Optimal Learning Protocols via Statistical Physics and Control Theory

Francesco Mori

UNIVERSITY OF OXFORD

NEW COLLEGE OXFORD

Statistical Physics & Machine Learning: moving forward

Institut d'Études Scientifiques de Cargèse

15th Aug 2025

General setup - Hyperparameter Optimization / Meta Learning

UNIVERSITY OF OXFORD

Model

$$f_{\mathbf{w}}(\mathbf{x})$$

Parameters

Input

Optimization algorithm

$$\mathbf{w}_{\text{final}} = F(\mathbf{w}_{\text{initial}}, \mathbf{u})$$

Hyperparameters

$$\mathbf{u}^* = \text{argmin}_{\mathbf{u}}\, \epsilon_g(\mathbf{w}_{\text{final}})$$

- In general very hard to compute
- Obtained protocols are hard to interpret

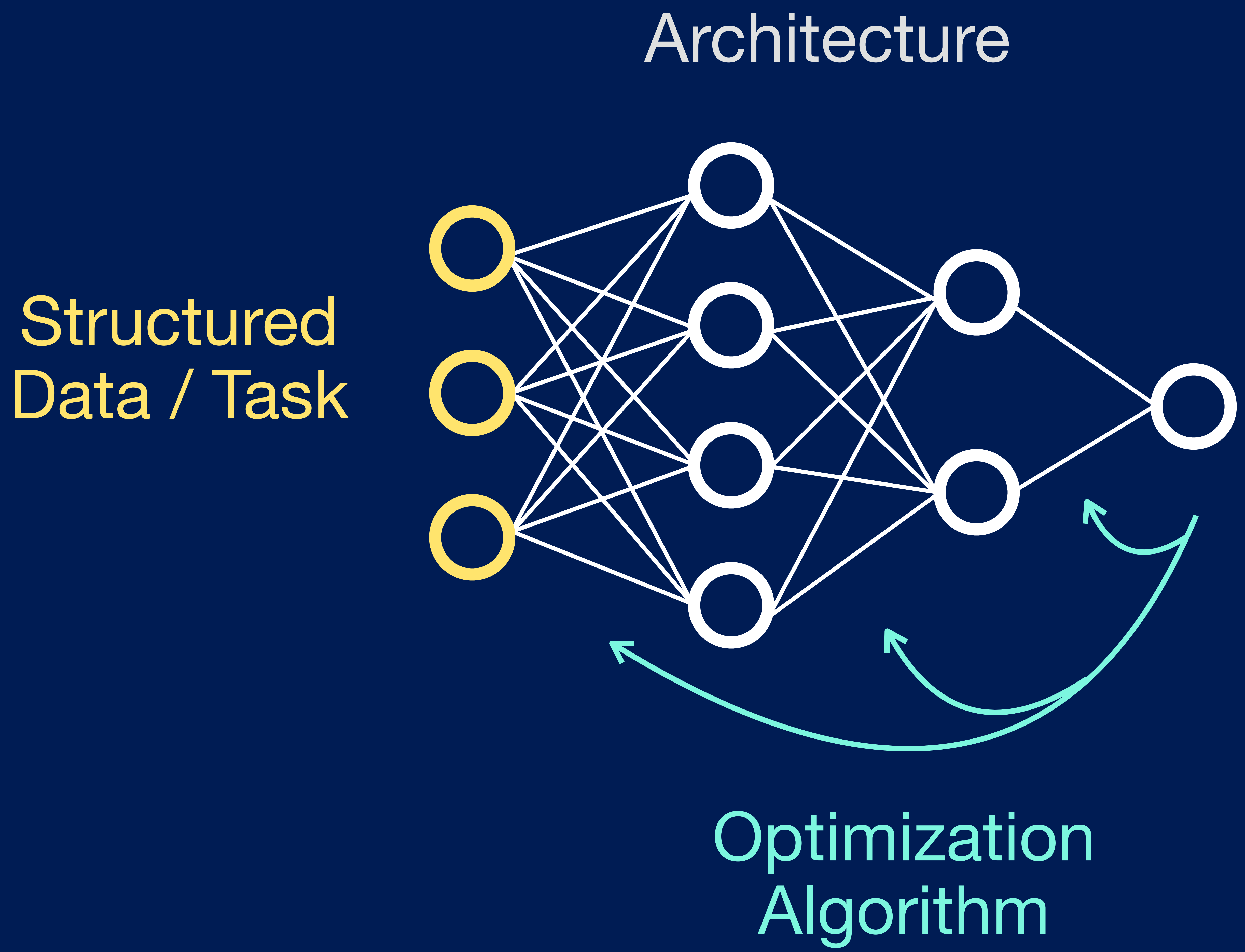
Grid/random searches

Gradient methods

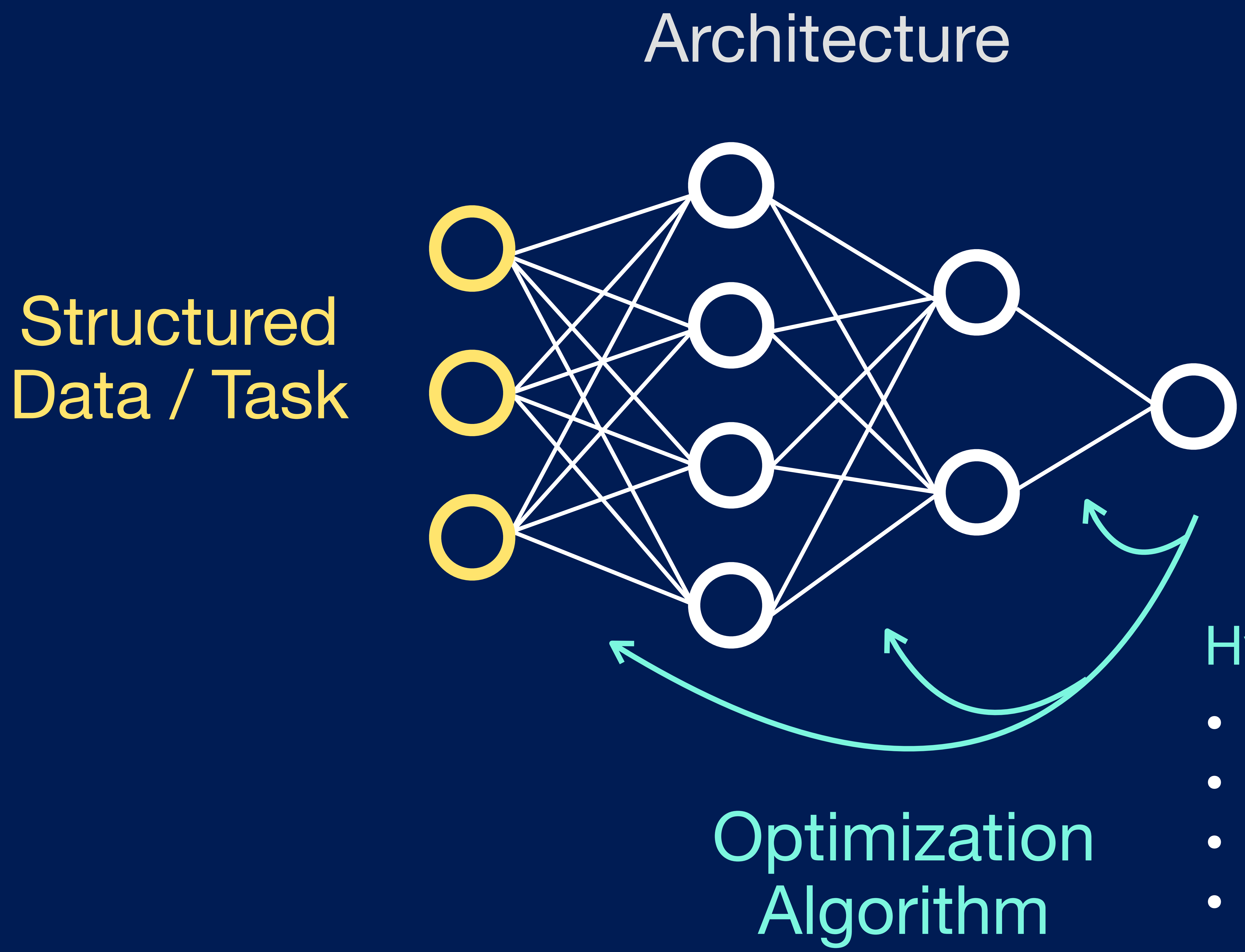
Bayesian approaches

…

Learning protocols for neural networks



Learning protocols for neural networks



Hyper-parameter schedules:

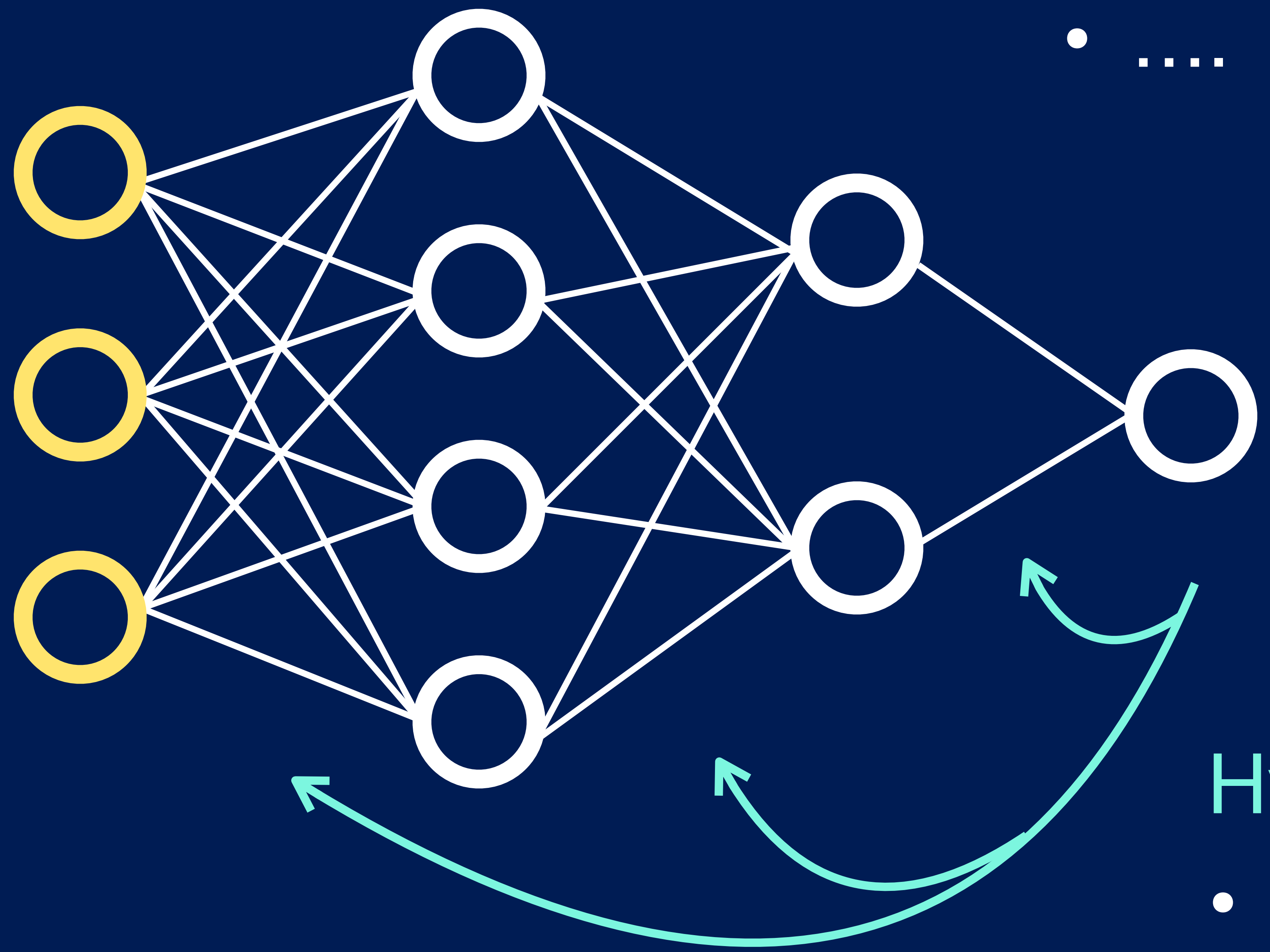
- Learning rate
- Momentum
- Batch size
- Regularization
- …

Learning protocols for neural networks

Model optimization:
- Gating
- Pruning
- Dropout
- ….

Architecture

Structured Data / Task



Hyper-parameter schedules:
- Learning rate
- Momentum
- Batch size
- Regularization
- …

Optimization Algorithm

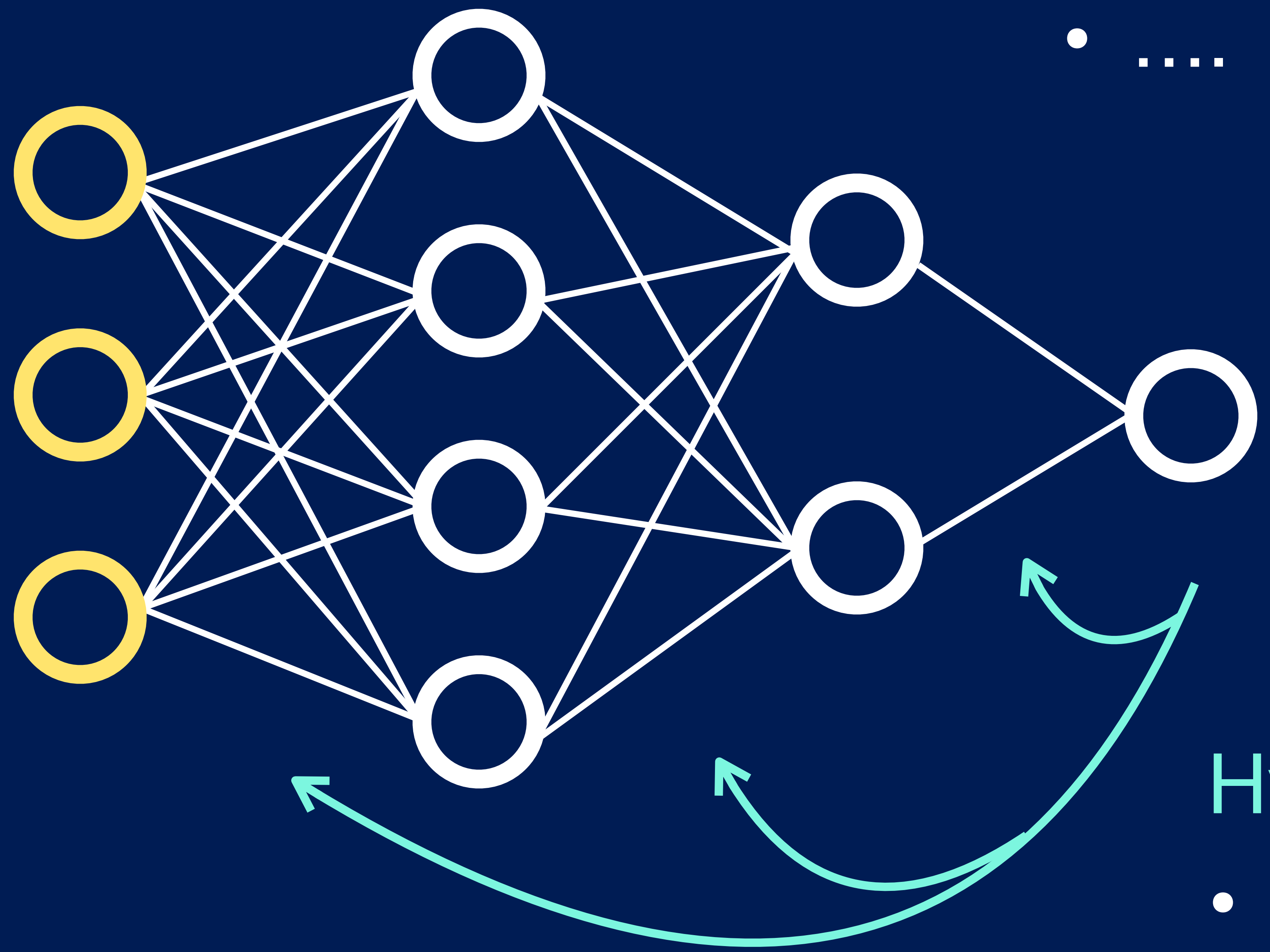
Learning protocols for neural networks

UNIVERSITY OF OXFORD

Architecture

Model optimization:
- Gating
- Pruning
- Dropout
- ….

Structured
Data / Task



Dynamic data / task selection:
- Active learning
- Curriculum learning
- Transfer learning
- Multi-task learning
- …

Hyper-parameter schedules:
- Learning rate
- Momentum
- Batch size
- Regularization
- …

Optimization
Algorithm

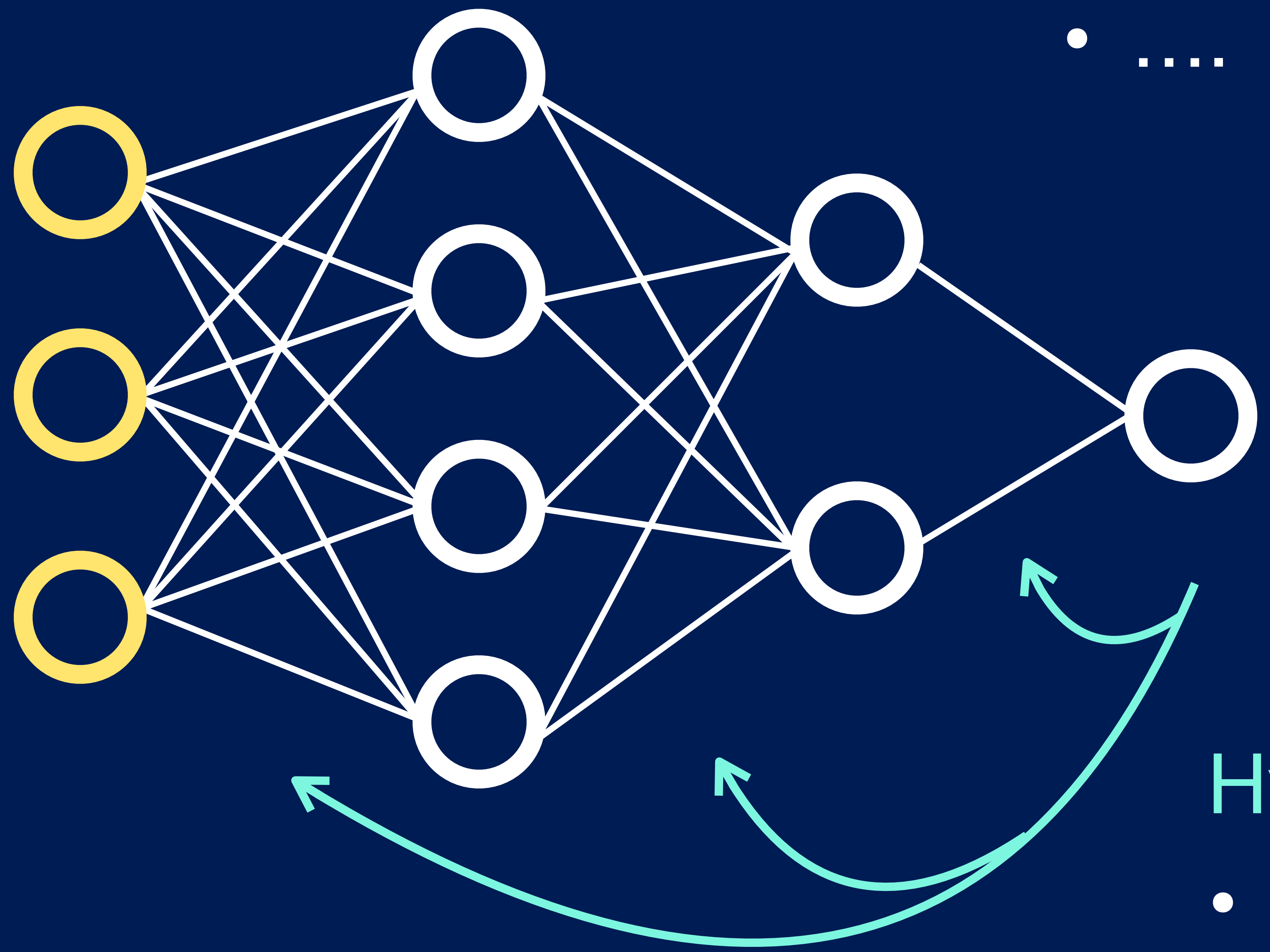
Learning protocols for neural networks

Complex interplay between time-dependent protocols and nonlinear learning dynamics

Model optimization:
- Gating
- Pruning
- Dropout
- ….

Architecture

Structured Data / Task



Dynamic data / task selection:
- Active learning
- Curriculum learning
- Transfer learning
- Multi-task learning
- …

Hyper-parameter schedules:
- Learning rate
- Momentum
- Batch size
- Regularization
- …

Optimization Algorithm

* Optimal learning protocols?

* in the final performance

UNIVERSITY OF OXFORD

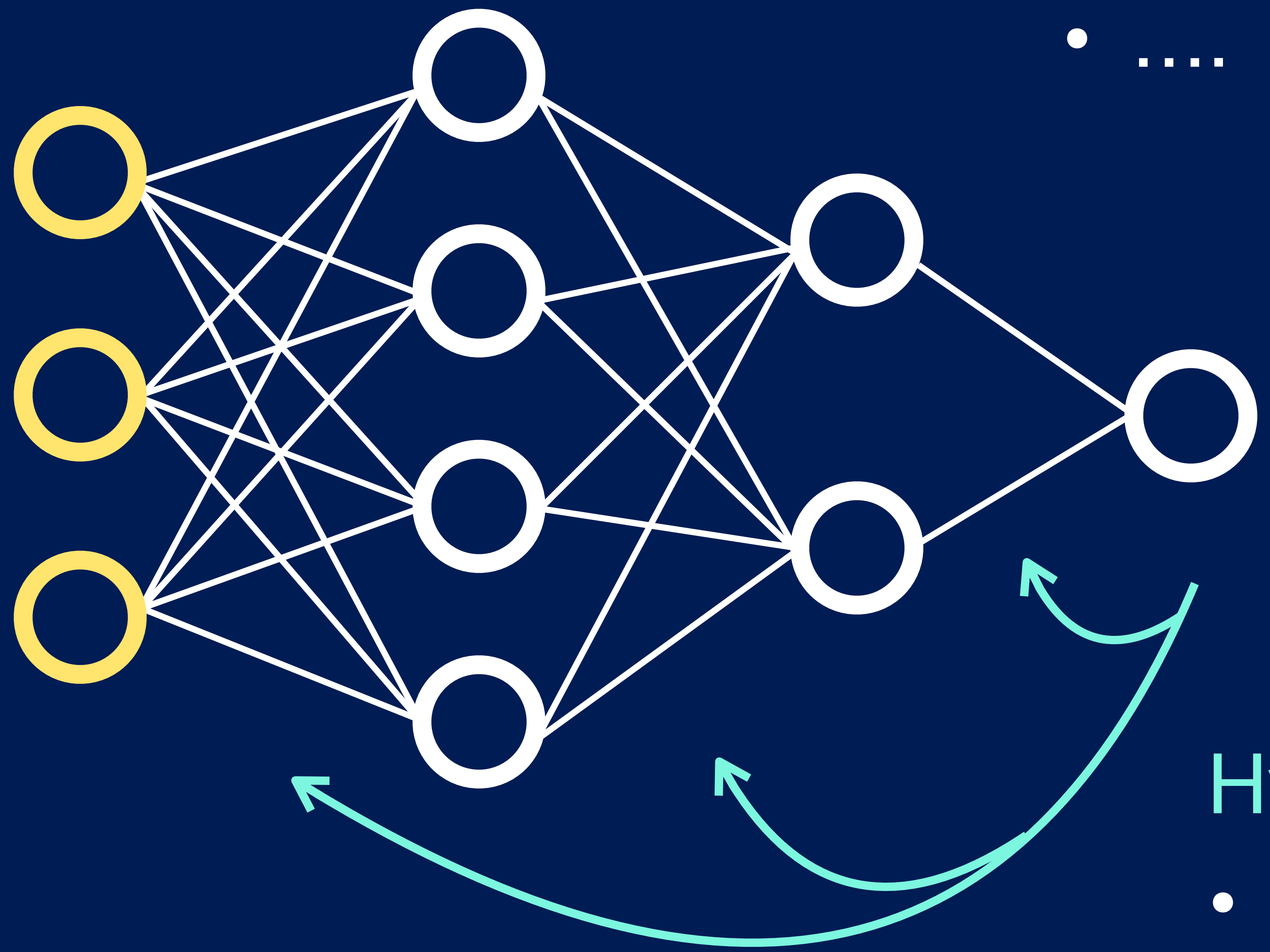
Complex interplay between time-dependent protocols and nonlinear learning dynamics

Model optimization:

- Gating
- Pruning
- Dropout
- ….

Architecture

Structured Data / Task



Dynamic data / task selection:

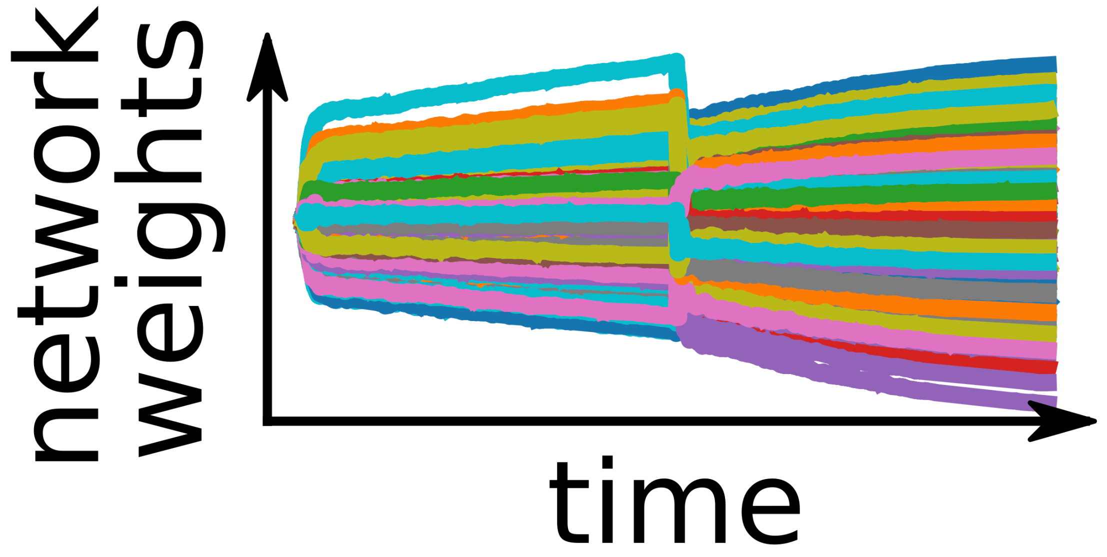
- Active learning
- Curriculum learning
- Transfer learning
- Multi-task learning
- …

Hyper-parameter schedules:

- Learning rate
- Momentum
- Batch size
- Regularization
- …

Optimization Algorithm

Dimensionality reduction

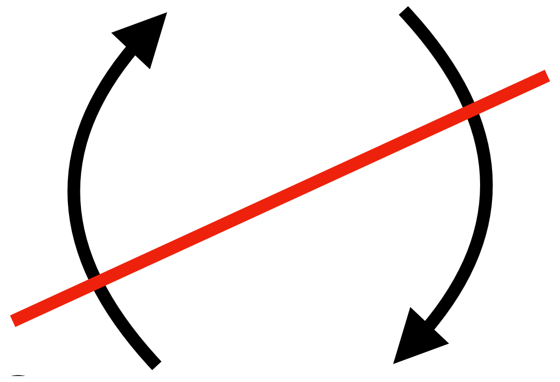


$$\mathbf{w}^{\mu+1} = \mathbf{w}^{\mu} - \nabla \mathscr{L}^{\mu}\left(\mathbf{u}^{\mu}\right)$$

$$\mathbf{w} \in \mathbb{R}^{N}$$

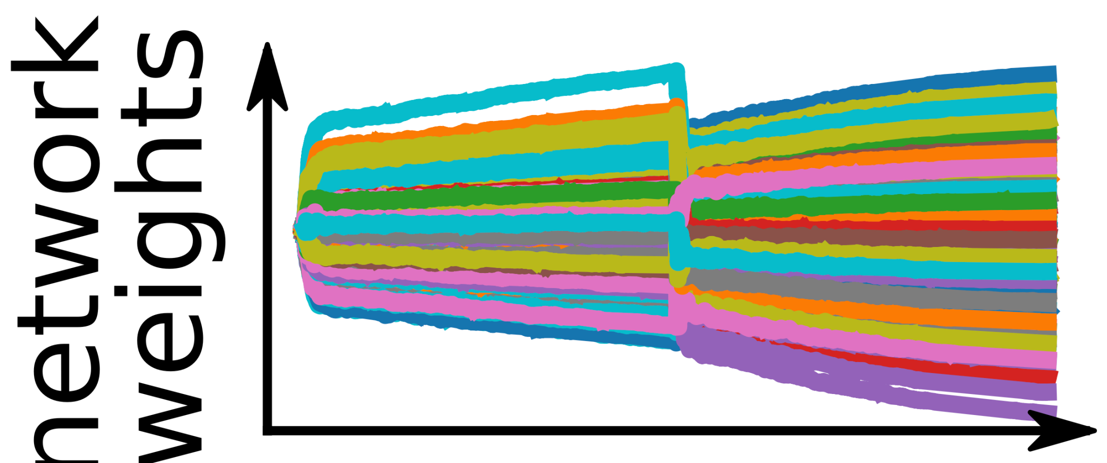
control variables

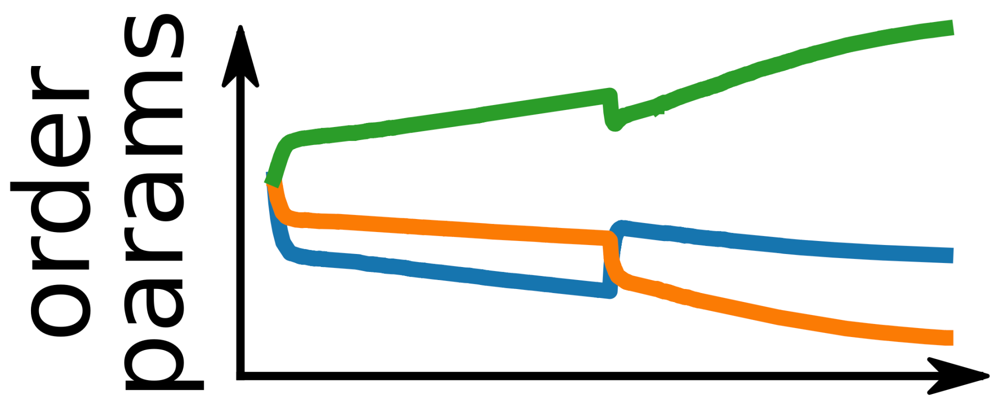
High-dimensional learning dynamics



Control theory is computationally demanding

Dimensionality reduction





$$\mathbf{w}^{\mu+1} = \mathbf{w}^{\mu} - \nabla \mathscr{L}^{\mu}\left(\mathbf{u}^{\mu}\right)$$

$$\mathbf{w} \in \mathbb{R}^N$$

control
variables

$$\dot{\mathbf{Q}} = f(\mathbf{Q}, \mathbf{u})$$

High-dimensional
learning dynamics

Statistical
Physics

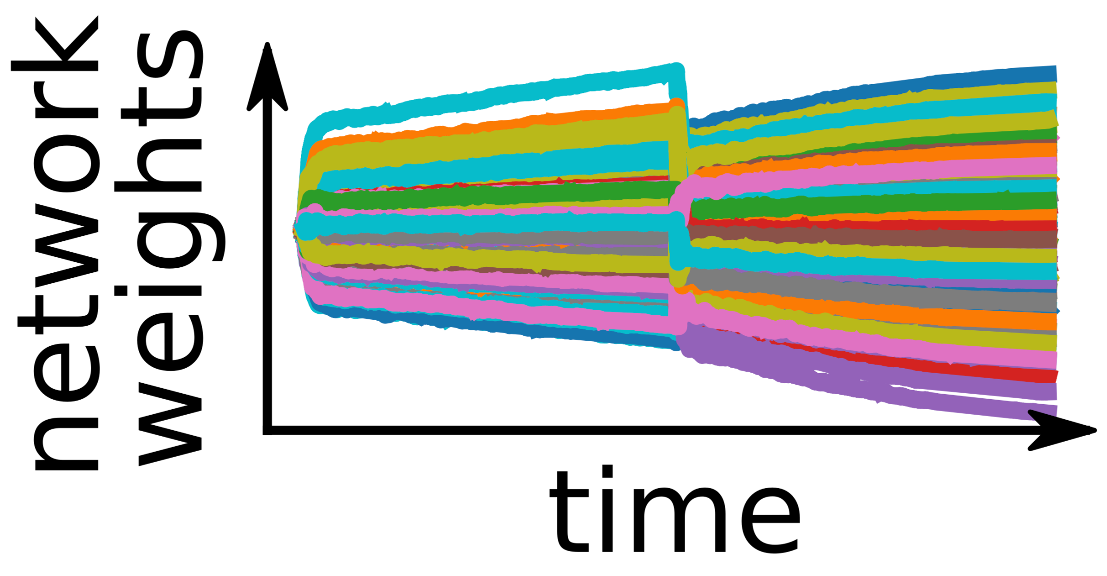
$N \to \infty$

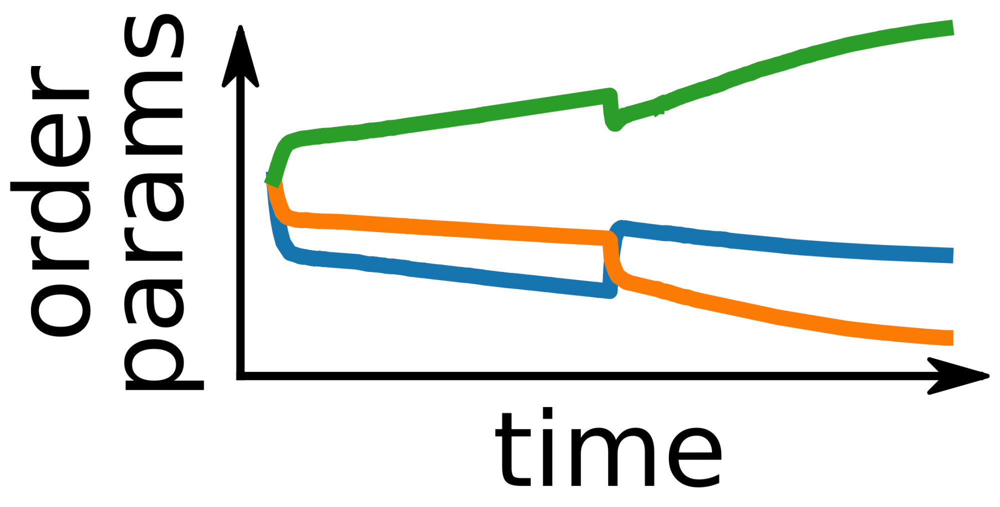
effective dynamics of
order parameters

Control theory is
computationally demanding

Control theory can
be applied
+ interpretability

Dimensionality reduction





$$\mathbf{w}^{\mu+1} = \mathbf{w}^{\mu} - \nabla \mathscr{L}^{\mu}\left(\mathbf{u}^{\mu}\right)$$

$$\mathbf{w} \in \mathbb{R}^N$$

control variables

$$\dot{\mathbf{Q}} = f(\mathbf{Q}, \mathbf{u})$$

Optimal control

High-dimensional learning dynamics

Statistical Physics

$N \to \infty$

effective dynamics of order parameters

$\mathbf{u}^*$

Control theory is computationally demanding

Control theory can be applied

\+ interpretability

NEW

Francesca Mignacco, FM, *arXiv:2507.07907*

Setup — Data model

Dataset

$$\mathscr{D} = \{\mathbf{x}_\mu, y_\mu\}_{\mu=1}^{P}$$

$$\mathbf{x}_\mu \sim \sum_{c=1}^{C} p_c \, \mathcal{N}\left(\frac{\mu_c}{\sqrt{N}}, \sigma_c^2 \mathbf{I}_N\right) \in \mathbb{R}^N$$

$$y_\mu = f^*_{\mathbf{w}_*}(\mathbf{x}_\mu) + \sigma_n \, z_\mu$$

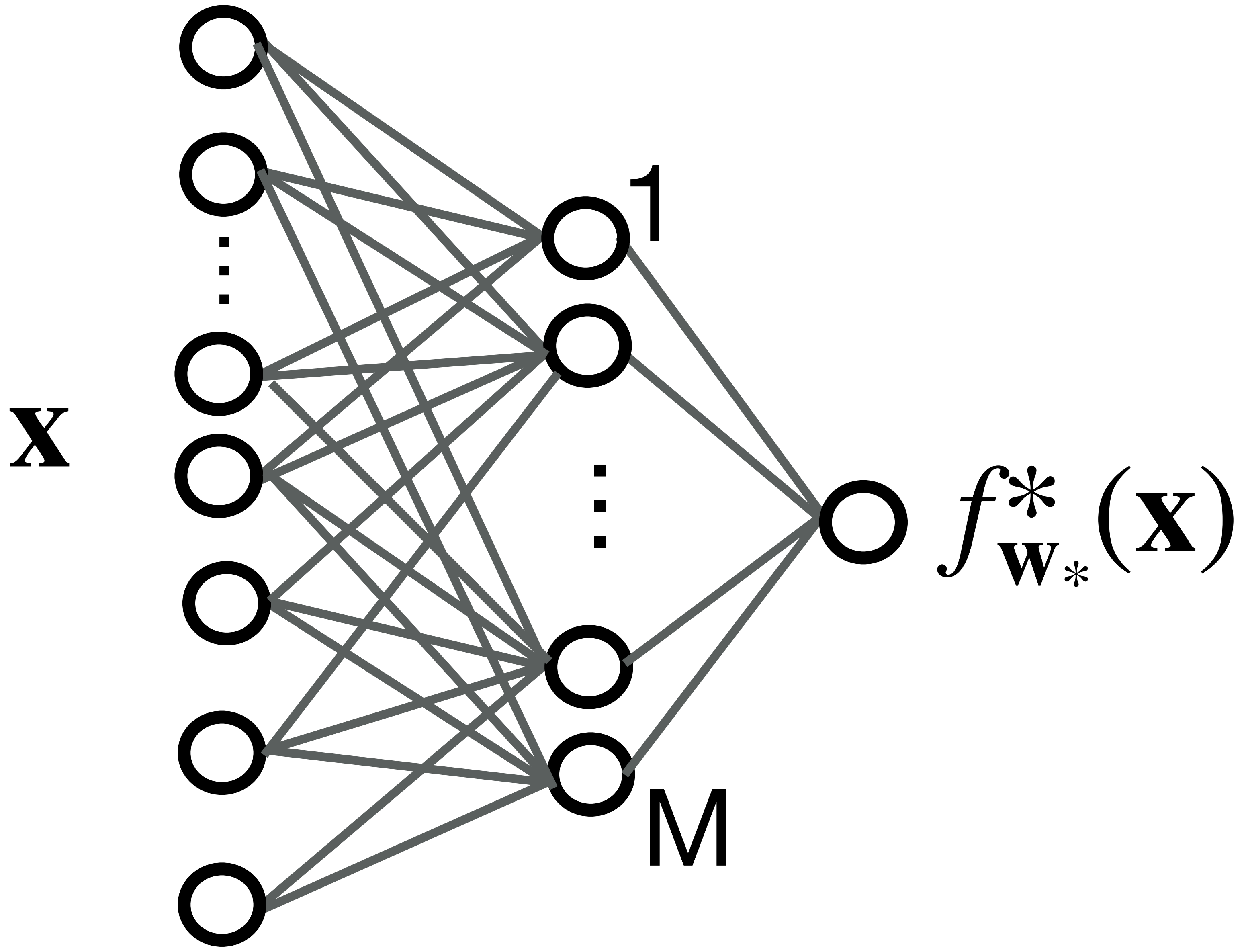
Teacher Network

(M hidden nodes)

$\mathbf{w}_* \in \mathbb{R}^{N\times M}$

Label noise

$z_\mu \sim \mathcal{N}(0,1)$



[H. Cui, J. Stat. Mech (2025)]

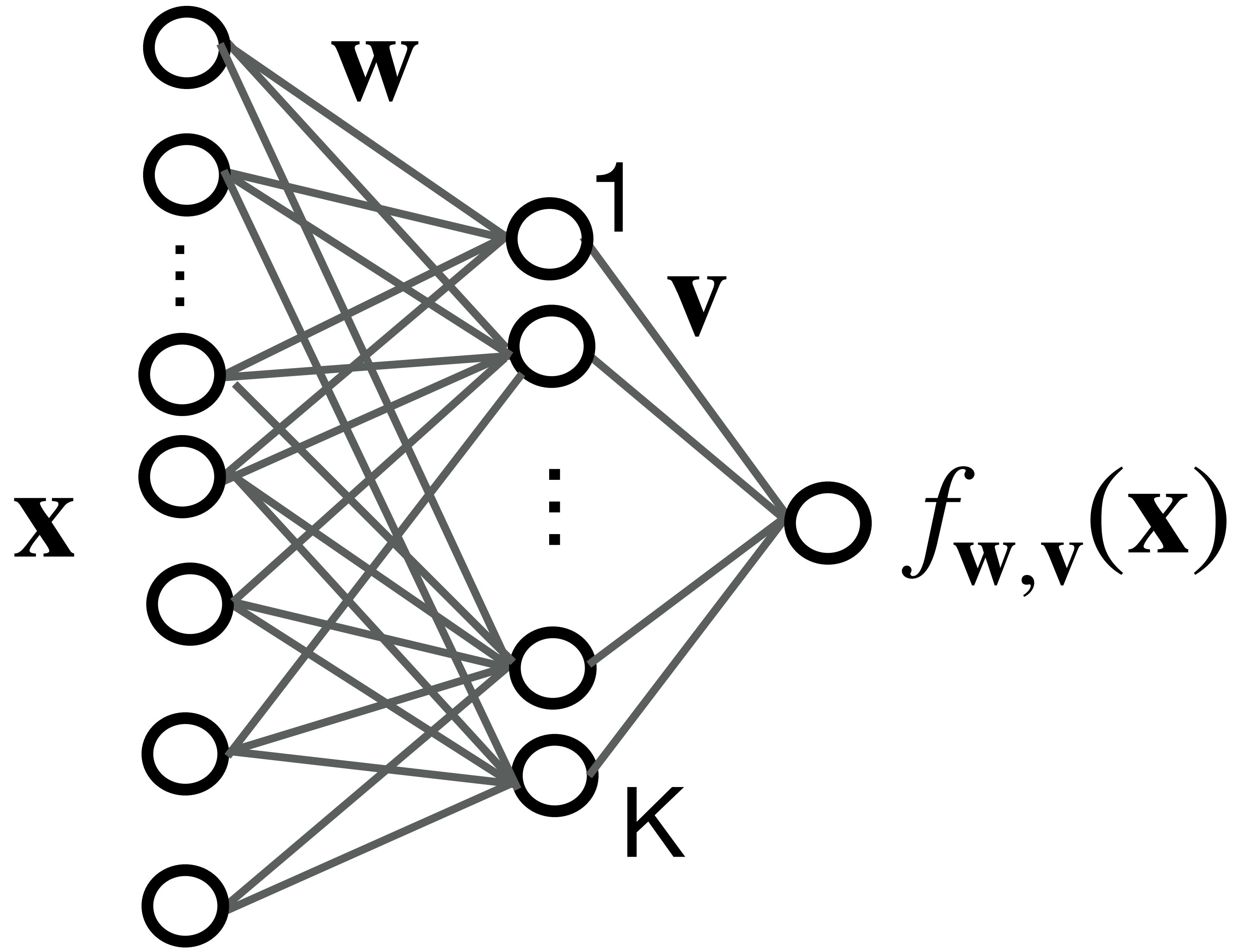
Setup — Architecture and algorithm

Two-layer neural network

(K hidden nodes)

$$f_{\mathbf{w},\mathbf{v}}(\mathbf{x}) = \frac{1}{\sqrt{K}} \sum_{k=1}^{K} v_k \; g\left(\frac{\mathbf{w}_k \cdot \mathbf{x}}{\sqrt{N}}\right)$$

$$\mathbf{w} \in \mathbb{R}^{K \times M}$$



Setup — Architecture and algorithm

UNIVERSITY OF OXFORD

Two-layer neural network

(K hidden nodes)

$$f_{\mathbf{w},\mathbf{v}}(\mathbf{x}) = \frac{1}{\sqrt{K}} \sum_{k=1}^{K} v_k \; g\left(\frac{\mathbf{w}_k \cdot \mathbf{x}}{\sqrt{N}} \right)$$

$$\mathbf{w} \in \mathbb{R}^{K \times M}$$

Per-sample cost function

$$\mathscr{L}(\mathbf{w}, \mathbf{v} \,|\, \mathbf{x}, y) = \underbrace{\ell}_{\text{Loss function}} \left(f_{\mathbf{w},\mathbf{v}}(\mathbf{x}), y \right) + \underbrace{g(\mathbf{w}, \mathbf{v})}_{\text{Regularization function}}$$

Setup — Architecture and algorithm

UNIVERSITY OF OXFORD

Two-layer neural network

(K hidden nodes)

$$f_{\mathbf{w},\mathbf{v}}(\mathbf{x}) = \frac{1}{\sqrt{K}} \sum_{k=1}^{K} v_k \; g\left(\frac{\mathbf{w}_k \cdot \mathbf{x}}{\sqrt{N}} \right)$$

$$\mathbf{w} \in \mathbb{R}^{K \times M}$$

Per-sample cost function

$$\mathscr{L}(\mathbf{w}, \mathbf{v} \,|\, \mathbf{x}, y) = \ell \left(f_{\mathbf{w},\mathbf{v}}(\mathbf{x}), y \right) + g(\mathbf{w}, \mathbf{v})$$

Loss function

Regularization function

One-pass SGD

$$\mathbf{w}^{\mu+1} = \mathbf{w}^{\mu} - \eta \, \nabla_{\mathbf{w}} \mathscr{L}(\mathbf{w}^{\mu}, \mathbf{v}^{\mu} \,|\, \mathbf{x}^{\mu}, y^{\mu})$$

Dynamics of order parameters

$$\mathbf{S} \in \mathbb{R}^{K\times K} \qquad \mathbf{M} \in \mathbb{R}^{K\times M} \qquad \mathbf{R} \in \mathbb{R}^{K\times C}$$

$$S_{kk'} = \frac{\mathbf{w}_k \cdot \mathbf{w}_{k'}}{N} \qquad M_{km} = \frac{\mathbf{w}_k \cdot \mathbf{w}_{*,m}}{N} \qquad R_{kc} = \frac{\mathbf{w}_k \cdot \mu_c}{N}$$

$$\mathbf{Q} \equiv (\mathrm{vec}(\mathbf{S}), \mathrm{vec}(\mathbf{M}), \mathrm{vec}(\mathbf{R})) \in \mathbb{R}^{K(K+M+C)}$$

One-pass SGD

ODEs for the order parameters

$$\mathbf{w}^{\mu+1} = \mathbf{w}^{\mu} - \eta \nabla_{\mathbf{w}} \mathscr{L}^{\mu}(\mathbf{u}_{\mu}) \xrightarrow[\text{with } \alpha = \mu/N]{N \to \infty\,, \mu \to \infty} \frac{d\mathbf{Q}(\alpha)}{d\alpha} = f_{\mathbf{Q}}\left(\mathbf{Q}(\alpha), \mathbf{u}(\alpha)\right)$$

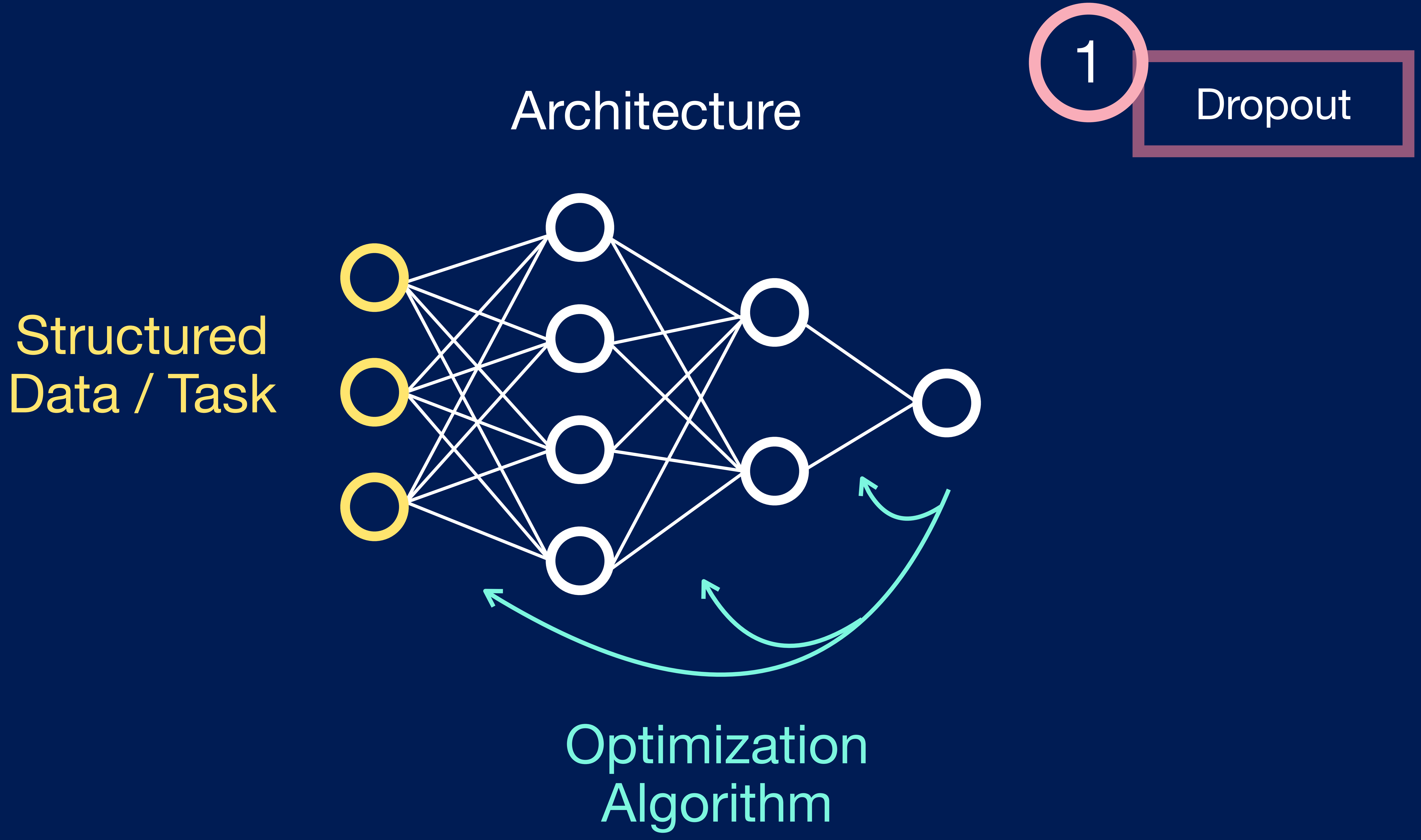
$\mu = 1\,, \ldots, P$

$\alpha \in [0,\alpha_F]$

[Saad & Solla (1995); Biehl & Schwarze (1995);Goldt et al (2019); Veiga et al. (2022), Arnaboldi et al (2023) … and many more]

Optimal learning protocols

Francesca Mignacco
CUNY & Priceton



FM, F. Mignacco, *arXiv:2505.07792*

F. Mignacco, **FM**, *arXiv:2507.07907*

Optimal learning protocols

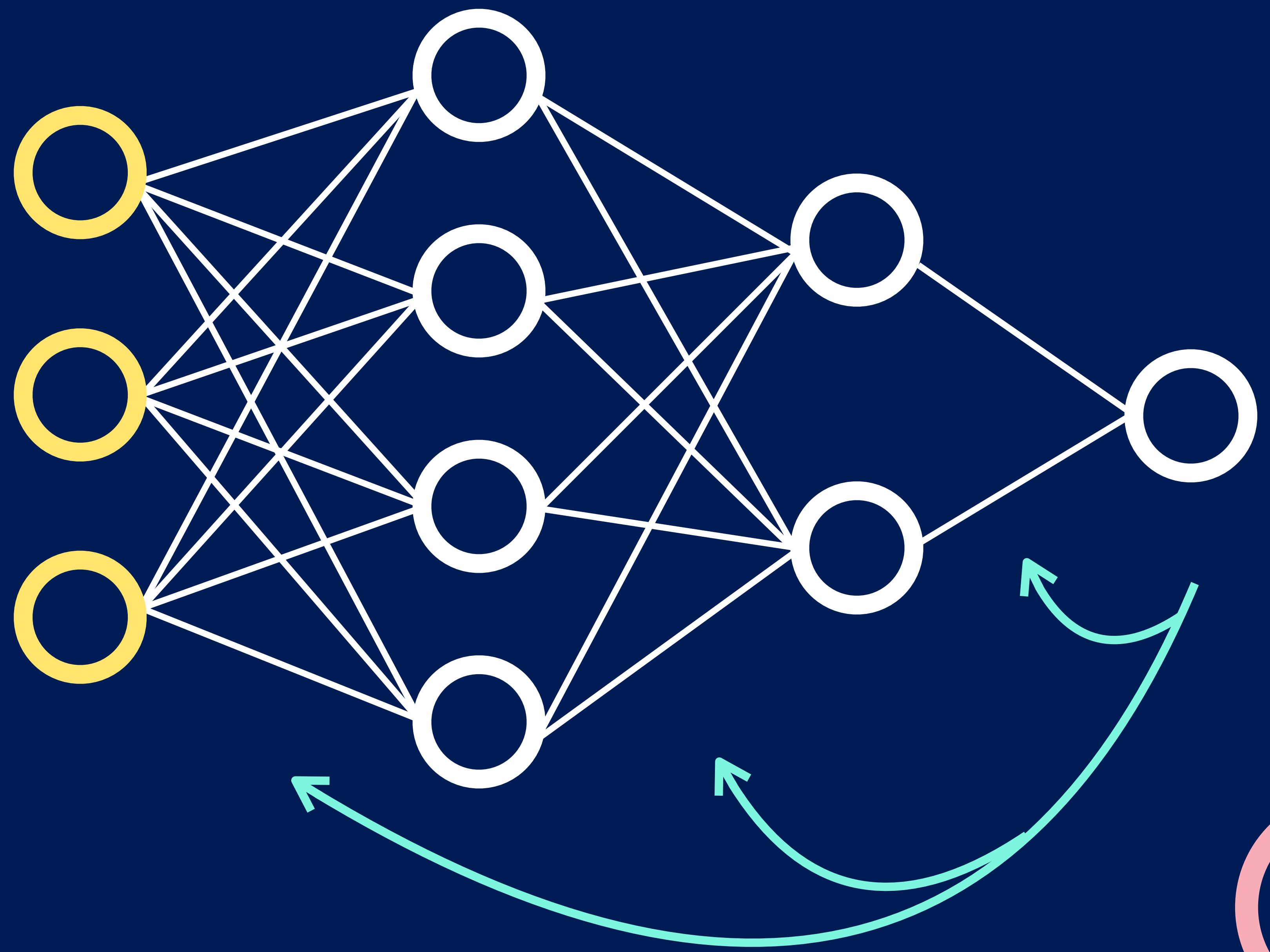
Francesca
Mignacco
CUNY & Priceton

Stefano Sarao Mannelli
Chalmers University
Gothenburg University

Architecture

1 Dropout

Structured
Data / Task



2 Continual learning

2 Learning rate

Optimization
Algorithm

FM, F. Mignacco, *arXiv:2505.07792*

F. Mignacco, **FM**, *arXiv:2507.07907*

FM, S. Sarao Mannelli, F. Mignacco, *ICLR 2025*

Optimal learning protocols

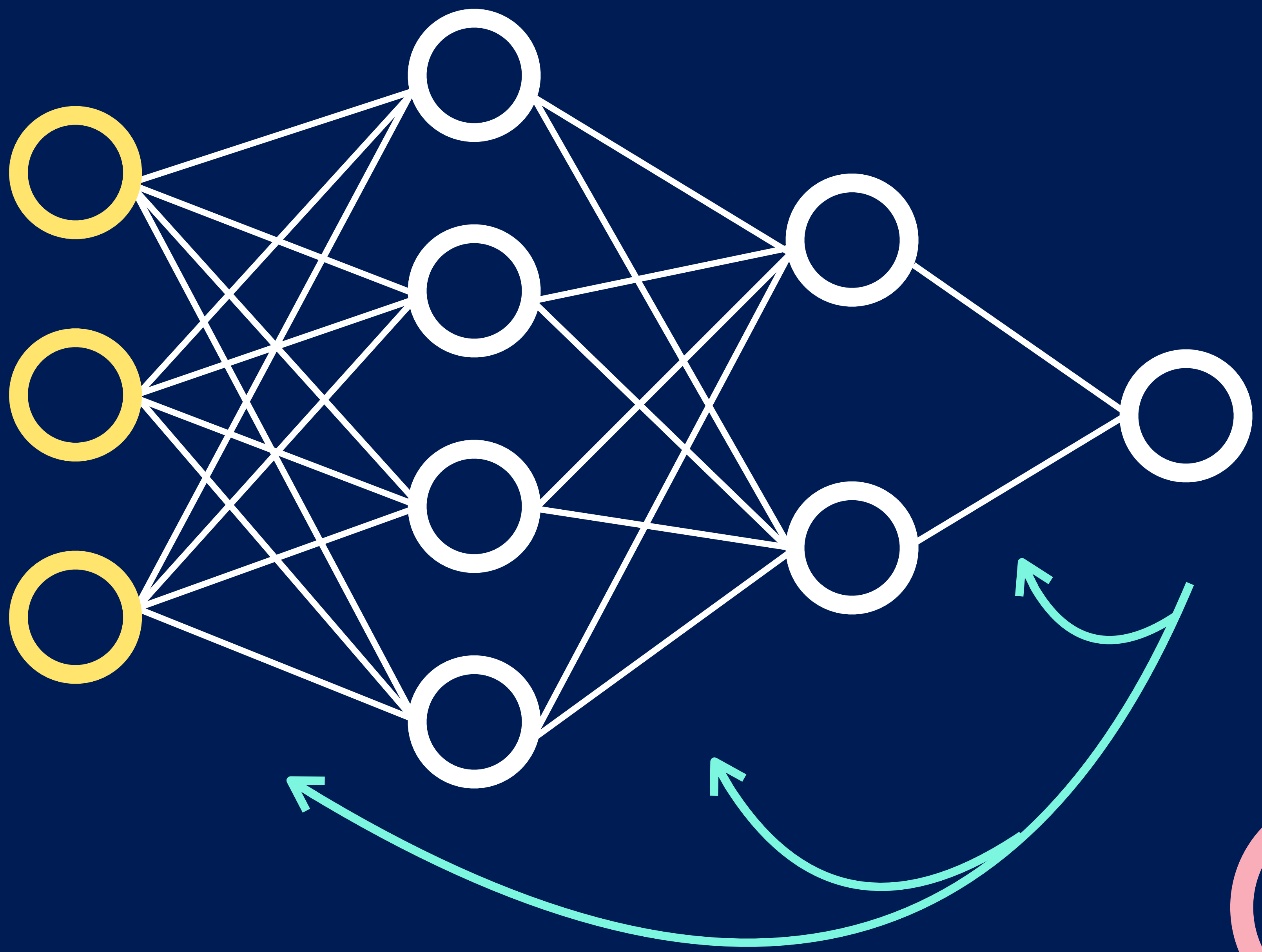
Francesca Mignacco
CUNY & Priceton

Stefano Sarao Mannelli
Chalmers University
Gothenburg University

Architecture

1 Dropout

Structured Data / Task



2 Continual learning

3 Denoising autoencoder

Optimization Algorithm

2 Learning rate

3 Batch size

FM, F. Mignacco, *arXiv:2505.07792*

F. Mignacco, **FM**, *arXiv:2507.07907*

FM, S. Sarao Mannelli, F. Mignacco, *ICLR 2025*

Can we compute optimal meta-parameter schedules?

UNIVERSITY OF OXFORD

Forward learning dynamics

$$\dot{\mathbf{Q}} = f(\mathbf{Q}, \mathbf{u})$$

Control variable

Order parameters

Pontryagin's maximum principle

Variatonal approach

Cost functional: $\mathscr{F}[\mathbf{Q}, \mathbf{u}] = \langle\ \epsilon_g(\alpha_F)\ \rangle$

Forward learning dynamics

$$\dot{\mathbf{Q}} = f(\mathbf{Q}, \mathbf{u})$$

Control variable

Order parameters

Pontryagin’s maximum principle

Variatonal approach
Cost functional:

$$\mathscr{F}[\mathbf{Q}, \hat{\mathbf{Q}}, \mathbf{u}] = \langle\ \epsilon_g(\alpha_F)\ \rangle + \int_0^{\alpha_F} \mathrm{d}\alpha\ \hat{\mathbf{Q}}(\alpha) \cdot \left[-\dot{\mathbf{Q}}(\alpha) + f\left(\mathbf{Q}(\alpha), \mathbf{u}(\alpha)\right)\right]$$

Forward learning dynamics

$$\dot{\mathbf{Q}} = f(\mathbf{Q}, \mathbf{u})$$

Control variable

Order parameters

Pontryagin's maximum principle

Variatonal approach
Cost functional: $$\mathscr{F}[\mathbf{Q},\hat{\mathbf{Q}},\mathbf{u}] = \langle\ \epsilon_g(\alpha_F)\ \rangle + \int_0^{\alpha_F} \mathrm{d}\alpha\ \hat{\mathbf{Q}}(\alpha)\cdot\left[-\dot{\mathbf{Q}}(\alpha) + f\big(\mathbf{Q}(\alpha),\mathbf{u}(\alpha)\big)\right]$$

Forward learning dynamics

$$\dot{\mathbf{Q}} = f(\mathbf{Q},\mathbf{u})$$

Control variable

Order parameters

Optimal control

Backward dynamics

$$-\dot{\hat{\mathbf{Q}}}^\top = \hat{\mathbf{Q}}^\top \nabla_{\mathbf{Q}} f(\mathbf{Q},\mathbf{u})$$

Costate variables

Pontryagin's maximum principle

Variatonal approach
Cost functional:

$$\mathscr{F}[\mathbf{Q}, \hat{\mathbf{Q}}, \mathbf{u}] = \langle \, \epsilon_g(\alpha_F) \, \rangle + \int_0^{\alpha_F} \mathrm{d}\alpha \; \hat{\mathbf{Q}}(\alpha) \cdot \left[-\dot{\mathbf{Q}}(\alpha) + f\left(\mathbf{Q}(\alpha), \mathbf{u}(\alpha)\right) \right]$$

Forward learning dynamics

$$\dot{\mathbf{Q}} = f(\mathbf{Q}, \mathbf{u})$$

Order parameters

Control variable

Optimal control

Backward dynamics

$$-\dot{\hat{\mathbf{Q}}}^\top = \hat{\mathbf{Q}}^\top \nabla_{\mathbf{Q}} f(\mathbf{Q}, \mathbf{u})$$

Costate variables

Optimality condition

$$\mathbf{u}^*(\alpha) = \mathrm{argmin}_{\mathbf{u}(\alpha)} \hat{\mathbf{Q}}(\alpha)^\top f\left(\mathbf{Q}(\alpha), \mathbf{u}(\alpha)\right)$$

Part I: *Dropout Regularization*

FM, F. Mignacco, *arXiv:2505.07792*

F. Mignacco, **FM**, *arXiv:2507.07907*

Dropout regularization

Journal of Machine Learning Research 15 (2014) 1929-1958 Submitted 11/13; Published 6/14

Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava NITISH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Alex Krizhevsky KRIZ@CS.TORONTO.EDU
Ilya Sutskever ILYA@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU

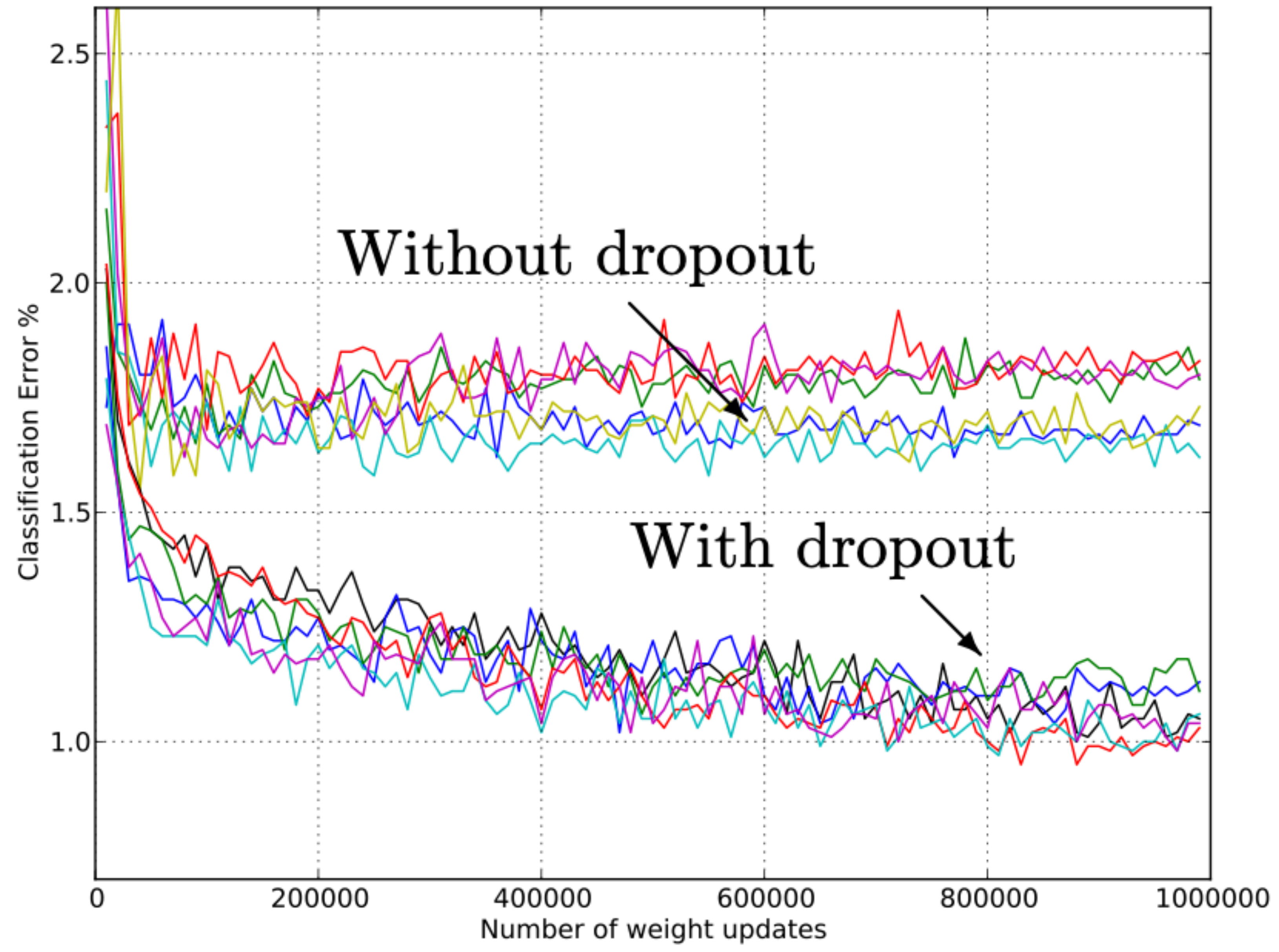
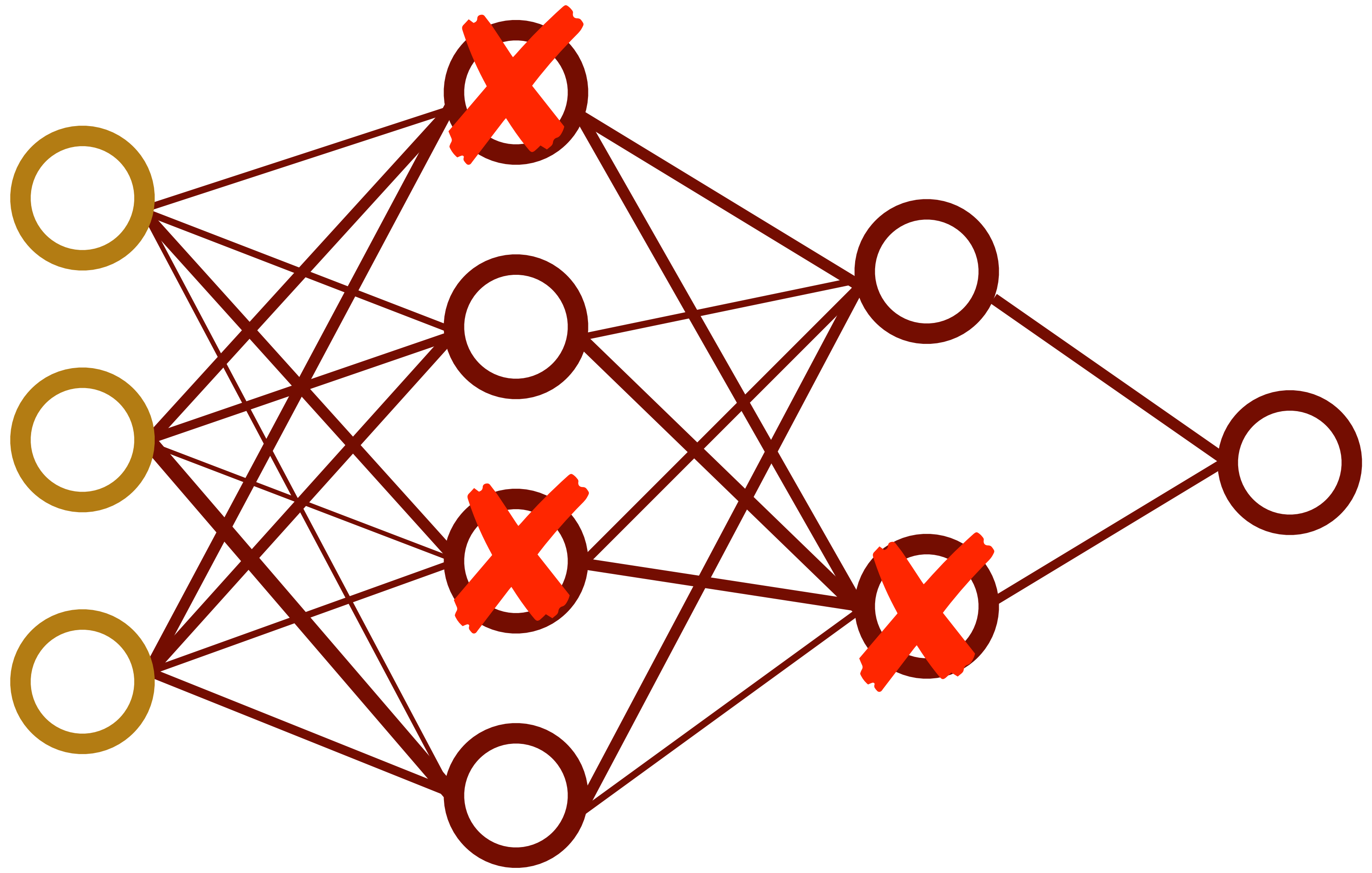


Figure 4: Test error for different architectures with and without dropout. The networks have 2 to 4 hidden layers each with 1024 to 2048 units.

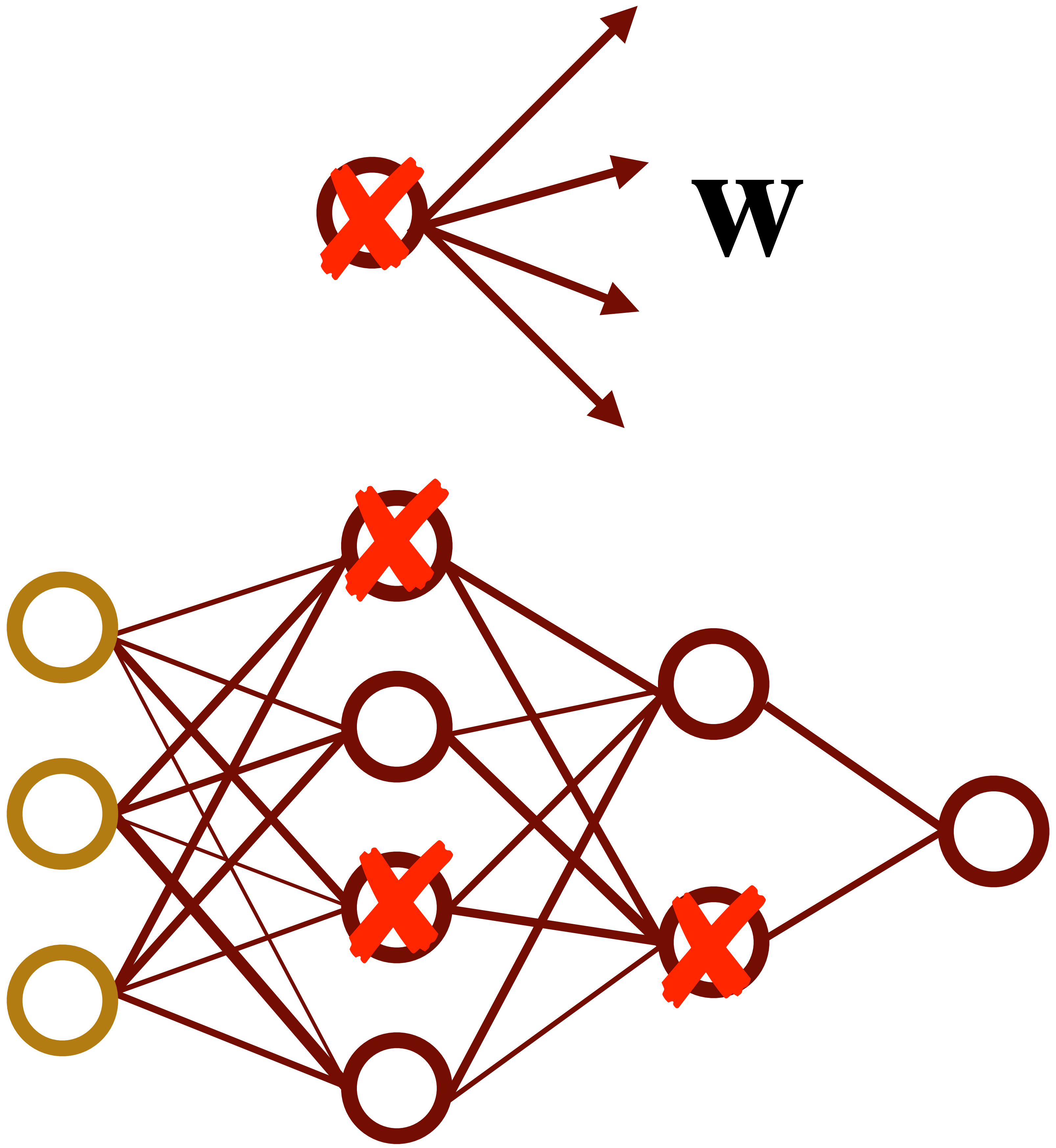


= remove node with probability $1 - p$

Dropout regularization

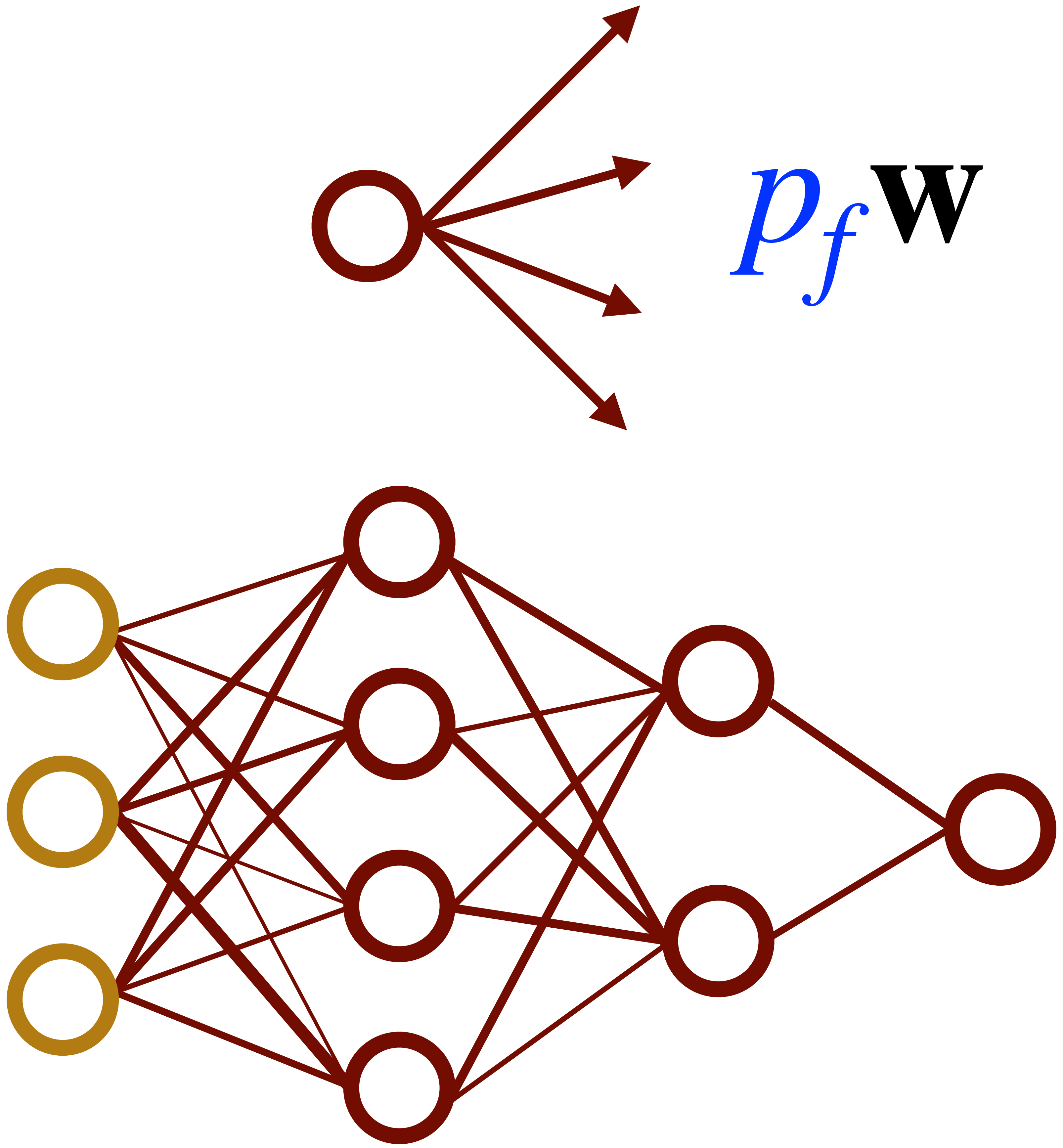
Train

= remove node (with probability $1-p$)



Test

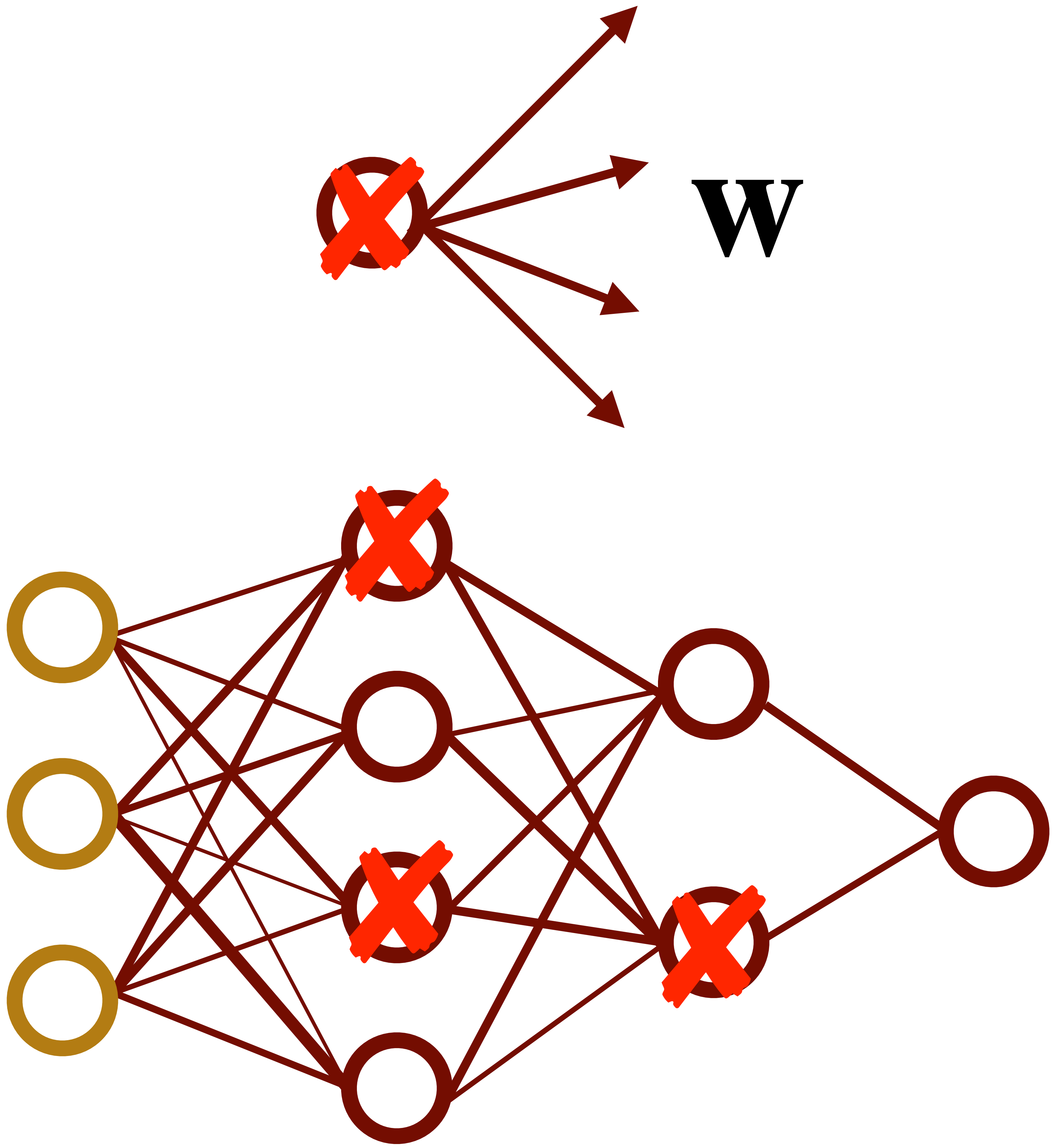
Always present but *rescaled* weights



What is the *optimal* dropout probability?

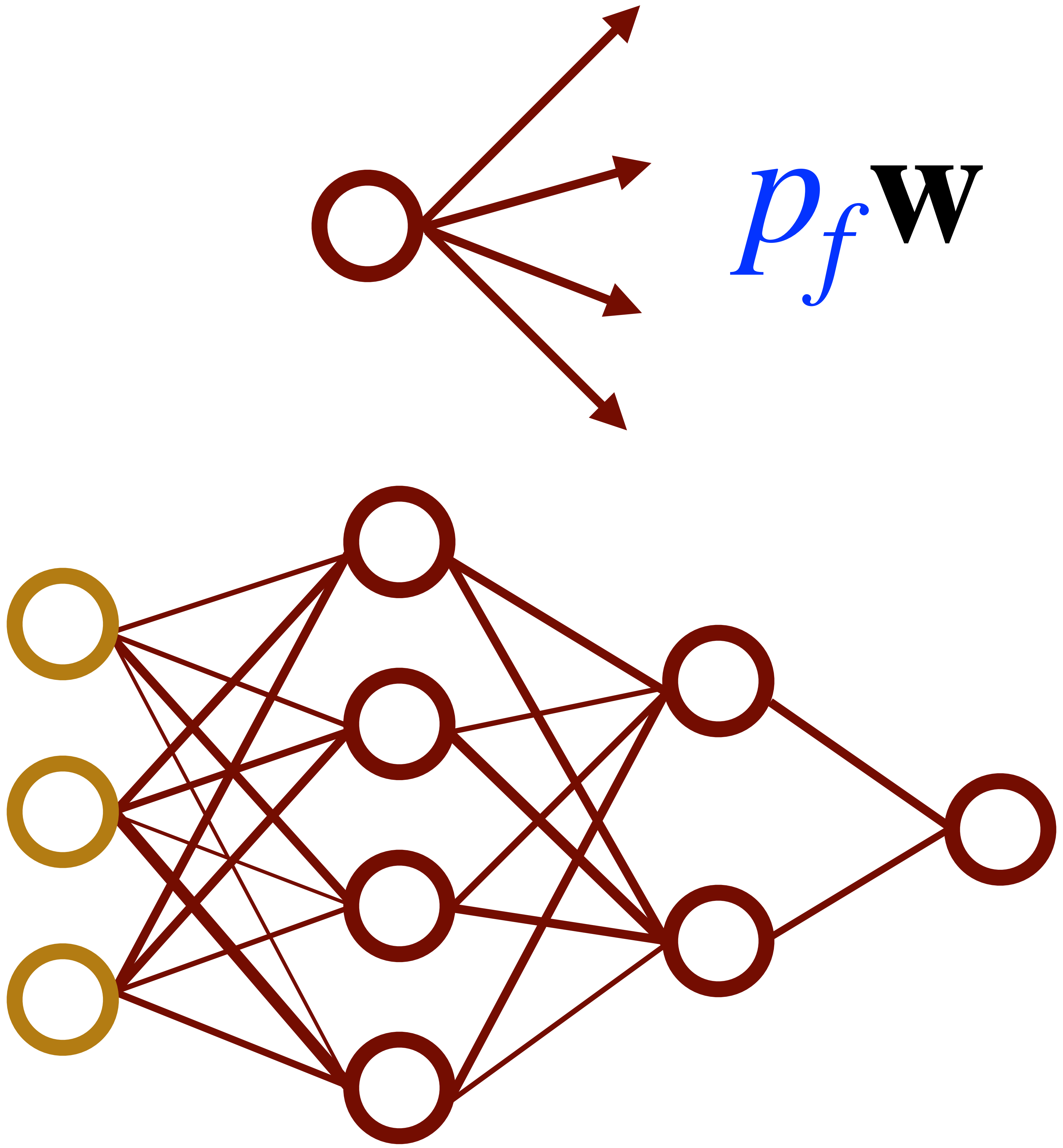
Train

= remove node (with probability $1-p$)



Test

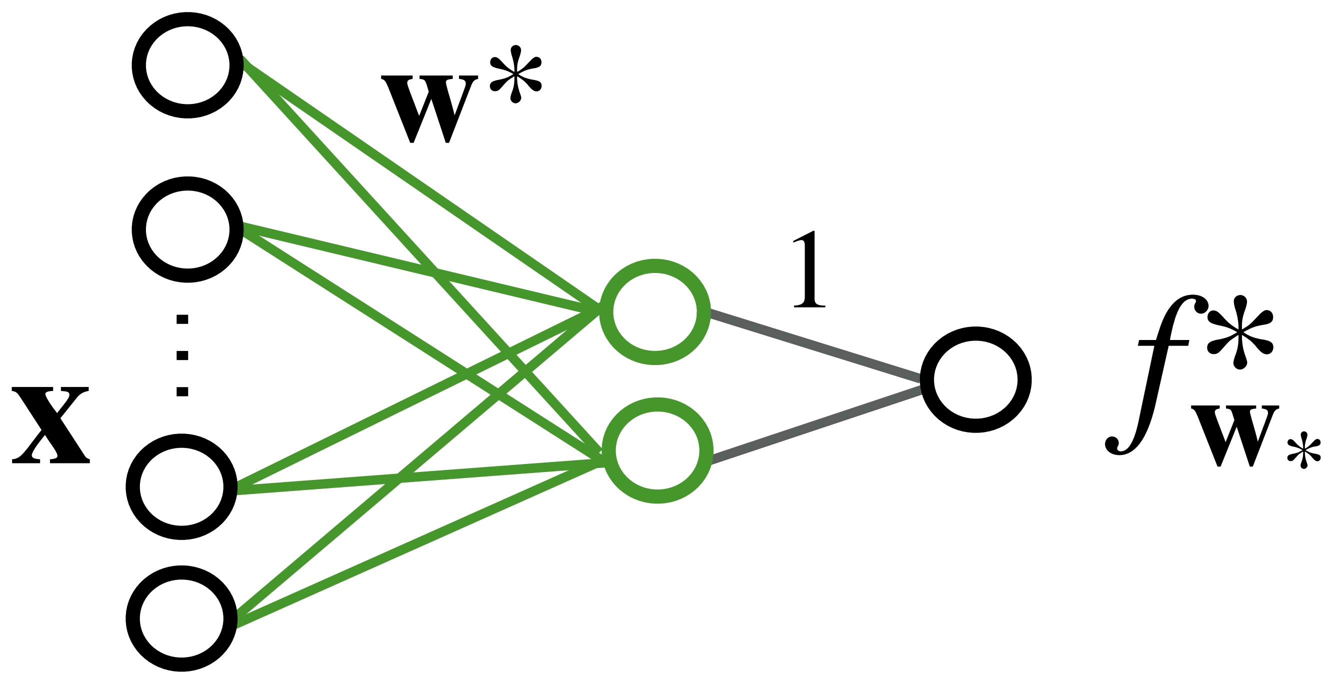
Always present but *rescaled* weights



A teacher-student model of dropout

$\mathbf{x} \sim \mathcal{N}(0, I_N)$

Teacher



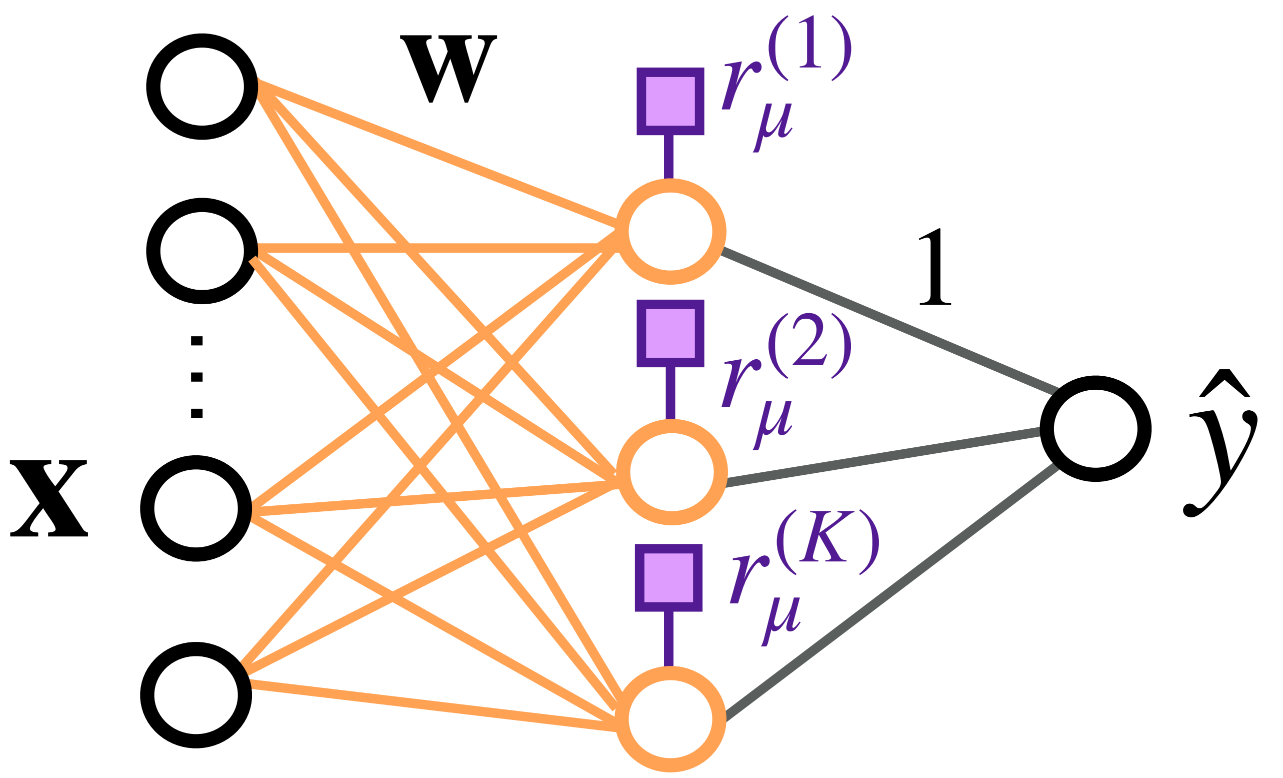
M hidden nodes

$$y = f^*_{\mathbf{w}_*}(\mathbf{x}) + \sigma_n\, z$$

Label noise: $z \sim \mathcal{N}(0,1)$

Student

(at training step μ)



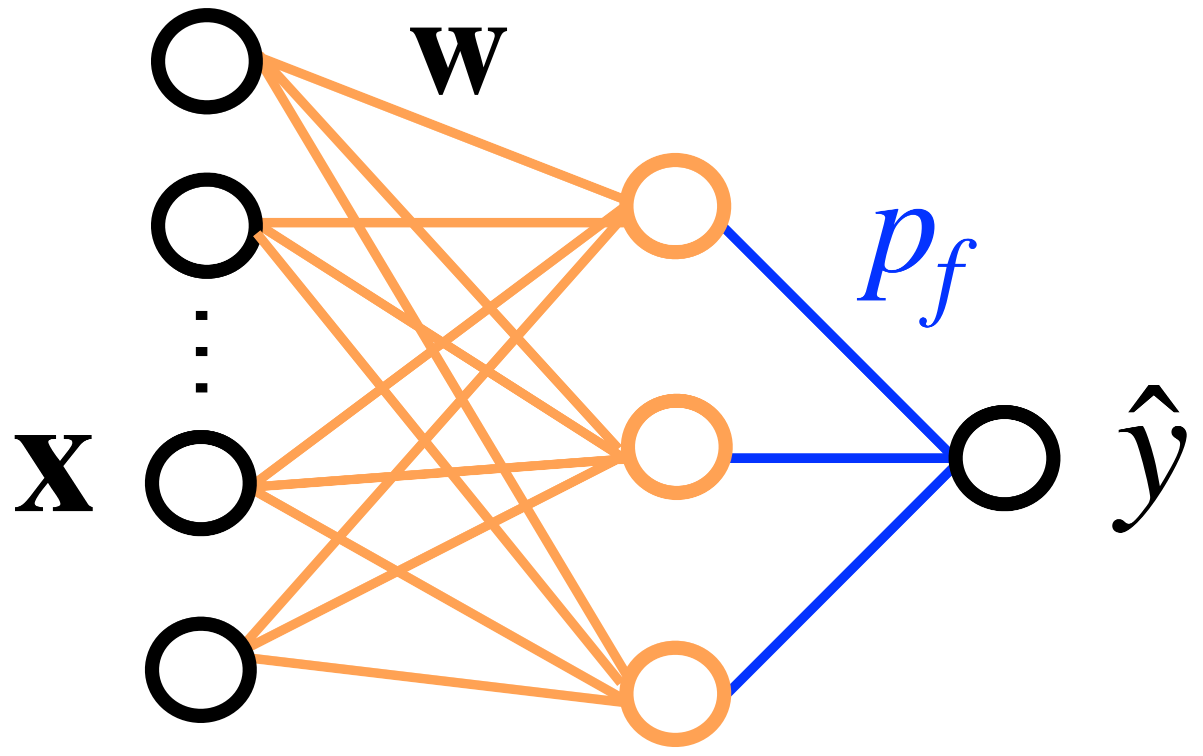
K hidden nodes

Node-activation variables

$r_\mu^{(1)}, r_\mu^{(2)}, \dots, r_\mu^{(K)} \sim$ Bernoulli(p_μ)

Student

(at testing time)



K hidden nodes

Rescaling factor: p_f

FM, F. Mignacco, *arXiv:2505.07792*

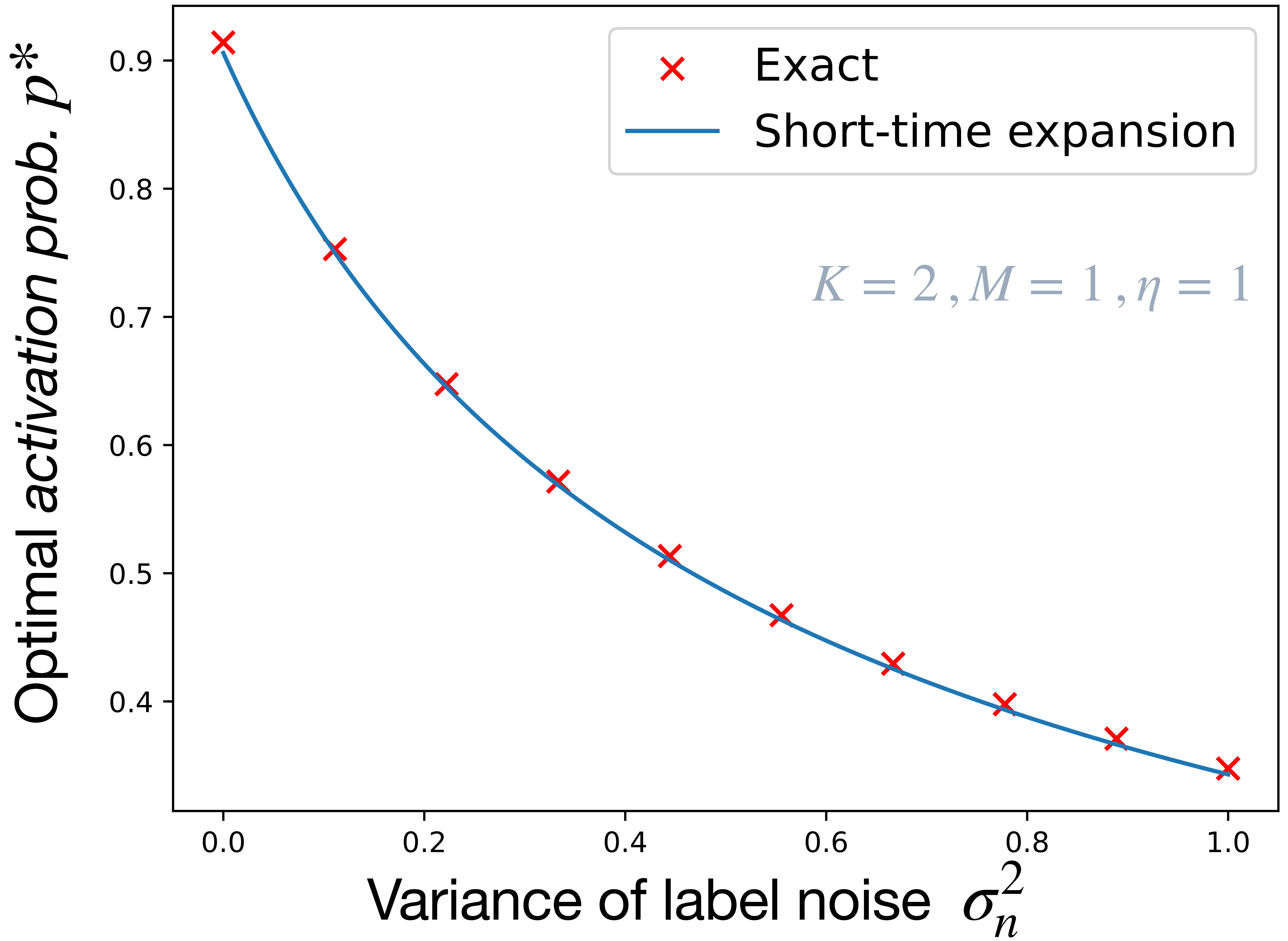
Online learning with constant node-activation prob.

1. The optimal *constant* activation probability depends on label noise, width, and learning rate.

Online learning with constant node-activation prob.

1. The optimal *constant* activation probability depends on label noise, width, and learning rate.

Early-time dynamics: $$p^* = \min\{1; \frac{12M}{3M\eta + 9\eta\sigma_n^2 + \sqrt{3\eta(M+3\sigma_n^2)\left[M(32(K-1)+3\eta)+9\eta\sigma_n^2\right]}}\}$$



FM, F. Mignacco
arXiv:2505.07792

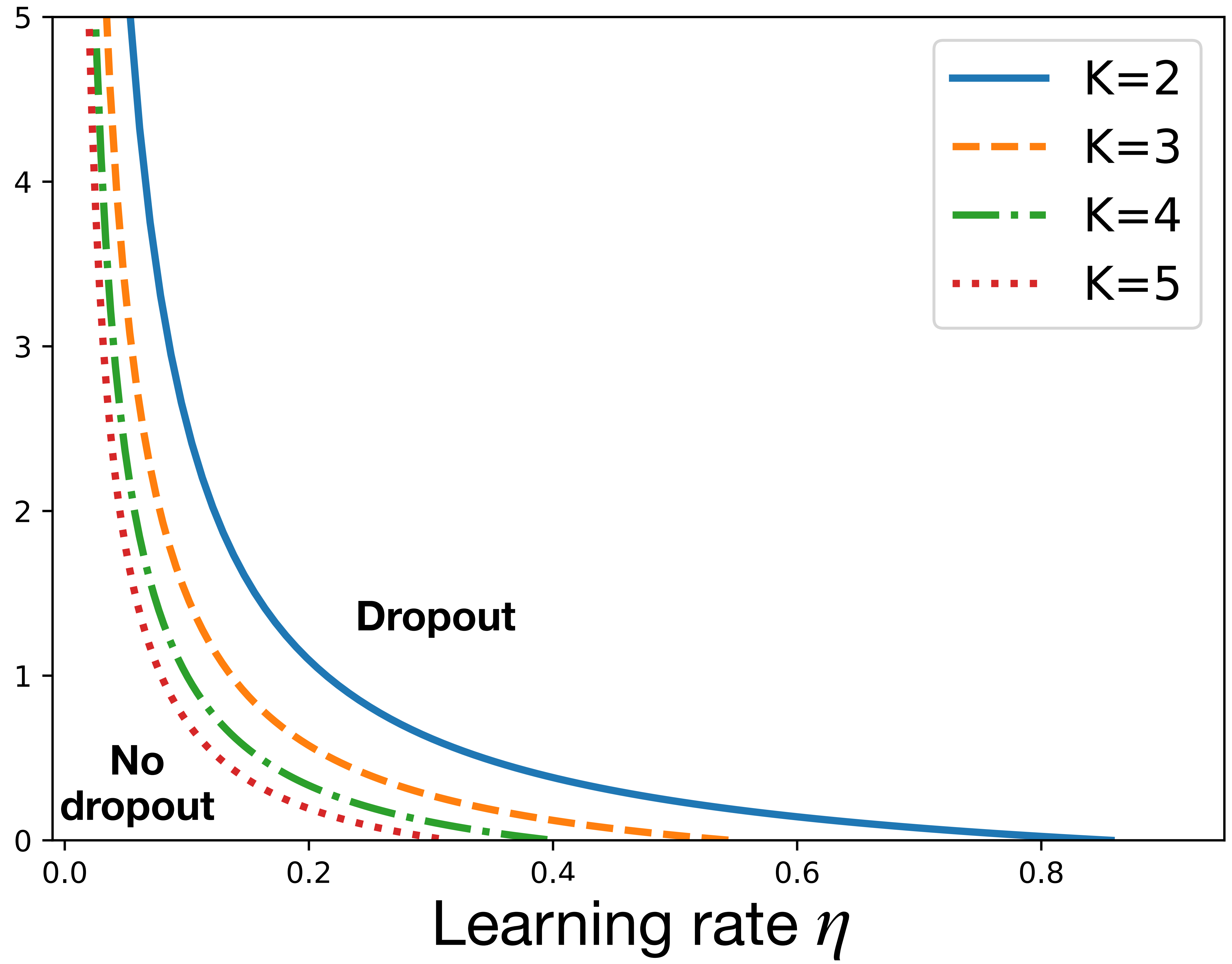
Online learning with constant node-activation prob.

UNIVERSITY OF OXFORD

1. The optimal *constant* activation probability depends on label noise, width, and learning rate.

Early-time dynamics: $$p^* = \min\{1; \frac{12M}{3M\eta + 9\eta\sigma_n^2 + \sqrt{3\eta(M+3\sigma_n^2)\left[M(32(K-1)+3\eta)+9\eta\sigma_n^2\right]}}\}$$

Label noise σ_n/M



Dropout is optimal if:

$$\sigma_n^2 \geq M\left[\frac{2}{(4K-1)\eta} - \frac{1}{3}\right]$$

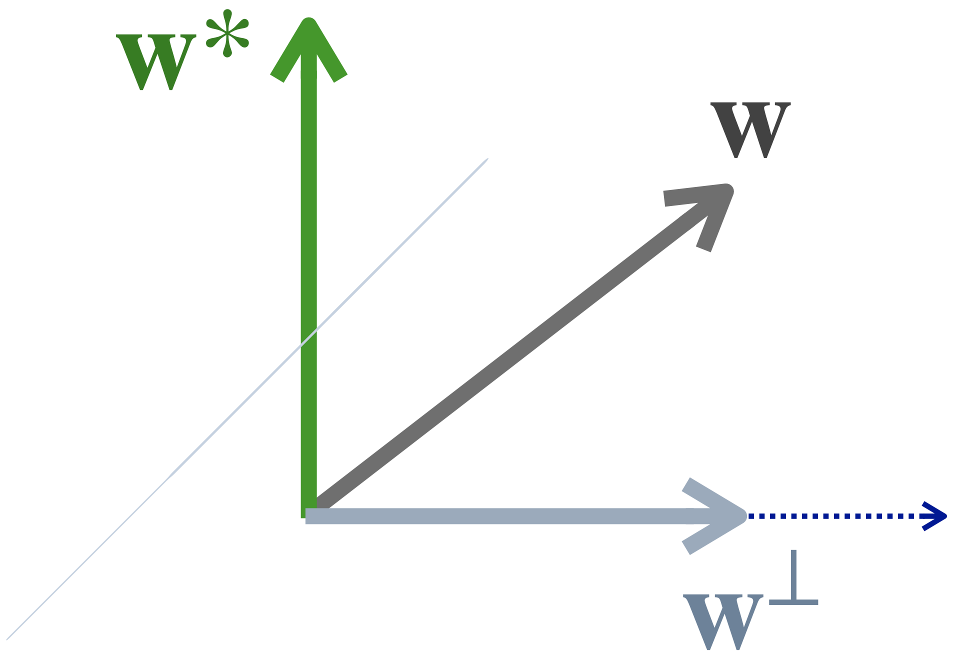
FM, F. Mignacco
arXiv:2505.07792

Online learning with constant node-activation prob.

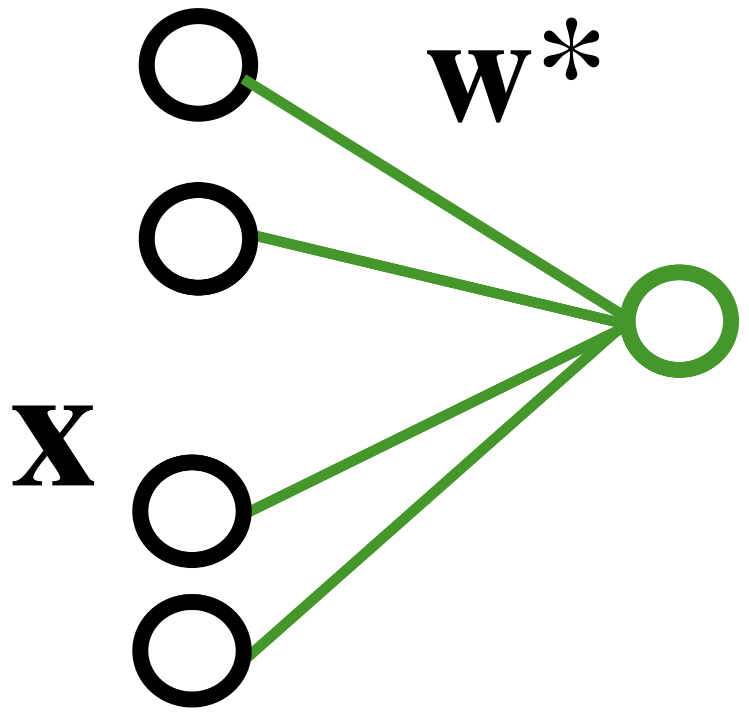
1. The optimal *constant* activation probability depends on label noise, width, and learning rate.
2. Dropout effectively decorrelates the hidden units of the student network.

e.g., take $M = 1$ and $K = 2$

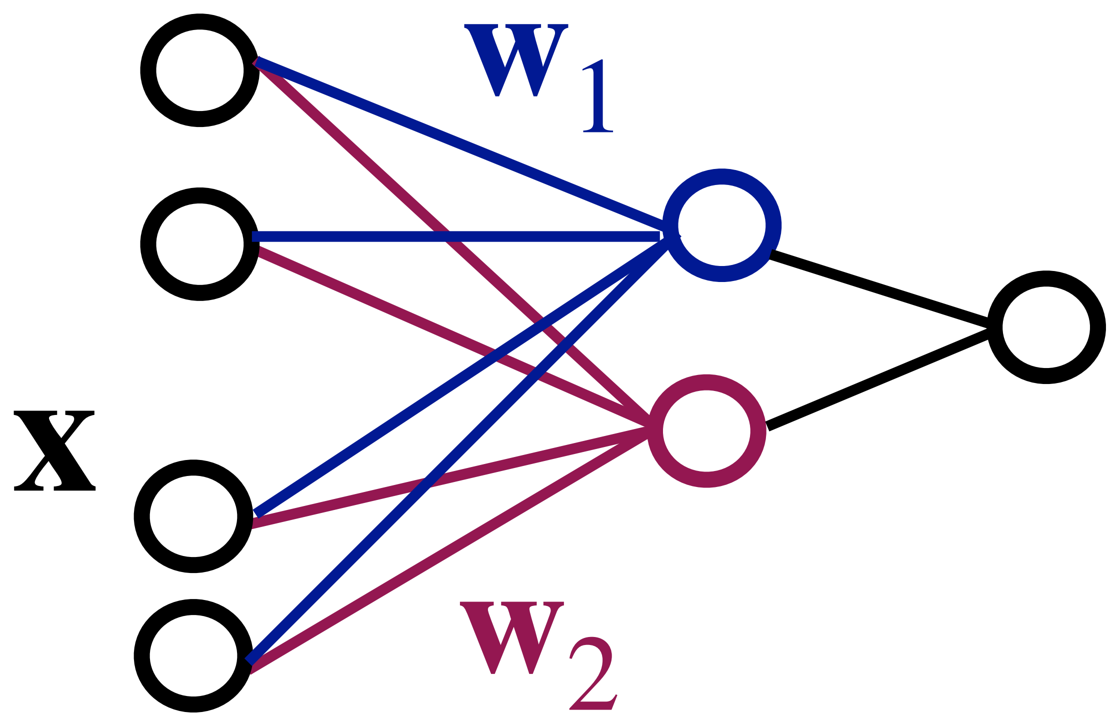
$\|\mathbf{w}^*\| = 1$



Teacher



Student



$$\mathbf{w}_1 = R_1 \mathbf{w}^* + \mathbf{w}_1^{\perp}$$

$$\mathbf{w}_2 = R_2 \mathbf{w}^* + \mathbf{w}_2^{\perp}$$

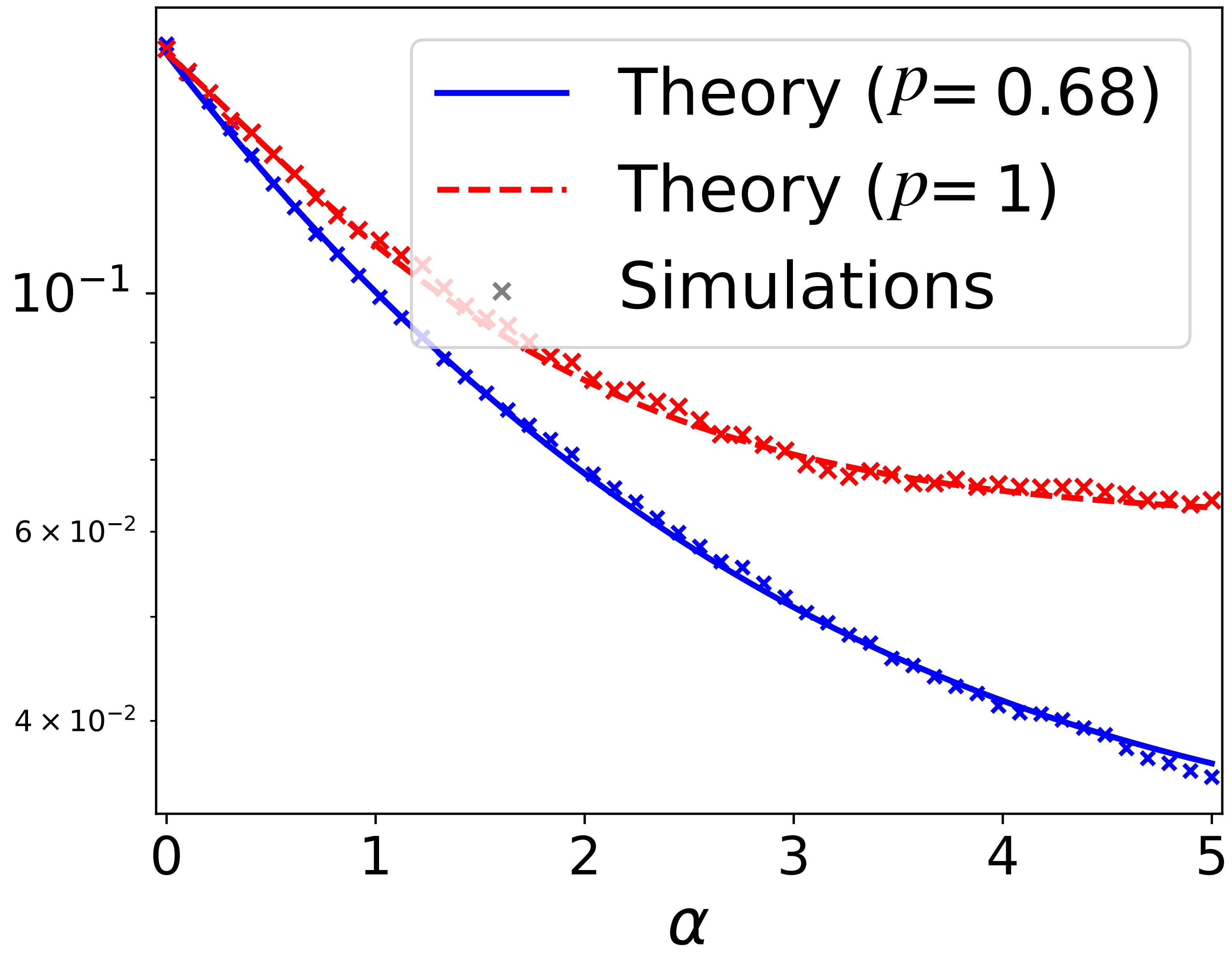
Noise overfitting

Online learning with constant node-activation prob.

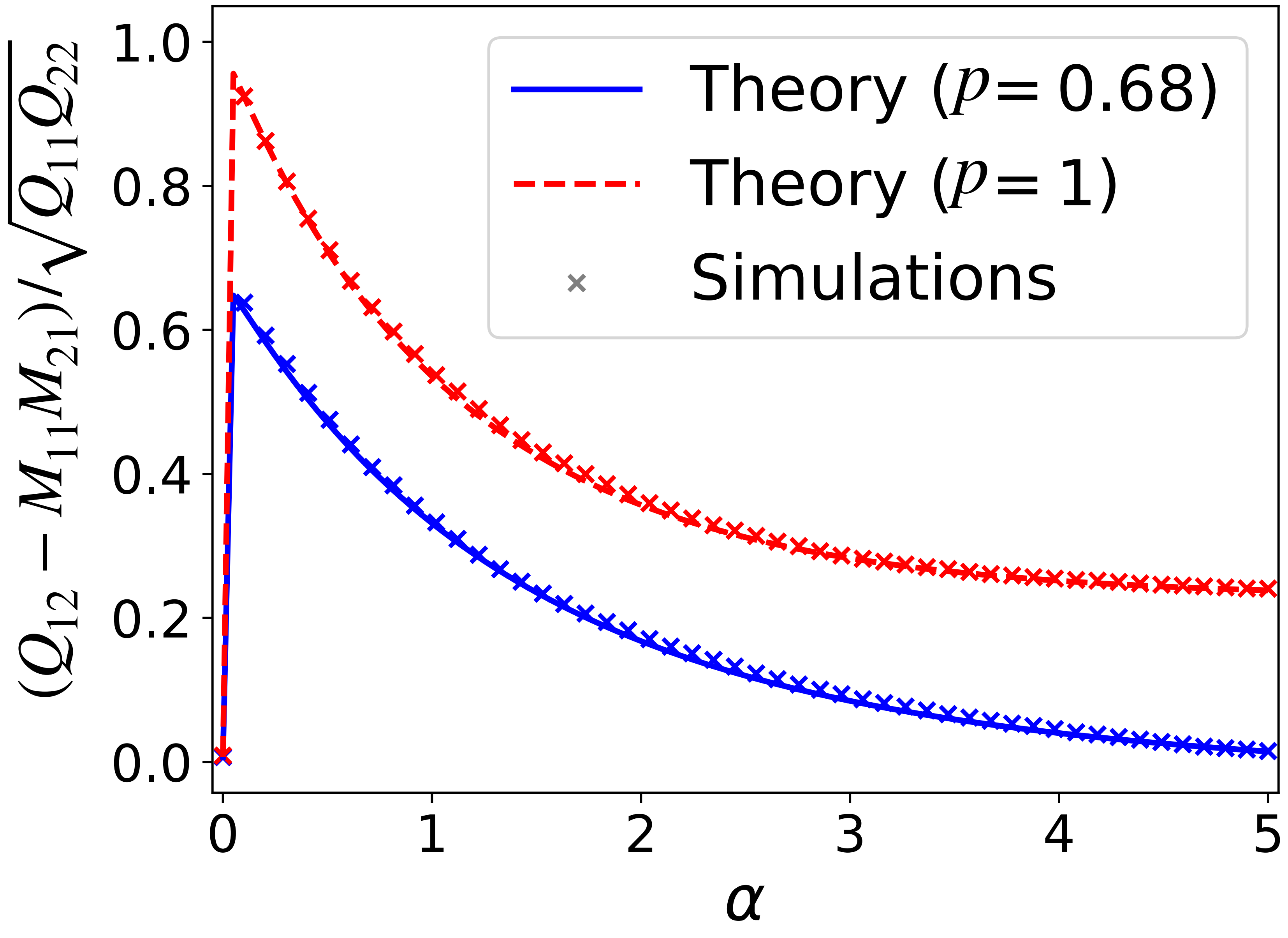
1. The optimal *constant* activation probability depends on label noise, width, and learning rate.
2. Dropout effectively decorrelates the hidden units of the student network.

e.g., take $M = 1$ and $K = 2$

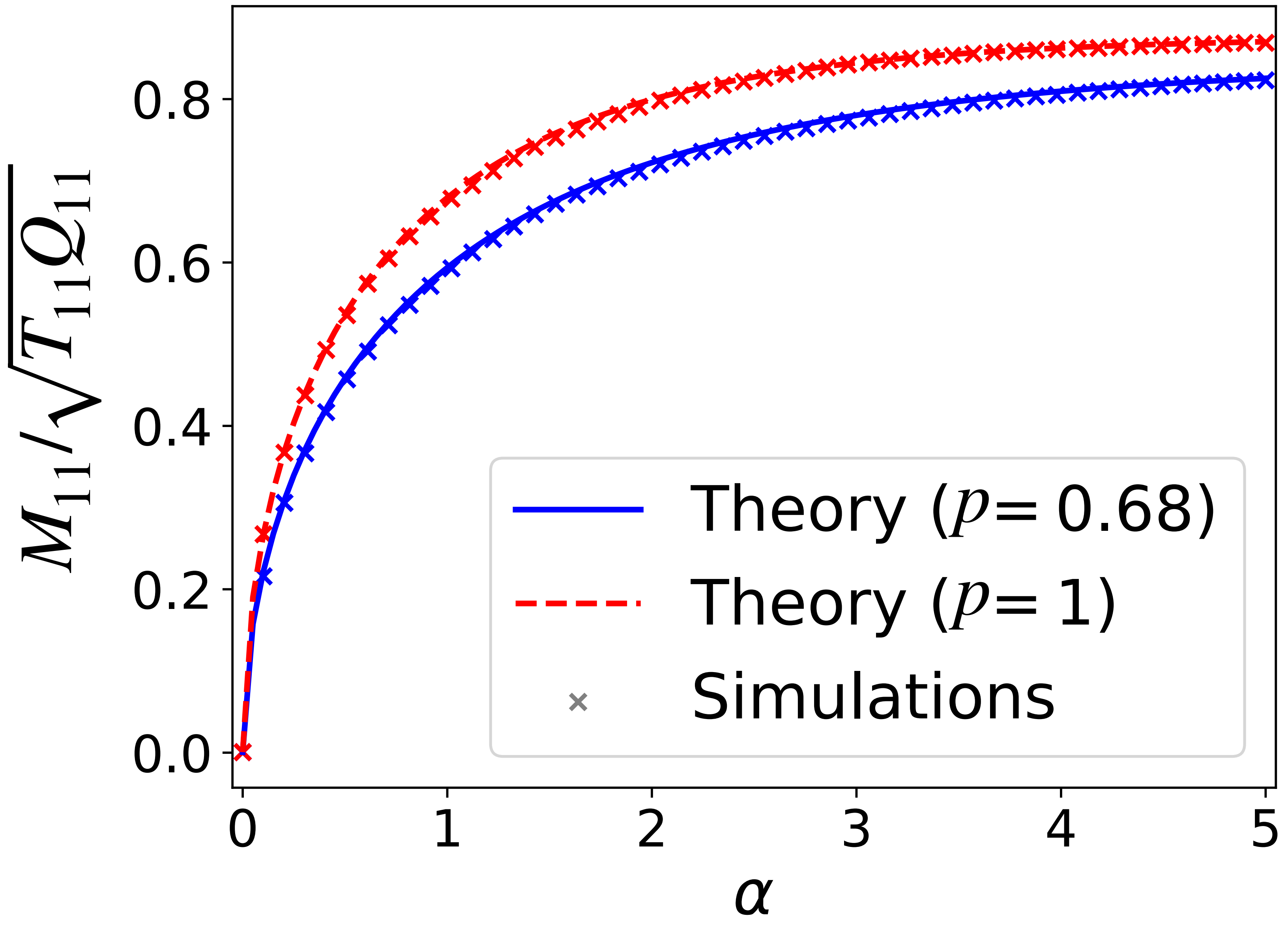
Generalization error



Negative co-adaptations



Cosine similarity with teacher



Adaptive dropout probability

ANNEALED DROPOUT TRAINING OF DEEP NETWORKS

Steven J. Rennie, Vaibhava Goel, and Samuel Thomas

IBM Thomas J. Watson Research Center
NY, USA
{sjrennie, vgoel, sthomas}@us.ibm.com

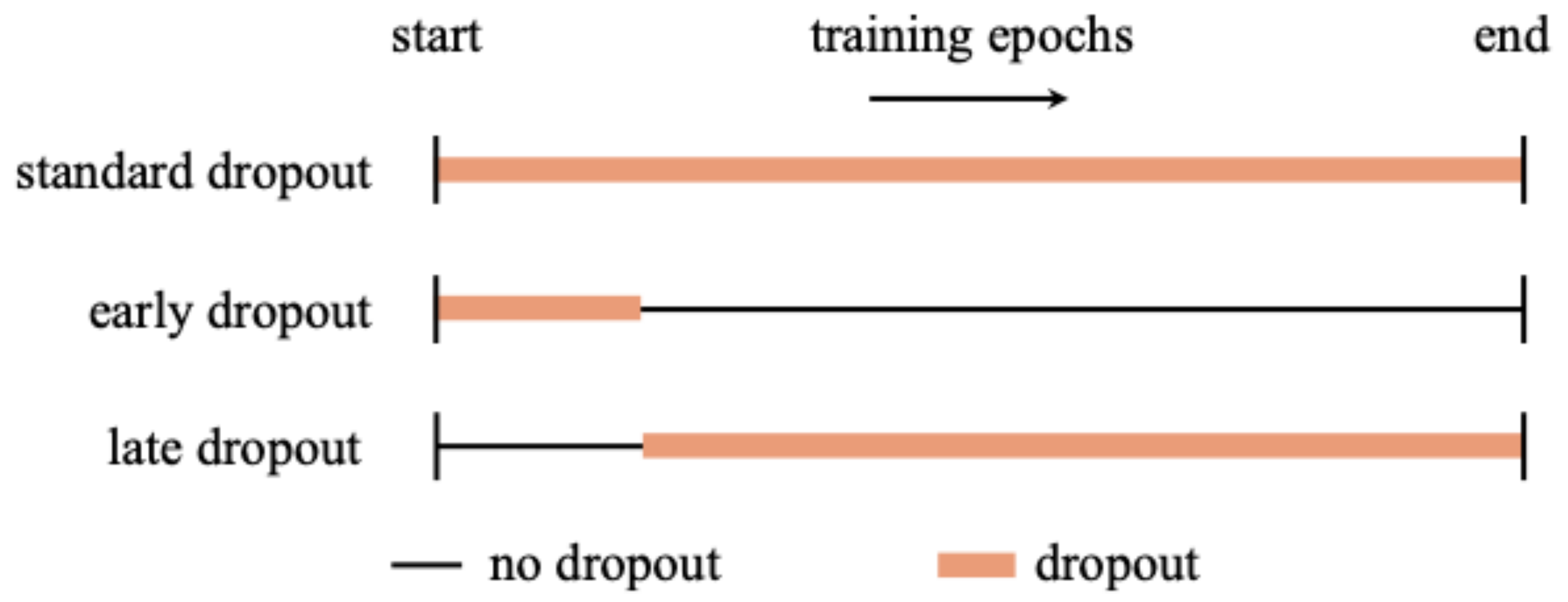
Curriculum Dropout

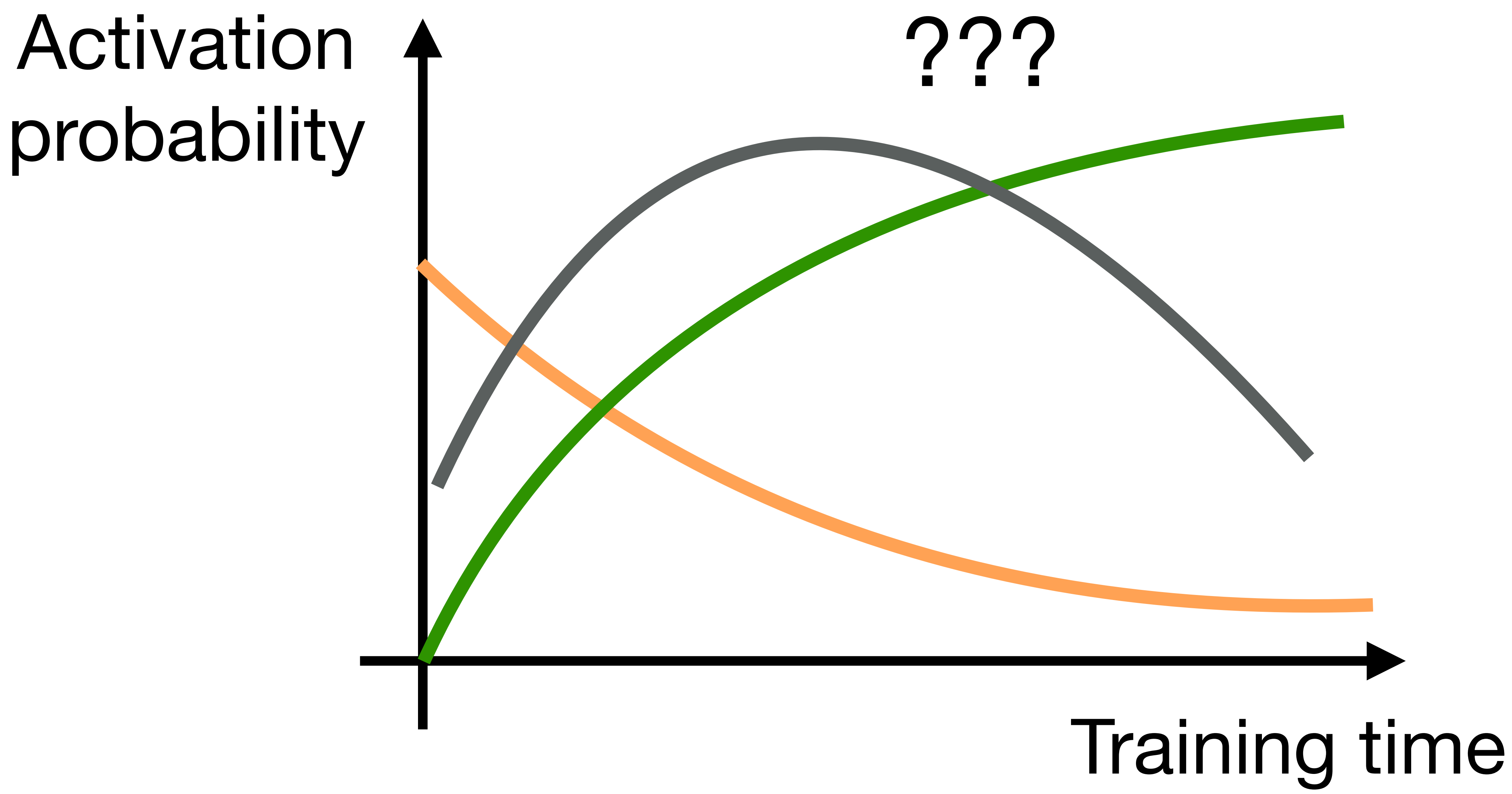
Pietro Morerio[1], Jacopo Cavazza[1,2], Riccardo Volpi[1,2], René Vidal[3] and Vittorio Murino[1,4]

[1]Pattern Analysis & Computer Vision (PAVIS) – Istituto Italiano di Tecnologia – *Genova, 16163, Italy*
[2]Electrical, Electronics and Telecommunication Engineering and Naval Architecture Department (DITEN) – Università degli Studi di Genova – *Genova, 16145, Italy*
[3]Department of Biomedial Engineering – Johns Hopkins University – *Baltimore, MD 21218, USA*
[4]Computer Science Department – Università di Verona – *Verona, 37134, Italy*
{pietro.morerio,jacopo.cavazza,riccardo.volpi,vittorio.murino}@iit.it, rvidal@cis.jhu.edu

Dropout Reduces Underfitting

Zhuang Liu [*1] **Zhiqiu Xu** [*2] **Joseph Jin** [2] **Zhiqiang Shen** [3] **Trevor Darrell** [2]

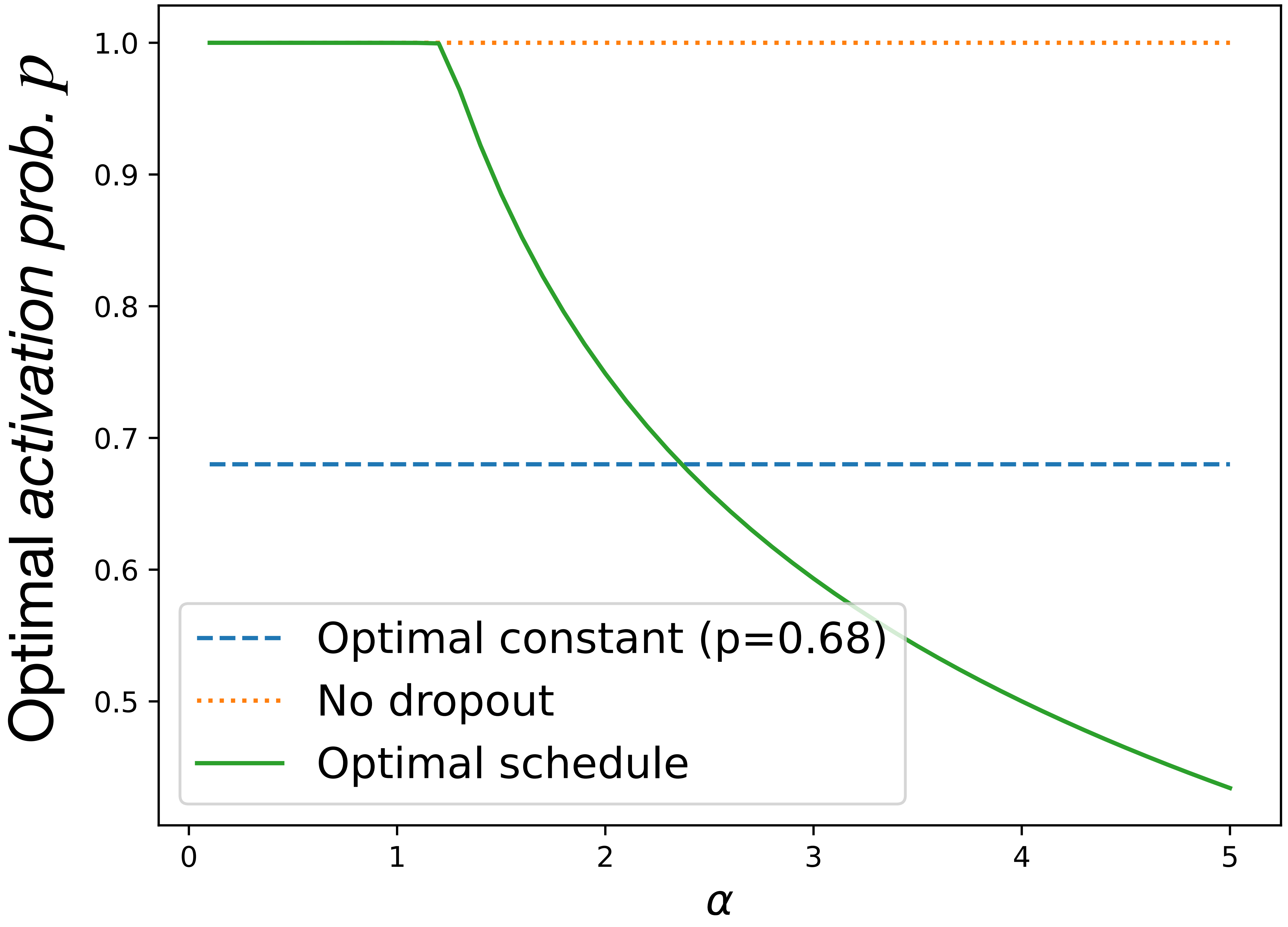




Optimal dropout schedules

Teacher-student setting $(K = 2\,, M = 1)$

1. The optimal schedule monotonically decreases the node-activation probability.



$\eta = 1\,, \alpha_F = 5\,, \sigma_n = 0.3\,, Q_0 = 0$

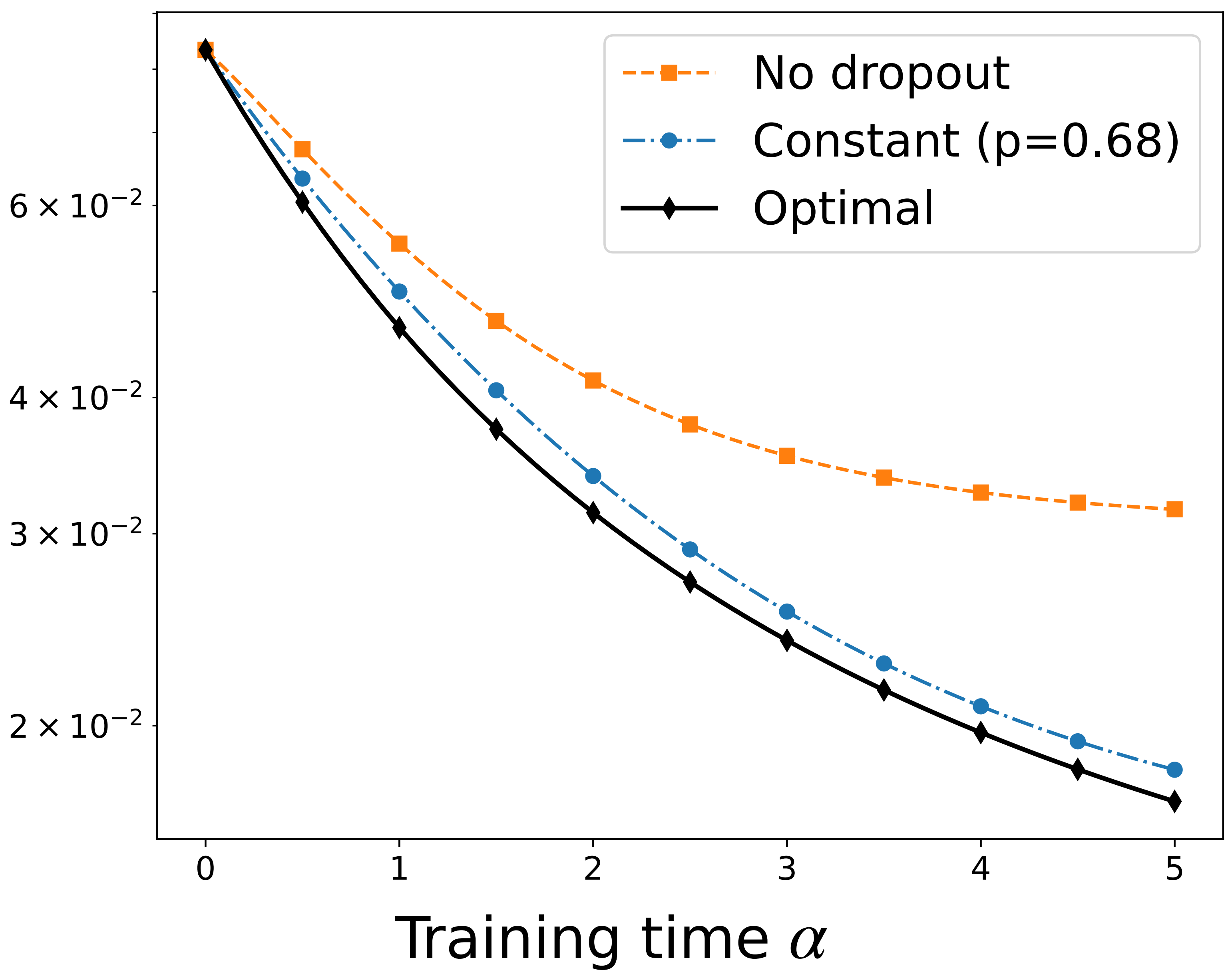
F. Mignacco, FM *arXiv:2507.07907*

Optimal dropout schedules

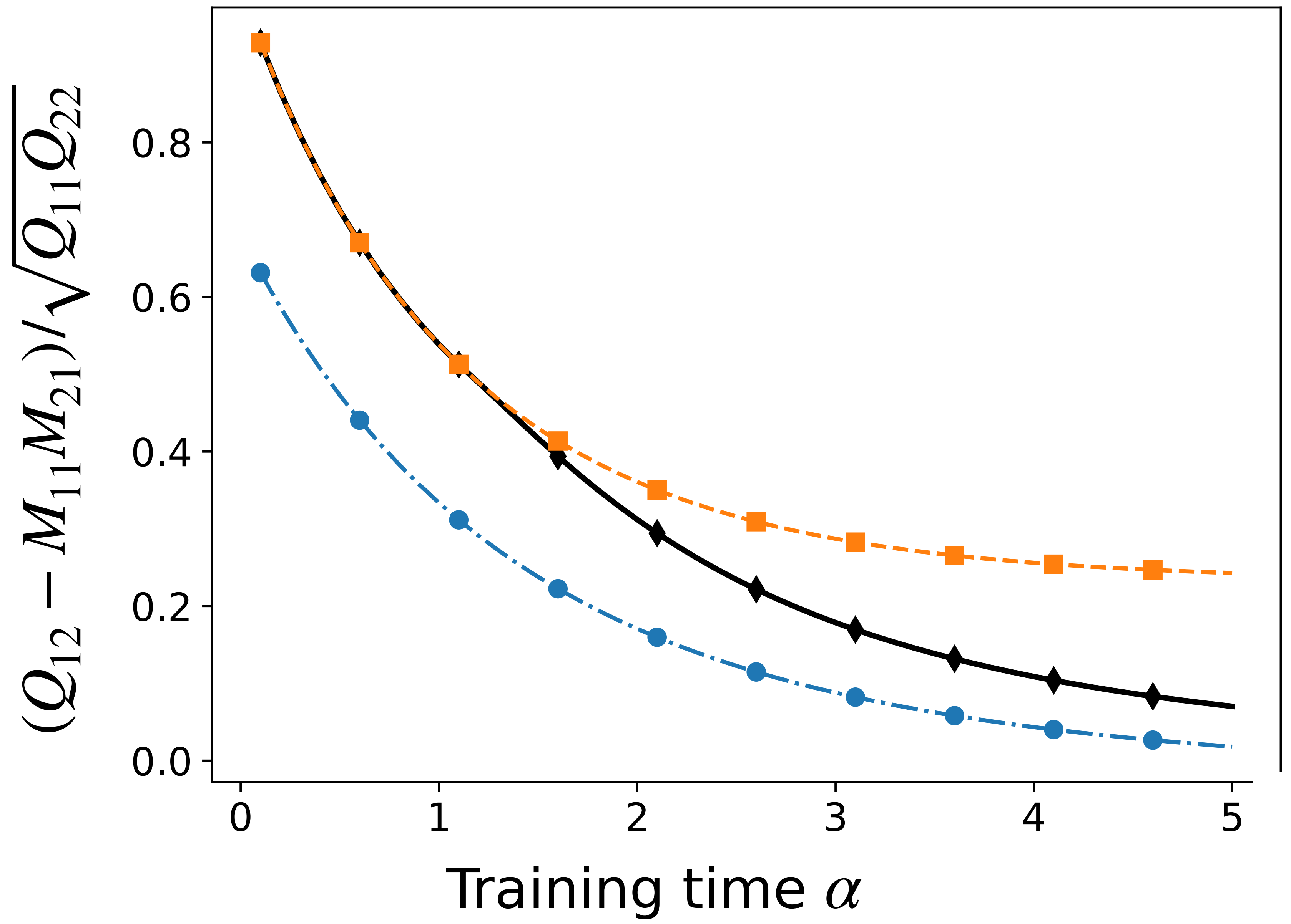
Teacher-student setting $(K = 2\,, M = 1)$

1. The optimal schedule monotonically decreases the node-activation probability.
2. Adaptive dropout achieves the optimal balance between node decorrelation and signal recovery.

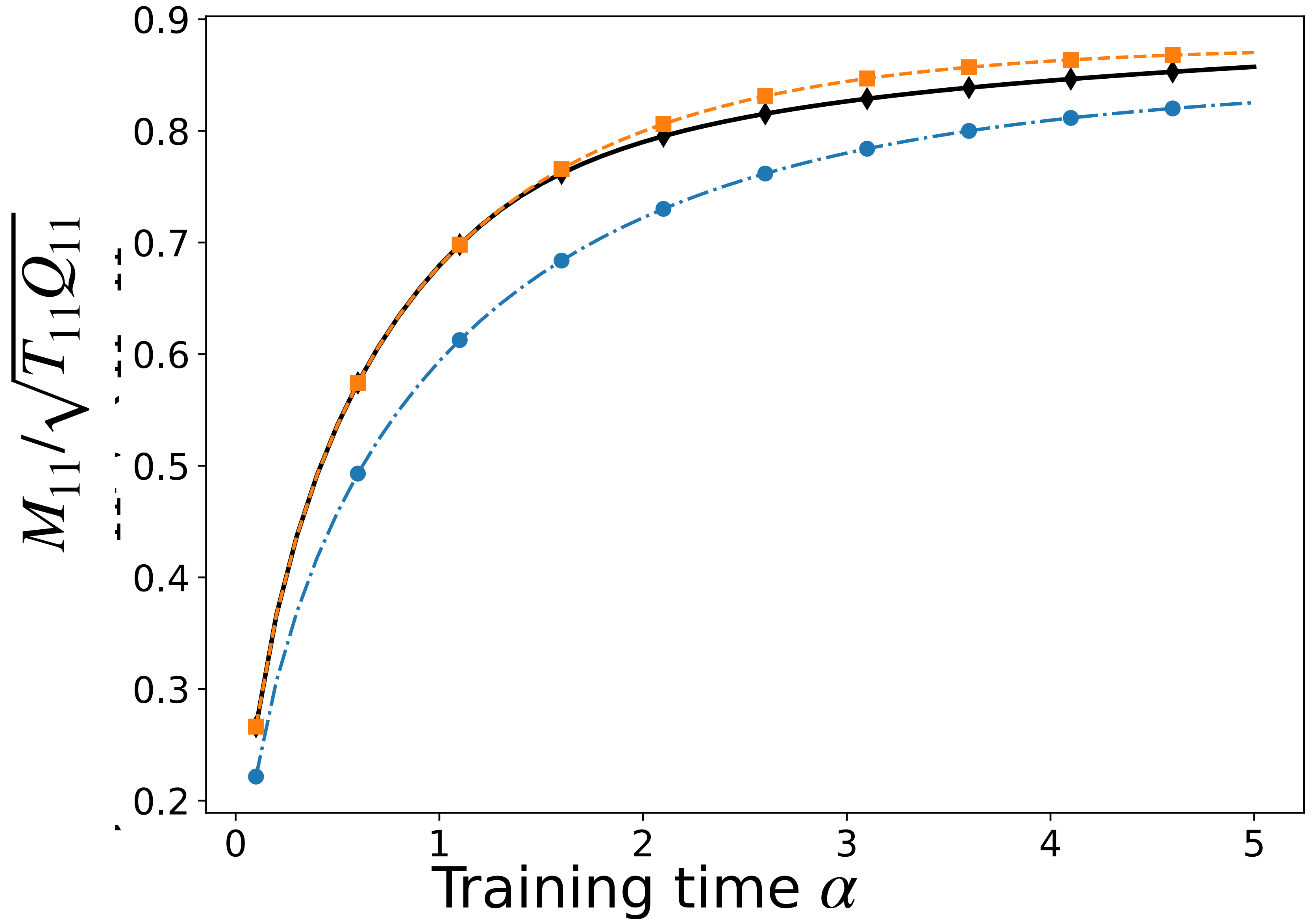
Generalization error



Negative co-adaptations



Cosine similarity with teacher



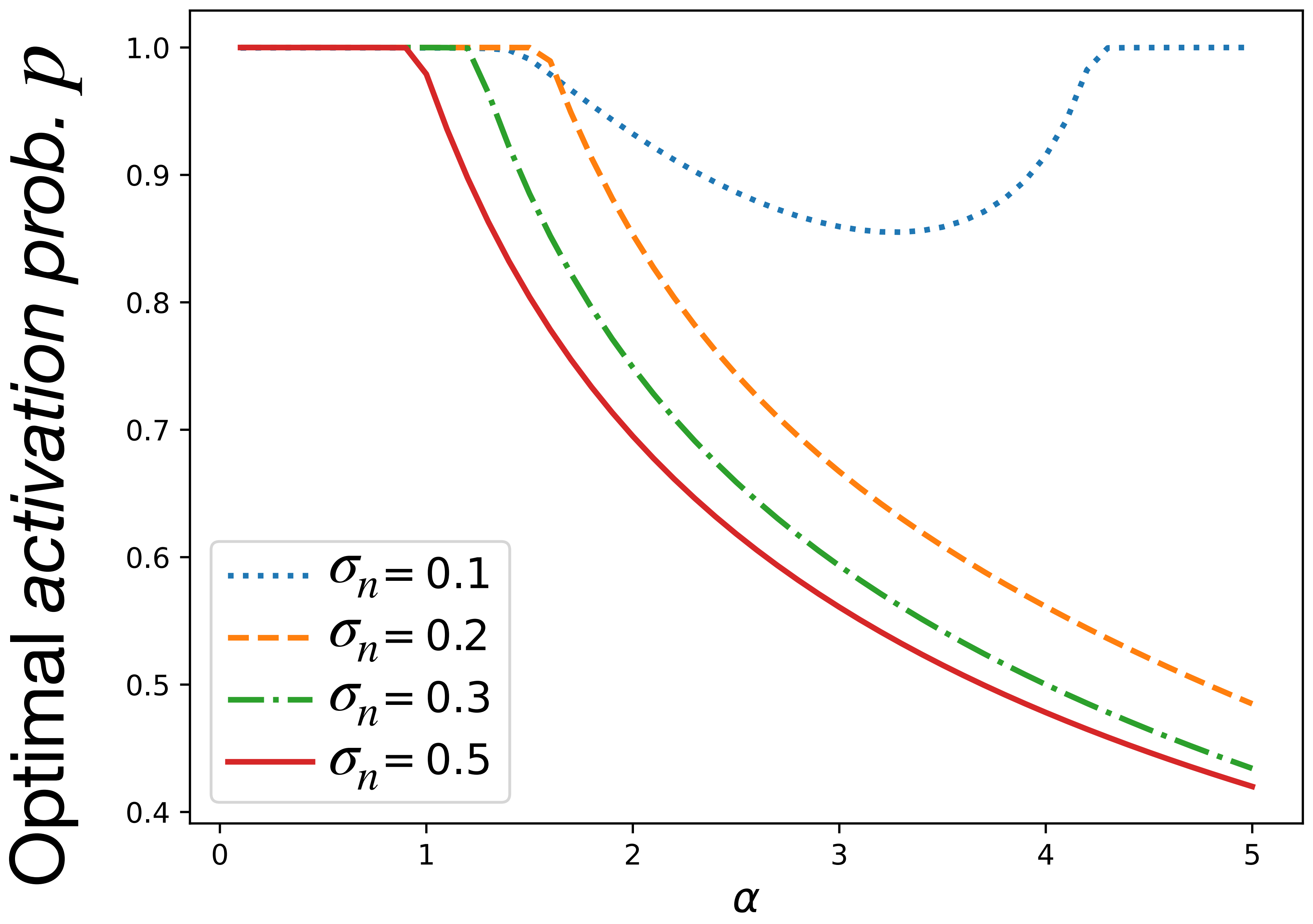
F. Mignacco, FM *arXiv:2507.07907*

$\eta = 1\,, \alpha_F = 5\,, \sigma_n = 0.3\,, Q_0 = 0$

Optimal dropout schedules

Teacher-student setting $(K = 2\,, M = 1)$

1. The optimal schedule monotonically decreases the node-activation probability.
2. Adaptive dropout achieves the optimal balance between node decorrelation and signal recovery.
3. The dropout schedule adapts to the noise level in the task.



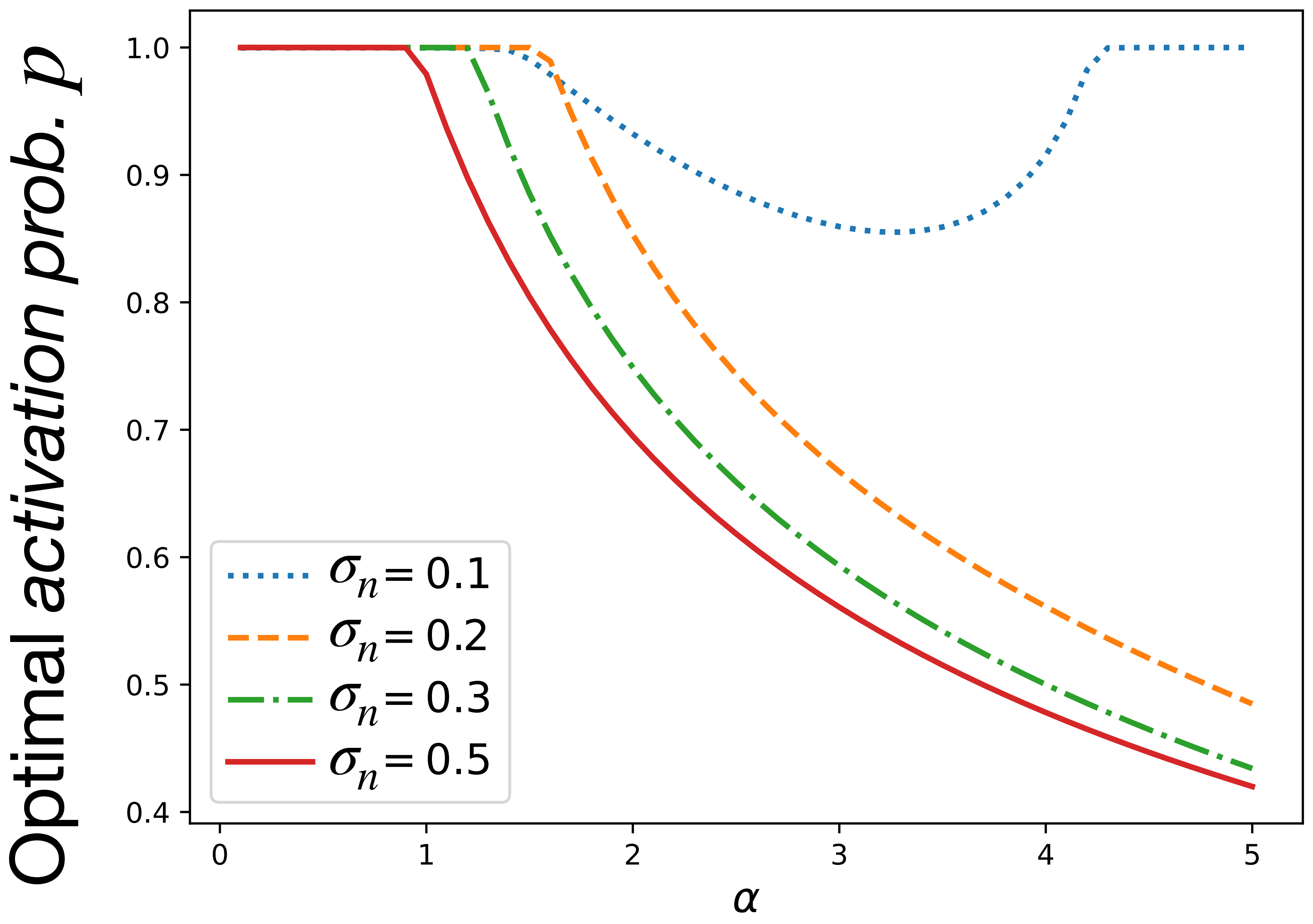
$\eta = 1\,, \alpha_F = 5\,, Q_0 = 0$

F. Mignacco, FM *arXiv:2507.07907*

Optimal dropout schedules

Teacher-student setting $(K = 2\,, M = 1)$

1. The optimal schedule monotonically decreases the node-activation rate.
2. Adaptive dropout achieves the optimal balance between node decorrelation and signal recovery.
3. The dropout schedule adapts to the noise level in the task.



$\eta = 1\,, \alpha_F = 5\,, Q_0 = 0$

F. Mignacco, FM *arXiv:2507.07907*

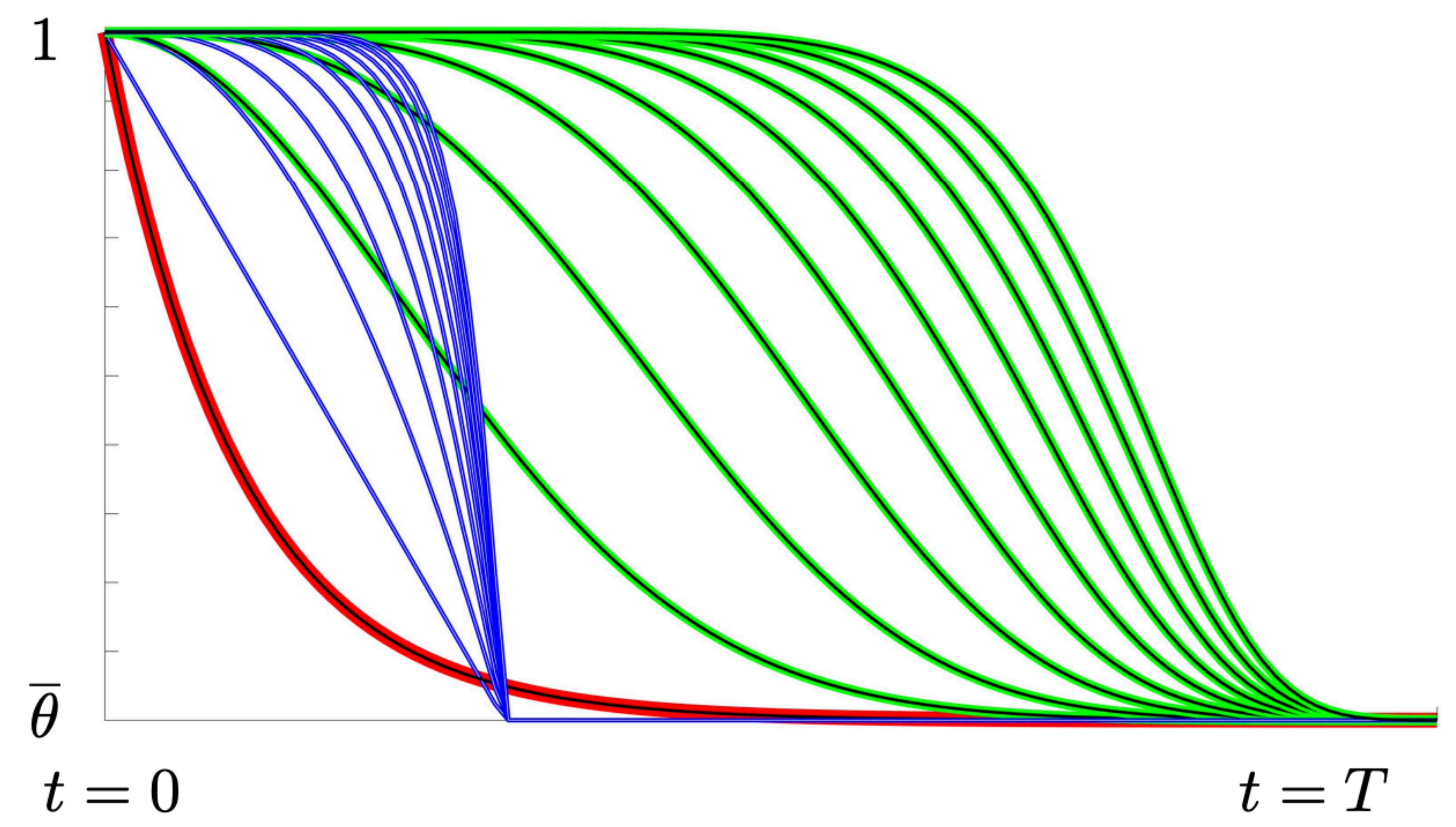


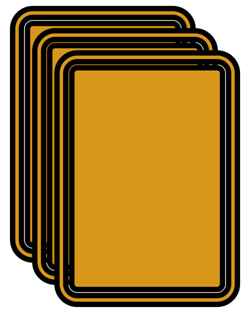
Figure 2. Curriculum functions. Eq. (1) (red), polynomial (blue) and exponential (green).

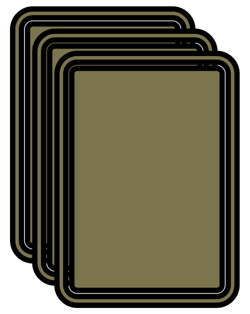
Morerio, P., Cavazza, J., Volpi, R., Vidal, R., & Murino, V. (2017). *Curriculum dropout*. In Proceedings of the IEEE international conference on computer vision (pp. 3544-3552)

Part II: *Continual learning*

FM, S. Sarao Mannelli, F. Mignacco, *ICLR* 2025

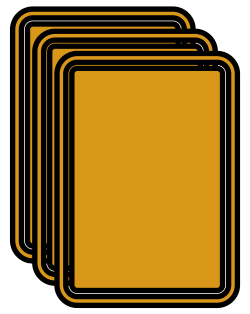
Continual learning

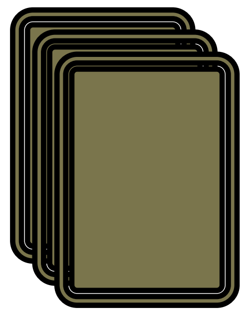
Task 1 

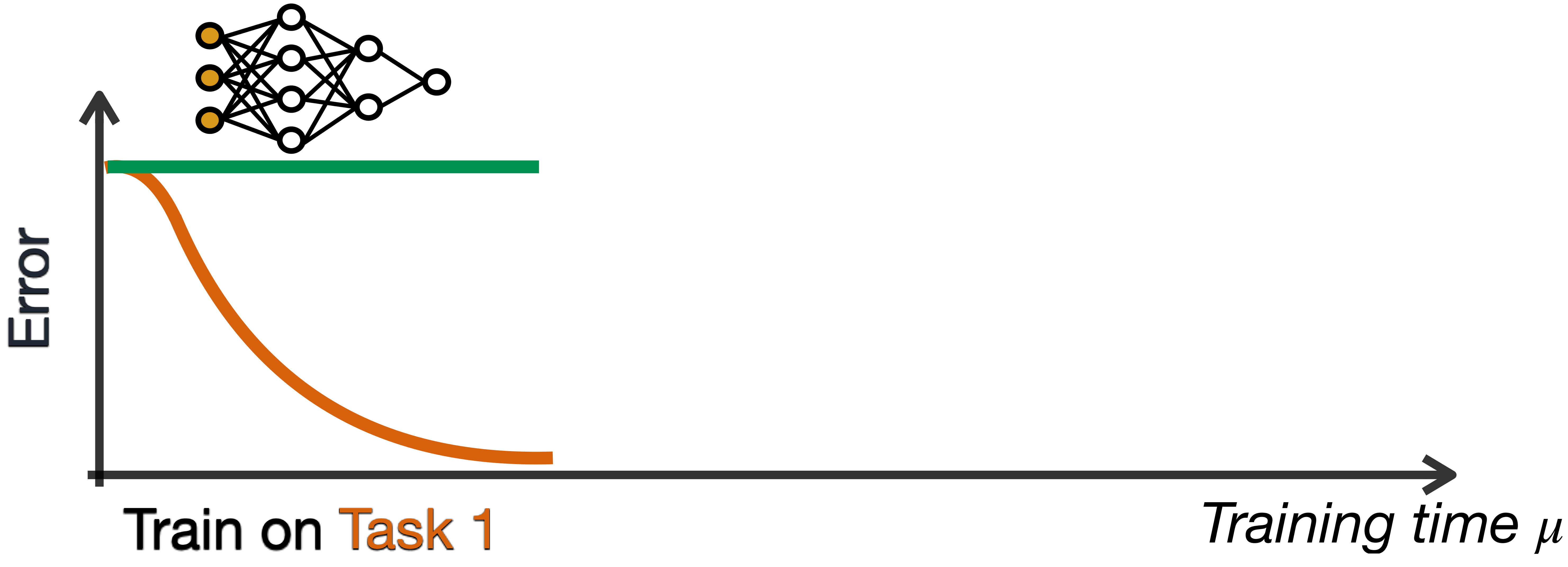
Task 2 

Task protocol u_μ **:**

Continual learning

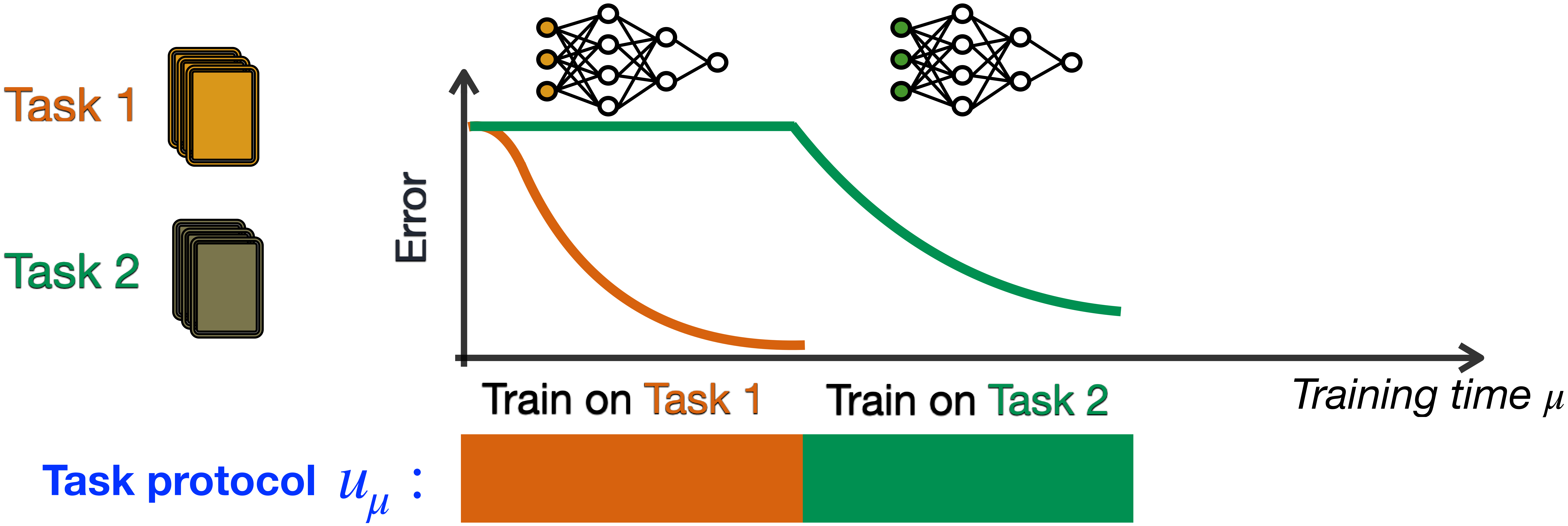
Task 1 

Task 2 

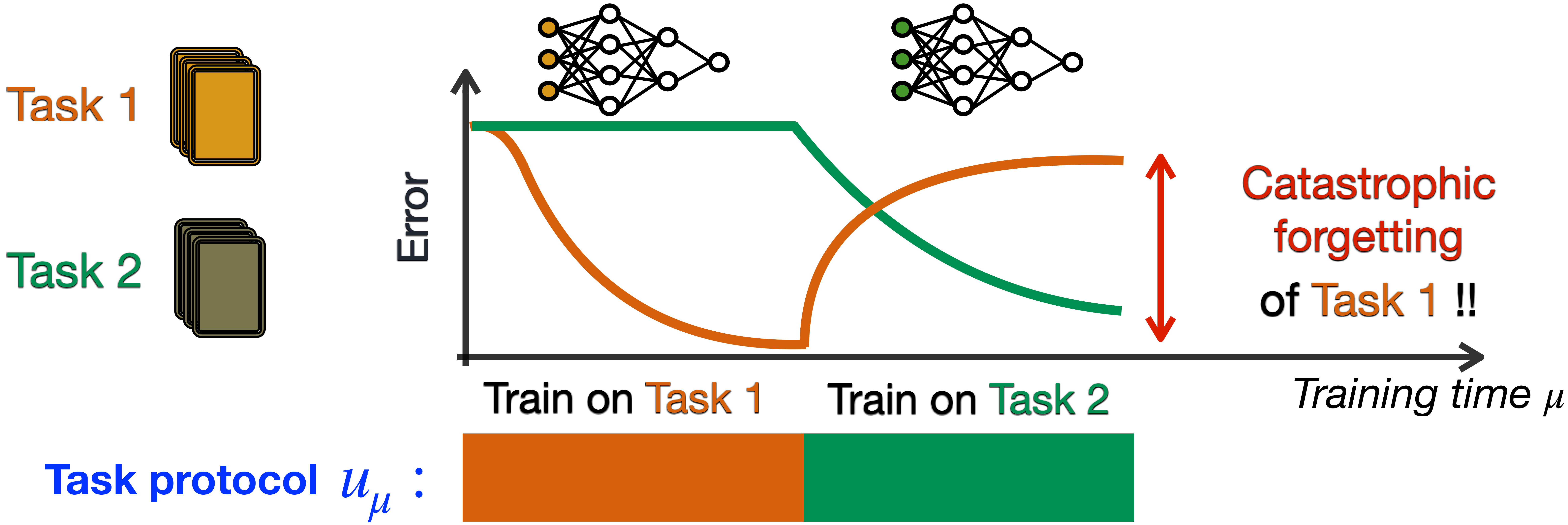


Task protocol u_μ : 

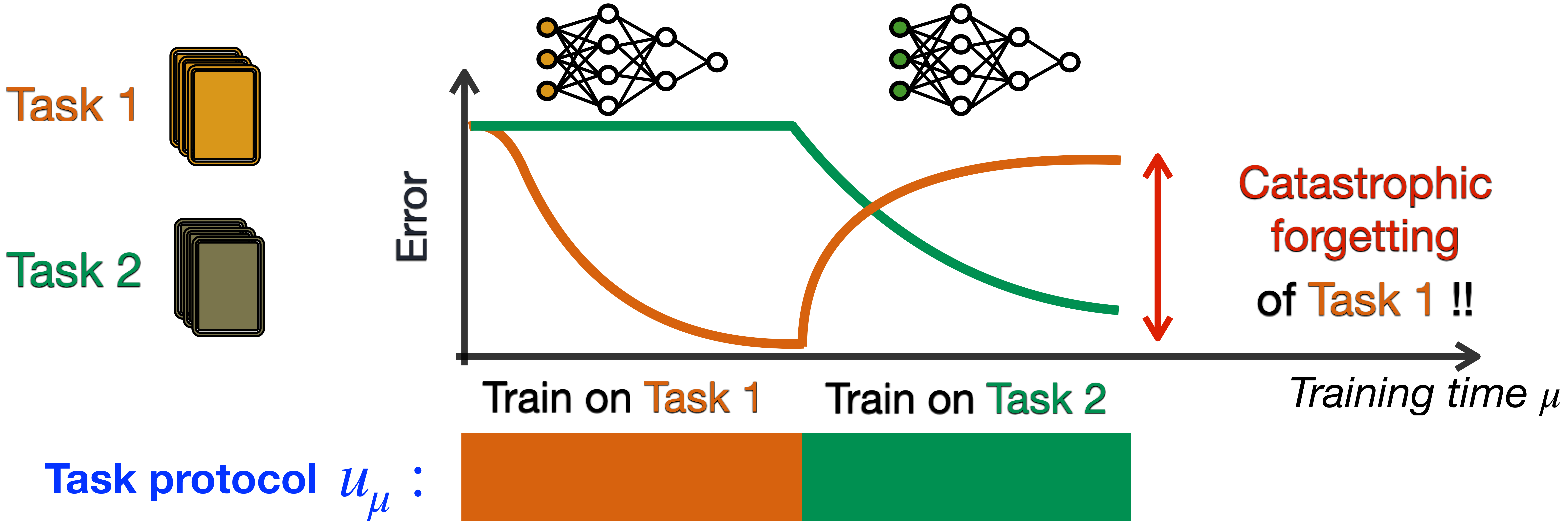
Continual learning



Continual learning



Continual learning

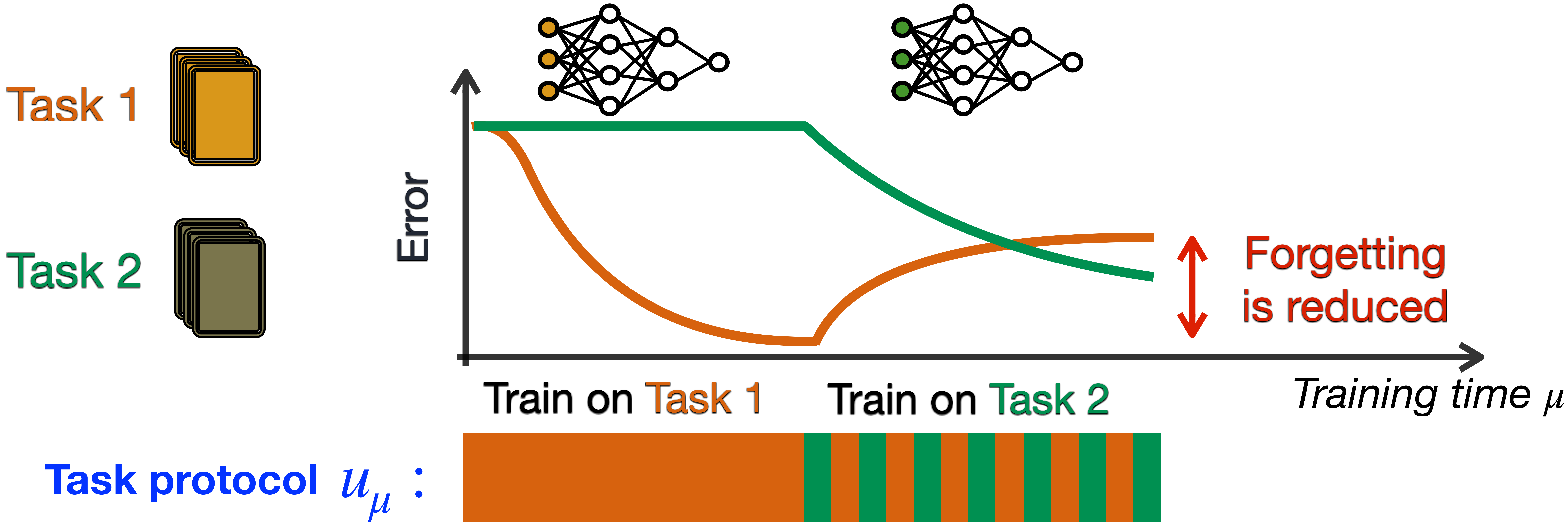


Humans & Animals

ML (empirical)

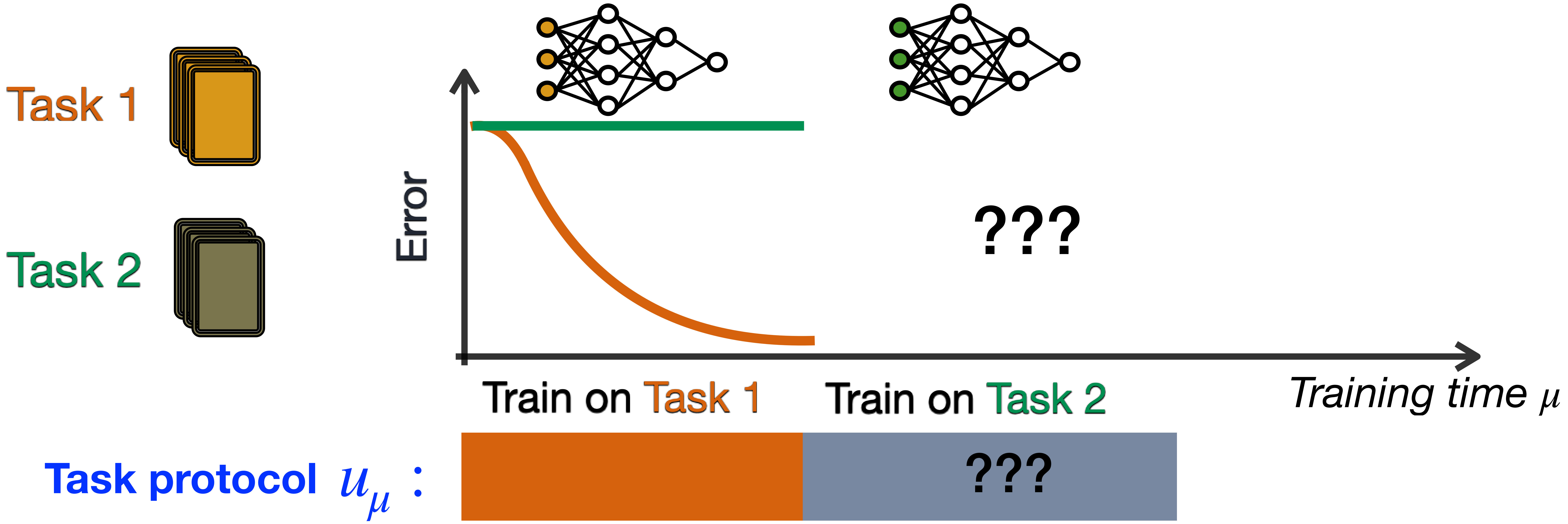
ML (theory)

Continual learning



(Heuristic) interleaved *replay* strategy

Continual learning



What is the *optimal* replay protocol?

A teacher-student model of continual learning

Introduced in: *Lee, Goldt, & Saxe (ICML 2021)*

Task 1

$$\mathscr{D}_1 = \{\mathbf{x}_\mu^{(1)}, y_\mu^{(1)}\}$$

$$\mathbf{x}_\mu^{(1)} \sim \mathscr{N}(0, I_N) \in \mathbf{R}^N$$

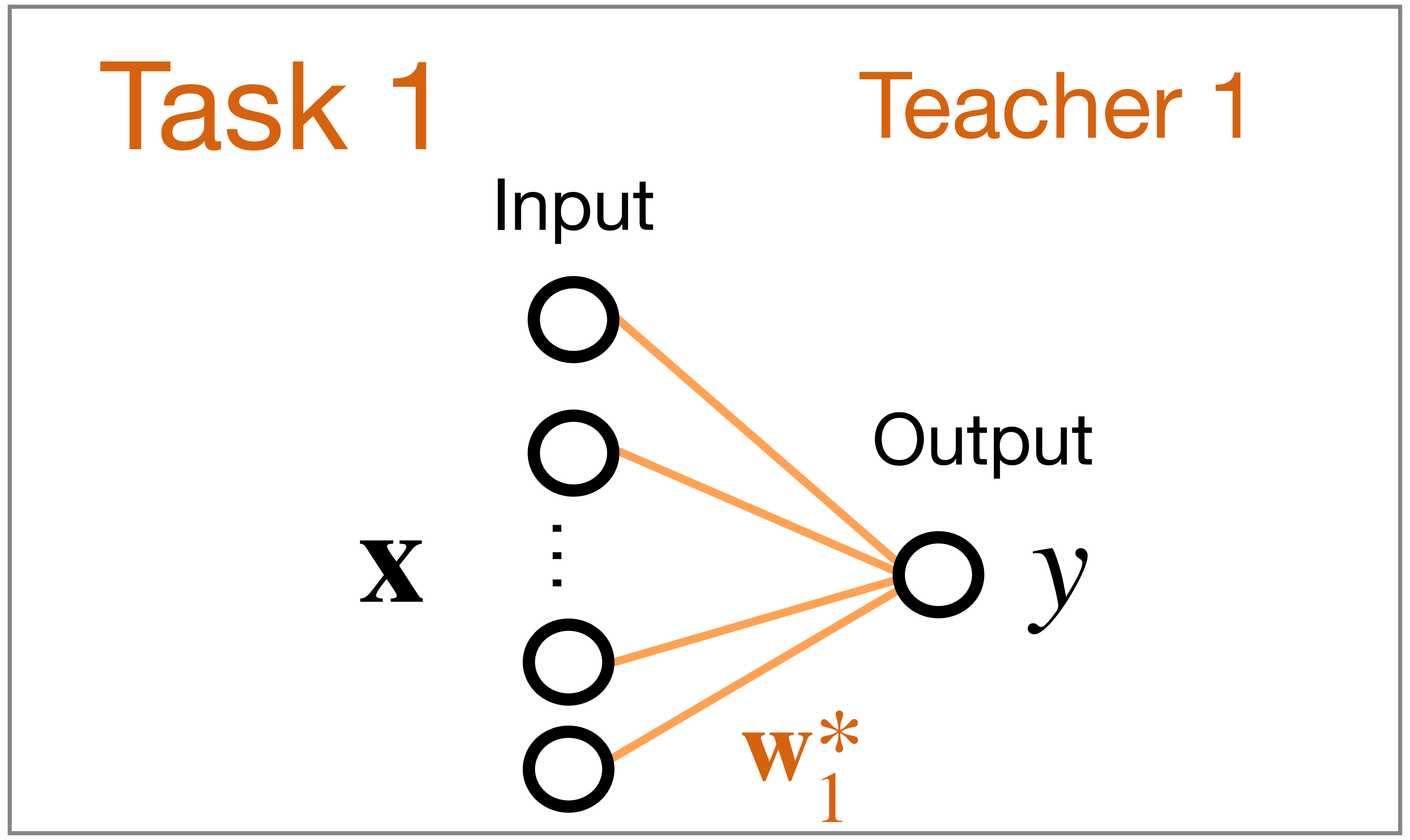
Task 2

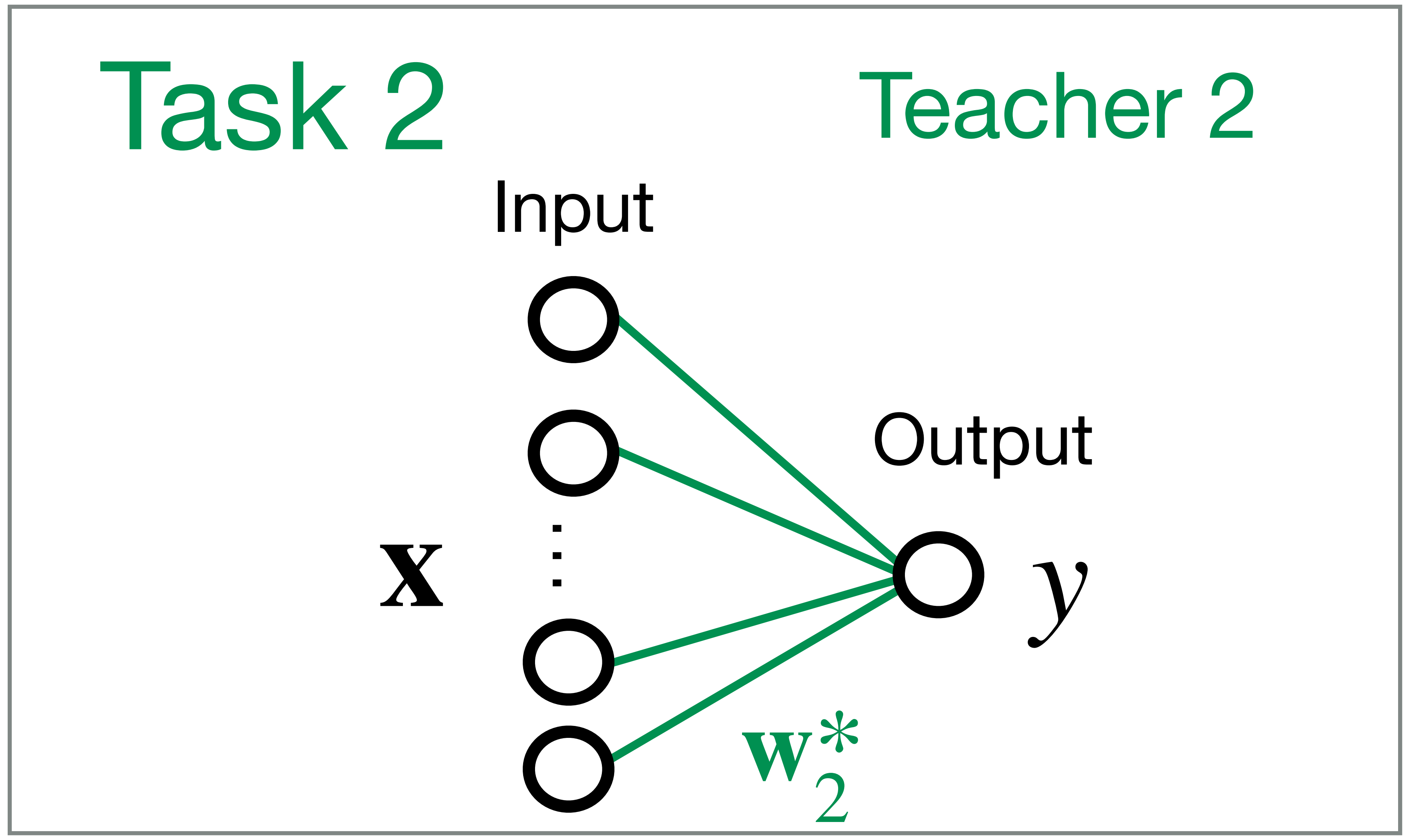
$$\mathscr{D}_2 = \{\mathbf{x}_\mu^{(2)}, y_\mu^{(2)}\}$$

$$\mathbf{x}_\mu^{(2)} \sim \mathscr{N}(0, I_N) \in \mathbf{R}^N$$

A teacher-student model of continual learning

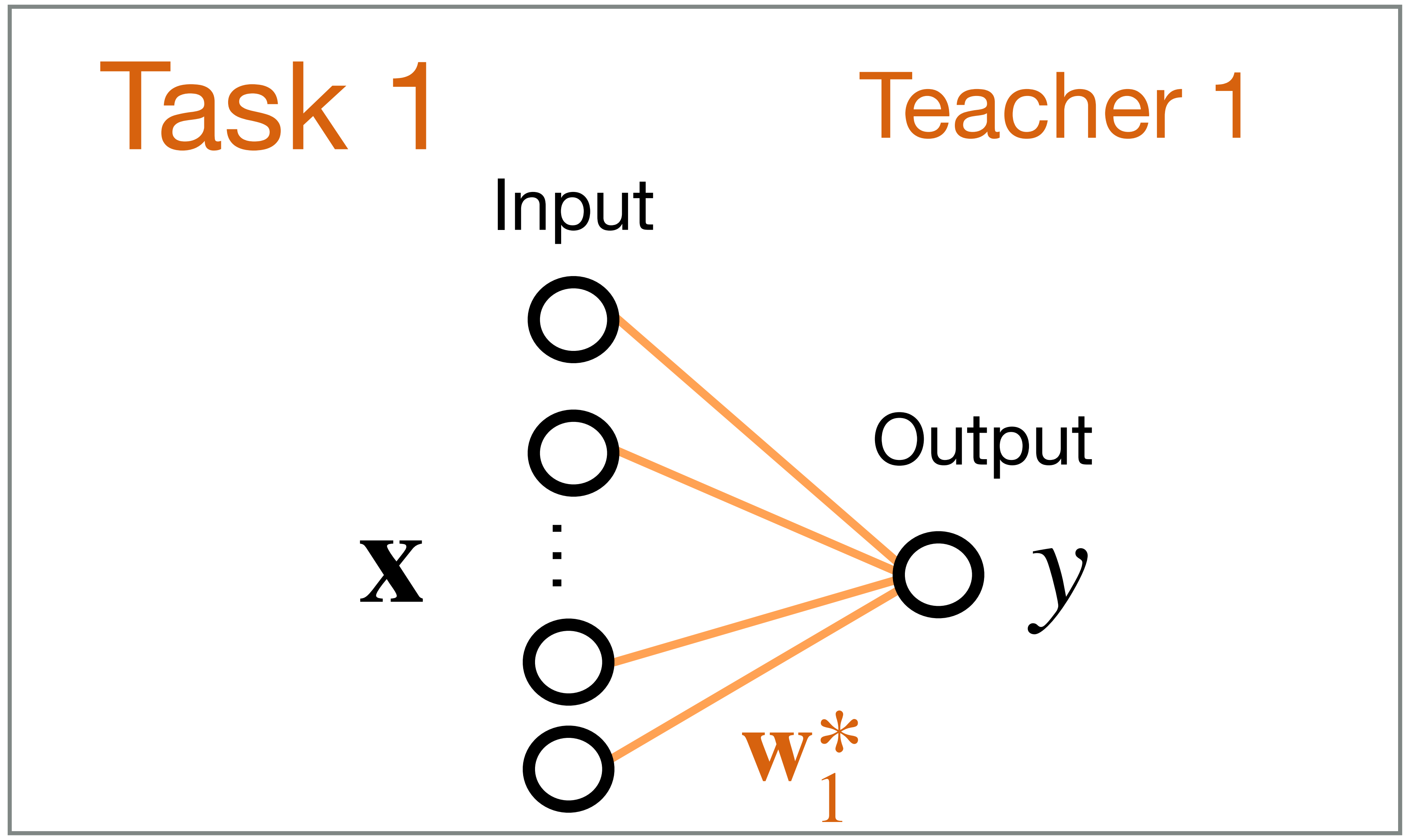
Introduced in: *Lee, Goldt, & Saxe (ICML 2021)*

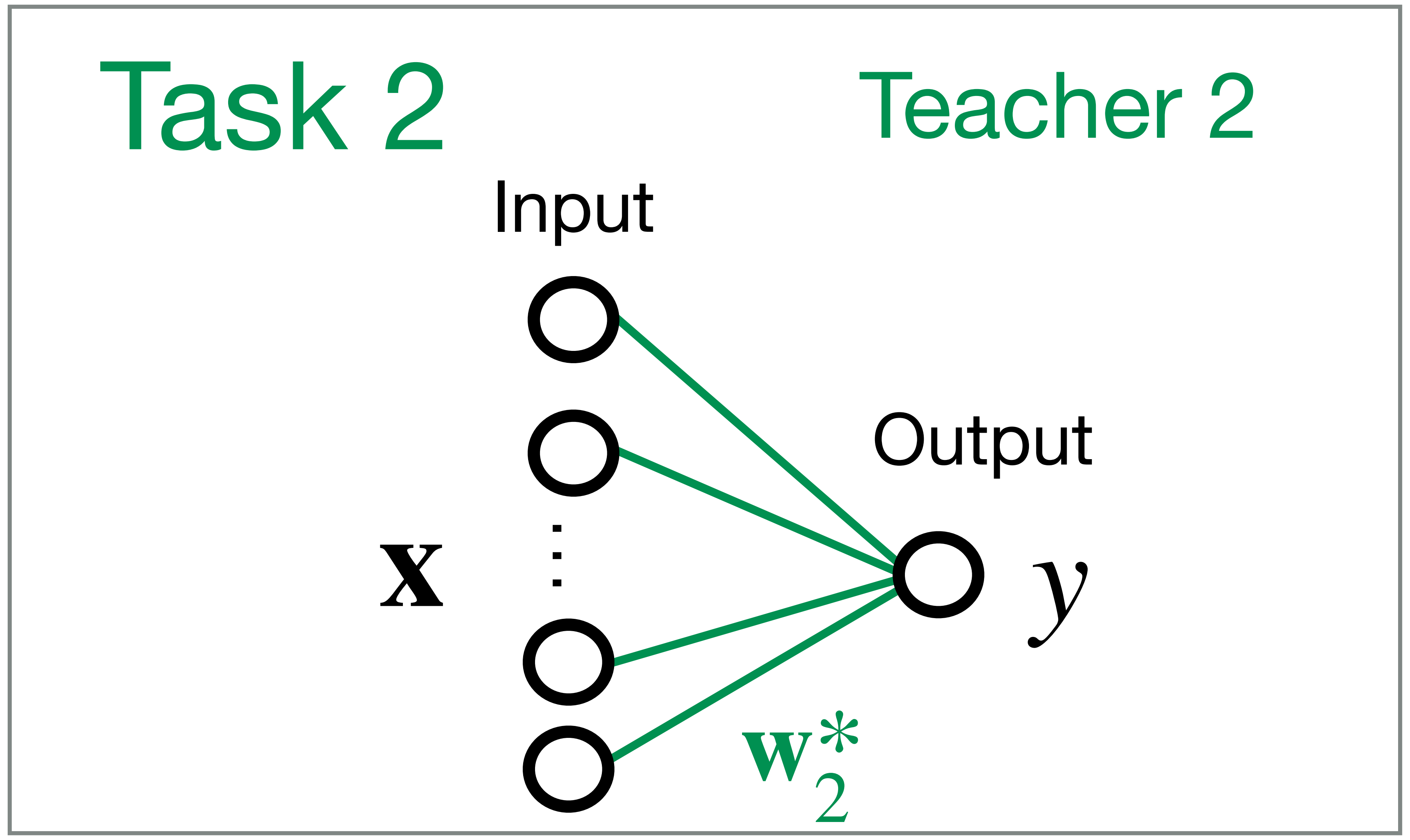


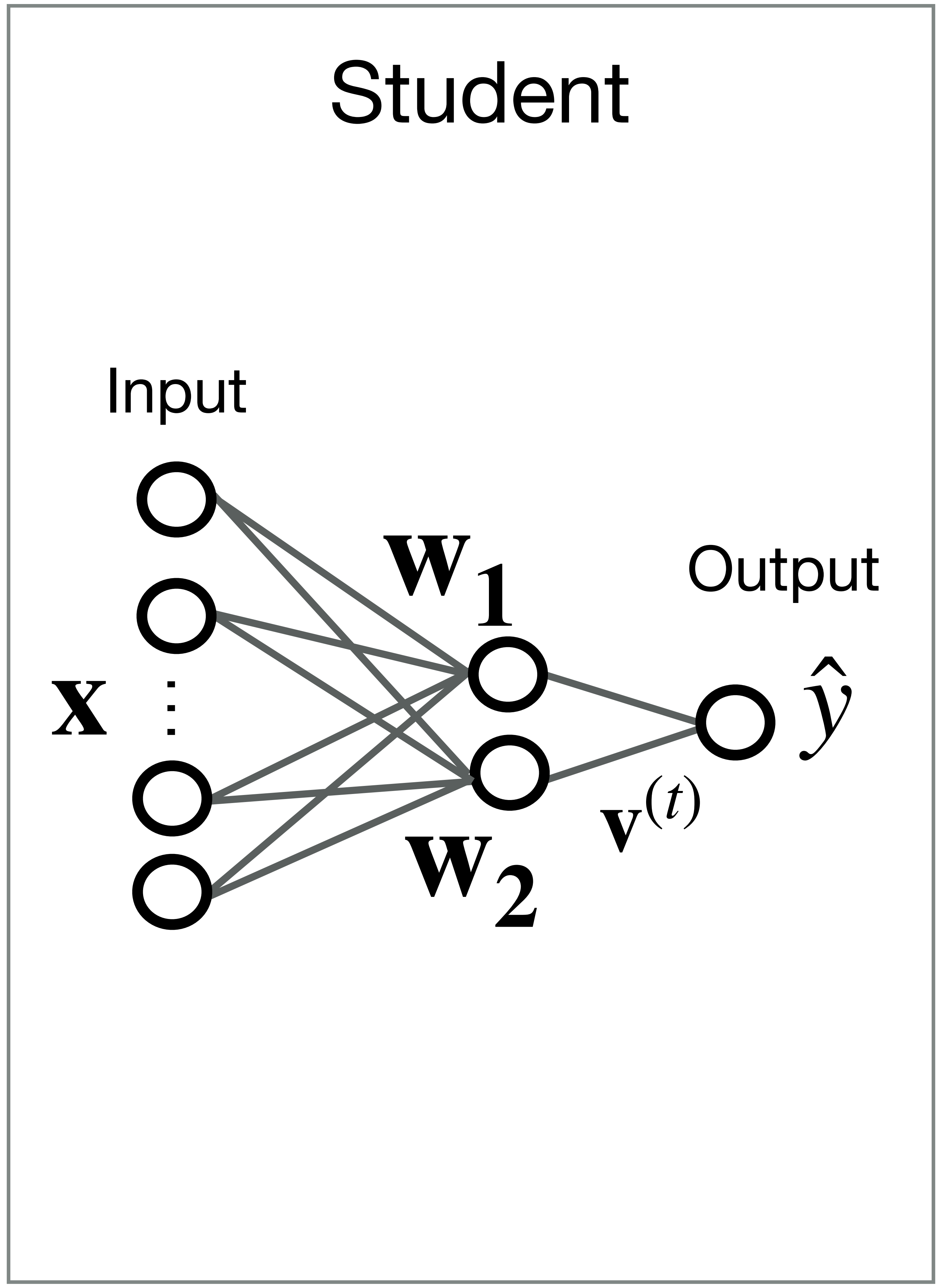


A teacher-student model of continual learning

Introduced in: *Lee, Goldt, & Saxe (ICML 2021)*

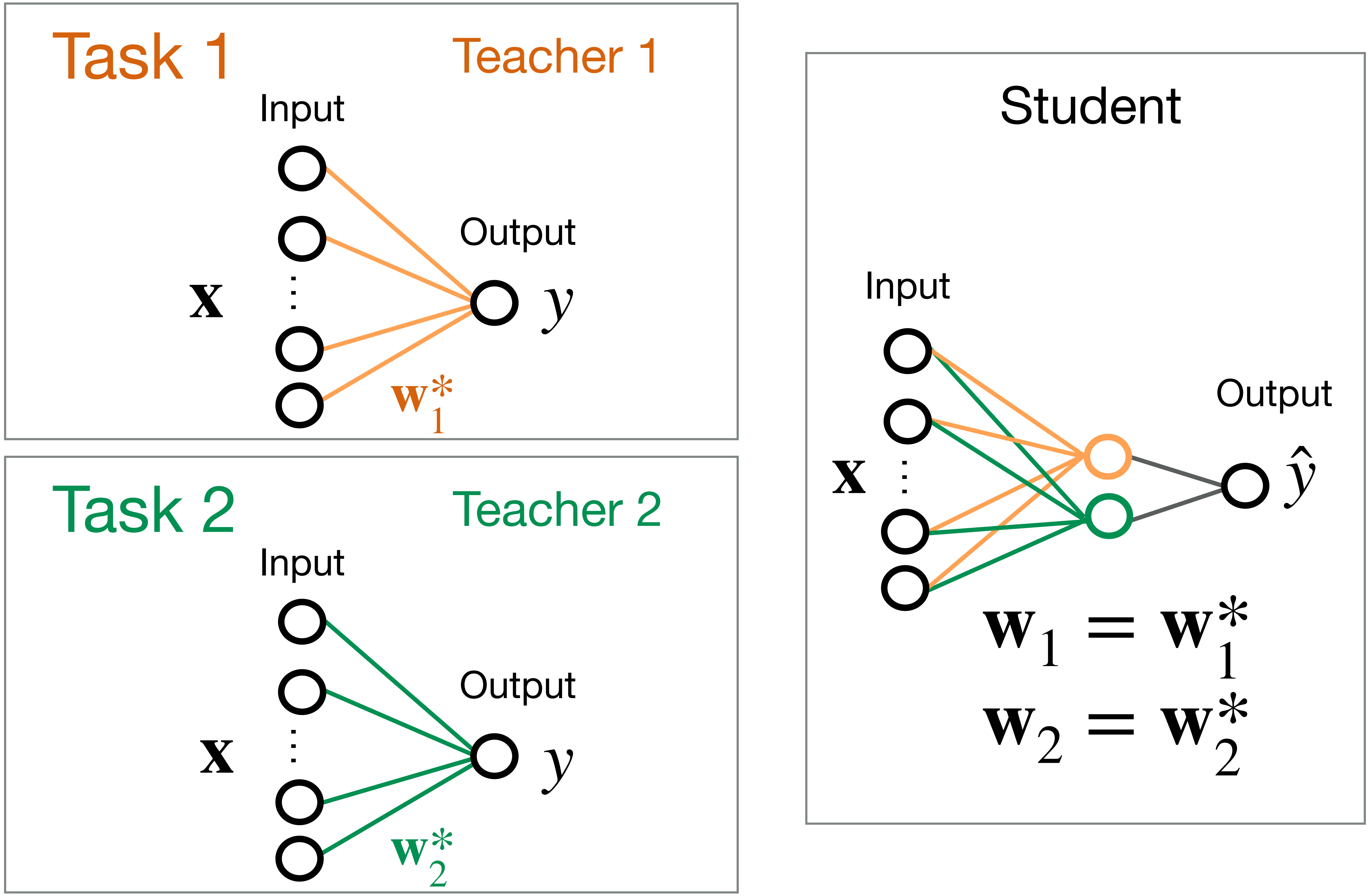






A teacher-student model of continual learning

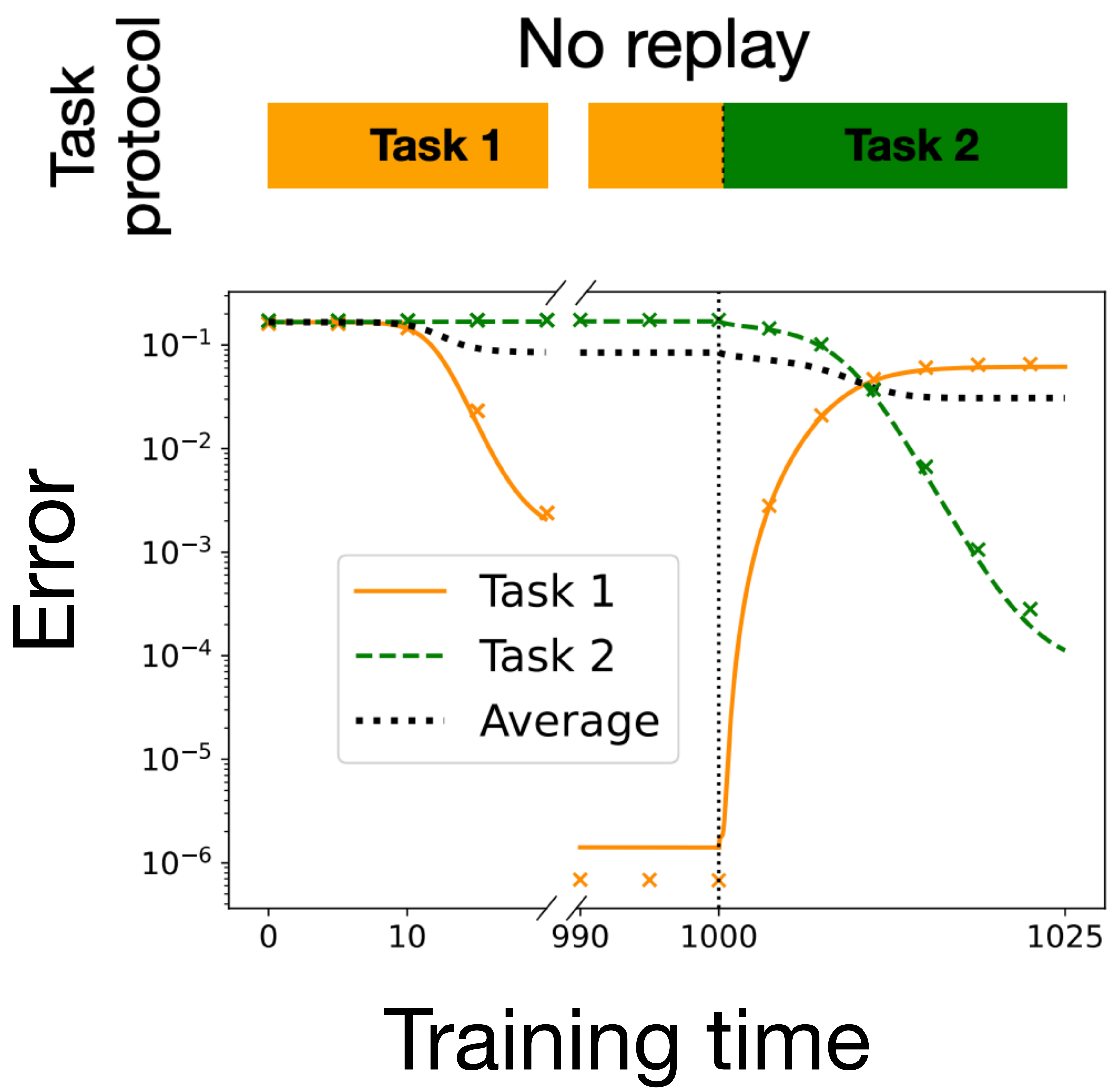
Introduced in: *Lee, Goldt, & Saxe (ICML 2021)*



The student has the capacity to learn both tasks.

Optimal strategy vs benchmarks

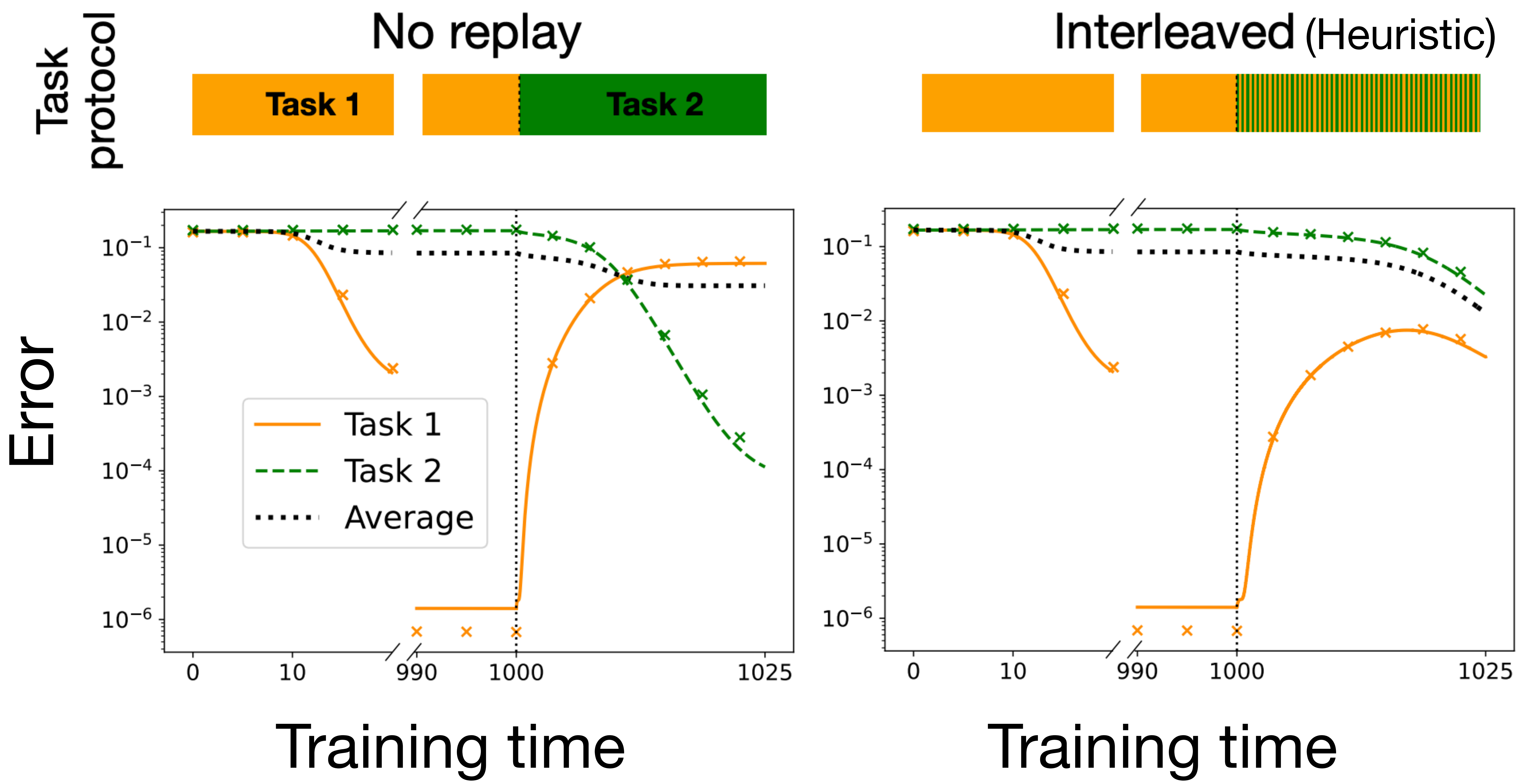
Control: $\mathbf{u} = \text{task}$



FM, S. Sarao Mannelli, F. Mignacco, ICLR 2025

Optimal strategy vs benchmarks

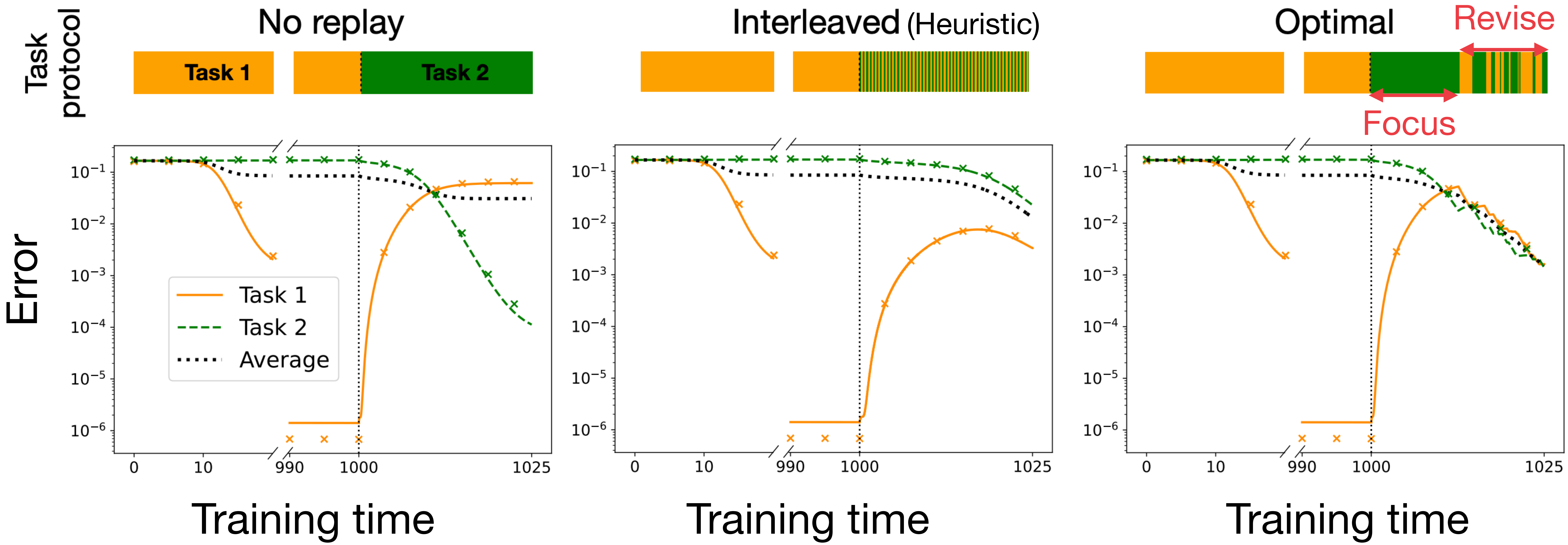
Control: $\mathbf{u} = \text{task}$



FM, S. Sarao Mannelli, F. Mignacco, ICLR 2025

Optimal strategy vs benchmarks

Control: $\mathbf{u} = \text{task}$



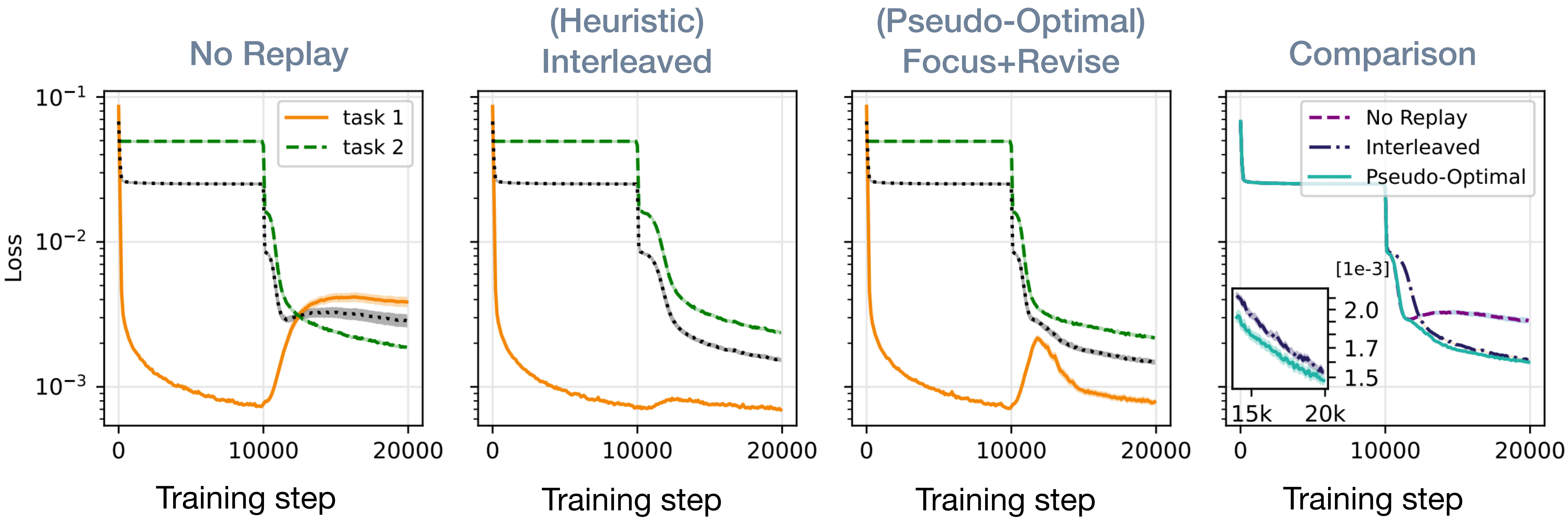
FM, S. Sarao Mannelli, F. Mignacco, ICLR 2025

Pseudo-optimal strategy on real data

Task 1: $\mathcal{D}_1 = \{\boldsymbol{x}_i^{(1)}, y_i^{(1)}\}_i$

Task 2: $\mathcal{D}_2 = \{\boldsymbol{x}_i^{(2)}, y_i^{(2)}\}_i = \{\gamma \boldsymbol{x}_i^{(1)} + (1-\gamma)\tilde{\boldsymbol{x}}_i, \gamma y_i^{(1)} + (1-\gamma)\tilde{y}_i\}_i$ $\quad \gamma = 0.5$

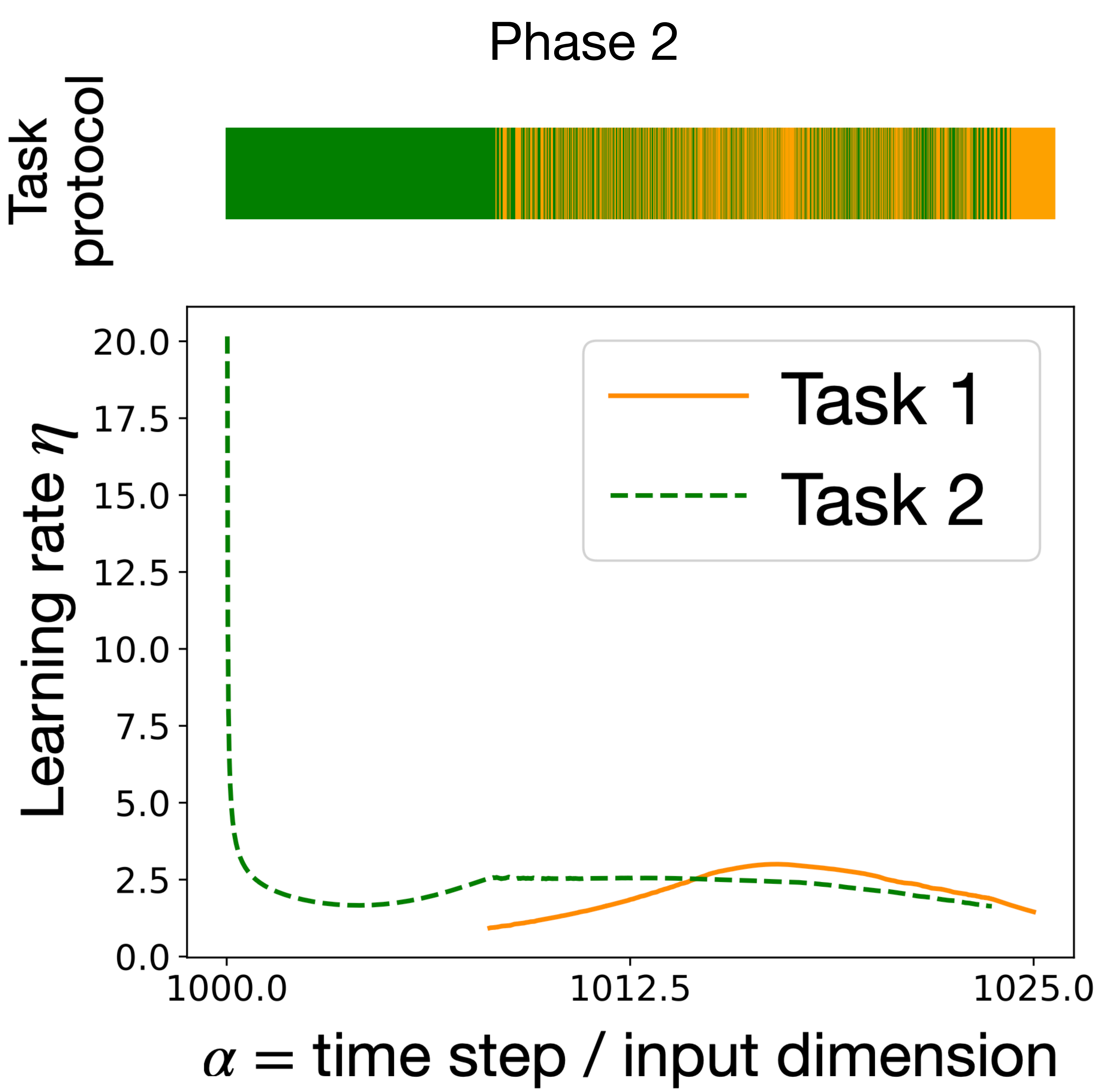
Experiments on Fashion-MNIST:



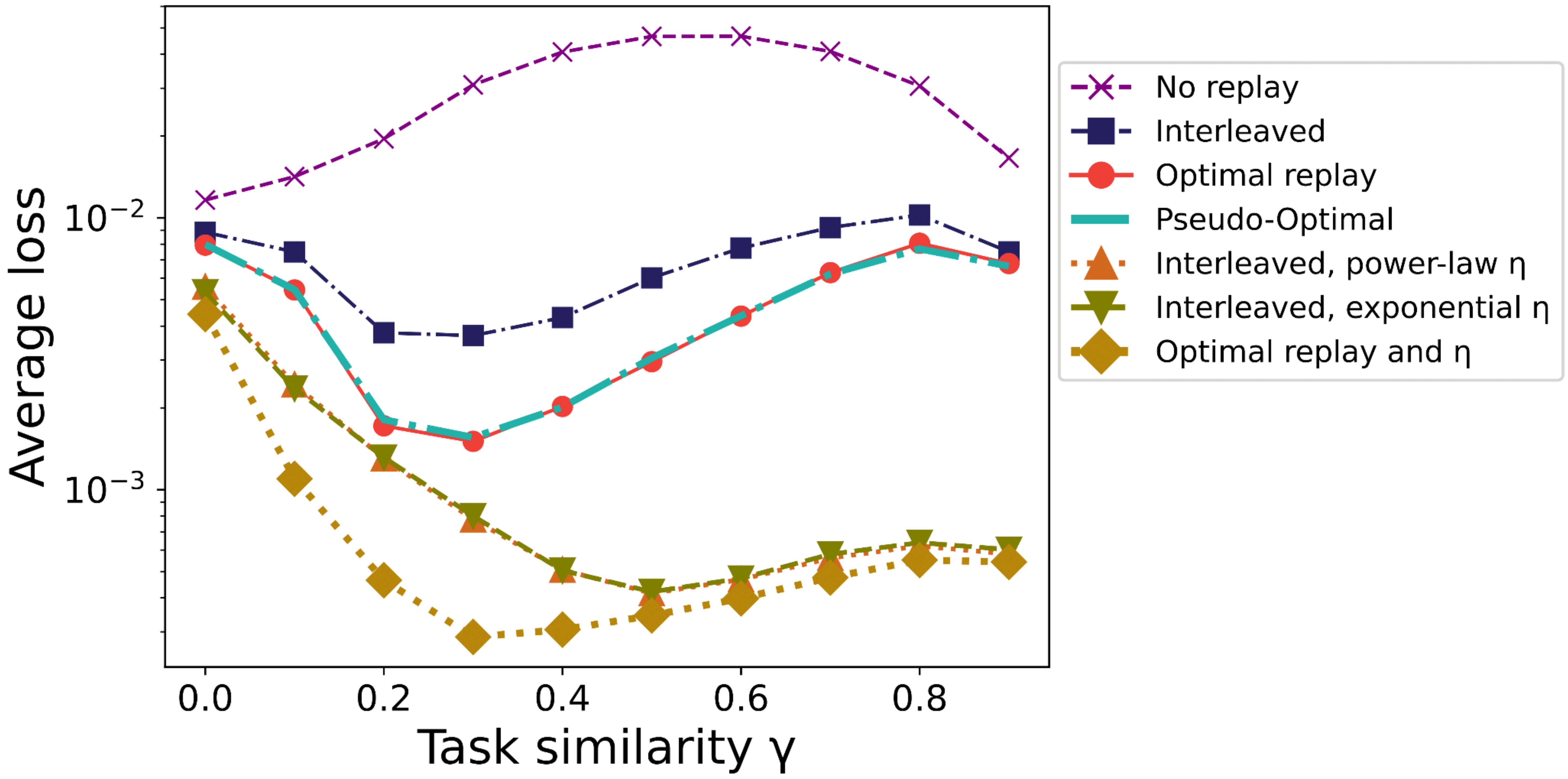
FM, S. Sarao Mannelli, F. Mignacco, ICLR 2025

Joint optimization of replay and learning rate schedule

Control: $\mathbf{u} = (\text{task}\ , \eta)$



$\gamma = 0.3$



FM, S. Sarao Mannelli, F. Mignacco, ICLR 2025

Part III: *Denoising autoencoders*

F. Mignacco, **FM**, *arXiv:2507.07907*

Denoising with autoencoders

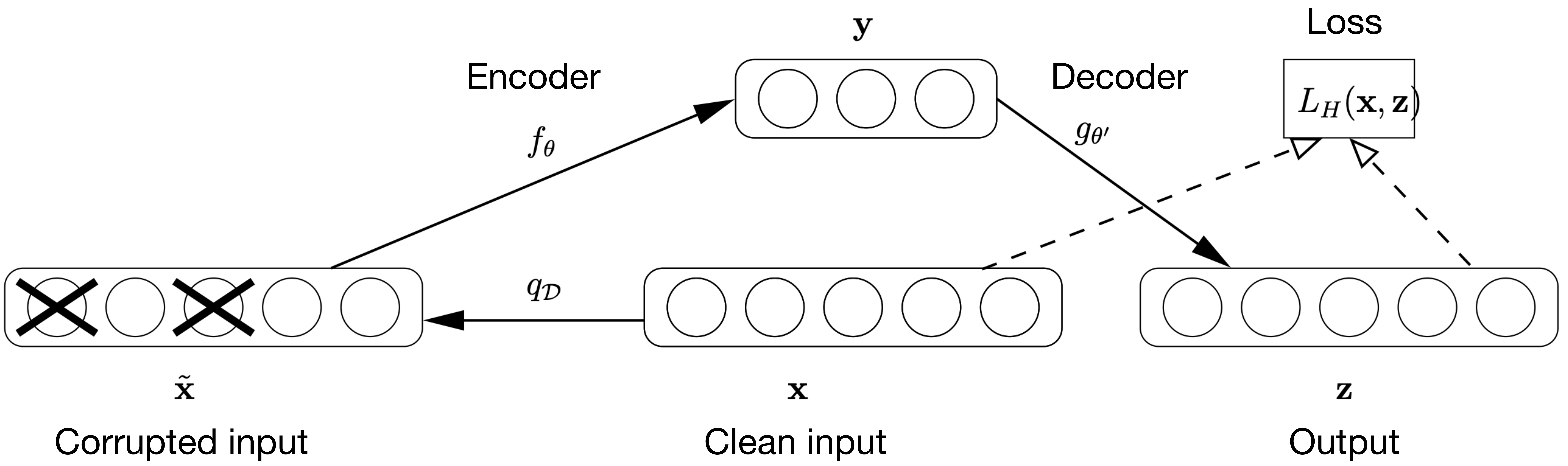


Image source: Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. *ICML* 2008

Denoising with autoencoders

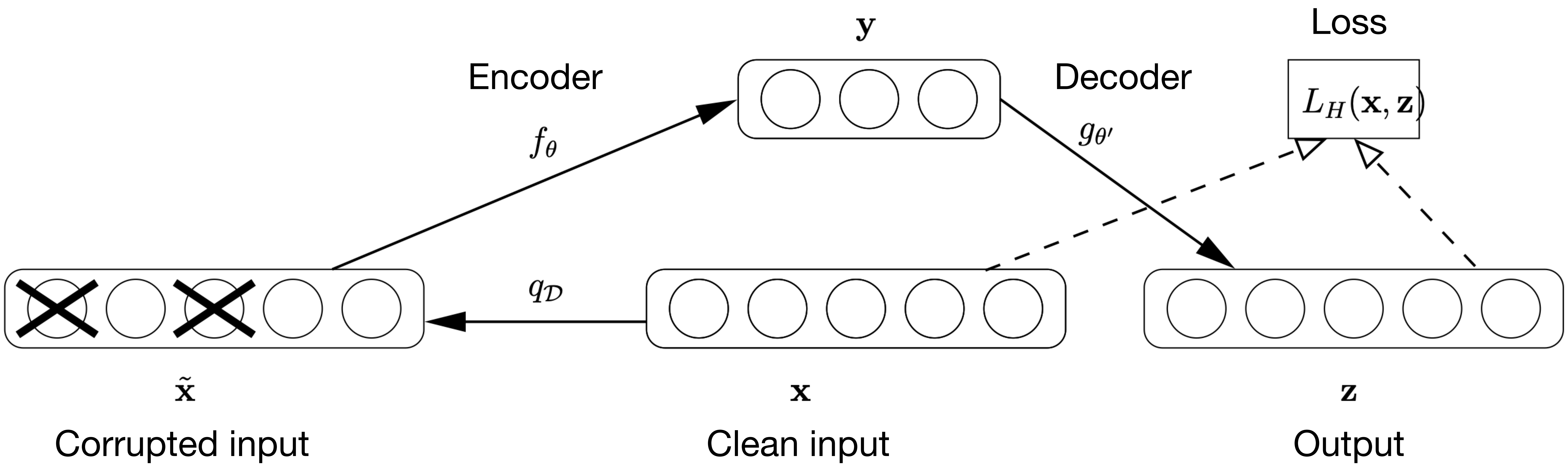


Image source: Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. ICML 2008

Theoretical works (non exhaustive list):

- **Learning dynamics in linear DAEs**
  [Pretorius, Kroon, and Kamper, ICML 2018]
- **Exact asymptotics in nonlinear DAEs**
  [Cui, and Zdeborová, NeurIPS 2023]
- **Online learning in shallow reconstruction autoencoders**
  [Refinetti, and Goldt, ICML 2022]
- **Diffusion models parametrized by DAEs**
  [Cui, Krzakala, Vanden-Eijnden, and Zdeborová, ICLR 2024; Cui, Pehlevan, and Lu, 2025]

A prototypical model of DAE

See also: [Cui, and Zdeborová, NeurIPS 2023]

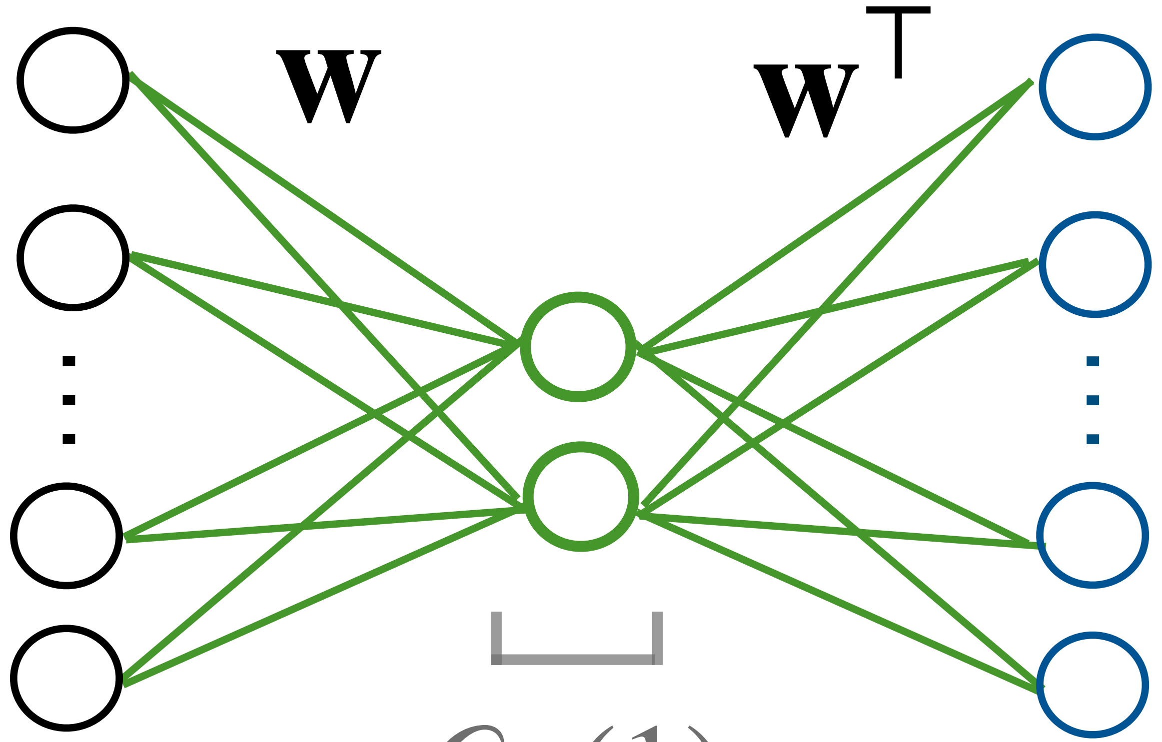
Clean input: $\mathbf{x} \sim \sum_{c=1}^{C} p_c \mathcal{N}(\mu_c, \sigma_c^2 \mathrm{I}_N)$

Noise level: $\Delta \in (0,1)$

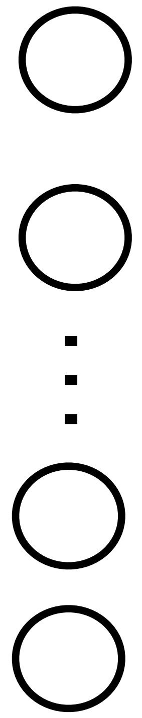
Corrupted input: $\tilde{\mathbf{x}} = \sqrt{1-\Delta}\,\mathbf{x} + \sqrt{\Delta}\,\xi \;,\qquad \xi \sim \mathcal{N}(0, \mathrm{I}_N)$

Two-layer DAE Bottleneck network Skip connection

$f_{\mathbf{w},b}(\tilde{\mathbf{x}}) =$



$\tilde{\mathbf{x}} \; + \; b$



$\tilde{\mathbf{x}}$

$\mathcal{O}_N(1)$

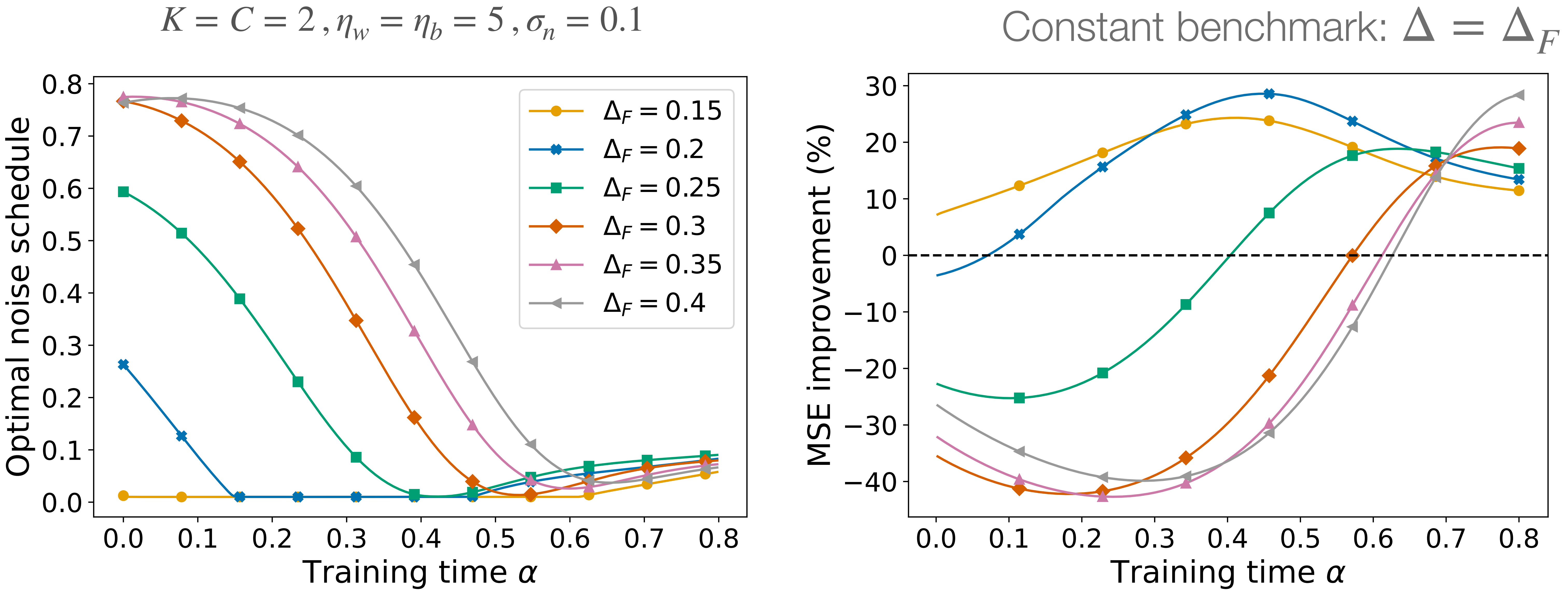
(Linear for simplicity)

One-pass SGD on the loss: $\mathcal{L}(\mathbf{w}, b) = \frac{1}{2} \| f_{\mathbf{w}}(\tilde{\mathbf{x}}) - \mathbf{x} \|^2$

Optimal noise schedule at fixed test noise level

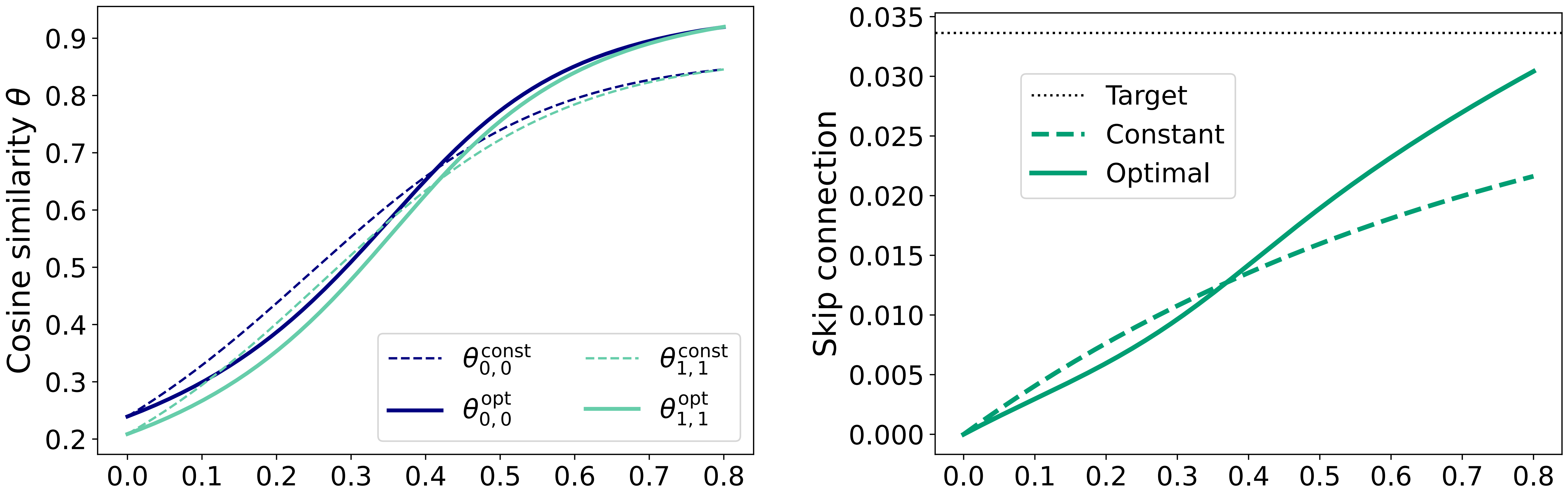
Control: $u(\alpha) = \Delta(\alpha)$

Final performance: at fixed noise level Δ_F



The optimal schedule:

1. Enhances the reconstruction capability of the bottleneck network
2. Accelerates the convergence of the skip connection



$$\theta_{k,c} = \frac{\mathbf{w}_k \cdot \mu_c}{\sqrt{\|\mathbf{w}_k\| \|\mu_c\|}}$$

Optimal batch size schedule

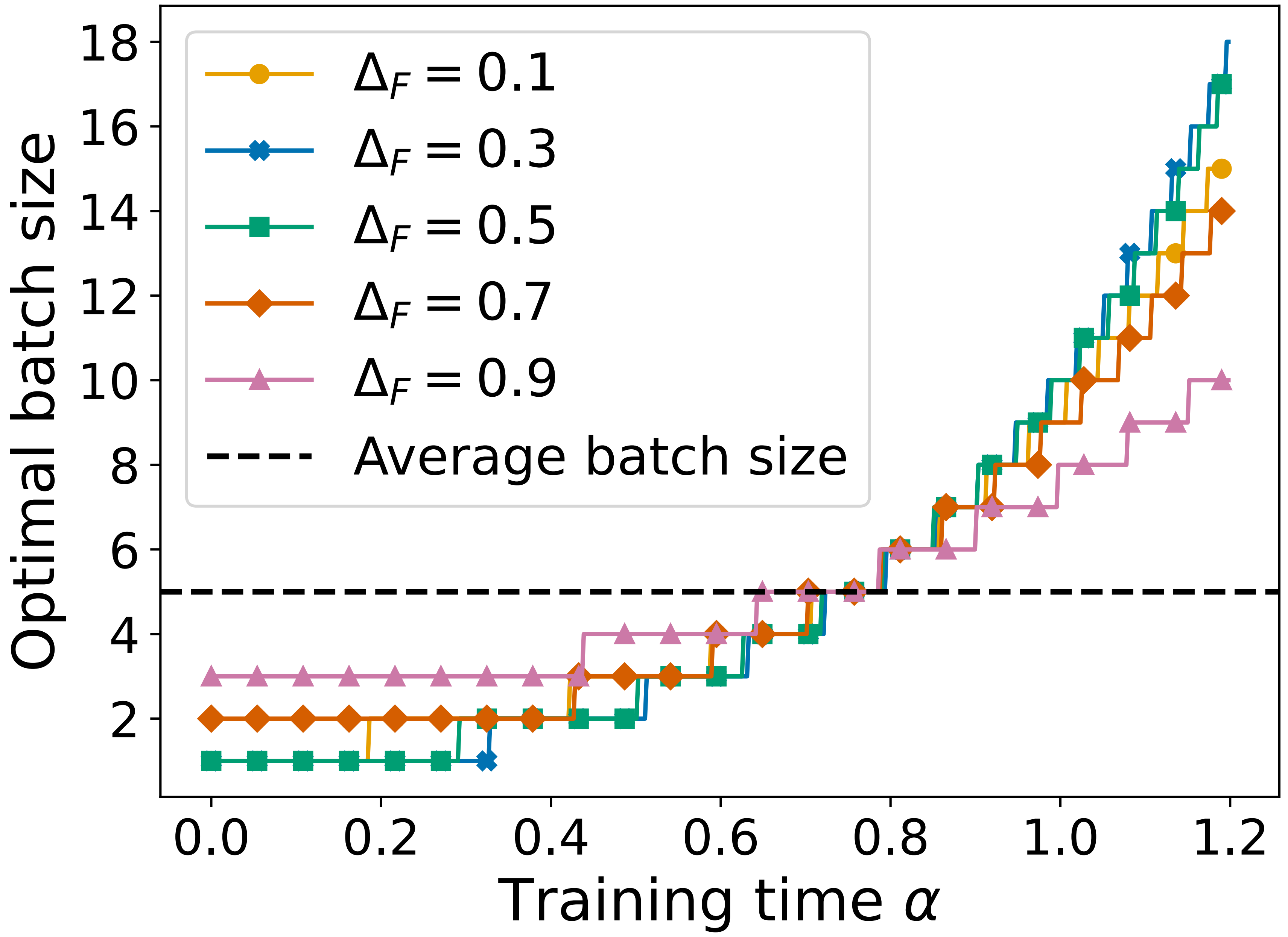
Batch augmentation: $\tilde{\mathbf{x}}_a = \sqrt{1-\Delta}\,\mathbf{x} + \sqrt{\Delta}\,\xi_a$, with $a = 1, \dots, B$ independent realizations of the noise

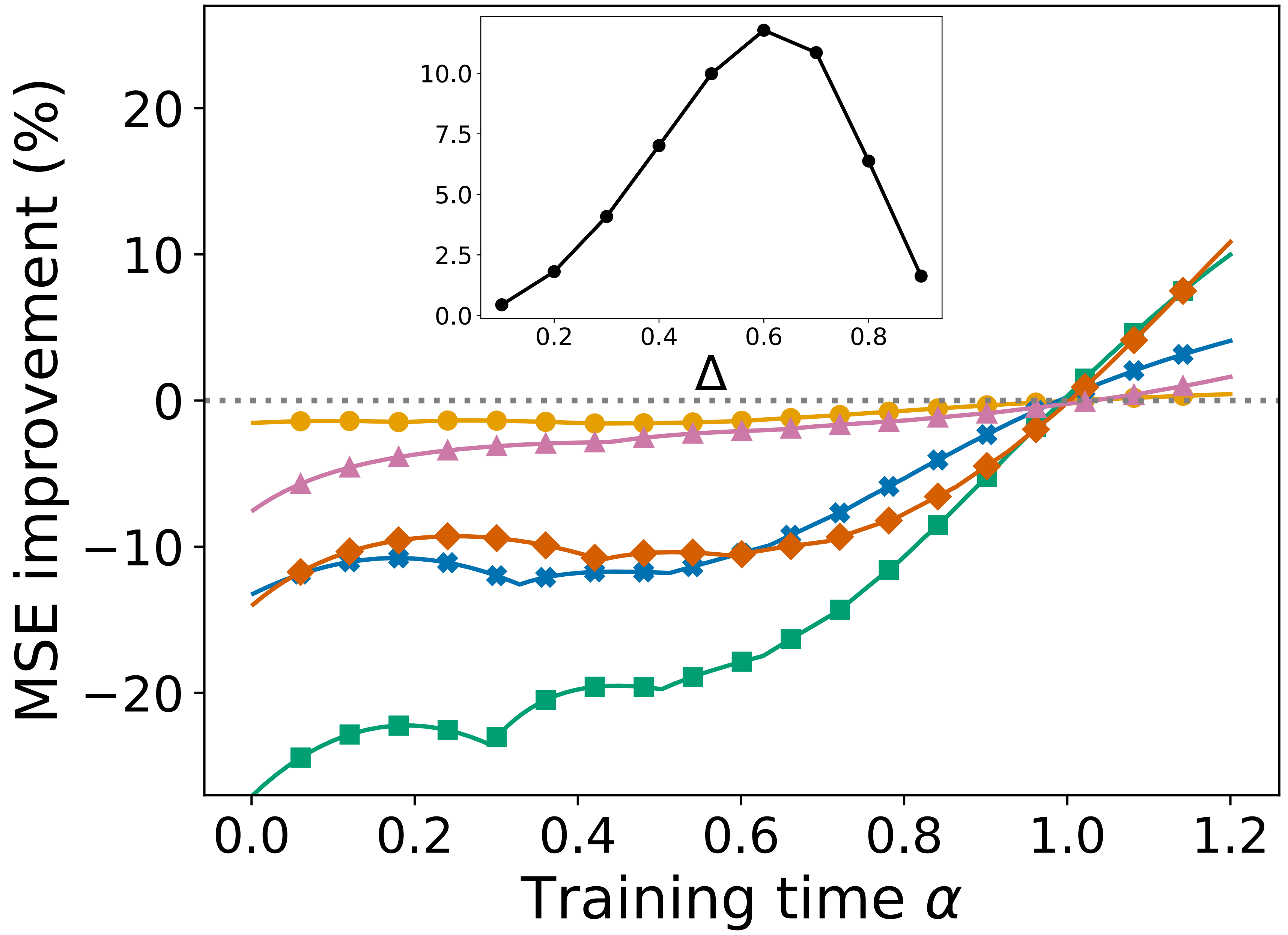
One-pass SGD on the loss: $\mathscr{L}(\mathbf{w}) = \frac{1}{2B} \sum_{a=1}^{B} \| f_{\mathbf{w}}(\tilde{\mathbf{x}}_a) - \mathbf{x} \|^2$

Constant benchmark: $B = \bar{B}$

Control: $u = B$

(time dependent batch size, **fixed total budget**)

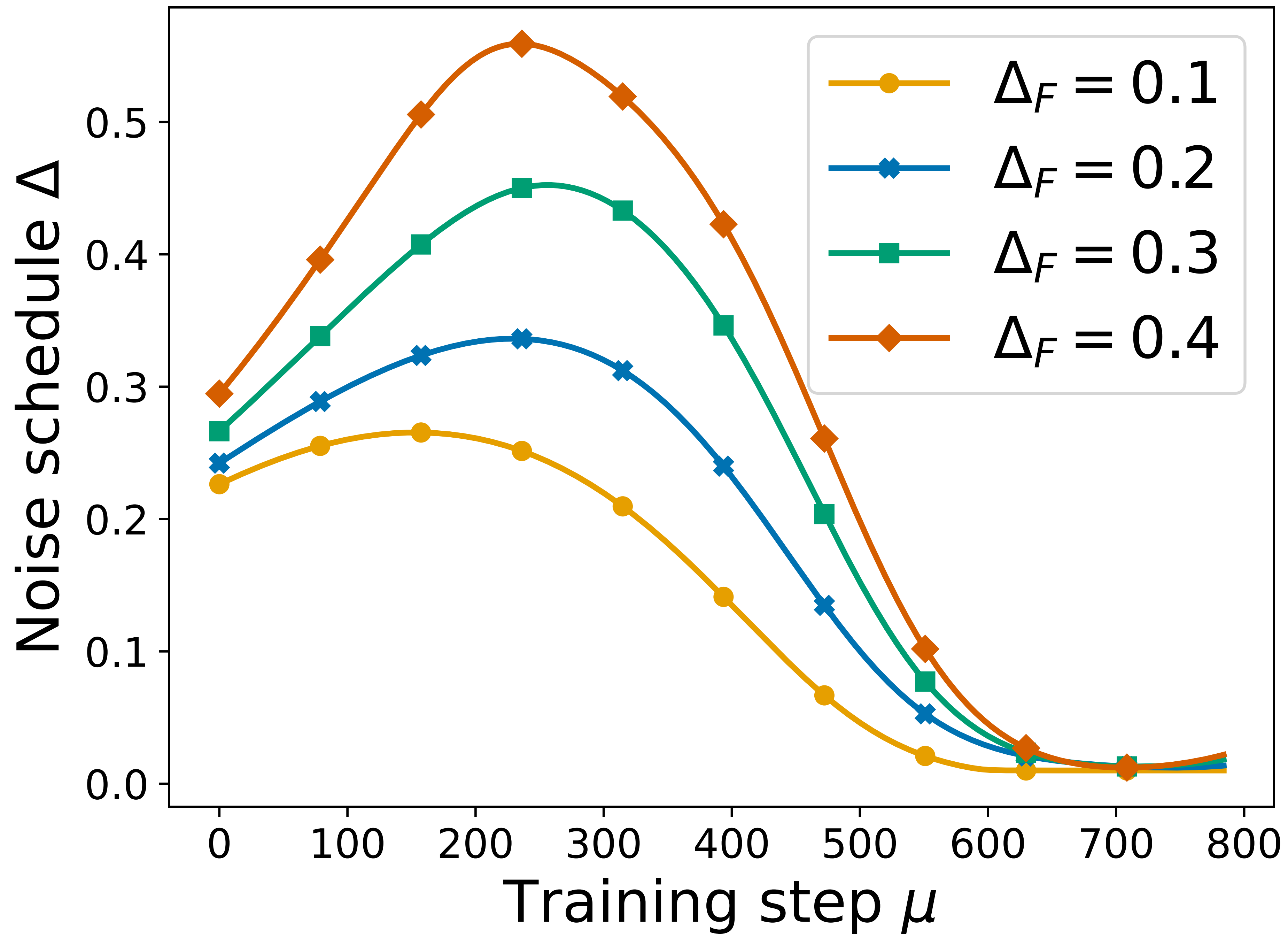


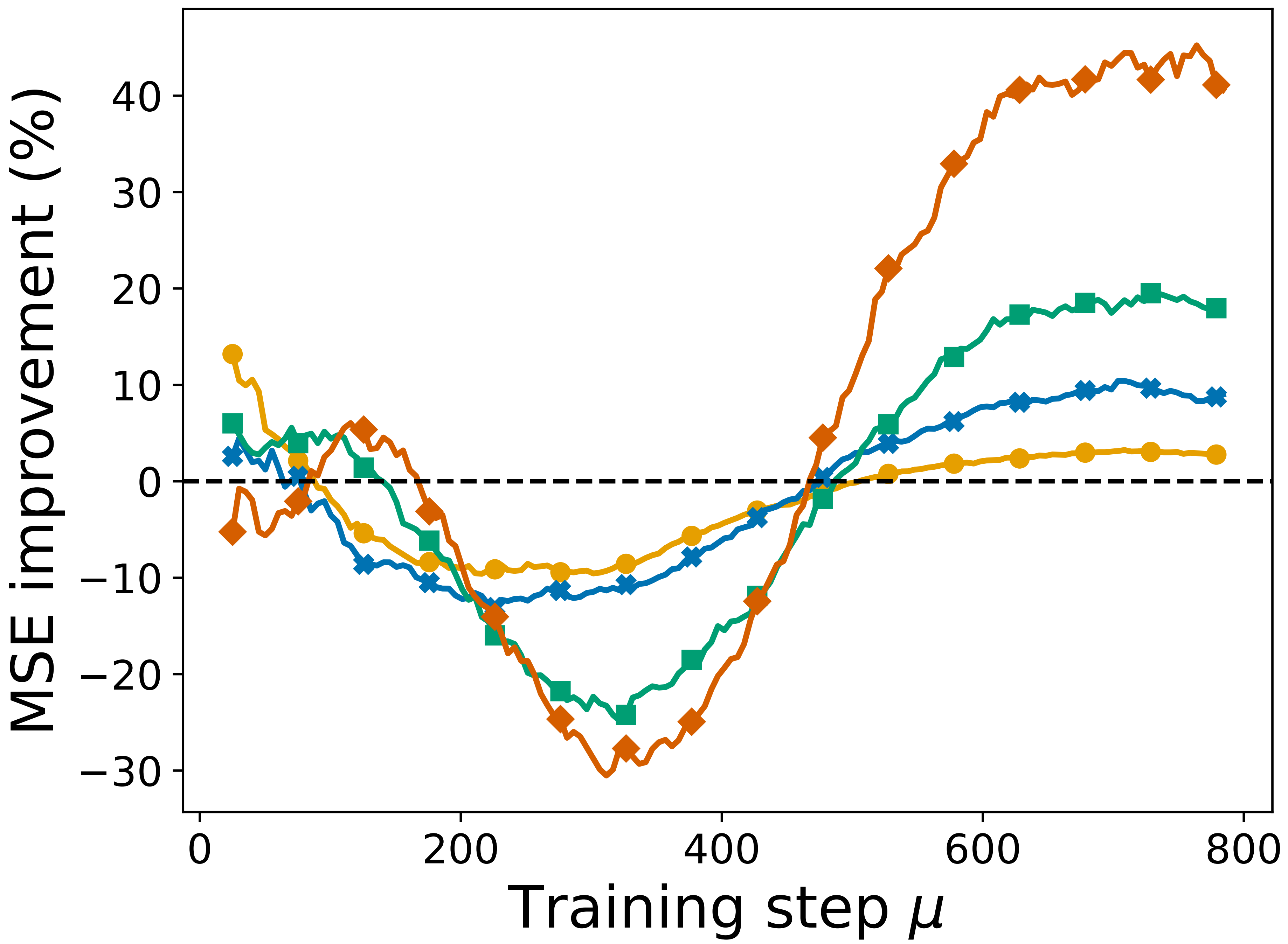


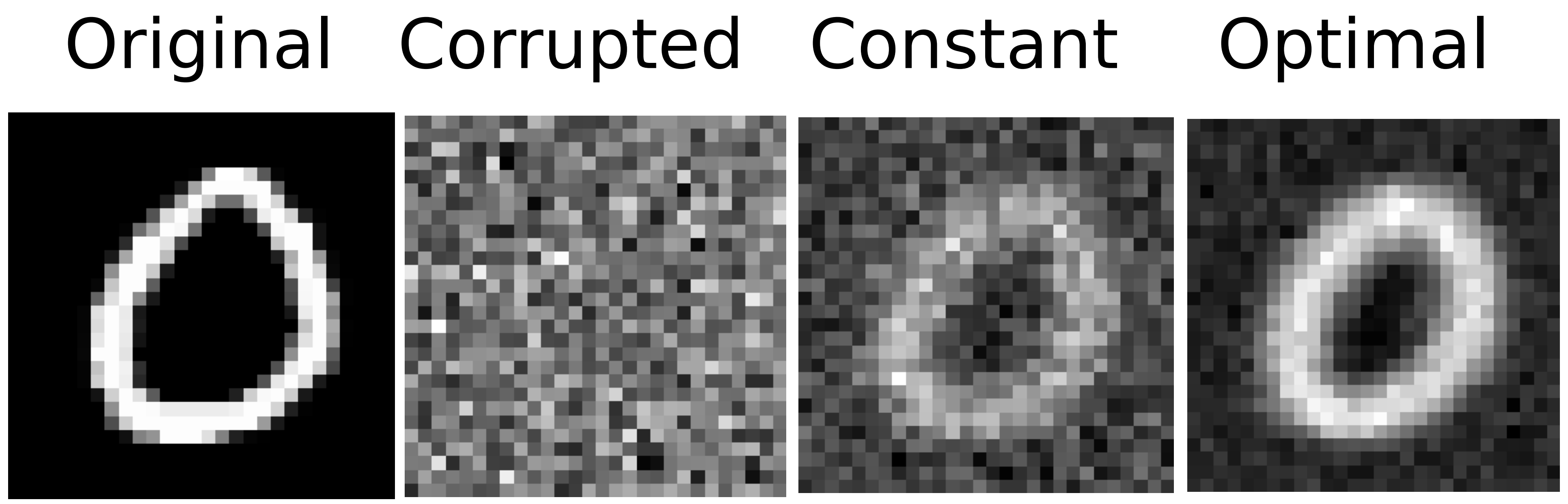
$K = C = 2\,, \eta_w = \eta_b = 5\,, \sigma_n = 0.1$

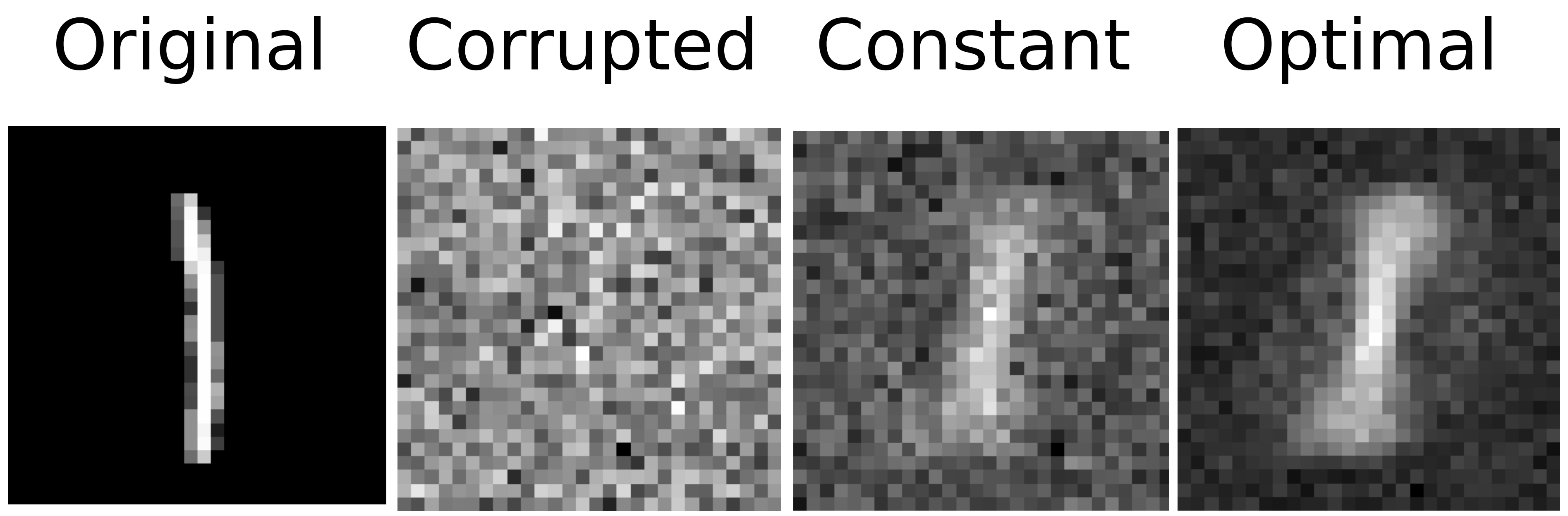
MNIST experiments

Optimal noise schedules fitting only the cluster means and variances:









Conclusions & Perspectives

- We formulate the design of learning protocols as an optimal control problem on the low-dimensional dynamics of the order parameters.
- Nontrivial yet interpretable strategies applicable to real data.
- Learning trade-offs

Many open directions, *e.g.,*

- Extend the theory to more realistic data models.
- Batch learning and memorization.
- Extend to different learning objectives (fairness metrics …).

Thank you!