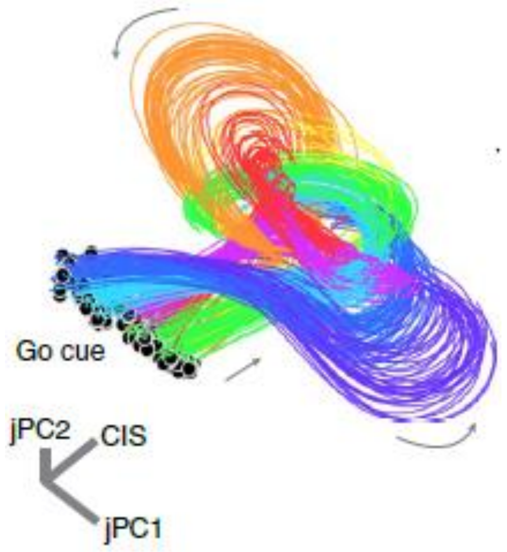
Inferring single-trial neural population dynamics using sequential auto-encoders

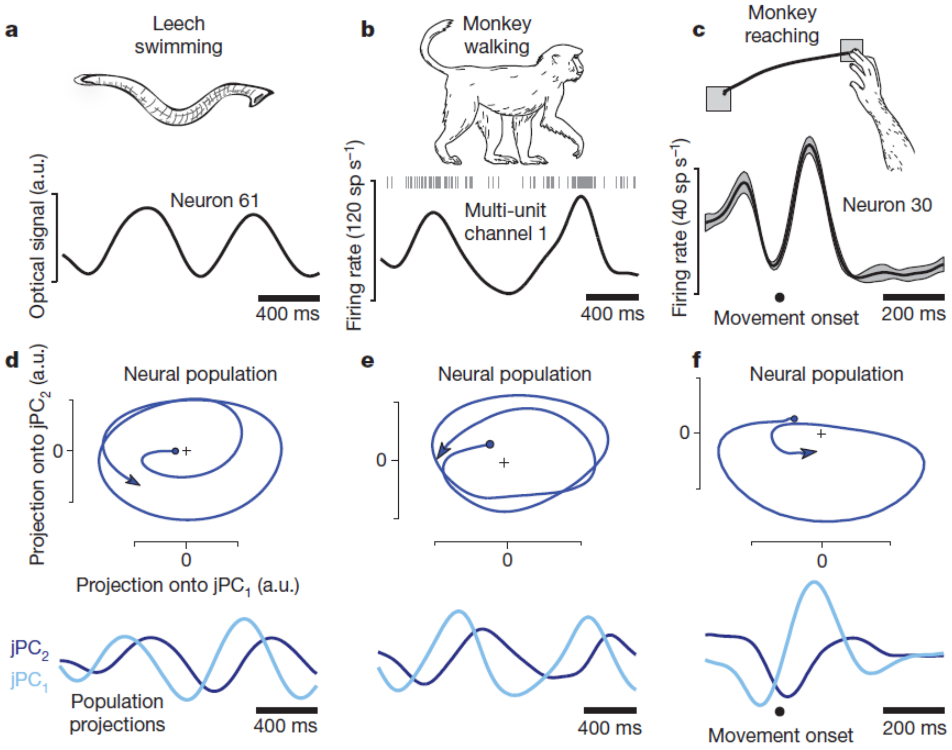
Pandarinath et al., 2018, *Nature methods*

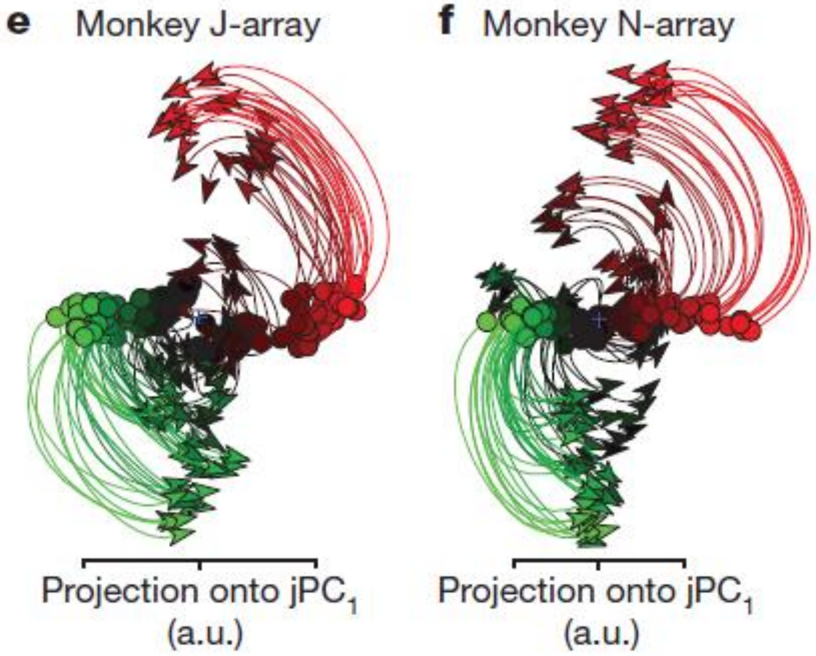
Nhat Le, NeuroComp meeting, Dec 11, 2018



Motivation

- Dynamical systems perspective: dynamical systems underlie the pattern of neural populations





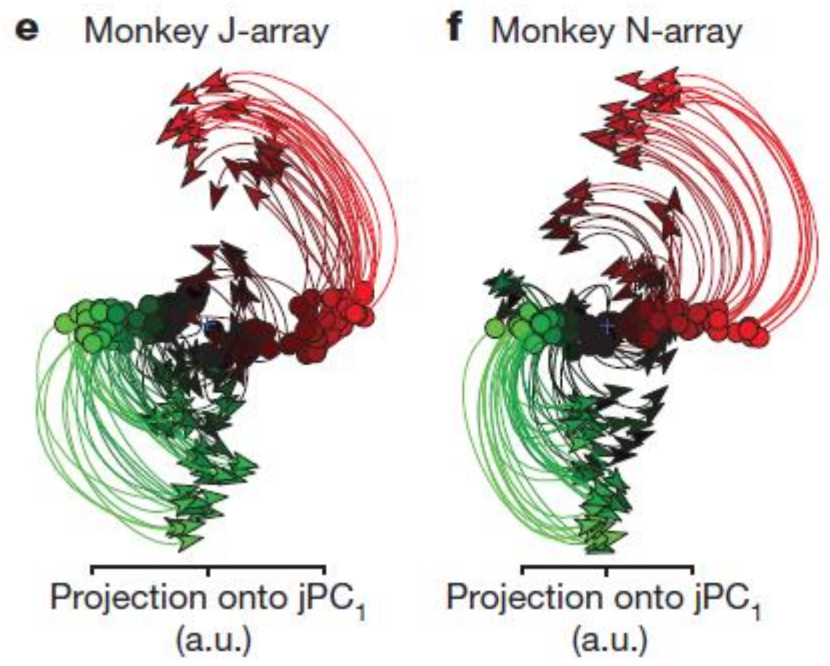
Churchland et al., 2012, *Nature*

Motivation

- Dynamical systems perspective: dynamical systems underlie the pattern of neural populations

Problem: trajectories often computed based on trial averages

→ Can we uncover underlying dynamics of *single trials*?



Churchland et al., 2012, *Nature*

Part I: The LFADS basic architecture

1. LFADS assumes an underlying ***dynamical system*** (***recurrent neural network***) that generates spike trains

2. LFADS as a form of ***factor analysis***

3. LFADS as a ***variational autoencoder***

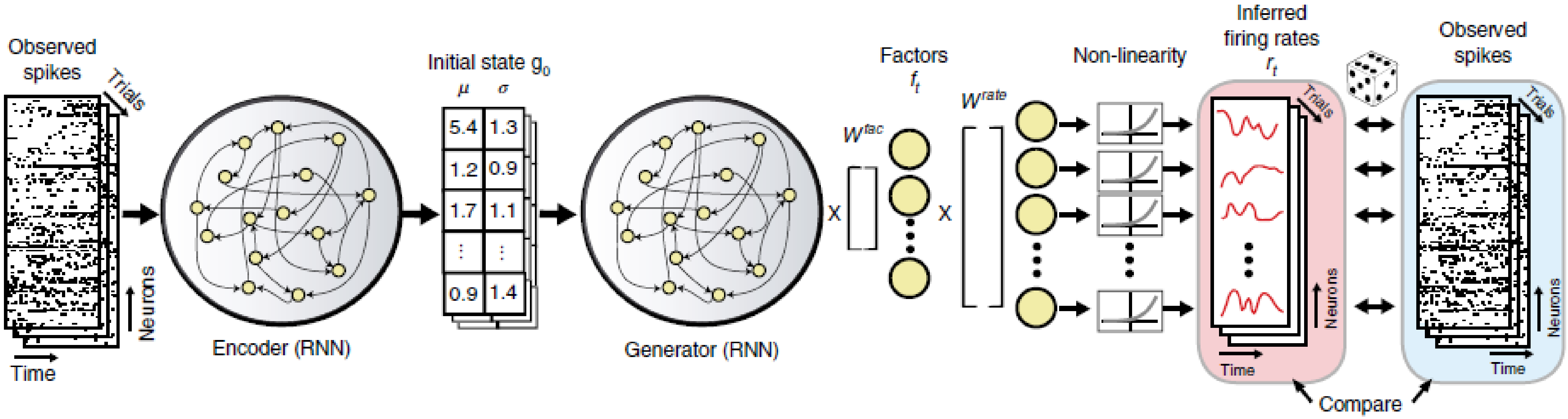
Part I: The LFADS basic architecture

1. **LFADS assumes an underlying *dynamical system* (*recurrent neural network*) that generates spike trains**

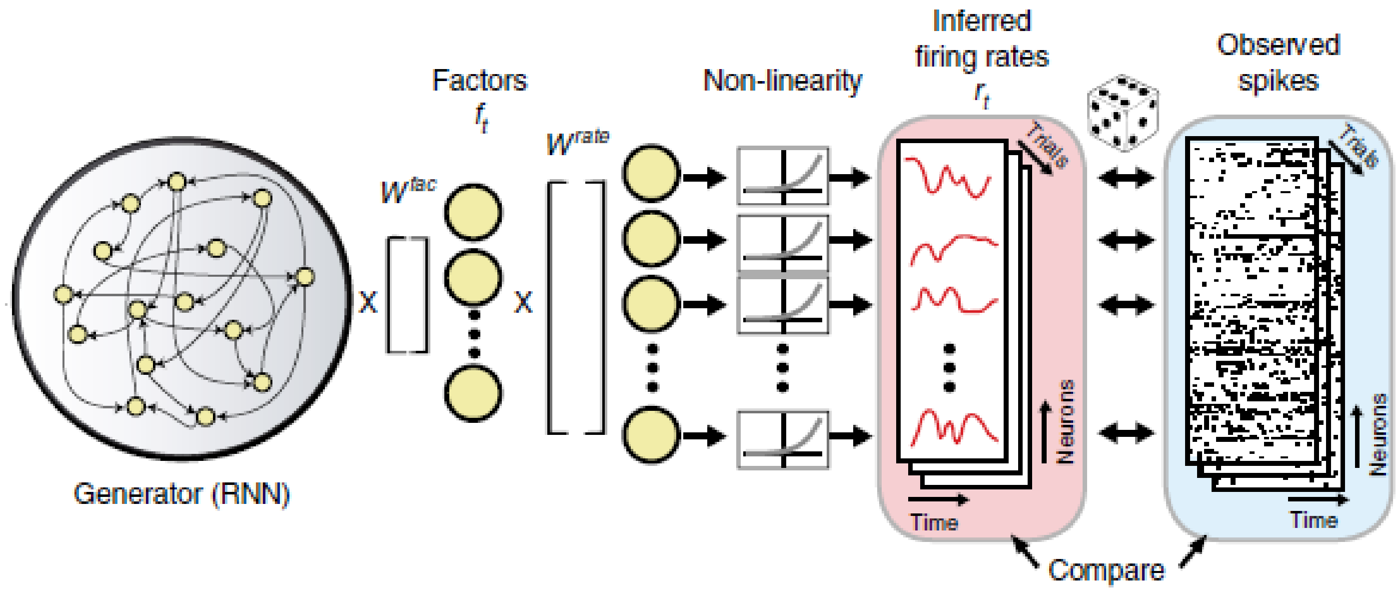
2. LFADS as a form of ***factor analysis***

3. LFADS as a ***variational autoencoder***

LFADS basic architecture



LFADS basic architecture



Underlying
dynamical system
g

which evolves
according to a rule

Firing rate
$\boldsymbol{r_t}$

$$\dot{g}(t) = F(g(t), u(t))$$

LFADS basic architecture

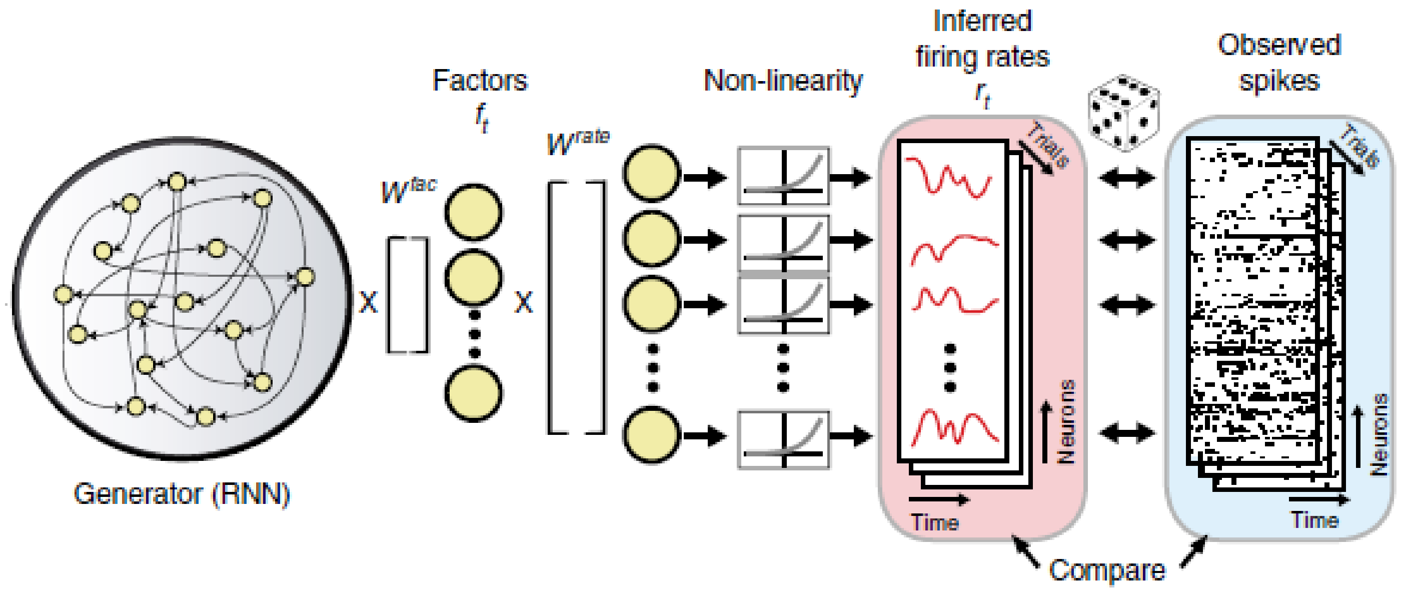
Observed spikes depend on

(1) Underlying dynamics (***F***)

(2) Initial conditions ($\boldsymbol{g_0}$)

(3) Inputs from other brain areas (***u***)

(4) Spiking variability



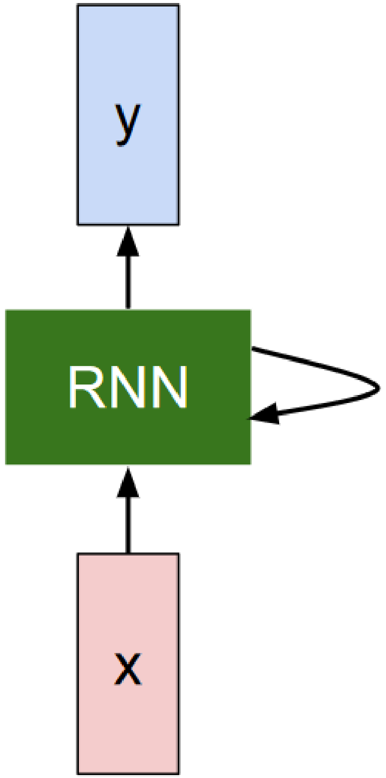
Underlying **dynamical system** $\boldsymbol{g_t}$, which evolves according to a rule

Firing rate $\boldsymbol{r_t}$

$$\dot{g}(t) = F(g(t), u(t))$$

To model underlying dynamics:

Recurrent neural networks (simple)



g_0: initial hidden state

hidden state

input

$$g_t = \tanh(W^g[g_{t-1}, x_t])$$

$$y_t = W^y g_t$$

output

Yeung, CS231n (Stanford) 2017, lecture13

RNN variant: Gated Recurrent Unit (GRU)

Simple

$$g_t = \tanh(W^g[g_{t-1}, x_t])$$

GRU

$$c_t = \tanh(W^c[x_t, r_t \odot g_{t-1}])$$

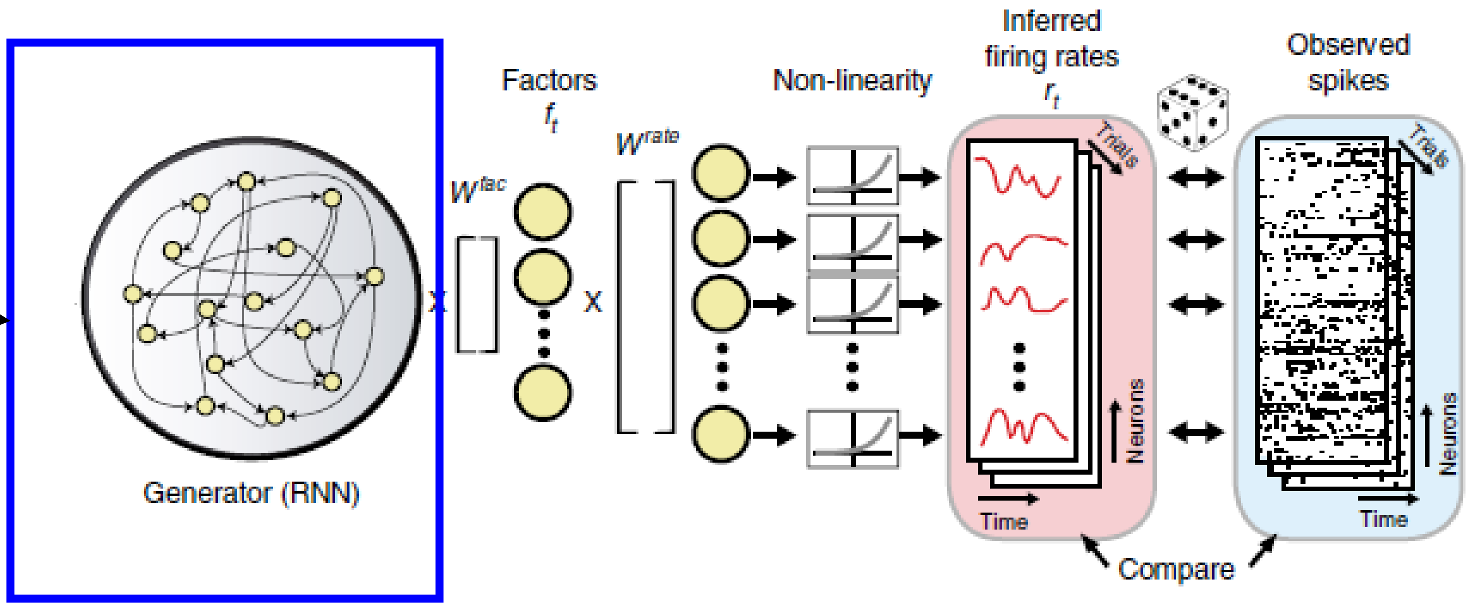
$$g_t = u_t \odot g_{t-1} + (1 - u_t) \odot c_t$$

- *A combination of previous state '**memory**' and **updated state***

- *Prevents vanishing gradients*

LFADS basic architecture

A **GRU**



Underlying
dynamical system
$\boldsymbol{g_t}$,
which evolves
according to a rule

Firing rate
$\boldsymbol{r_t}$

$$\dot{g}(t) = F(g(t), u(t))$$

Part I: The LFADS basic architecture

1. LFADS assumes an underlying ***dynamical system*** (***recurrent neural network***) that generates spike trains

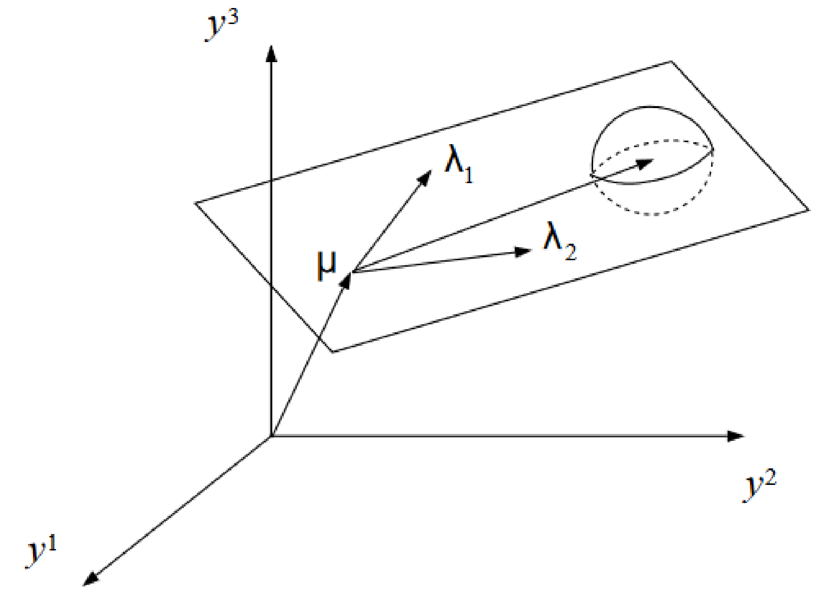
2. LFADS as a form of ***factor analysis***

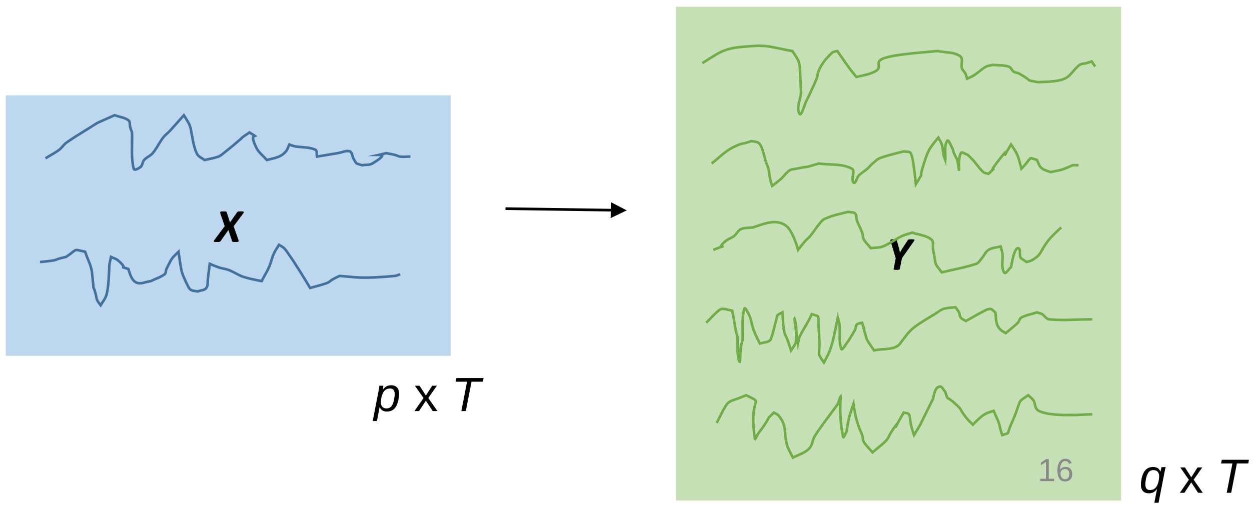
3. LFADS as a ***variational autoencoder***

Factor Analysis

Assumes data $\boldsymbol{y}$ (dimension q) is generated by a latent variable $\boldsymbol{x}$ (dimension p) where $p < q$

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





Roweis, 2004. Factor analysis and PCA

Previous methods to uncover latent factors

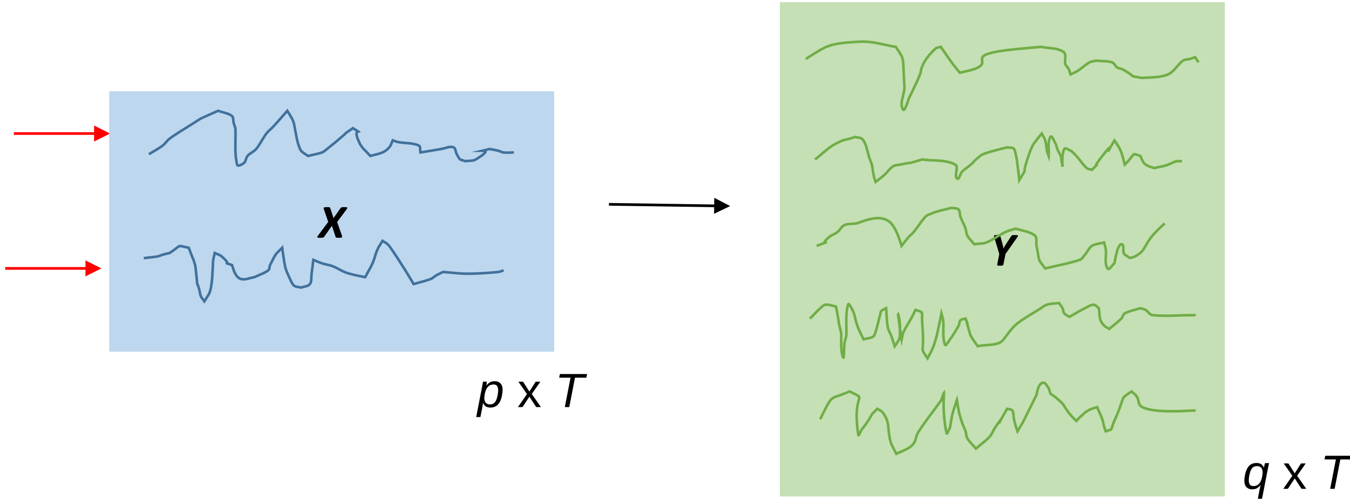
$$\mathbf{y}_{:,t} \mid \mathbf{x}_{:,t} \sim \mathcal{N}\left(C\mathbf{x}_{:,t} + \mathbf{d},\ R\right),$$

- *Gaussian process factor analysis*: each dimension of x is a Gaussian process

$$\mathbf{x}_{i,:} \sim \mathcal{N}\left(\mathbf{0},\ K_i\right)$$

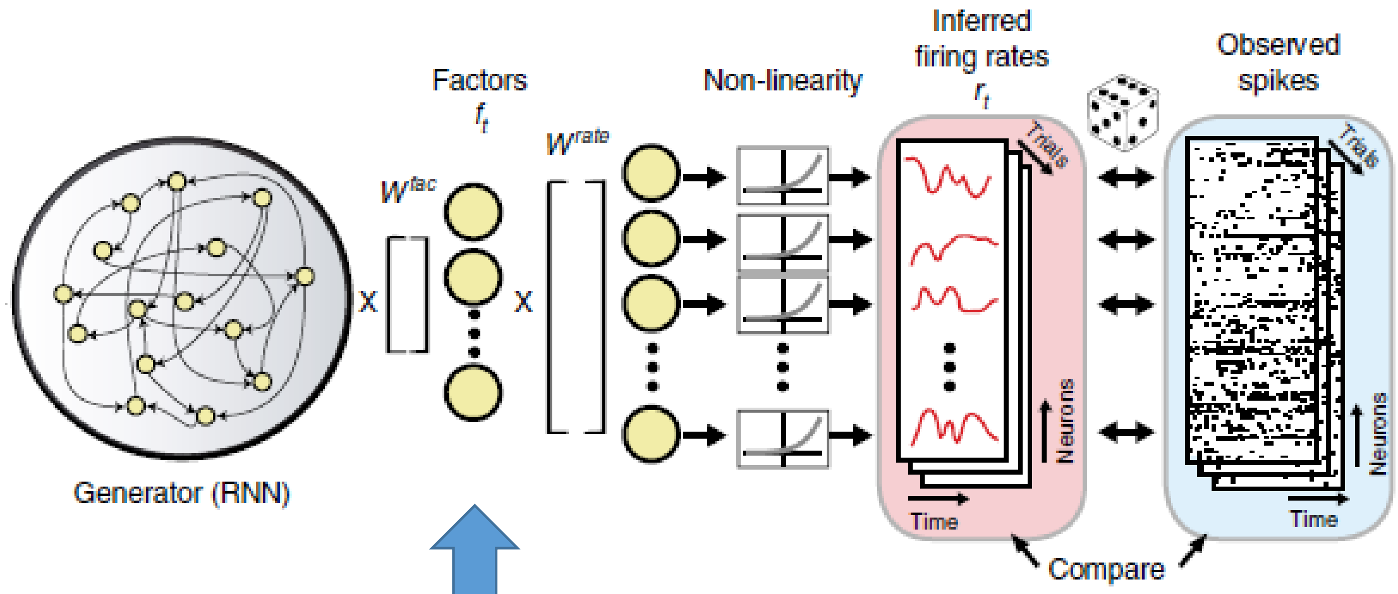
- *Poisson feed-forward linear dynamical system (PfLDS)*: *y* is generated from *x* through a rate λ and a noise model *P*

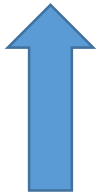
$$x_{rti} | \mathbf{z}_{rt} \sim \mathcal{P}_\lambda\left(\lambda_{rti} = [f_\psi(\mathbf{z}_{rt})]_i\right)$$



Roweis, 2004. Factor analysis and PCA

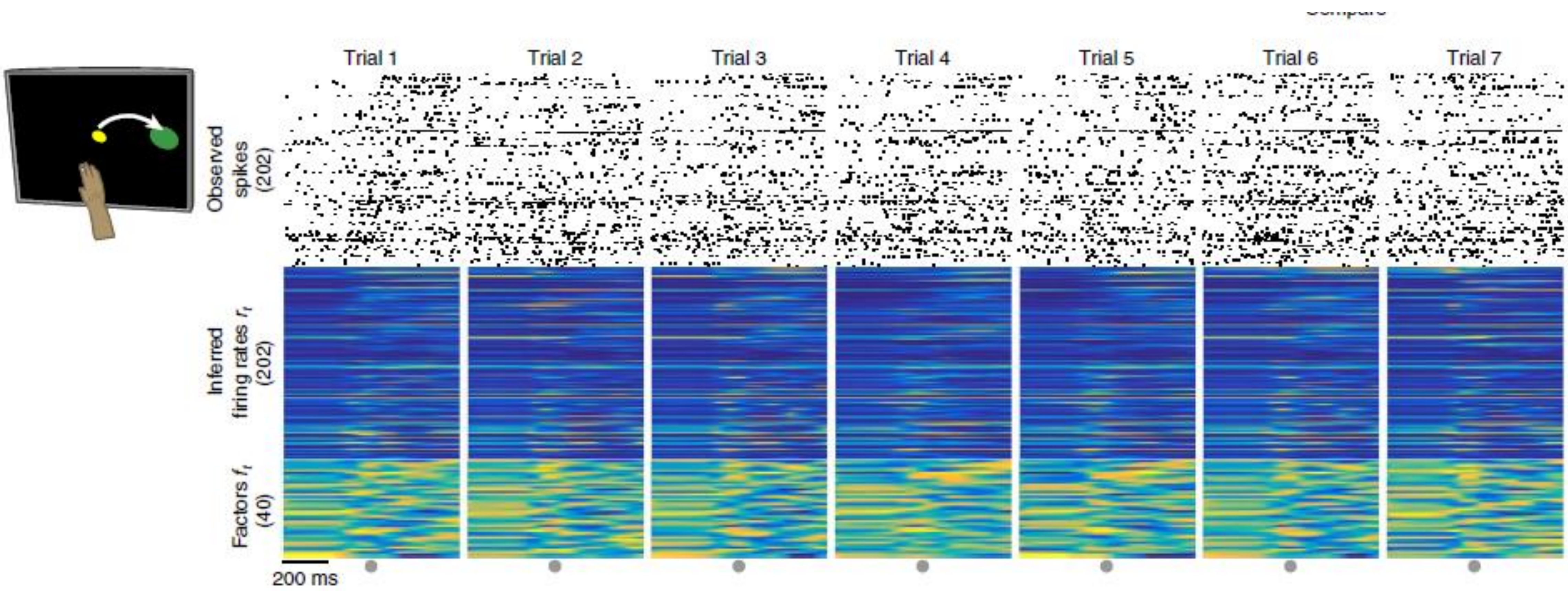
LFADS basic architecture

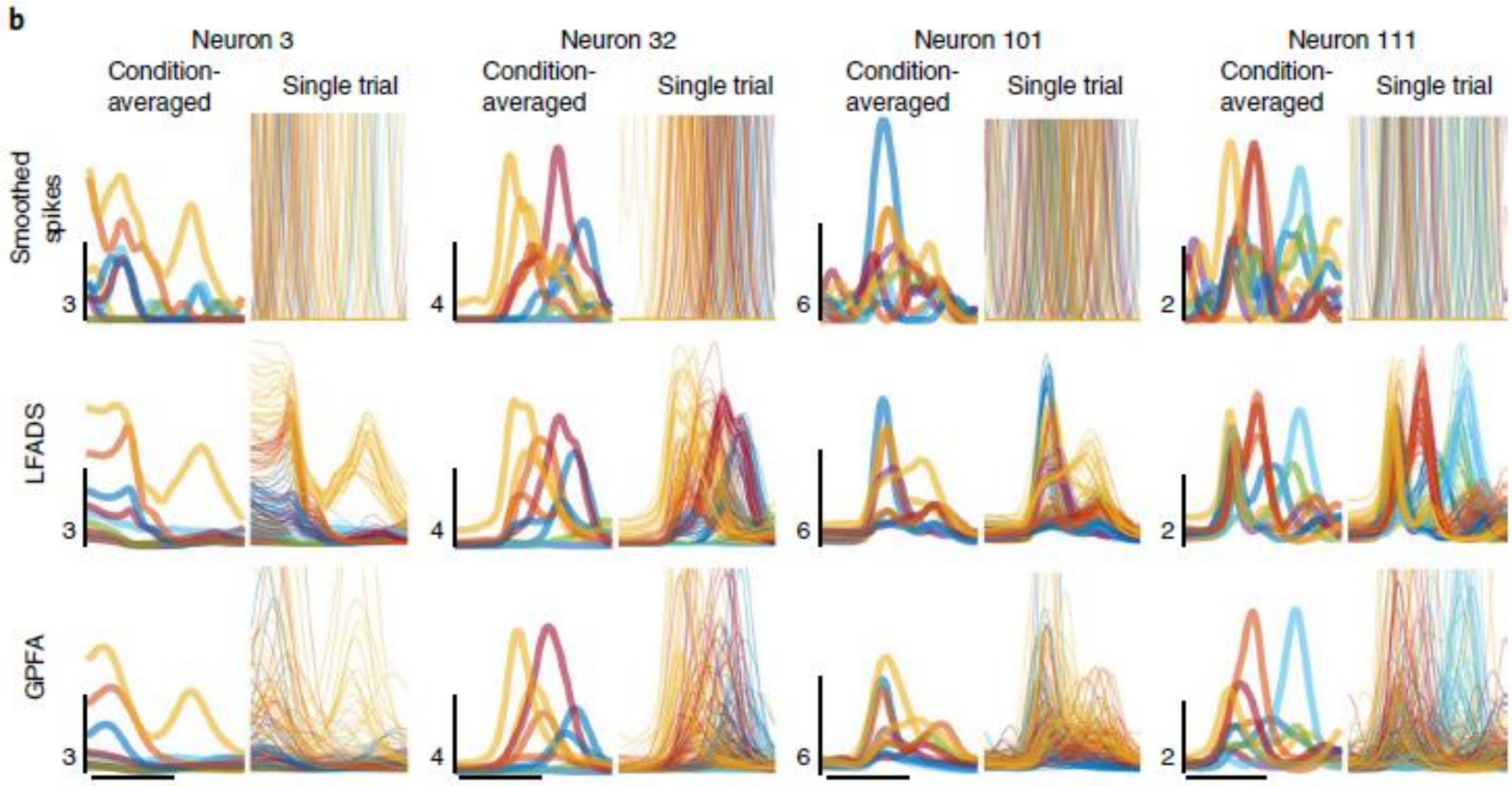




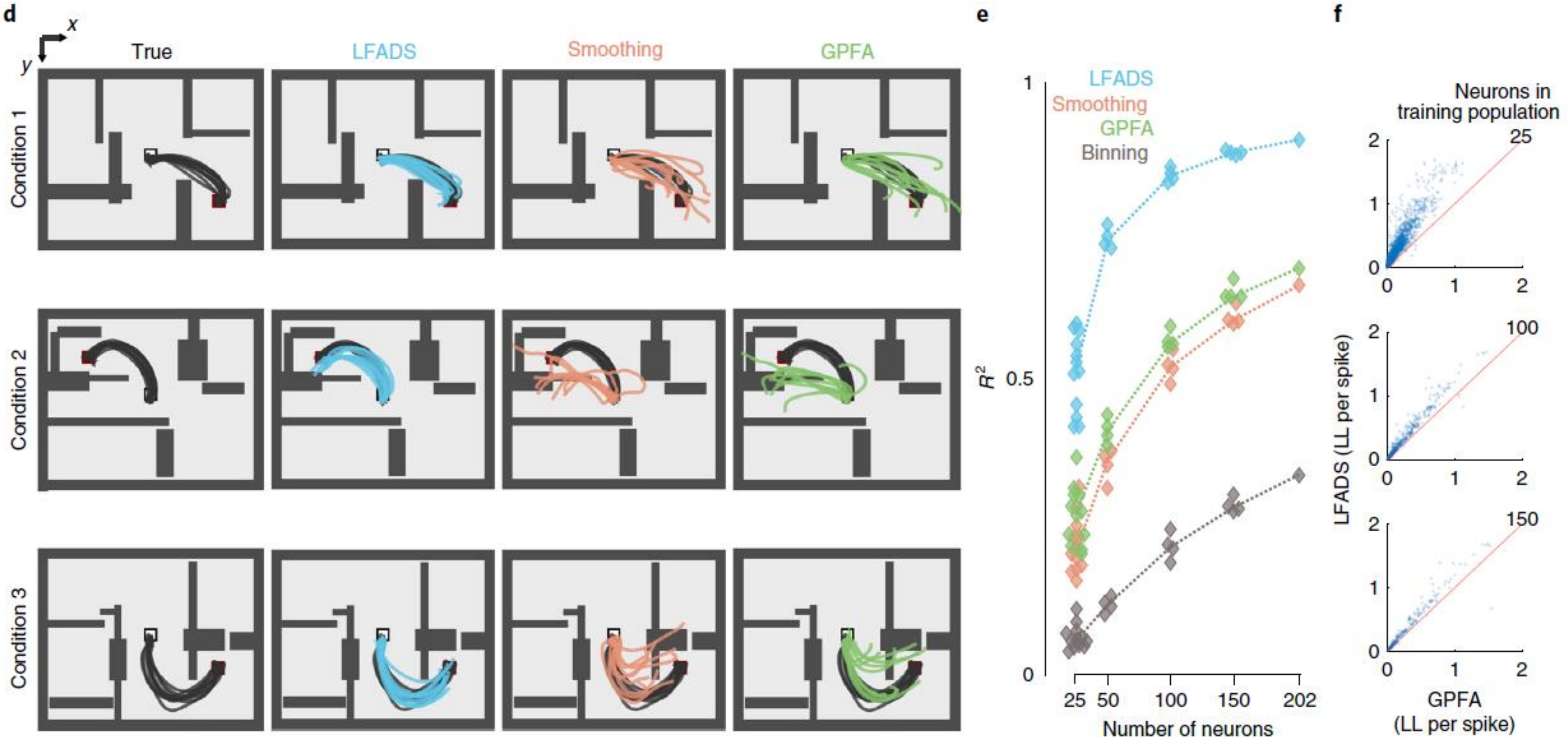
Low-dimensional factors which generate firing rates

LFADS can recover low-dimensional factors





b
Neuron 3
Neuron 32
Neuron 101
Neuron 111
Condition-averaged
Single trial
Smoothed spikes
LFADS
GPFA
3
4
6
2



d
x
y
True
LFADS
Smoothing
GPFA
Condition 1
Condition 2
Condition 3
e
LFADS
Smoothing
GPFA
Binning
R^2
1
0.5
0
25
50
100
150
202
Number of neurons
f
Neurons in training population
25
100
150
LFADS (LL per spike)
GPFA (LL per spike)
0
1
2

Part I: The LFADS basic architecture

1. LFADS assumes an underyling ***dynamical system*** (***recurrent neural network***) that generates spike trains

2. LFADS as a form of ***factor analysis***

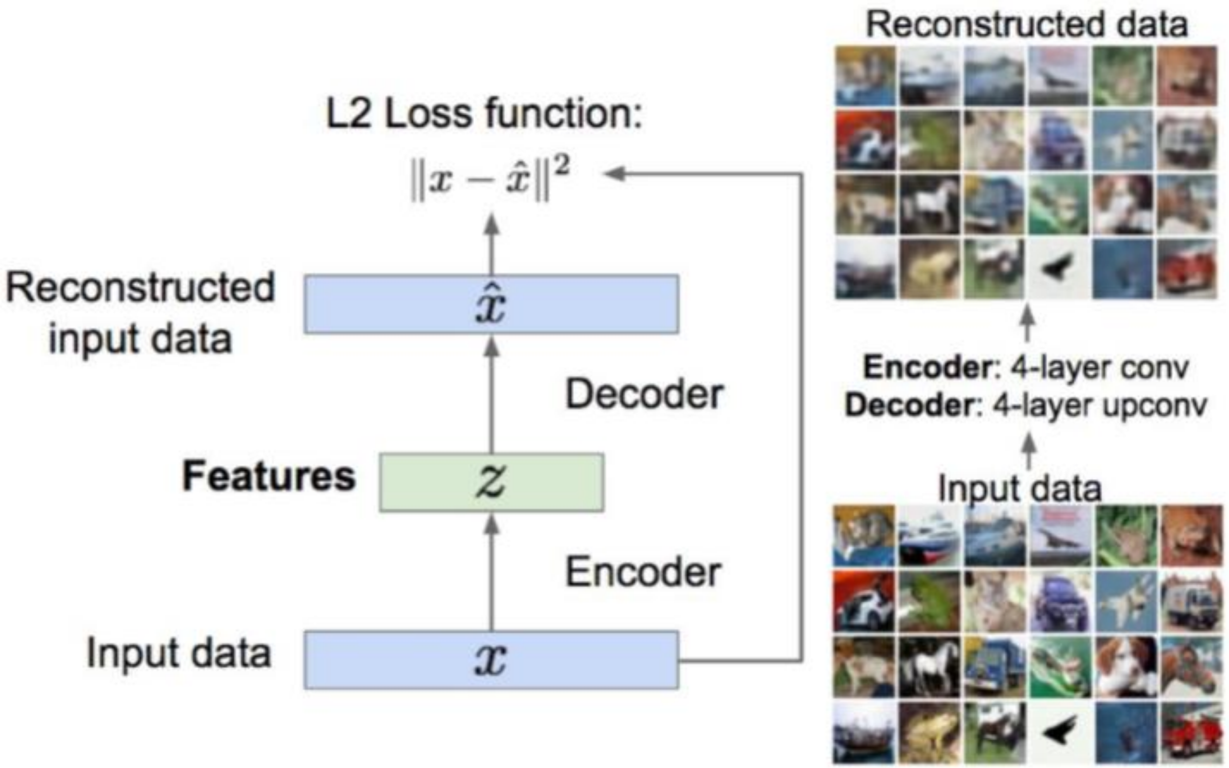
3. LFADS as a ***variational autoencoder***

Autoencoders

A form of **unsupervised learning**

Uncover hidden structure of data (in LFADS: spike trains)

After training, features (factors) will represent a compressed structure of the input data



Autoencoders
(Feature learning)

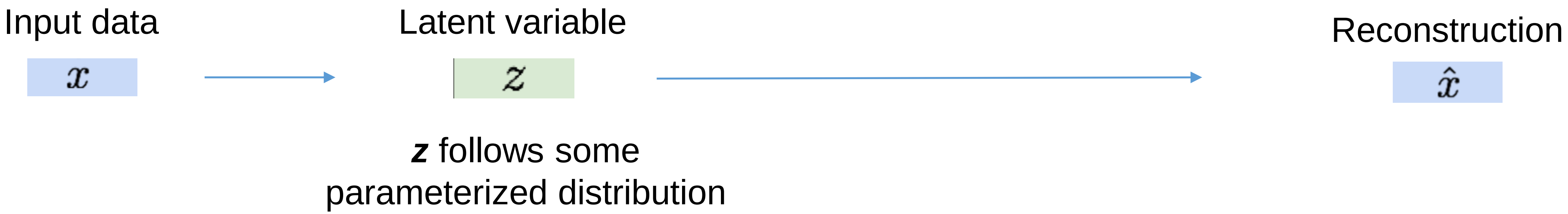
Yeung, CS231n (Stanford) 2017, lecture13

Variational Autoencoder (VAE)

- Goal: try to learn the probability distribution *p*(*x*) that generates training data *x*

VAEs define intractable density function with latent **z**:

$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$

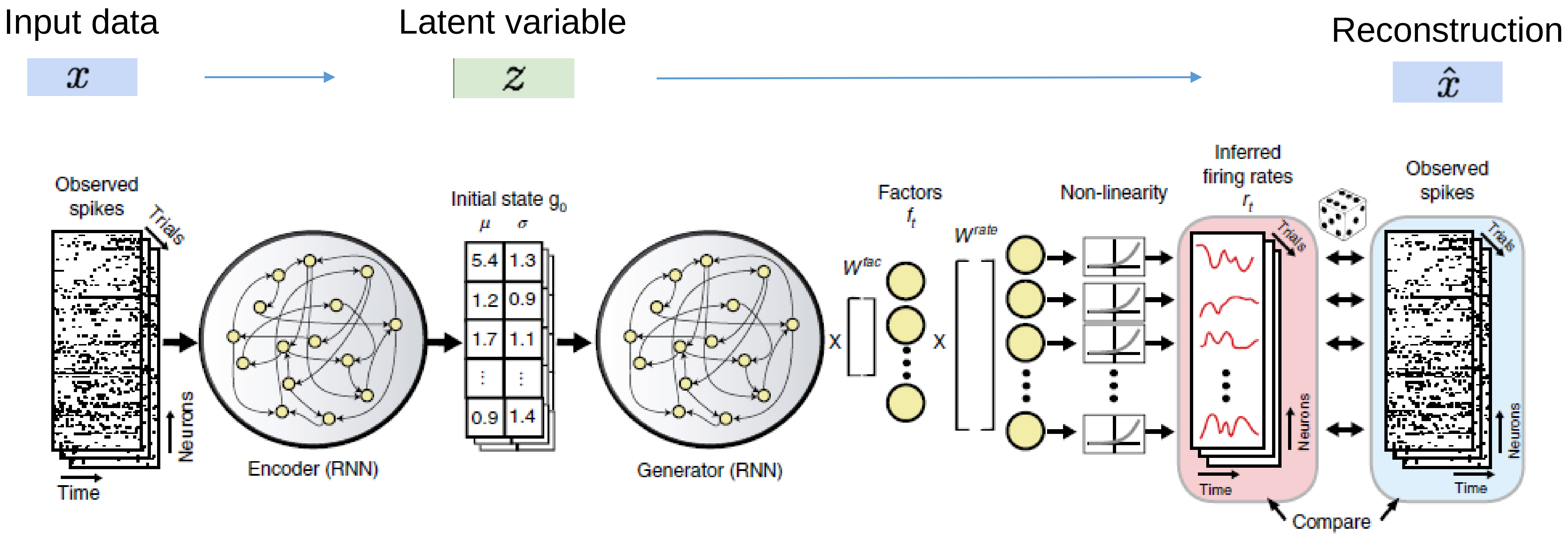


Variational Autoencoder (VAE)

- Goal: try to learn the probability distribution $p(x)$ that underlie training data x

VAEs define intractable density function with latent **z**:

$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$



Variational Autoencoder (VAE): Training

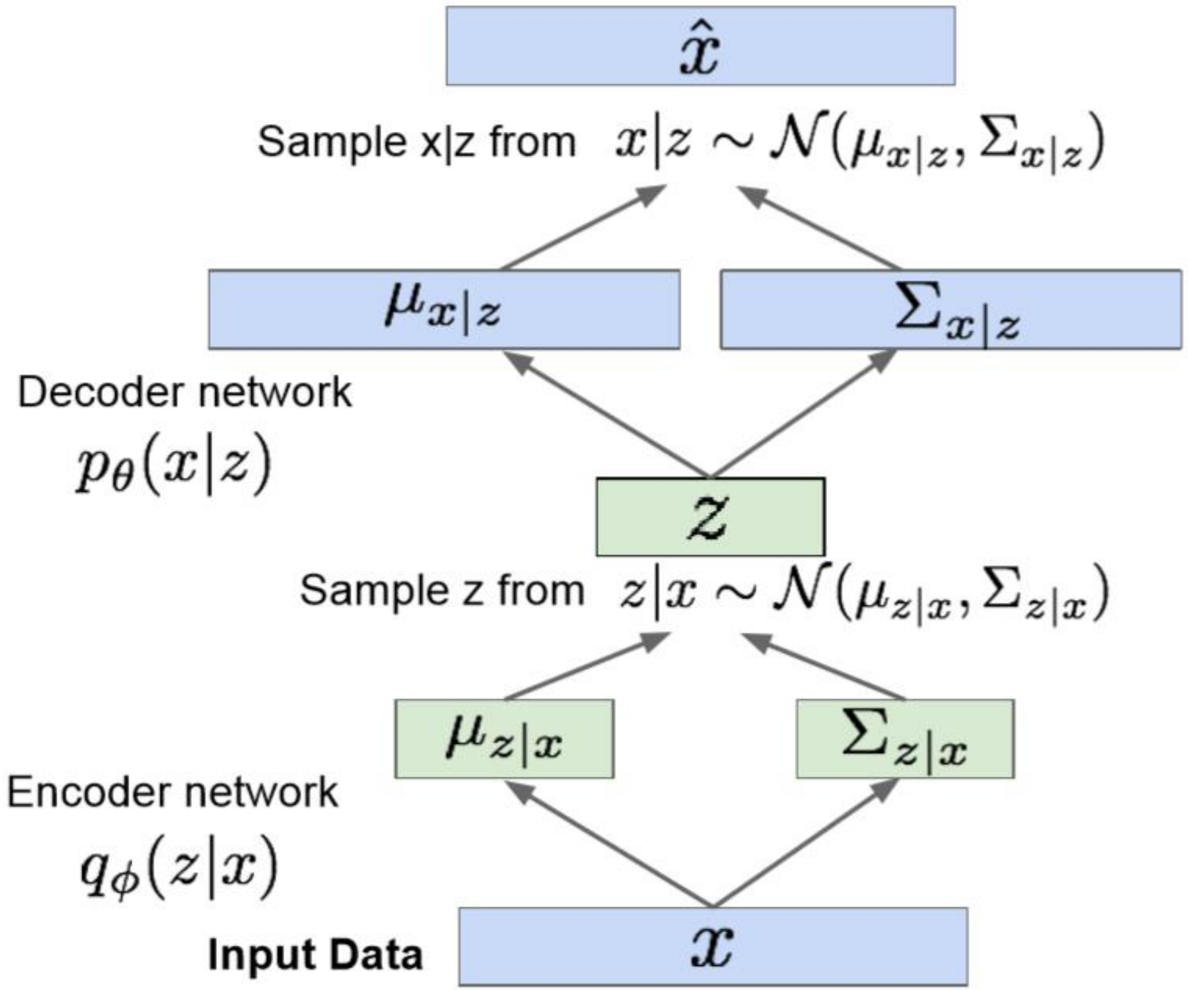
Reconstructed spike trains

RNN (decoder)

g_0

RNN (encoder)

Spike trains



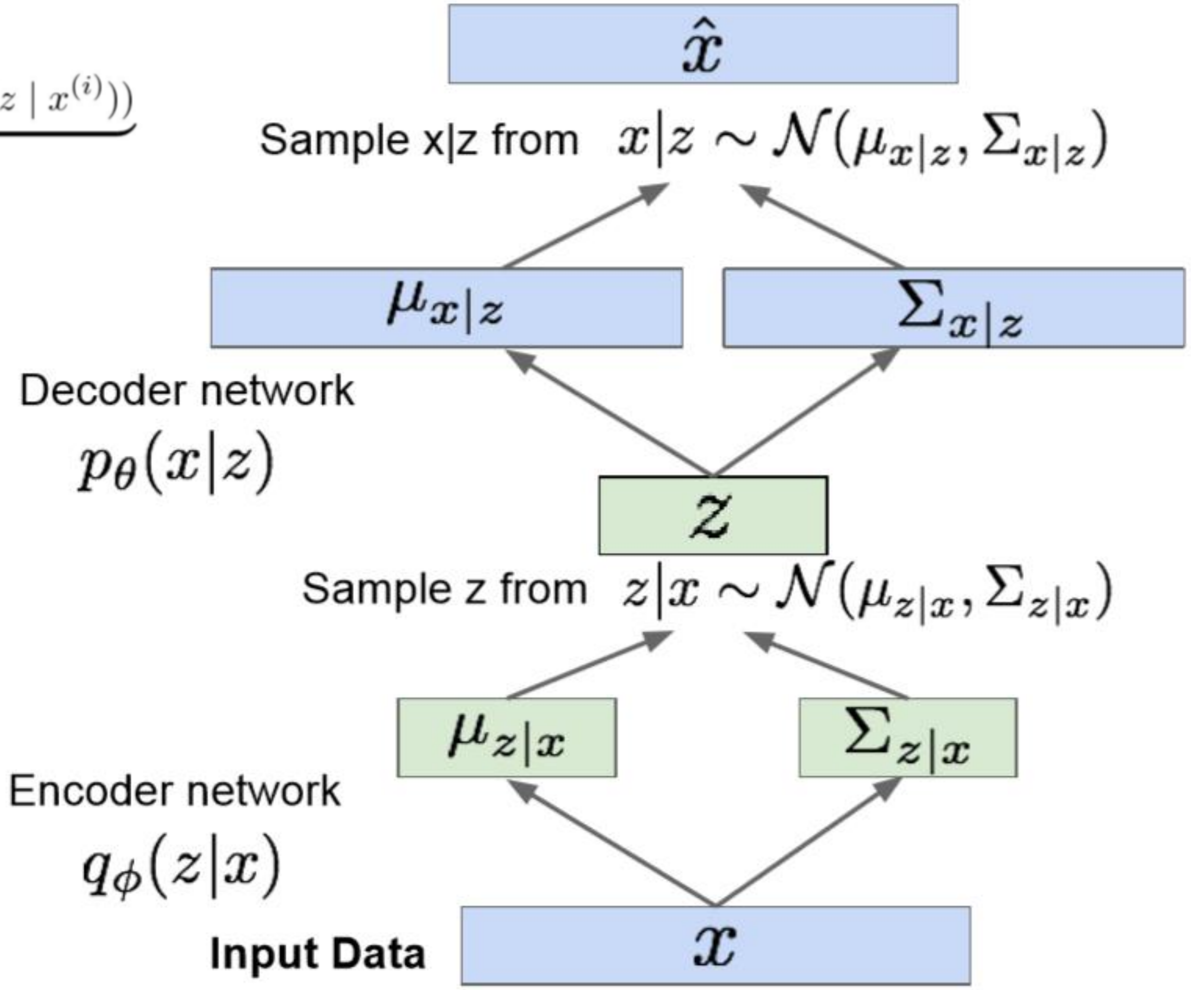
Yeung, CS231n (Stanford) 2017, lecture13

Variational Autoencoder (VAE): Training

$$\log p_\theta(x^{(i)}) =$$

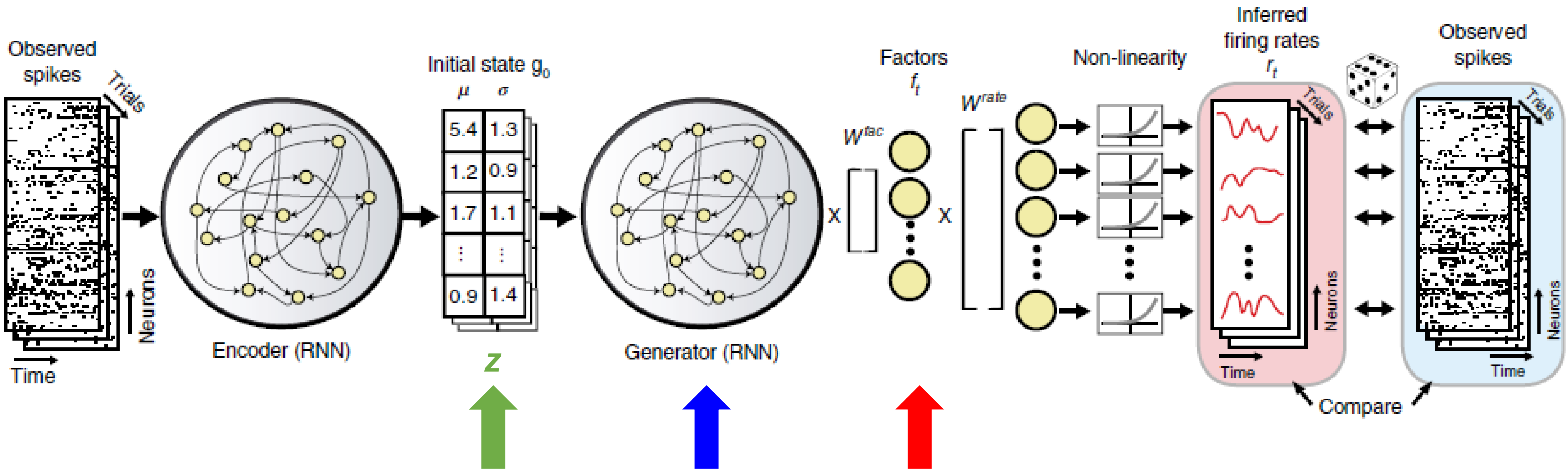
$$= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}$$

Tractable lower bound which we can take gradient of and optimize! ($p_\theta(x|z)$ differentiable, KL term differentiable)



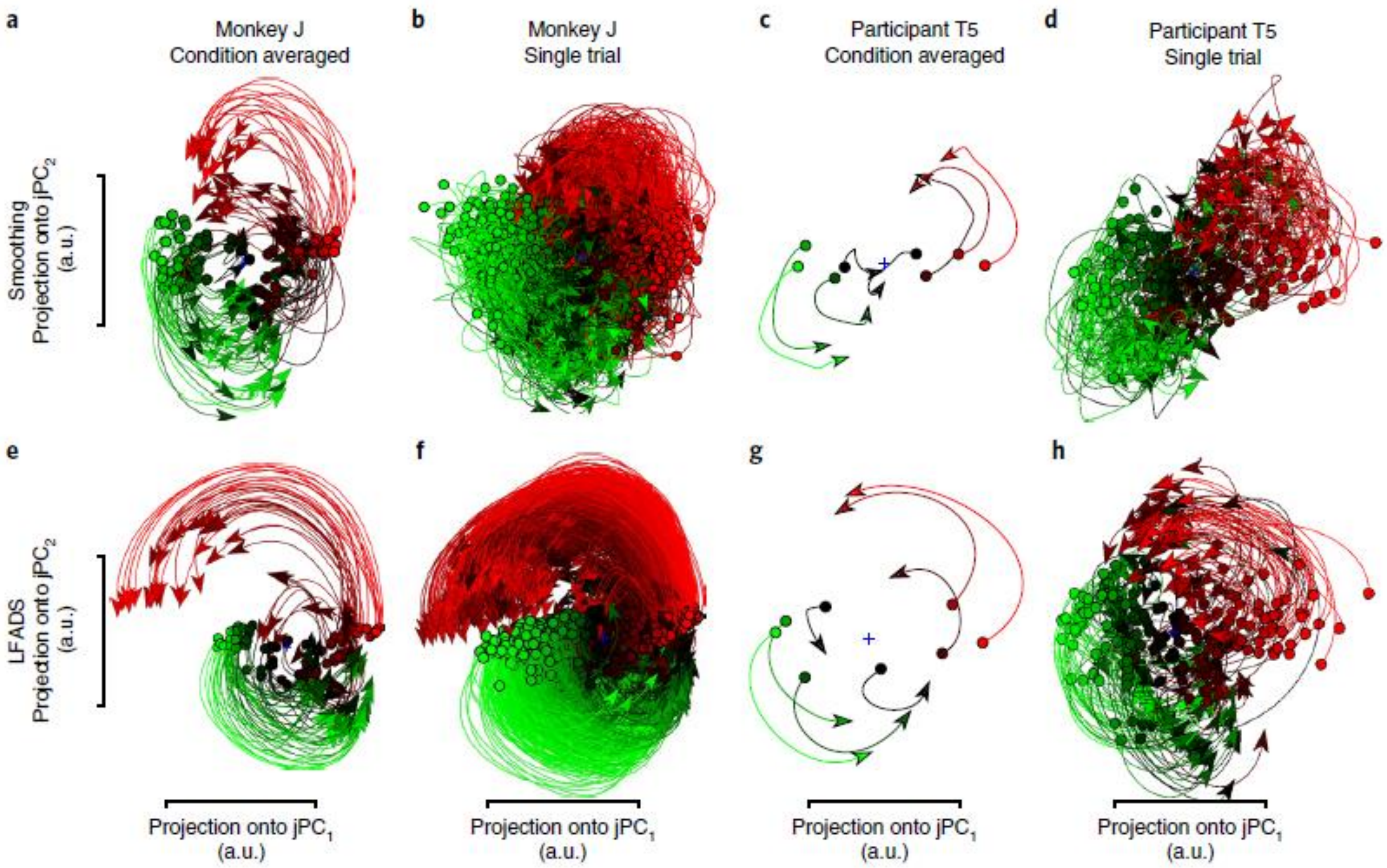
Yeung, CS231n (Stanford) 2017, lecture13

Summary: the LFADS basic architecture

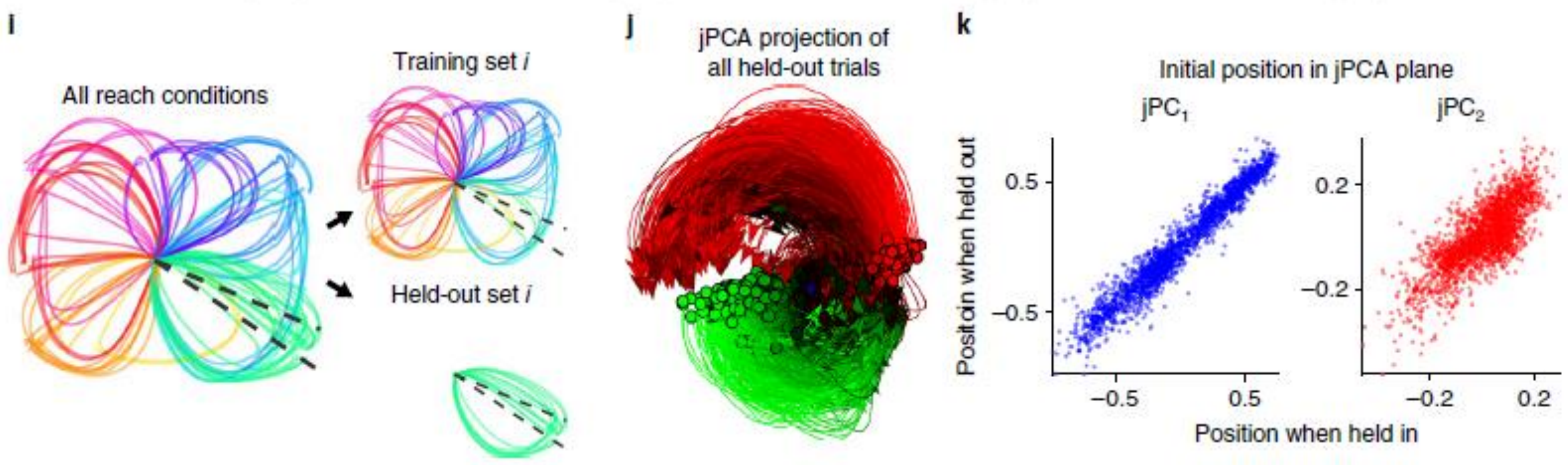


1. LFADS assumes a ***dynamical system*** underlying spike trains
2. LFADS uncovers ***low-dimensional factors*** that underlie firing rates
3. LFADS is trained as a variational autoencoder (***VAE***)

LFADS uncovers single-trial rotation dynamics



LFADS uncovers single-trial rotation dynamics

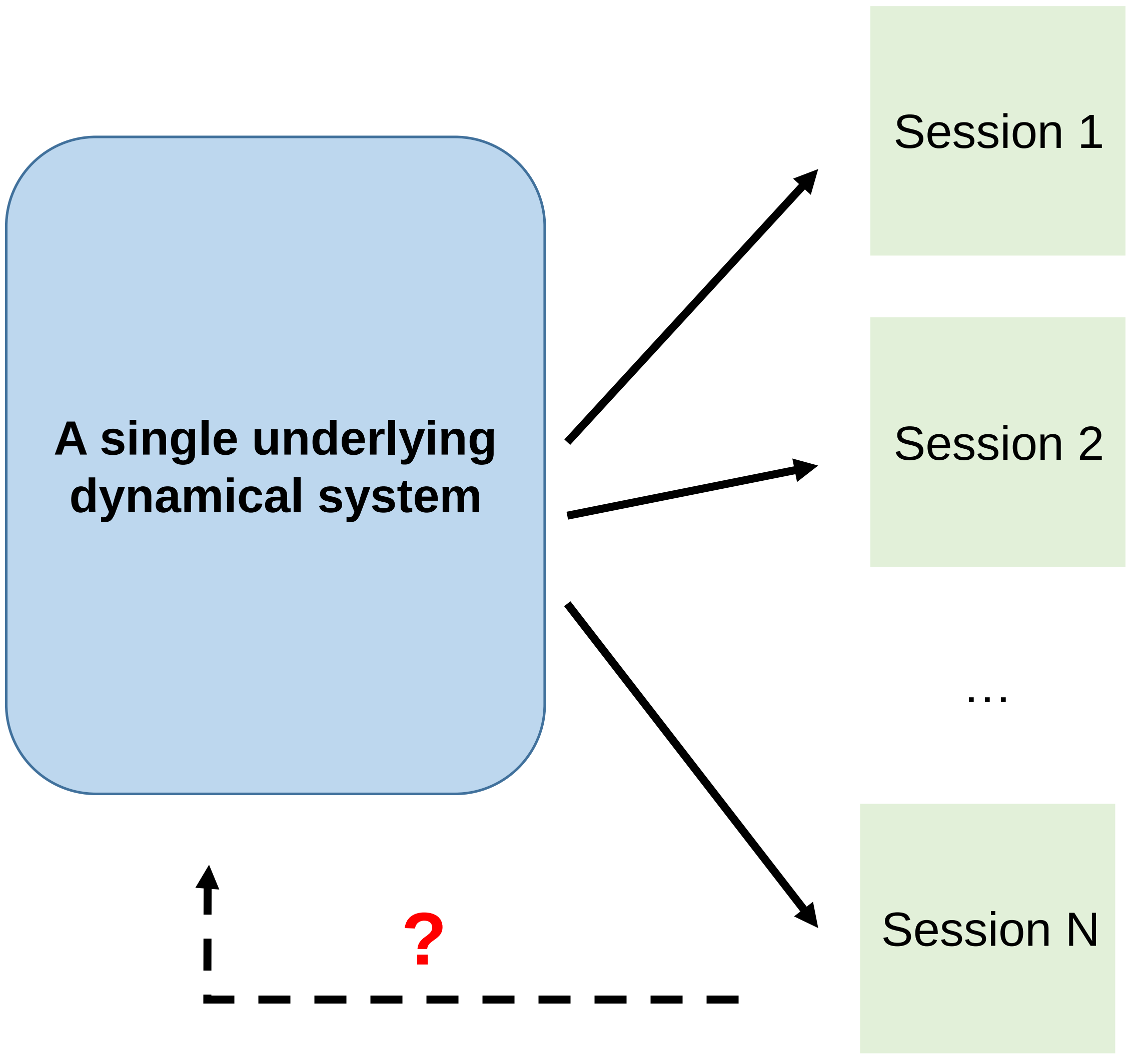


LFADS performs well on held-out trials

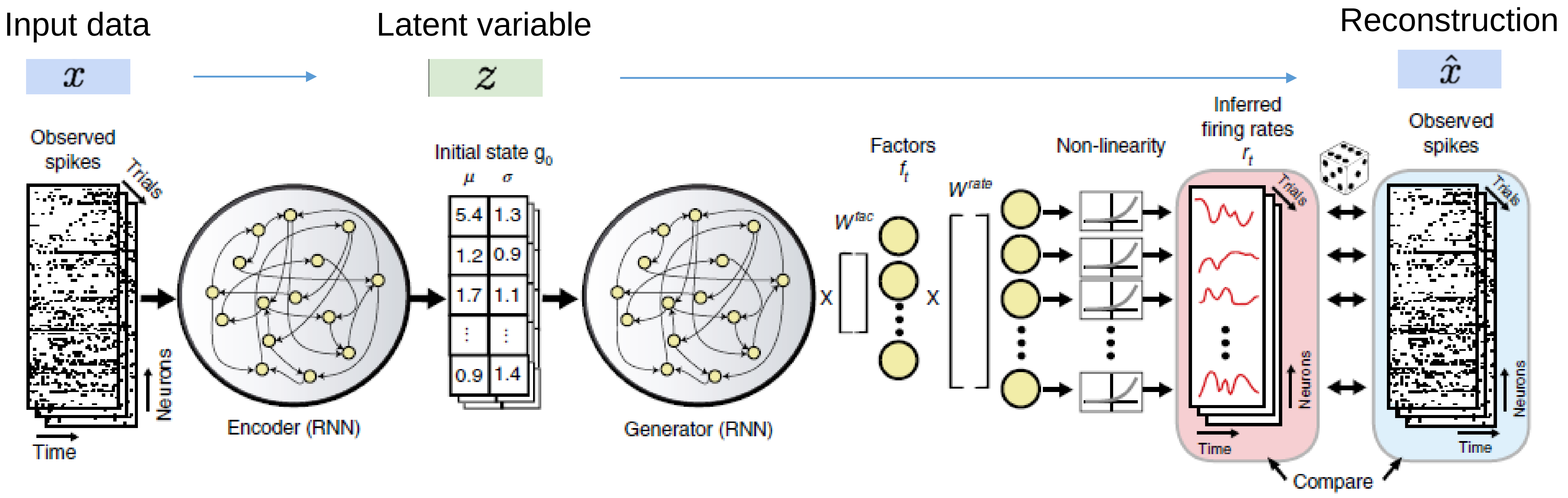
Part II: Variants of LFADS

1. Dynamic neural stitching
2. LFADS with external perturbations

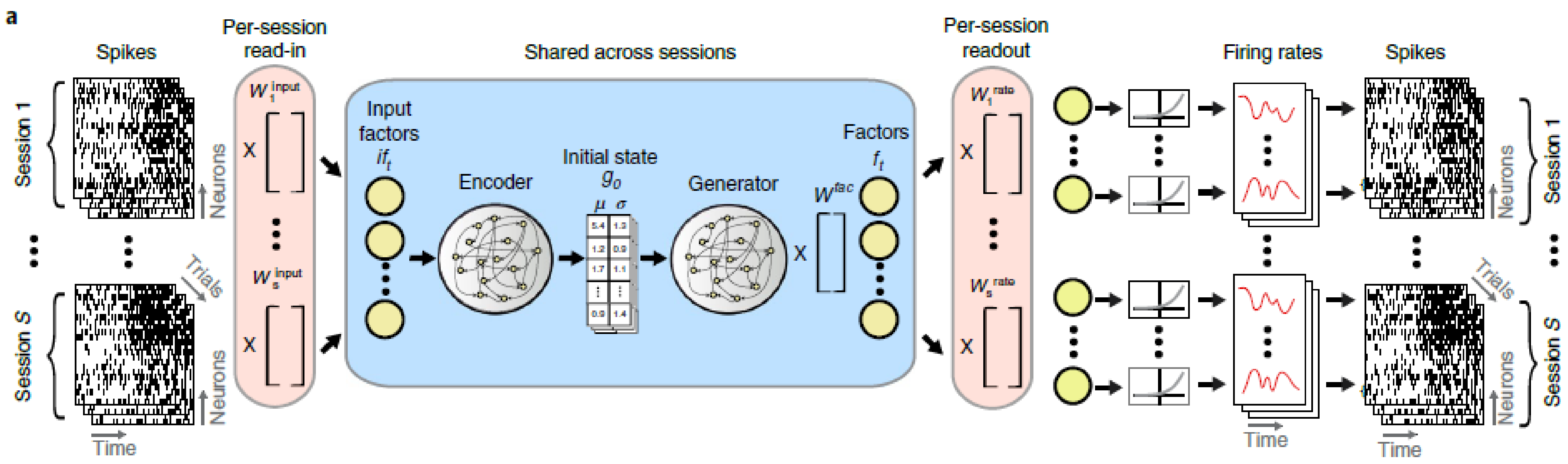
Dynamic neural stitching



LFADS basic architecture

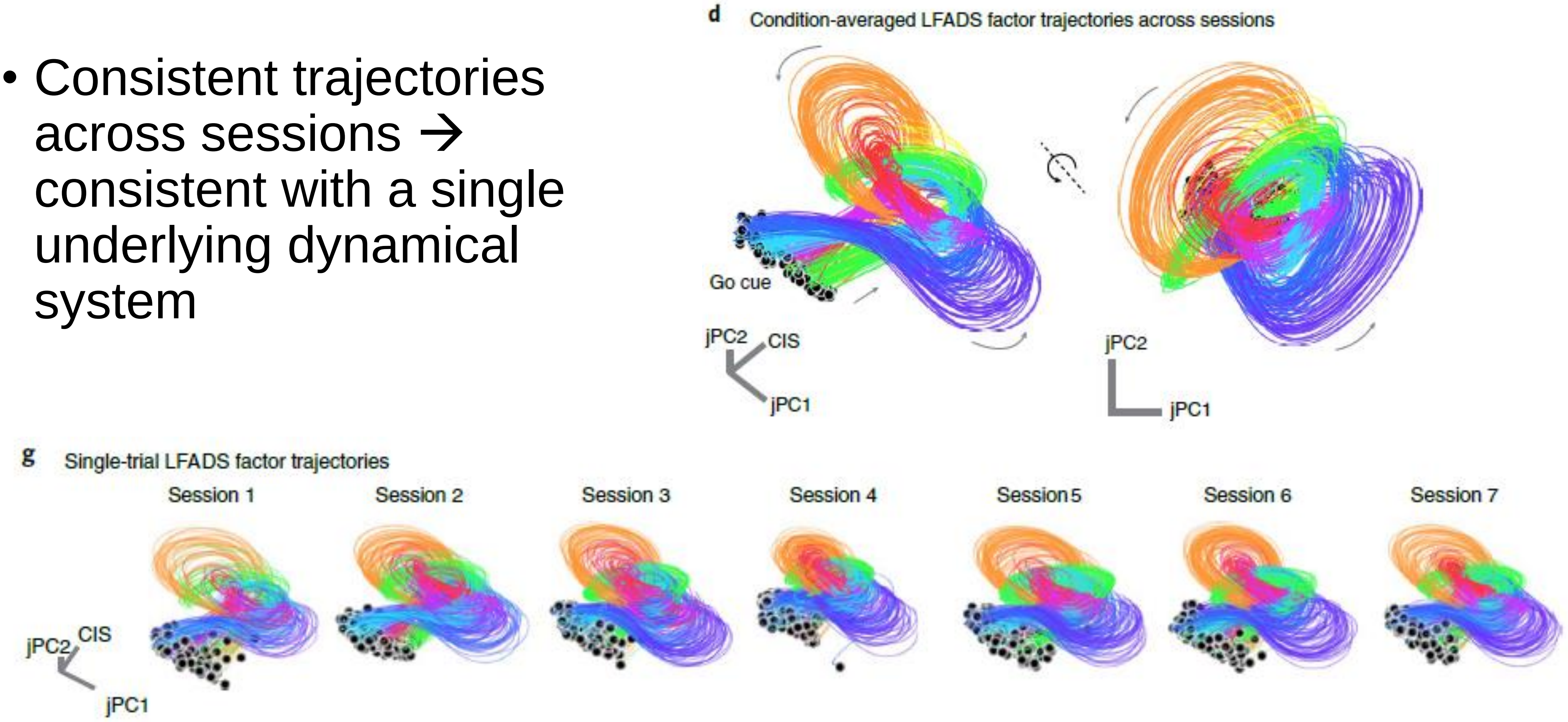


Dynamic neural stitching



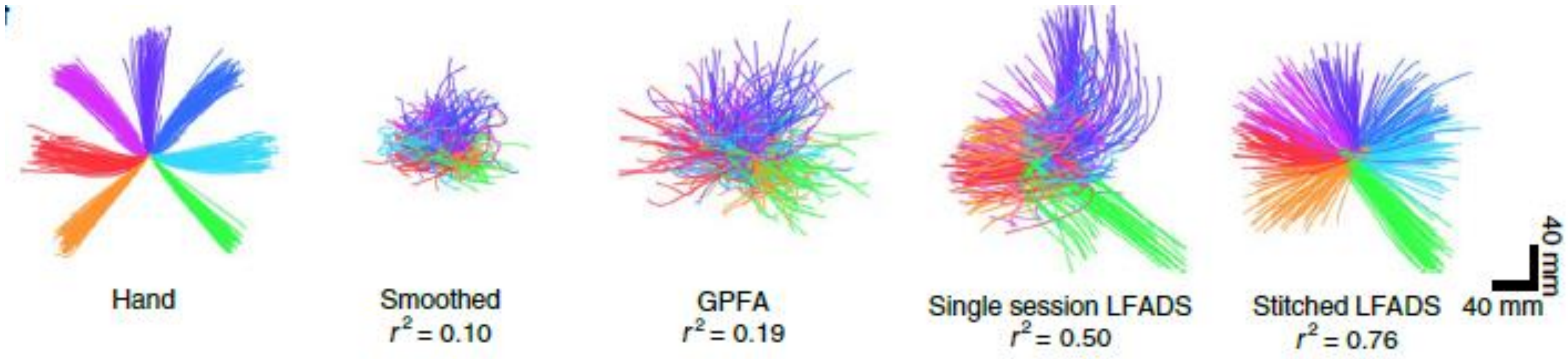
Dynamic neural stitching

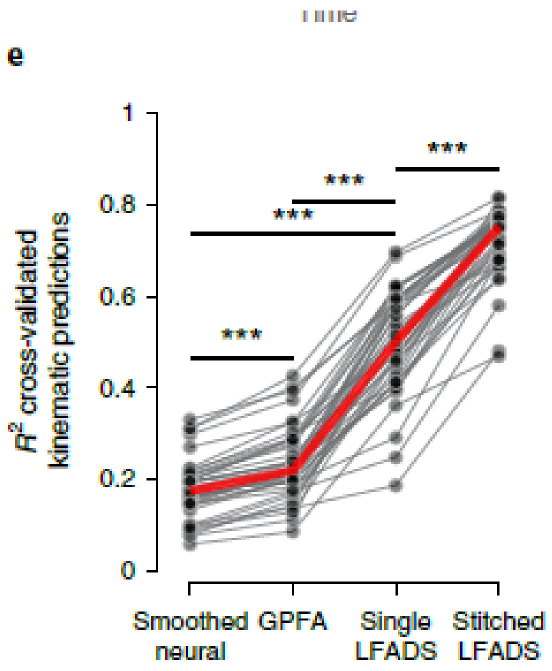
- Consistent trajectories across sessions → consistent with a single underlying dynamical system



Dynamic neural stitching

- Good decoding of kinematic variables using LFADS factors

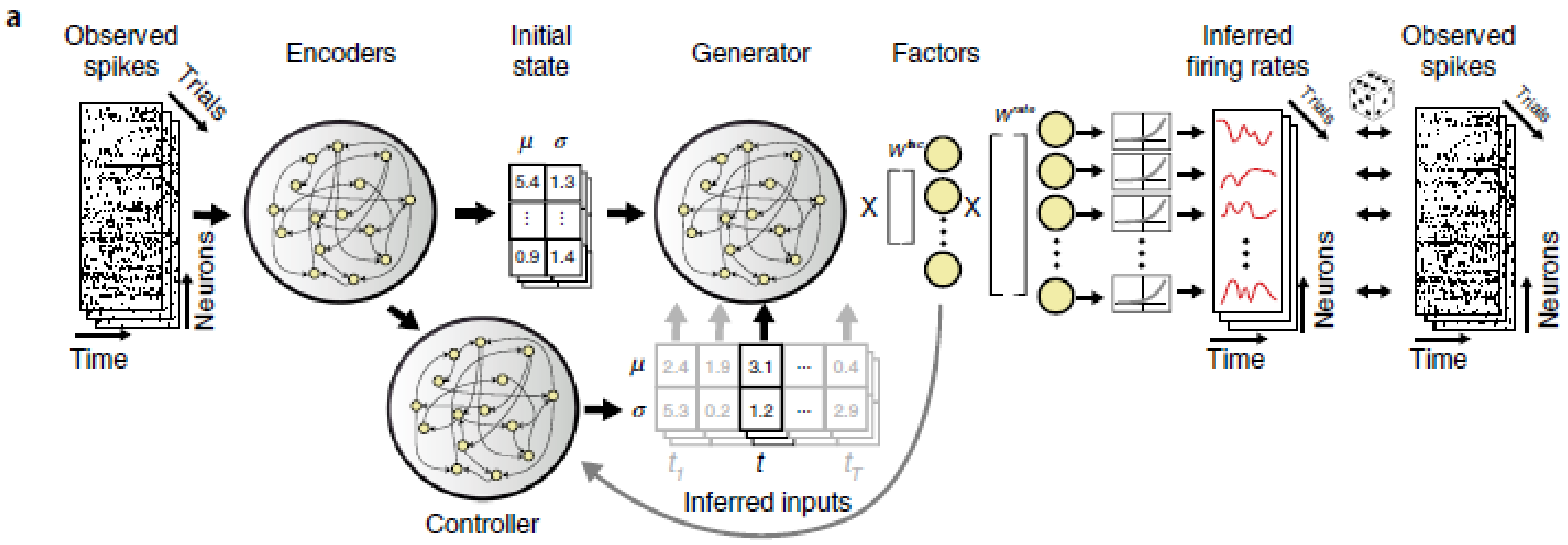




LFADS uncover structure of perturbations

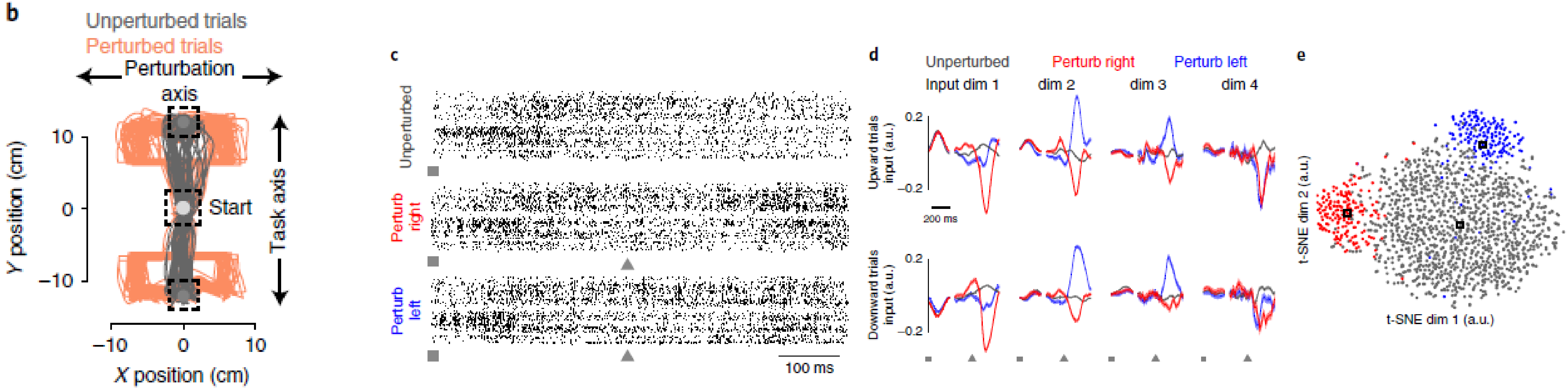
- Behavior of dynamical system often disturbed by **external signals** from other brain areas

→ Can we infer these signals on single-trials?



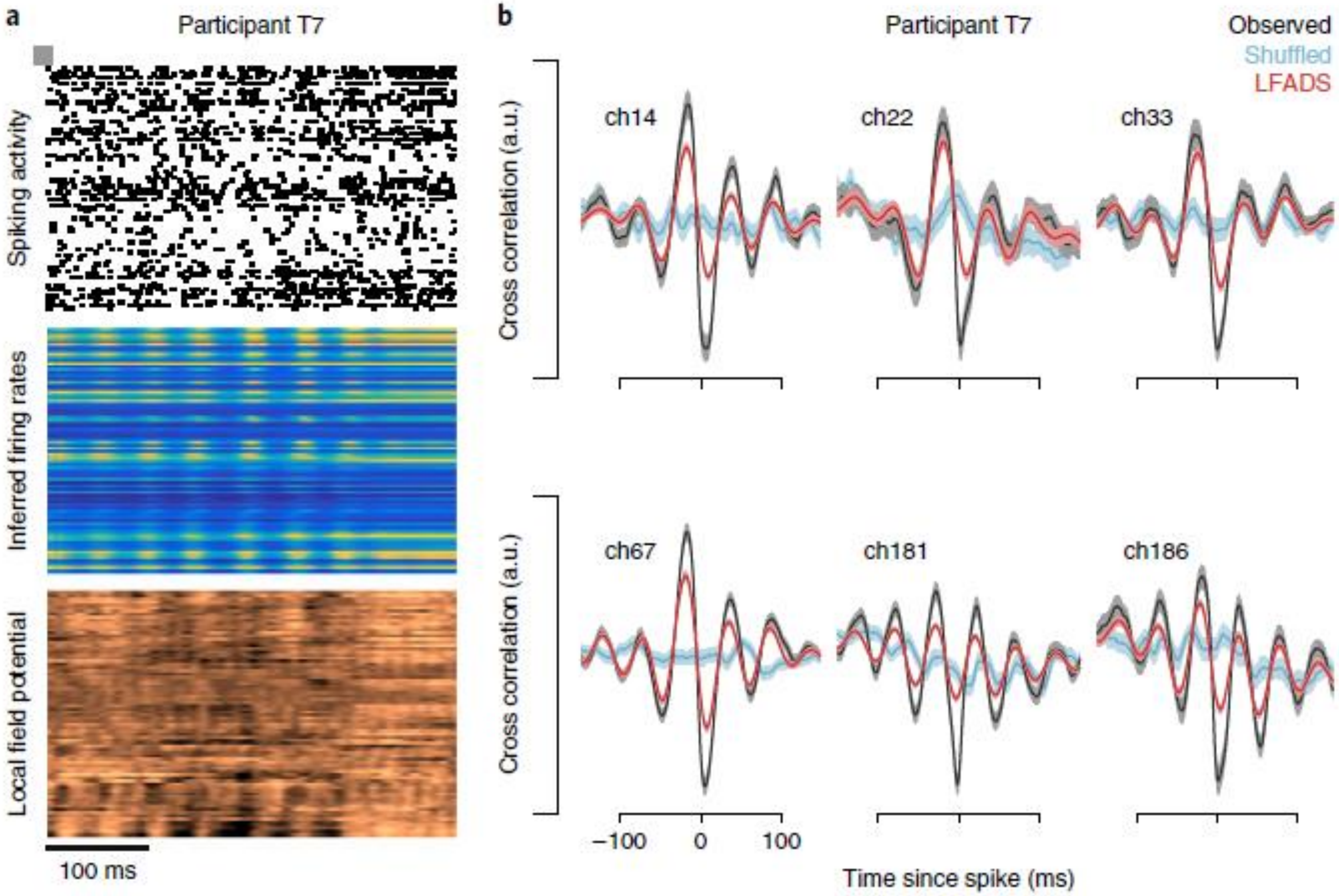
g_0 and ***u(t)*** are the latent variables ***z*** in the VAE

LFADS uncover structure of perturbations



External inputs during a cursor manipulation task when a perturbation is given in some of the trials

LFADS uncover structure of perturbations



Oscillatory inputs before movement initiation, synchronized to LFPs

Discussion

1. In what context would LFADS be appropriate to model neural data? When will it ***not*** be appropriate?