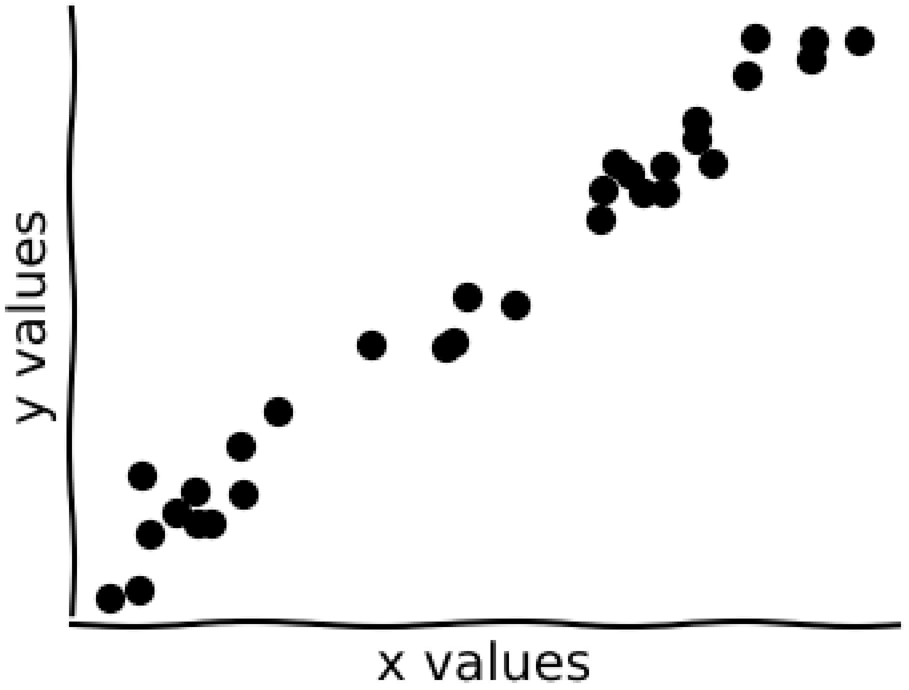
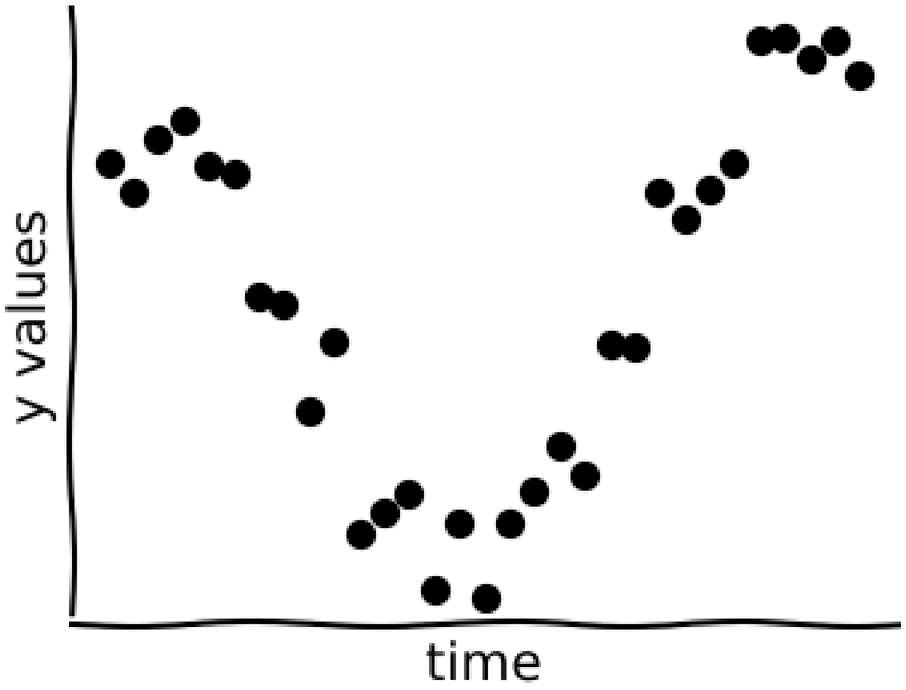
Linear dynamical systems

A tool for neural and behavioural data analysis

Kris Jensen, February 2025

Previously: linear regression

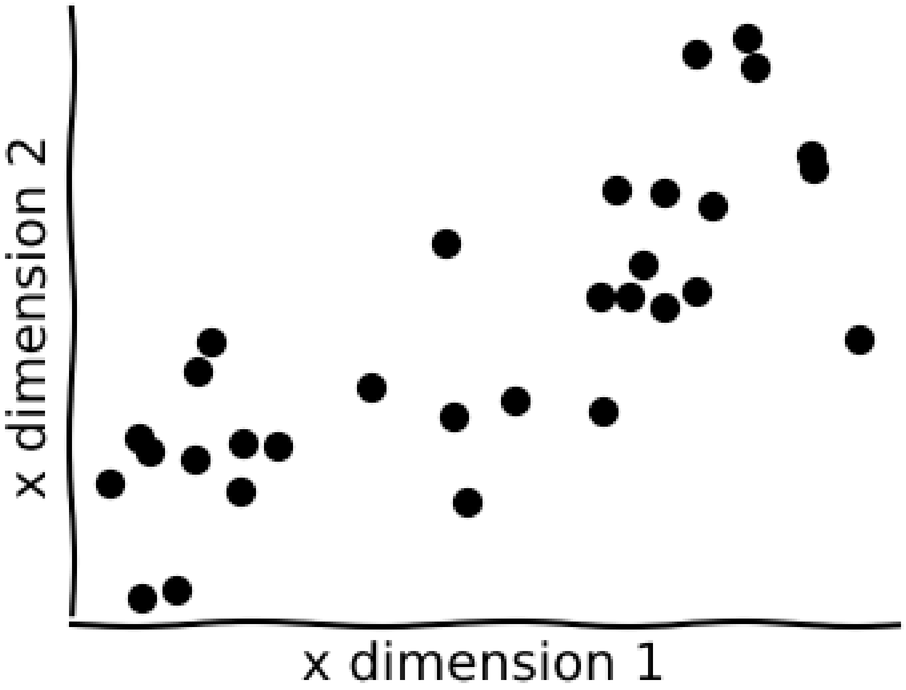


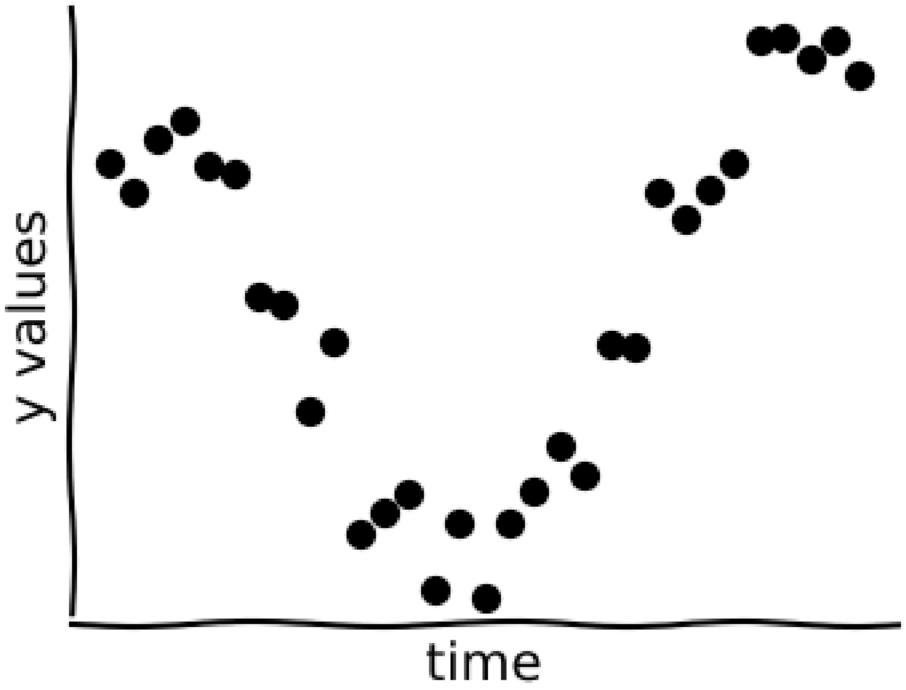


Assumptions:

- We have access to labelled data
- Data points are i.i.d. in time

Previously: PCA

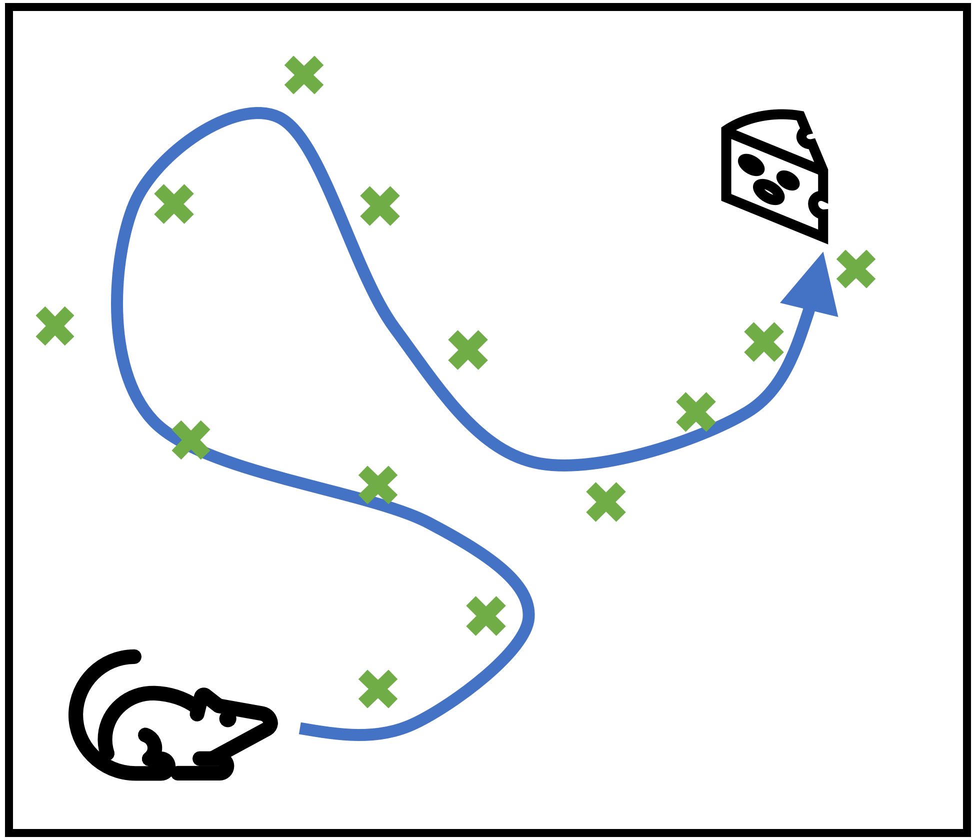




Assumptions:

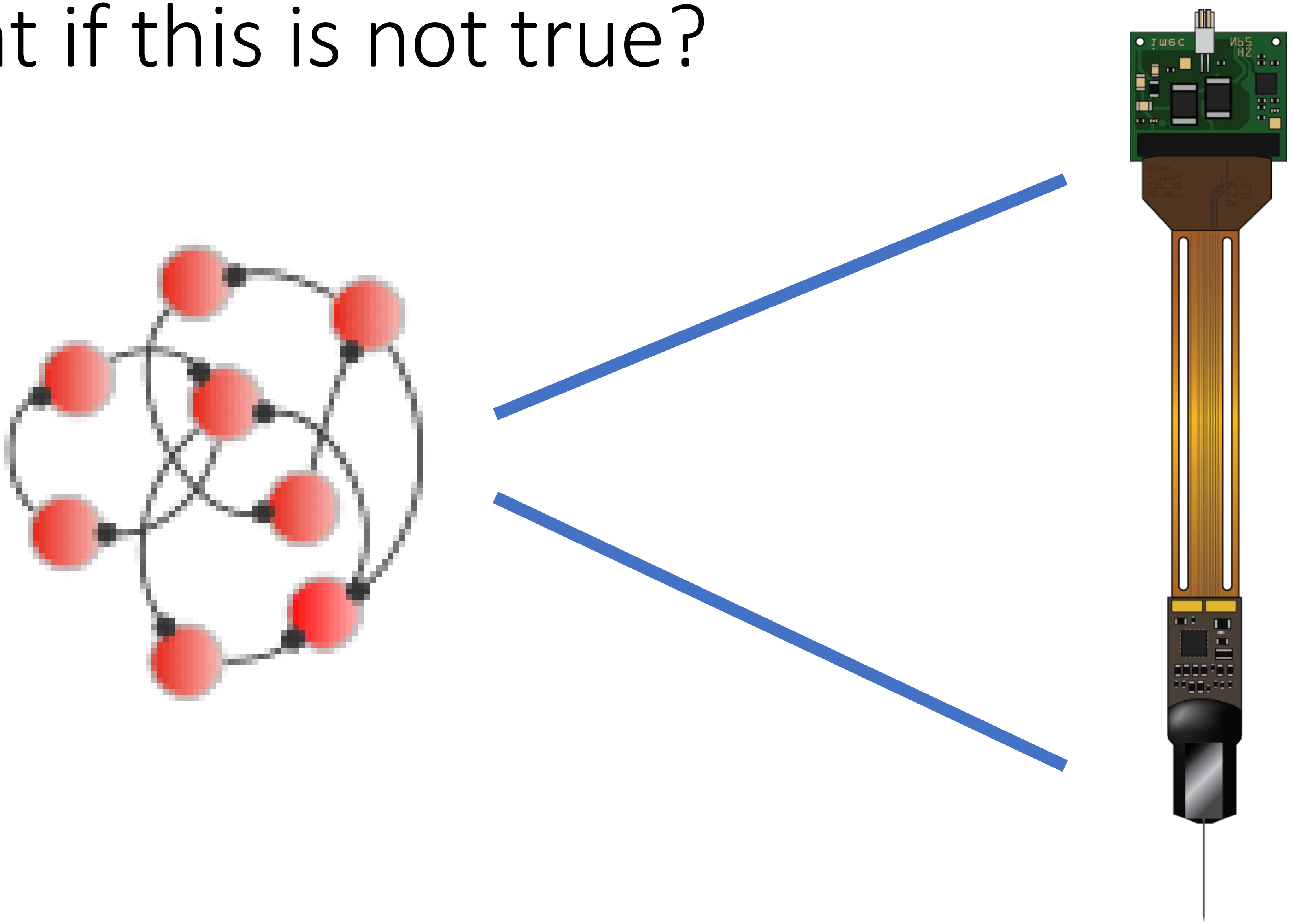
- ~~We have access to labelled data~~
- Data points are i.i.d. in time

What if this is not true?



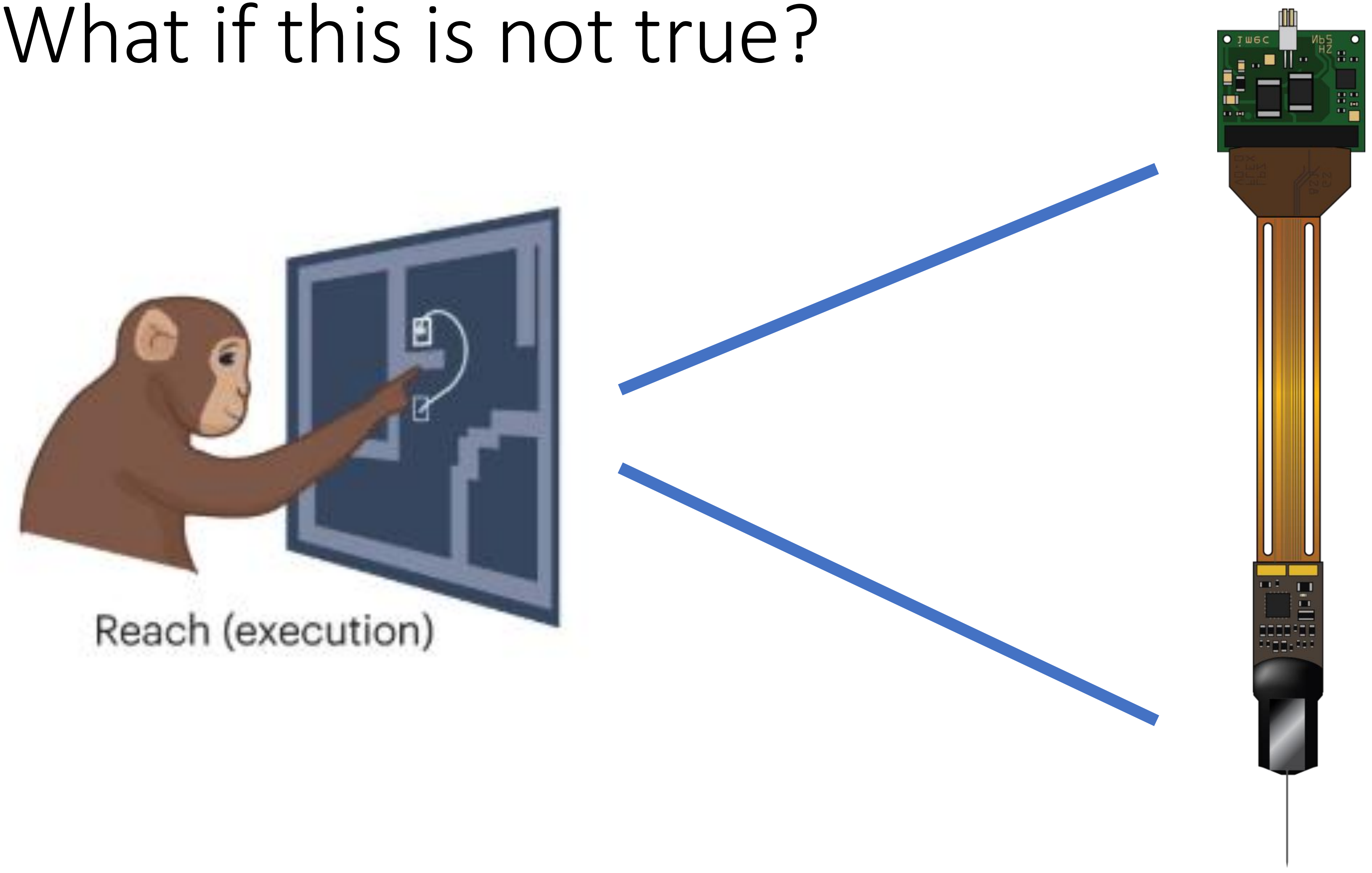
Example 1: behavioural tracking from noisy measurements

What if this is not true?



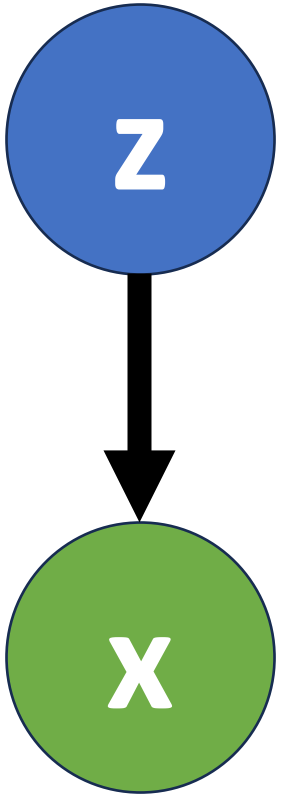
Example 2: neurons as a noisy readout of a low-dimensional system

What if this is not true?



Example 3: a simple ‘brain-computer interface’

PCA as a *probabilistic generative model*



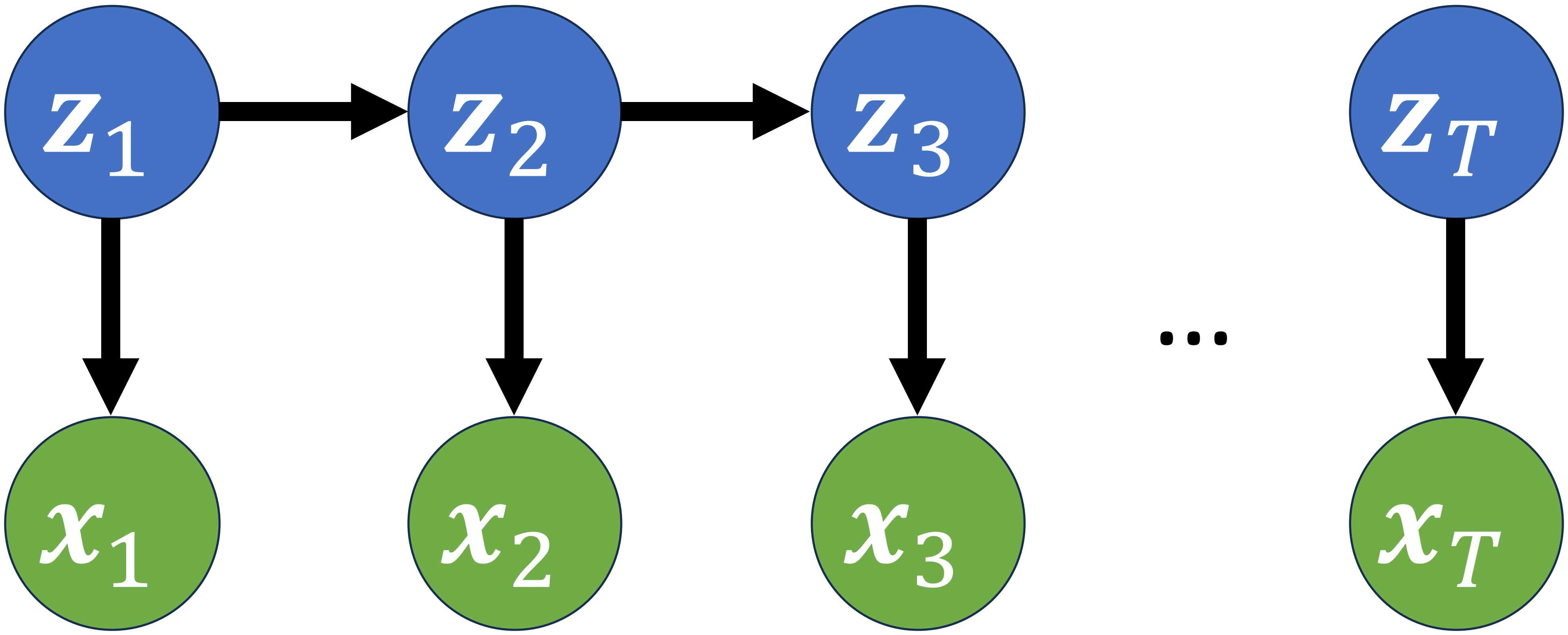
$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

$$p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{C}) = \mathcal{N}(\boldsymbol{C}\boldsymbol{z}, \epsilon\boldsymbol{I})$$

Optimize $\boldsymbol{C} = \underset{\boldsymbol{C}}{\operatorname{argmax}}\, p(\boldsymbol{x}; \boldsymbol{C}) = \int p(\boldsymbol{x}|\boldsymbol{z}; \boldsymbol{C}) p(\boldsymbol{z}) d\boldsymbol{z}$

Compute $p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{C})$

The Kalman filter



$$p(\boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{A}\boldsymbol{z}_{t-1}, \boldsymbol{Q})$$
$$p(\boldsymbol{x}_t|\boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{C}\boldsymbol{z}_t, \boldsymbol{R})$$

The Kalman filter – inference (1d)

$z_t \sim \mathcal{N}(az_{t-1}, q^2);\;\; x_t \sim \mathcal{N}(cz_t, r^2).$ Our goal: estimate $p(z_t|x_{1:t}) = \mathcal{N}(\mu_t, \sigma_t^2)$

- $p(z_t|x_{1:t}) \propto p(x_t|z_t, x_{1:t-1})p(z_t|x_{1:t-1}) = p(x_t|z_t\,)p(z_t|x_{1:t-1})$ (prediction & update)
- $p(z_t|x_{1:t-1}) = \int p(z_t, z_{t-1}|x_{1:t-1})dz_{t-1} = \int p(z_t|z_{t-1}\,)p(z_{t-1}|x_{1:t-1})dz_{t-1}$
- $p(z_t|z_{t-1}) = \mathcal{N}(az_{t-1}, q^2);\quad p(z_{t-1}|x_{1:t-1}) = \mathcal{N}(\mu_{t-1}, \sigma_{t-1}^2)$

- $p(z_t|x_{1:t-1}) = \mathcal{N}\left(\mu_{t|t-1}, \sigma_{t|t-1}^2\right)$
  - $\mu_{t|t-1} = \langle a(\mu_{t-1} + \sigma_{t-1}\epsilon_1) + q\epsilon_2\rangle_{\epsilon_1,\epsilon_2} = a\mu_{t-1}\,,$ where $\epsilon \sim \mathcal{N}(0,1)$
  - $\sigma_{t|t-1}^2 = \langle (a(\mu_{t-1} + \sigma_{t-1}\epsilon_1 - \mu_{t-1}) + q\epsilon_2)^2\rangle_{\epsilon_1,\epsilon_2} = a^2\sigma_{t-1}^2 + q^2$

- $p(x_t|z_t) = \mathcal{N}(cz_t, r^2)$
  - $\exp[-r^{-2}(x_t - cz_t)^2] = \exp[-(r/c)^{-2}(z_t - x_t/c)^2]$

- $p(z_t|x_{1:t}) = \mathcal{N}(\mu_t, \sigma_t^2)$
  - $\sigma_t^2 = \left((c/r)^2 + \sigma_{t|t-1}^{-2}\right)^{-2}$
  - $\mu_t = \sigma_t^2\left(\left(\frac{c}{r}\right)^2\frac{x_t}{c} + \sigma_{t|t-1}^{-2}\mu_{t|t-1}\right)$

The Kalman filter – inference (general)

The equations for higher-dimensional systems can be derived similarly.
$\boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{z}_{t-1}, \boldsymbol{Q})$; $\quad\quad\quad$ $\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{C}\boldsymbol{z}_t, \boldsymbol{R})$

- $p(\boldsymbol{z}_t | \boldsymbol{x}_{1:t-1}) = \mathcal{N}\left(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}\right)$
  - $\boldsymbol{\mu}_{t|t-1} = \boldsymbol{A}\boldsymbol{\mu}_{t-1}$
  - $\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{A}\boldsymbol{\Sigma}_{t-1}\boldsymbol{A}^T + \boldsymbol{Q}$

- $p(\boldsymbol{x}_t | \boldsymbol{z}_t) = \mathcal{N}(\boldsymbol{C}\boldsymbol{z}_t, \boldsymbol{R})$

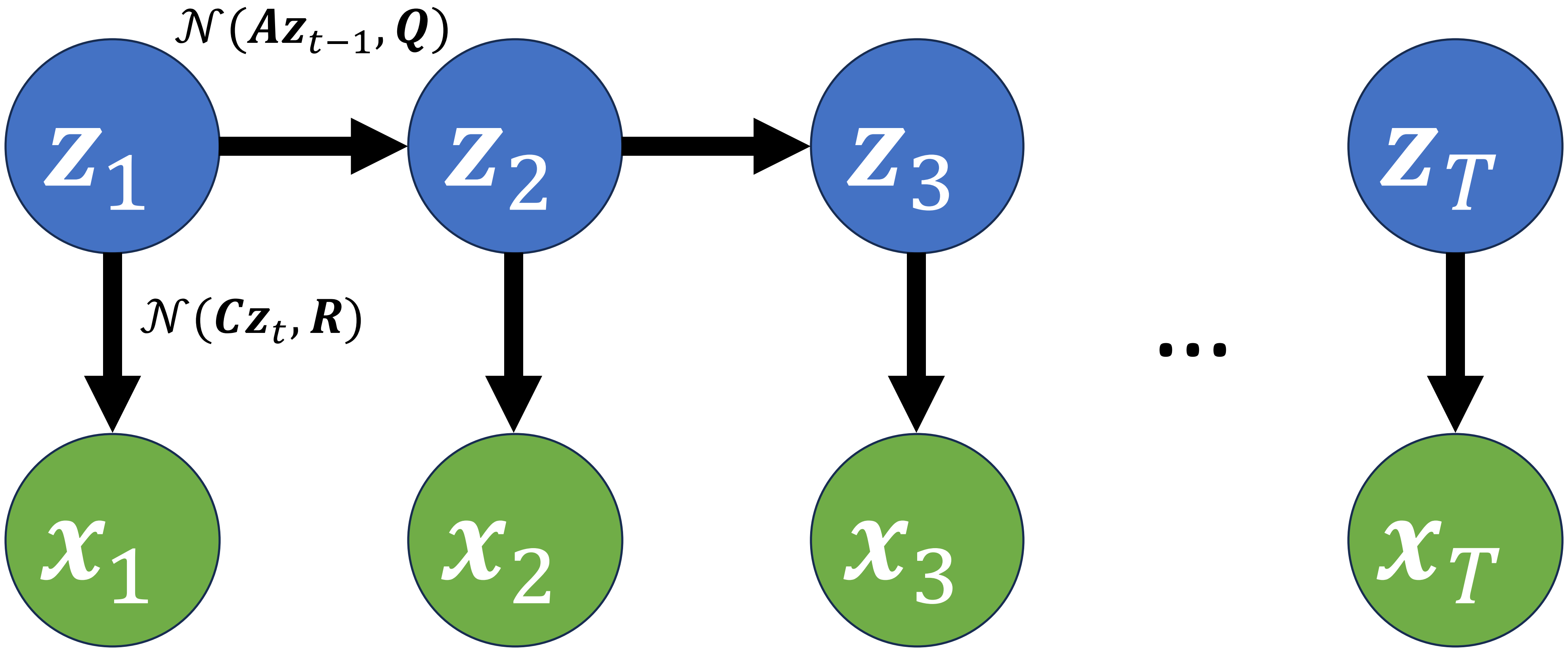
- $p(\boldsymbol{z}_t | \boldsymbol{x}_{1:t}) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$
  - $\boldsymbol{\Sigma}_t = \left(\boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{C} + \boldsymbol{\Sigma}_{t|t-1}^{-1}\right)^{-1}$
  - $\boldsymbol{\mu}_t = \boldsymbol{\Sigma}_t\left(\boldsymbol{C}^T\boldsymbol{R}^{-1}\boldsymbol{x}_t + \boldsymbol{\Sigma}_{t|t-1}^{-1}\boldsymbol{\mu}_{t|t-1}\right)$

Kalman *smoothing*

$$
\begin{aligned}
p(z_t|x_{1:T}) &= \int p(z_t, z_{t+1}|x_{1:T})dz_{t+1} \\
&= \int p(z_t|z_{t+1}, x_{1:T})p(z_{t+1}|x_{1:T})dz_{t+1} \\
&= \int p(z_t|z_{t+1}, x_{1:t})p(z_{t+1}|x_{1:T})dz_{t+1} \\
&\propto p(z_t|x_{1:t})\int p(z_{t+1}|z_t\ )p(z_{t+1}|x_{1:T})dz_{t+1}
\end{aligned}
$$

- This recursion expresses $p(z_t|x_{1:T}) = \mathcal{N}(\nu_t, \eta_t^2)$ as a function of $(\nu_{t+1}, \eta_{t+1}^2)$, our 'filtering distribution' $z_t|x_{1:t} \sim \mathcal{N}(\mu_t, \sigma_t^2)$ and dynamics $z_{t+1}|z_t \sim \mathcal{N}(az_t, q^2)$.
- Upside: we get a better estimate of our latents by using both past and future data.
- Downside: we have to wait until the experiment finishes before we can perform the computation.
- This is useful for post-hoc data processing, but not for online decoding/tracking.
- An intermediate approach computes $p(z_t|x_{1:t+n})$ for some choice of n.
- For more details, see *Roweis & Ghahramani (1999).*

Where do the parameters come from???

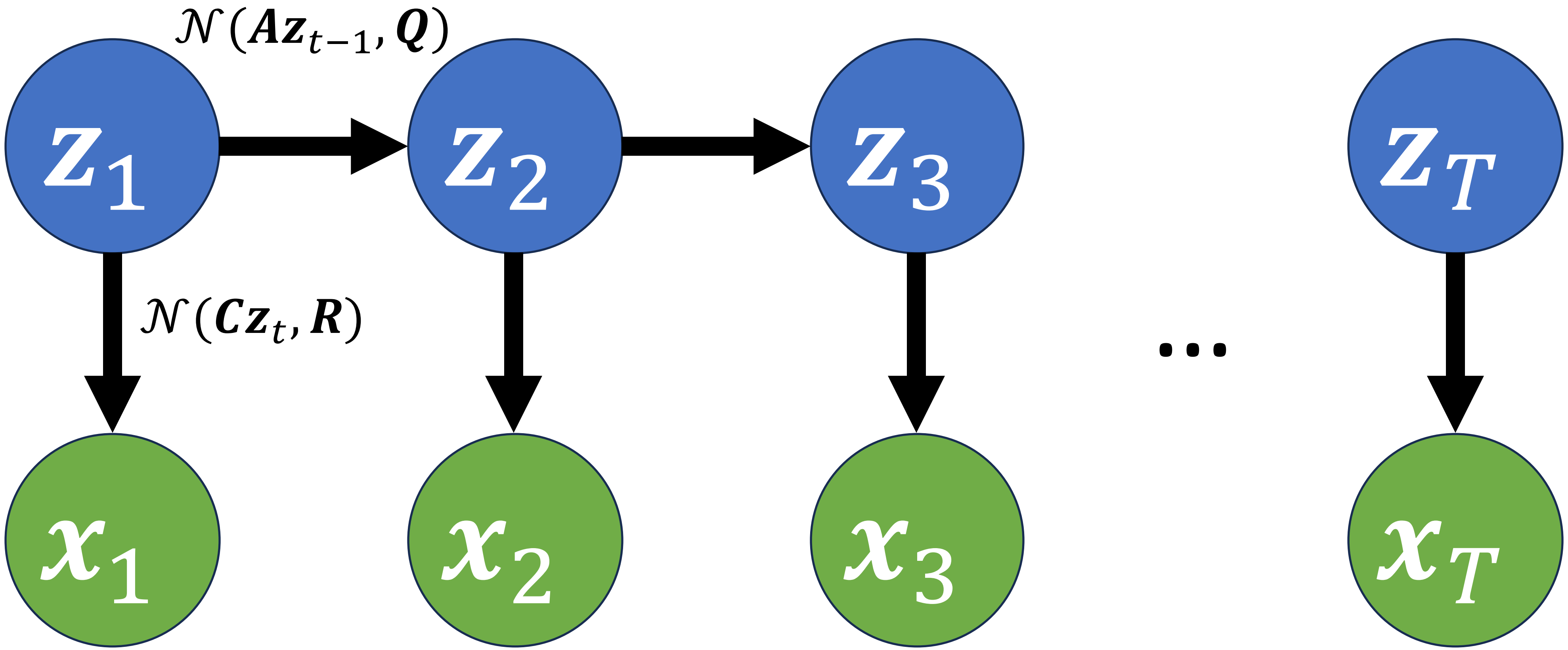


We might know the ‘true’ dynamics

We can *maximize the likelihood* of the parameters on some training data:

$L(\theta) = \sum_t \log p(\boldsymbol{z}_t | \boldsymbol{x}_{1:t}, \theta)$

What if we don't have any 'training data'?



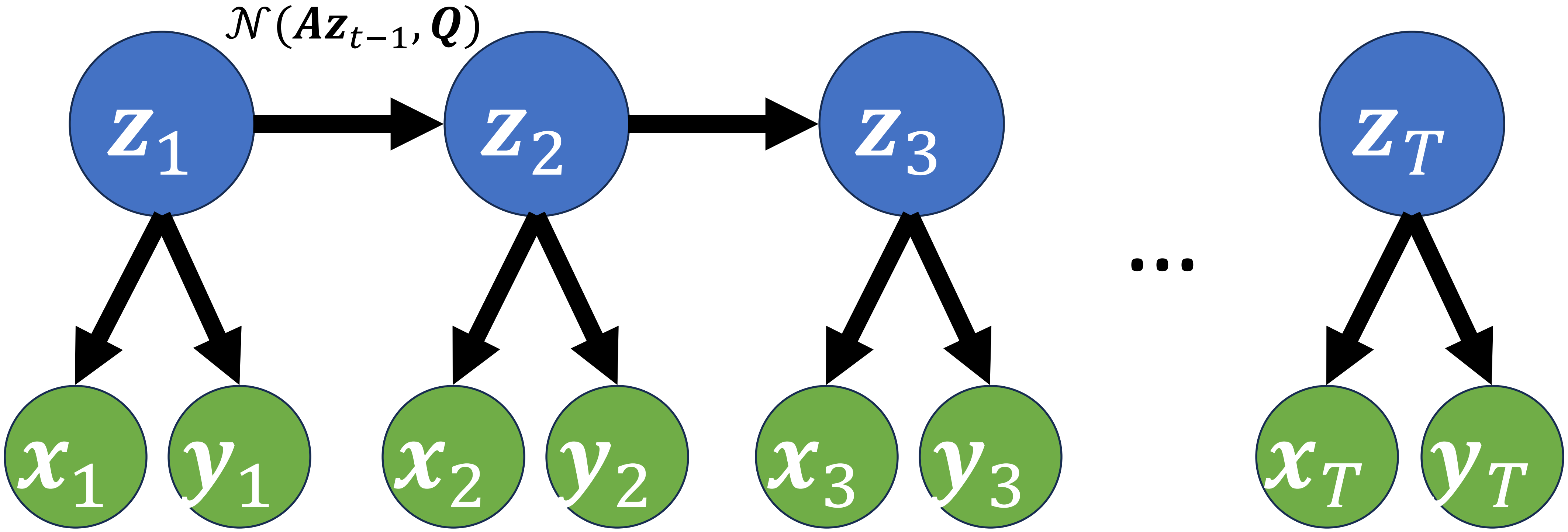
We can maximize the likelihood of the parameters on the *observations*:

$L(\theta) = \log p(\boldsymbol{X}|\theta) = \log \int_{\boldsymbol{z}} p\,(\boldsymbol{X}|\boldsymbol{Z}, \theta)p(\boldsymbol{Z}|\theta)d\boldsymbol{Z}$

We use *expectation maximization*, alternating between inferring $\boldsymbol{Z}$ and optimizing θ:

$$\theta_{i+1} \leftarrow \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{\boldsymbol{Z}\sim p(\boldsymbol{Z}|\boldsymbol{X};\, \theta_i)}[\log p(\boldsymbol{Z}, \boldsymbol{X}; \theta)]$$

What if we care about the *output* of an LDS?



$\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{C}\boldsymbol{z}_t, \boldsymbol{R})$

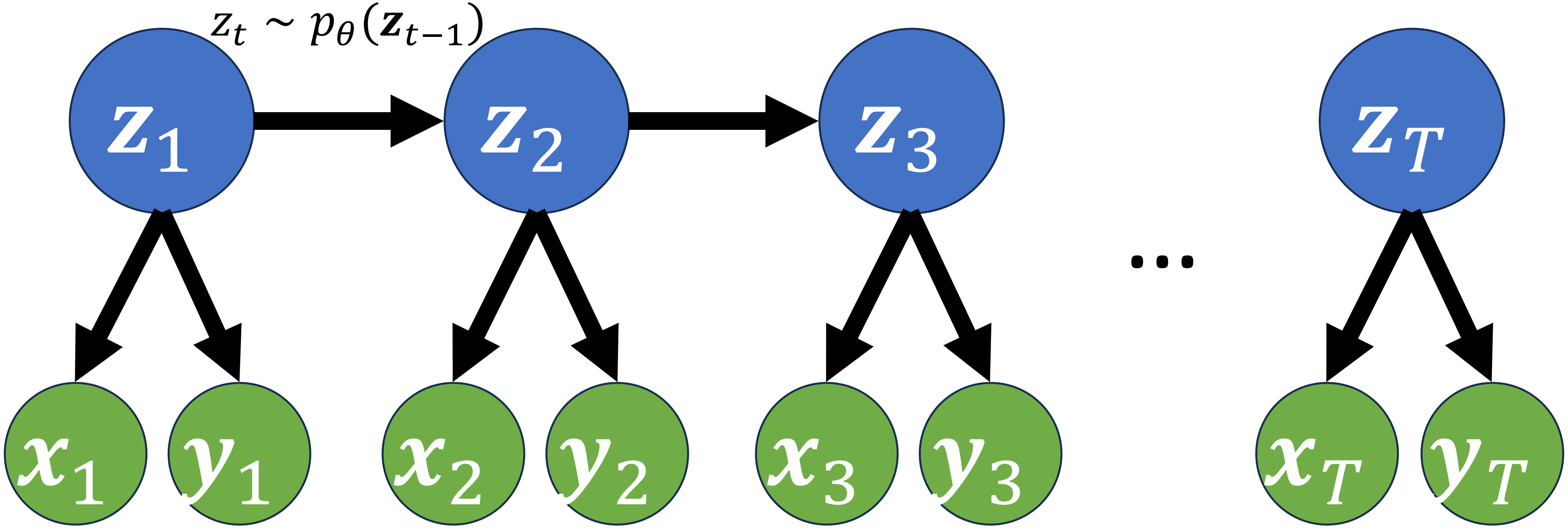
$y_t \sim \mathcal{N}(\boldsymbol{D}\boldsymbol{z}_t, \boldsymbol{S})$

We can compute $p(\boldsymbol{y}_t|\boldsymbol{x}_{1:t}) = \int_z p(\boldsymbol{y}_t|\boldsymbol{z}_t)p(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})dz$

$p(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})$ is given by our Kalman filter

$\{\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{S}\} = \underset{\theta}{\operatorname{argmax}} \log p(\boldsymbol{Y}, \boldsymbol{X})$

Why linear and why Gaussian?



$\boldsymbol{x}_t \sim p_\theta(\boldsymbol{z}_t)$ $\quad$ $\boldsymbol{y}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_\theta(\boldsymbol{z}_t), \boldsymbol{\Sigma}_\theta(\mathbf{z}_\mathrm{t})\right)$

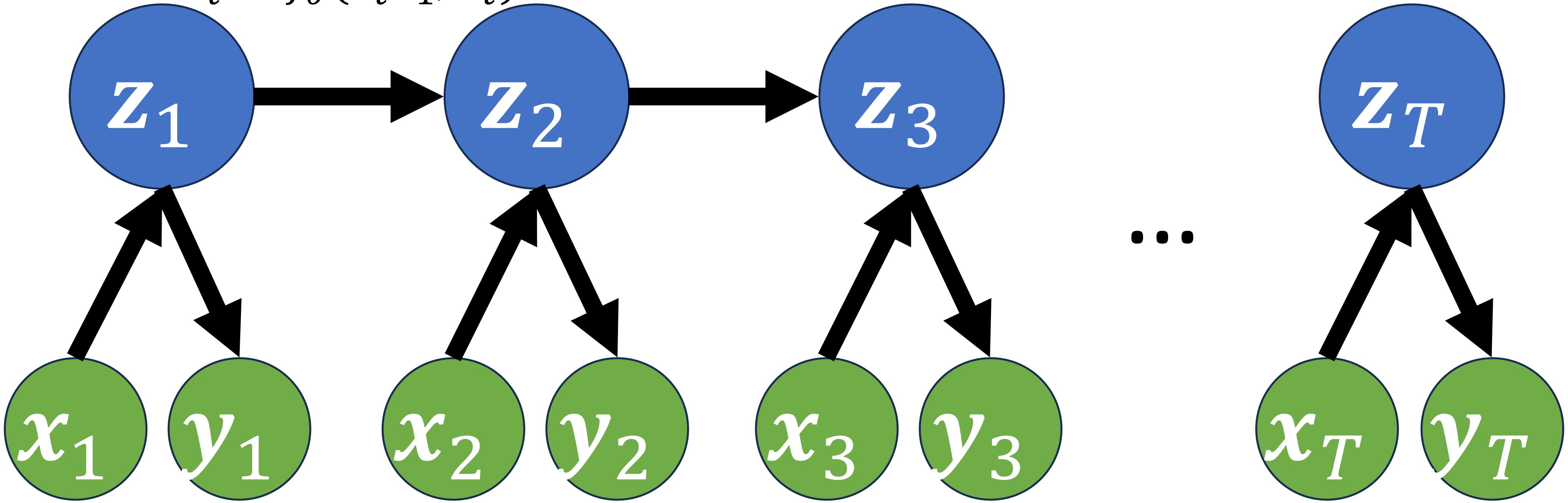
~~We can compute~~ $\cancel{p(\boldsymbol{y}_t|\boldsymbol{x}_{1:t}) = \int_z p(\boldsymbol{y}_t|\boldsymbol{z}_t)p(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})dz}$

$\cancel{p(\boldsymbol{z}_t|\boldsymbol{x}_{1:t})}$ ~~is given by our Kalman filter~~

$$\theta = \operatorname*{argmax}_\theta \log p(\boldsymbol{Y}, \boldsymbol{X})$$

Why linear and why Gaussian?

$\boldsymbol{z}_t = f_\theta(\boldsymbol{z}_{t-1}, \boldsymbol{x}_t)$



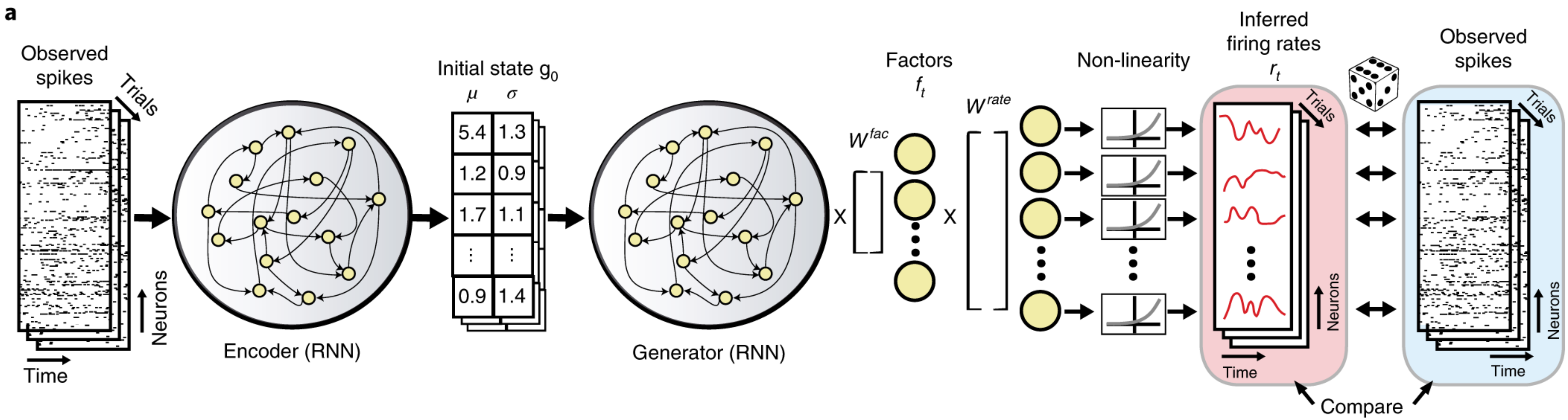
$\boldsymbol{y}_t \sim \mathcal{N}\left(\boldsymbol{\mu}_\theta(\boldsymbol{z}_t), \boldsymbol{\Sigma}_\theta(\mathbf{z}_\mathrm{t})\right)$

We can compute $p(\boldsymbol{y}_t | \boldsymbol{x}_{1:t}) = p(\boldsymbol{y}_t | z(\boldsymbol{x}_{1:t}))$

$\theta = \underset{\theta}{\operatorname{argmax}} \log p(\boldsymbol{Y}|\boldsymbol{X})$

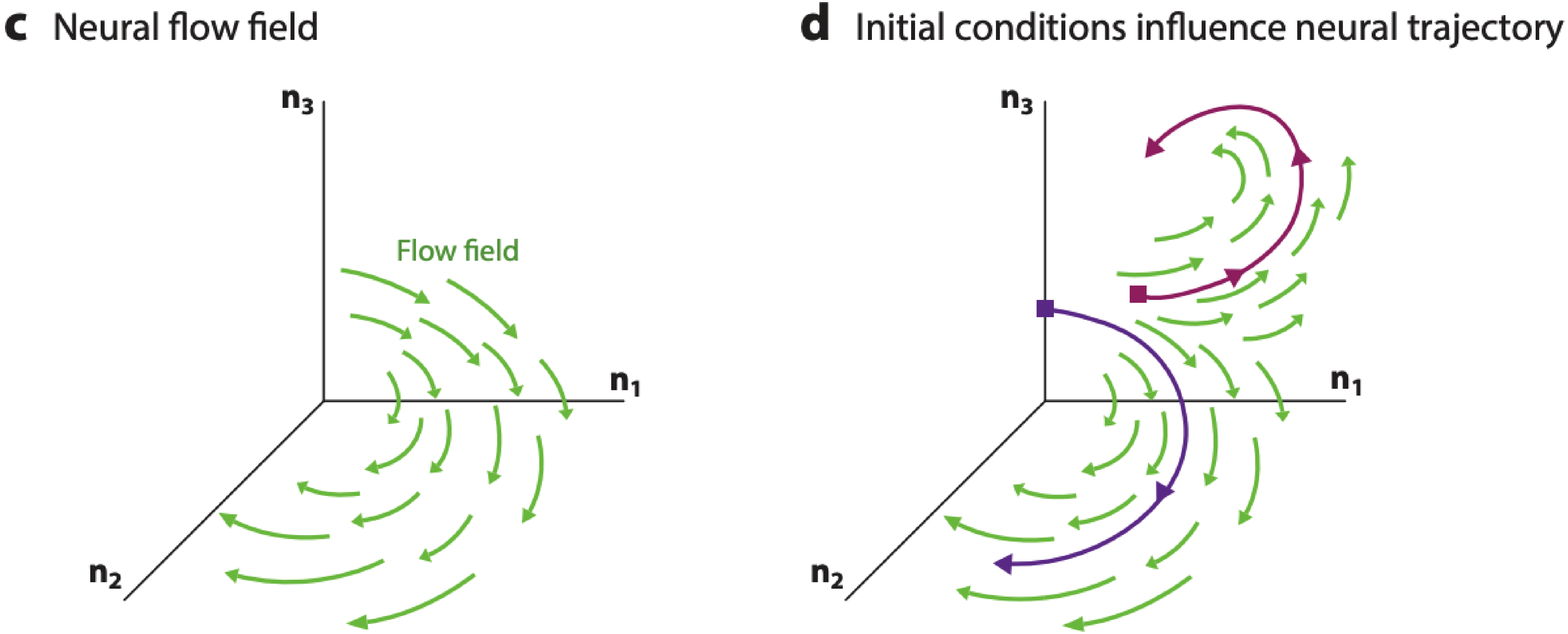
No more closed-form solutions!

Nonlinear dimensionality reduction



Pandarinath et al. (2018)

The brain as a ‘dynamical system’



Vyas et al. (2020)