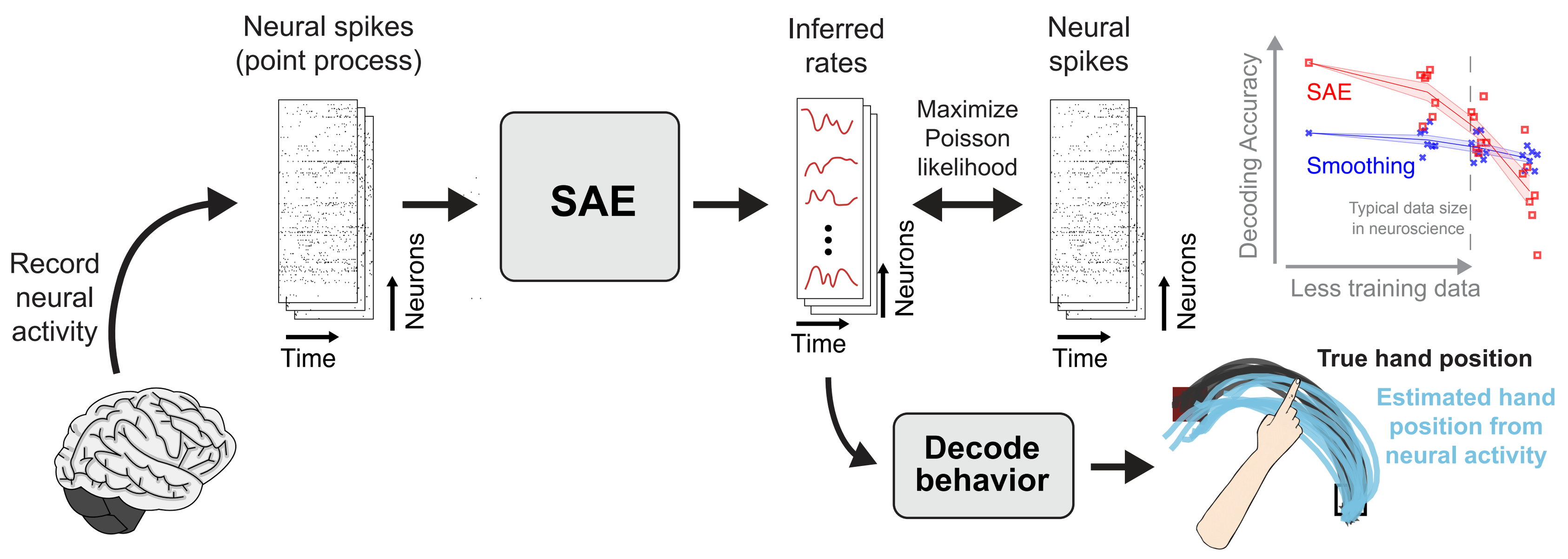
Enabling hyperparameter optimization in sequential autoencoders for spiking neural data

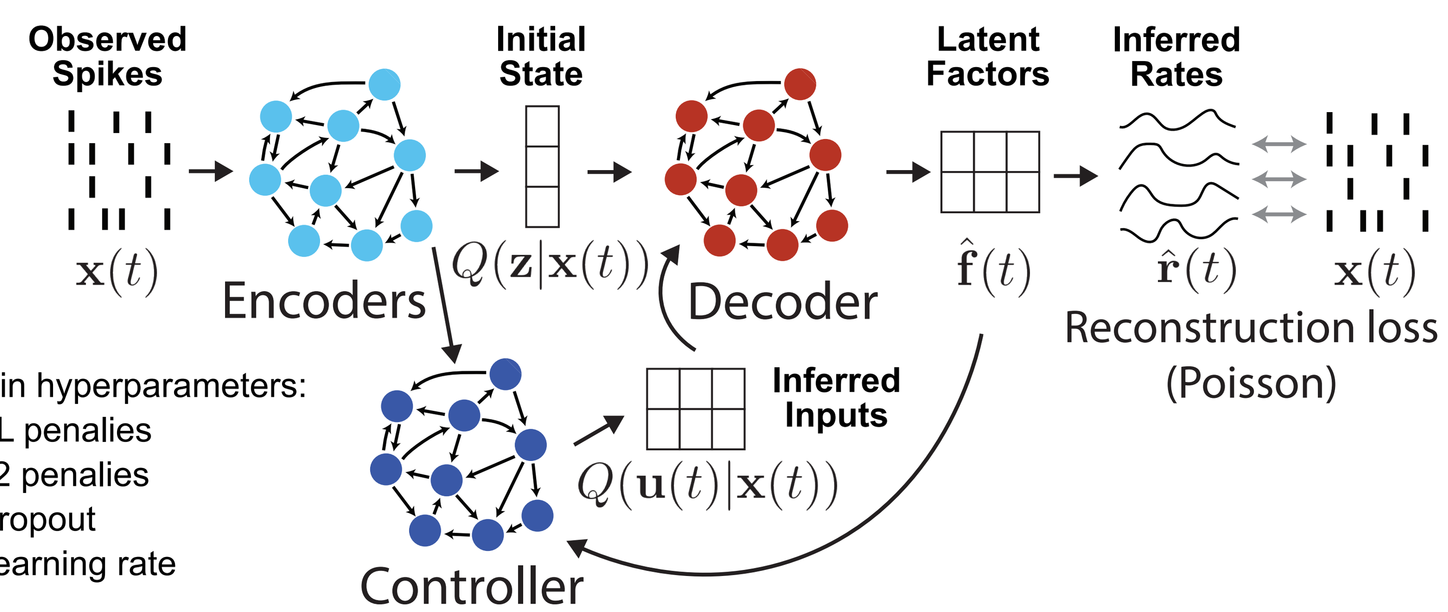
Mohammad Reza Keshtkaran, Chethan Pandarinath

Georgia Tech and Emory University Email: mkeshtk@emory.edu

Sequential autoencoder (SAE) for modeling neural dynamics



SAE Model: Latent factor analysis through dynamical systems (LFADS)[1]



Main hyperparameters:
- KL penalies
- L2 penalies
- Dropout
- Learning rate

Reconstruction log-likelihood: $\mathcal{L}^x = \left\langle \sum_{t=1}^{T} \log\left(\text{Poisson}(\mathbf{x}_t|\hat{\mathbf{r}}_t)\right) \right\rangle_{\mathbf{z},\mathbf{u}}$

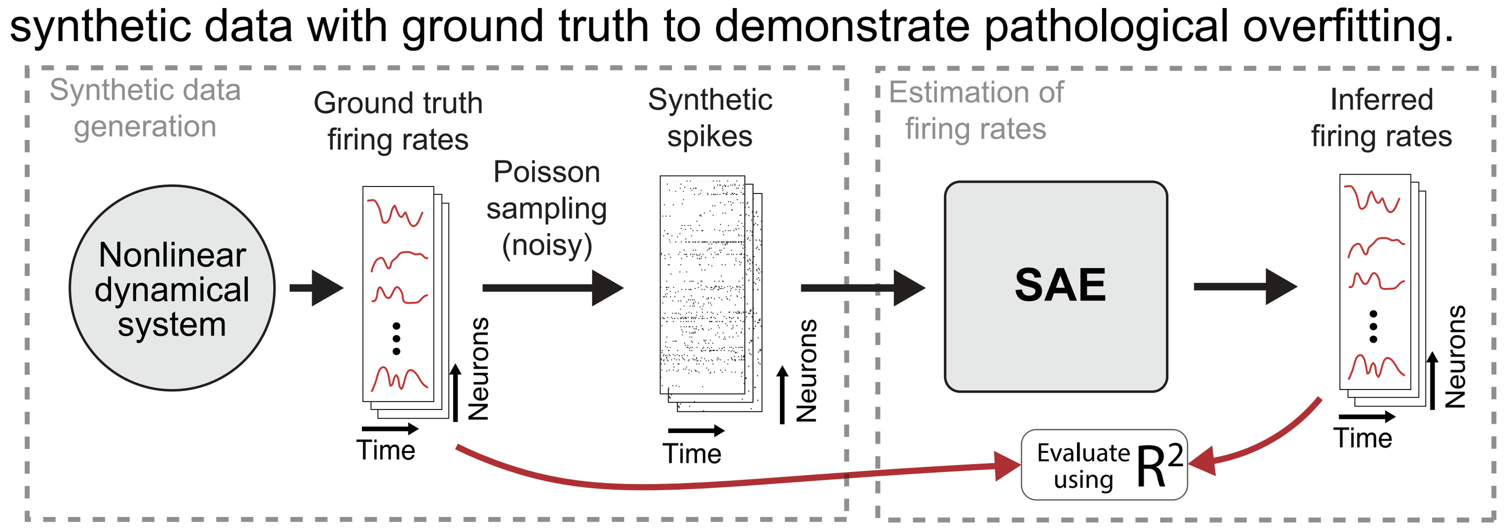
KL penalty: $\mathcal{L}^{KL} = \left\langle D_{KL}\left(\mathcal{N}(\mathbf{g}_0 \mid \boldsymbol{\mu}^{g_0}, \boldsymbol{\sigma}^{g_0}) \parallel P^{g_0}(\mathbf{g}_0)\right)\right\rangle_{\mathbf{z}} + \left\langle D_{KL}\left(\mathcal{N}(\mathbf{u}_1 \mid \boldsymbol{\mu}_1^{u}, \boldsymbol{\sigma}_1^{u}) \parallel P^{u_1}(\mathbf{u}_1)\right)\right\rangle_{\mathbf{z},\mathbf{u}_1} + \left\langle \sum_{t=2}^{T} D_{KL}\left(\mathcal{N}(\mathbf{u}_t \mid \boldsymbol{\mu}_t^{u}, \boldsymbol{\sigma}_t^{u}) \parallel P^{u}(\mathbf{u}_t|\mathbf{u}_{t-1})\right)\right\rangle_{\mathbf{z},\mathbf{u}},$

Maximize Evidence Lower Bound ELBO (L):

$\log P(\mathbf{x}_{1:T}) \geq \mathcal{L} = \mathcal{L}^x - \mathcal{L}^{KL}$

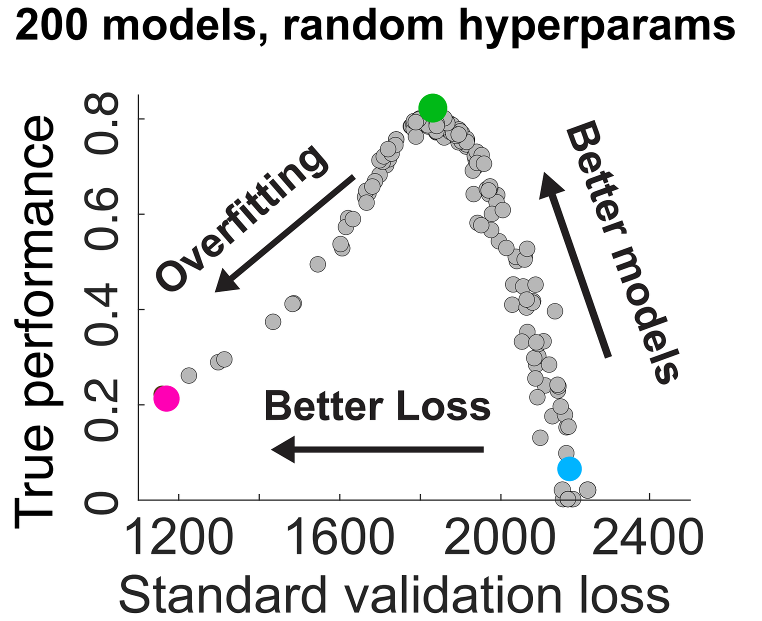
Simulation setup[2]

Used synthetic data with ground truth to demonstrate pathological overfitting.

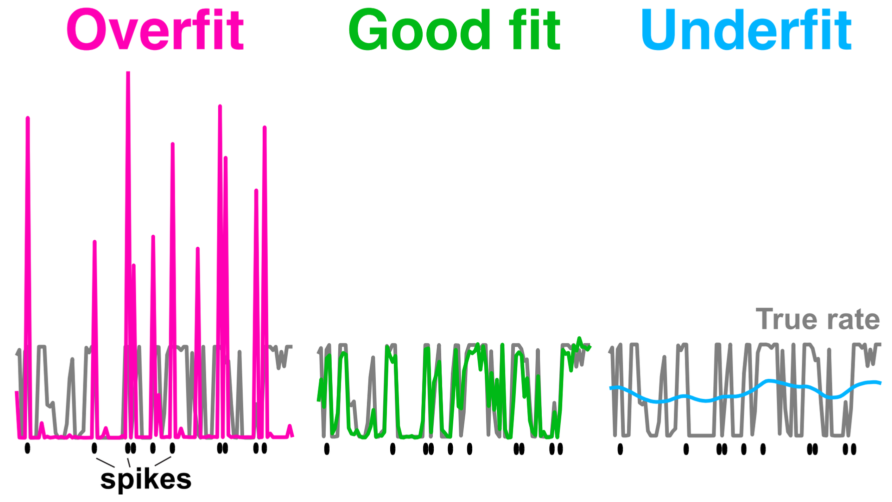


Pathological overfitting in SAEs

200 models, random hyperparams



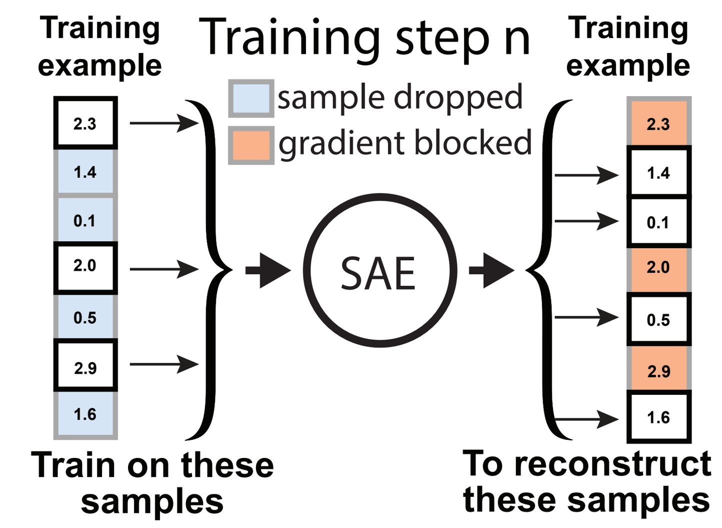
Example rate for one neuron

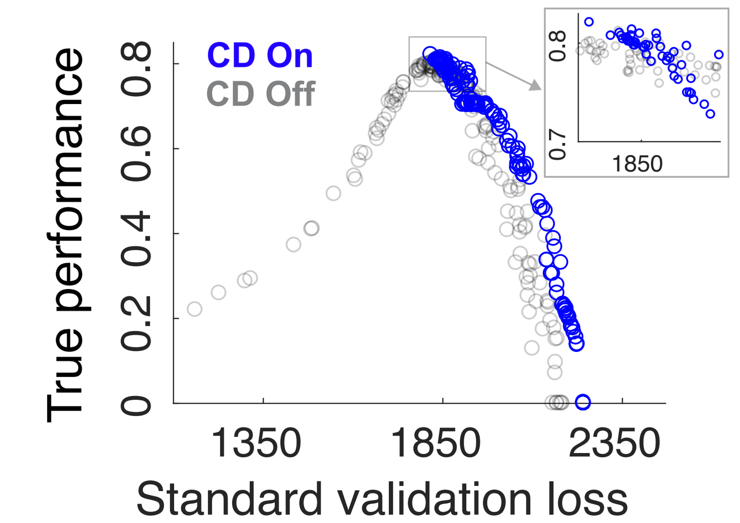


Two solutions to address pathological overfitting in SAEs

1- Coordinated dropout (CD)

CD forces the network to only model shared structure underlying the observations. CD first passes in a subset of samples at the input (by applying dropout). Next, to update network weights, CD only uses gradients calculated for reconstruction of the complementary subset of samples.





Simulation example, linear over-complete autoencoder:

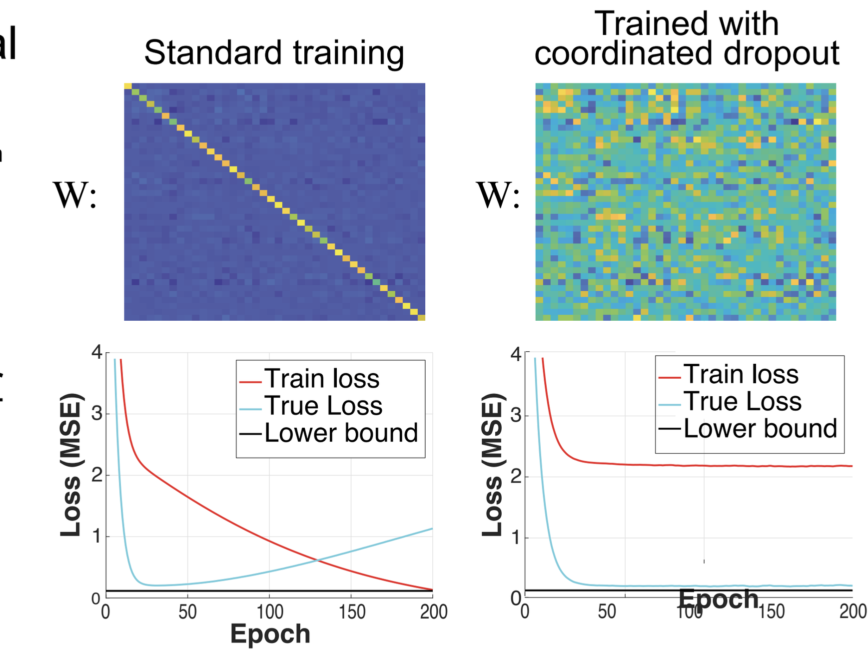
CD is equivalent to preventing diagonal weights to be trained.

$Y = W_p^T Q$ low-d data: $Q \in \mathbb{R}^{m\times n}$; high-d observation: $X \in \mathbb{R}^{o\times n}$; latent variable: $Z \in \mathbb{R}^{h\times n}$; $m < o \leq h$

$X = Y + N$ where $N \sim \mathcal{N}(0, \sigma)$

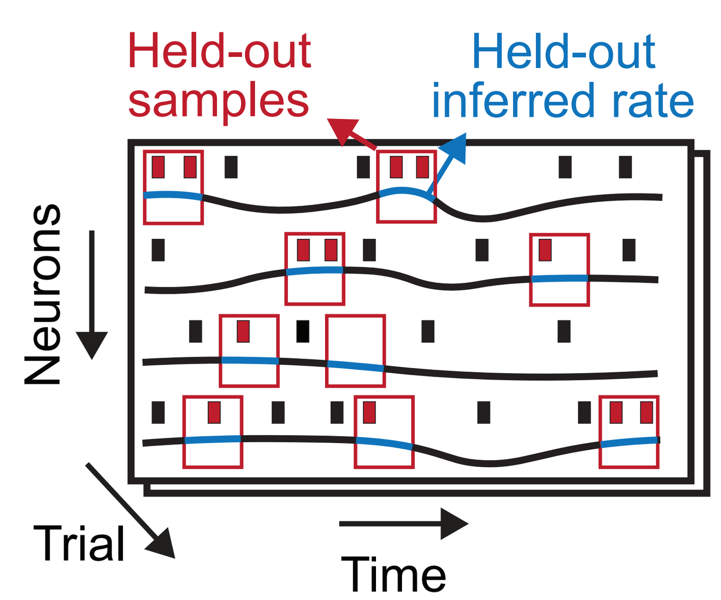
encoder	decoder	auto-encoder
$Z = W_1^T X$ (latent variable)	$X' = W_2 Z$ (Reconstruction)	$X' = W_2 W_1^T X$ (W)

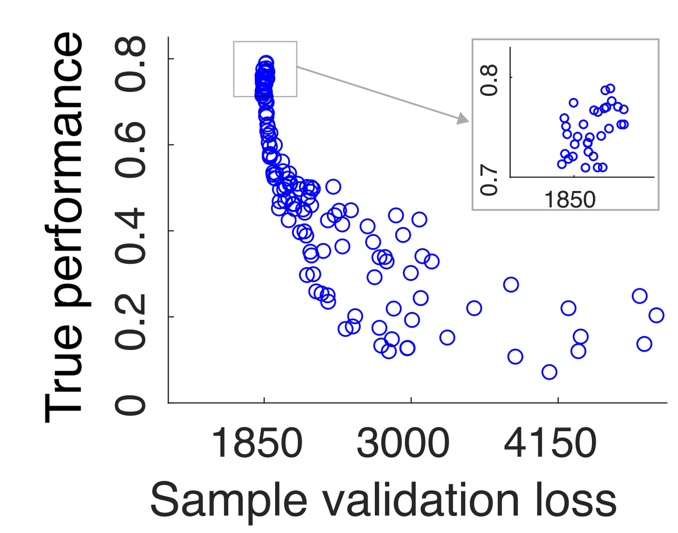
Train loss: $\|X-X'\|^2$ True loss: $\|Y-X'\|^2$



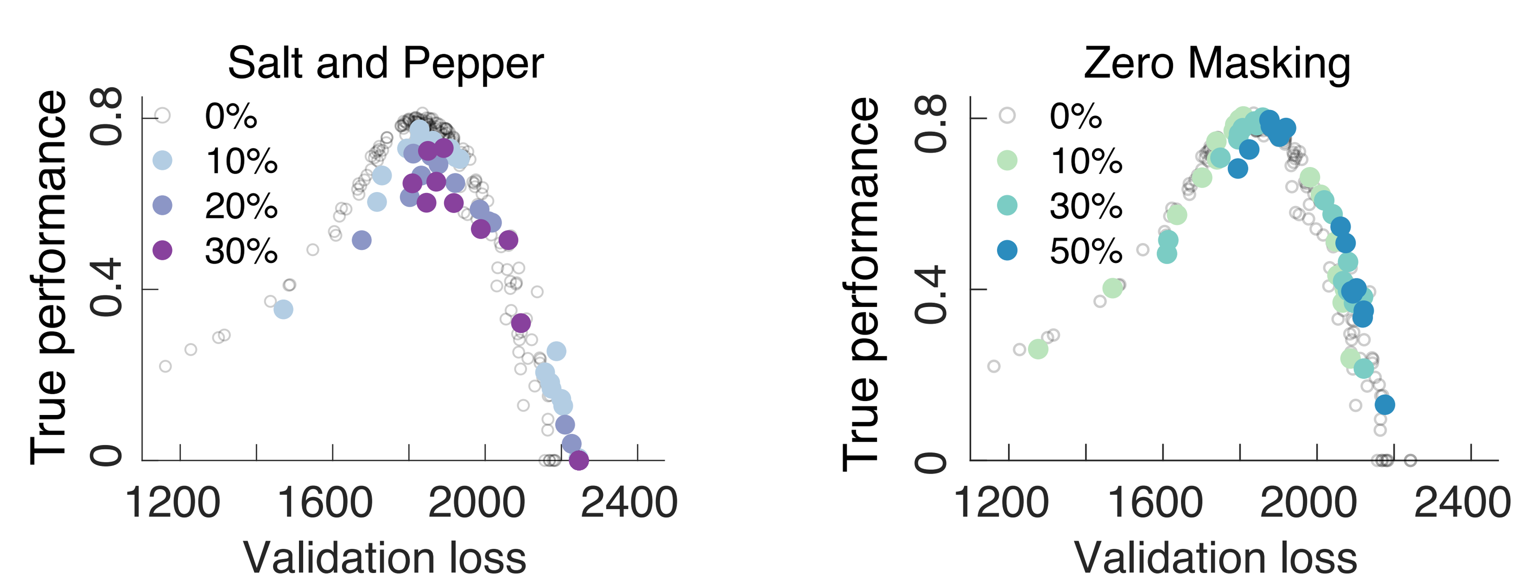
2- Sample validation

Evaluate the network by how well it can predict the rates for the samples it has never seen during training or evaluation.

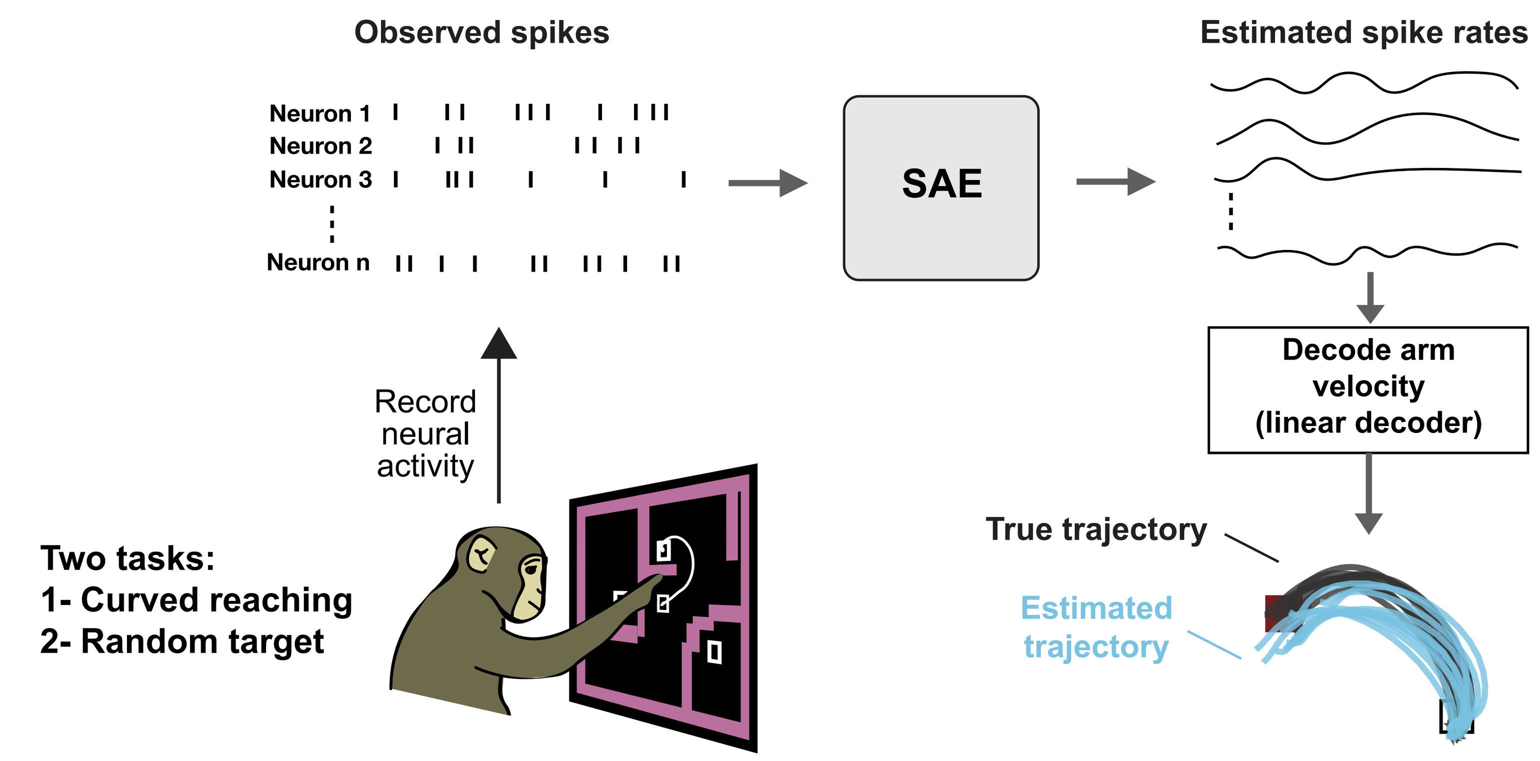




Denoising autoencoders do not address overfitting in SAEs



Experimental setup - decoding arm velocity

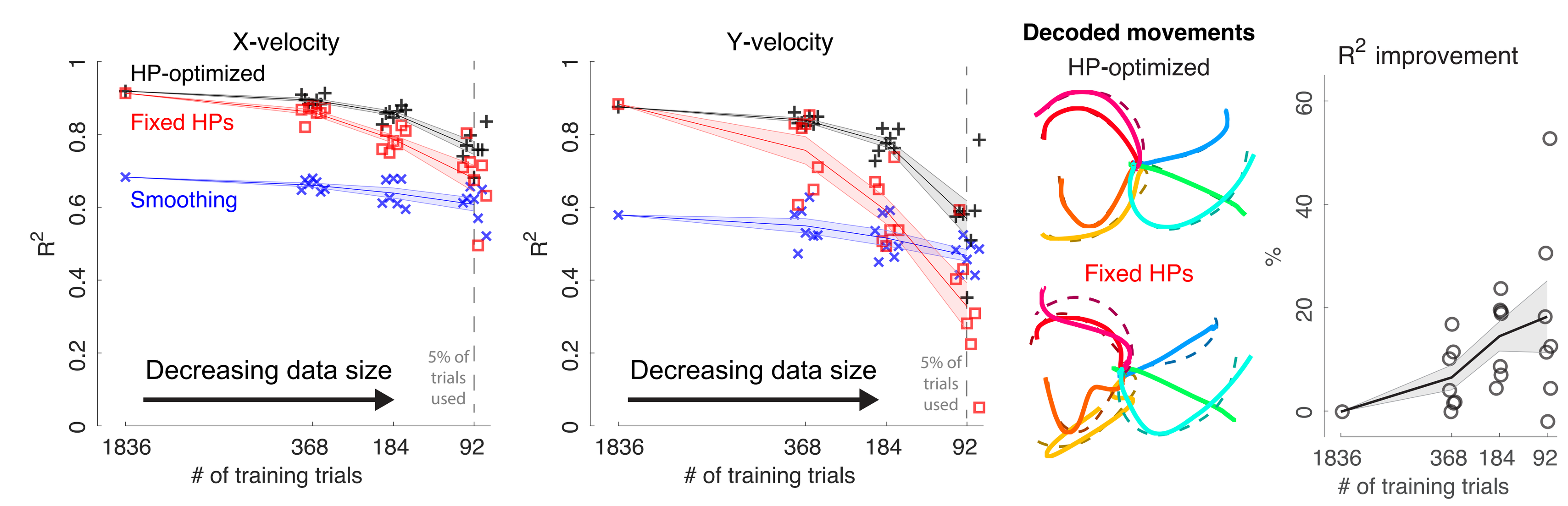


Two tasks:
1- Curved reaching
2- Random target

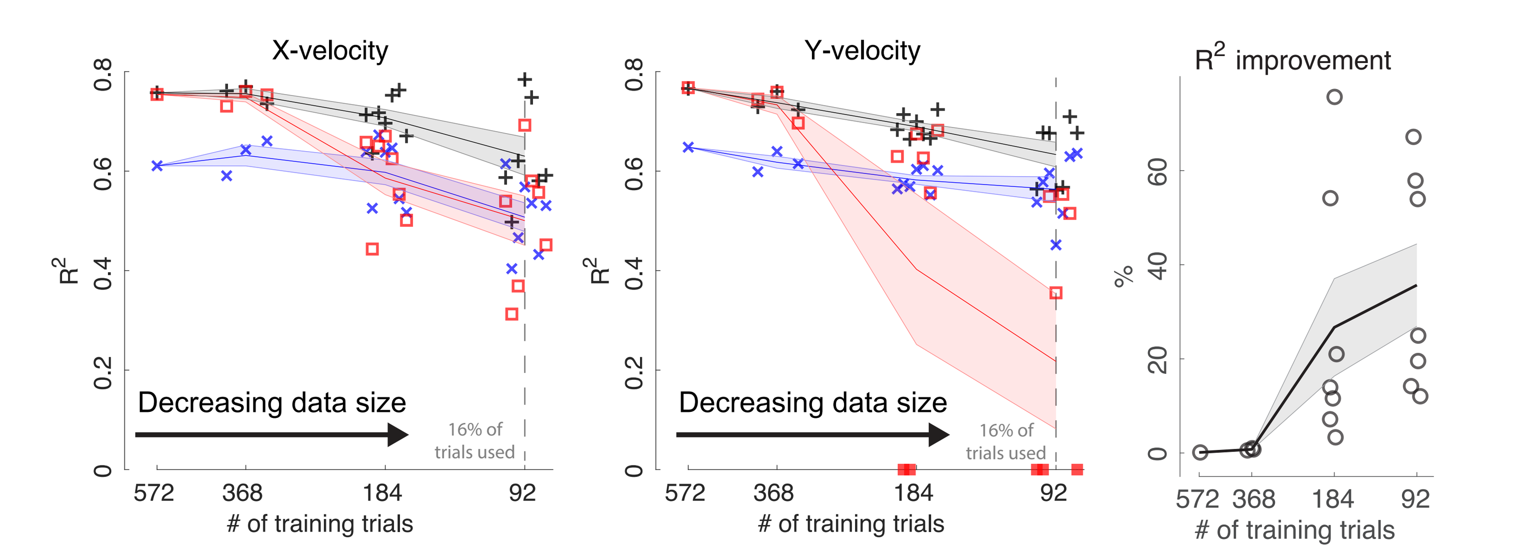
HP optimization trains accurate models with 5-10x less data

HP optimization performed using Population based training (PBT)[3]

Dataset 1: Curved reaching task



Dataset 2: Random target task



Conclusions

- SAEs are prone to a particular type of overfitting that cannot be detected through using standard validation loss.
- Lack of a reliable validation metric prevented HP optimization in SAEs.
- We developed two solutions “Coordinated dropout” and “Sample validation” to address pathological overfitting in SAEs and enable HP optimization.
- HP optimization led to accurate models while using 5-10-fold less training data.

References

[1] Pandarinath, Chethan, Daniel J. O’Shea, Jasmine Collins, et al. “Inferring Single-Trial Neural Population Dynamics Using Sequential Auto-Encoders.” Nature Methods, 2018.

[2] Sussillo, David, Rafal Jozefowicz, L. F. Abbott, and Chethan Pandarinath. "Lfads-latent factor analysis via dynamical systems." arXiv preprint arXiv:1608.06315 (2016).

[3] Jaderberg, Max, Valentin Dalibard, Simon Osindero, et al. "Population based training of neural networks." arXiv preprint arXiv:1711.09846 (2017).

Acknowledgements: This work was supported NSF NCS 1835364, and DARPA Intelligent Neural Interfaces program.