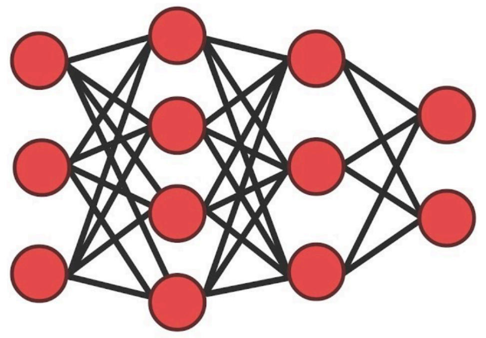
The topology of representation teleportation, regularized Oja's rule, and symmetric weights

Jon Bloom
Institute Scientist, Hail Engineer
Director of Models, Inference, and Algorithms

Co-conspirators: Daniel Kunin, Aleks Goeva, Cotton Seed



Prediction in artificial neural networks is inspired by the brain.
Is *learning* in the brain inspired by artificial neural networks?

COGNITIVE SCIENCE 11, 23–63 (1987)

Competitive Learning: From Interactive Activation to Adaptive Resonance

STEPHEN GROSSBERG
Boston University

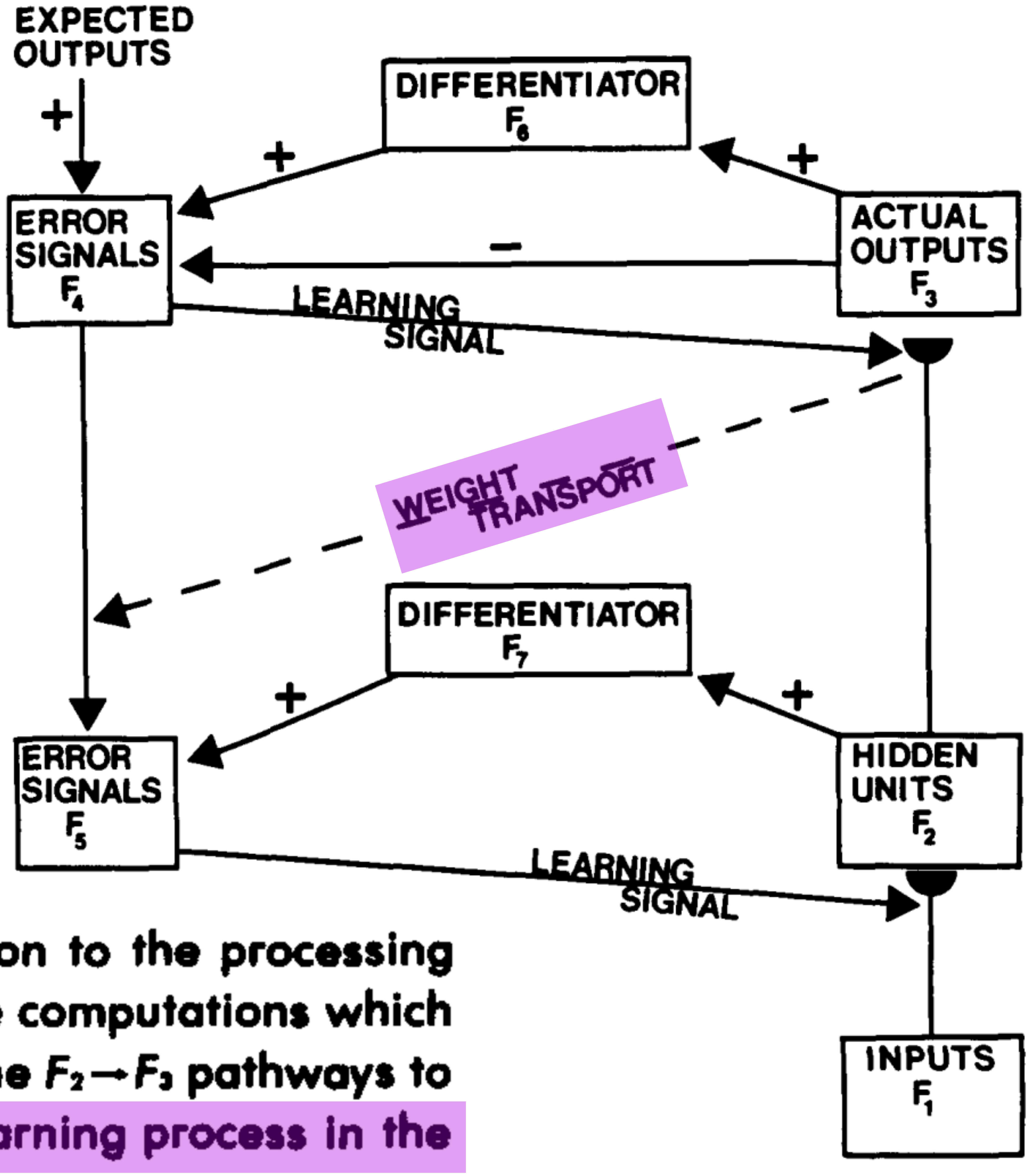
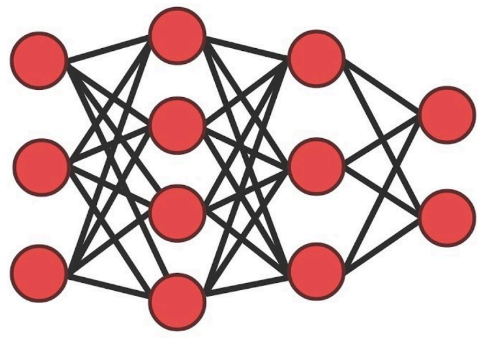
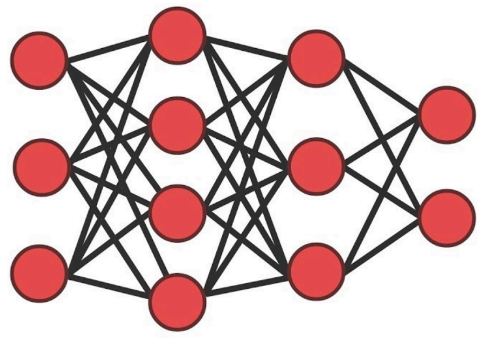


Figure 8. Circuit diagram of the back propagation model: In addition to the processing levels F_1, F_2, F_3, there are also levels F_4, F_5, F_6, and F_7 to carry out the computations which control the learning process. The transport of learned weights from the $F_2 \rightarrow F_3$ pathways to the $F_4 \rightarrow F_5$ pathways shows that this algorithm cannot represent a learning process in the brain.



Prediction in artificial neural networks is inspired by the brain.
Is *learning* in the brain inspired by artificial neural networks?

Such a physical transport of weights has no plausible physical interpretation. The weights in the $F_2 \rightarrow F_3$ pathways must be computed *within* these pathways in order to multiply signals from F_2 to F_3. These weights cannot also exist *within* the pathways from F_4 to F_5 in order to multiply signals from F_4 to F_5 without being physically transported from $(F_2 \rightarrow F_3)$ to $(F_4 \rightarrow F_5)$ pathways, thereby violating basic properties of locality. Moreover, the levels F_3 and F_4 cannot be lumped together, because F_3 must record actual outputs, whereas F_4 must record differences between expected and actual outputs. The *BP* model is thus not a model of a brain process.

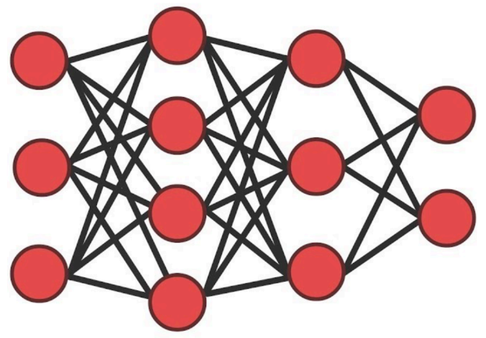


Prediction in artificial neural networks is inspired by the brain.

Is *learning* in the brain inspired by artificial neural networks?

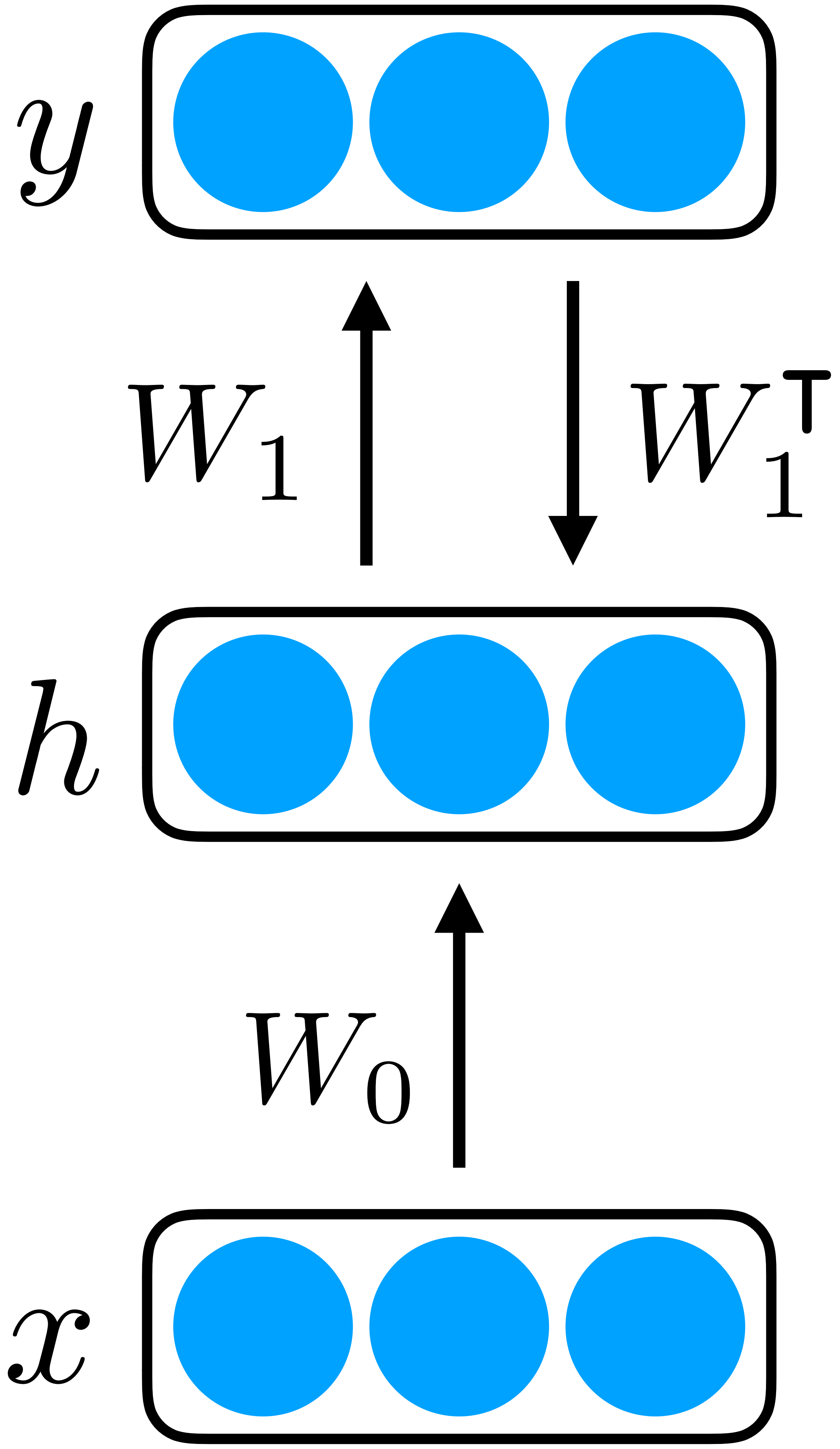
Nope.

I hope you enjoyed my talk!



Prediction in artificial neural networks is inspired by the brain.

Is *learning* in the brain inspired by artificial neural networks?



Backprop

$$e = y - W_1 W_0 x$$

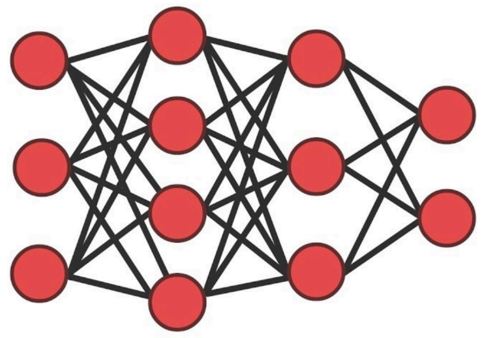
$$\Delta W_1 \propto e h^\intercal$$

$$\Delta W_0 \propto W_1^\intercal e x^\intercal$$

weight transport problem

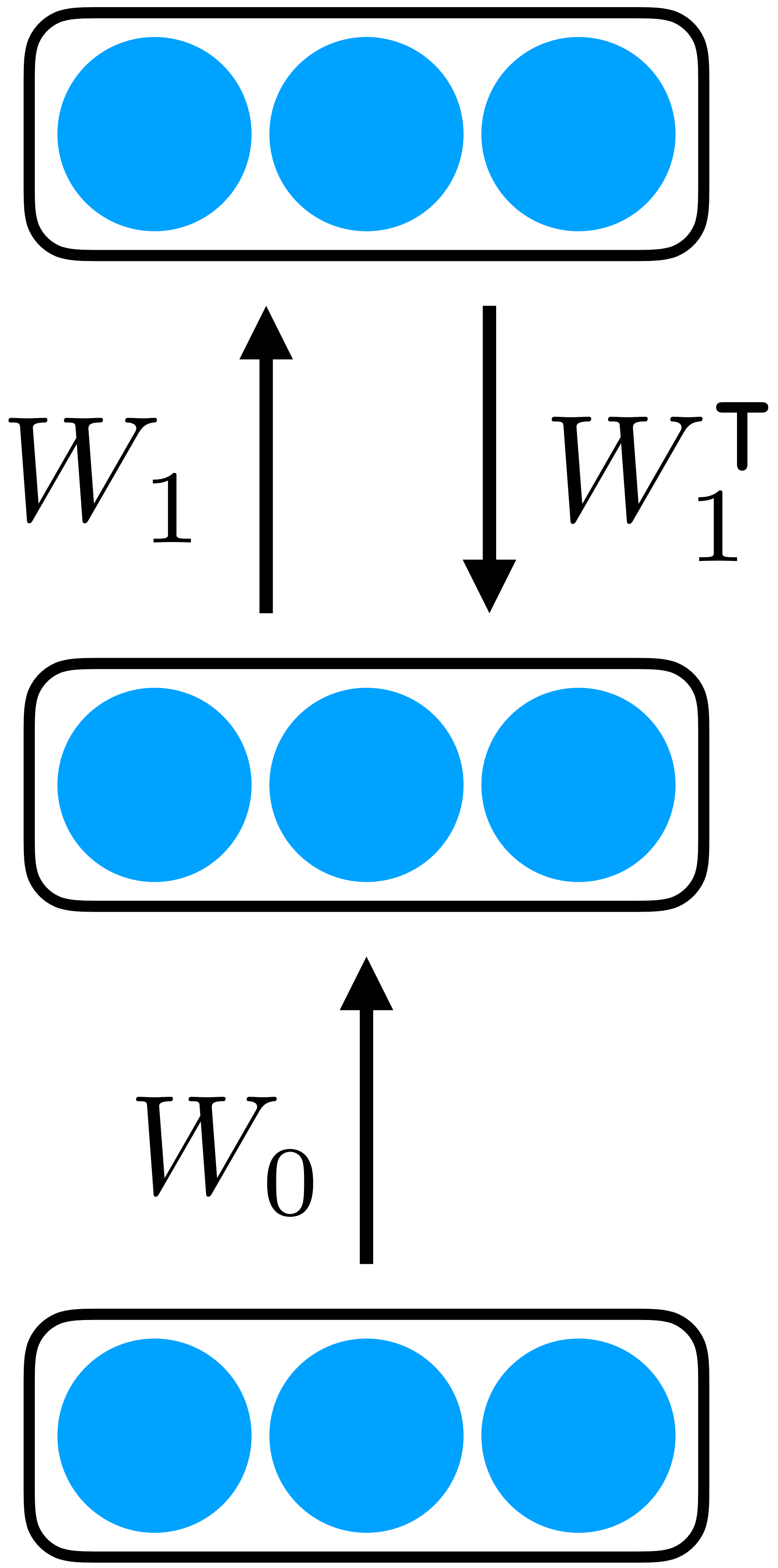
symmetric weights problem

Individual neurons are unidirectional.

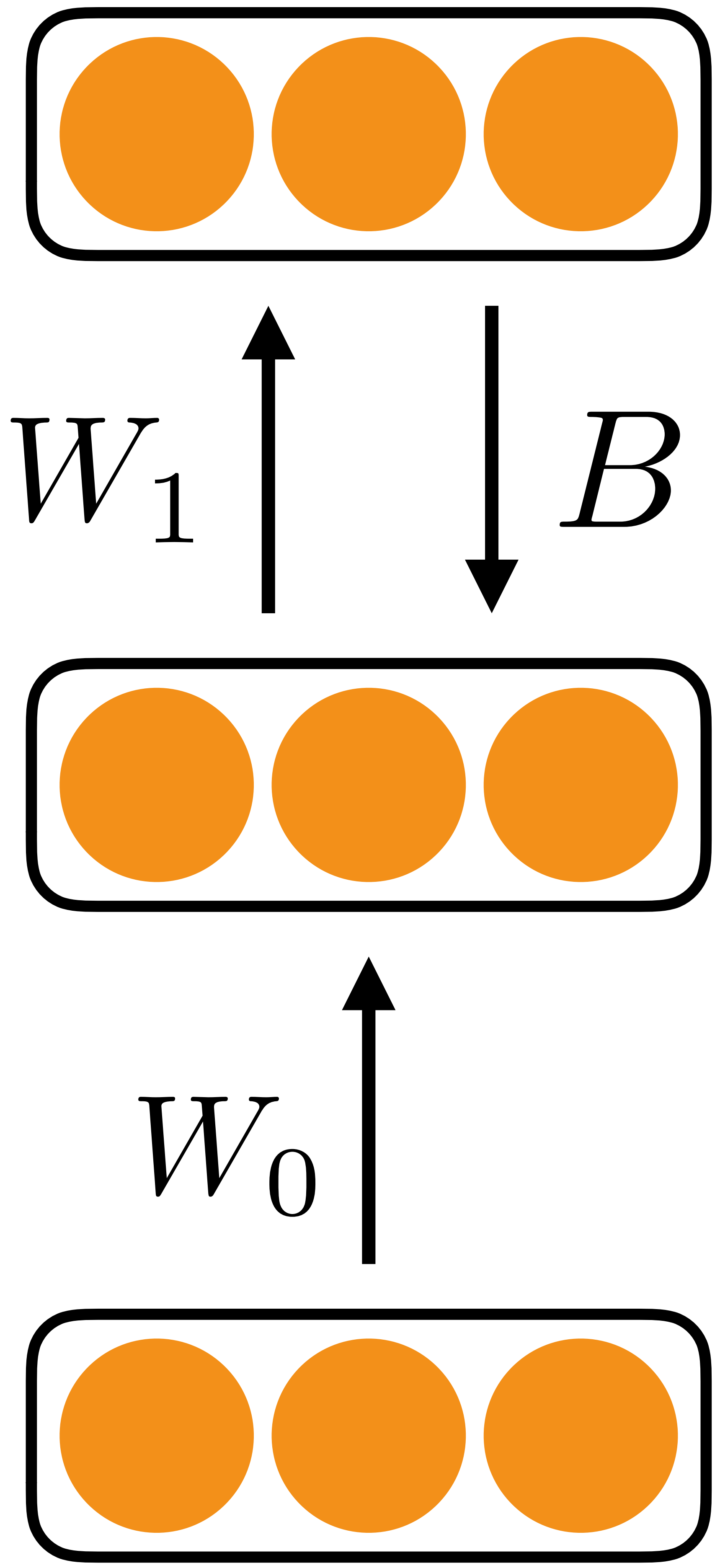


Prediction in artificial neural networks is inspired by the brain.

Is *learning* in the brain inspired by artificial neural networks?



Backprop



Feedback Alignment

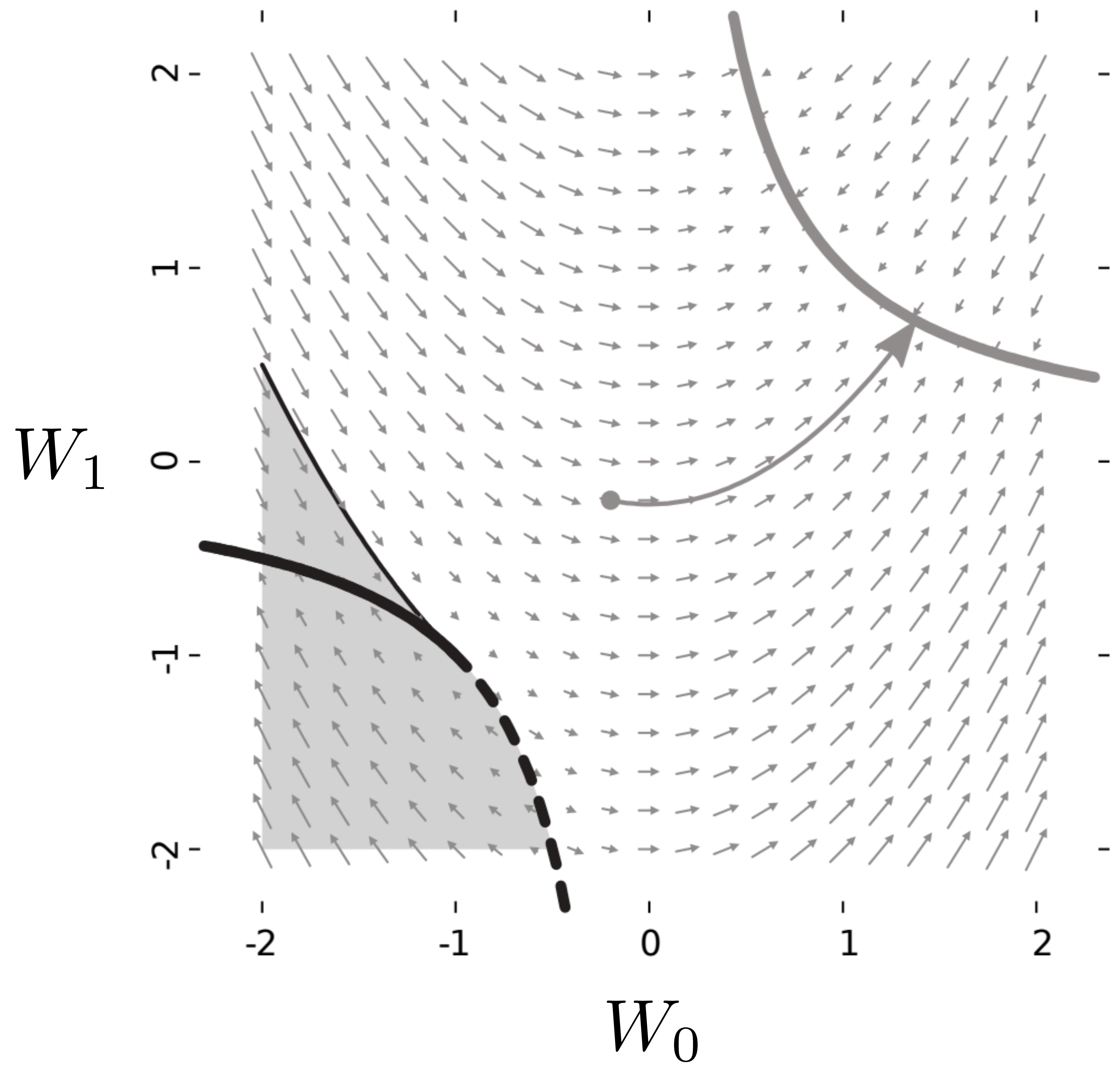
ARTICLE

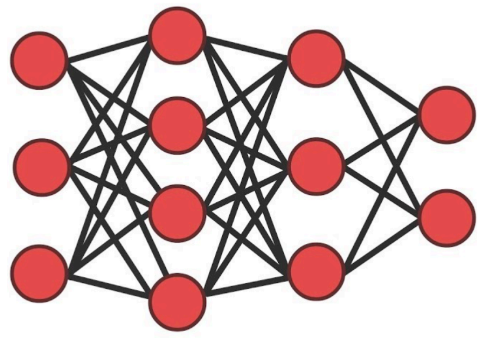
Received 7 Jan 2016 | Accepted 16 Sep 2016 | Published 8 Nov 2016

DOI: 10.1038/ncomms13276 OPEN

Random synaptic feedback weights support error backpropagation for deep learning

Timothy P. Lillicrap[1,2], Daniel Cownden[3], Douglas B. Tweed[4,5] & Colin J. Akerman[1]





Prediction in artificial neural networks is inspired by the brain.

Is *learning* in the brain inspired by artificial neural networks?

Random feedback weights support learning in deep neural networks

Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed, Colin J. Akerman

Difference Target Propagation

Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, Yoshua Bengio

Towards Biologically Plausible Deep Learning

Yoshua Bengio, Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard, Zhouhan Lin

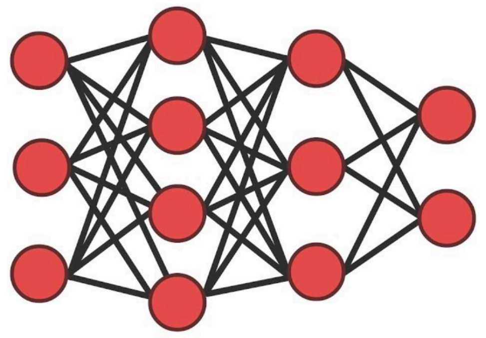
The weight symmetry problem [is] arguably the crux of BP's biological implausibility.

How Important is Weight Symmetry in Backpropagation?

Qianli Liao, Joel Z. Leibo, Tomaso Poggio

Equivalence of Equilibrium Propagation and Recurrent Backpropagation

Benjamin Scellier, Yoshua Bengio



Prediction in artificial neural networks is inspired by the brain.

Is *learning* in the brain inspired by artificial neural networks?

NeurIPS 2018

Assessing the Scalability of Biologically-Motivated Deep Learning Algorithms and Architectures

Sergey Bartunov
DeepMind

Adam Santoro
DeepMind

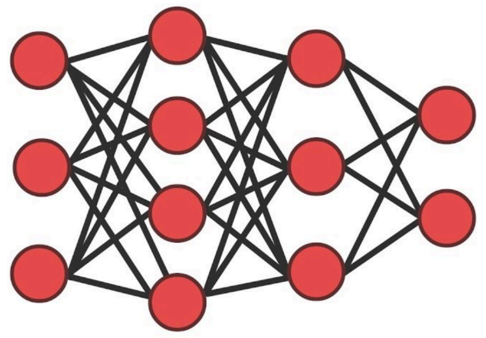
Blake A. Richards
University of Toronto

Luke Marris
DeepMind

Geoffrey E. Hinton
Google Brain

Timothy P. Lillicrap
DeepMind, University College London

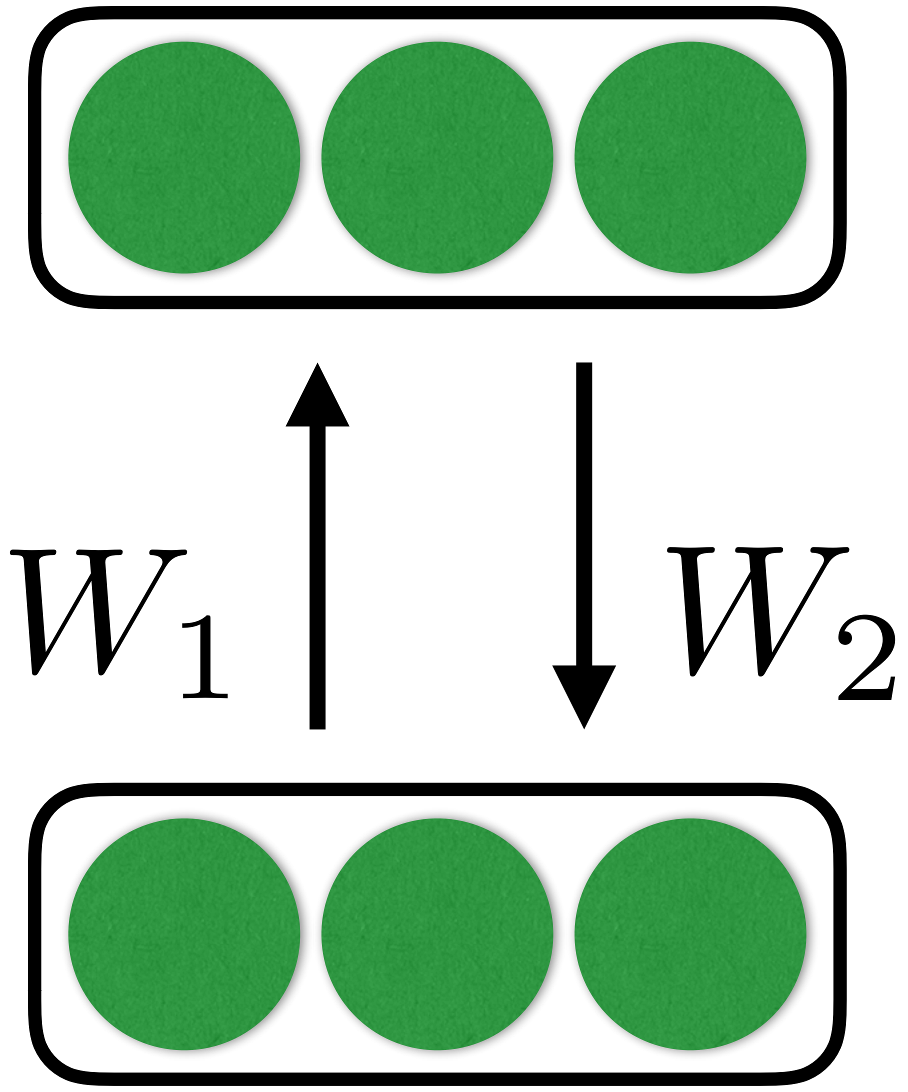
Many of these algorithms perform well for MNIST, but for CIFAR and ImageNet we find that TP and FA variants perform significantly worse than BP, especially for networks composed of locally connected units, opening questions about whether new architectures and algorithms are required to scale these approaches.



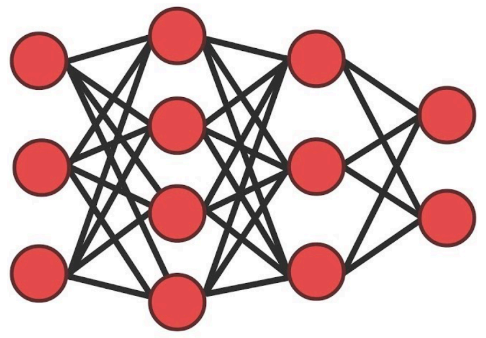
Prediction in artificial neural networks is inspired by the brain.

Is *learning* in the brain inspired by artificial neural networks?

Theorem: L_2-regularized linear autoencoders are *symmetric* at all critical points.



$$W_2 = W_1^{+} \xrightarrow{\text{weight decay}} W_2 = W_1^{T}$$



Prediction in artificial neural networks is inspired by the brain. Is *learning* in the brain inspired by artificial neural networks?

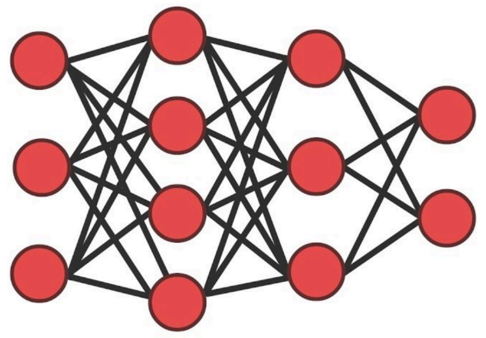
Theorem: L_2-regularized linear autoencoders are *symmetric* at all critical points.

On Sun, Feb 10, 2019 at 2:16 PM Yoshua Bengio <yoshua.bengio@mila.quebec> wrote:

Thanks for reaching out, this is interesting.

The question of obtaining the transpose is actually pretty important for research on a biologically plausible version of backprop, because if you obtain approximate transposes, then several local learning rules give rise to gradient estimator analogues of backdrop.

Cheers,
-- Yoshua

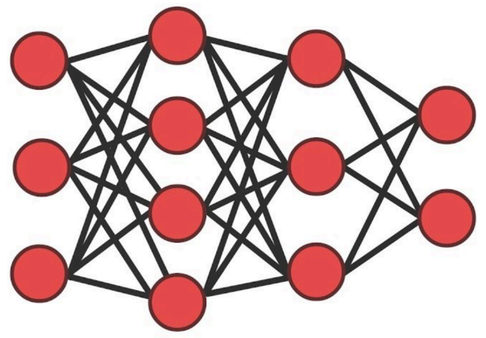


Prediction in artificial neural networks is inspired by the brain.

Is *learning* in the brain inspired by artificial neural networks?

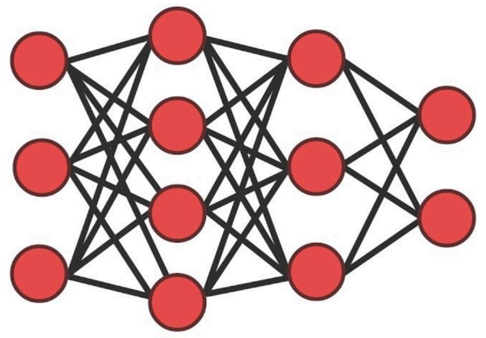
Yes.

By pure logic.



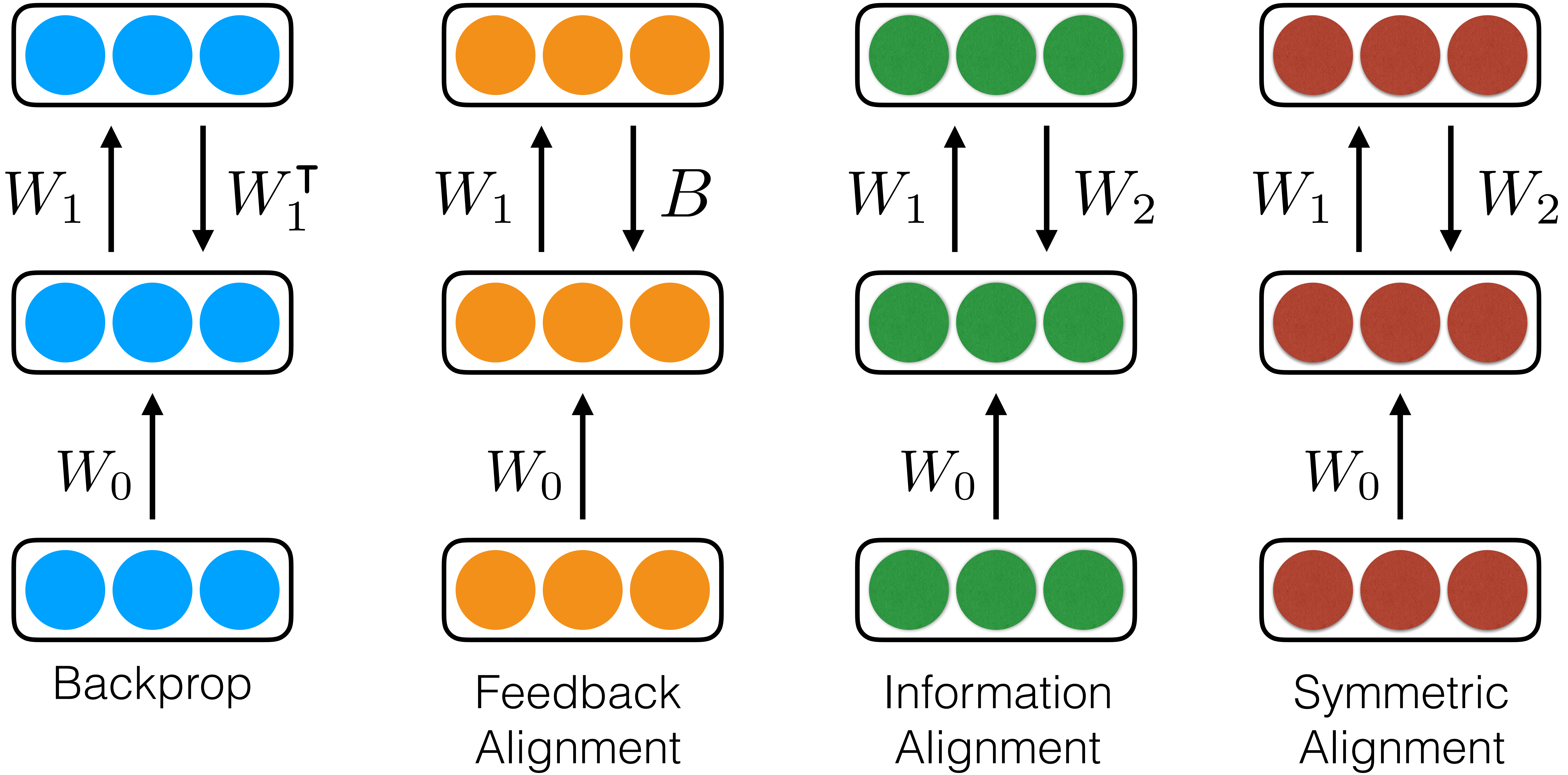
Maybe.

And wouldn't it be fun to build toward statements that could be verified or falsified with rigor?



Prediction in artificial neural networks is inspired by the brain.

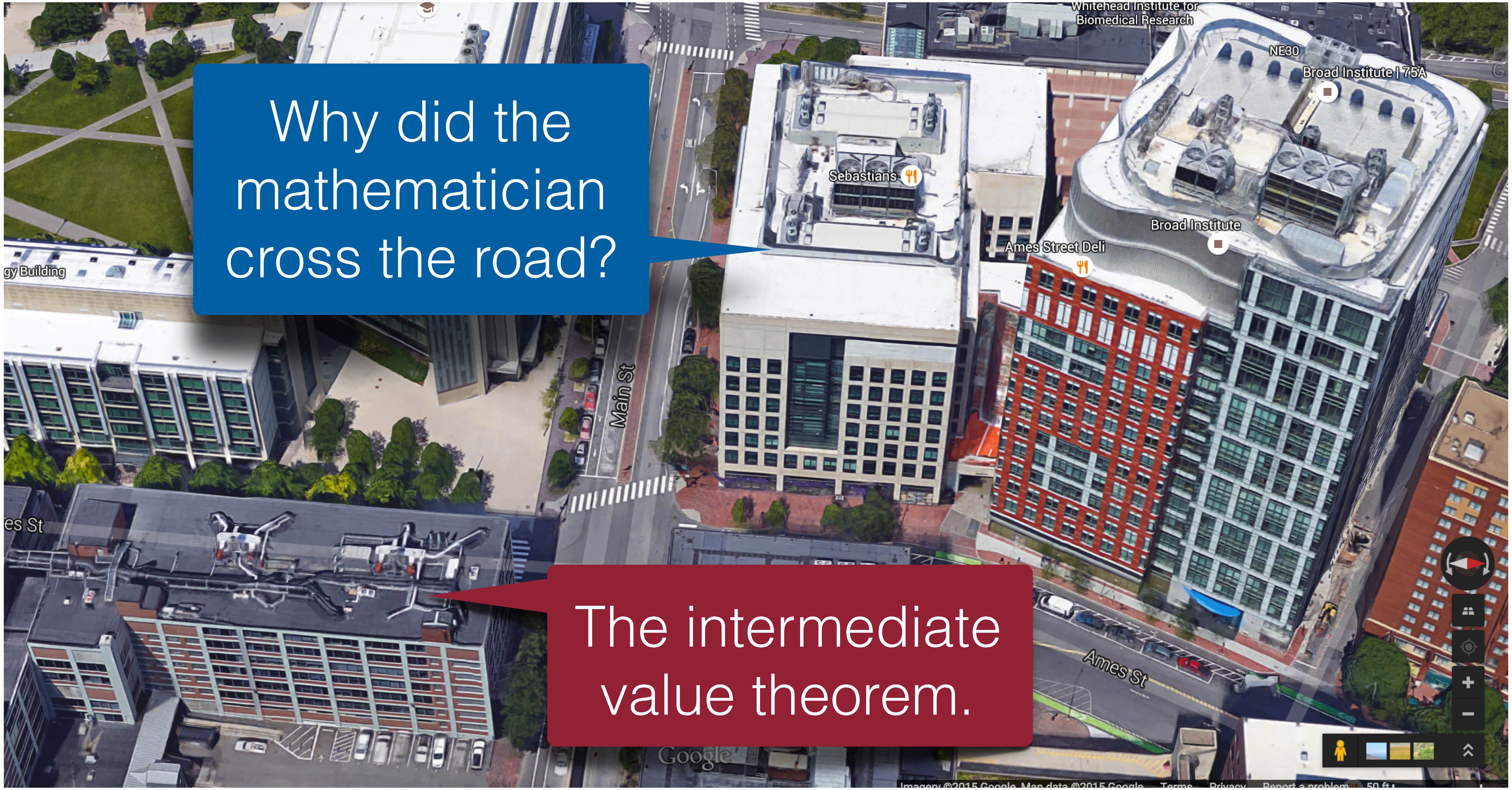
Is *learning* in the brain inspired by artificial neural networks?



CBMM

BROAD INSTITUTE

STANLEY CENTER FOR PSYCHIATRIC RESEARCH AT BROAD INSTITUTE

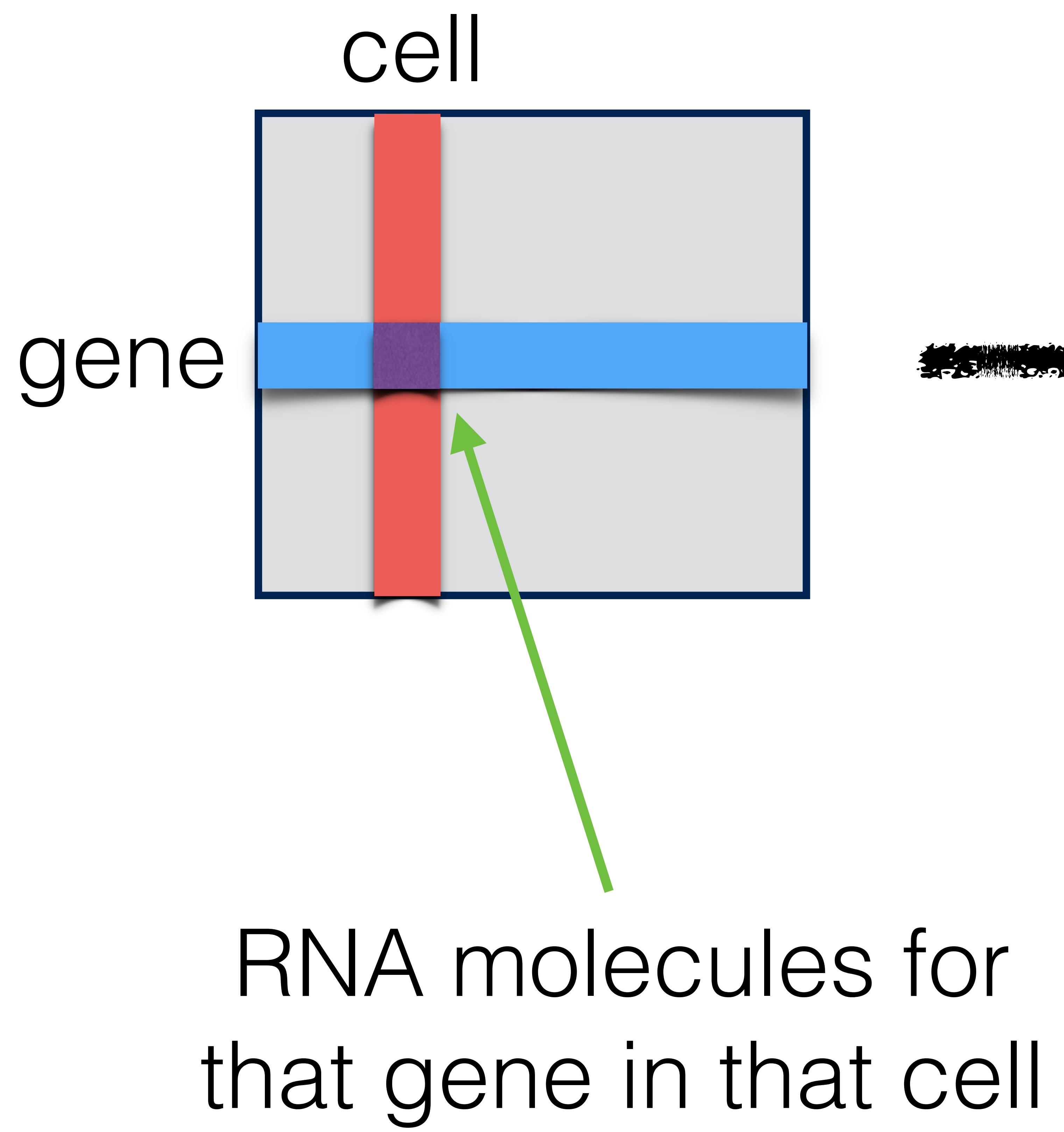


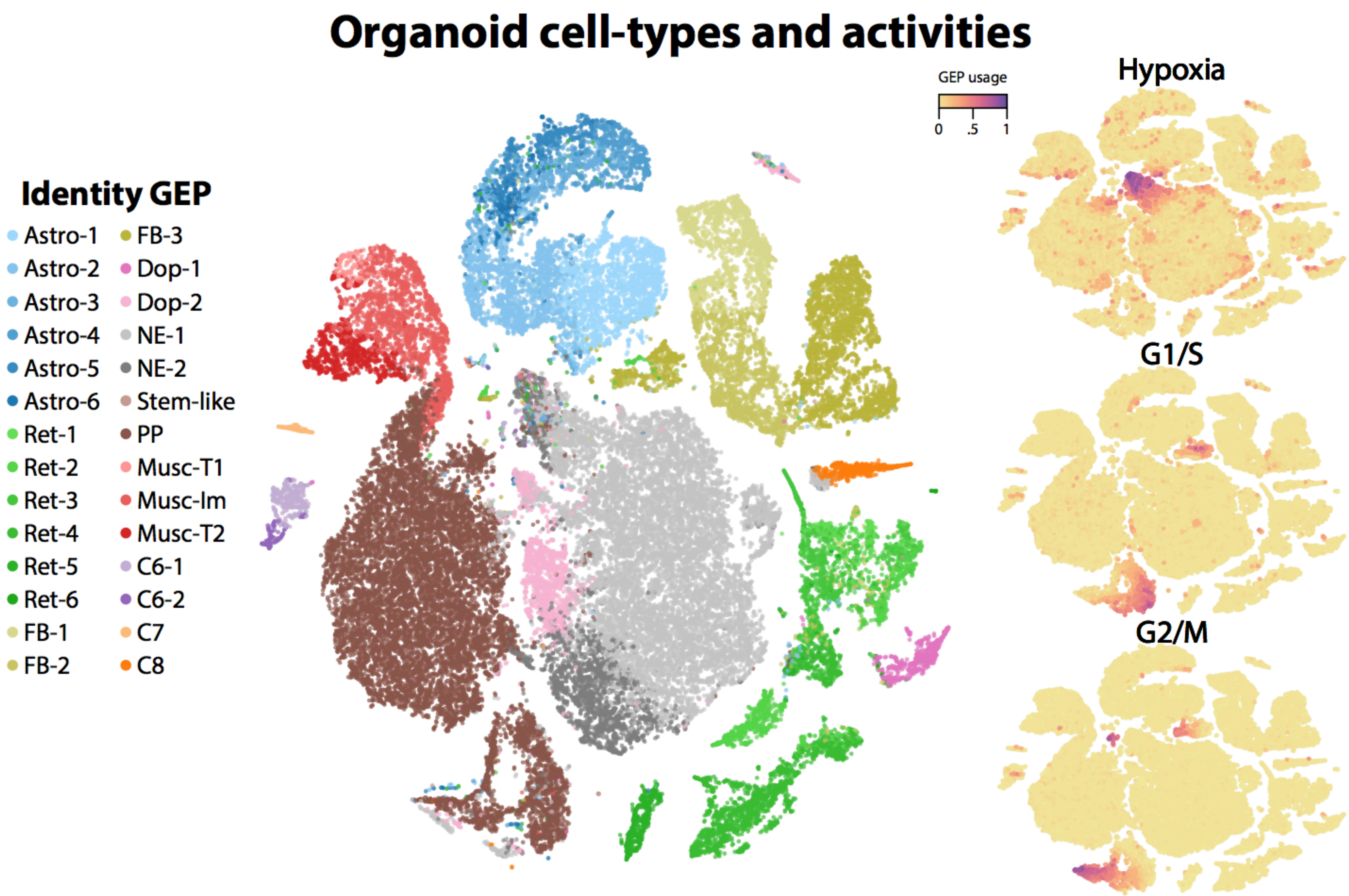
MIT MATHEMATICS

Google AI

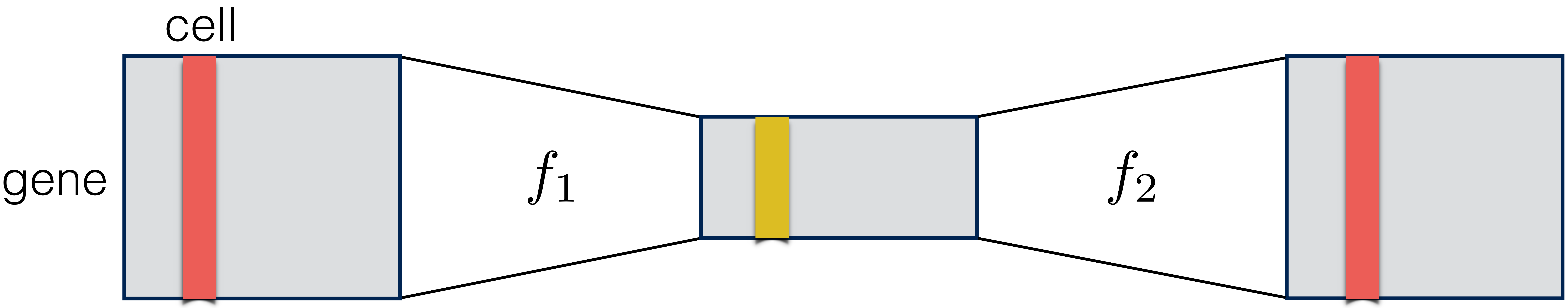
STANLEY CENTER
FOR PSYCHIATRIC RESEARCH
AT BROAD INSTITUTE

STANLEY CENTER
FOR PSYCHIATRIC RESEARCH
AT BROAD INSTITUTE

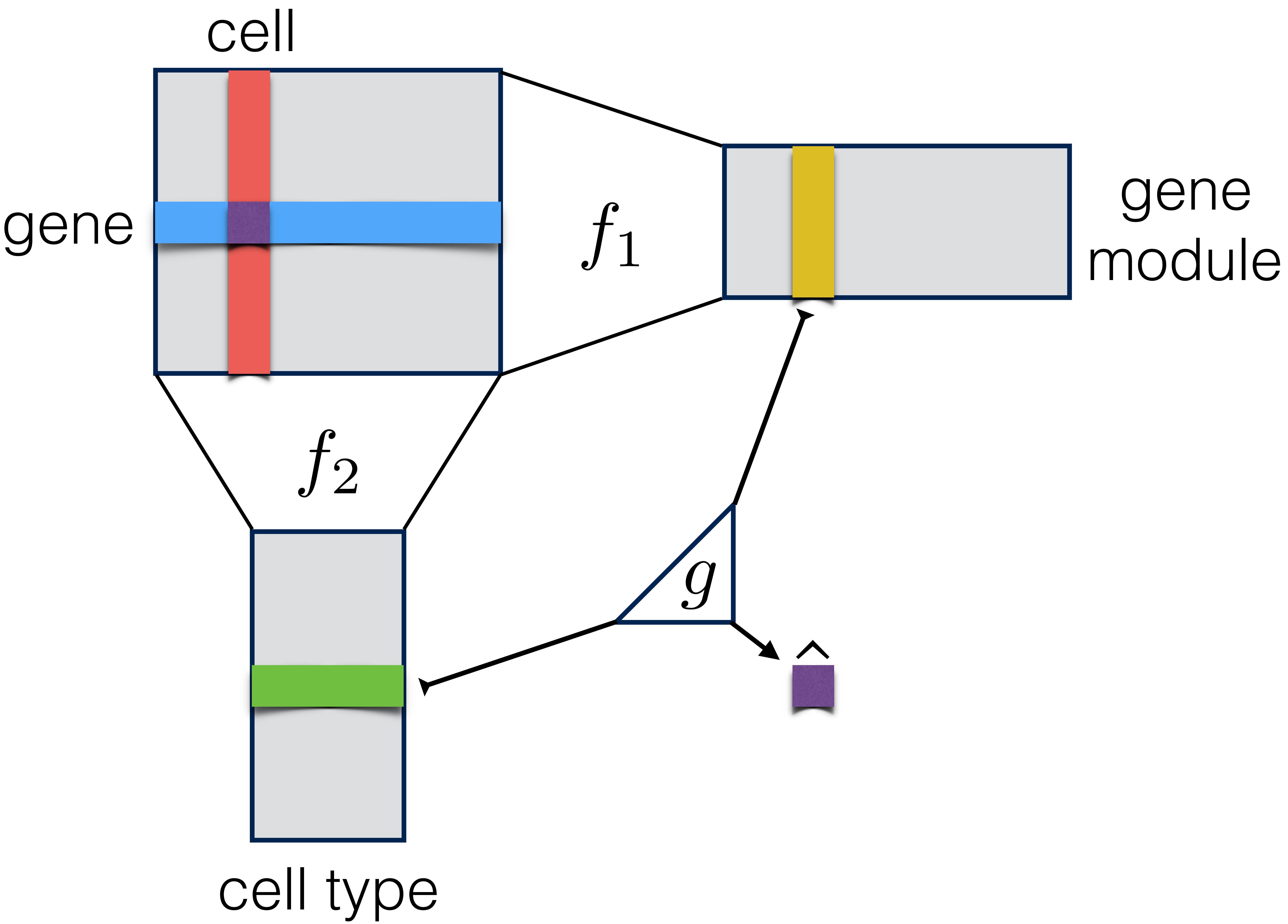




Representation Learning of Cell



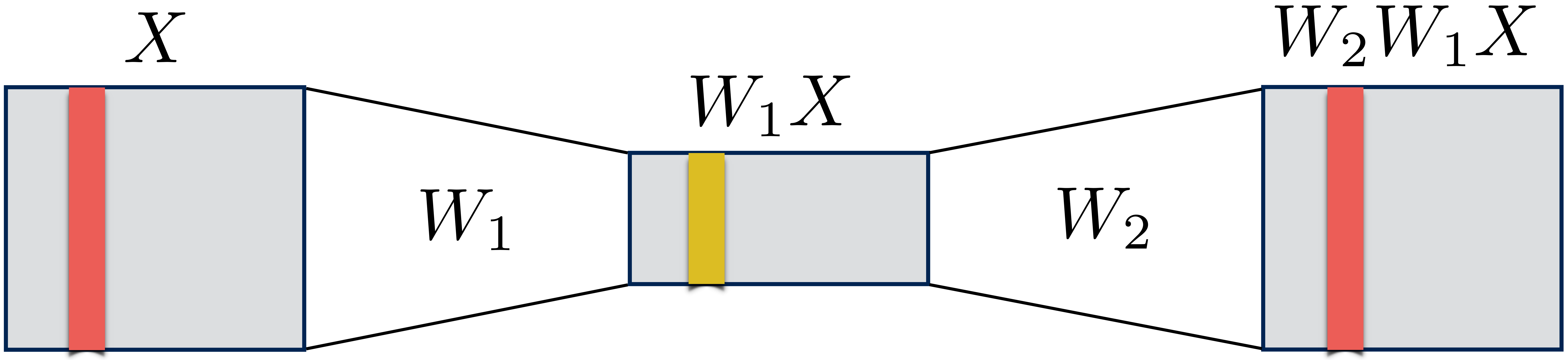
Representation Learning of Cell++



Non-linear
Singular Value
Decomposition

$$X = U\Sigma V^\intercal$$

Linear Autoencoder

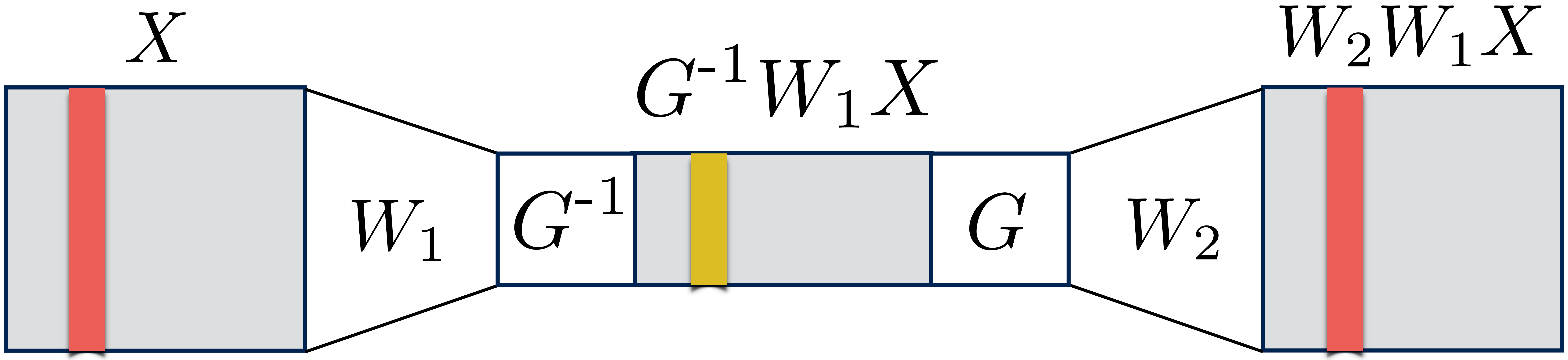


$$\mathcal{L}(W_1, W_2) = ||X - W_2 W_1 X||^2$$

$$X = U \Sigma V^\intercal \qquad W_1 = U_k^\intercal \qquad W_2 W_1 = U_k U_k^\intercal$$

$$X X^\intercal = U \Sigma^2 U^\intercal \qquad W_2 = U_k$$

Linear Autoencoder



$$\mathcal{L}(W_1, W_2) = ||X - W_2W_1X||^2 = ||X - (W_2G)(G^{-1}W_1)X||^2$$

$$X = U\Sigma V^\intercal \qquad W_1 = G^{-1}U_k^\intercal \qquad W_2W_1 = U_kU_k^\intercal$$

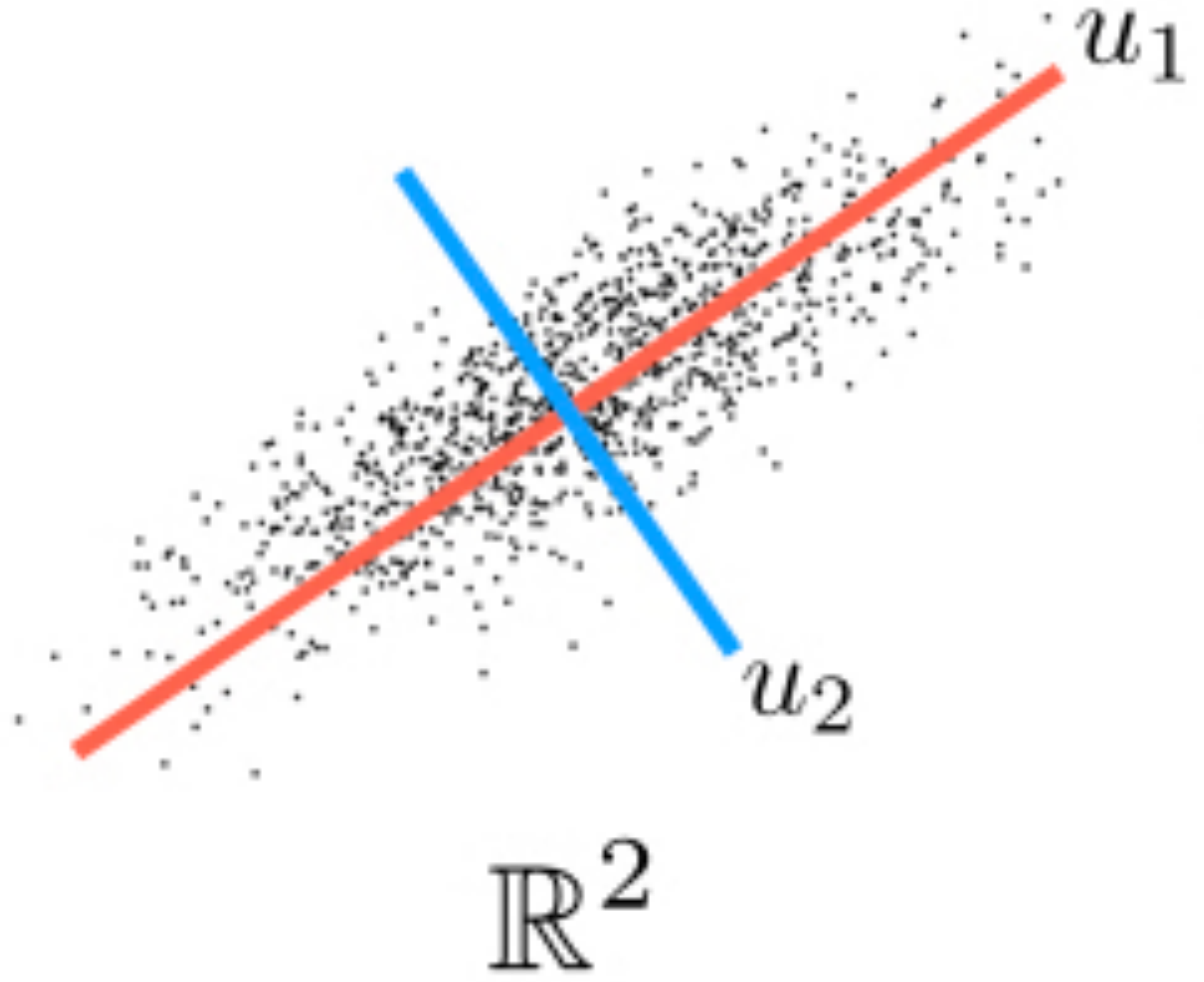
$$XX^\intercal = U\Sigma^2U^\intercal \qquad W_2 = U_kG$$

Fact: Linear autoencoders are *pseudoinverses* at all critical points: $W_2 = W_1^+$

Topology of PCA

<u>Problem</u>
Find the k-plane closest to a point cloud in $\mathbb{R}^m$.

<u>Domain</u>
The manifold whose points are k-planes in $\mathbb{R}^m$.

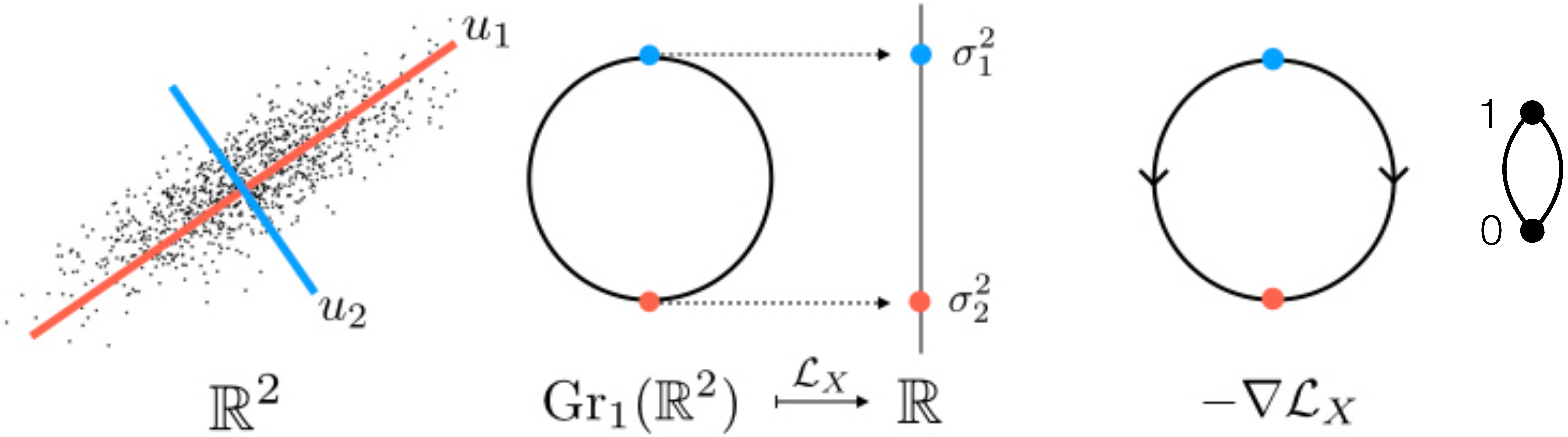


I.e., the *Grassmannian* manifold:

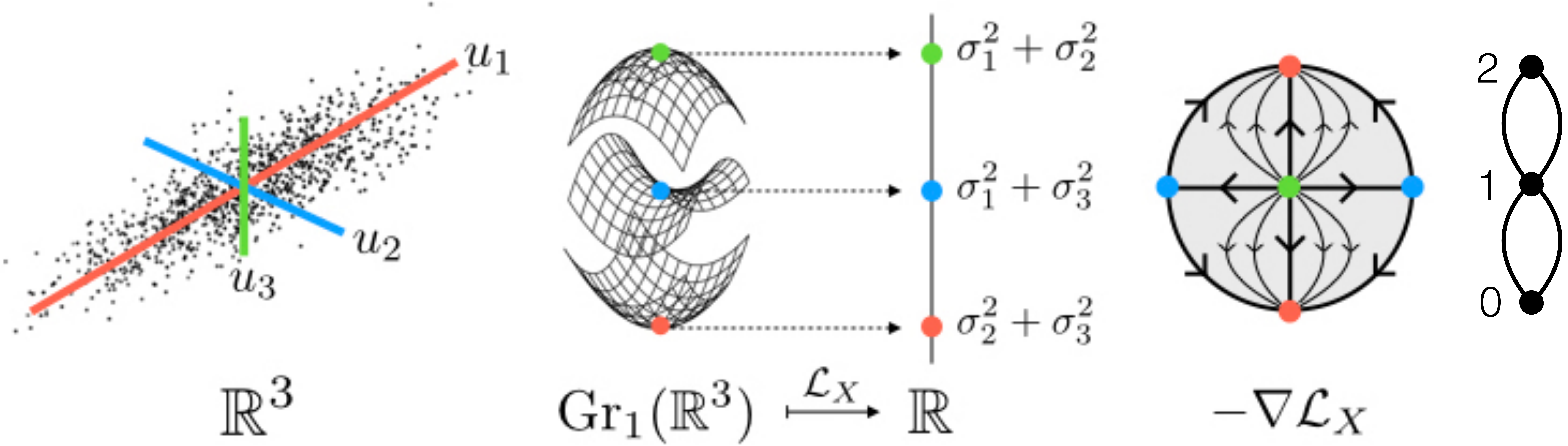
$$\mathrm{Gr}_k(\mathbb{R}^m) \cong \{P = P^2,\ P = P^\intercal,\ \mathrm{tr}\, P = k\} \subset \mathbb{R}^{m \times m}$$

$$\mathcal{L}_X : \mathrm{Gr}_k(\mathbb{R}^m) \to \mathbb{R} \qquad \mathcal{L}_X(P) = ||X - PX||^2$$

Topology of PCA



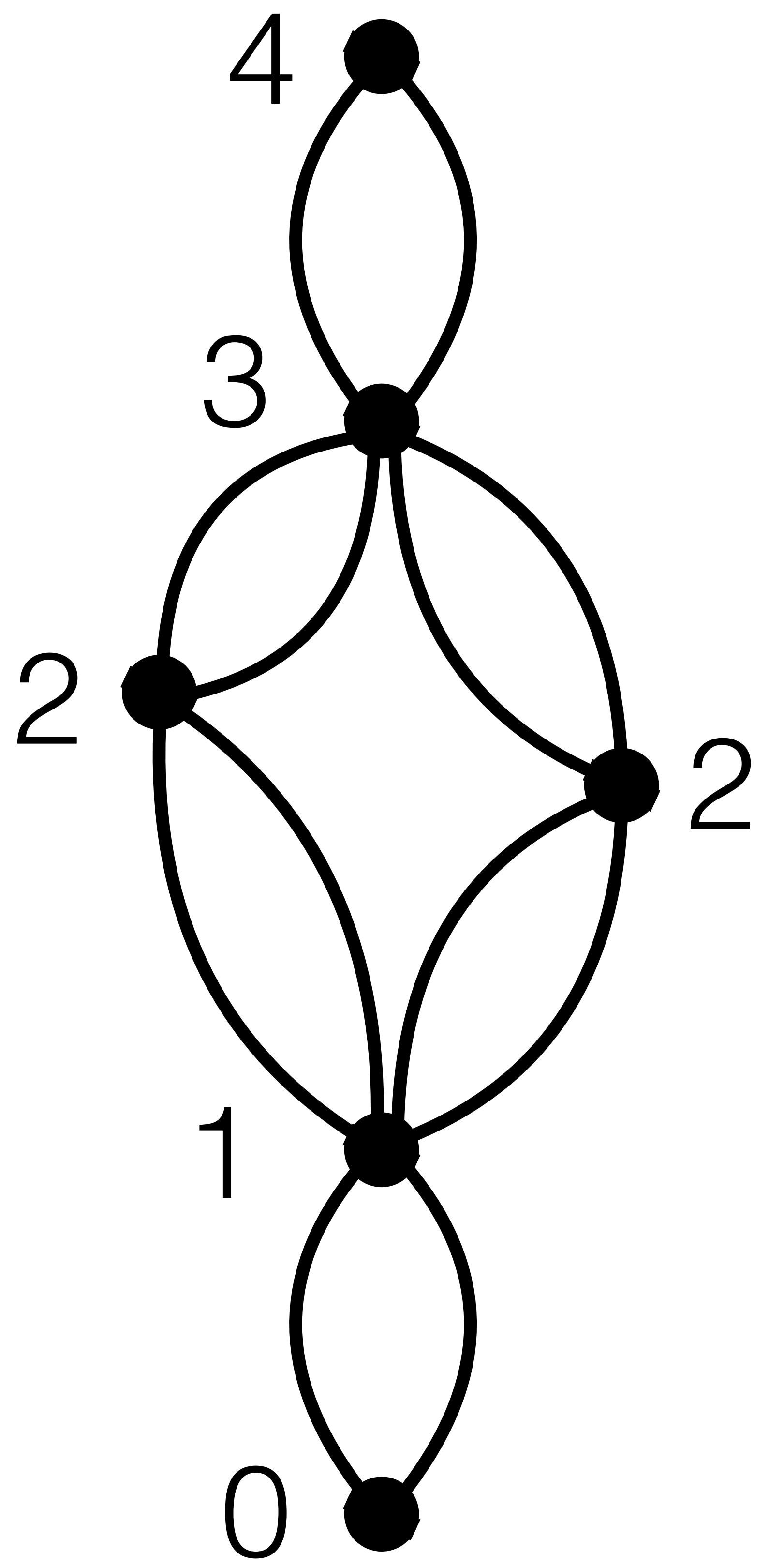
Topology of PCA



Topology of PCA

$$\mathcal{L}_X : \mathrm{Gr}_2(\mathbb{R}^4) \to \mathbb{R}$$

d	u_1	u_2	u_3	u_4
4			•	•
3		•		•
2		•	•	
2	•			•
1	•		•	
0	•	•		

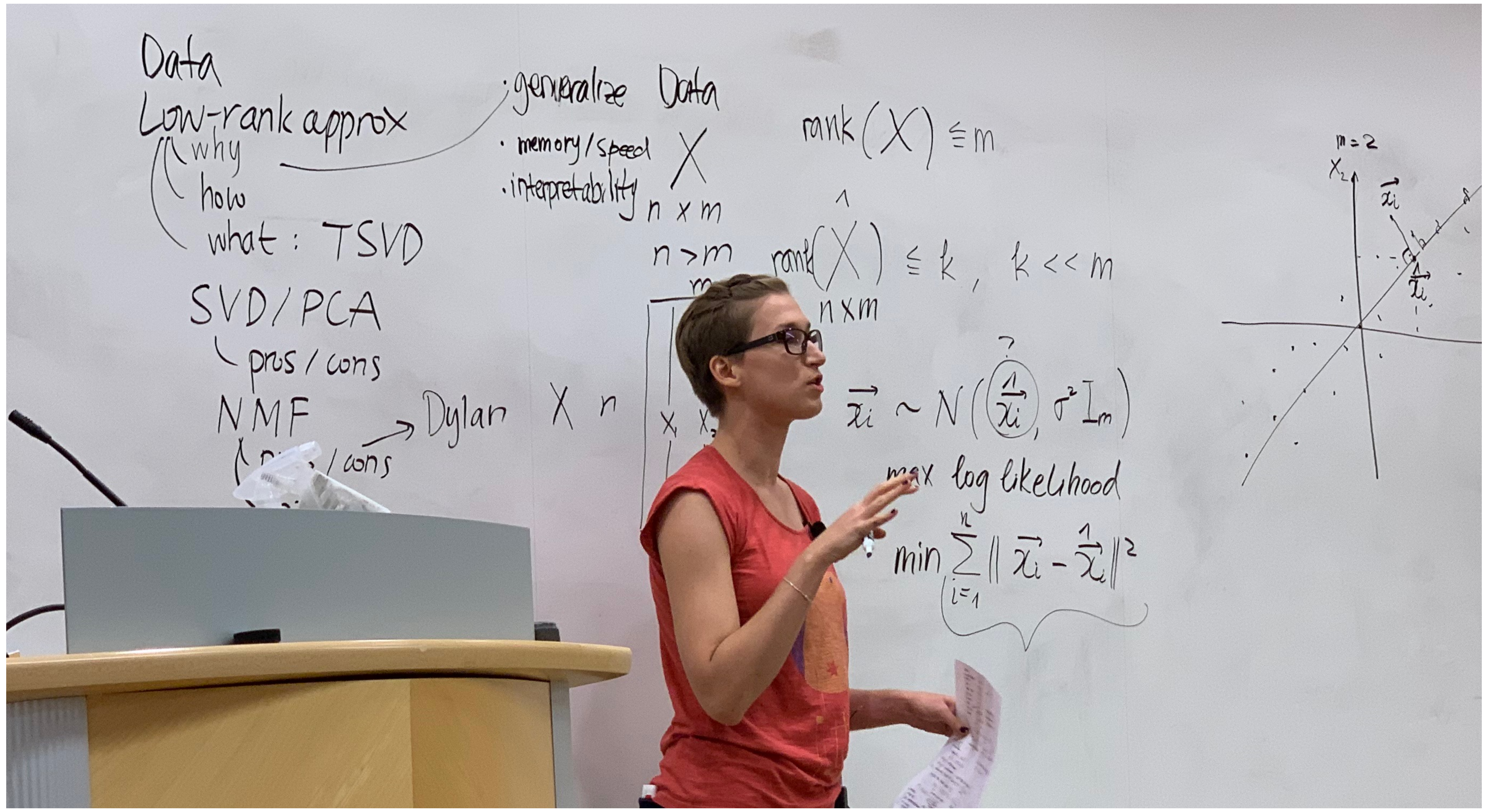


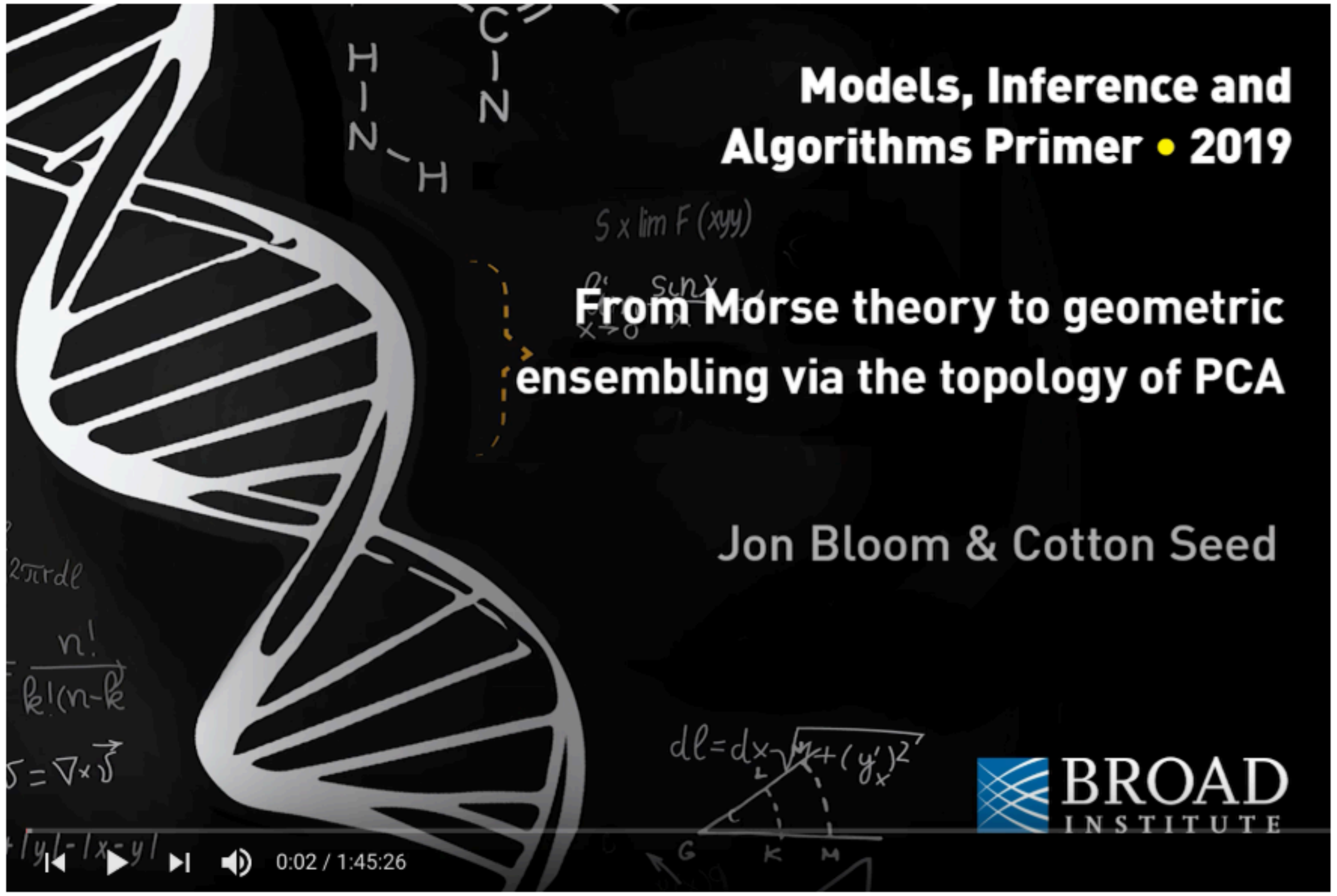
$$\dim \mathrm{Gr}_k(\mathbb{R}^m) = k(m-k)$$

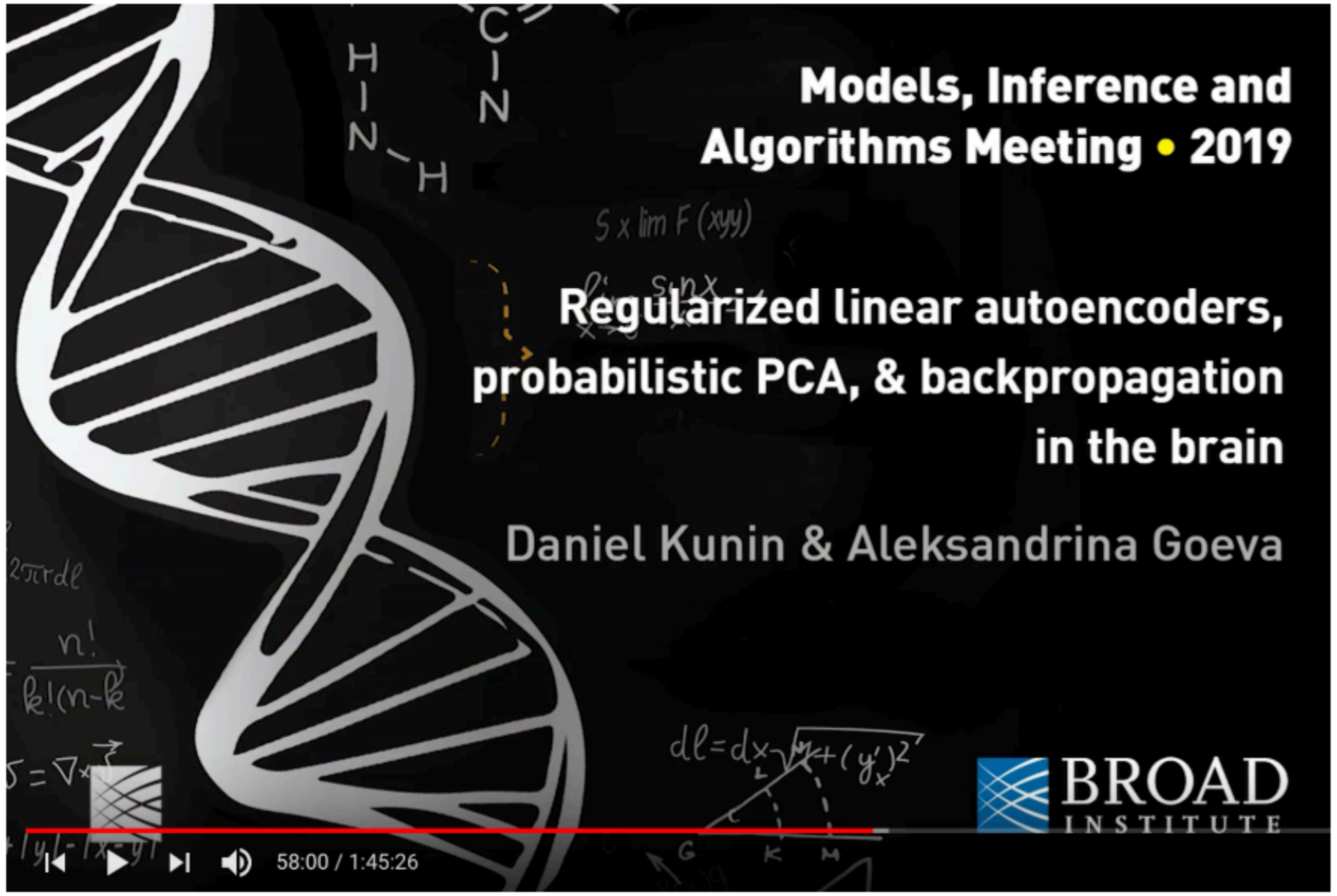
*To visualize 4 dimensions, first visualize n dimensions and then let n = 4.

Topology of PCA

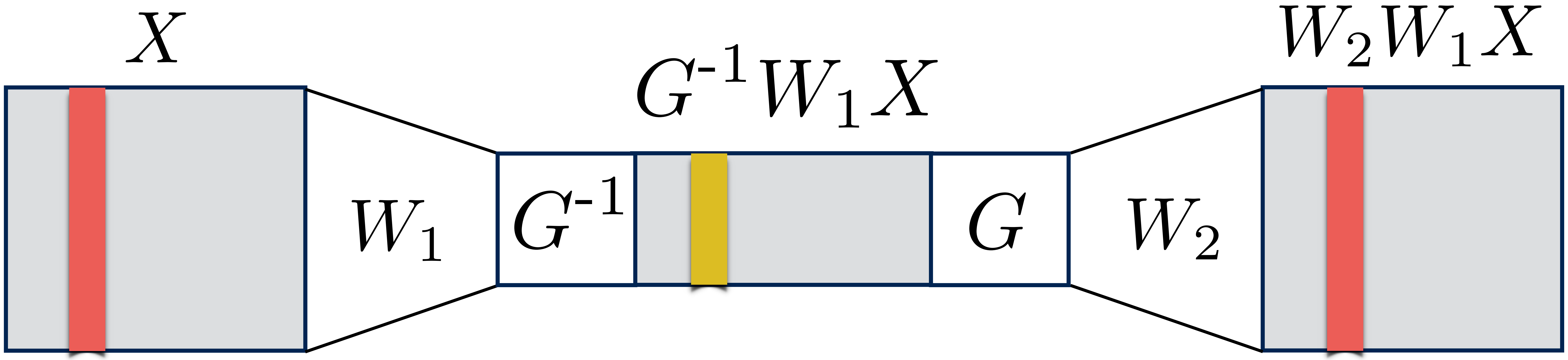
- $\mathcal{L}_X : \mathrm{Gr}_k(\mathbb{R}^m) \to \mathbb{R}$ is Morse iff singular values are positive and distinct.
- $\binom{m}{k}$ critical points are the principal k-planes.
- Critical values are sums of eigenvalues => toy model for random matrix theory.
- Gradient trajectories between principal planes in adjacent index rotate one principal direction in first plane to another in second plane fixing the rest.
- There are exactly two such trajectories $\Longrightarrow$ perfect $\mathbb{F}_2$-Morse function.
- This rabbit hole goes far *deeper*…


Data
Low-rank approx
why
how
what : TSVD
SVD / PCA
pros / cons
NMF
Dylan
generalize Data
memory / speed
interpretability
n x m
n > m
max log likelihood


Models, Inference and
Algorithms Primer • 2019
From Morse theory to geometric
ensembling via the topology of PCA
Jon Bloom & Cotton Seed
BROAD
INSTITUTE
0:02 / 1:45:26


Models, Inference and
Algorithms Meeting • 2019
Regularized linear autoencoders,
probabilistic PCA, & backpropagation
in the brain
Daniel Kunin & Aleksandrina Goeva
BROAD
INSTITUTE
58:00 / 1:45:26

Linear Autoencoder



$$\mathcal{L}(W_1, W_2) = ||X - W_2W_1X||^2 = ||X - (W_2G)(G^{-1}W_1)X||^2$$

$$X = U\Sigma V^\top \qquad W_1 = G^{-1}U_k^\top \qquad W_2W_1 = U_kU_k^\top$$

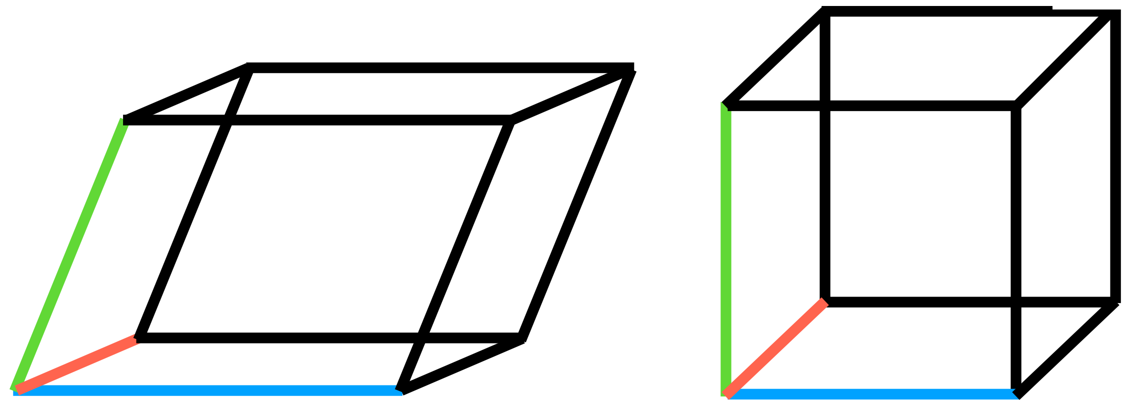
$$W_2 = U_kG$$

Fact: Linear autoencoders are *pseudoinverses* at all critical points: $W_2 = W_1^+$

Regularization  Orthogonality

- Orthogonal matrices are the volume-preserving matrices of minimal Frobenius norm.

$$\sum \sigma_i^2 \qquad \min_{A} ||A||_F^2 \quad \text{s.t.} \quad \det(A)^2 = 1 \qquad \prod \sigma_i^2$$

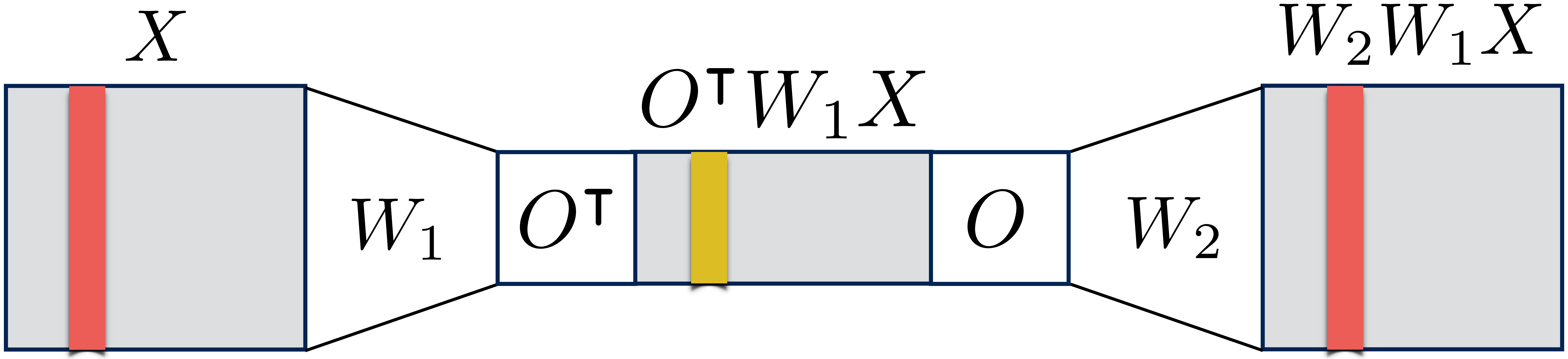


- Orthogonal matrices are the inverse matrices of minimum total Frobenius norm.

$$\sum(\sigma_i^2 + \sigma_i^{-2}) \quad \min_{A,B} ||A||_F^2 + ||B||_F^2 \quad \text{s.t.} \quad AB = I$$

In particular, $A = B^\intercal$.

Linear Autoencoder



$$\mathcal{L}_\sigma(W_1, W_2) = ||X - W_2 W_1 X||^2 + \lambda(||W_1||^2 + ||W_2||^2)$$

$$X = U\Sigma V^\intercal$$

$$W_2 W_1 = U_k (I - \lambda\Sigma^{-2})^{-1} U_k^\intercal$$

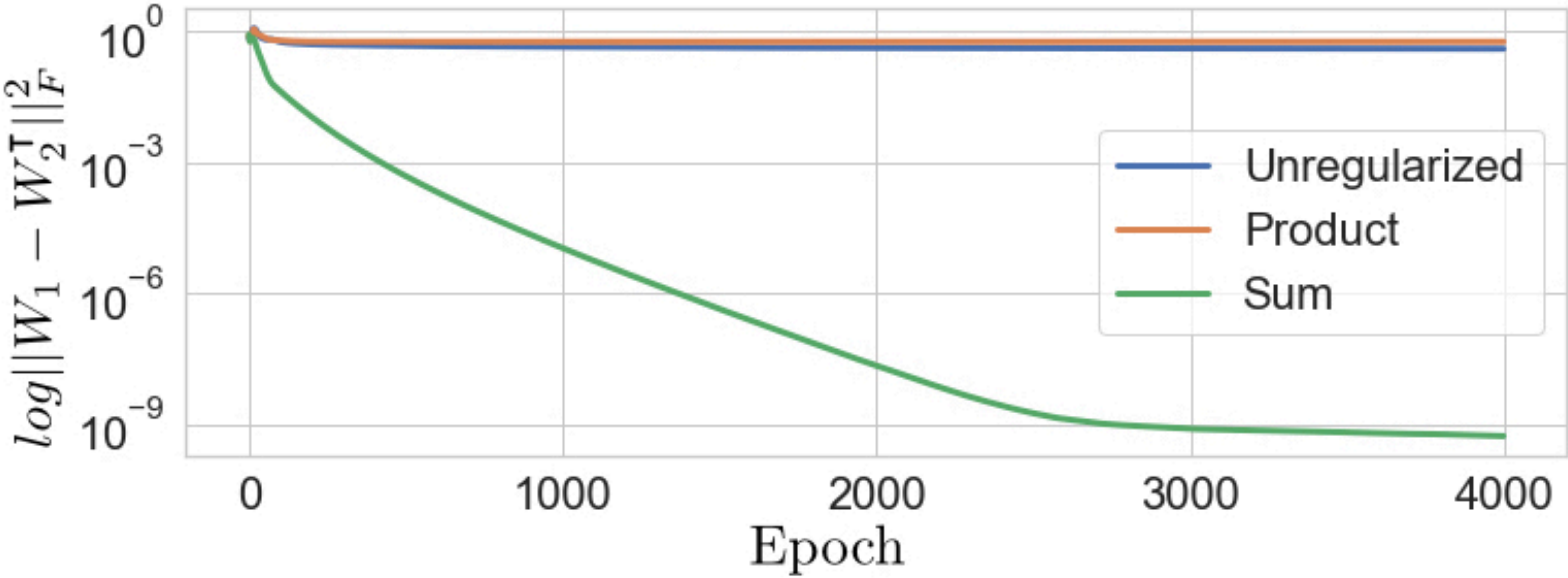
$$W_1 = O^\intercal (I - \lambda\Sigma^{-2})^{-\frac{1}{2}} U_k^\intercal$$

$$W_2 = U_k (I - \lambda\Sigma^{-2})^{-\frac{1}{2}} O$$

Theorem: L_2-regularized linear autoencoders are *symmetric* at all critical points.

$$W_2 = W_1^T$$

Linear Autoencoder



Linear Autoencoder

$$X = U\Sigma V^{\intercal} \implies W_2 = U_k(I - \lambda\Sigma^{-2})^{-\frac{1}{2}}O$$

Theorem: L_2-regularized linear autoencoders are *symmetric* at all critical points.

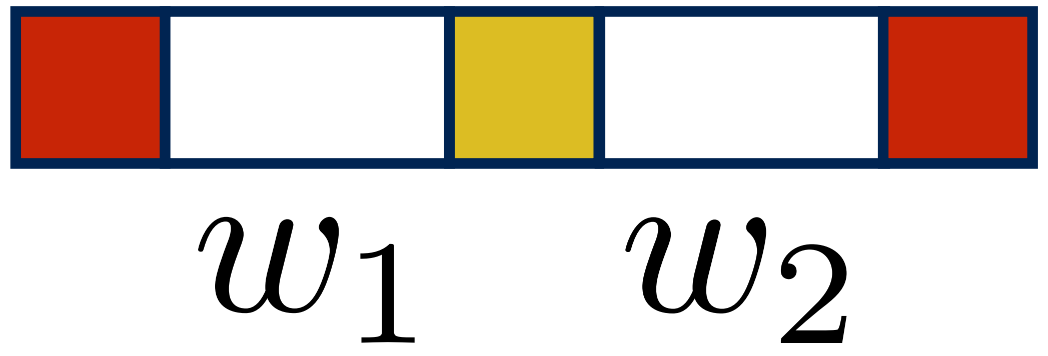
Theorem: The loss is strictly saddle (*Morse-Bott*). All minima are global.

Theorem: The top principal directions of X with eigenvalue greater than λ coincide with the left singular vectors of the trained decoder W_2. These eigenvalues are determined by the singular values of W_2.

PCA algo: Fit a regularized LAE and SVD the decoder.

+ weight decay

Scalar Autoencoder



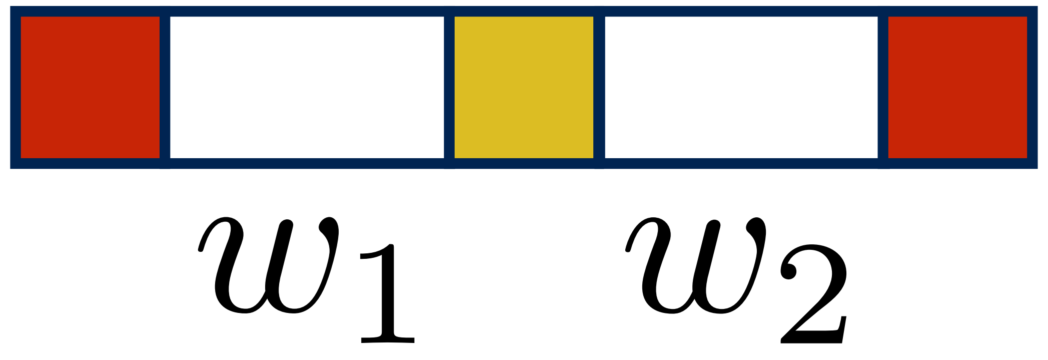
unregularized $$\mathcal{L}(w_1, w_2) = (x - w_2 w_1 x)^2$$

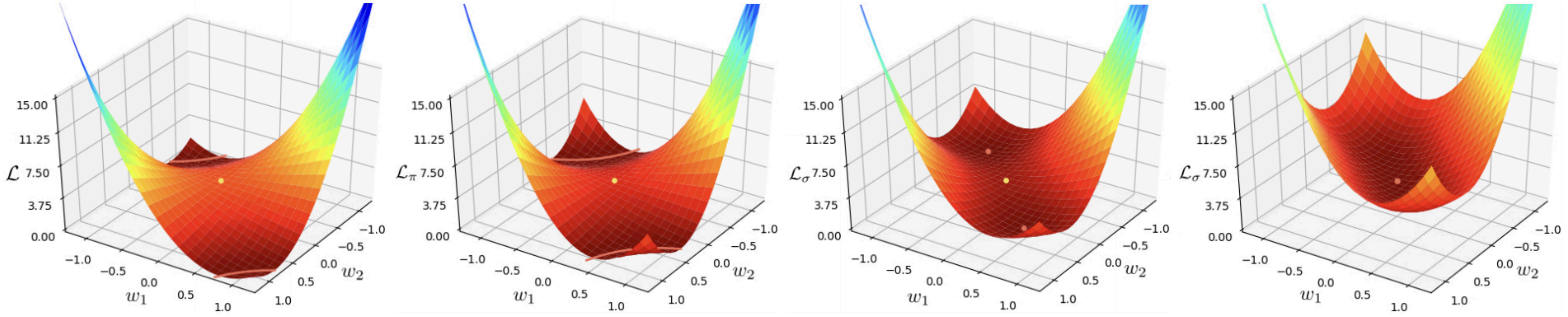
product $$\mathcal{L}_\pi(w_1, w_2) = (x - w_2 w_1 x)^2 + \lambda (w_2 w_1)^2$$

sum $$\mathcal{L}_\sigma(w_1, w_2) = (x - w_2 w_1 x)^2 + \lambda (w_1^2 + w_2^2)$$

[Scalar AE Visualization]

Scalar Autoencoder





(a) Unregularized (b) Product ($\lambda = 2$) (c) Sum ($\lambda = 2$) (d) Sum ($\lambda = 4$)

Figure 1. Scalar loss landscapes with $x^2 = 4$. Yellow points are saddles and red curves and points are global minima.

Theorem 4.2 (Landscape Theorem).

The critical landscape is diffeomorphic to the space of pairs $(\mathcal{I}, G)$ or $(\mathcal{I}, O)$ with

- $\mathcal{I} \subset \{1, \ldots, m\}$ of size $0 \leq l \leq k$,
- $G \in \mathbb{R}^{k \times l}$ with independent columns,
- $O \in \mathbb{R}^{k \times l}$ with orthonormal columns.

	W_2	W_1
$\mathcal{L}$	$U_{\mathcal{I}} G^{+}$	$G U_{\mathcal{I}}^{\intercal}$
$\mathcal{L}_{\pi}$	$U_{\mathcal{I}}(I_{\ell} + \lambda \Sigma_{\mathcal{I}}^{-2})^{-\frac{1}{2}} G^{+}$	$G(I_{\ell} + \lambda \Sigma_{\mathcal{I}}^{-2})^{-\frac{1}{2}} U_{\mathcal{I}}^{\intercal}$
$\mathcal{L}_{\sigma}$	$U_{\mathcal{I}}(I_{\ell} - \lambda \Sigma_{\mathcal{I}}^{-2})^{\frac{1}{2}} O^{\intercal}$	$O(I_{\ell} - \lambda \Sigma_{\mathcal{I}}^{-2})^{\frac{1}{2}} U_{\mathcal{I}}^{\intercal}$

Shrinkage

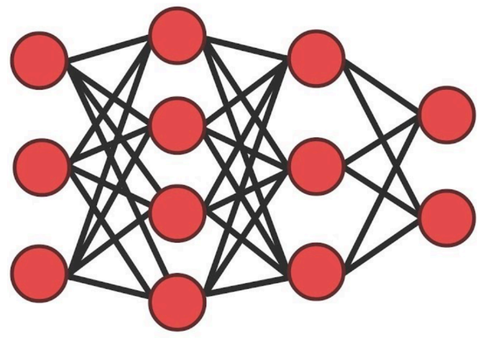
$\lambda = 1$
$\lambda = 9$
$\lambda = 25$
$\lambda = 49$
$\lambda = 81$
$\lambda = 121$
$\lambda = 169$
$\lambda = 225$
$\lambda = 289$
$\lambda = 361$
$\lambda = 441$
None
Ridge
pPCA-like
1.0
0.8
0.6
0.4
0.2
0.0
τ^2
150
200
250
300
350
400
σ^2
Unregularized
Product
Sum

Theorem 3.1 (pPCA Theorem). *With $\sigma^2 = \lambda$, the critical points of*

$$\mathcal{L}_\sigma^0(W_0) = \mathcal{L}_\sigma(W_0^\intercal (XX^\intercal)^{-\frac{1}{2}}, (XX^\intercal)^{-\frac{1}{2}} W_0)$$

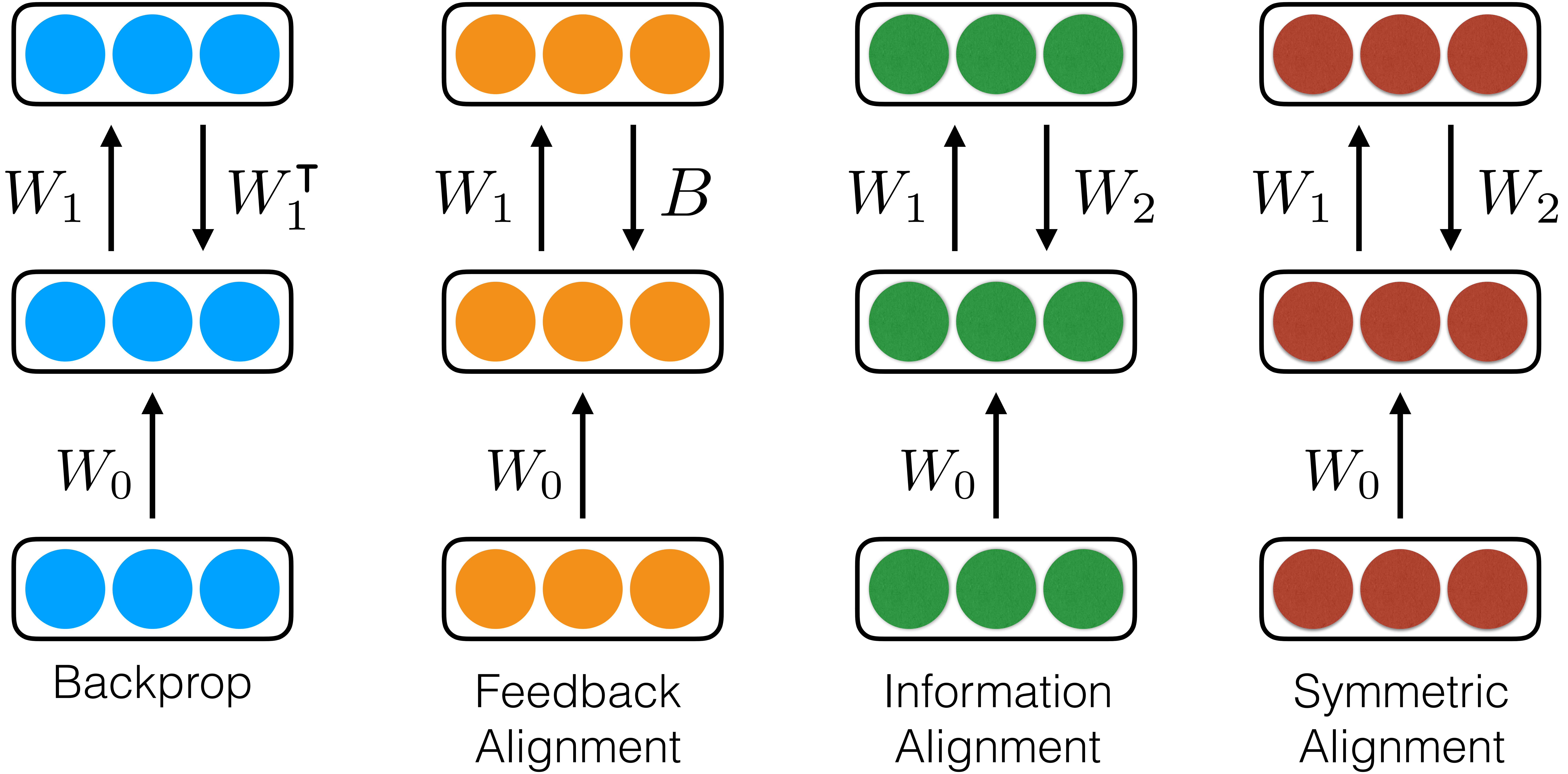
coincide with the critical points of pPCA.

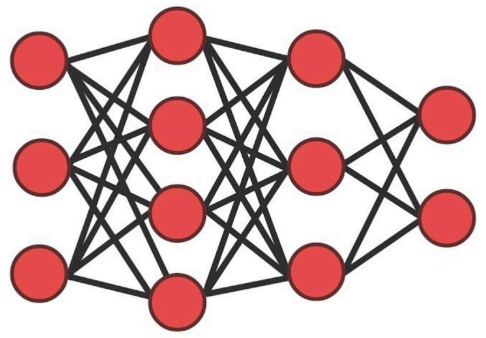
Bayesian $\mathcal{L}_\sigma$	pPCA
$W_1, W_2^\intercal \sim \mathcal{N}_{k\times m}(0, \lambda^{-1})$	$z_i \sim \mathcal{N}_k(0, 1)$
$\varepsilon_i \sim \mathcal{N}_m(0, 1)$	$\epsilon_i \sim \mathcal{N}_m(0, \sigma^2)$
$x_i = W_2 W_1 x_i + \varepsilon_i$	$x_i = W_0 z_i + \epsilon_i$



Prediction in artificial neural networks is inspired by the brain.

Is *learning* in the brain inspired by artificial neural networks?





Prediction in artificial neural networks is inspired by the brain.
Is *learning* in the brain inspired by artificial neural networks?

Prediction
Supervised

$$\mathcal{L}_{\text{pred}} = ||y - W_1 W_0 x||^2$$

Representation
Unsupervised

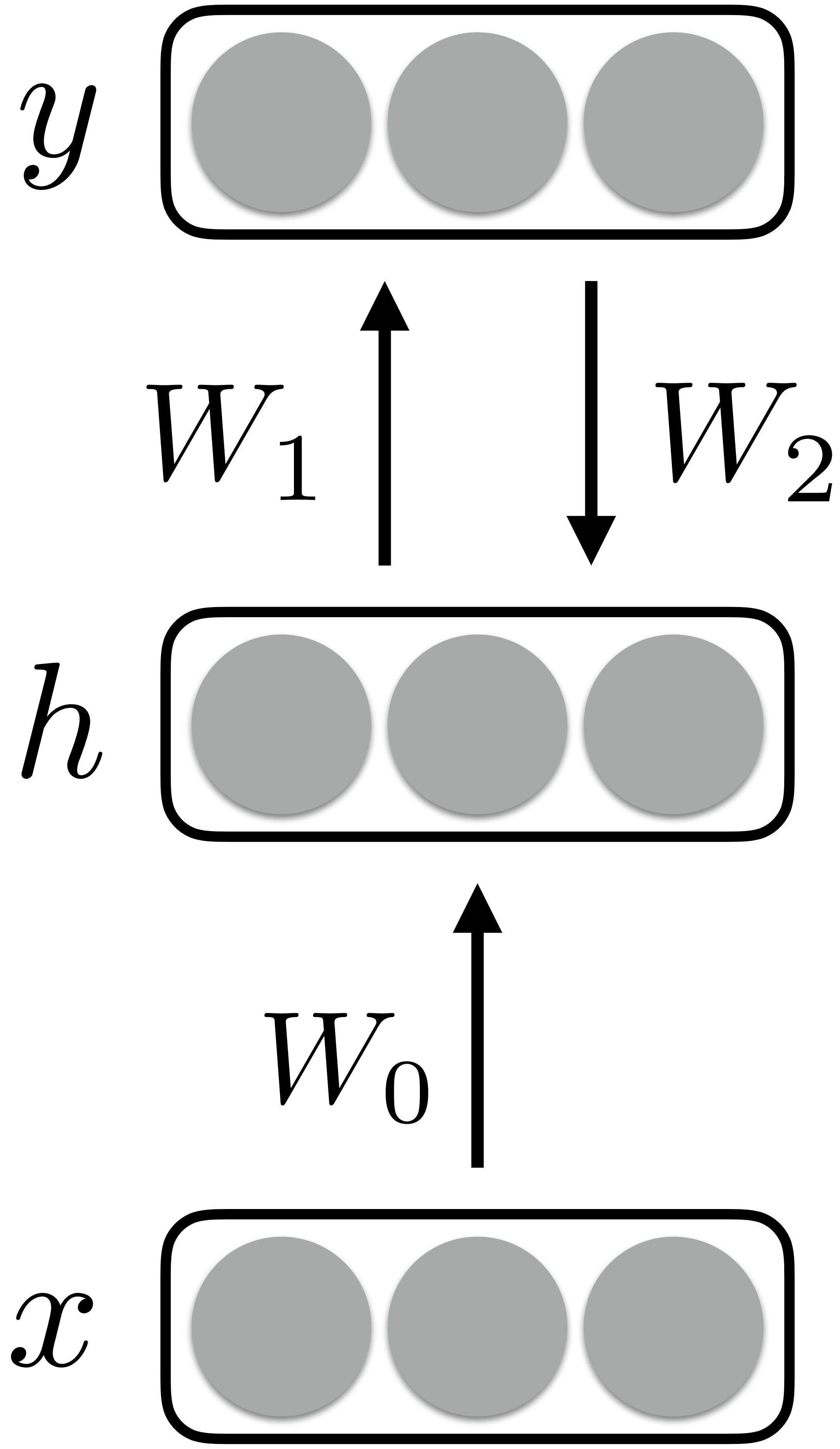
$$\mathcal{L}_{\text{info}} = ||h - W_2 W_1 h||^2$$

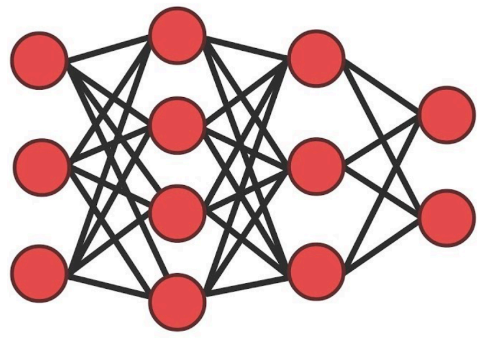
Efficiency
Sparsity

$$\mathcal{L}_{\text{reg}} = ||W_1||^2 + ||W_2||^2$$

Self-amplification
Feedback control

$$\mathcal{L}_{\text{self}} = -2\text{tr}(W_2 W_1)$$





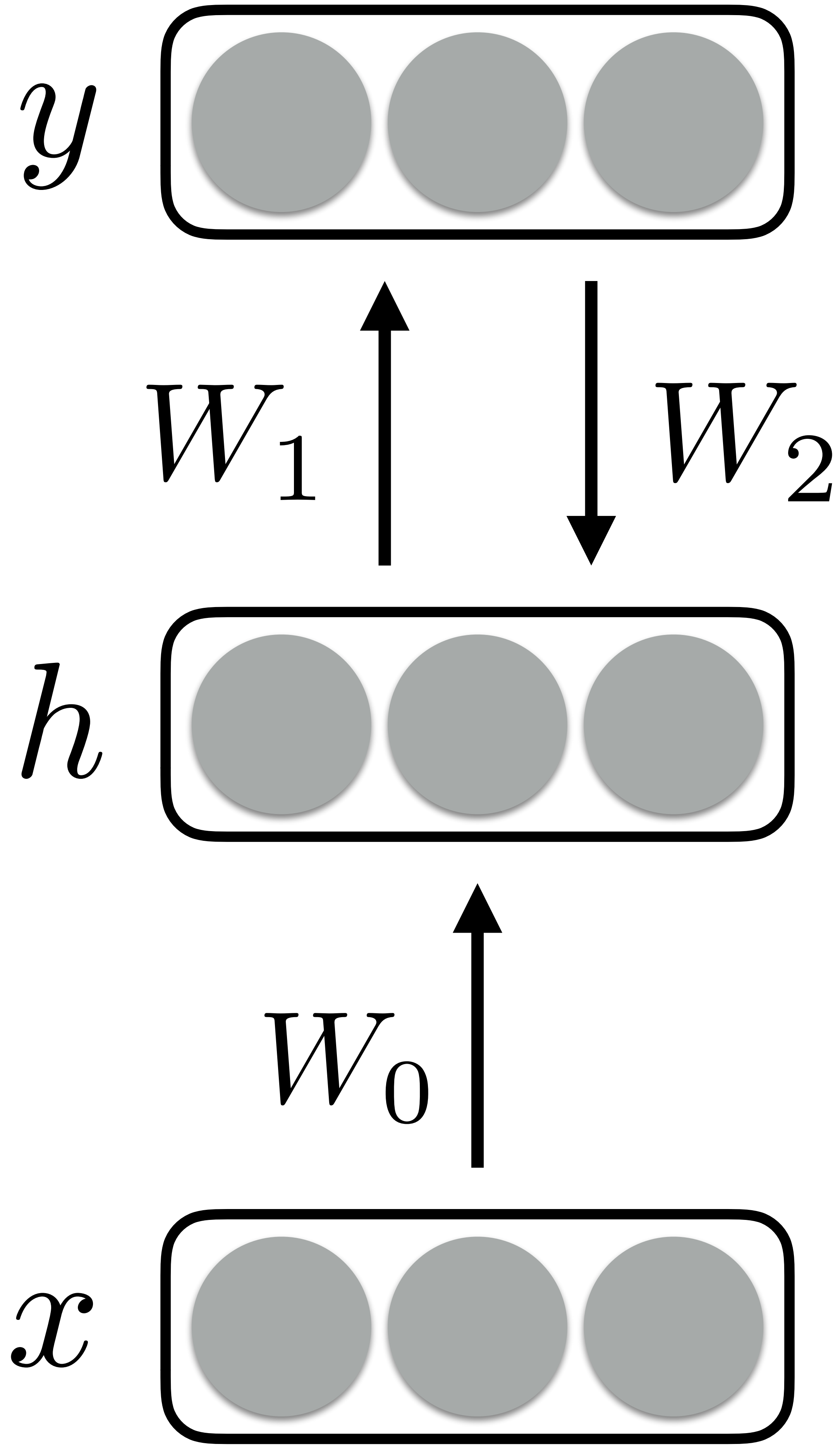
Prediction in artificial neural networks is inspired by the brain.

Is *learning* in the brain inspired by artificial neural networks?

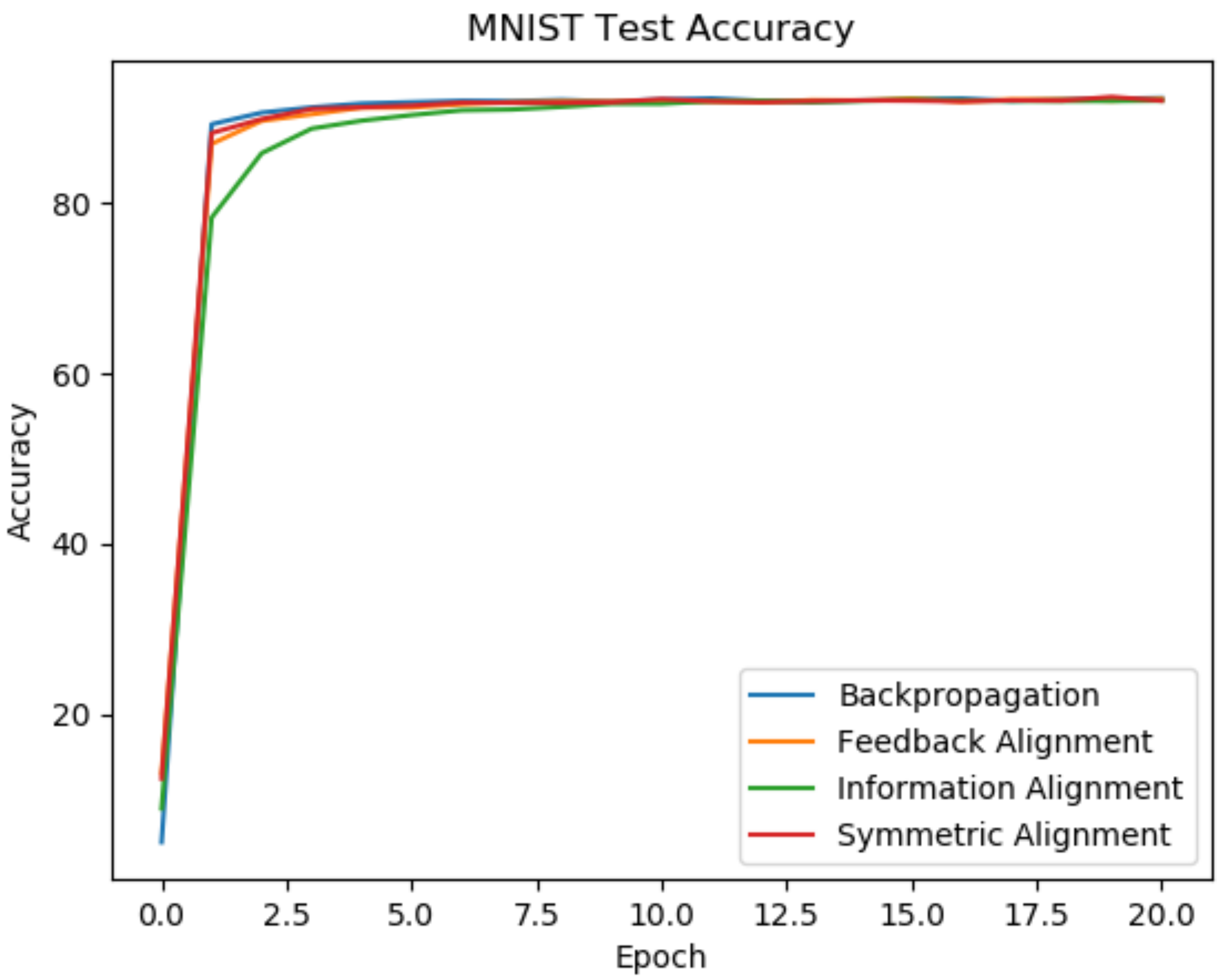
$$\mathcal{L}_{\text{BP}} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{reg}}$$

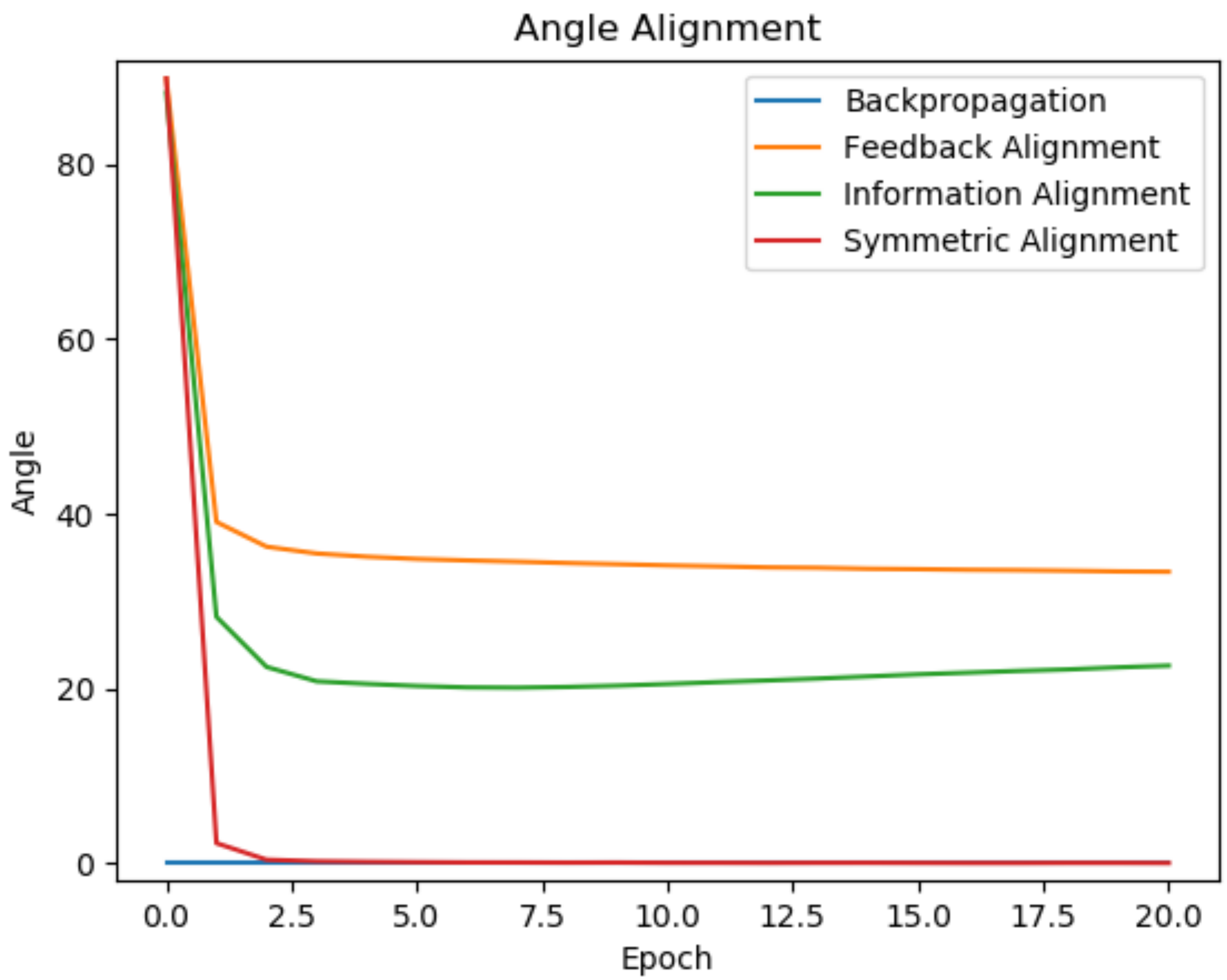
$$\mathcal{L}_{\text{IA}} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{info}} + \mathcal{L}_{\text{reg}}$$

$$\mathcal{L}_{\text{SA}} = \mathcal{L}_{\text{pred}} + \underbrace{\mathcal{L}_{\text{self}} + \mathcal{L}_{\text{reg}}}_{||W_2 - W_1^{\intercal}||^2}$$

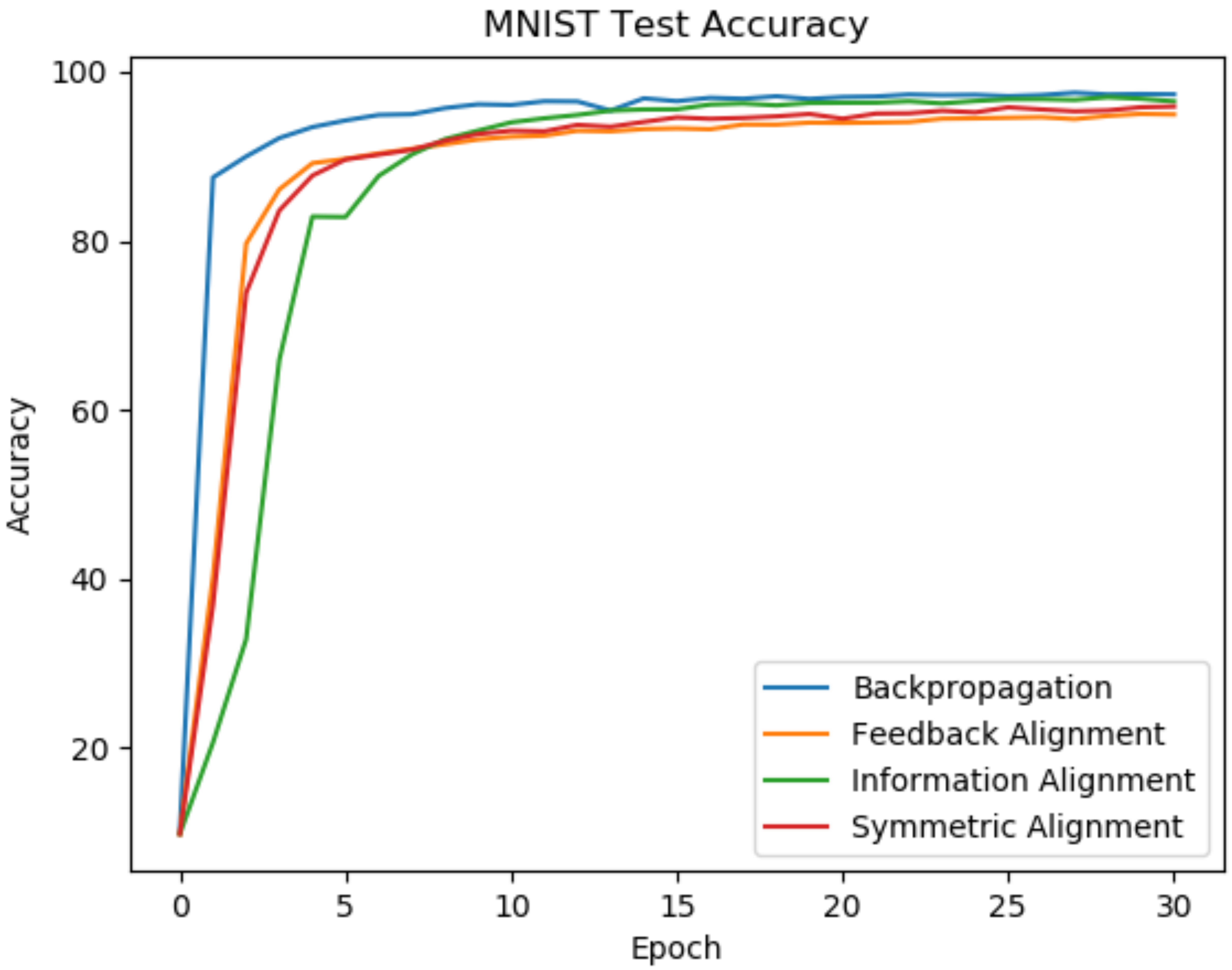


Linear, 1 hidden layer

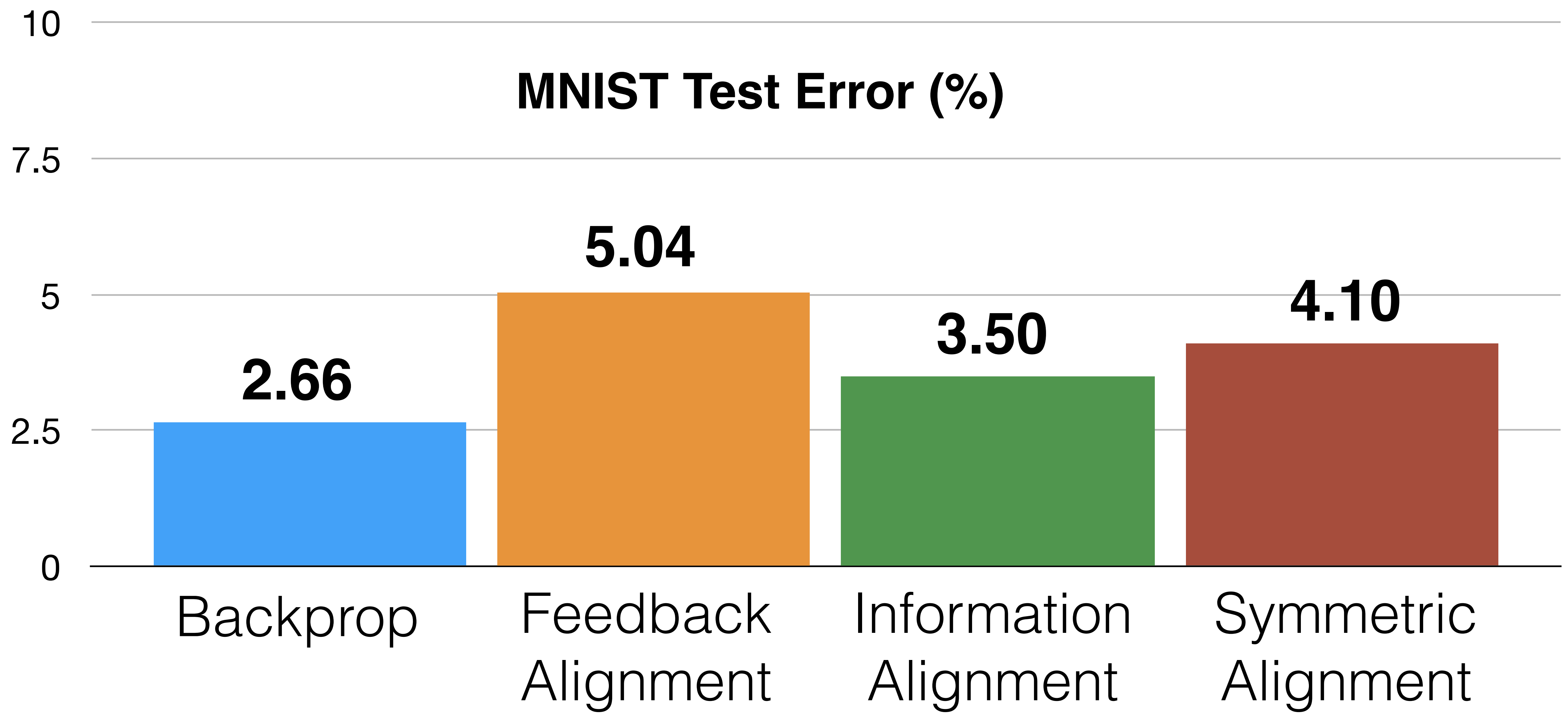




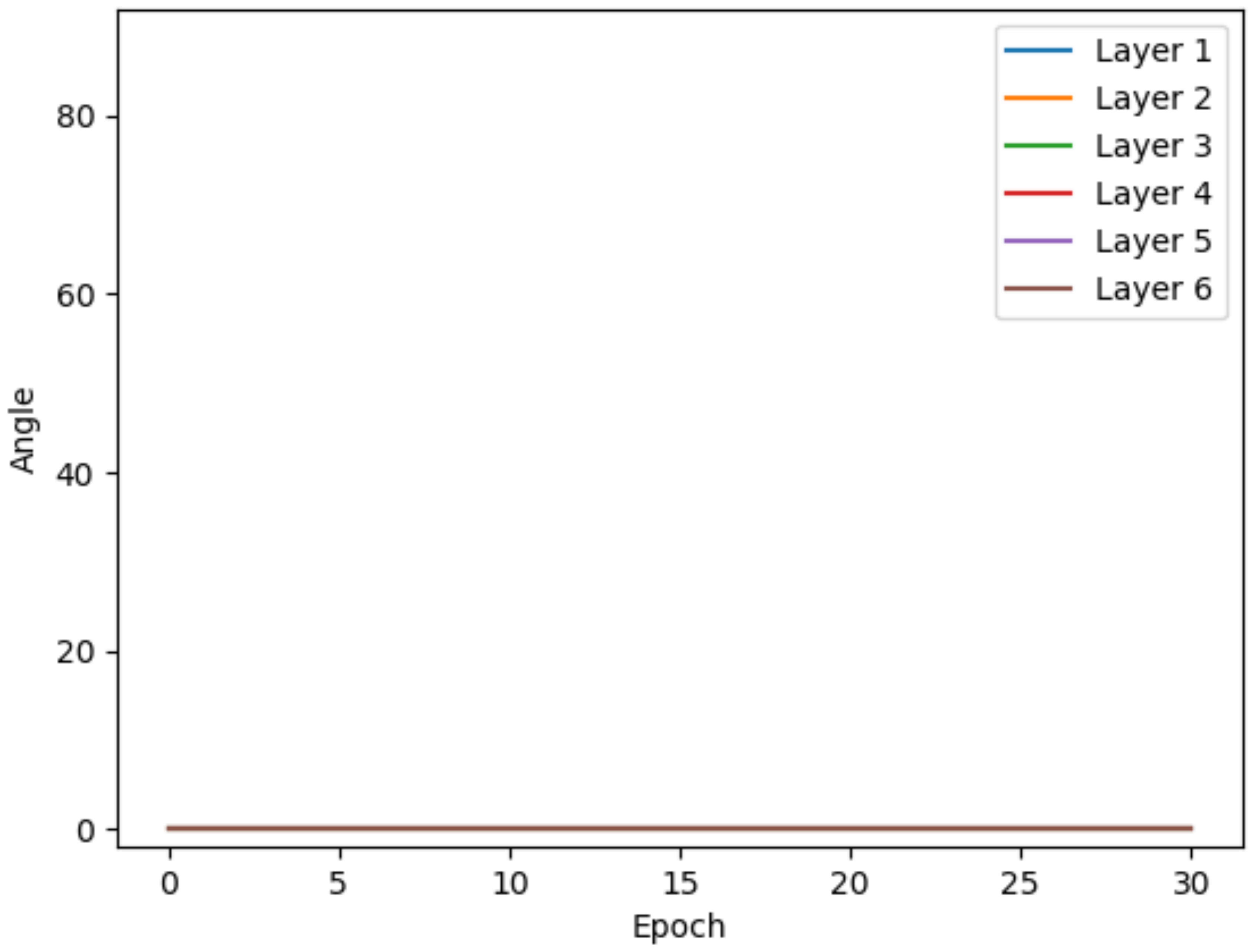
ReLU, 6 hidden layers

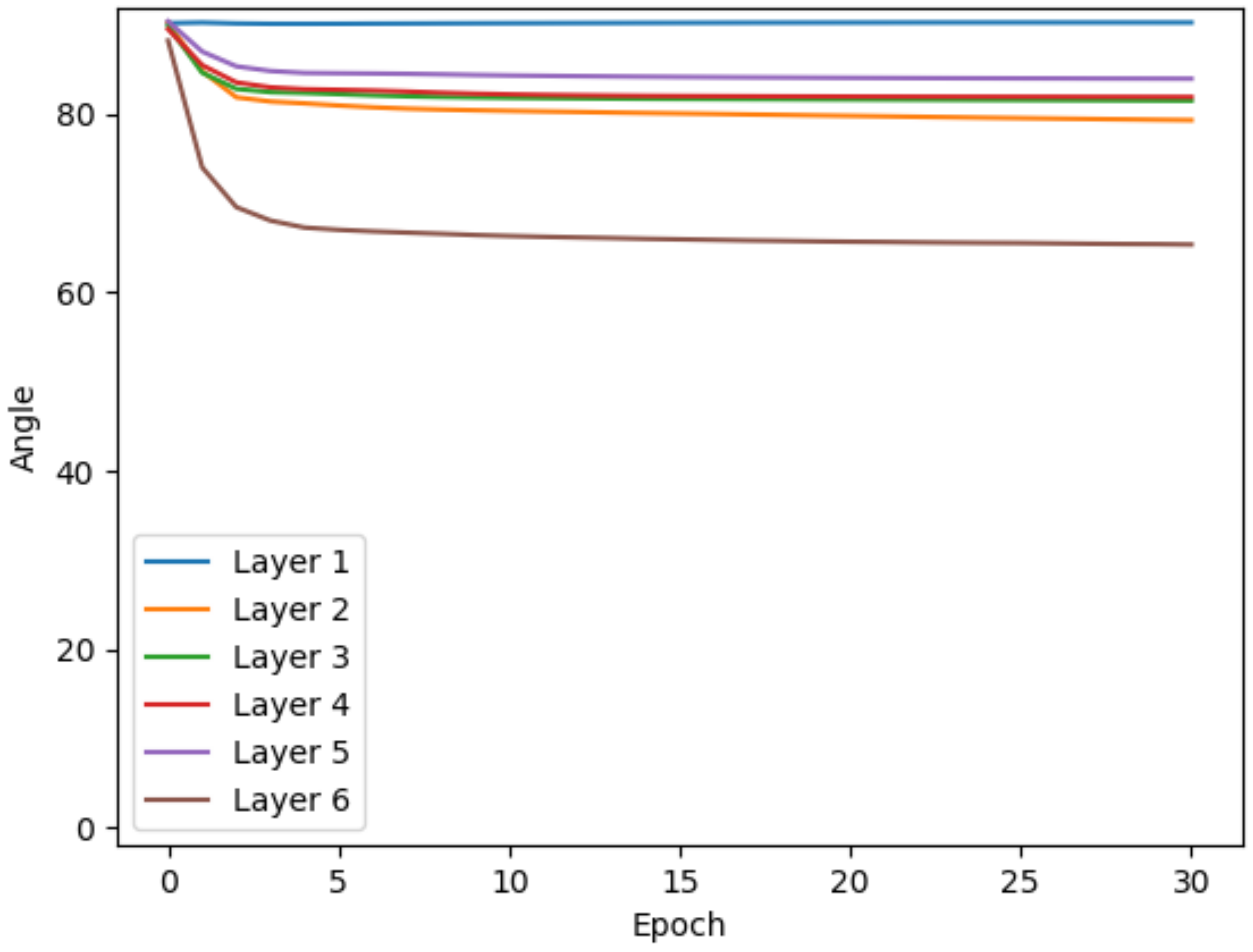


- 5e-5 learning rate
- 1000 batch size
- 30 epochs
- ReLu activations
- 784 - 256 - 256 - 256 - 256 - 256 - 10 architecture
- Lambda = 1e-6



BP

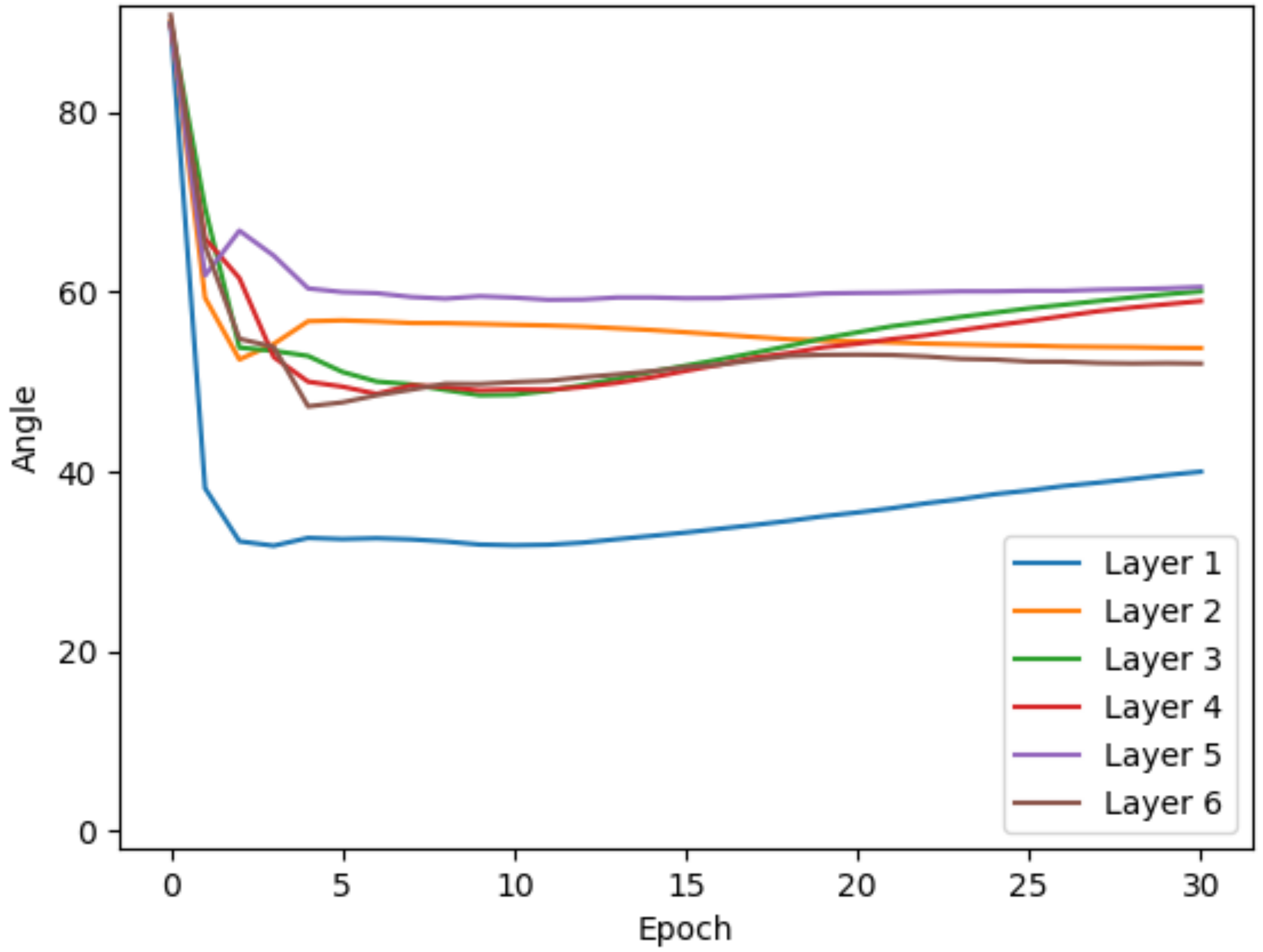


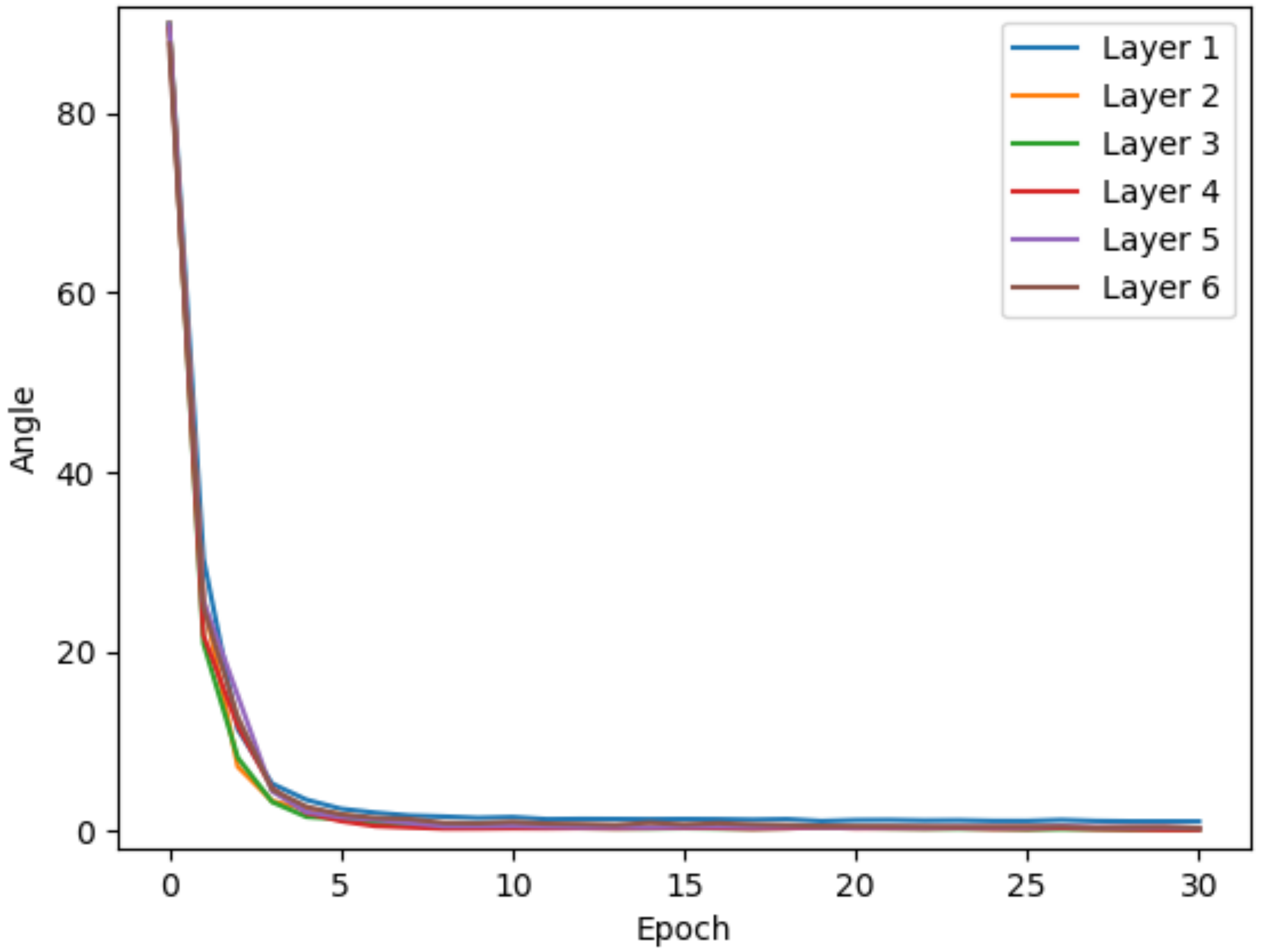


FA

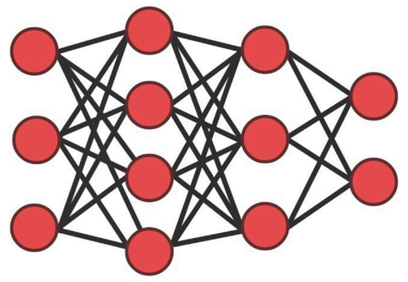
Alignment

IA





SA



in silico

- architecture search for less-biologically-implausible learning at scale
- adversarial robustness, information bottleneck, multitask learning

π

in puro

- LAE from X to Y: closely related to PLS, CCA
- derive and study continuous flows on linear alignment networks

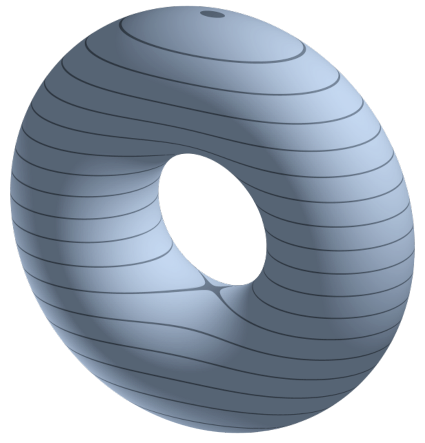
Σ

in algo

- bio-inspired take down of randomized SVD

in vivo

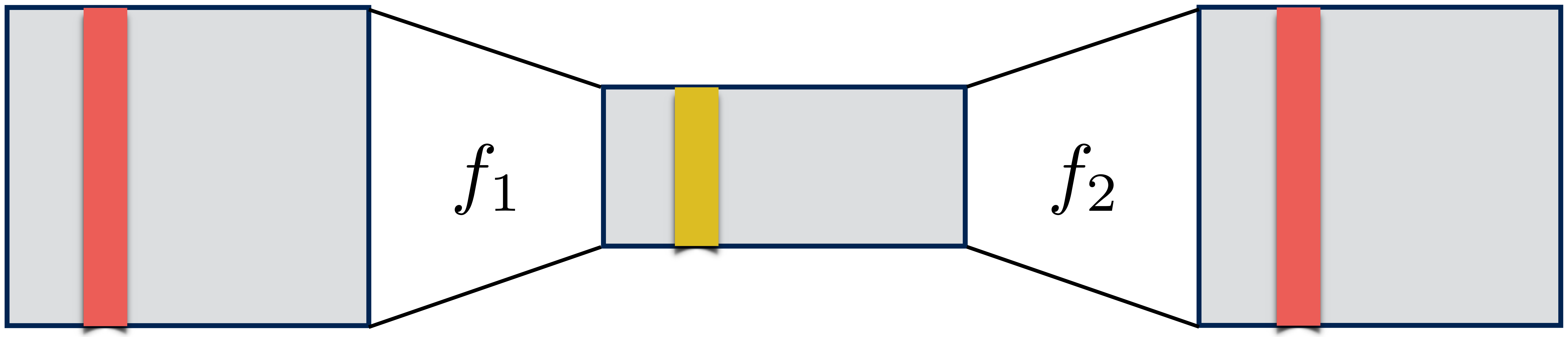
- dynamics of representation teleportation in development and learning
- neural implementation factored through genomic / molecular ontogeny in space and time



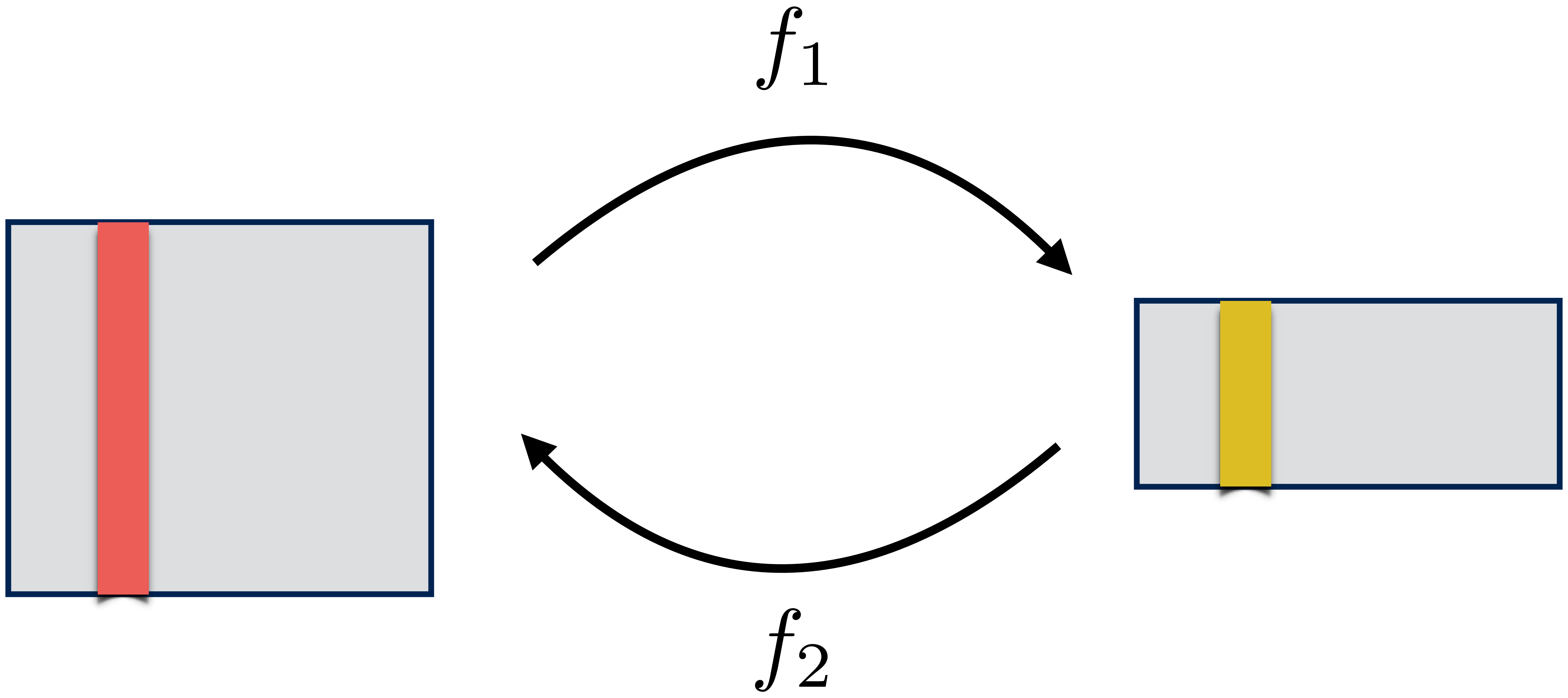
in categorico / physico

- *ground learning in functors between algebra, topology, and geometry (Morse homology, ensemble and consensus learning, TQFT)*

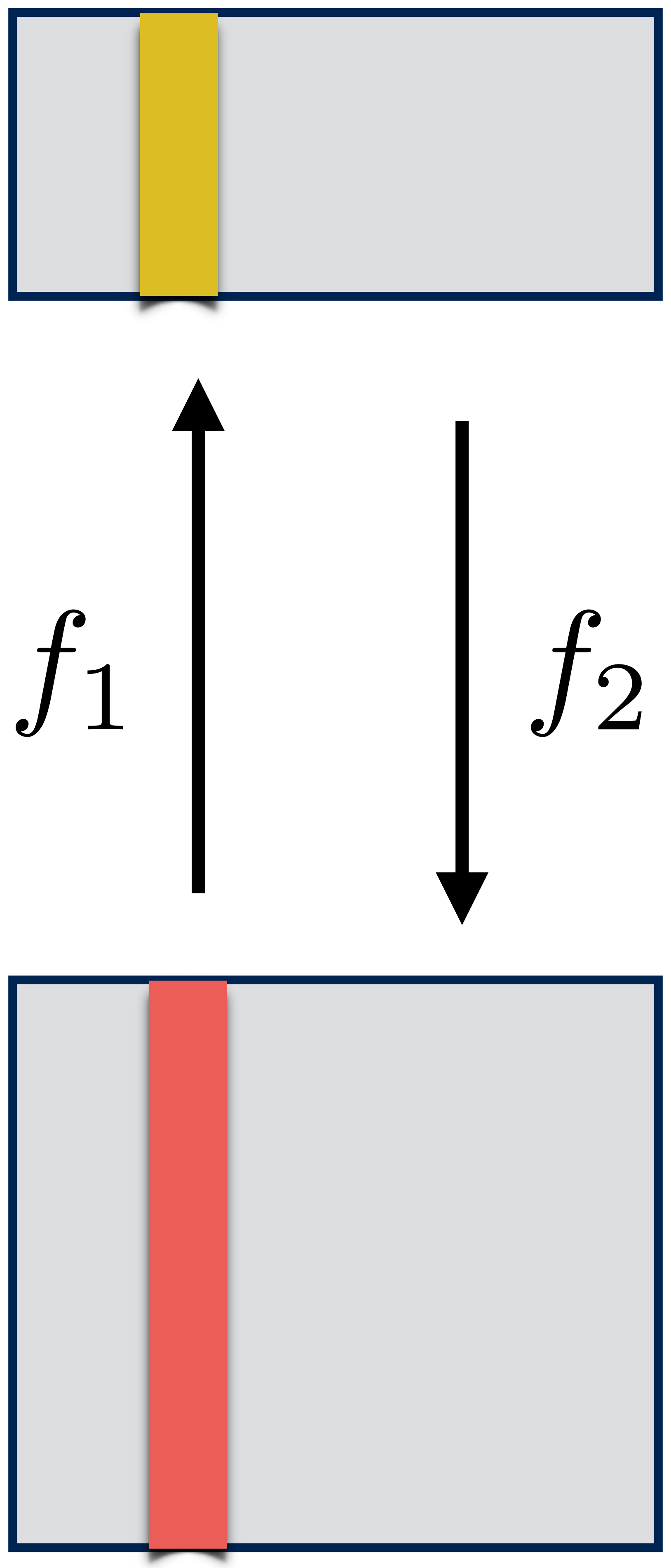
Compressor



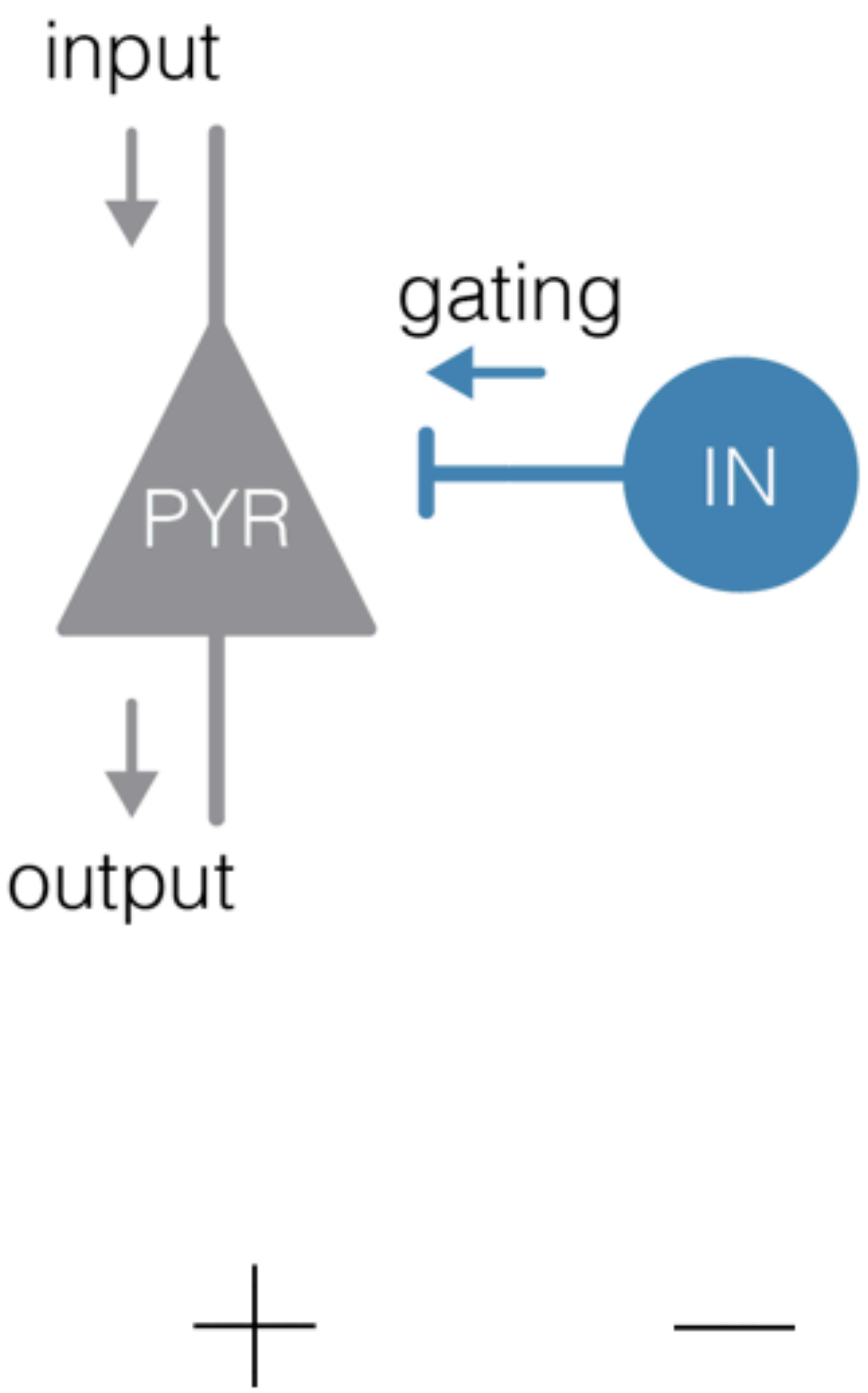
Transporter



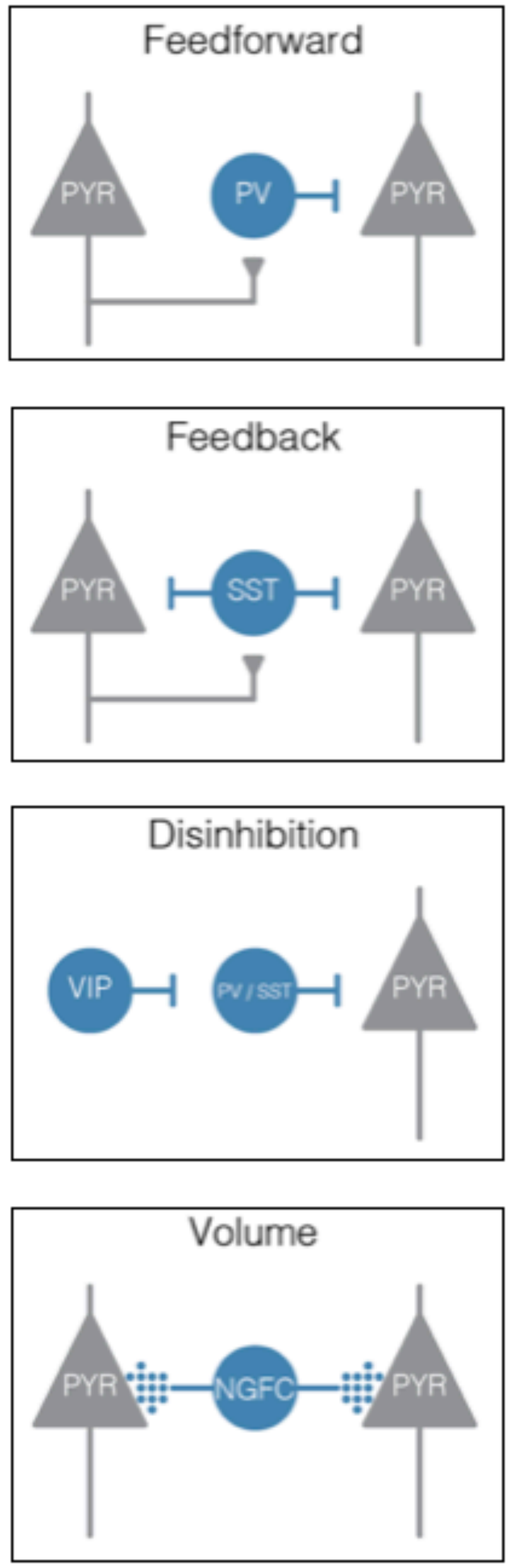
Stabilizer



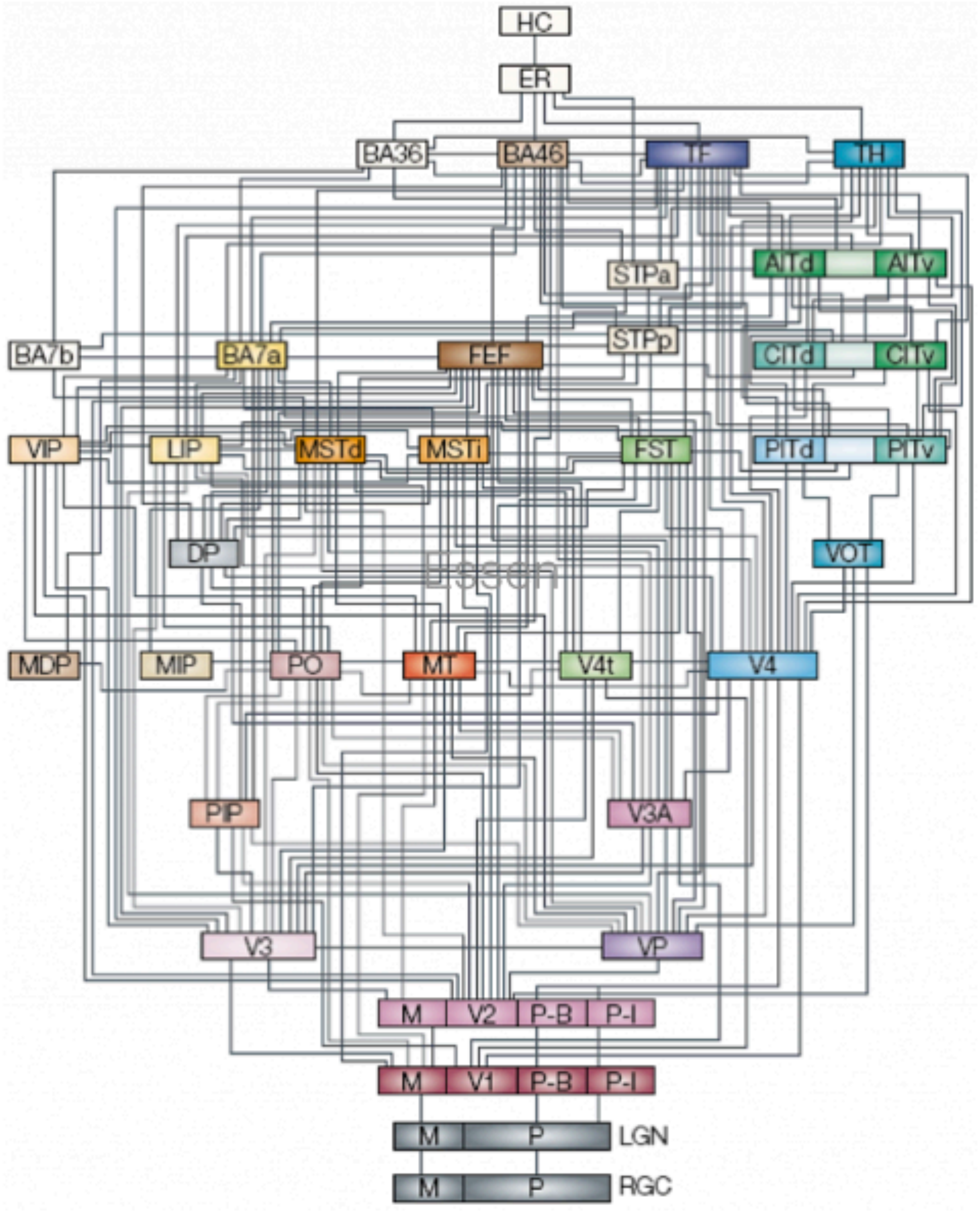
Cellular interactions



Circuit motifs



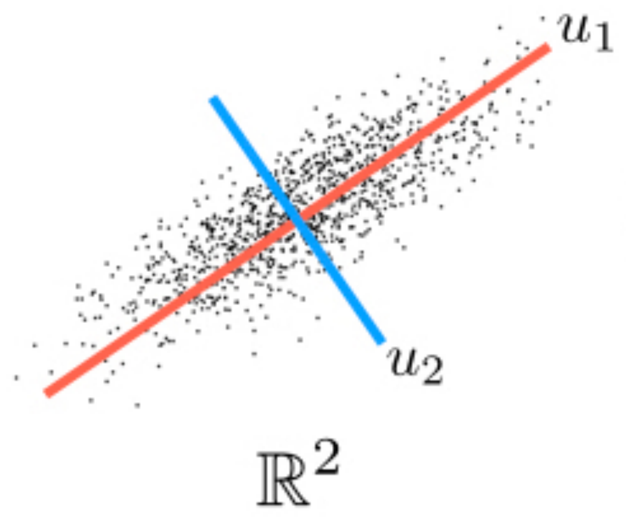
Connectivity across regions



Felleman and Van Essen, Cer. Cor. 1991


MAY THE
MORSE
BE WITH YOU

Thanks!



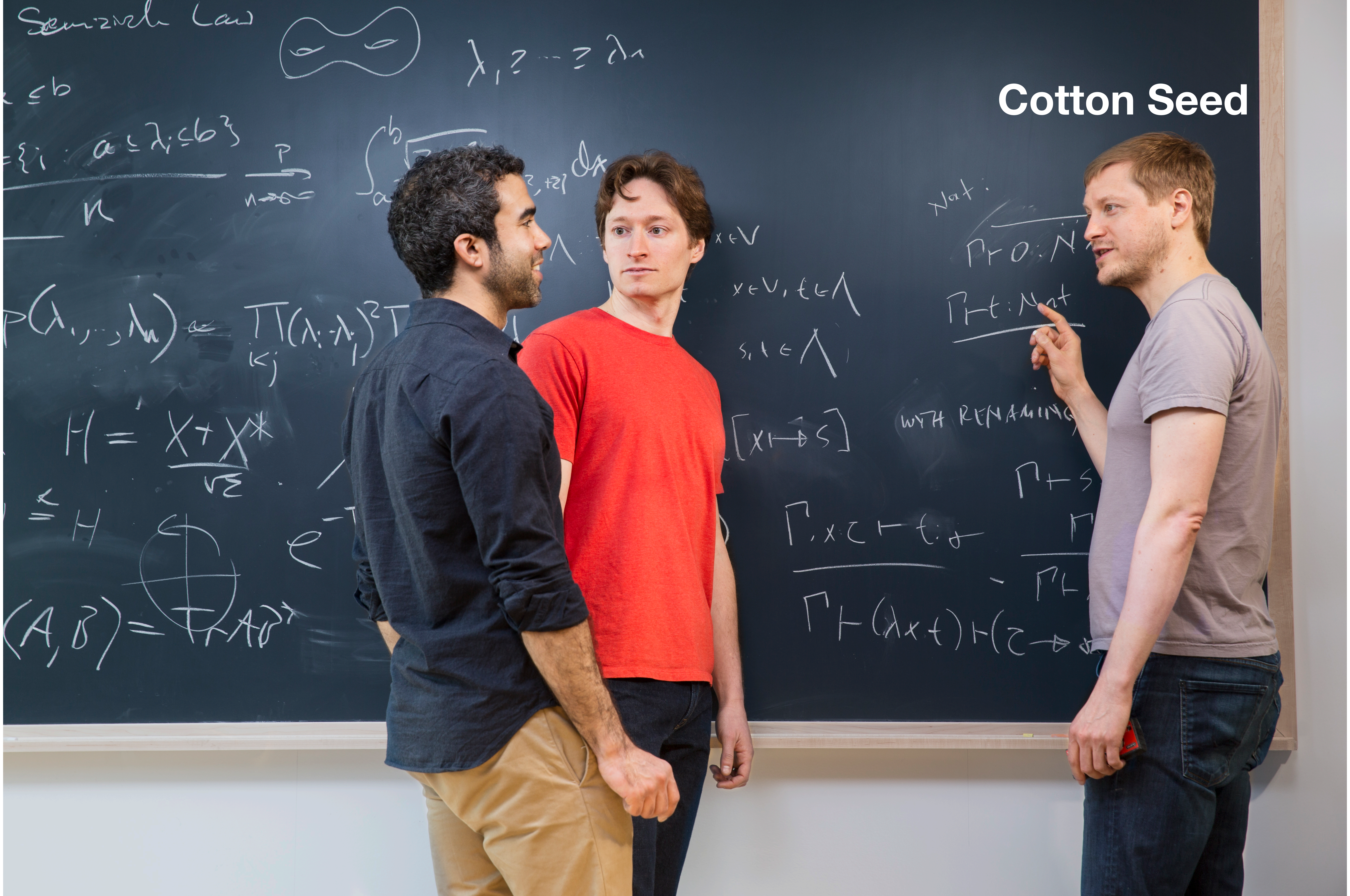
LAE: github.com/danielkunin/Regularized-Linear-Autoencoders

homepage: broadinstitute.org/mia
playlist: bit.ly/2I18EvO
overview: youtube.com/watch?v=gWcFJiYZNZ0

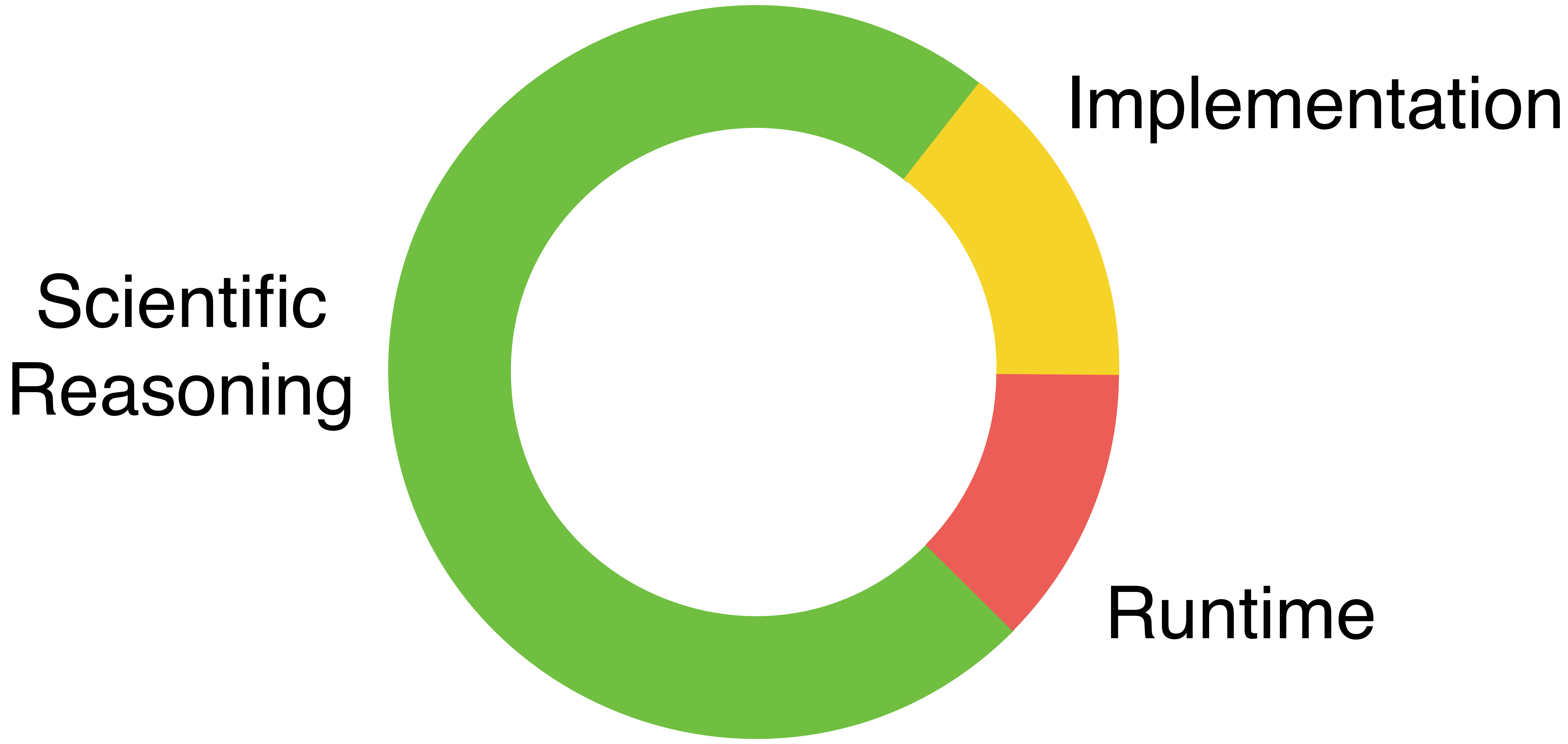
homepage: hail.is/about.html
code: github.com/hail-is/hail

article: thecrimson.com/article/2019/2/28/broad-institute-scrut/

Hail Slides


Cotton Seed


hail


Implementation
Scientific
Reasoning
Runtime

Statistical Genetics Tools

Custom Python/R scripts
- Filter genotypes with bad allele balance
- Call *de novo* variants
- Compute transmission disequilibrium
- Dominance-encoded GWAS
- Gene count permutation tests

Doesn't Scale

PLINK
- Detect sample duplicates or ID swaps
- Call Mendelian violations
- Relatedness
- GWAS
- …

Doesn't Scale

SNPSift
- Genotype concordan

Doesn't Scale

EMMAX
- Sequence kernel association test
- Rare variant burden t

Scales ish

bcftools
- Split multiallelic variants
- Filter on GQ, AD, PASS

Doesn't Scale

tabix
- Subset VCFs to interv

Doesn't Scale

vcffilterjdk
- Filter variants

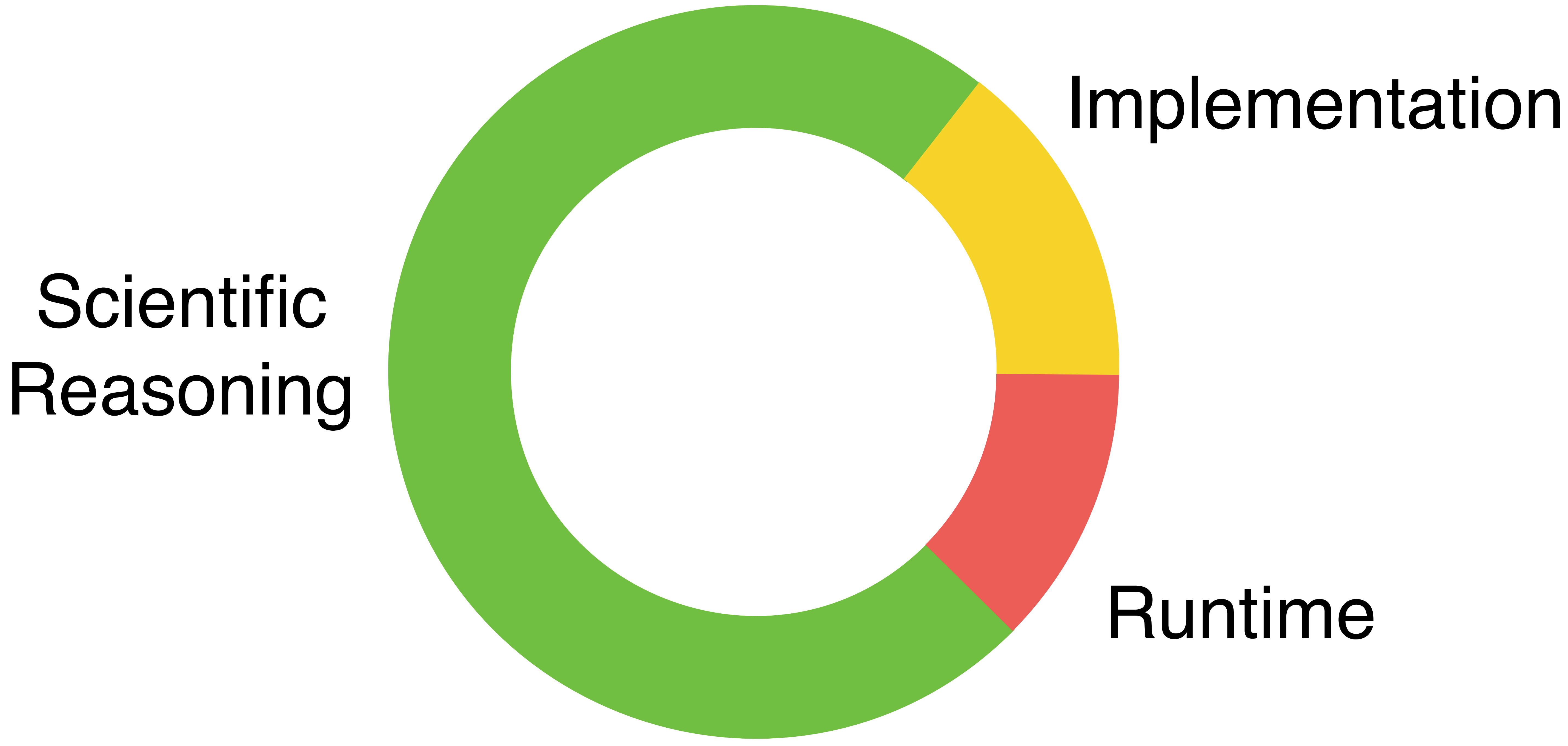
Doesn't Scale

bedtools
- Interval annotation

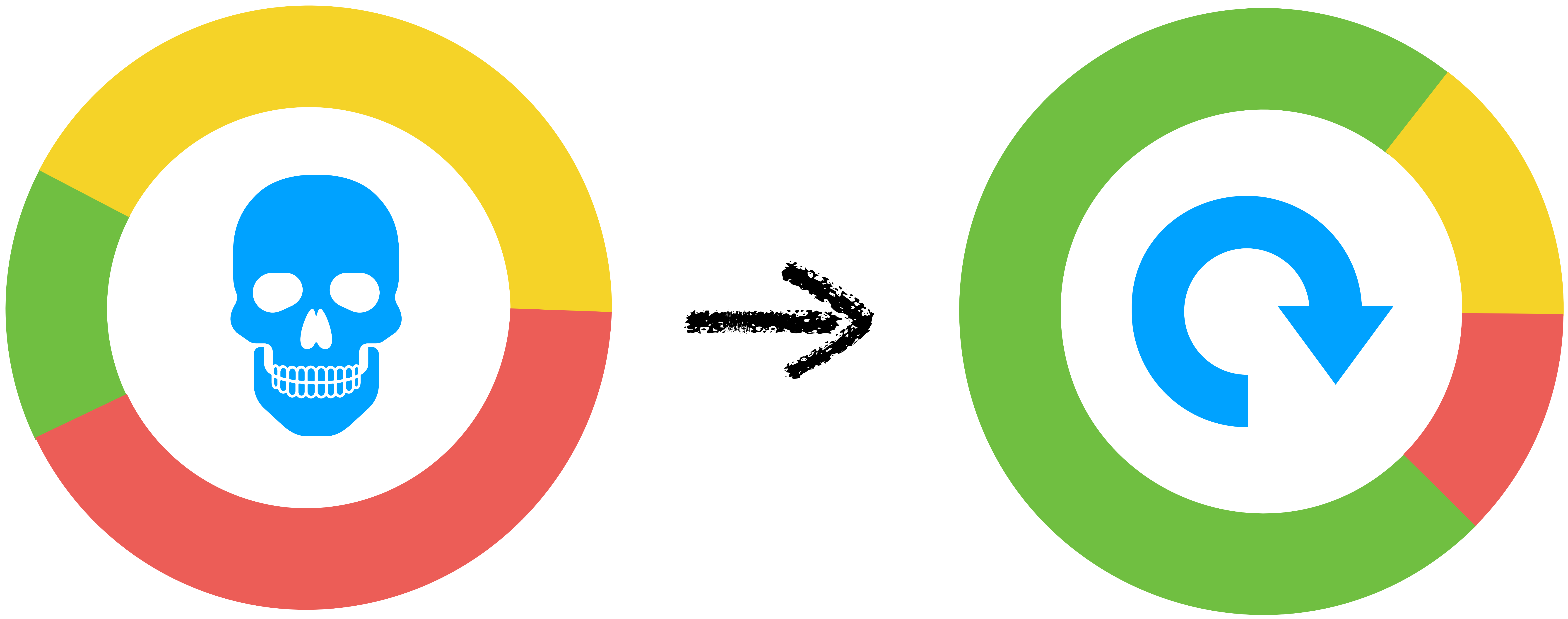
Doesn't Scale

Eigenstrat
- PCA

Doesn't Scale


Implementation
Scientific
Reasoning
Runtime


Implementation
Scientific
Reasoning
Runtime

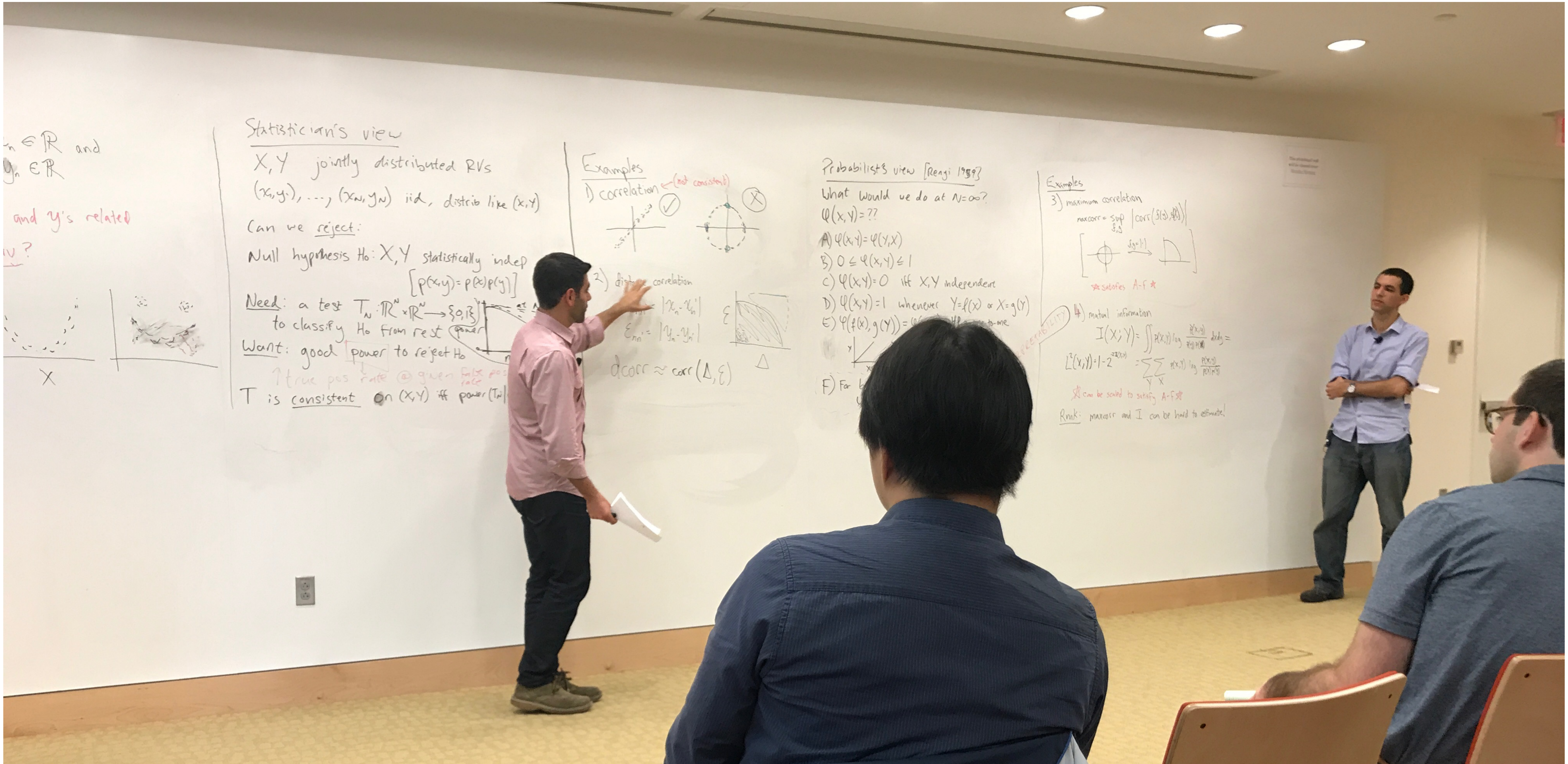


MIA Slides


Alex Bloemendal

MODELS, INFERENCE & ALGORITHMS

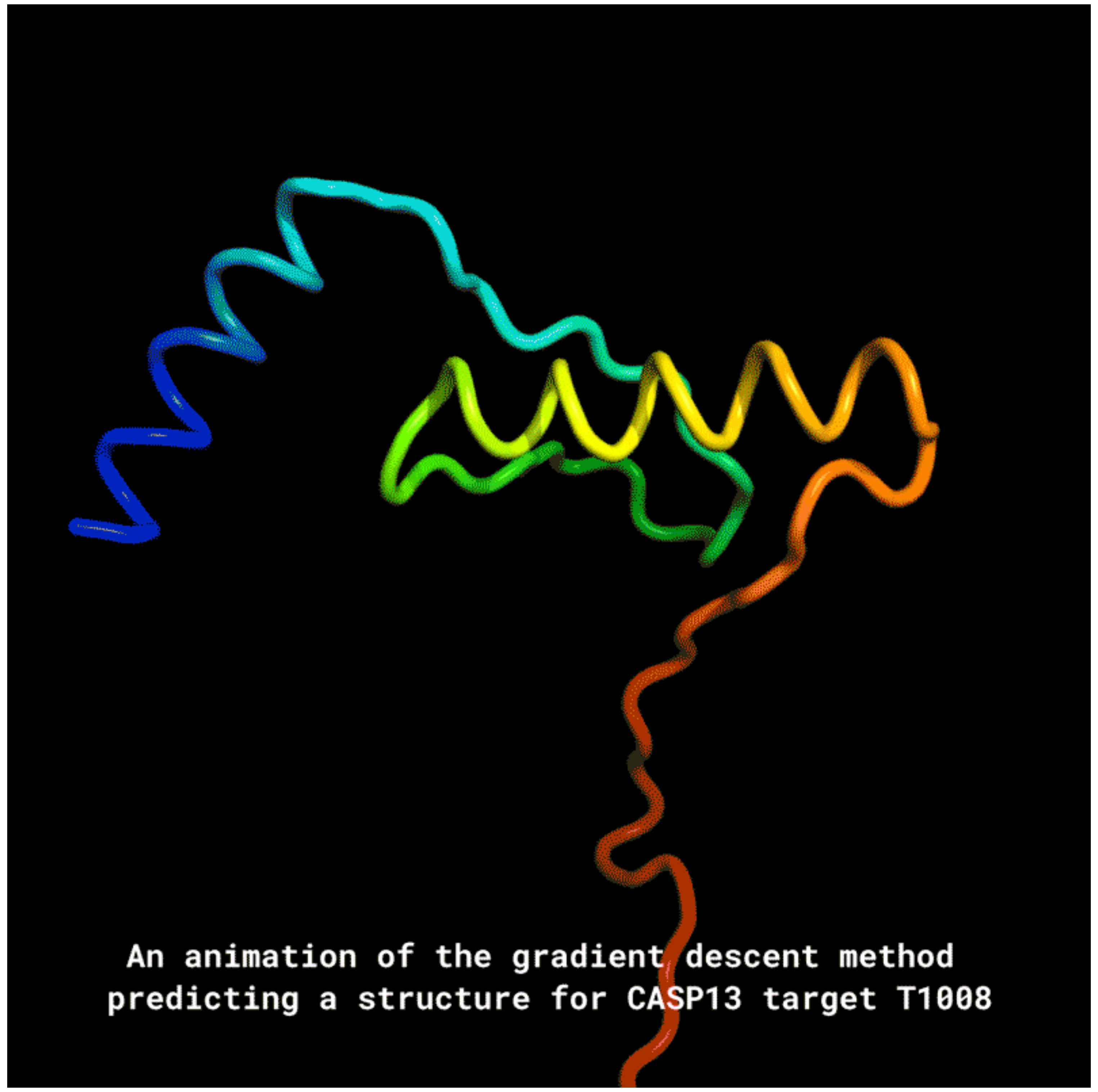
broadinstitute.org/

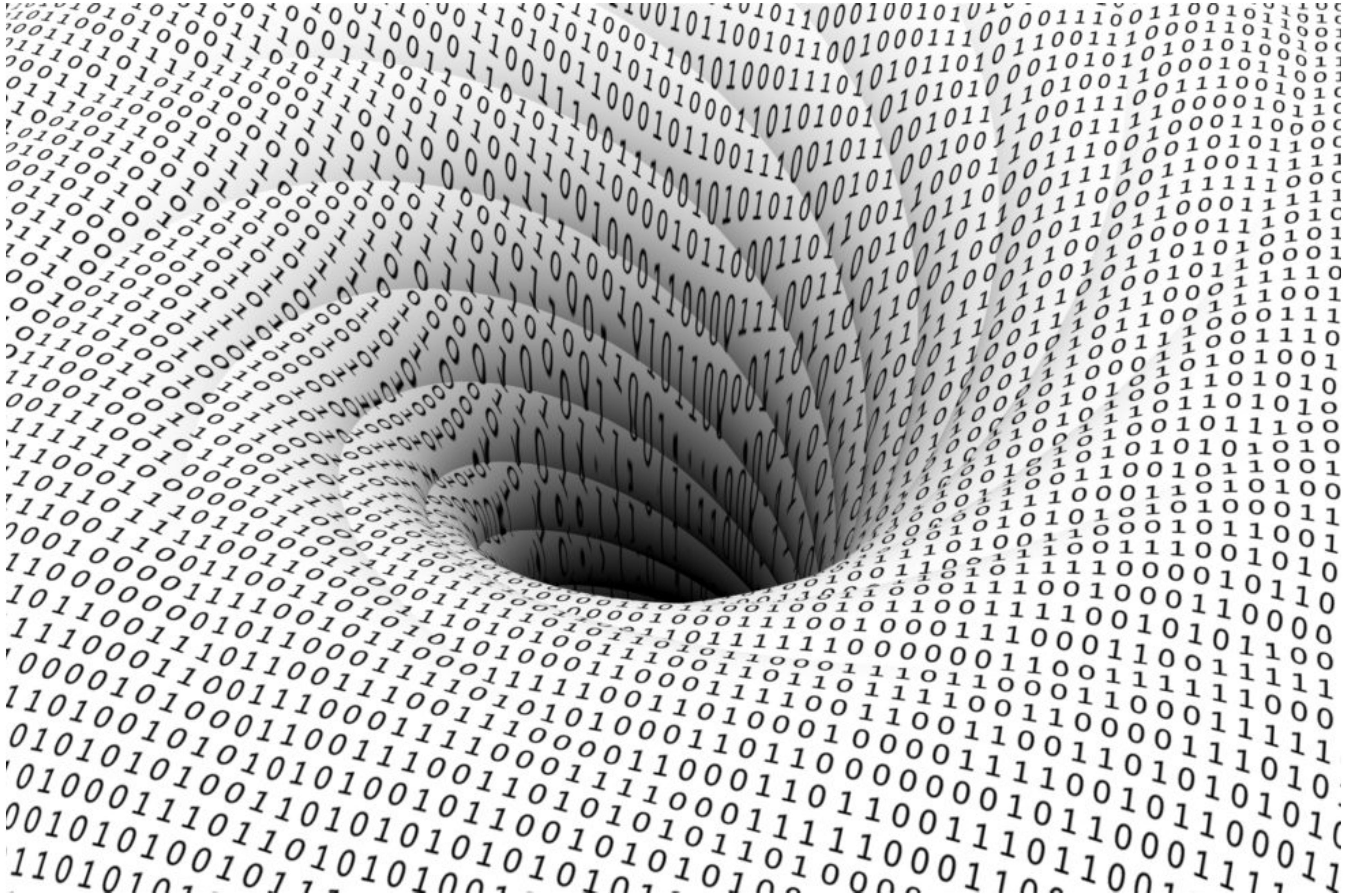


DeepMind

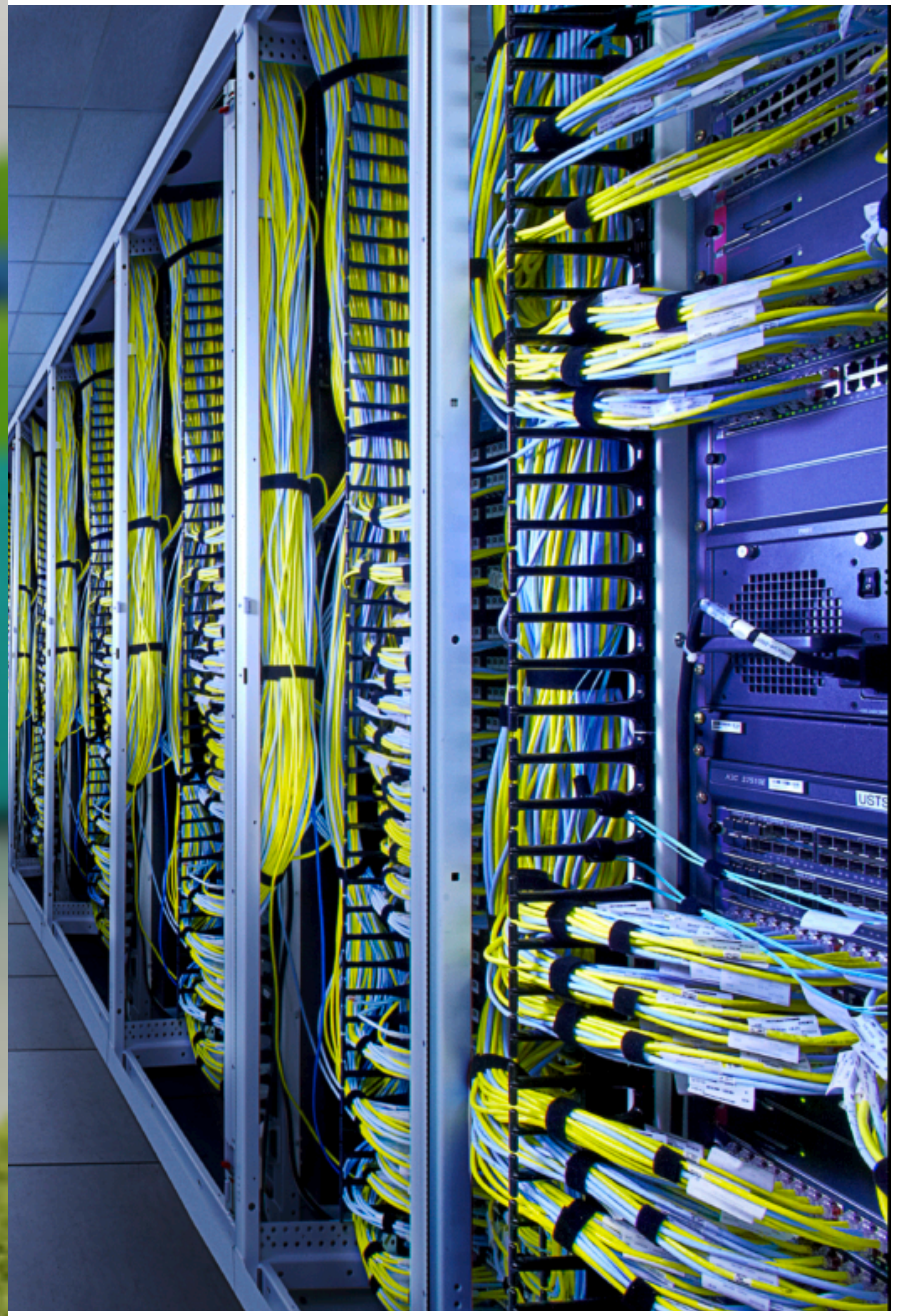

INNOVATIONS IN
The microbiome
nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
LEARNING CURVE
Self-taught AI software attains human-level performance in video games
PAGES 486 & 529
EPIDEMIOLOGY
SHARE DATA IN OUTBREAKS
Forge open access to sequences and more
PAGE 477
COSMOLOGY
A GIANT IN THE EARLY UNIVERSE
A supermassive black hole at a redshift of 6.3
PAGES 490 & 512
QUANTUM PHYSICS
TELEPORTATION FOR TWO
Transferring two properties of a single photon
PAGES 491 & 516
NATURE.COM/NATURE
26 February 2015 £10
Vol. 518, No. 7540


nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO
CONSERVATION
SONGBIRDS À LA CARTE
Illegal harvest of millions of Mediterranean birds
PAGE 452
RESEARCH ETHICS
SAFEGUARD TRANSPARENCY
Don't let openness backfire on individuals
PAGE 459
POPULAR SCIENCE
WHEN GENES GOT 'SELFISH'
Dawkins's calling card 40 years on
PAGE 462
NATUREASIA.COM
28 January 2016
Vol. 529, No. 7587


An animation of the gradient descent method
predicting a structure for CASP13 target T1008





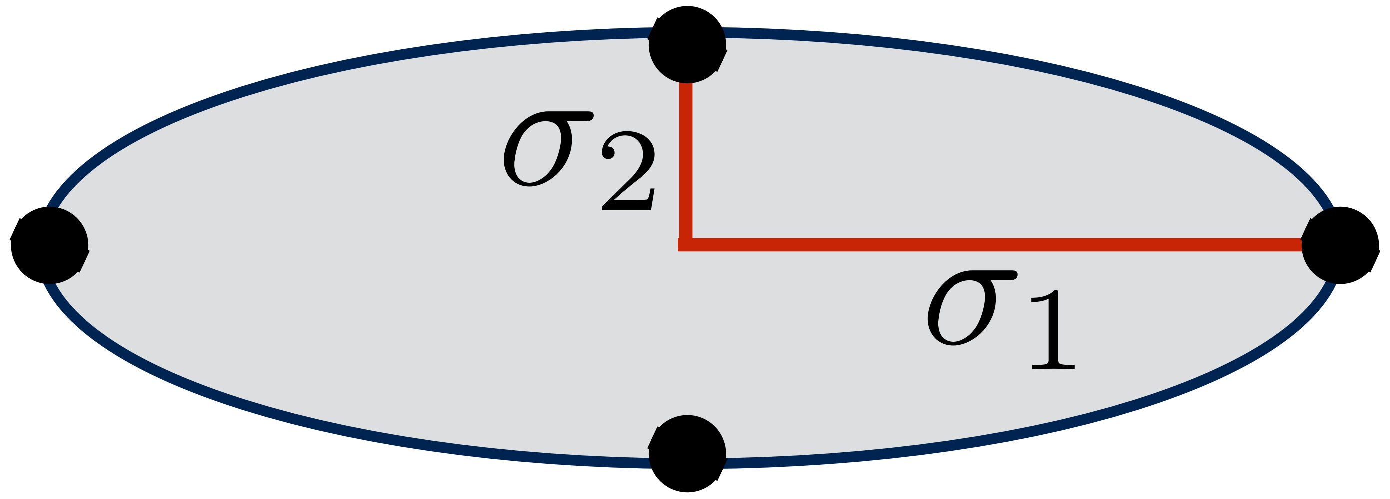


ML / AI needs biomedicine / neuroscience.

Backup Slides

Topology of PCA: Proof options

- Reduce to a simpler problem.
  - Replace X with Σ for analysis or $[\Sigma \mid -\Sigma]$ for symmetry.

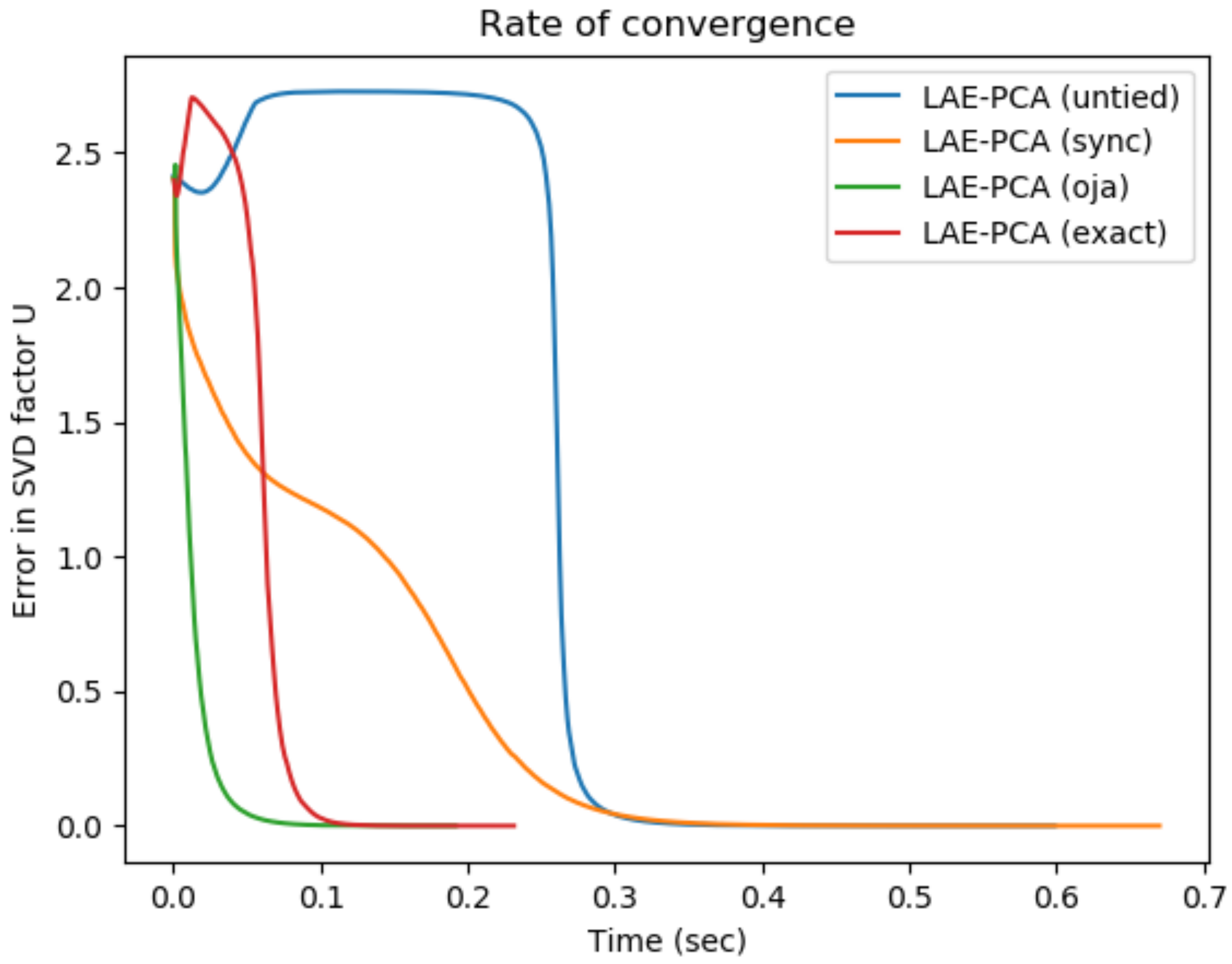


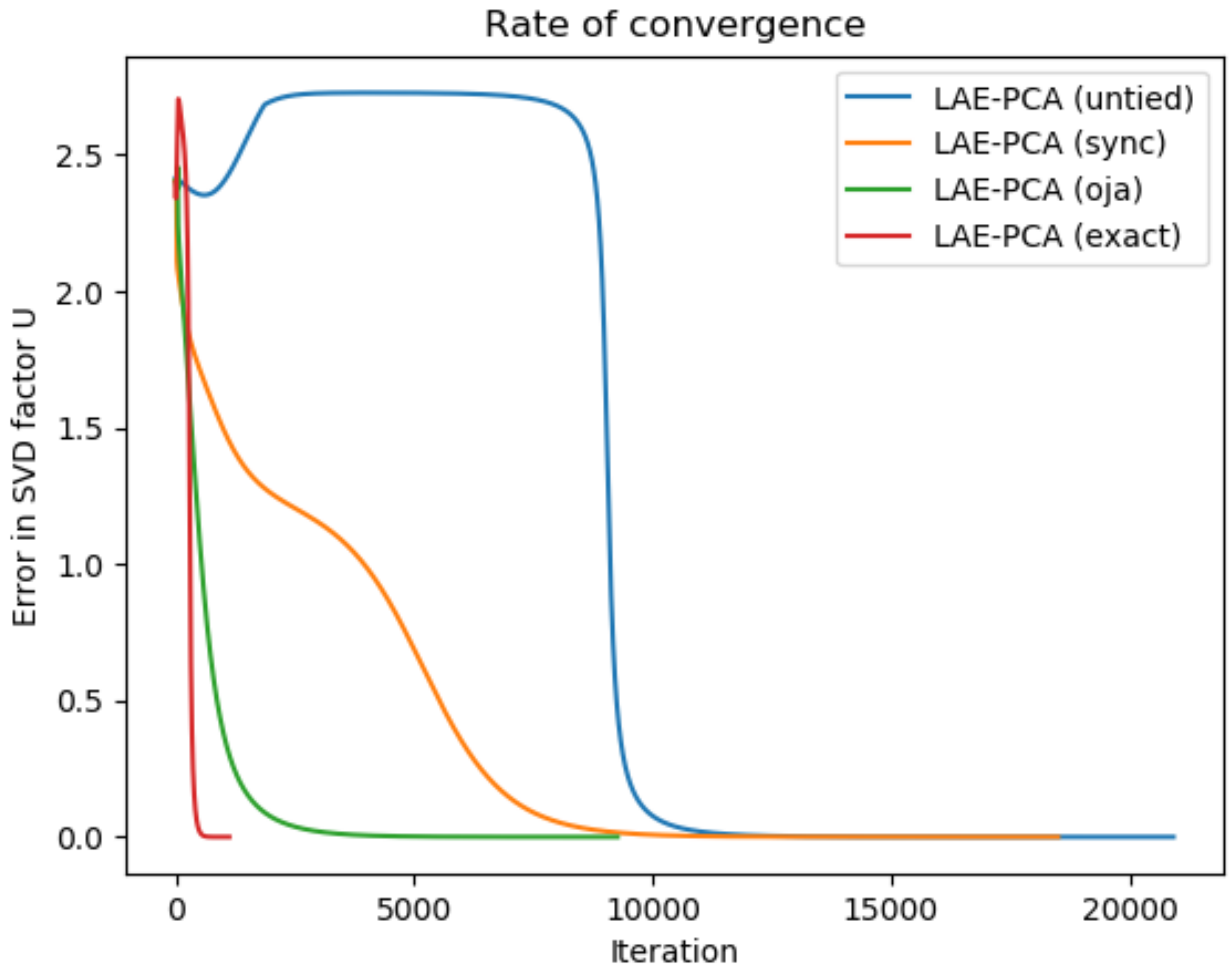
- Reduce to a harder problem that's already been solved: the LAE!

$$\begin{array}{ccc} \mathrm{V}_k(\mathbb{R}^m) & \xrightarrow{\pi: O \mapsto \mathrm{Im}(OO^\intercal)} & \mathrm{Gr}_k(\mathbb{R}^m) \\ {\scriptstyle \iota: O \mapsto (O^\intercal, O)} \big\downarrow & & \big\downarrow {\scriptstyle \mathcal{L}_X} \\ \mathbb{R}^{k\times m} \times \mathbb{R}^{m \times k} & \xrightarrow[\mathcal{L}]{} & \mathbb{R} \end{array}$$

LAE-SVD optimization

- Algorithm: optimize L_2-regularized LAE and then take SVD of the decoder.

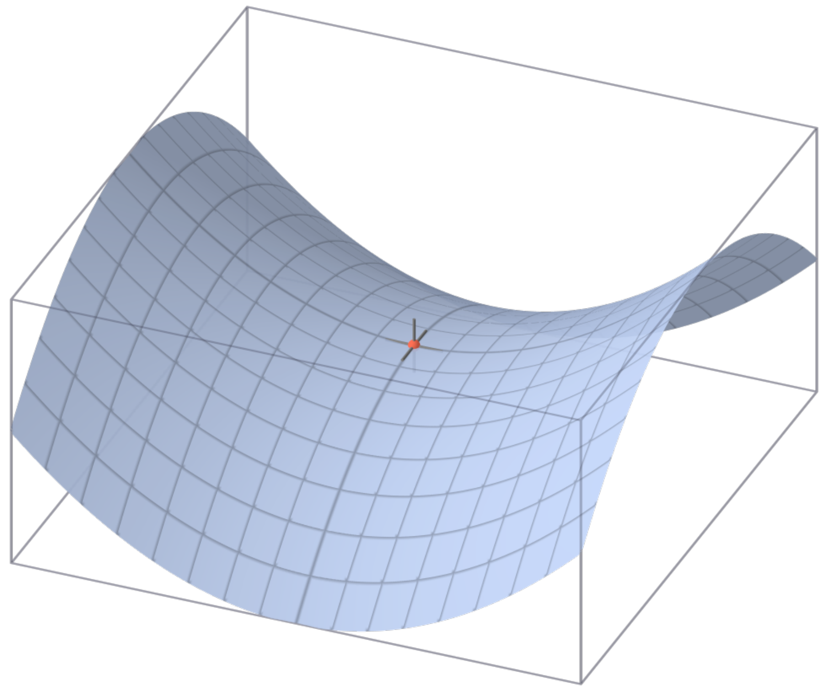




- Using SGD, algorithm resembles randomized SVD.

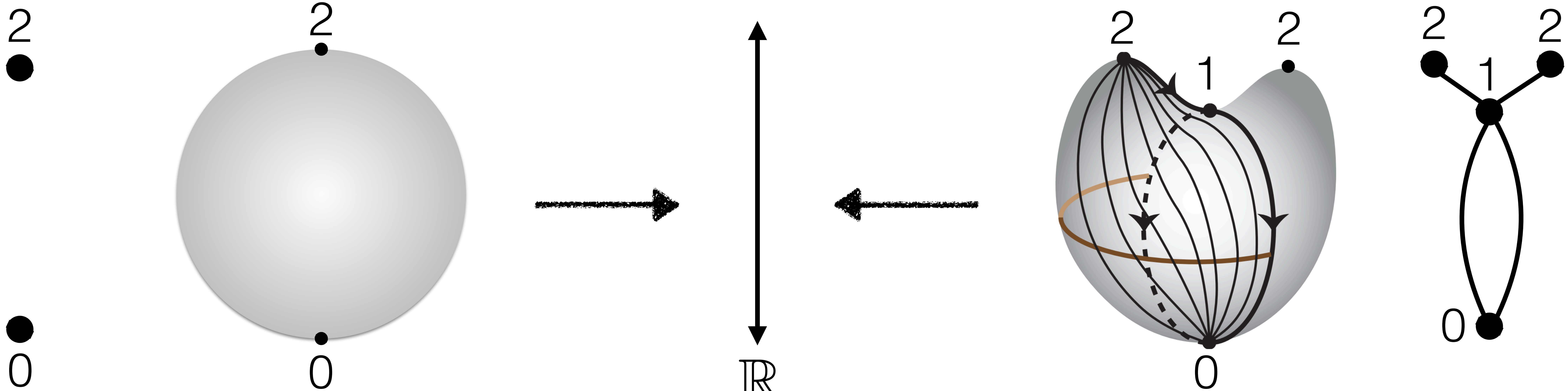
Morse theory

- Idea: Study the topology of a space via smooth functions on the space.
- A function is *Morse* if all critical points are non-degenerate:



$$f(x_1, \ldots, x_m) = c - x_1^2 - \ldots - x_d^2 + x_{d+1}^2 + \ldots + x_m^2$$

- The *Morse index* d is number of negative eigenvalues of Hessian.

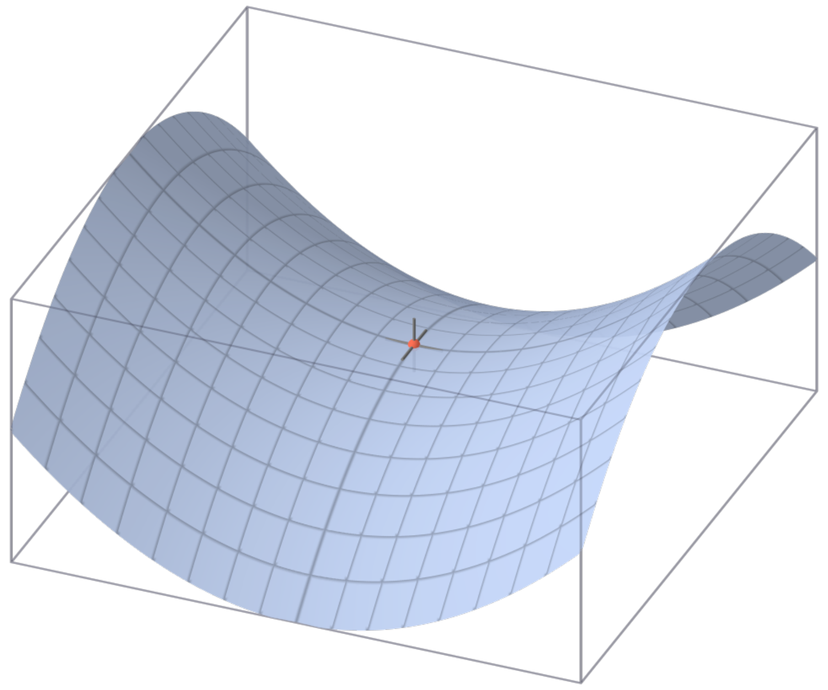


Morse theory

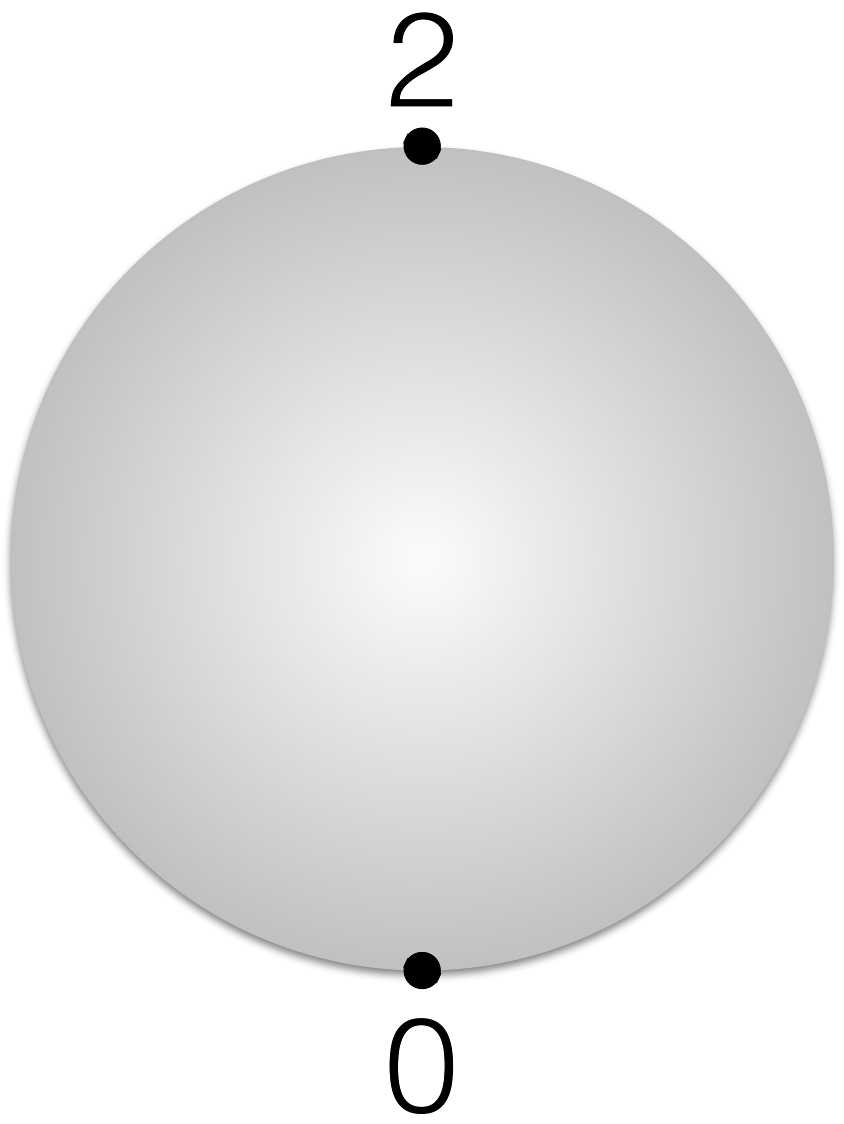
- Idea: Study the topology of a space via smooth functions on the space.

- A function is *Morse* if all critical points are non-degenerate:

$$f(x_1, \ldots, x_m) = c - x_1^2 - \ldots - x_d^2 + x_{d+1}^2 + \ldots + x_m^2$$



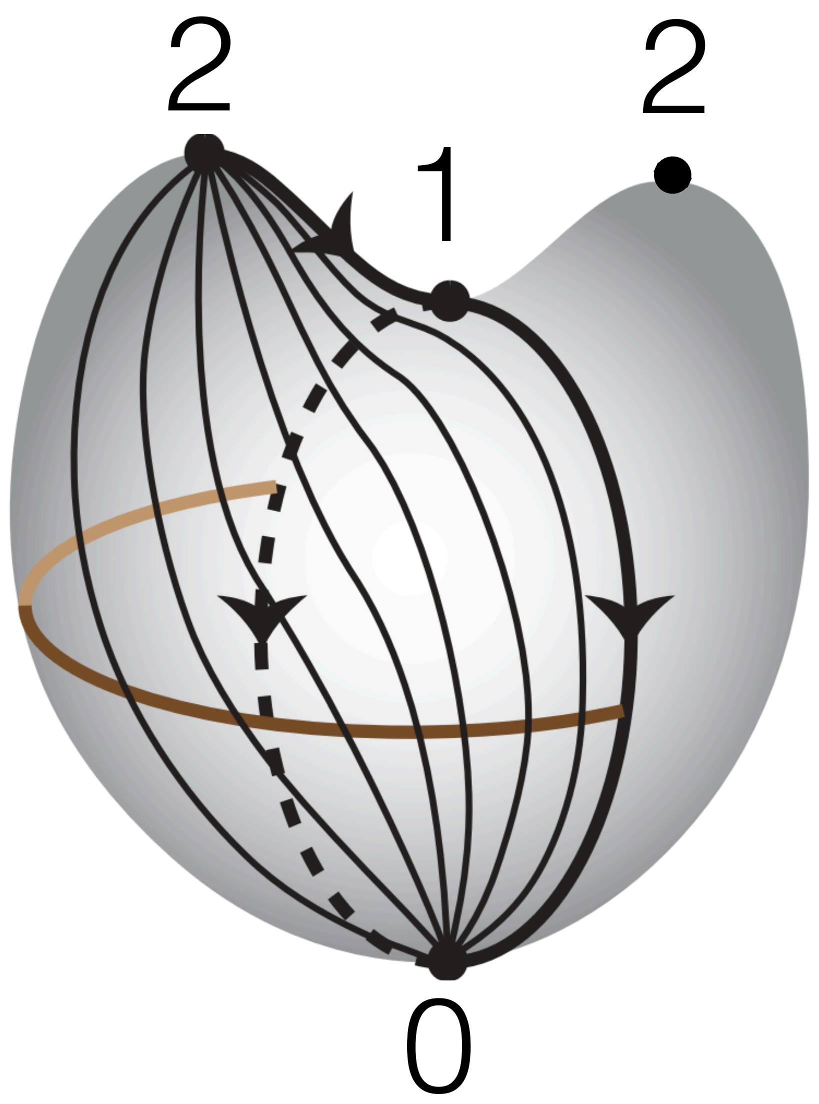
- The *Morse index* d is number of negative eigenvalues of Hessian.

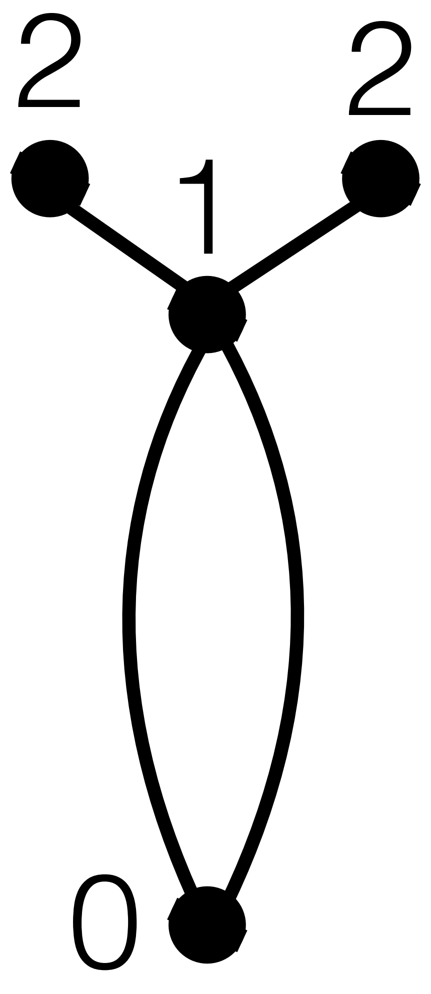


Euler characteristic

$$\chi = \sum (\text{-}1)^{d_i}$$

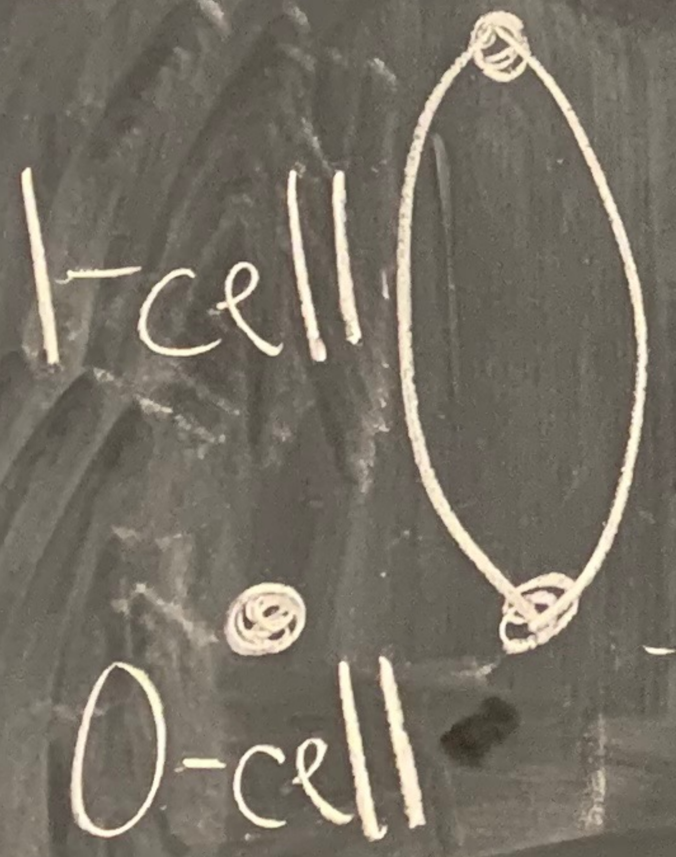
$$1 - 0 + 1 = 2 = 2 - 1 + 1$$

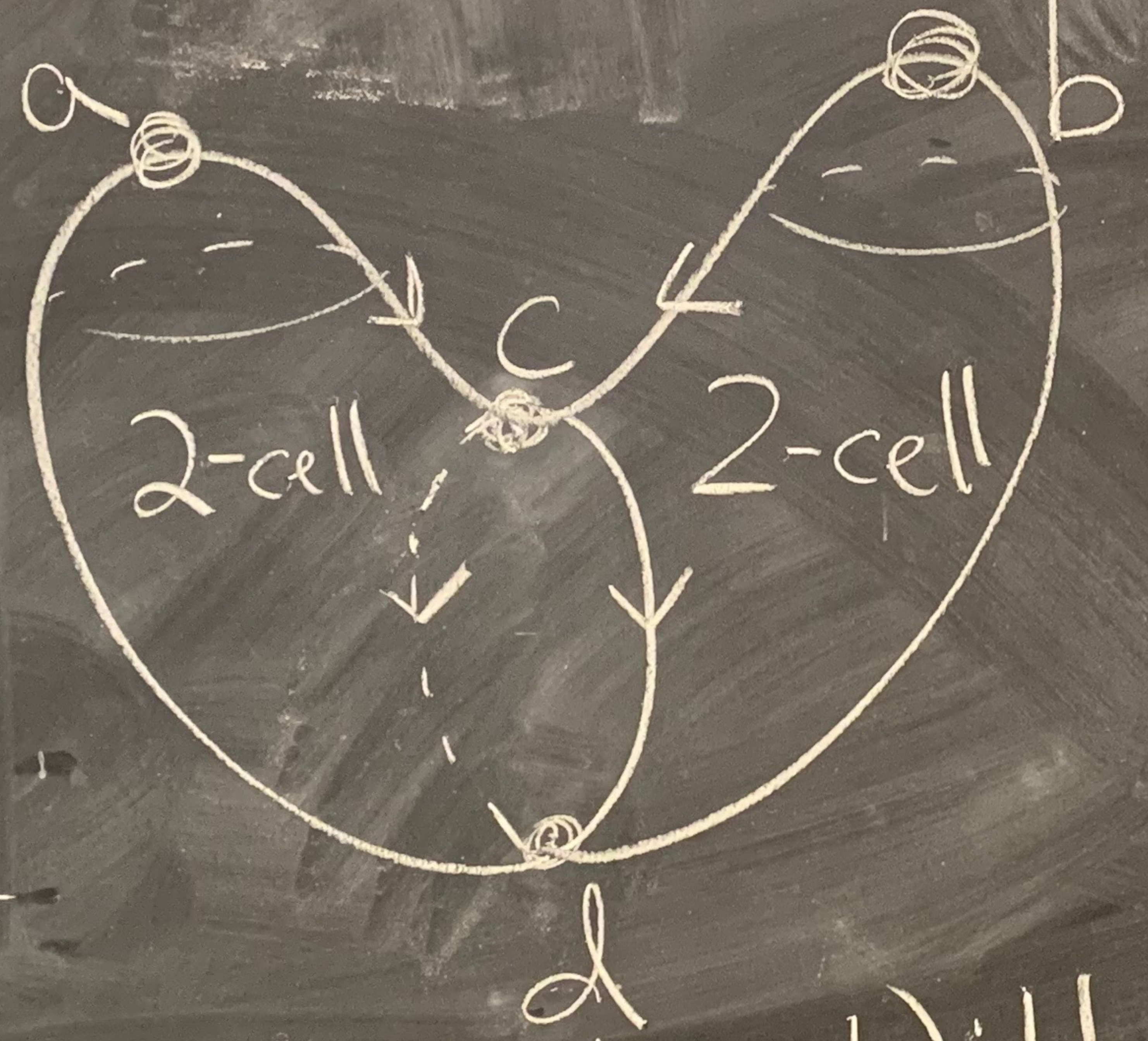




$$\langle \partial x, y \rangle = \#\left\{ \begin{array}{c} \text{gradient} \\ \text{trajectory} \\ \gamma : x \rightsquigarrow y \end{array} \right\}$$

$$\partial^2 = 0$$





Thom, Smale, Milnor, Witten

Chain Complex	(Morse) Homology
$\mathbb{F}_a \oplus \mathbb{F}_b$	$\mathbb{F}(a+b)$
$\partial_2 \downarrow [1\ \ 1]$	
$\mathbb{F}_c$	0
$\partial_1 \downarrow [0]$	
$\mathbb{F}_d$	$\mathbb{F}d$

$$H_i(M; \mathbb{F}) = \frac{\operatorname{Ker}(\partial_i)}{\operatorname{Im}(\partial_{i+1})}$$


Morse ensembling = Morsembling?
$\mathcal{L}: \mathbb{R}^P \to \mathbb{R}$
quality vs quantity
1
1/2
1/4
1/8
0
Epoch
log(n) compute
n models
Compose with FGE
Boundary (NMF)?

LAE-SVD optimization

- Algorithm: optimize L_2-regularized LAE and then take SVD of the decoder.

```
XXt = X @ X.T
while np.linalg.norm(W1 - W2.T) > epsilon:
    W1 -= alpha * ((W2.T @ (W2 @ W1 - I)) @ XXt + lamb * W1)
    W2 -= alpha * (((W2 @ W1 - I) @ XXt) @ W1.T + lamb * W2)

principal_directions, s,  _ = np.linalg.svd(W2, full_matrices = False)
eigenvalues = np.sqrt(lamb / (1 - s**2))
```

- This is a regularized version of Oja’s rule.

LAE-SVD optimization

- Algorithm: optimize L_2-regularized LAE and then take SVD of the decoder.

```
XXt = X @ X.T
diff = np.inf
while diff > epsilon:
    update = alpha * (((W2 @ W2.T - I) @ XXt) @ W2 + lamb * W2)
    W2 -= update
    diff = np.linalg.norm(update)

principal_directions, s,  _ = np.linalg.svd(W2, full_matrices = False)
eigenvalues = np.sqrt(lamb / (1 - s**2))
```

- This is a regularized version of Oja's rule.