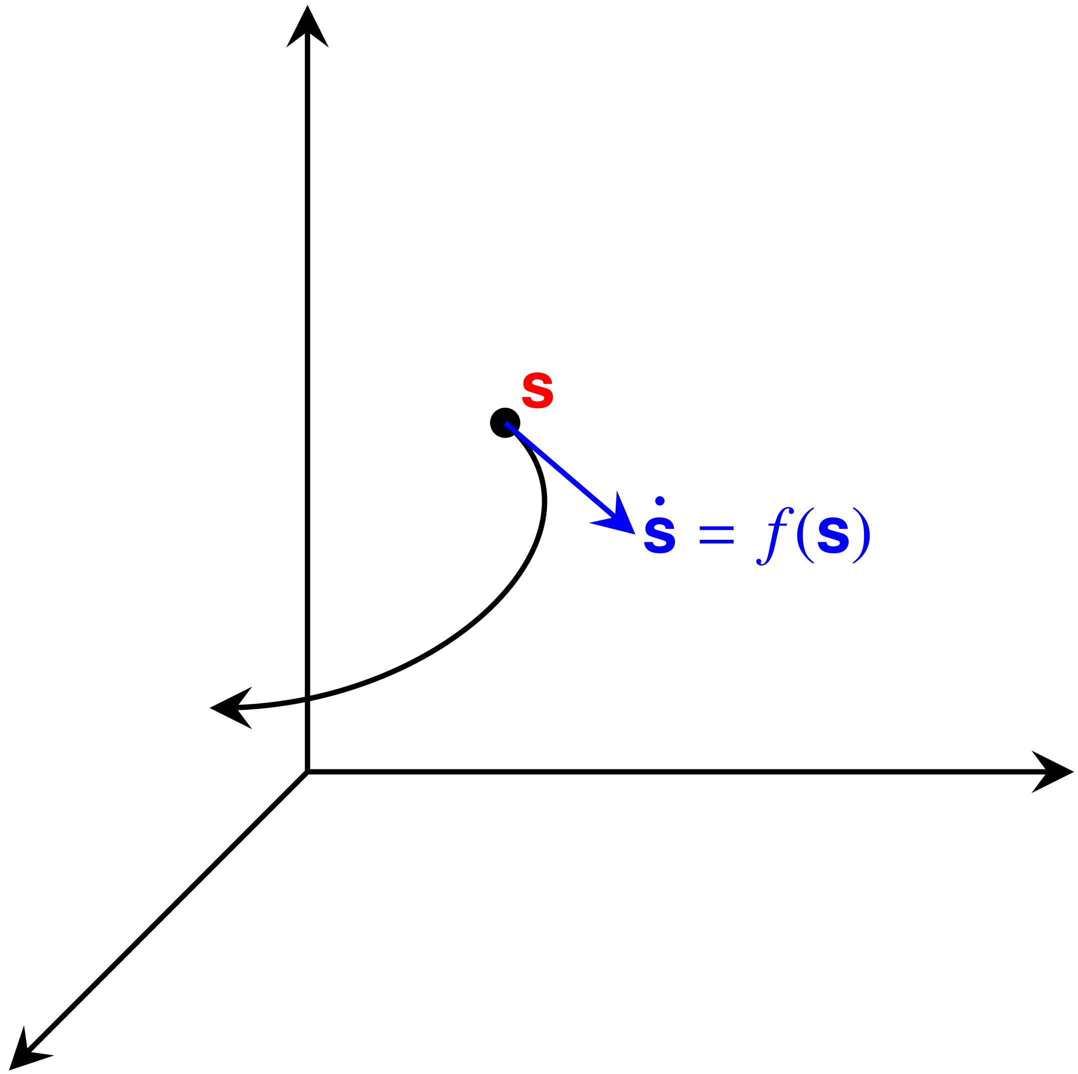
Characterising a dynamical system

- dynamical state



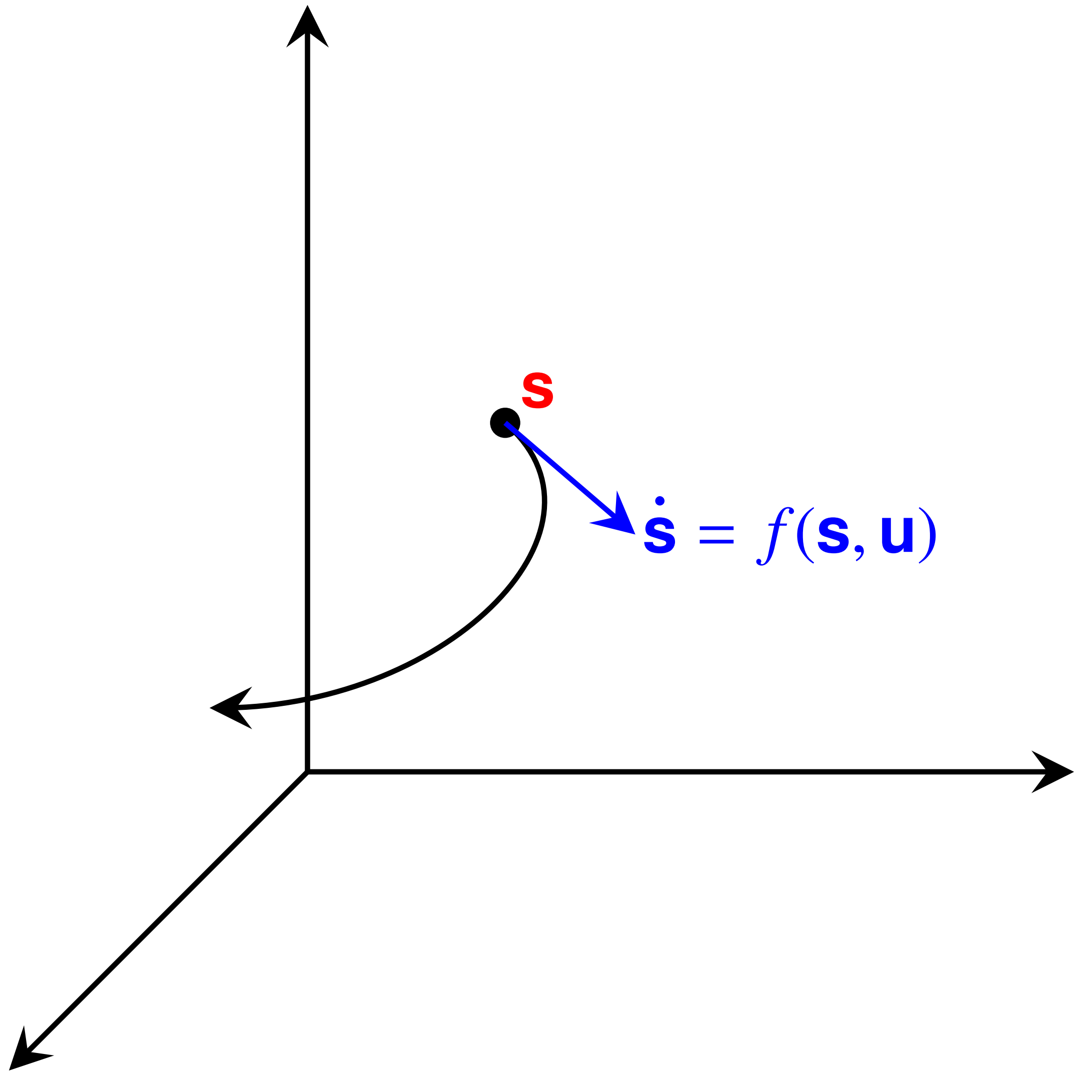
Characterising a dynamical system

- dynamical state
- equations of motion



Characterising a dynamical system

- dynamical state
- equations of motion



But . . .

But . . .

- What is the state?

- What are the equations of motion?

But . . .

- What is the state?
  - spike times of every neuron?

- What are the equations of motion?
  - integrate-and-fire or similar?

But . . .

- What is the state?
  - spike times of every neuron?
  - membrane, channel & ion state of every neuron?
- What are the equations of motion?
  - integrate-and-fire or similar?
  - biophysics & biochemistry?

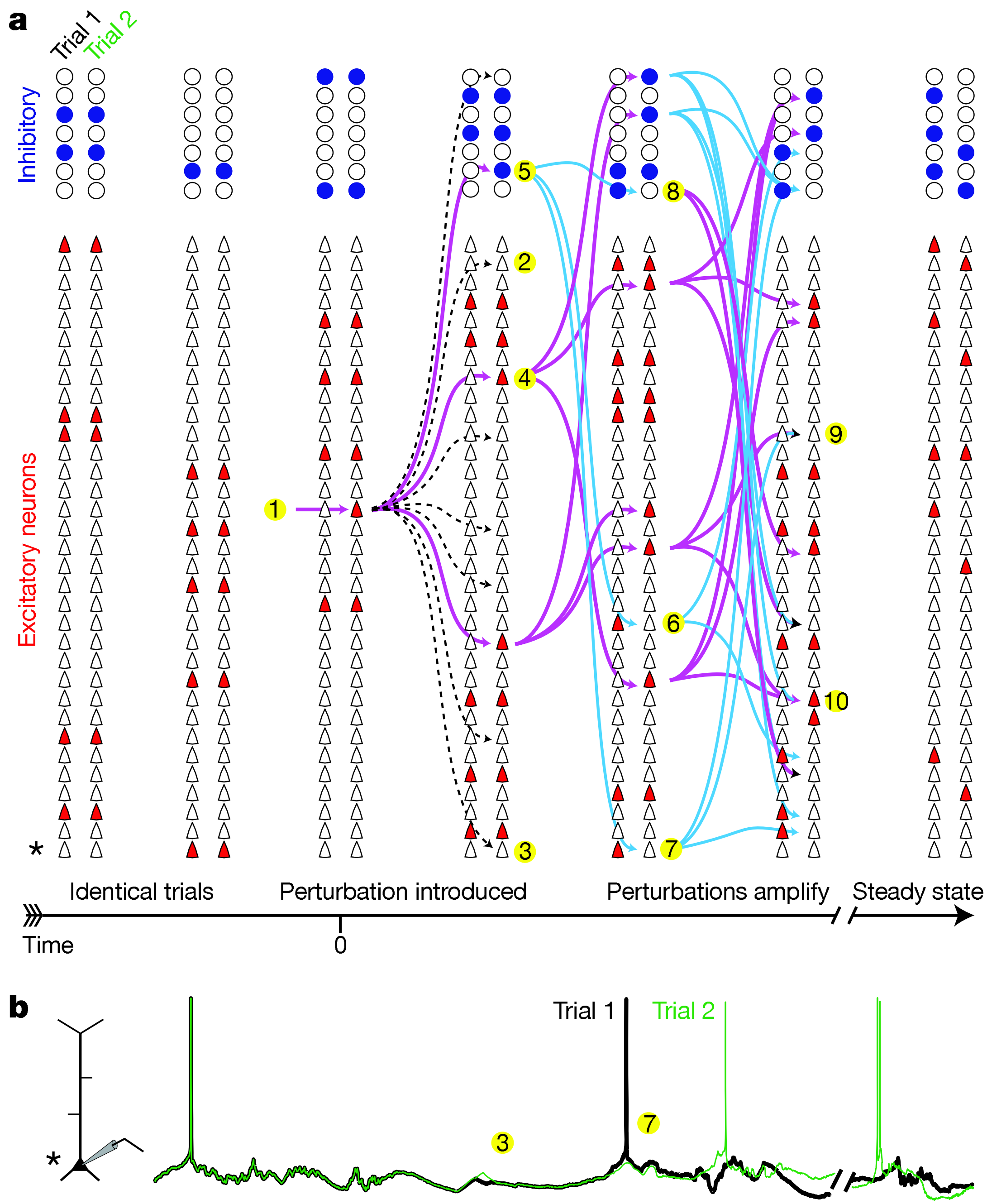
But . . .

- What is the state?
  - spike times of every neuron?
  - membrane, channel & ion state of every neuron?
- What are the equations of motion?
  - integrate-and-fire or similar?
  - biophysics & biochemistry?

– robustness to noise (kT is large)

– stable computations on a divergent substrate

Divergence?



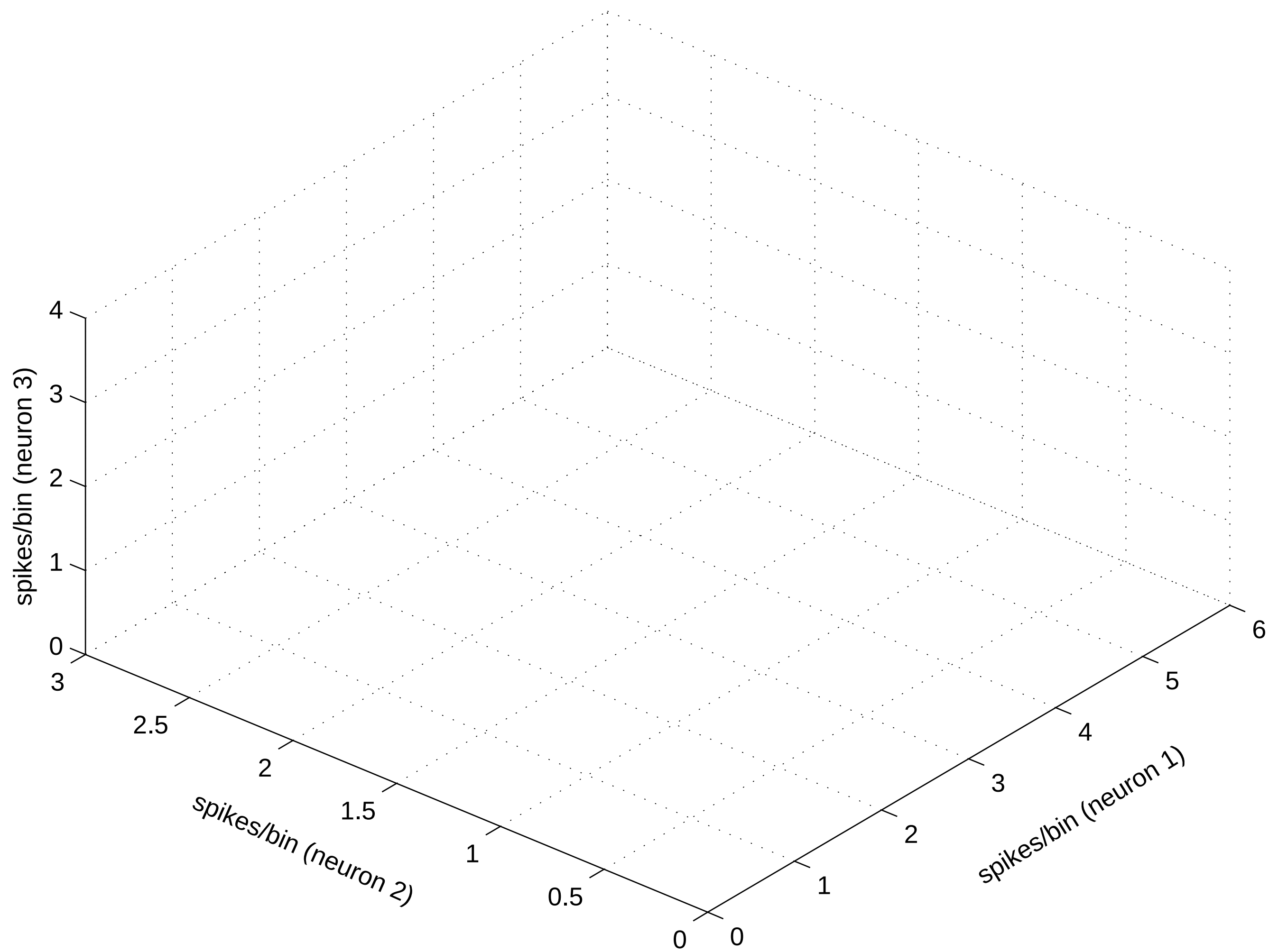
London *et al.* *Nature* 2010

A hope

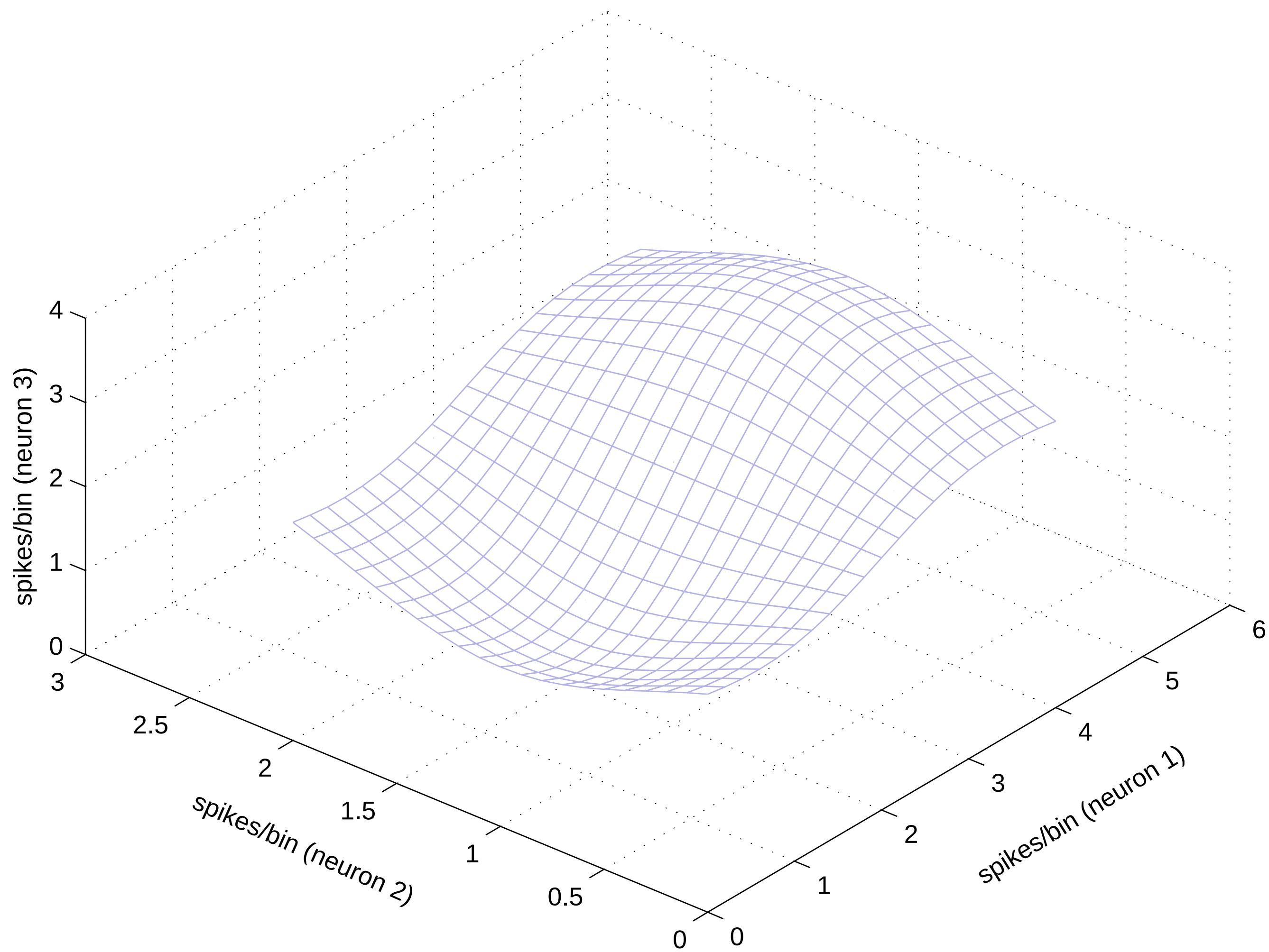
Computational state (order parameters) may be lower dimensional, with simpler dynamics

Latent spaces

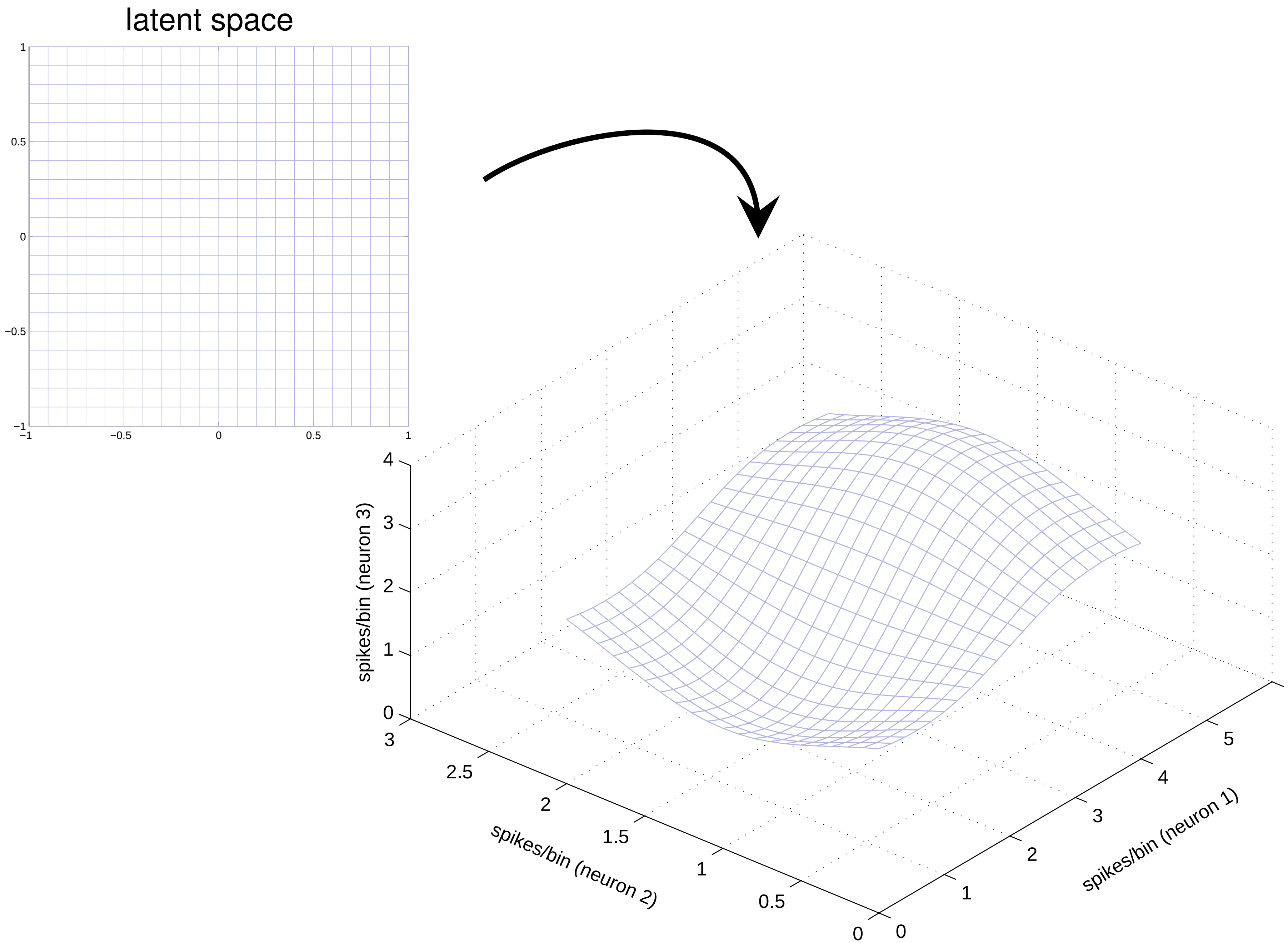
Latent spaces



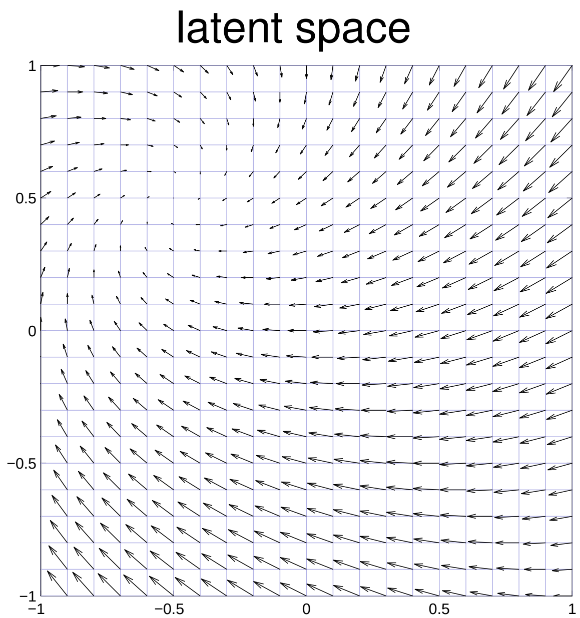
Latent spaces

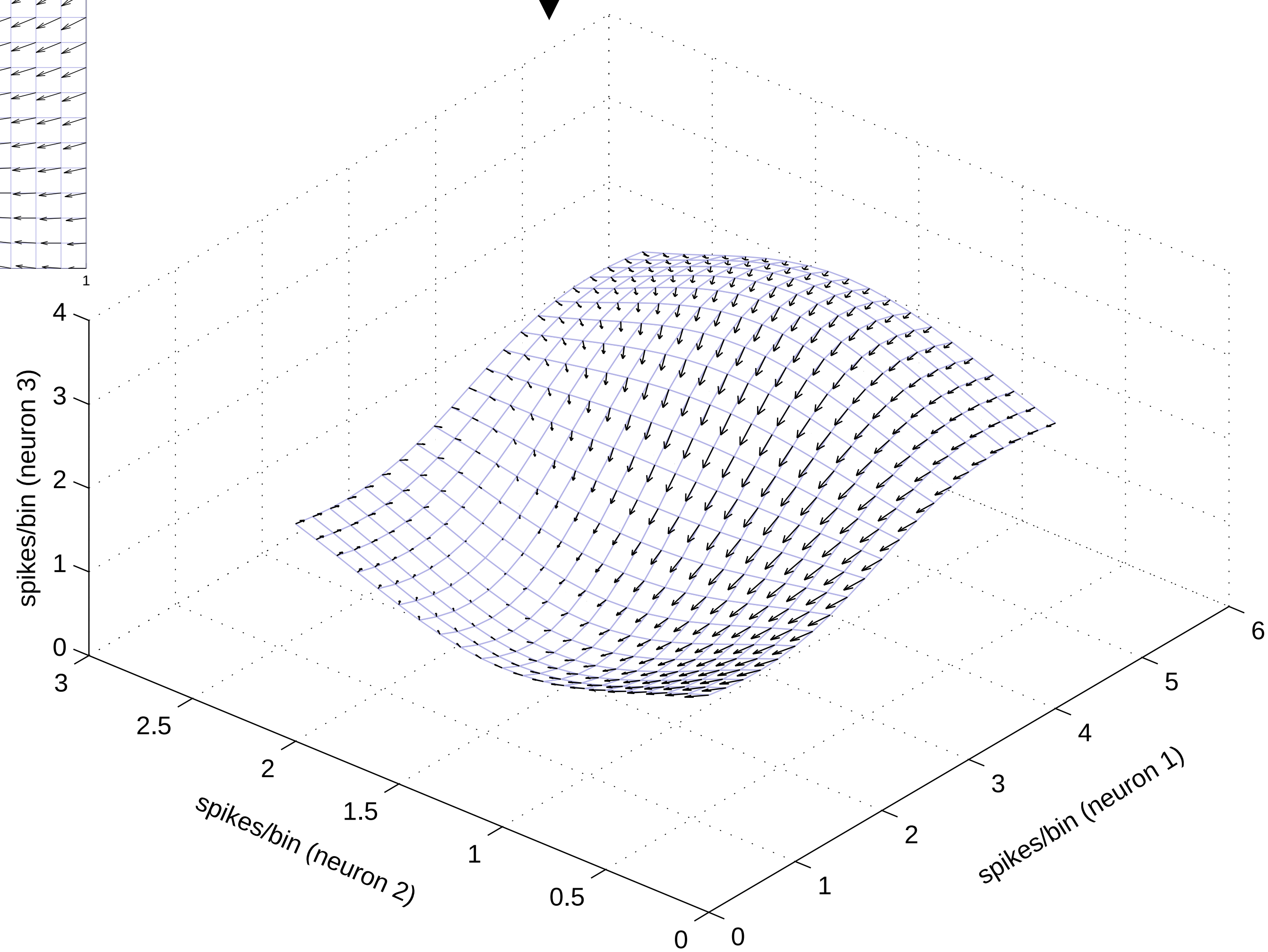


Latent spaces

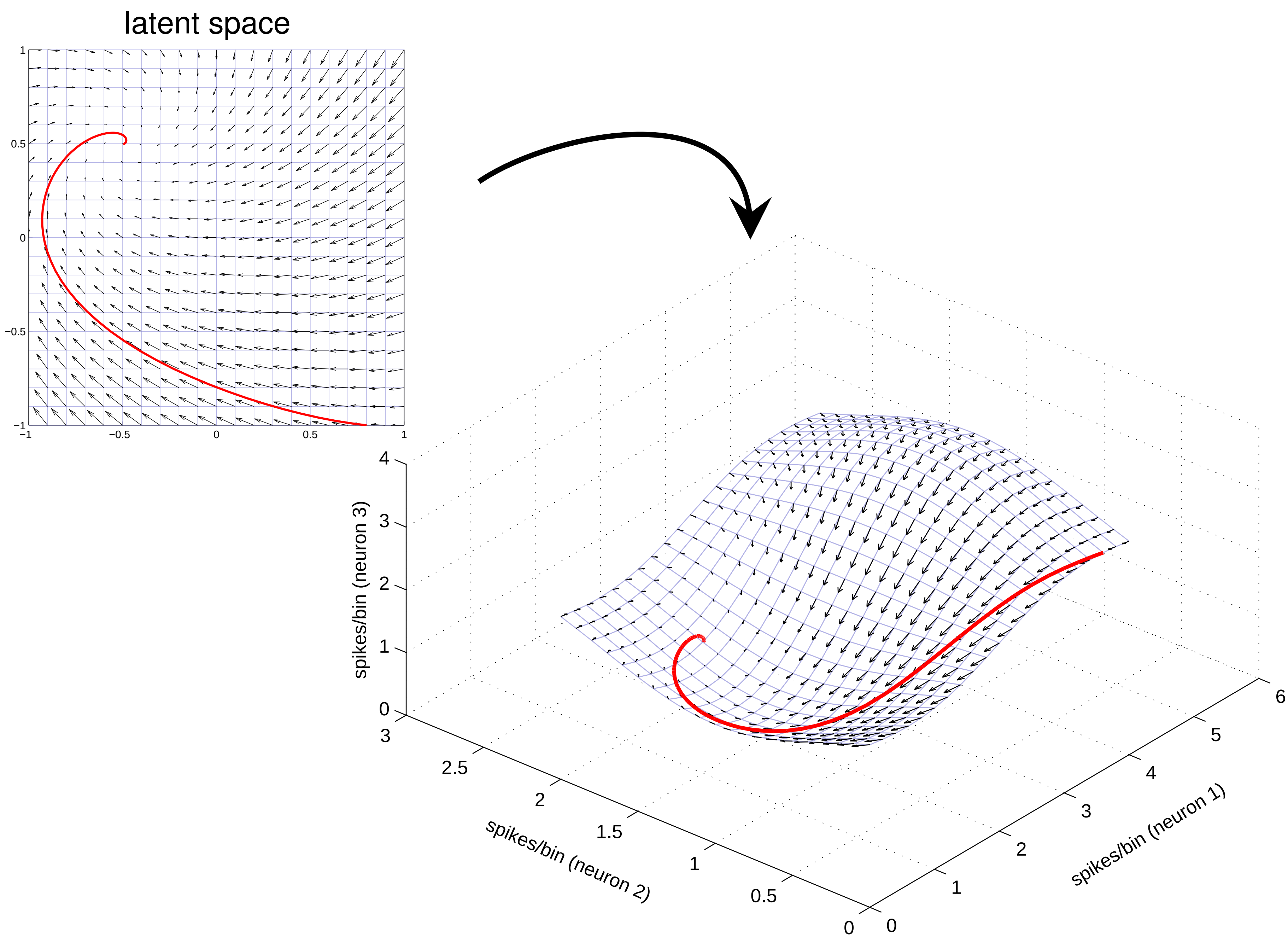


Latent spaces

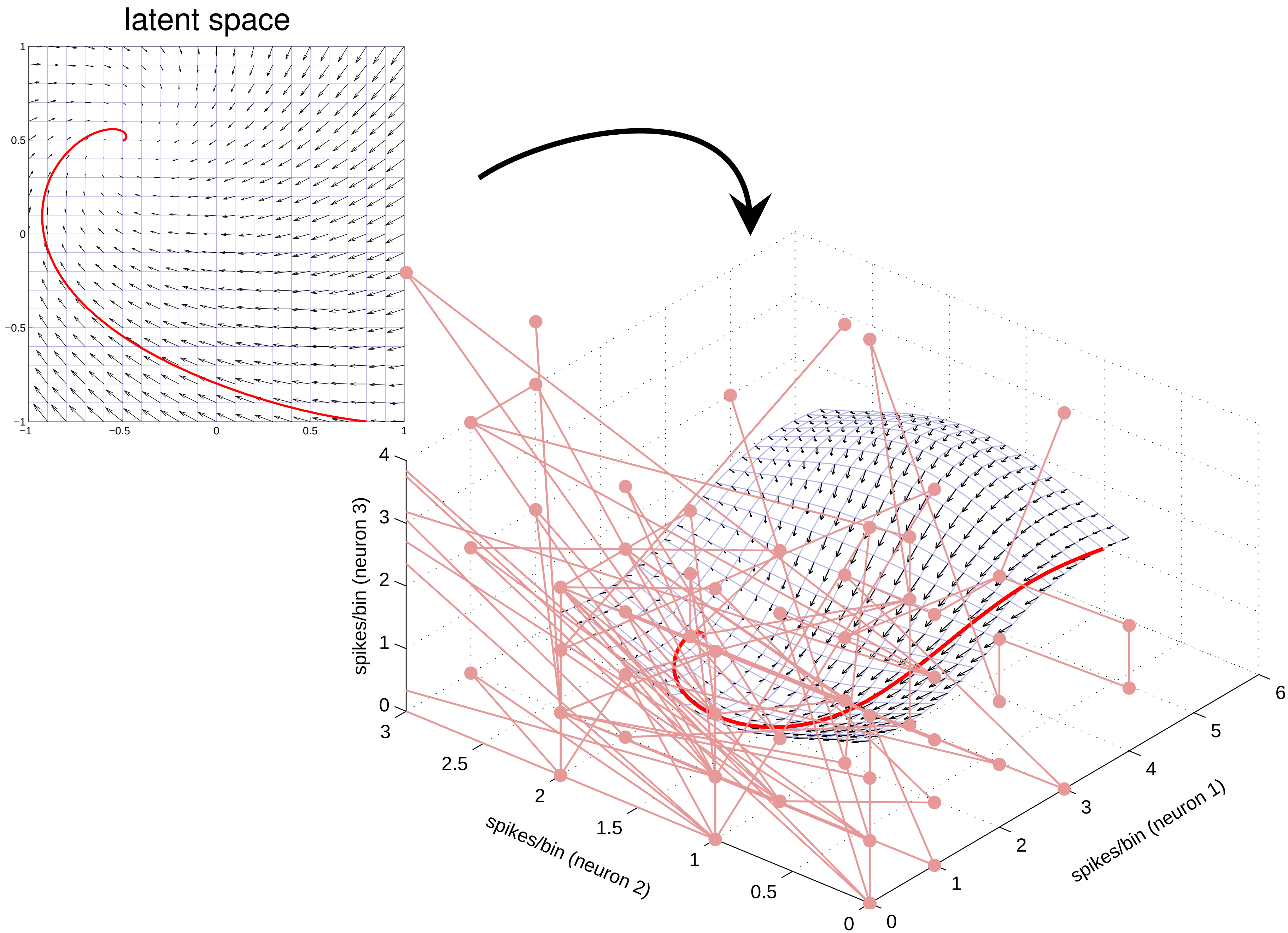




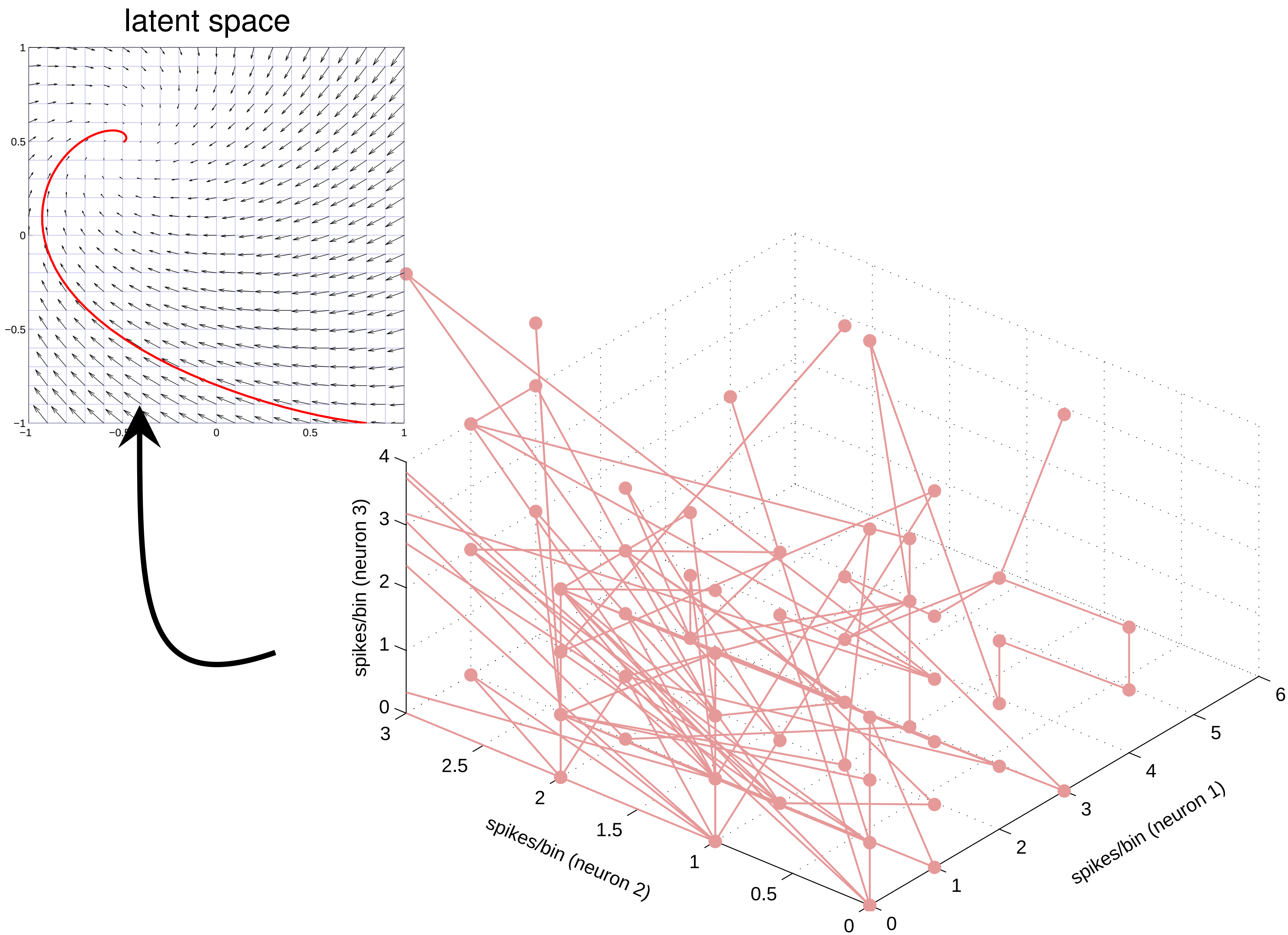
Latent spaces



Latent spaces



Latent spaces



What are we faced with?

Imagine trying to learn the laws of inertia from video of a ball . . .

What are we faced with?

Imagine trying to learn the laws of inertia from video of a ball . . .

– with a (very!) noisy camera

What are we faced with?

Imagine trying to learn the laws of inertia from video of a ball . . .

– with a (very!) noisy camera

– with only 100 working pixels

What are we faced with?

Imagine trying to learn the laws of inertia from video of a ball . . .

– with a (very!) noisy camera

– with only 100 working pixels

– in random places

What are we faced with?

Imagine trying to learn the laws of inertia from video of a ball . . .

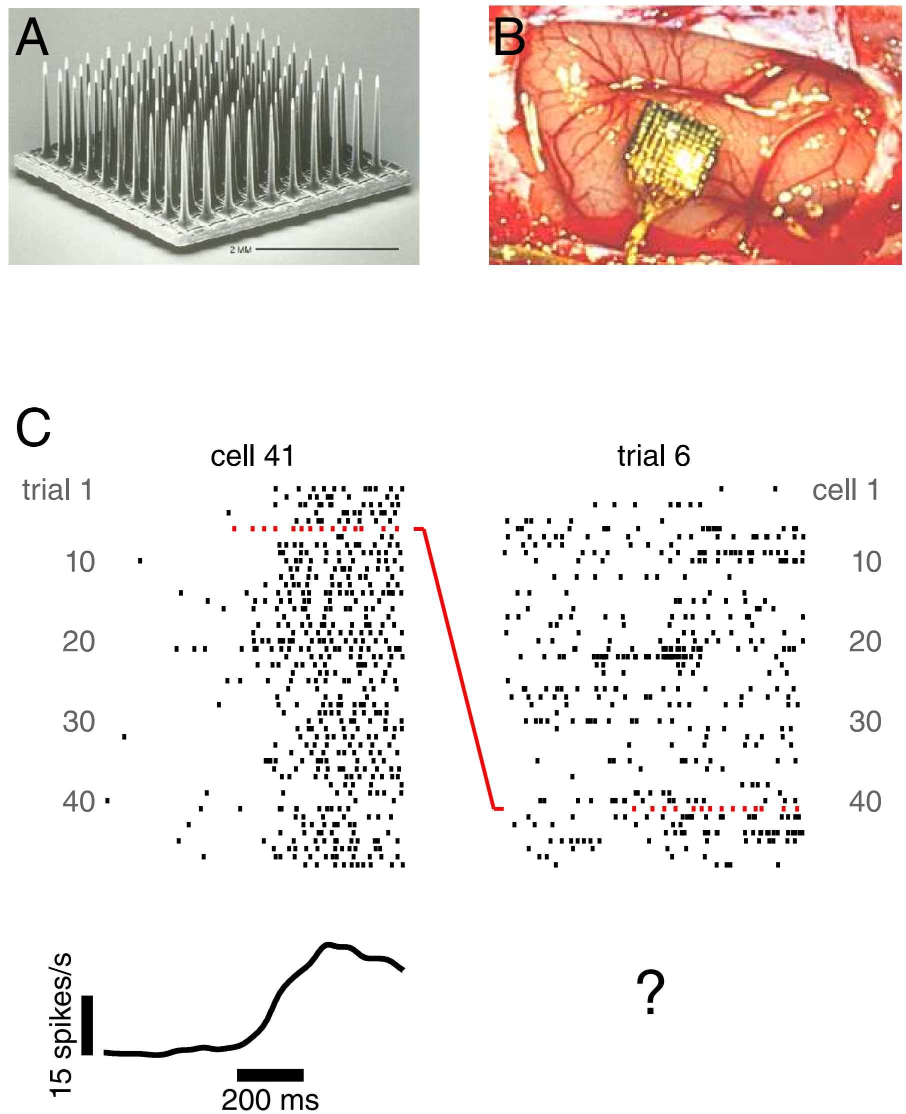
– with a (very!) noisy camera

– with only 100 working pixels

– in random places

– without location labels

The new PSTH?



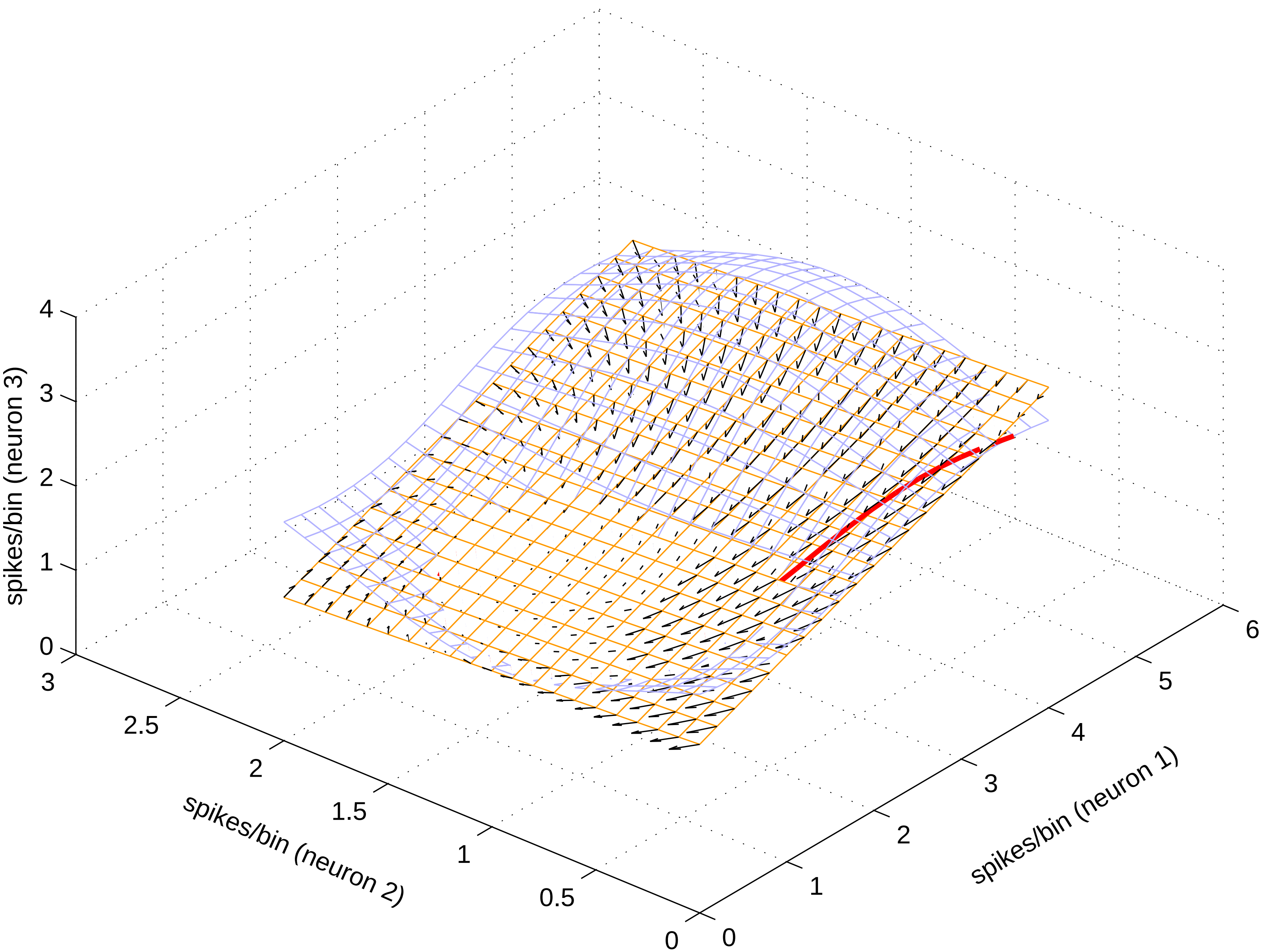
Churchland *et al. Curr Opinions Neurobiol* 2008

Methods

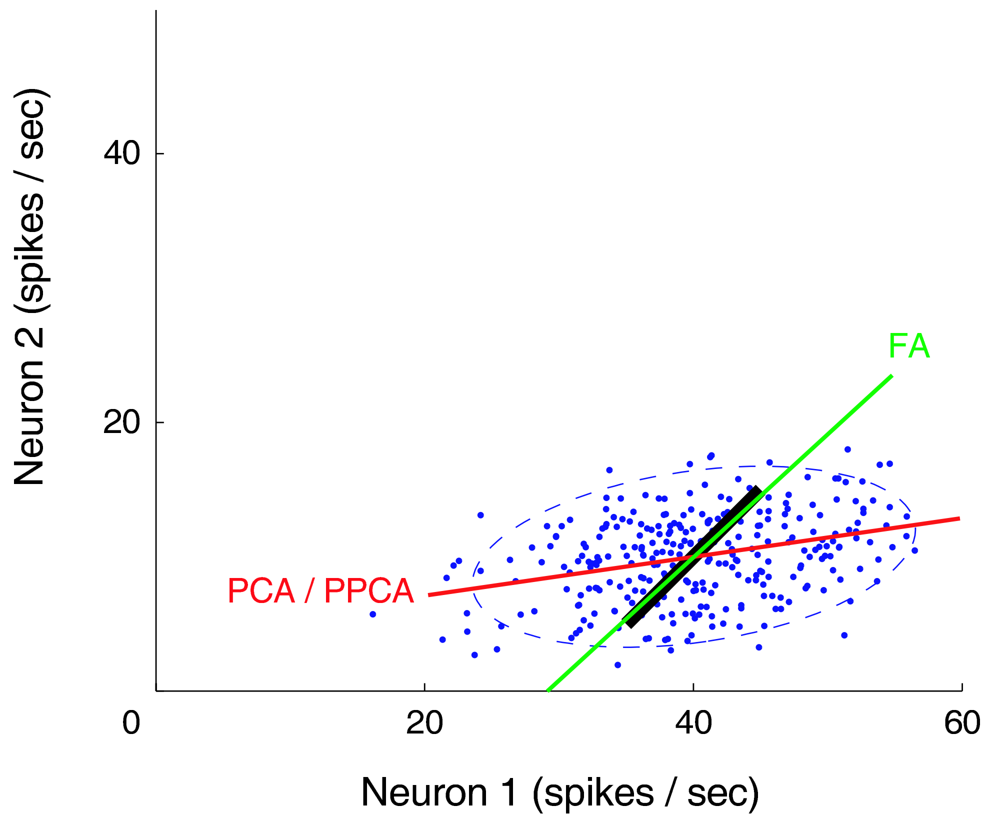
The necessary technology can be found in both old and new algorithms from computational statistics and machine learning:

- PCA and Factor Analysis
- Non-linear dimensionality reduction (ISOMAP, LLE, MVU . . .).
- GPFA (Yu *et al*. *NIPS*, *J Neurophysiol*, 2009)
- Hidden (non)linear dynamics ("Kalman filters") for point processes (Smith and Brown *Neural Comput* 2003; Yu *et al*. *NIPS, NSSPW* 2006; Macke *et al*. *NIPS* 2011)
- Hidden switching LDS ("switching Kalman filters") (Petreska et al. *NIPS* 2011)

Linear subspaces



PCA vs Factor Analysis



Gaussian Process Methods

Latent prior

$$\mathbf{x}(t) \sim \mathcal{GP}\left[\boldsymbol{\mu}(t); \mathsf{K}_{\boldsymbol{\theta}}(t, t')\right] \qquad \text{state model}$$
$$\mathbf{y}(t) \sim \textit{Dist}\left[f(\mathbf{x}(t))\right] \qquad \text{observation model}$$

$\mathcal{GP}$ is a Gaussian process: this implies that any finite set of measurements at fixed times is jointly normal.

- Includes linear-Gaussian dynamical systems (LDS).

$$\mathbf{x}_t \sim \mathcal{N}\left(A\mathbf{x}_{t-1}, Q\right)$$

- Allows generalisation to non-(first-order-)Markov systems.

Gaussian process dynamics

$$\mathbf{x}(t) \sim \mathcal{GP}\left[\mu(t); \mathrm{K}_{\boldsymbol{\theta}}(t, t')\right]$$
$$\mathbf{y}(t) \sim \mathit{Dist}\left[f(\mathbf{x}(t))\right]$$

- $K_{\boldsymbol{\theta}}(t, t')$ gives the covariance between values of $\mathbf{x}(t)$ and $\mathbf{x}(t')$.
- Parameterised by covariance. LDS (or auto-regressive models) are parameterised by precision (inverse covariance).
- Easier to specify priors wih interesting properties:
  - LDS: $\mathrm{K}(t, t') \propto a^{|t-t'|}$
  - Smooth: $\mathrm{K}(t, t') \propto \exp((t-t')^2/2\lambda)$
  - Oscillatory: $\mathrm{K}(t, t') \propto \sin(2\pi\omega(t-t'))$
  - Stationary "Brownian": $\mathrm{K}(t, t') \propto [1 - |t-t'|/\lambda]^+$
- Inference naively $O(T^3)$ instead of $O(T)$.
  - Numerical methods based on regularities in matrices.
  - Sparsifying methods select (or create) subset of data with similar predictive power.

Link functions

$$\mathbf{x}(t) \sim \mathcal{GP}\left[\boldsymbol{\mu}(t); \mathsf{K}_{\boldsymbol{\theta}}(t, t')\right]$$
$$\mathbf{y}(t) \sim \textit{Dist}\left[f(\mathbf{x}(t))\right]$$

f maps the latent GP values to (mean) intensity.

- Nonlinear
  - Exponential – danger: emphasises variability at high values.
  - Threshold-linear or soft-threshold.
- Linear
  - Requires observation model tolerant of negative values.
  - Alternatively, can use a truncated prior.
    * Requires approximation (but so does non-linearity).
    * Posterior often not far from Gaussian (multi-d truncation – draws are suprisingly smooth).
    * EP can be powerful approximation technique.

Observation models

$$\mathbf{x}(t) \sim \mathcal{GP}\left[\boldsymbol{\mu}(t); \mathsf{K}_{\boldsymbol{\theta}}(t, t')\right]$$
$$\mathbf{y}(t) \sim \mathit{Dist}\left[f(\mathbf{x}(t))\right]$$

- Point process (continuous time)
  - Rescaled renewal process. (next)
  - Inhomogeneous Markov-interval.

$$\lambda(t) = f(\mathbf{x}(t), s_{last}) \qquad \left(\text{often } = f(\mathbf{x}(t)) \cdot h(s_{last})\right)$$

  - GLM-like sum.

$$\lambda(t) = f\left(\mathbf{x}(t) + \sum_i \alpha_i h(s_i)\right)$$

- Spike count (discrete time)
  - Poisson counts.
  - (Square-rooted) Gaussian counts.

Examples

- Example 1: GP-based intensity estimates

  Cunningham, Yu, Shenoy, and Sahani. Inferring neural firing rates from spike trains using Gaussian processes. In *Adv. Neural Info. Proc. Sys. 20*, Cambridge, MA, 2008. MIT Press.

  Cunningham, Shenoy, and Sahani. Fast Gaussian process methods for point process intensity estimation. In *ICML '08*, pp. 192–199, Helsinki Finland, 2008. Omni Press.

- Example 2: Gaussian process factor analysis

  Yu, Cunningham, Santhanam, Ryu, Shenoy, and Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *J. Neurophysiol.* 102: 614-635, 2009.

Example 1: GP-based intensity estimates

Spike train discretised in (arbitrarily small) time-bins.

$$\mathbf{x} \sim \mathcal{N}(\mu\mathbf{1}, \mathsf{K}_{\boldsymbol{\theta}})$$

$$p(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{N} \left[\frac{\gamma x_{y_i}}{\Gamma(\gamma)} \left(\gamma \sum_{k=y_{i-1}}^{y_i - 1} x_k \Delta \right)^{\gamma-1} \exp\left\{ -\gamma \sum_{k=y_{i-1}}^{y_i - 1} x_k \Delta \right\} \right]$$

- This is a Gamma-interval process likelihood

$$p(\tau) = \frac{\gamma^\gamma}{\Gamma(\gamma)} \tau^{\gamma-1} e^{-\gamma\tau}$$

with order γ and mean 1, with time rescaled according to GP rate.

Example 1: GP-based intensity estimates

Modal Inference:

$$\mathbf{x}^* = \operatorname*{argmax}_{\mathbf{x} \geq \mathbf{0}} p(\mathbf{x} \mid \mathbf{y}) = \operatorname*{argmax}_{\mathbf{x} \geq \mathbf{0}} p(\mathbf{y} \mid \mathbf{x}) p(\mathbf{x}).$$

- Note that the nonnegativity constraint eliminates need for a space warping link function (equivalent to truncated prior).
- Convex. Solve using a log barrier Newton Method.
- Computational complexity is a major challenge. We exploit problem structure to minimize run-time and memory requirements.

Example 1: GP-based intensity estimates

Learning

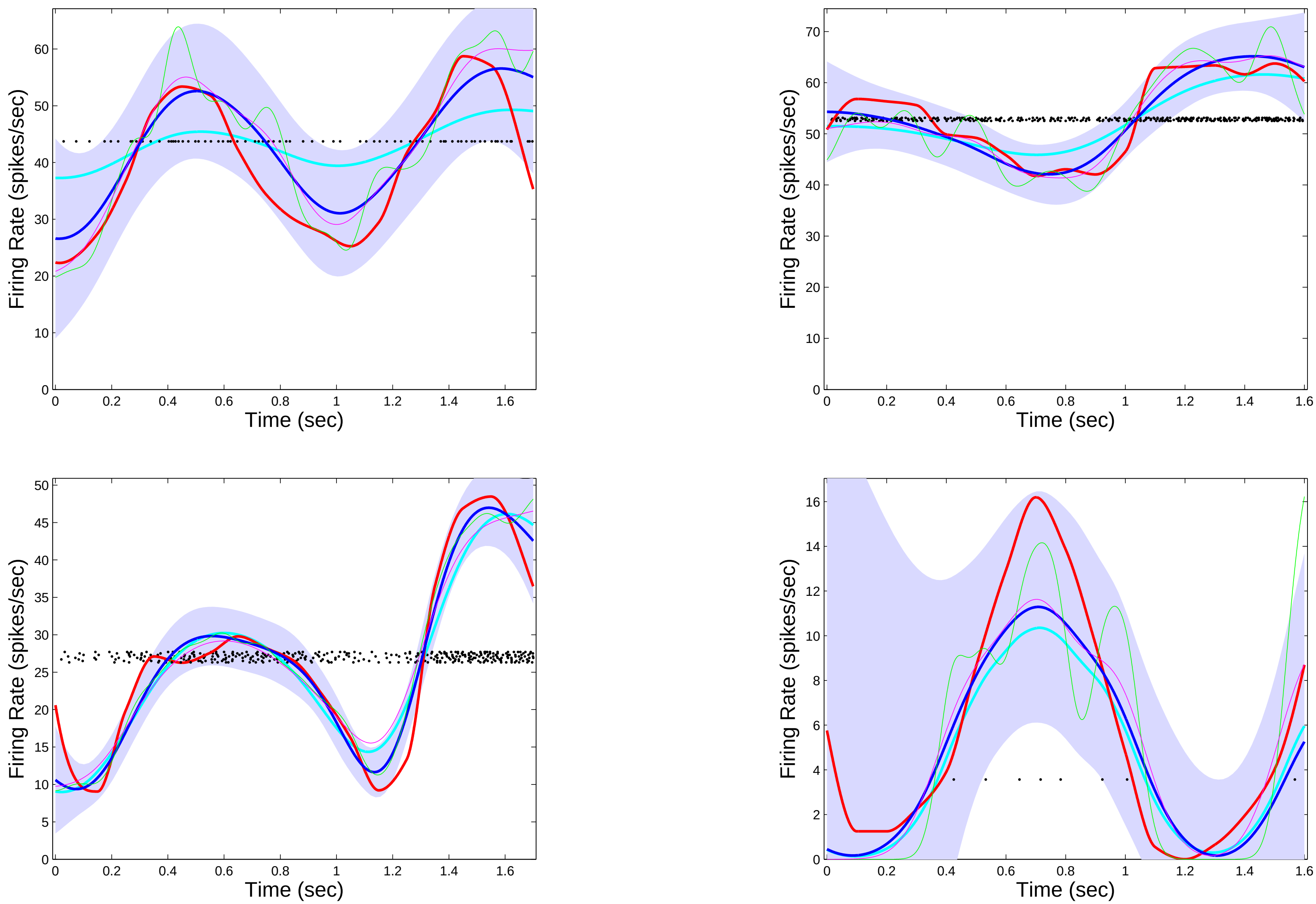
- The hyperparameters are $\theta = [\sigma_f^2, \kappa, \gamma, \mu]$ (where σ_f^2 and κ are the variance and lengthscale of the covariance kernel).
- Laplace approximation to approximate the intractable integral over $\mathbf{x}$:

$$p(\mathbf{y} \mid \theta) = \int_{\mathbf{x}} p(\mathbf{y} \mid \mathbf{x}, \theta) p(\mathbf{x} \mid \theta) d\mathbf{x} \quad \approx \quad p(\mathbf{y} \mid \mathbf{x}^*, \theta) p(\mathbf{x}^* \mid \theta) \frac{(2\pi)^{\frac{n}{2}}}{|\Lambda^* + \mathsf{K}^{-1}|^{\frac{1}{2}}}$$

- This can be optimised to find "best" parameter values. Or can be used to weight different parameter values on a grid to integrate approximately over parameter settings.

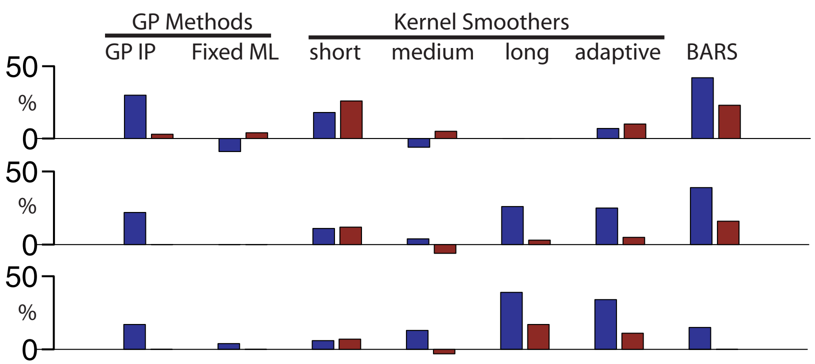
Example 1: GP-based intensity estimates

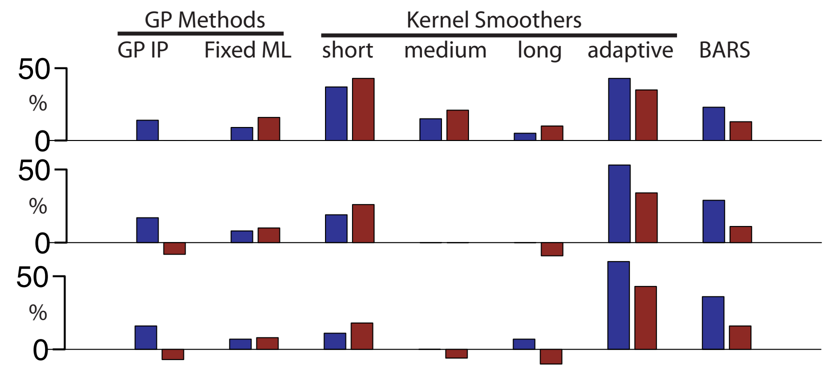
Results

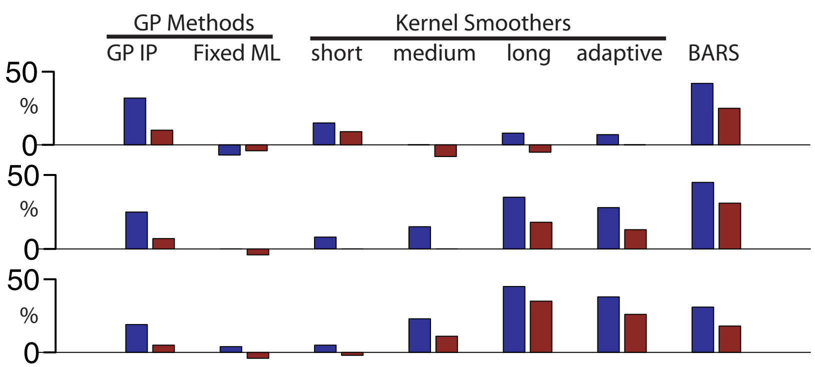


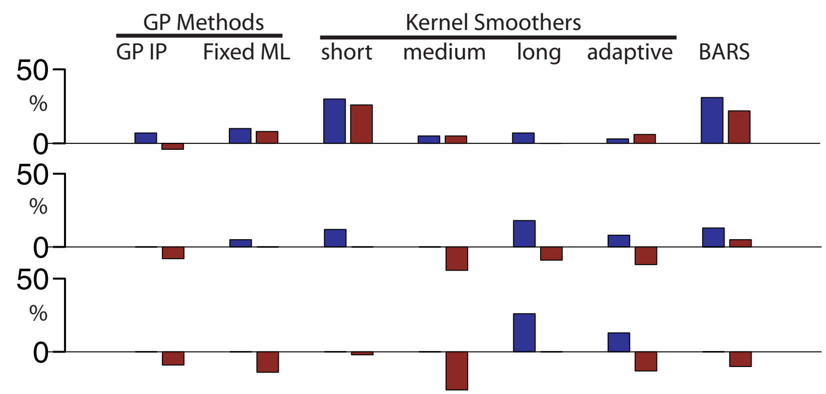
Example 1: GP-based intensity estimates

Results









Example 2: GPFA

Spike train binned (10 – 20 ms) to yield spike counts.

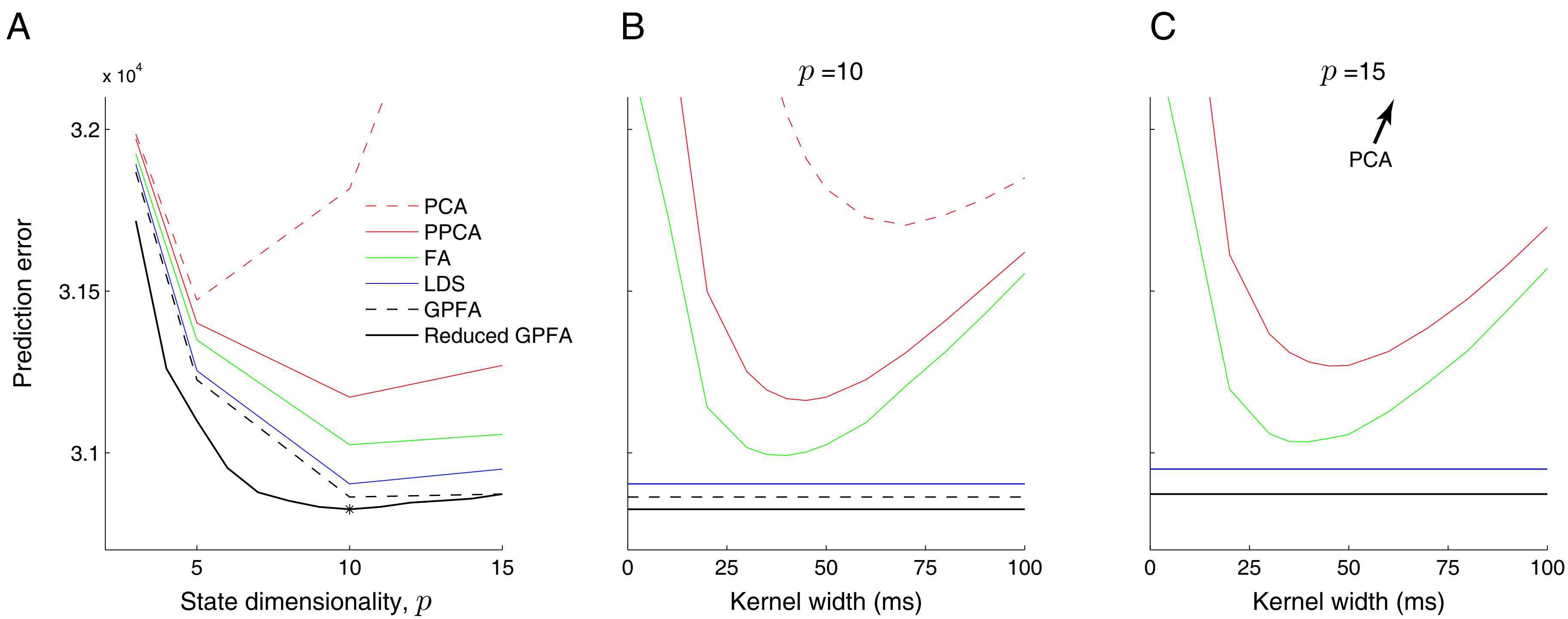
$$x_i(t) \sim \mathcal{GP}[\mathbf{0}; K_i]$$

$$K_i(t_1, t_2) = (1 - \sigma_n^2) \exp\left(-\frac{(t_1 - t_2)^2}{2\tau_i^2}\right) + \sigma_n^2 \delta_{t_1, t_2}$$

$$\mathbf{y}(t)|\mathbf{x}(t) \sim \mathcal{N}\left(C\mathbf{x}(t) + \mathbf{d}, R\right)$$

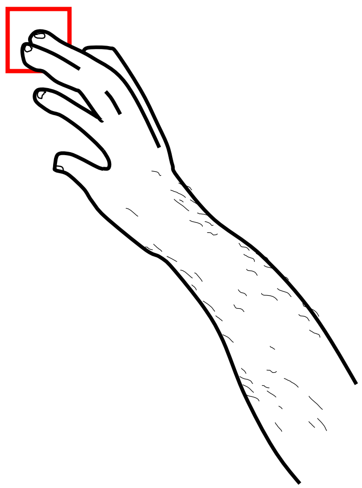
- Spike counts may be square-rooted to stabilise variance of (and Gaussianise) Poisson counts
- The model is jointly Gaussian! Exact inference and learning is possible using Factor-Analysis-like methods.

Example 2: GPFA



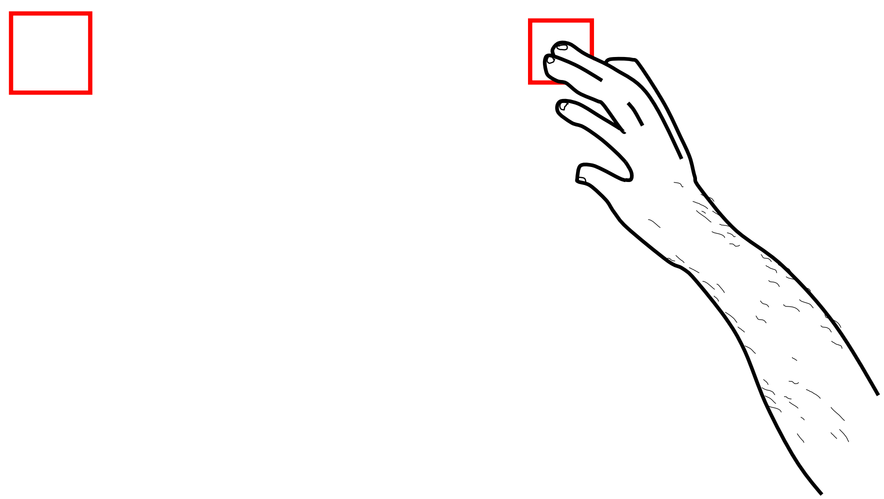
On to some data . . .

The delayed reach task



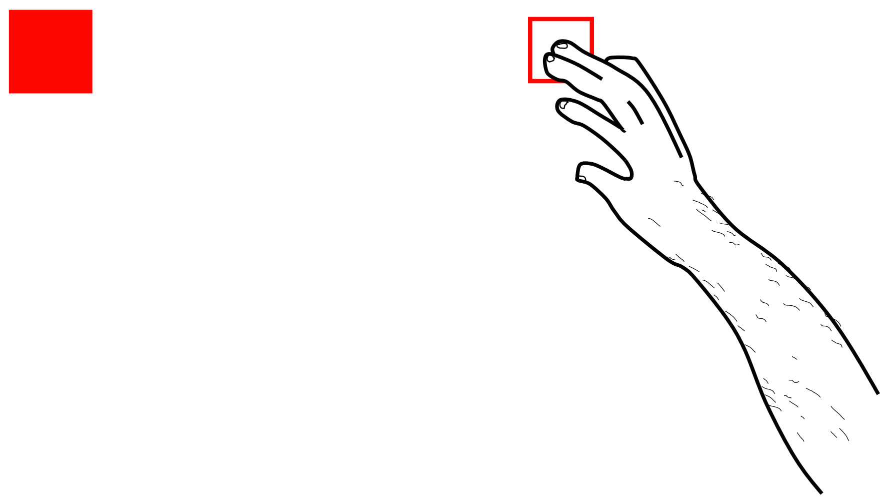
The delayed reach task

Delay period

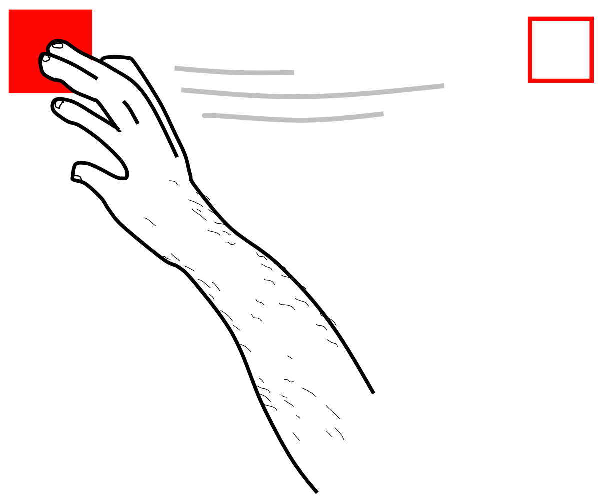


The delayed reach task

go

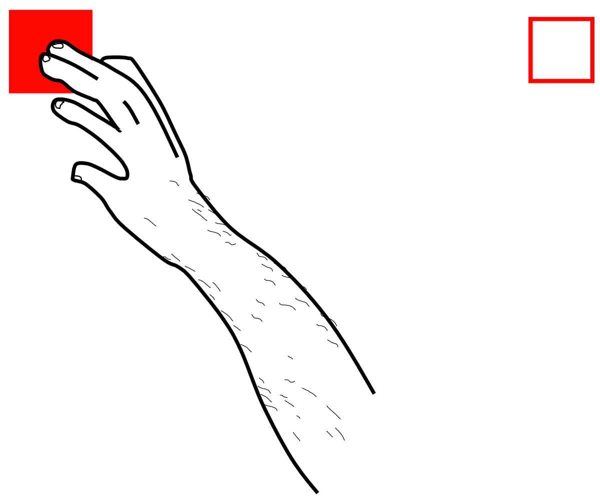


The delayed reach task



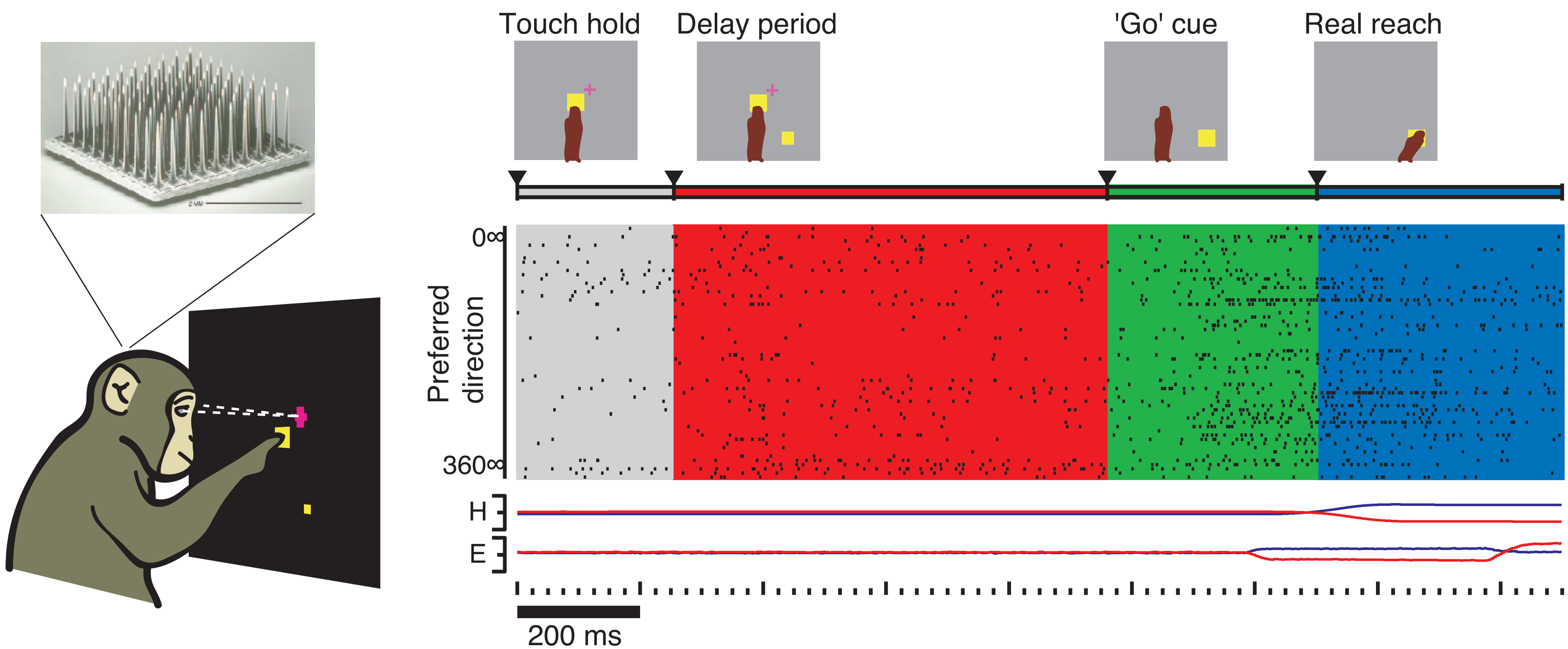
The delayed reach task

reward

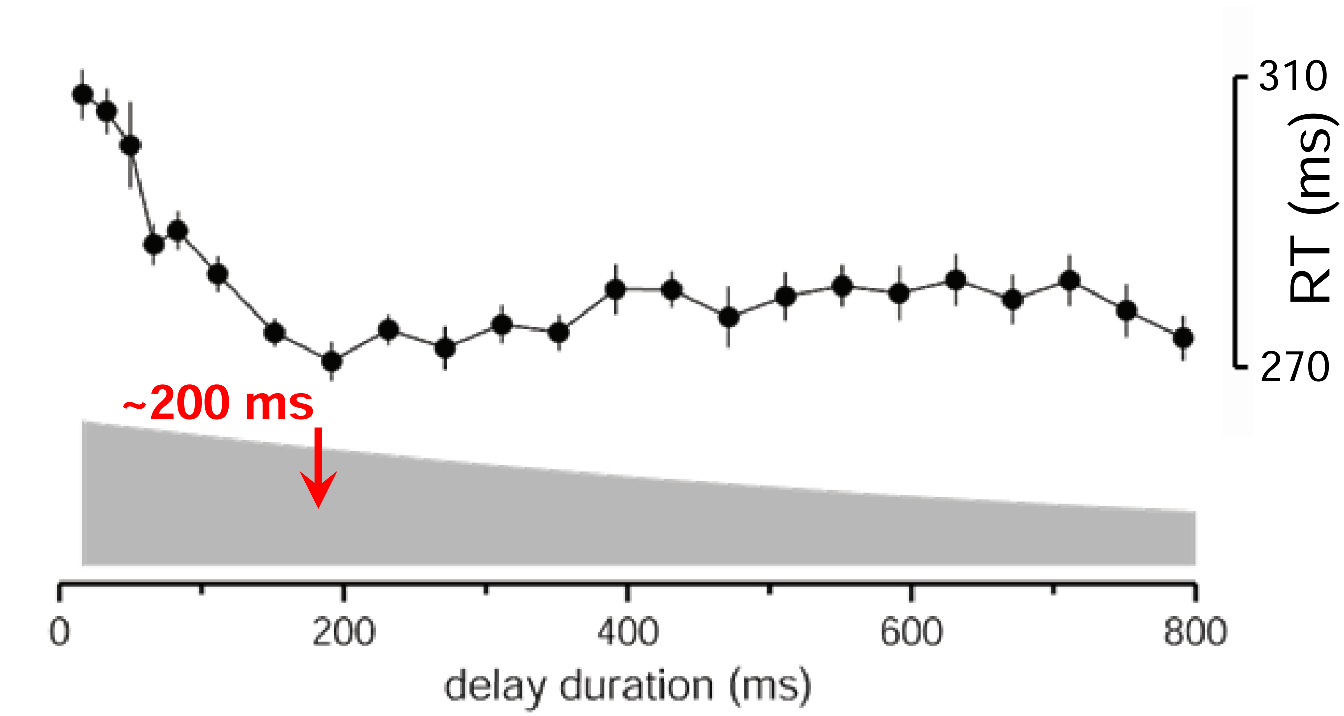


The delayed reach task

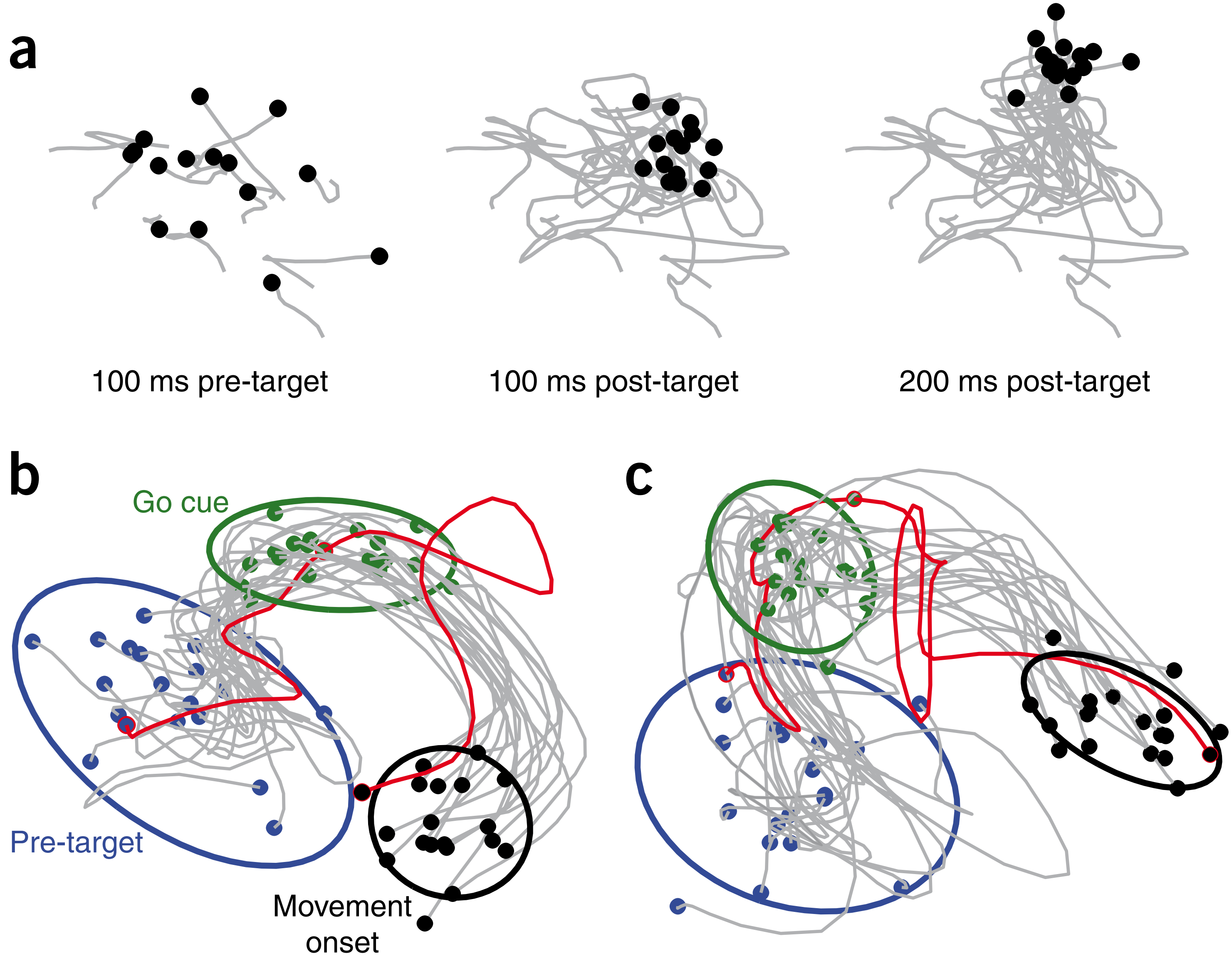
a



Reaction times



Single trial behaviour



Yu *et al.* *J Neurophysiol* 2009; Churchland *et al.* *Nat Neurosci* 2010

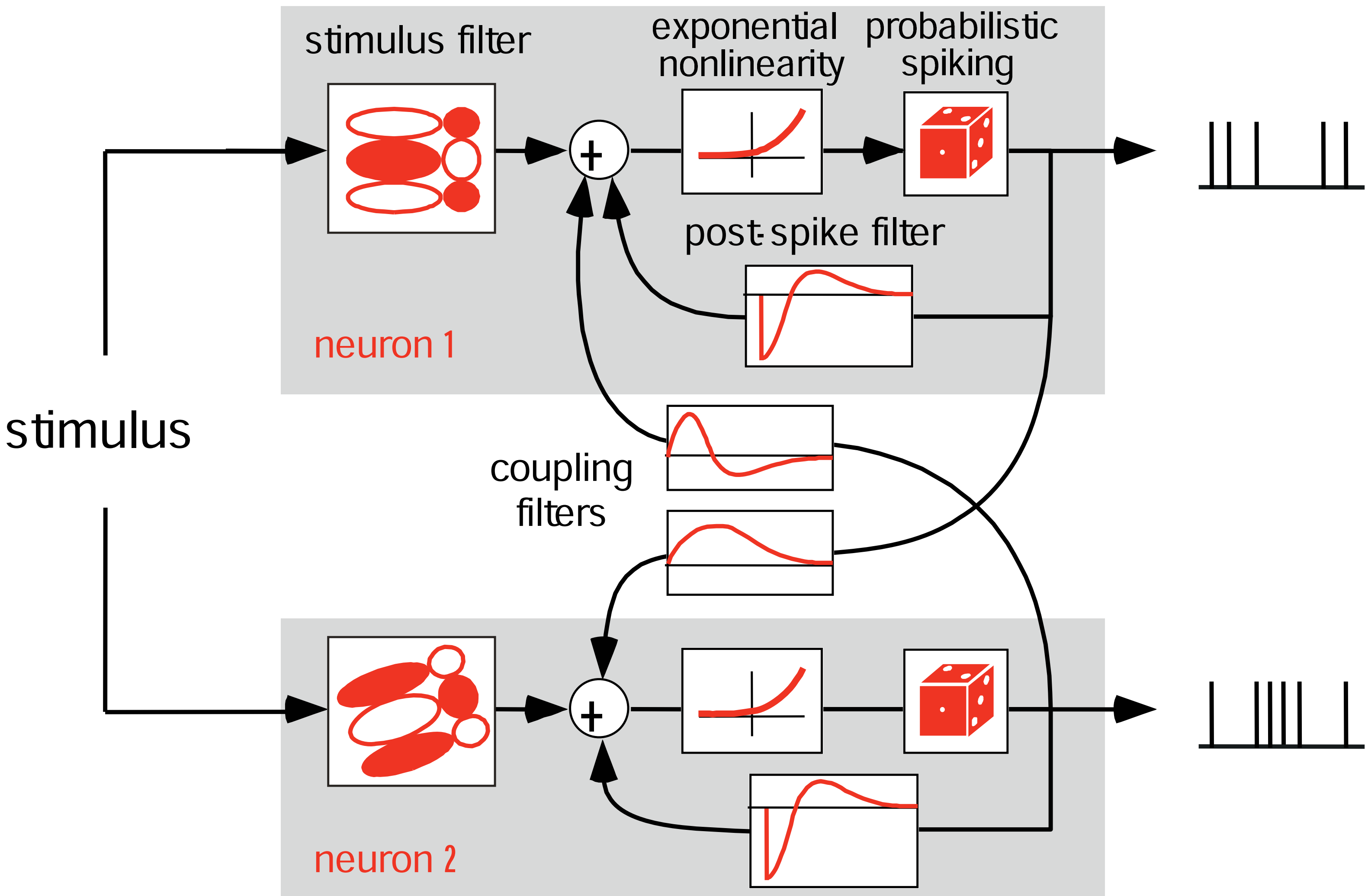
Is this space relevant?

- Statistics of firing
- The “initial condition” planning hypothesis
- Dynamical segmentation

Is this space relevant?

- Statistics of firing
- The “initial condition” planning hypothesis
- Dynamical segmentation

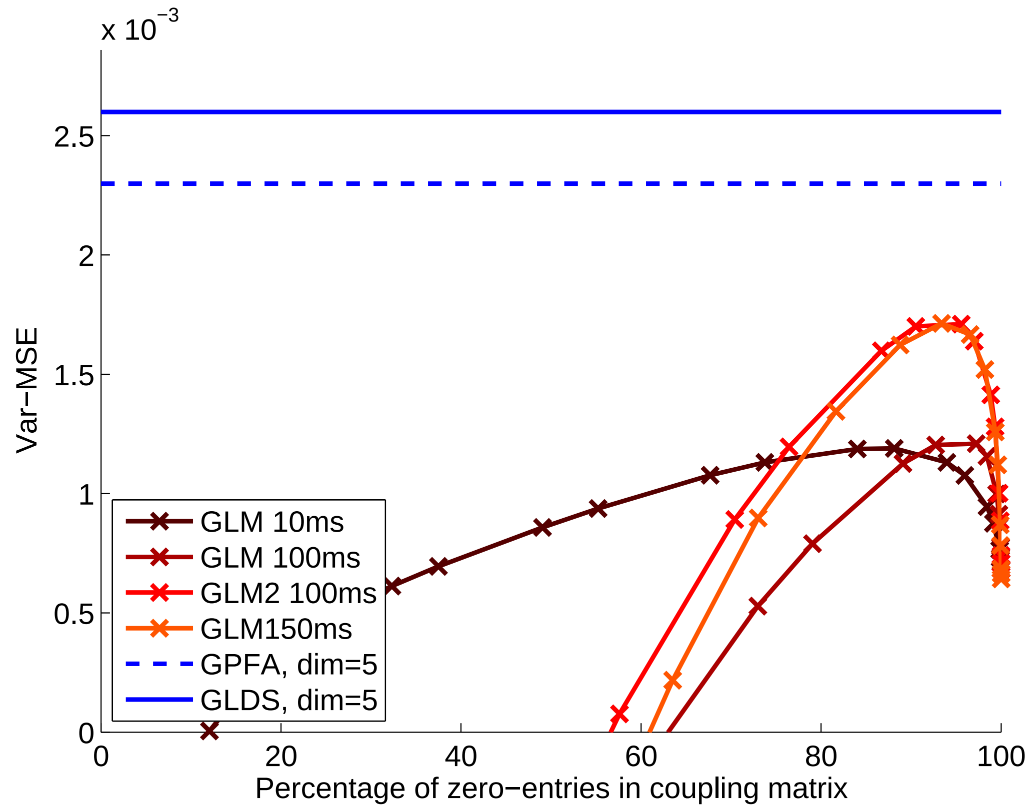
Direct connections?



Truccolo, Pillow, Paninski, and others

Direct connections?

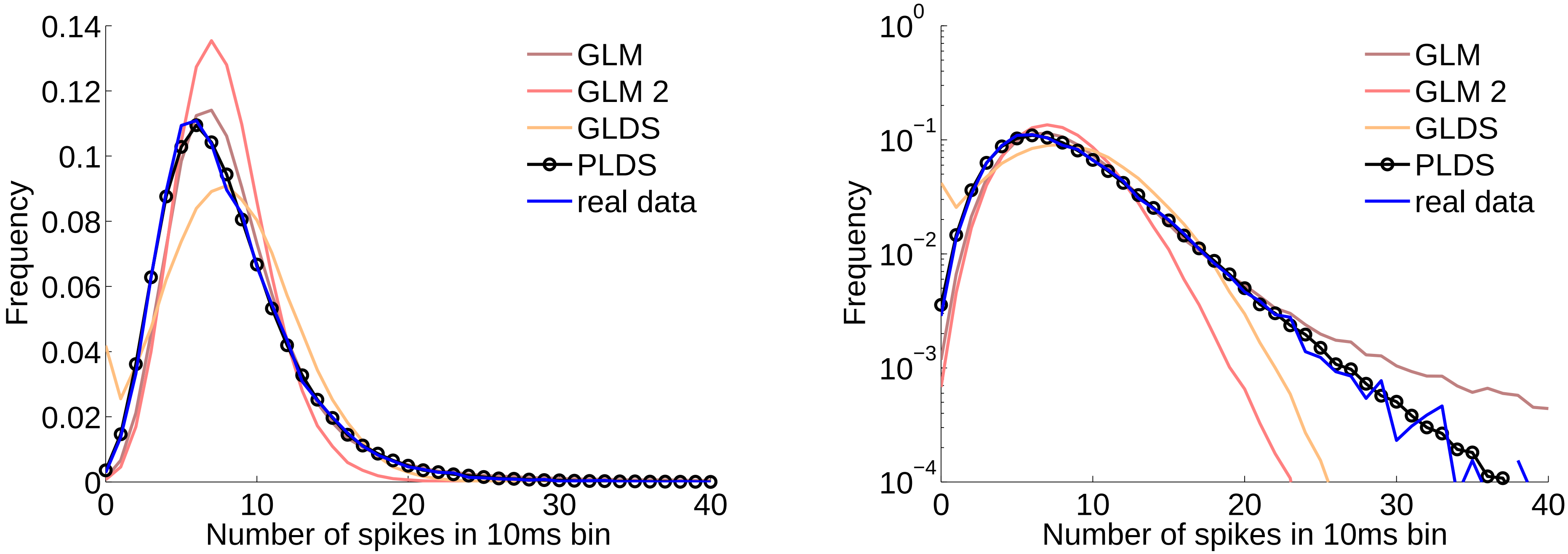
Latent models predict firing of one neuron from others better than GLMs.



Macke *et al.* *NIPS* 2011

Direct connections?

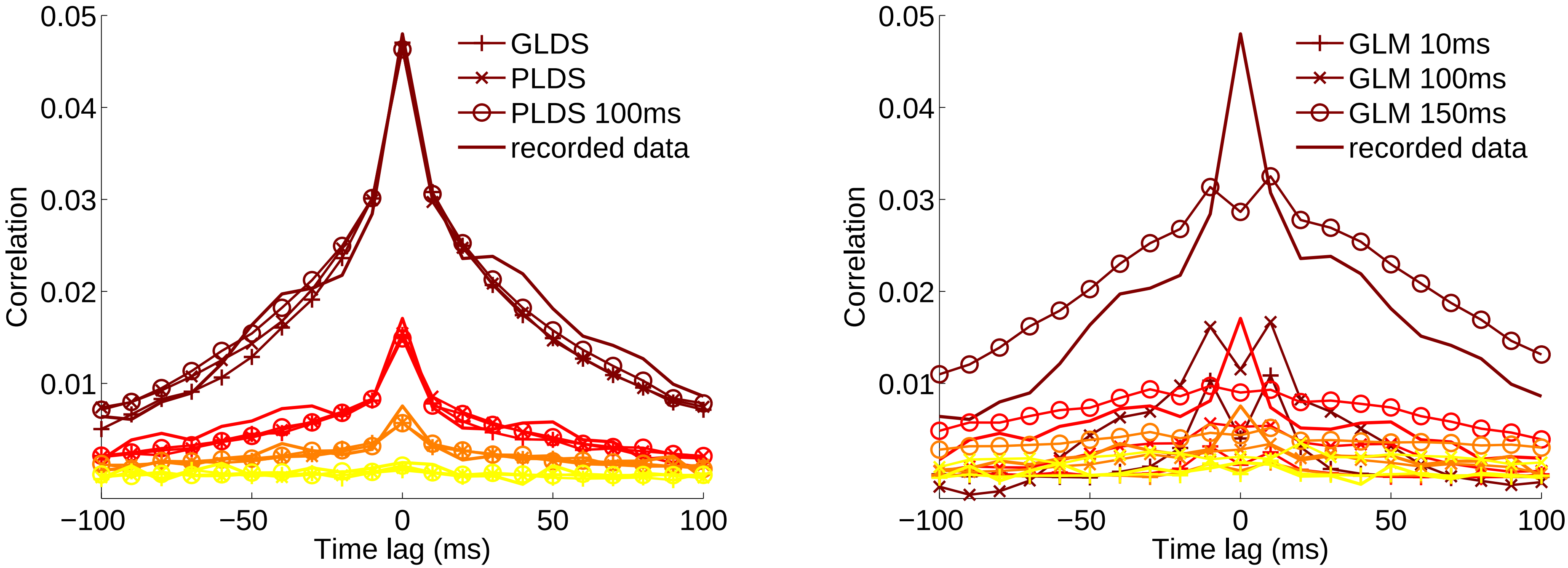
Latent models capture population statistics better than GLMs.



Macke *et al. NIPS* 2011

Direct connections?

Latent models capture population statistics better than GLMs.

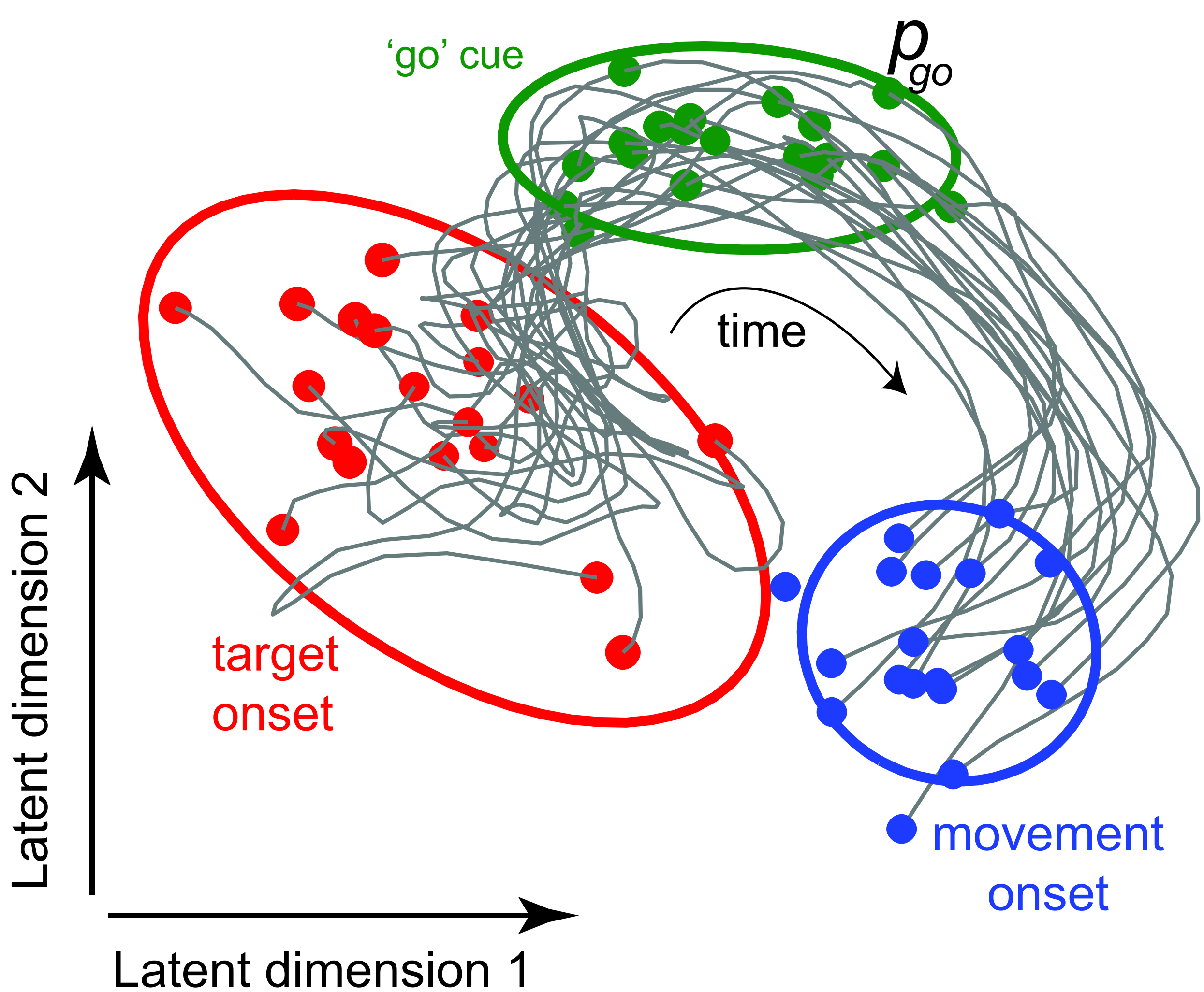


Macke *et al. NIPS* 2011

Is this space relevant?

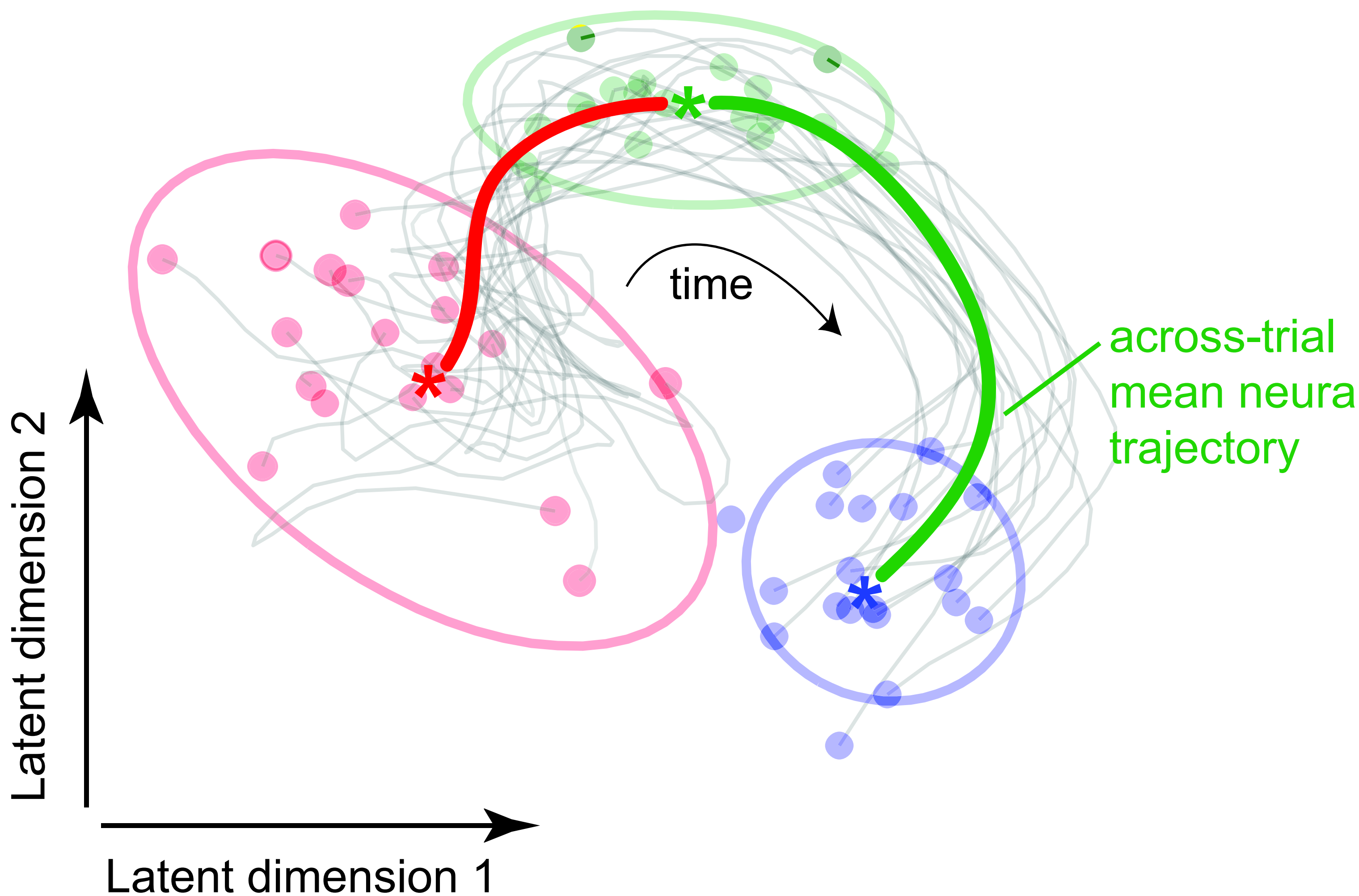
- Statistics of firing
- The “initial condition” planning hypothesis
- Dynamical segmentation

Movement initiation



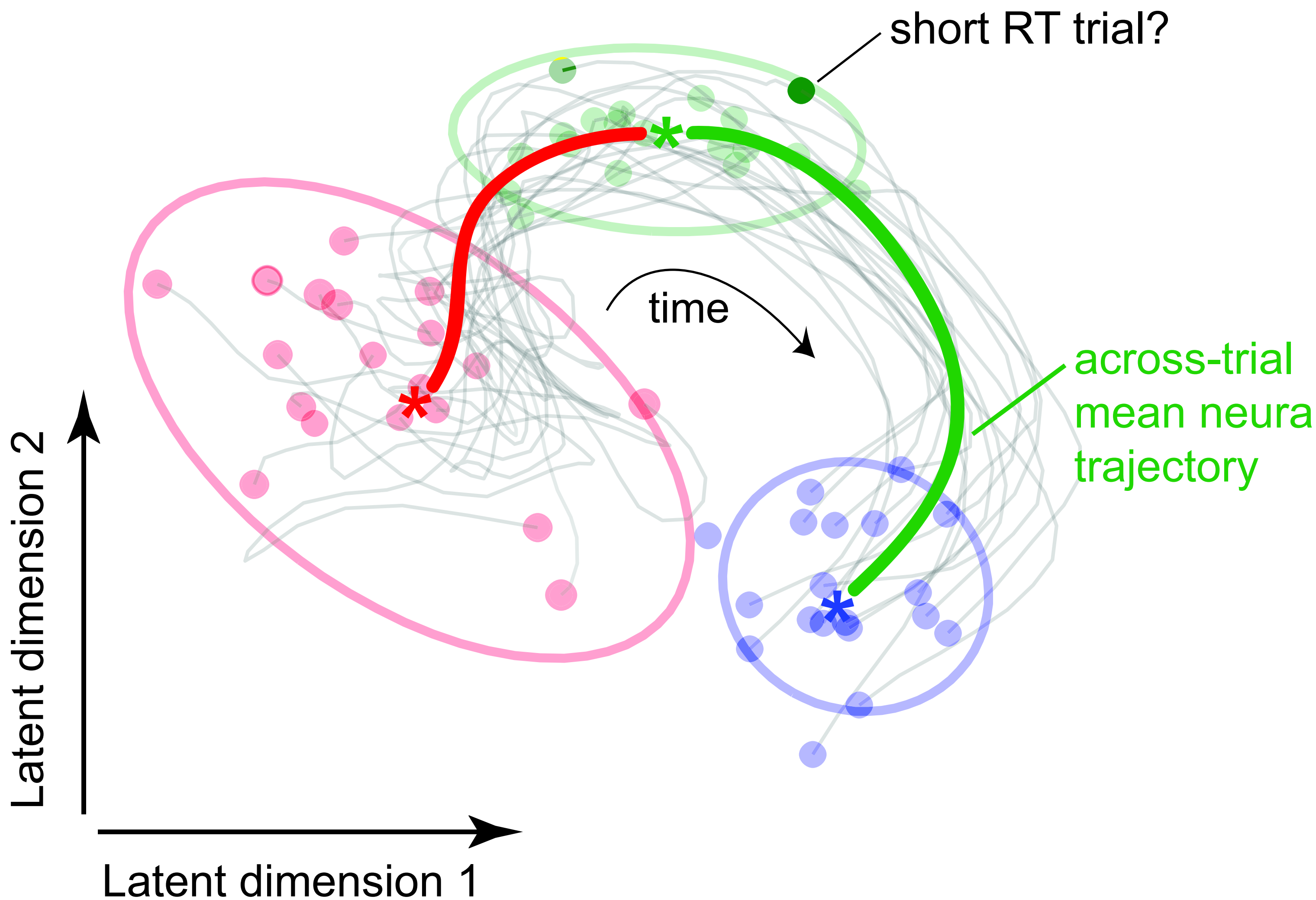
Afshar *et al. Neuron* 2011

Movement initiation



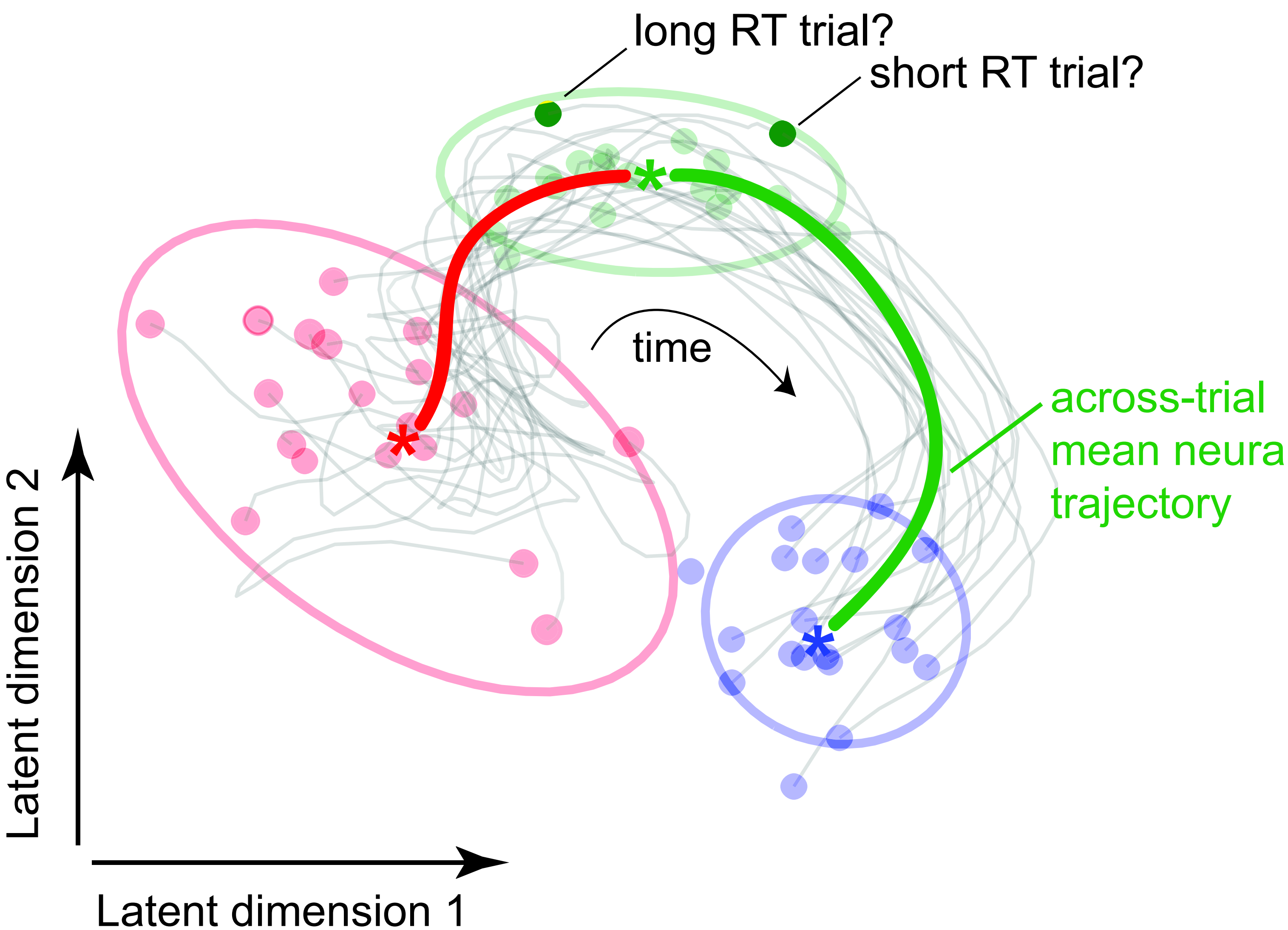
Afshar *et al. Neuron* 2011

Movement initiation



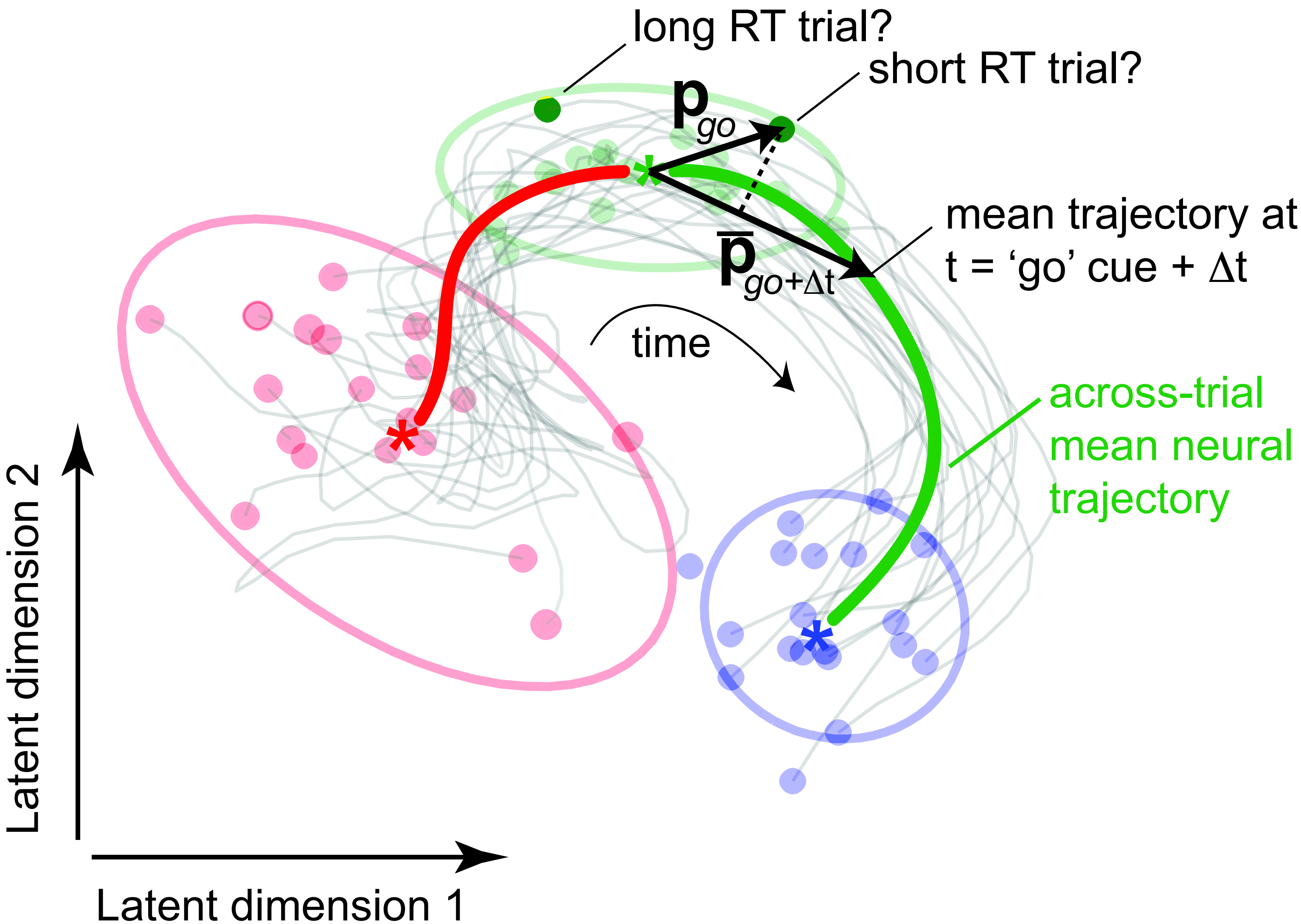
Afshar *et al. Neuron* 2011

Movement initiation



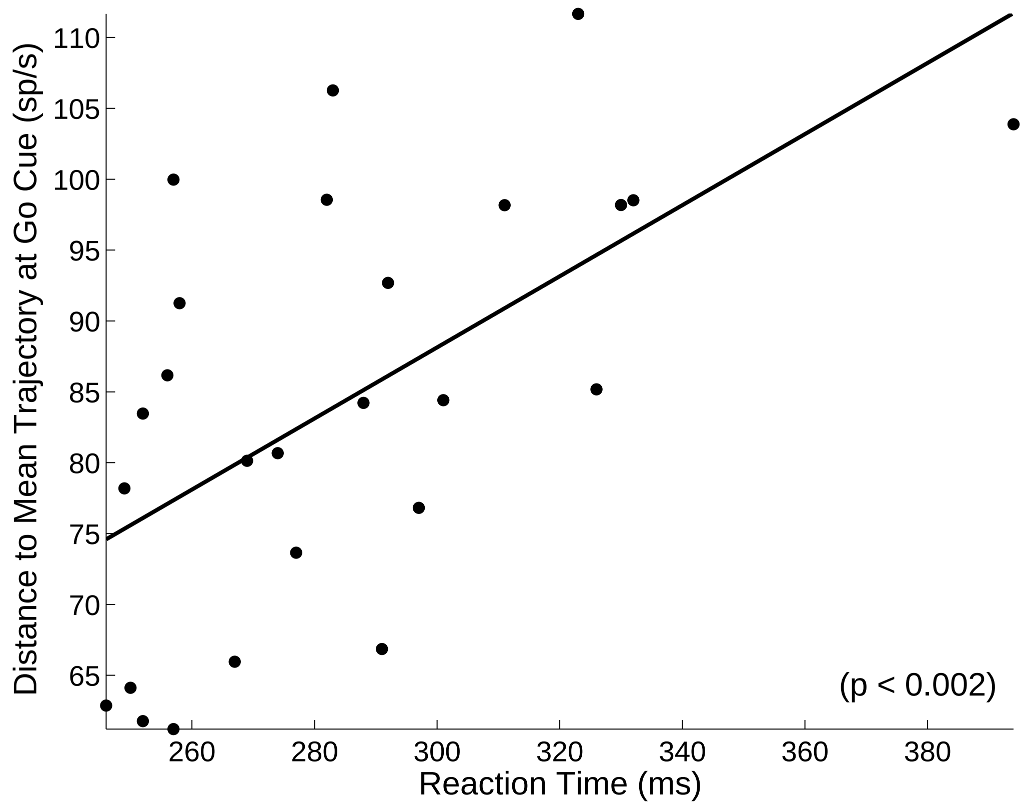
Afshar *et al.* *Neuron* 2011

Movement initiation



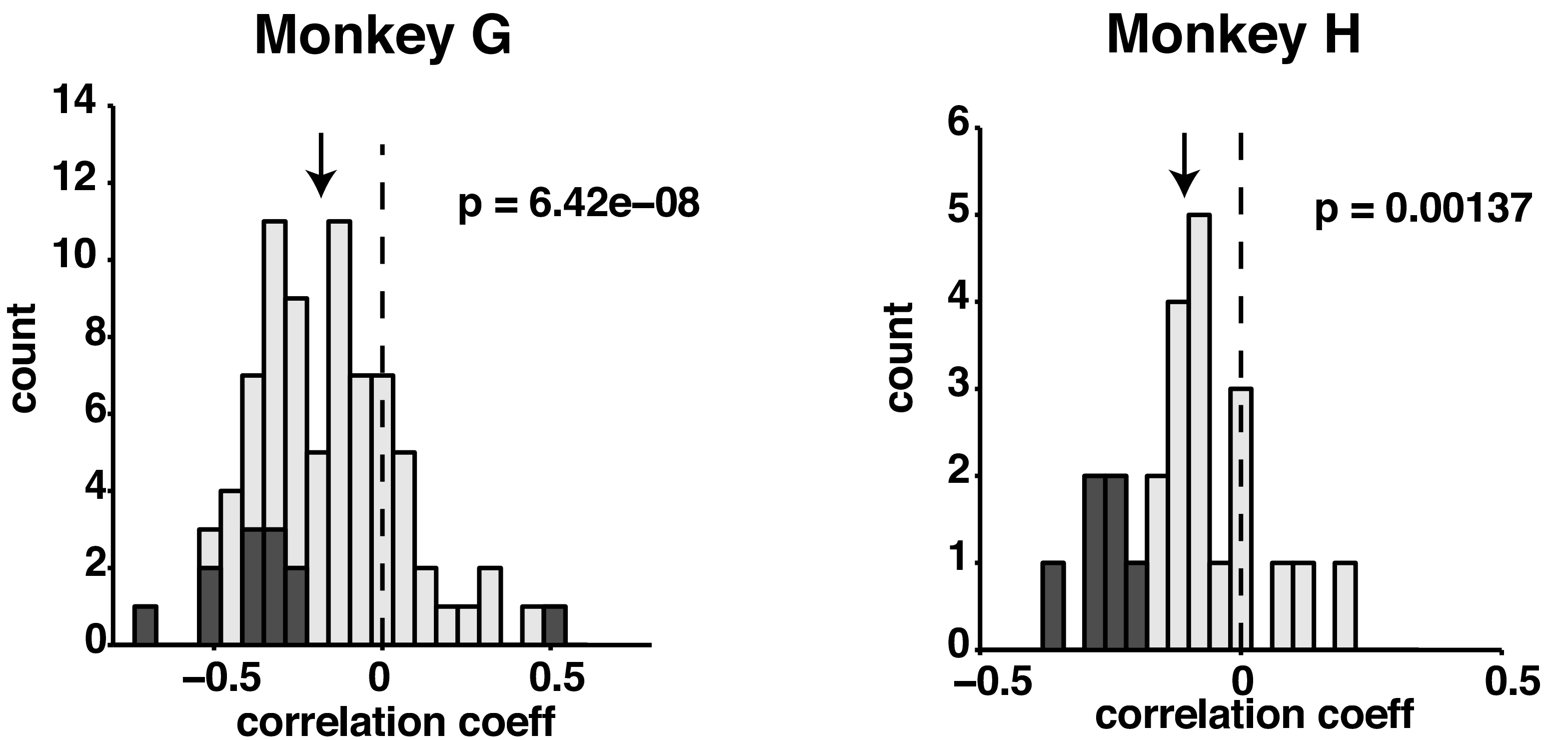
Afshar *et al. Neuron* 2011

Initial point correlated with RT



Afshar *et al. Neuron* 2011

Initial point correlated with RT

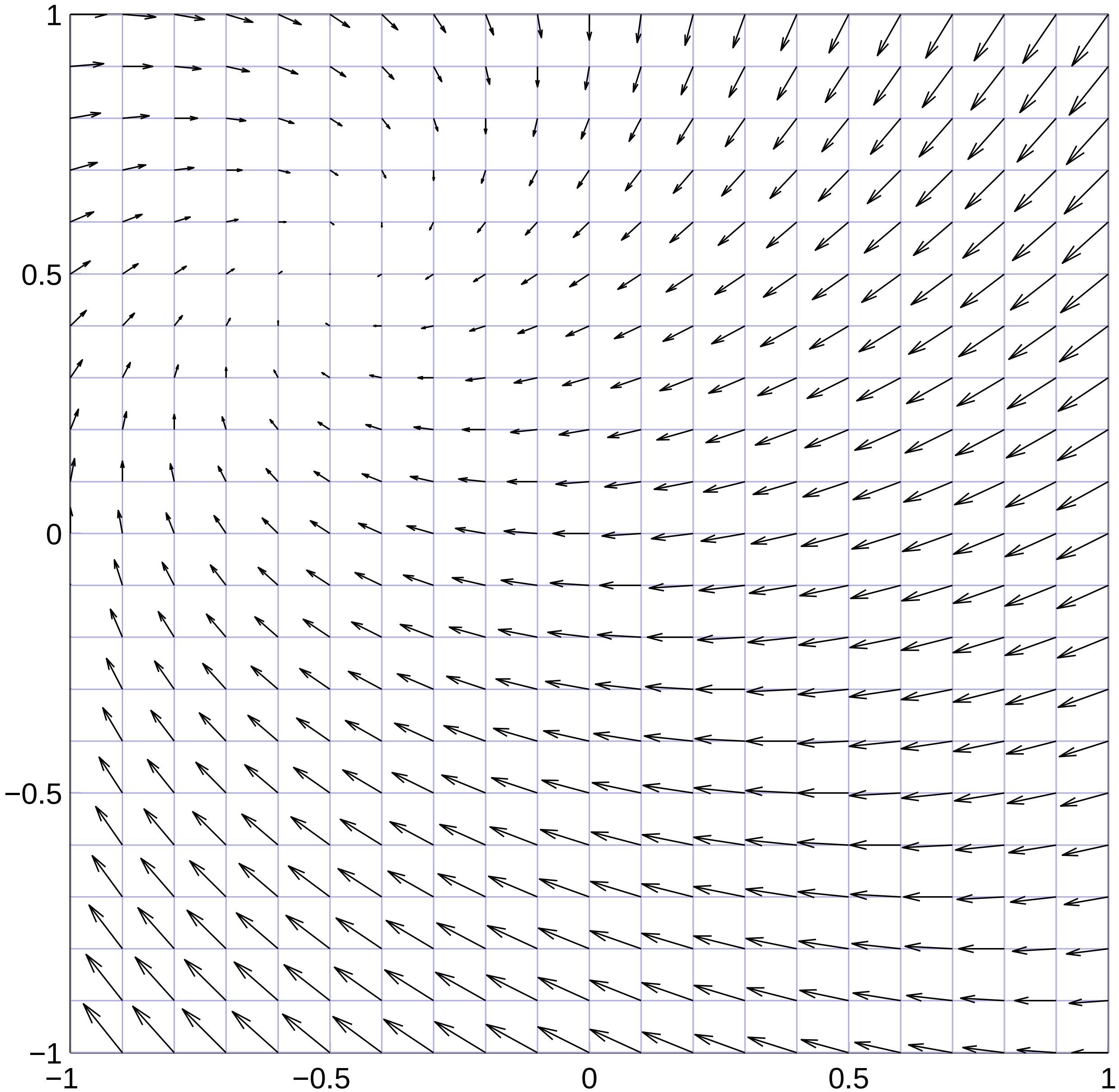


Afshar *et al.* *Neuron* 2011

Is this space relevant?

- Statistics of firing
- The “initial condition” planning hypothesis
- Dynamical segmentation

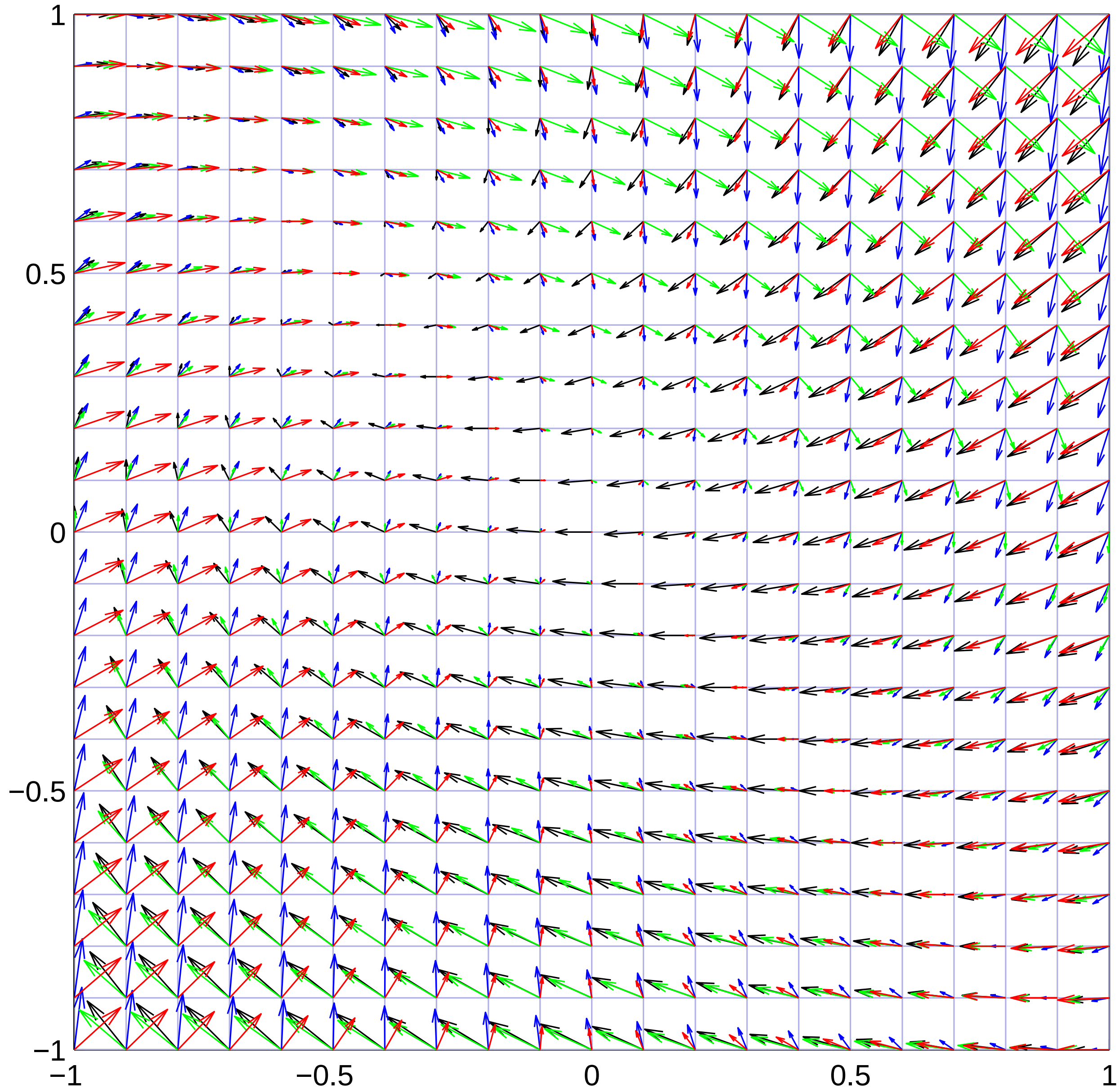
Switching Dynamics



Petreska *et al.* *NIPS* 2011

Switching Dynamics

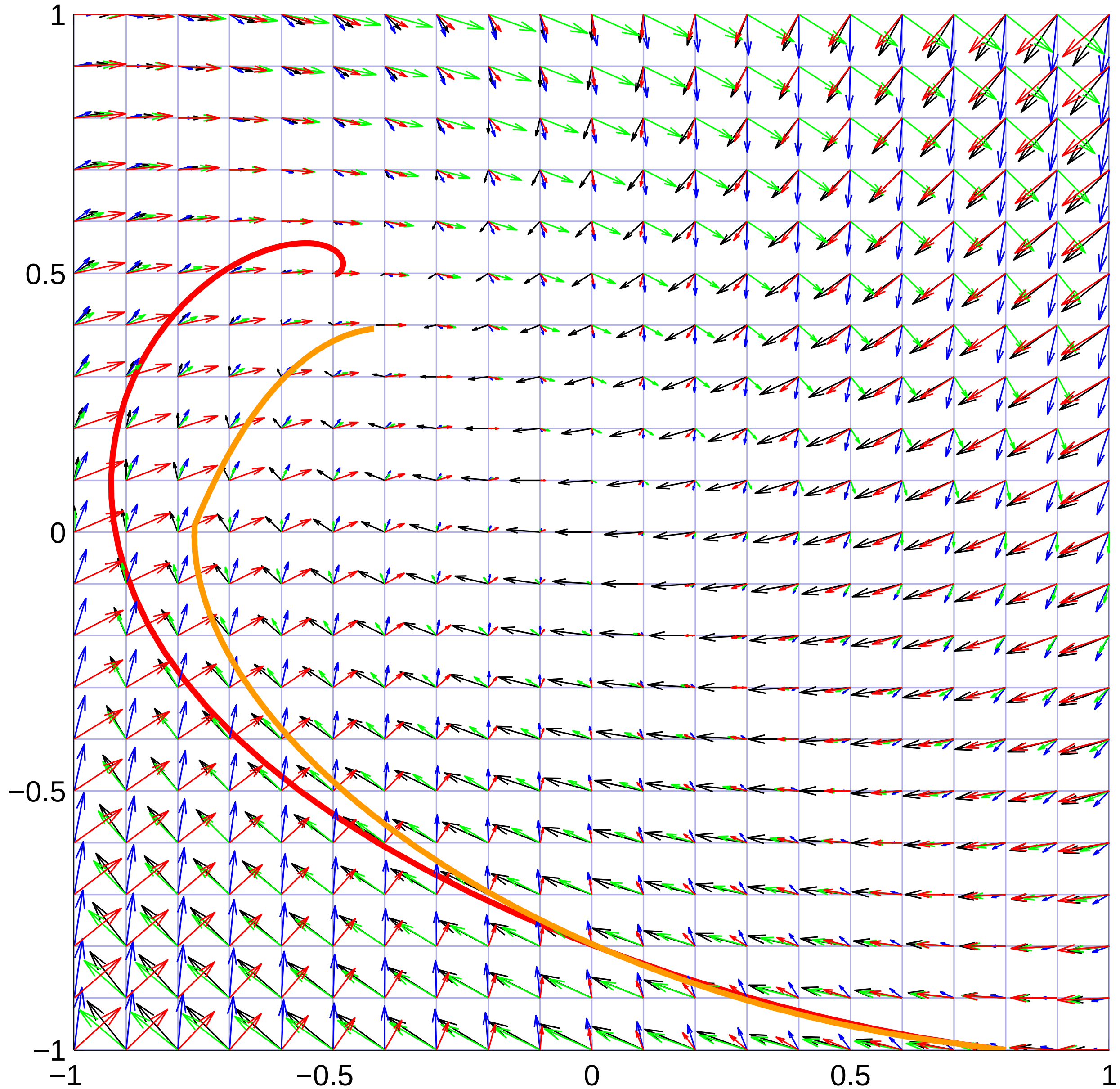
$$s_t \sim T_{s_{t-1}}$$
$$\mathbf{x}_t = A_{s_t}\mathbf{x}_{t-1} + \text{innovations}$$



Petreska *et al. NIPS* 2011

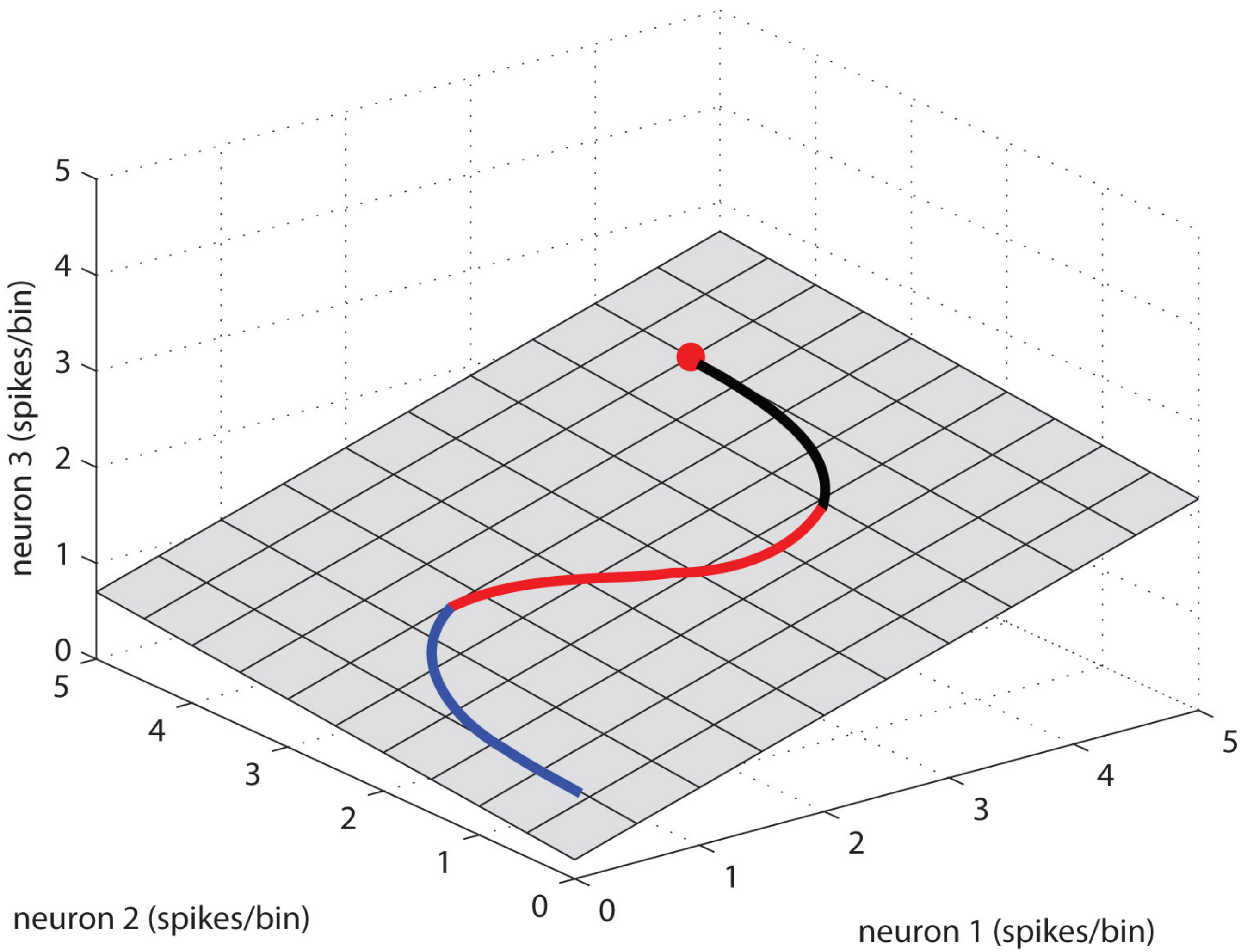
Switching Dynamics

$$s_t \sim T_{s_{t-1}}$$
$$\mathbf{x}_t = A_{s_t}\mathbf{x}_{t-1} + \text{innovations}$$



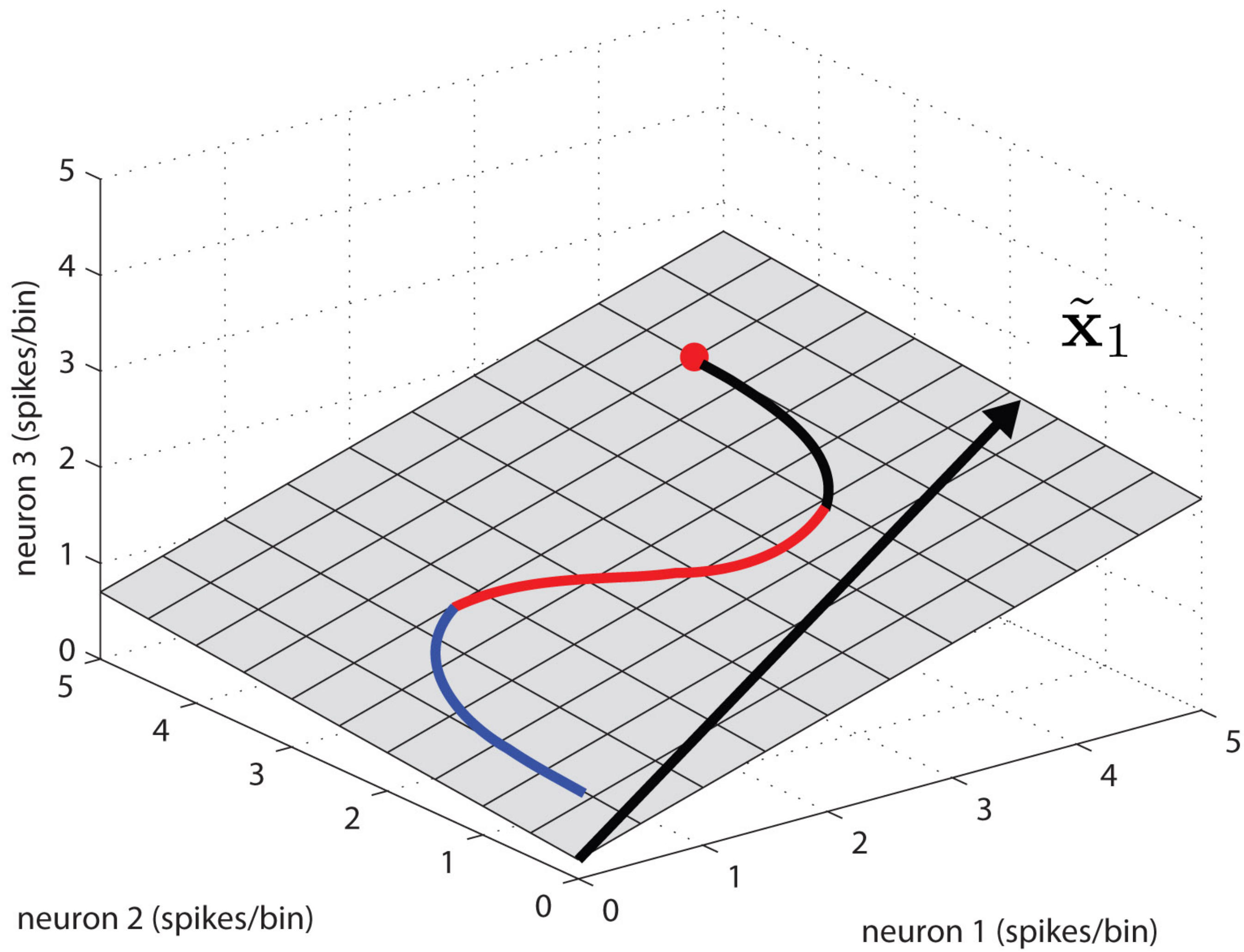
Petreska *et al.* *NIPS* 2011

Single-trial segmentation



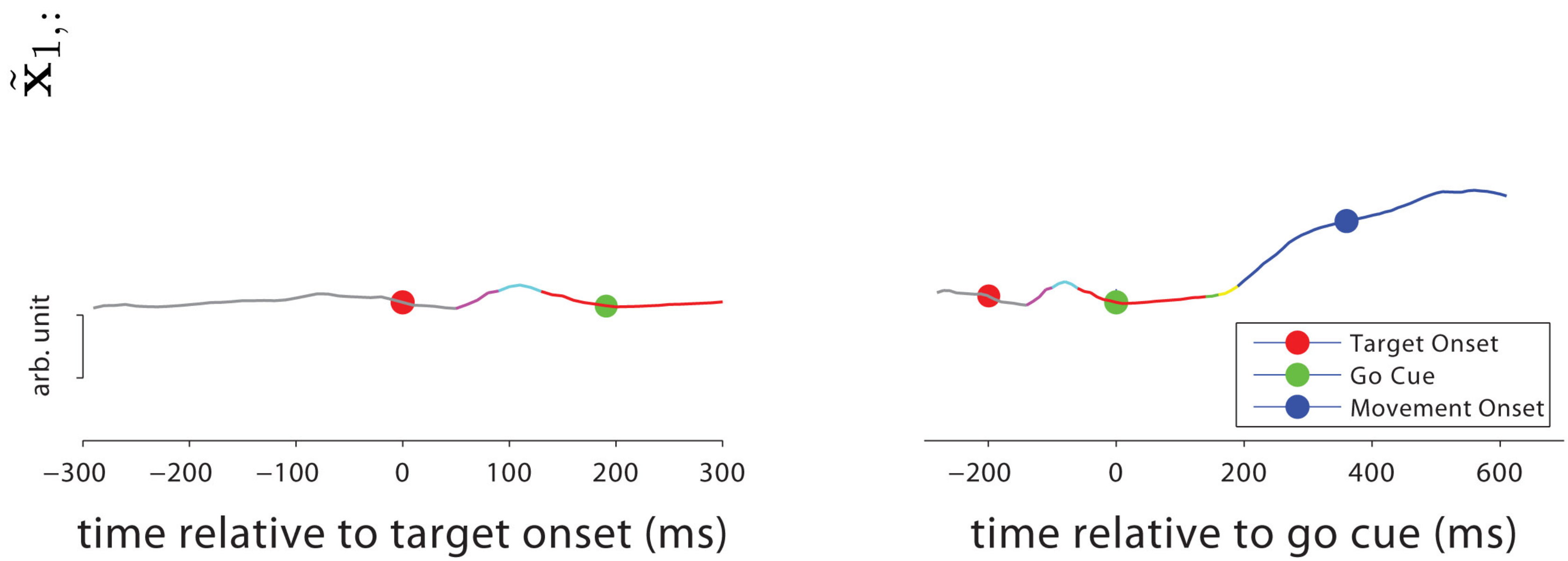
Petreska *et al. NIPS* 2011

Single-trial segmentation



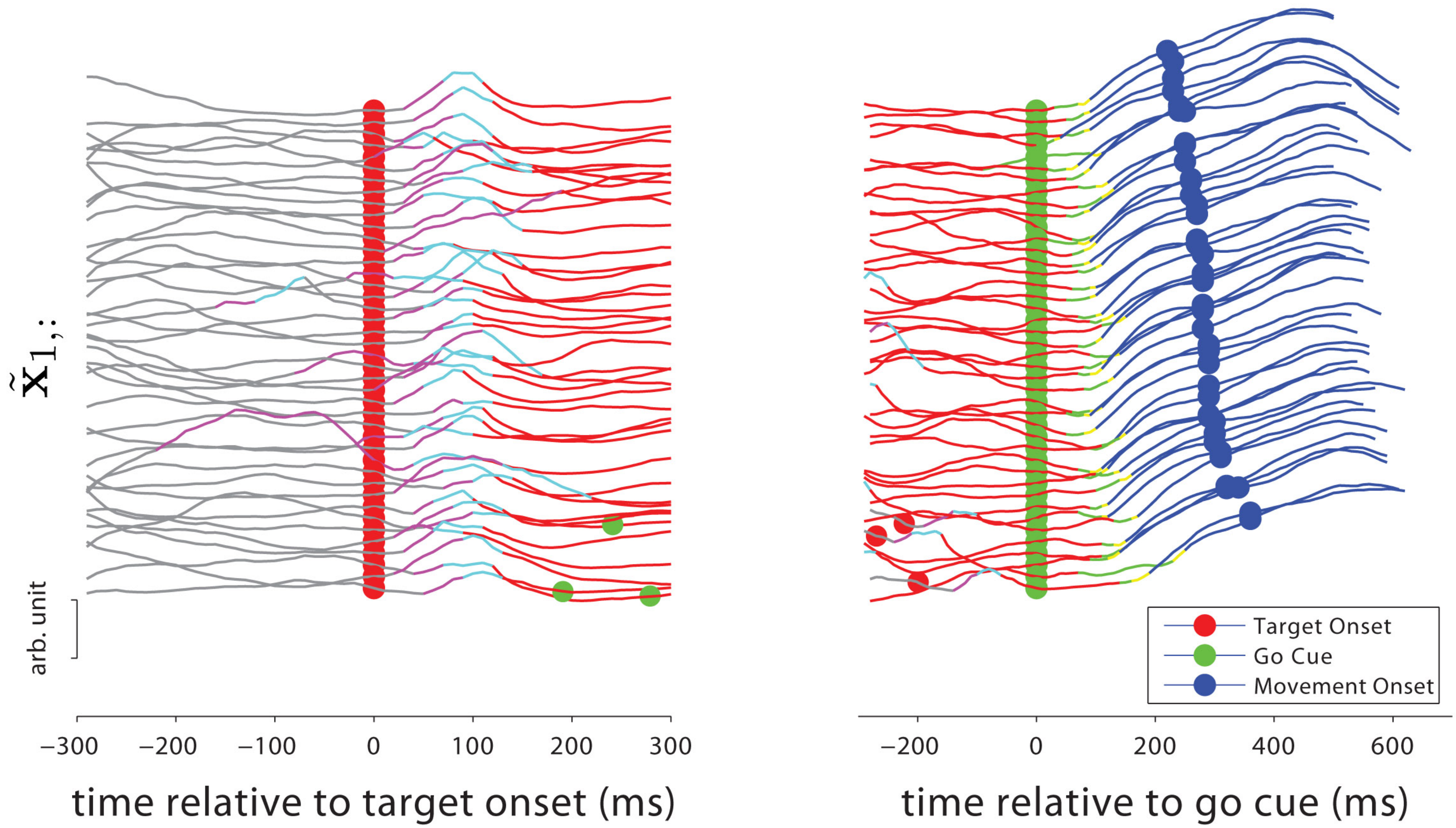
Petreska *et al.* *NIPS* 2011

Single-trial segmentation



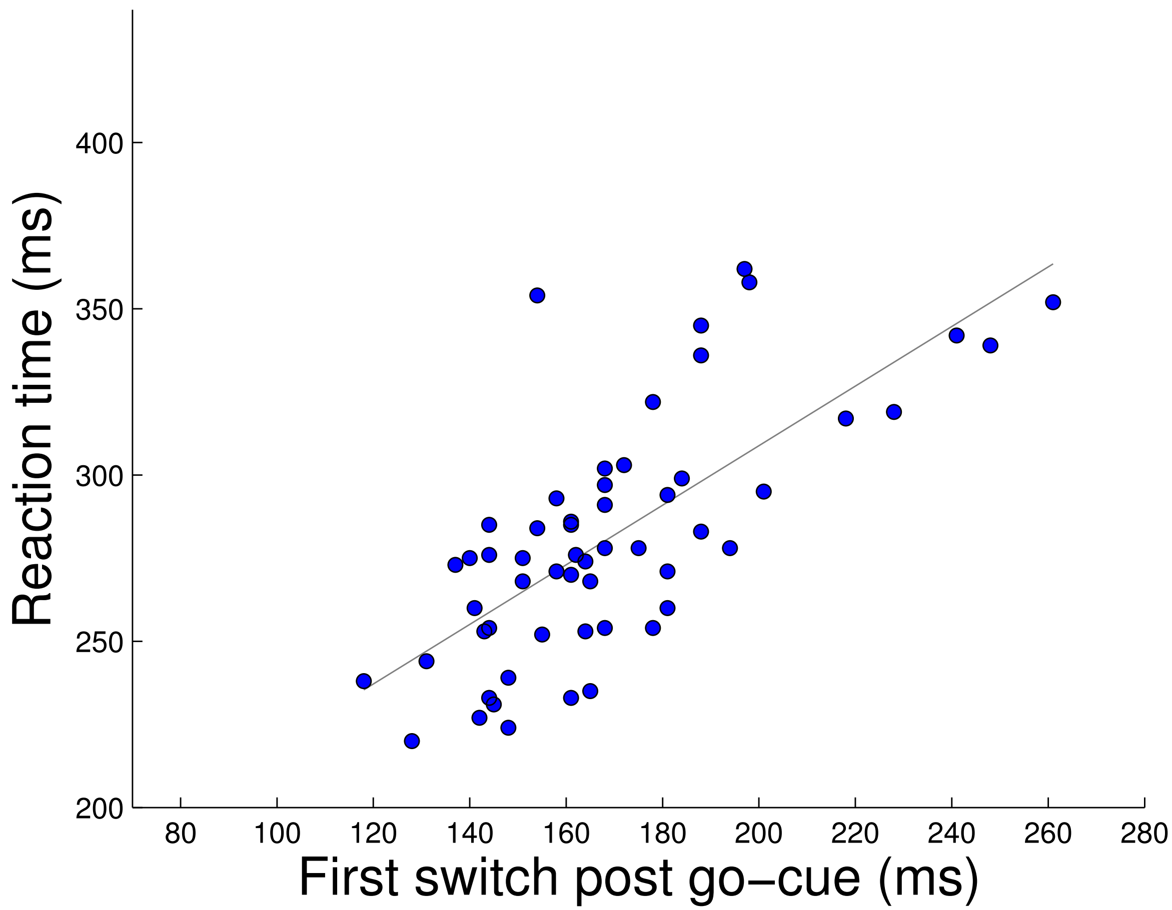
Petreska *et al. NIPS* 2011

Single-trial segmentation



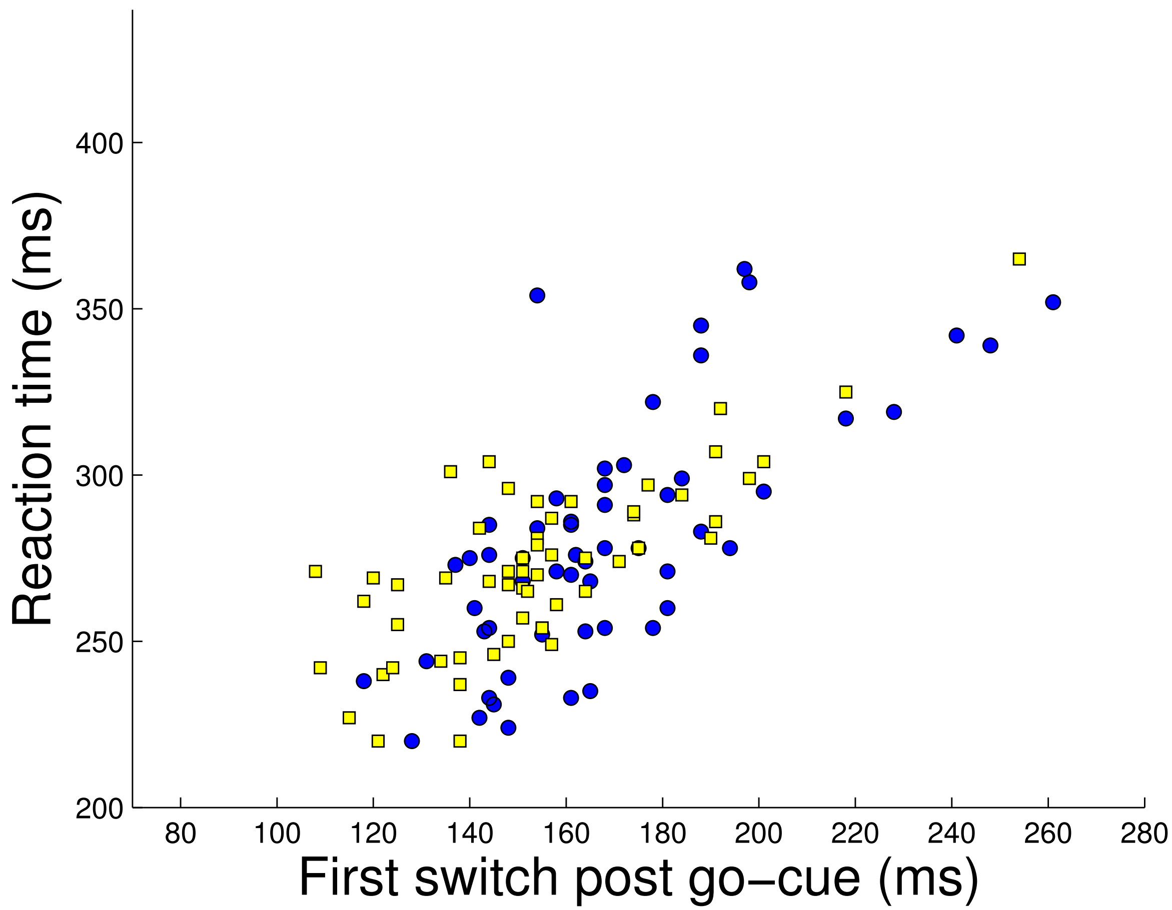
Petreska *et al. NIPS* 2011

Event predictions



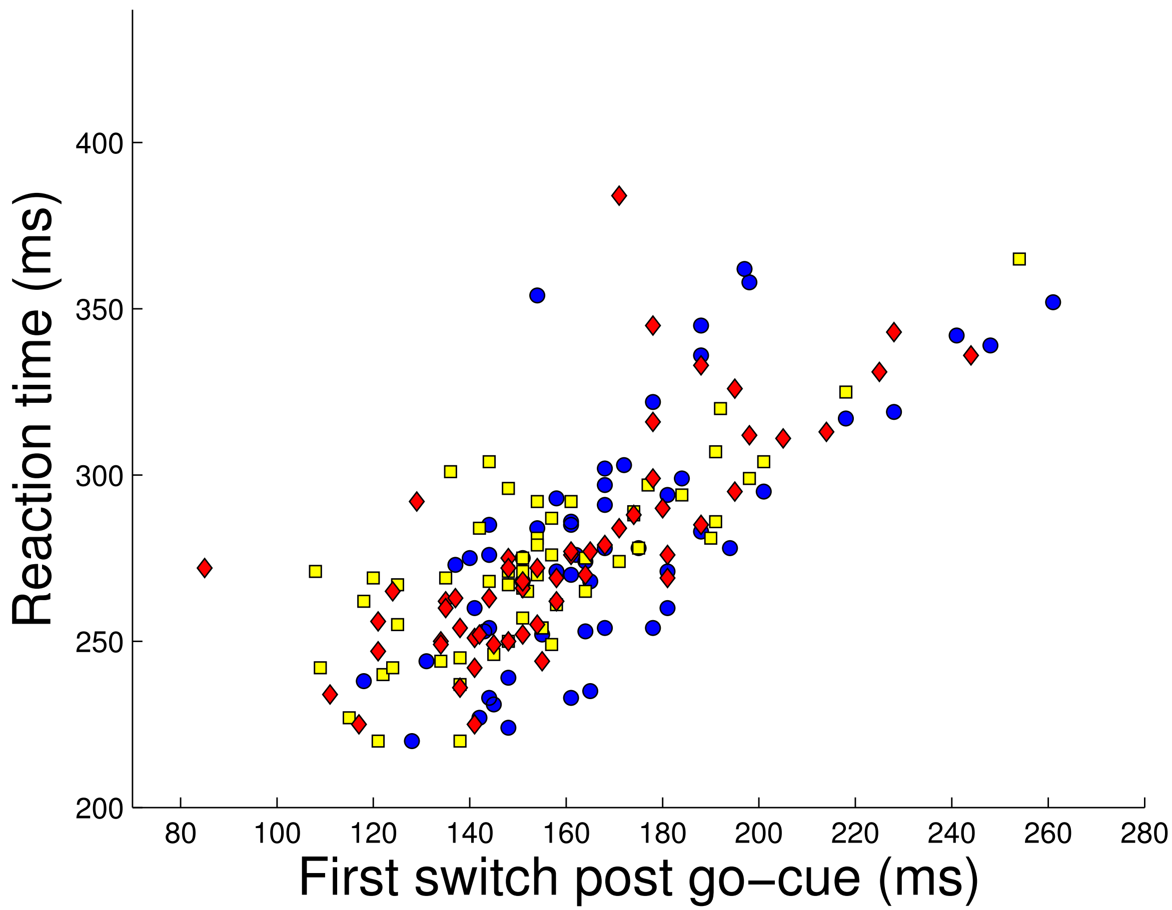
Petreska *et al*. *NIPS* 2011

Event predictions



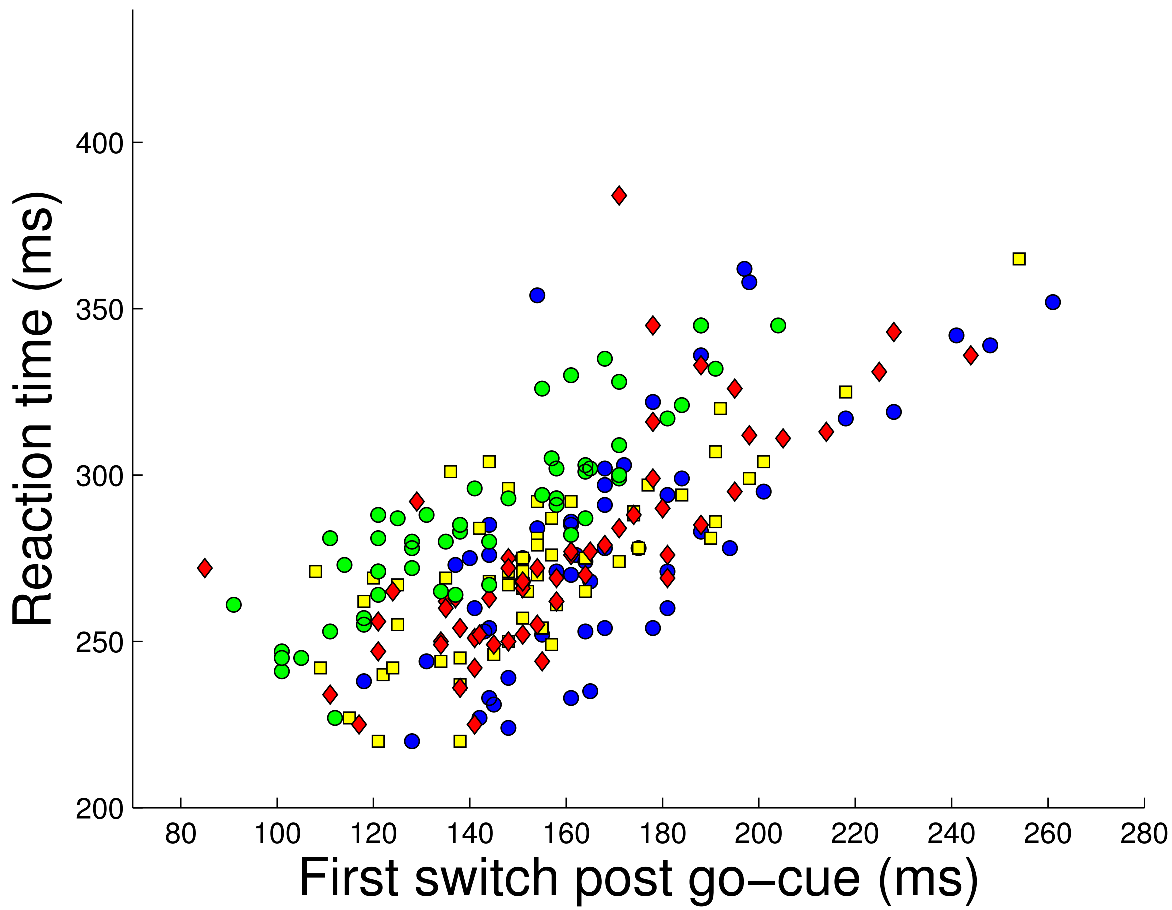
Petreska *et al.* *NIPS* 2011

Event predictions



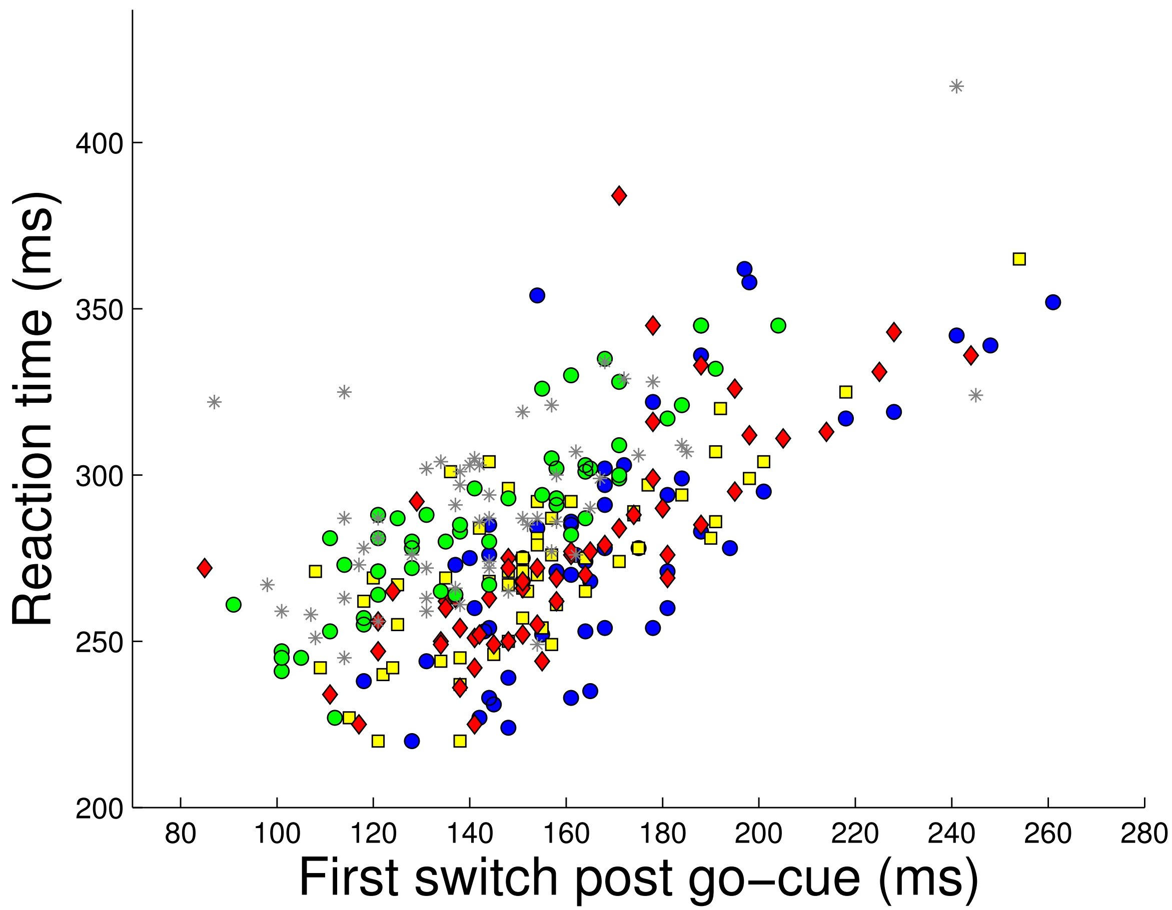
Petreska *et al.* *NIPS* 2011

Event predictions



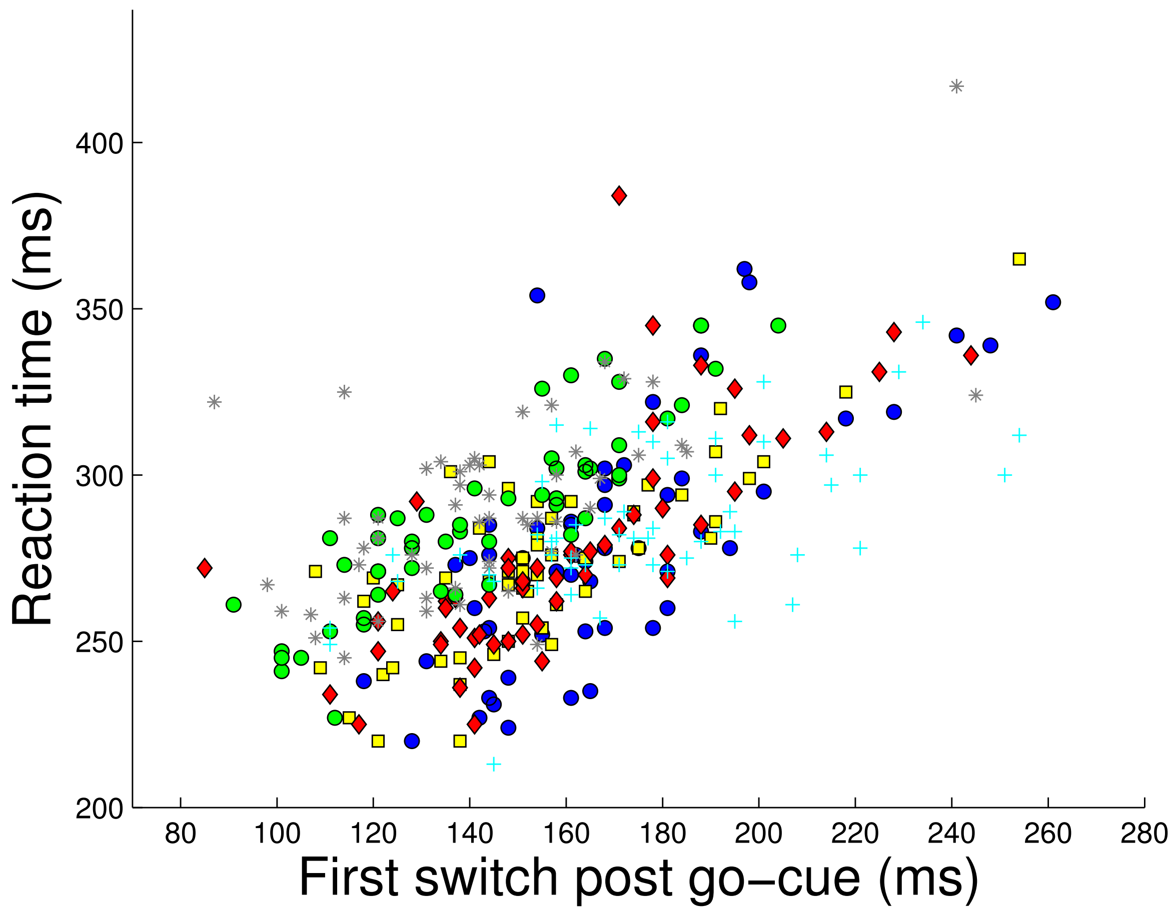
Petreska *et al.* *NIPS* 2011

Event predictions



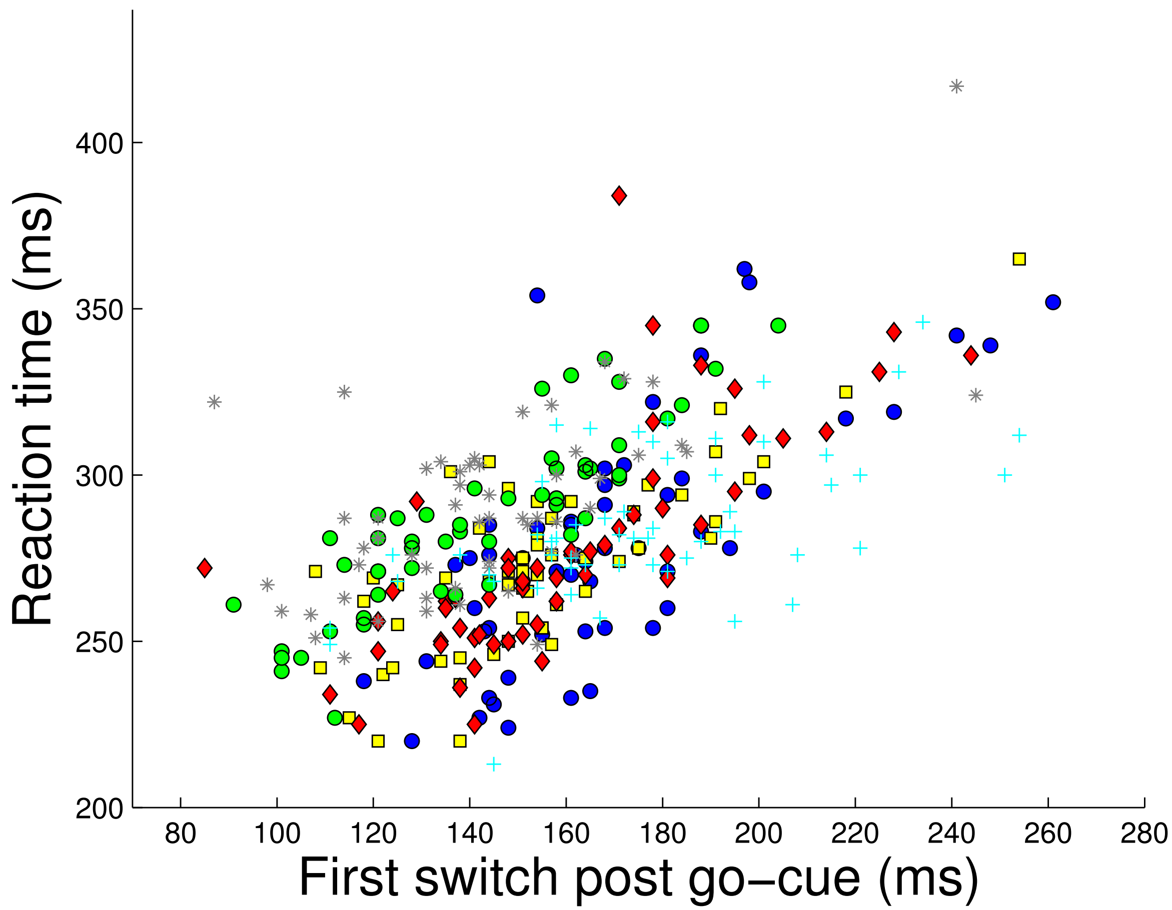
Petreska *et al. NIPS* 2011

Event predictions



Petreska *et al.* *NIPS* 2011

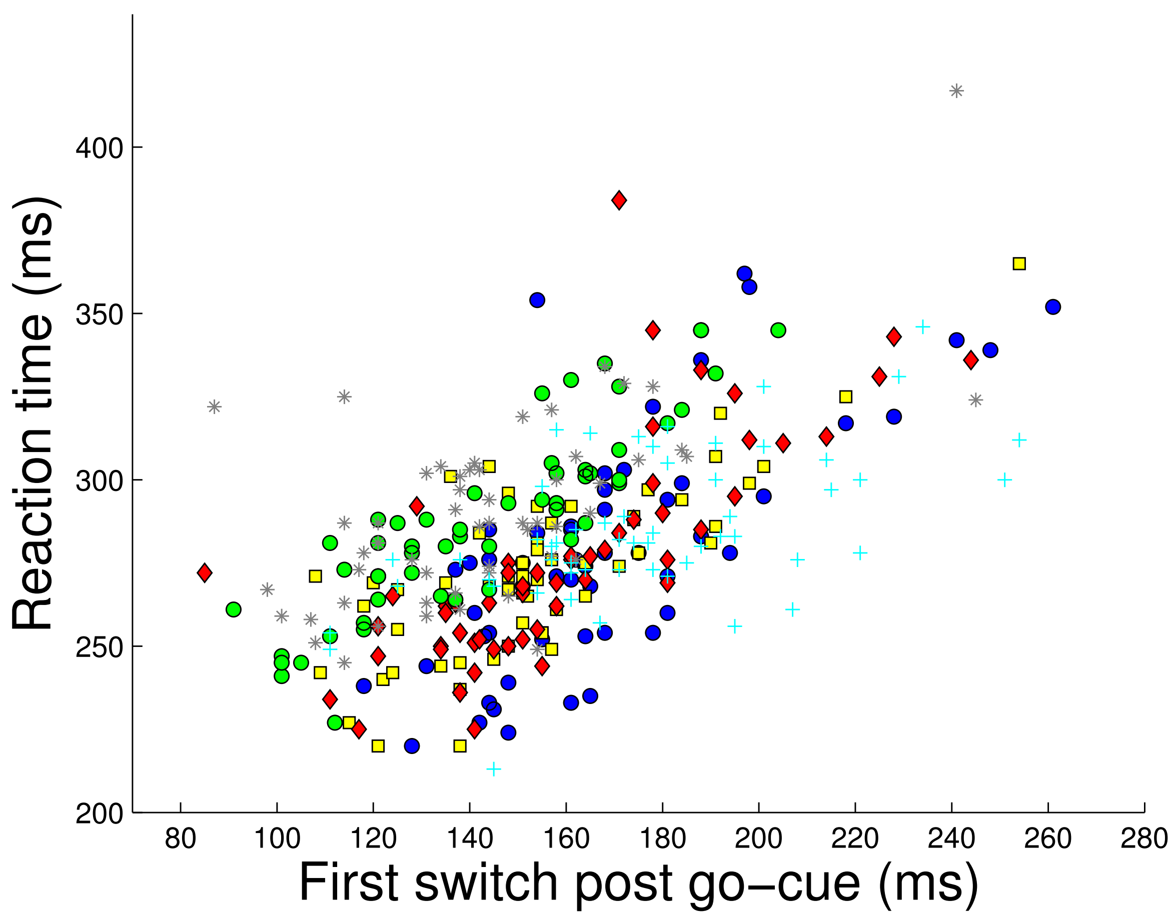
Event predictions



The first switch following “go” explains 52% of reaction time variance.

Petreska *et al.* *NIPS* 2011

Event predictions



The first switch following “go” explains 52% of reaction time variance.
Similar or better results in further sessions (and with other monkeys).

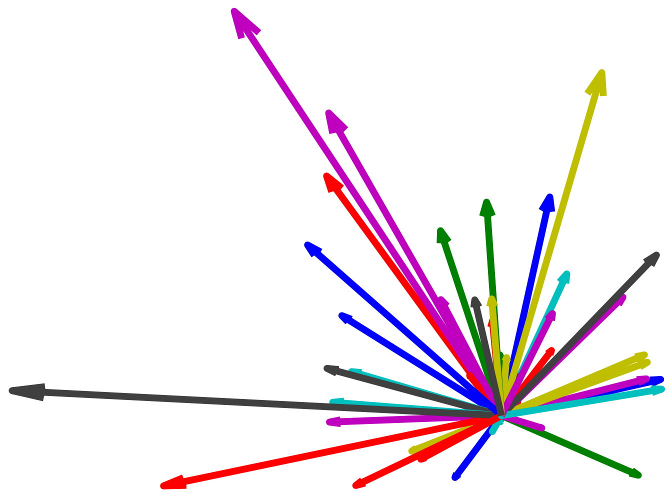
Petreska *et al. NIPS* 2011

Population vector – rise to threshold

Is this just about changes in firing rate?

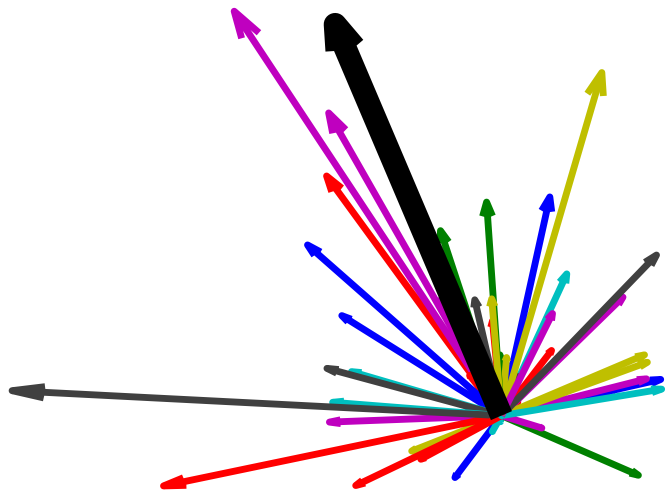
Population vector – rise to threshold

Is this just about changes in firing rate?



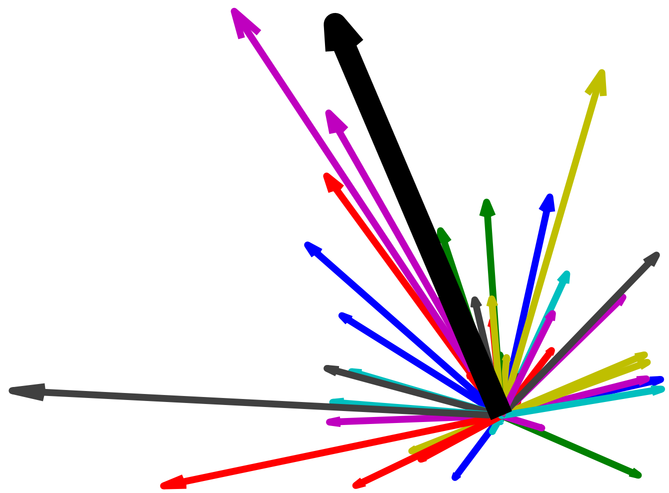
Population vector – rise to threshold

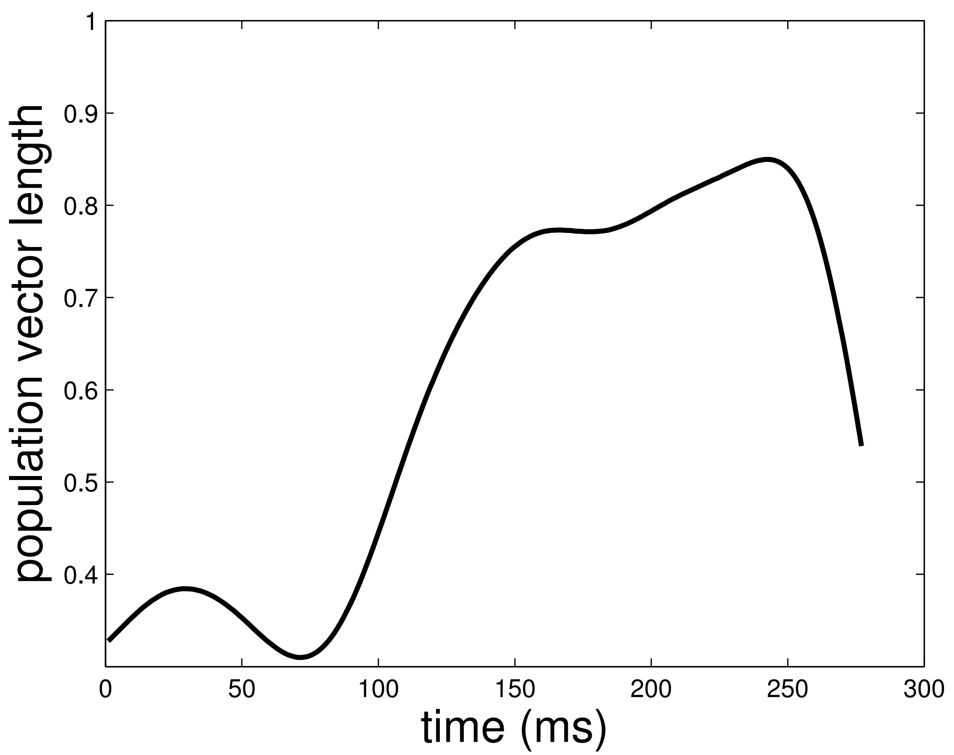
Is this just about changes in firing rate?



Population vector – rise to threshold

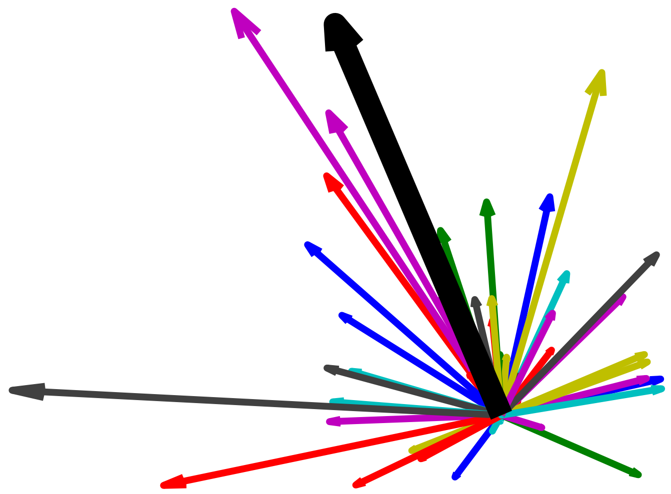
Is this just about changes in firing rate?

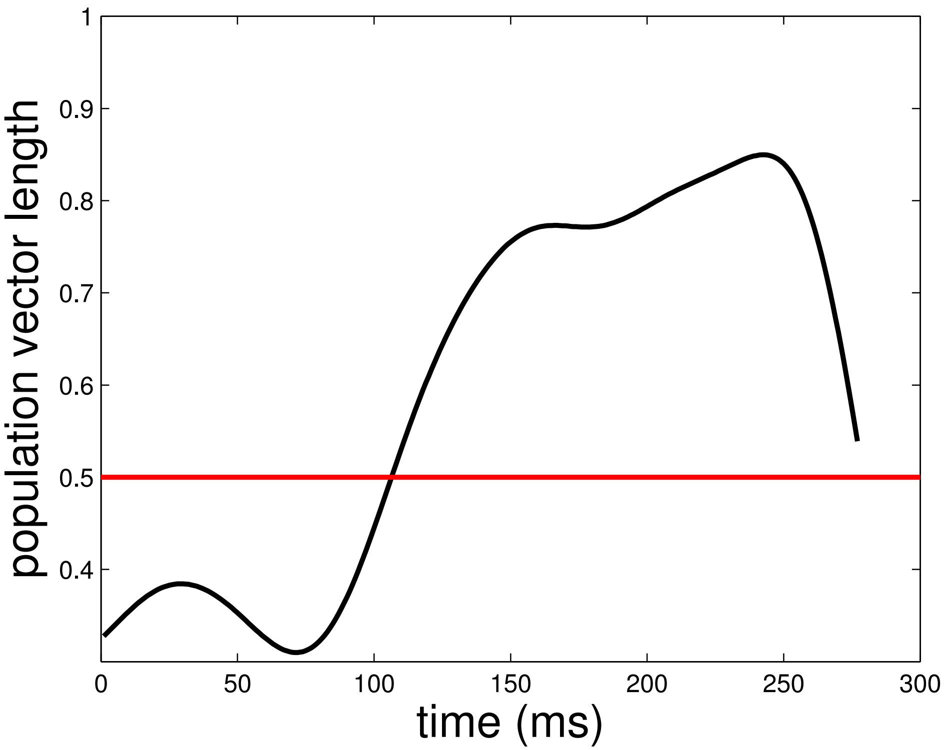




Population vector – rise to threshold

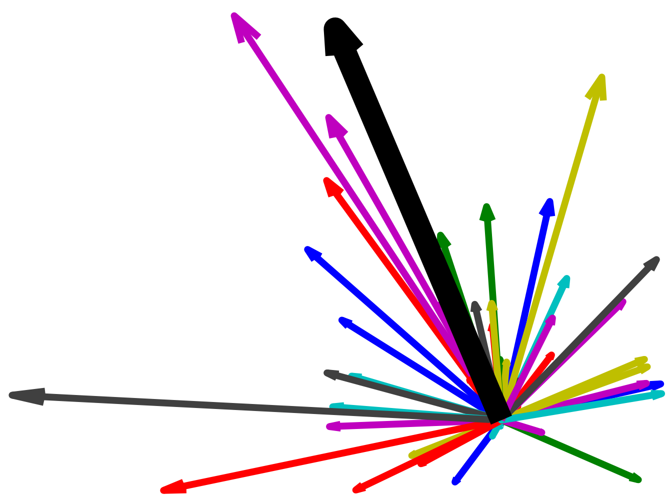
Is this just about changes in firing rate?

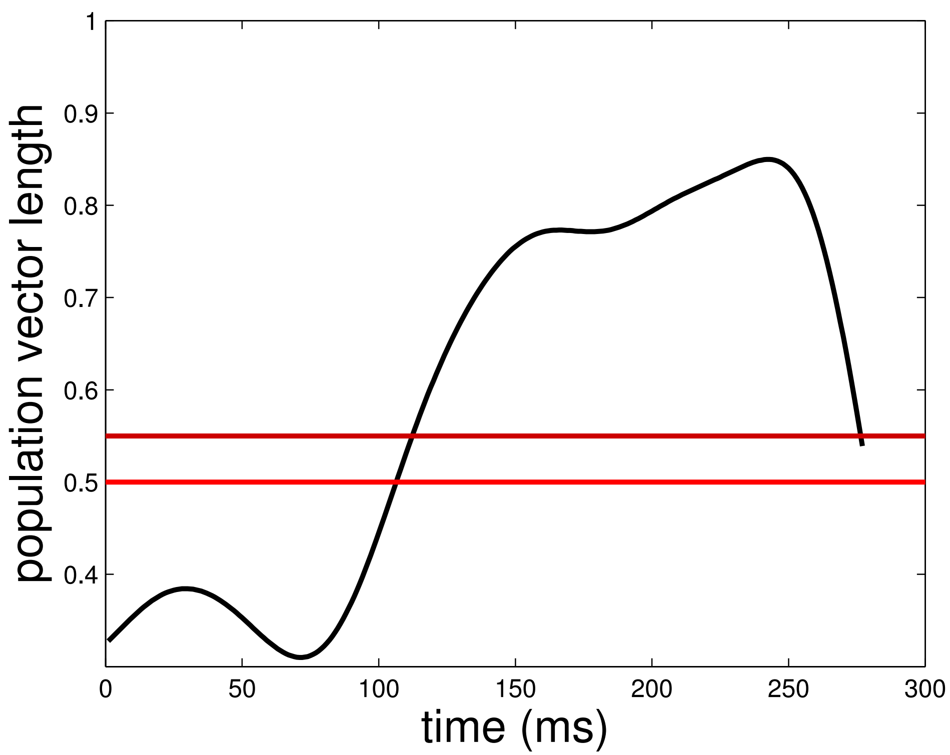




Population vector – rise to threshold

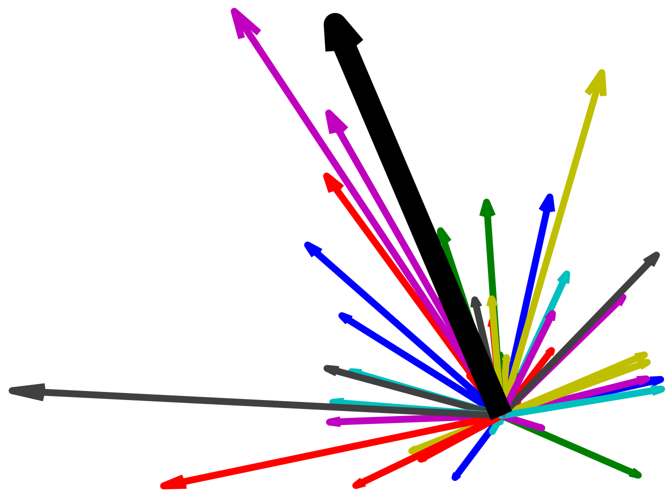
Is this just about changes in firing rate?

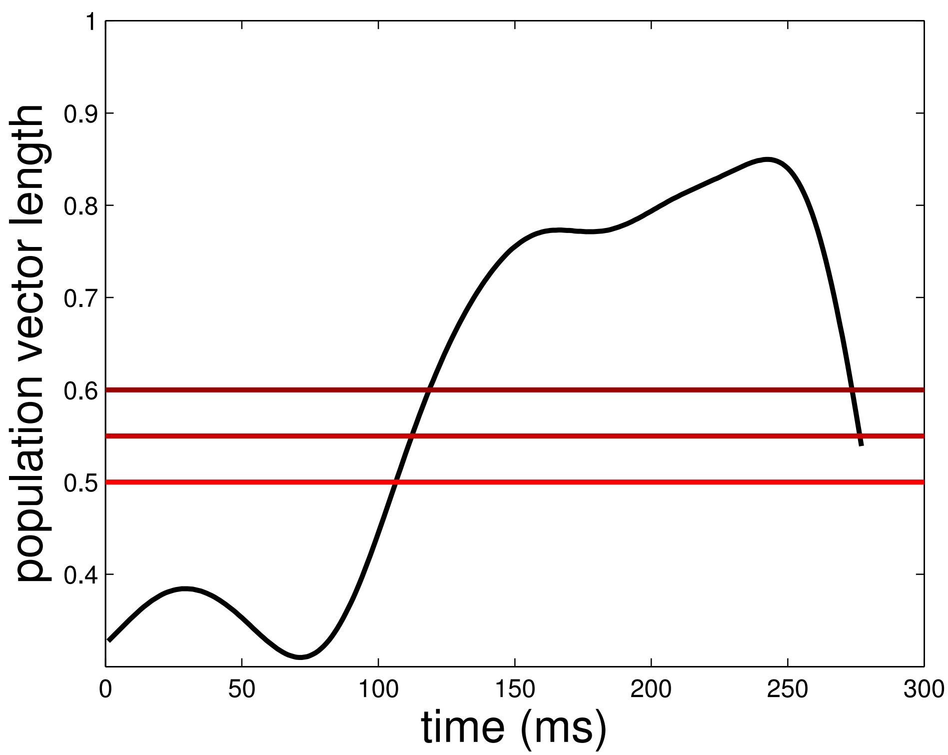




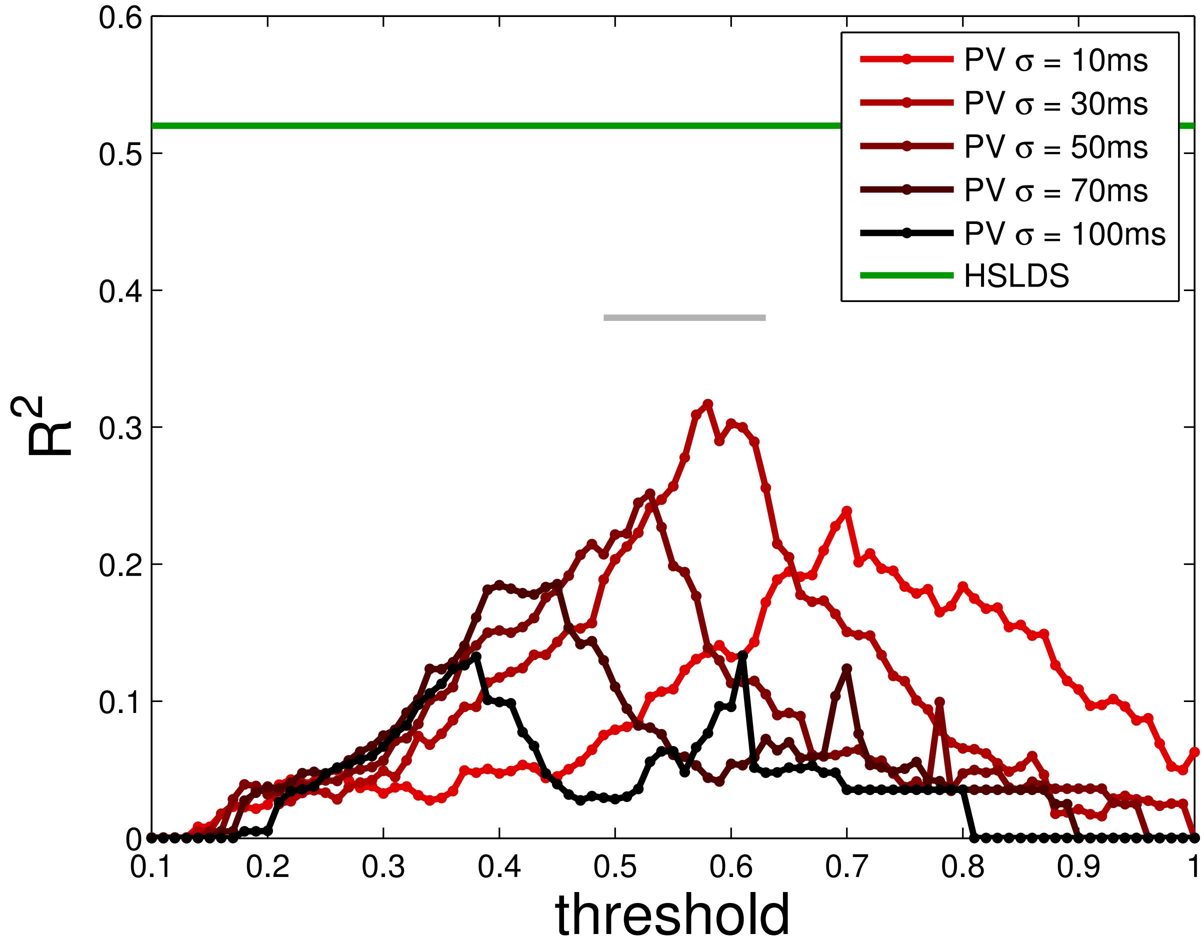
Population vector – rise to threshold

Is this just about changes in firing rate?

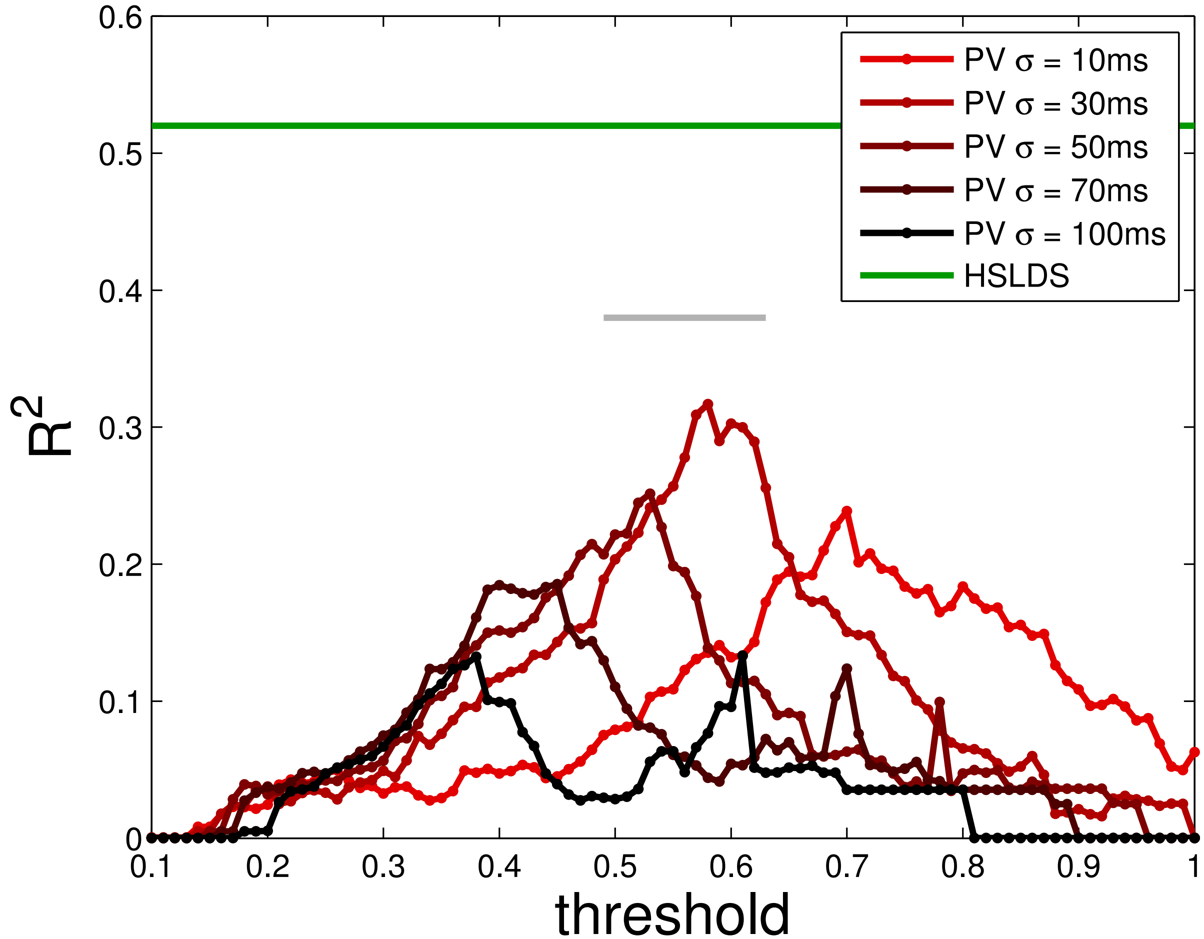




Population vector method

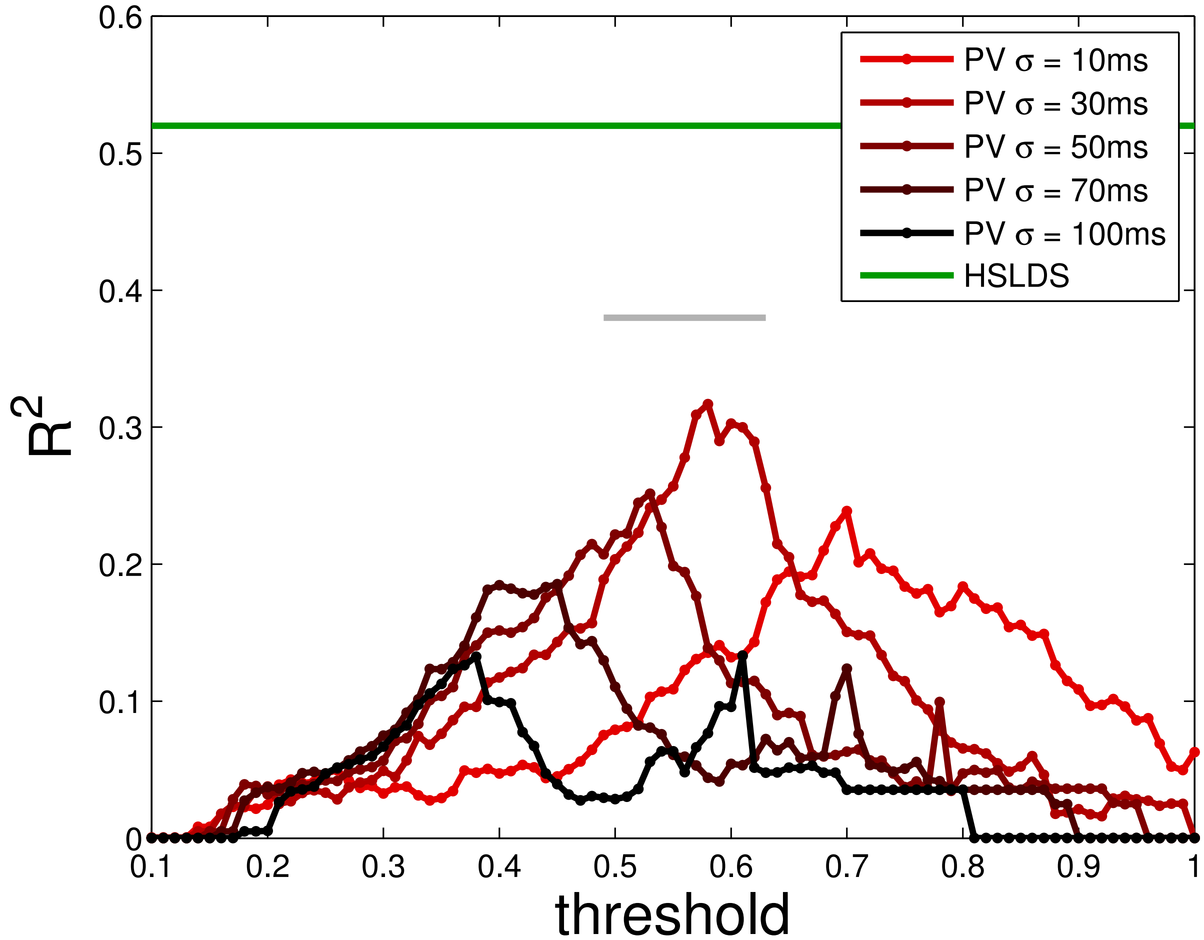


Population vector method



Allowing different thresholds per direction gives R^2 of 0.38 (< 0.52 for HSLDS).

Population vector method



Allowing different thresholds per direction gives R^2 of 0.38 (< 0.52 for HSLDS). Poorer performance despite considerable supervision.

Discussion

- Multineuron array recordings provide a substrate for empirical access to neural population dynamics on single trials.

Discussion

- Multineuron array recordings provide a substrate for empirical access to neural population dynamics on single trials.

- Latent space models provide a powerful and flexible set of models for estimating time-varying hidden states underlying neural firing.

Discussion

- Multineuron array recordings provide a substrate for empirical access to neural population dynamics on single trials.
- Latent space models provide a powerful and flexible set of models for estimating time-varying hidden states underlying neural firing.
- Firing rate dynamics in (pre)motor cortex seems to be smooth (and perhaps even shows signs of inertia).

Discussion

- Multineuron array recordings provide a substrate for empirical access to neural population dynamics on single trials.

- Latent space models provide a powerful and flexible set of models for estimating time-varying hidden states underlying neural firing.

- Firing rate dynamics in (pre)motor cortex seems to be smooth (and perhaps even shows signs of inertia).

- Do not know extent of network involved — other areas, basal ganglia, spine, cerebellum?

Discussion

- Multineuron array recordings provide a substrate for empirical access to neural population dynamics on single trials.

- Latent space models provide a powerful and flexible set of models for estimating time-varying hidden states underlying neural firing.

- Firing rate dynamics in (pre)motor cortex seems to be smooth (and perhaps even shows signs of inertia).

- Do not know extent of network involved — other areas, basal ganglia, spine, cerebellum?

- Do not know source of variability in neural state at go-cue.

Discussion

- Multineuron array recordings provide a substrate for empirical access to neural population dynamics on single trials.
- Latent space models provide a powerful and flexible set of models for estimating time-varying hidden states underlying neural firing.
- Firing rate dynamics in (pre)motor cortex seems to be smooth (and perhaps even shows signs of inertia).
- Do not know extent of network involved — other areas, basal ganglia, spine, cerebellum?
- Do not know source of variability in neural state at go-cue.
- An unsupervised approach to dynamical segmentation finds behaviourally relevant transitions on individual trials.

Acknowledgements

Krishna Shenoy

Byron Yu

Mark Churchland

Afsheen Afshar

Biljana Petreska

Jakob Macke

Lars Büsing

John Cunningham

Matt Kaufmann

Zuley Rivera

Rachel Kalmar

Stephen Ryu

Gopal Santhanam

Alex Lerchner