CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Smarter and Sparser AI with Complex Network Science

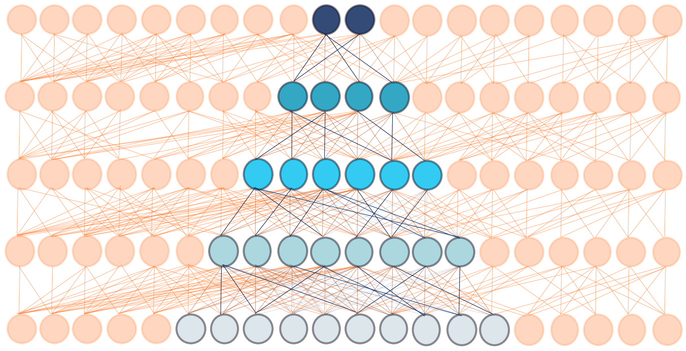
Rebekka Burkholz

Network Science Informs AI (NSIA)

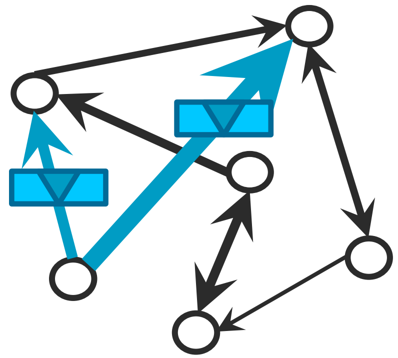
June 1, 2026

Outline: Complex network science for AI

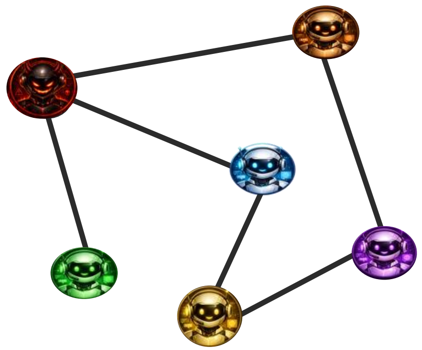
1. **Sparse** deep learning
    - Random graph **ensembles**
    - **Learning networks** (by sparsification)



2. **Graph** Neural Networks:
    - Spectral rewiring (**Braess** paradox)
    - Features by running cascade **process**
    - **Learning on** networks

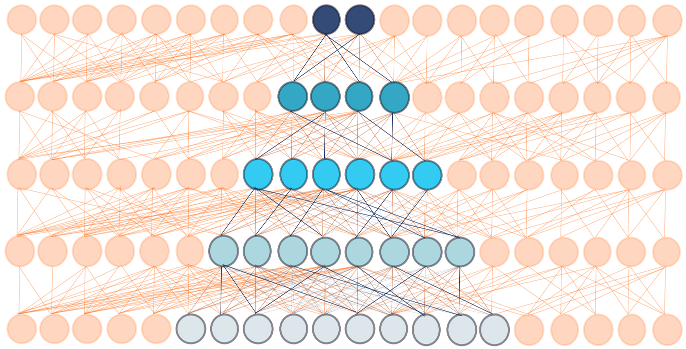


3. **Agentic** networks
    - Robustness of agentic LLM system
    - Friedkin-Johnsen **opinion formation**
    - Process/network weights **emerge**

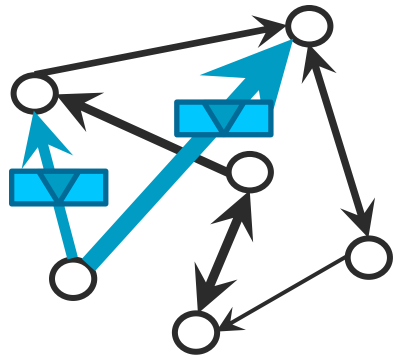


Outline: Complex network science for AI

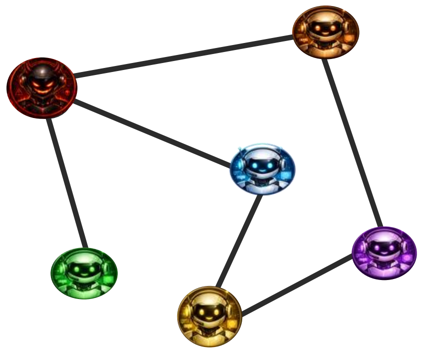
1. **Sparse** deep learning
   - Random graph **ensembles**
   - **Learning networks** (by sparsification)



2. **Graph** Neural Networks:
   - Spectral rewiring (**Braess** paradox)
   - Features by running cascade **process**
   - **Learning on** networks

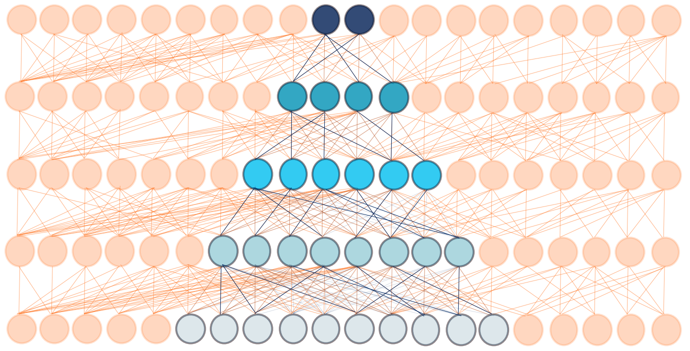


3. **Agentic** networks
   - Robustness of agentic LLM system
   - Friedkin-Johnsen **opinion formation**
   - Process/network weights **emerge**

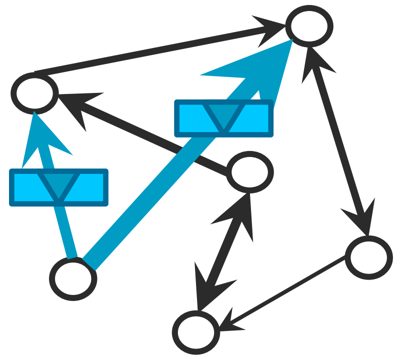


Outline: Complex network science for AI

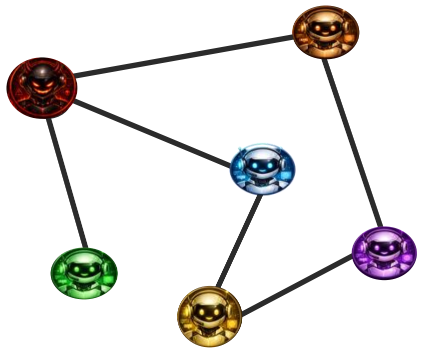
1. **Sparse** deep learning
    - Random graph **ensembles**
    - **Learning networks** (by sparsification)



2. **Graph** Neural Networks:
    - Spectral rewiring (**Braess** paradox)
    - Features by running cascade **process**
    - **Learning on** networks

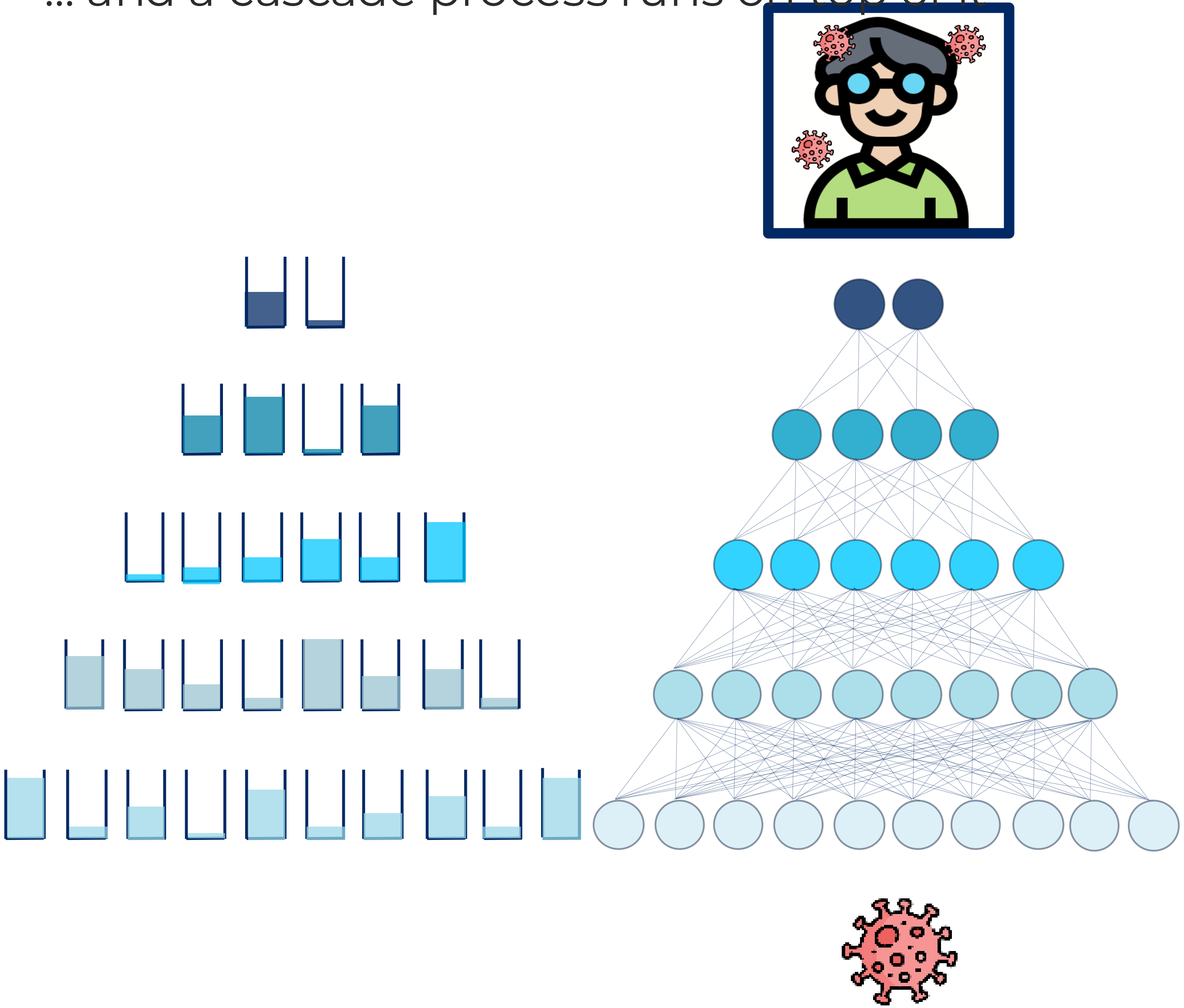


3. **Agentic** networks
    - Robustness of agentic LLM system
    - Friedkin-Johnsen **opinion formation**
    - Process/network weights **emerge**

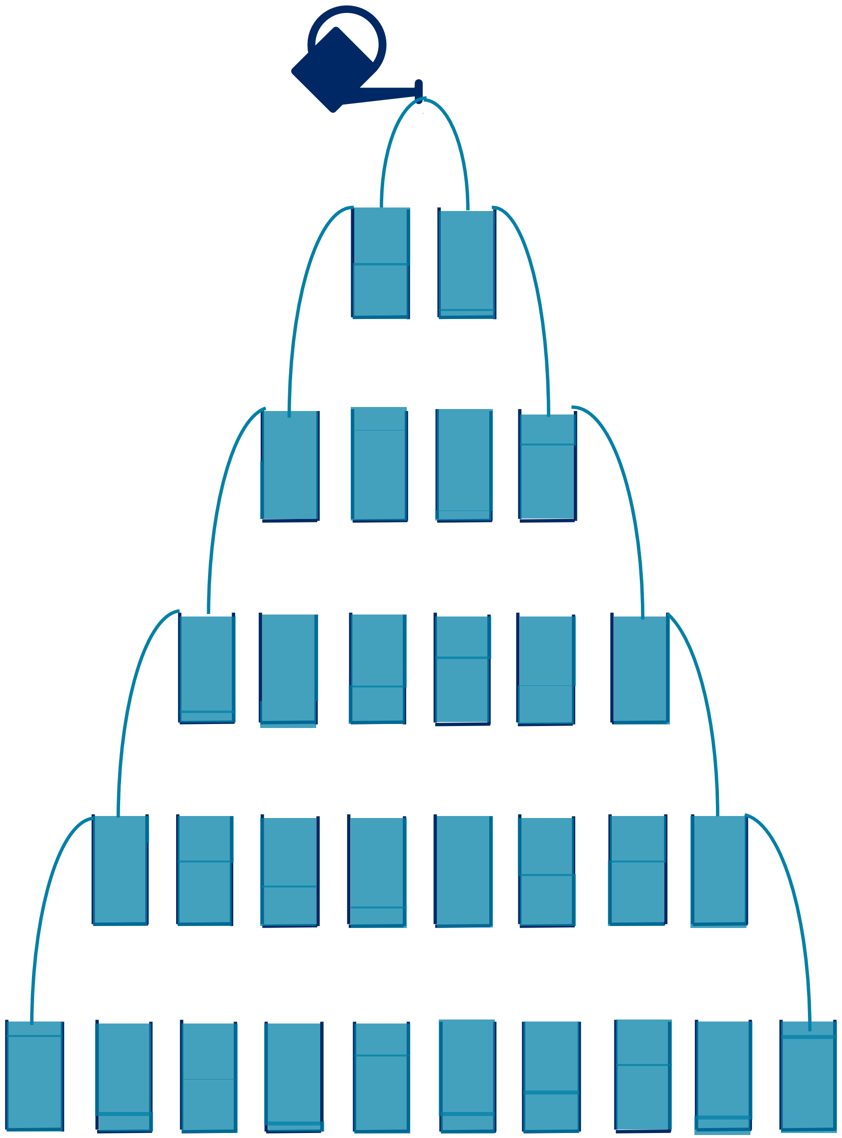


Deep neural networks are complex networks

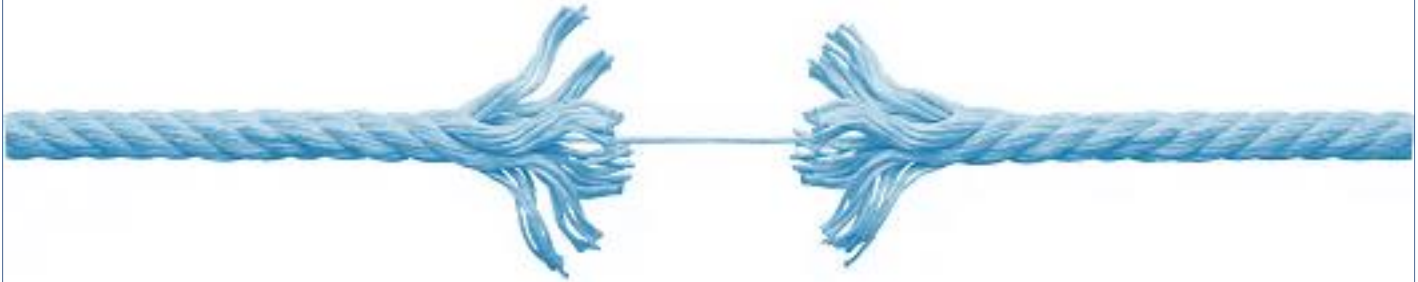
... and a cascade process runs on top of it



Cascade process: overload distribution



Fiber bundles / cracking materials



Burkholz, Schweitzer.
Framework for cascade size calculations on random networks. PRE 2018.

Neural network = cascade process

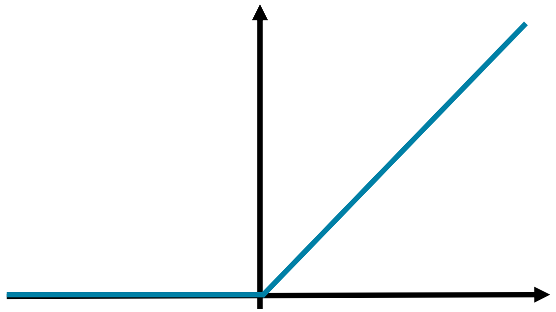
Neural network

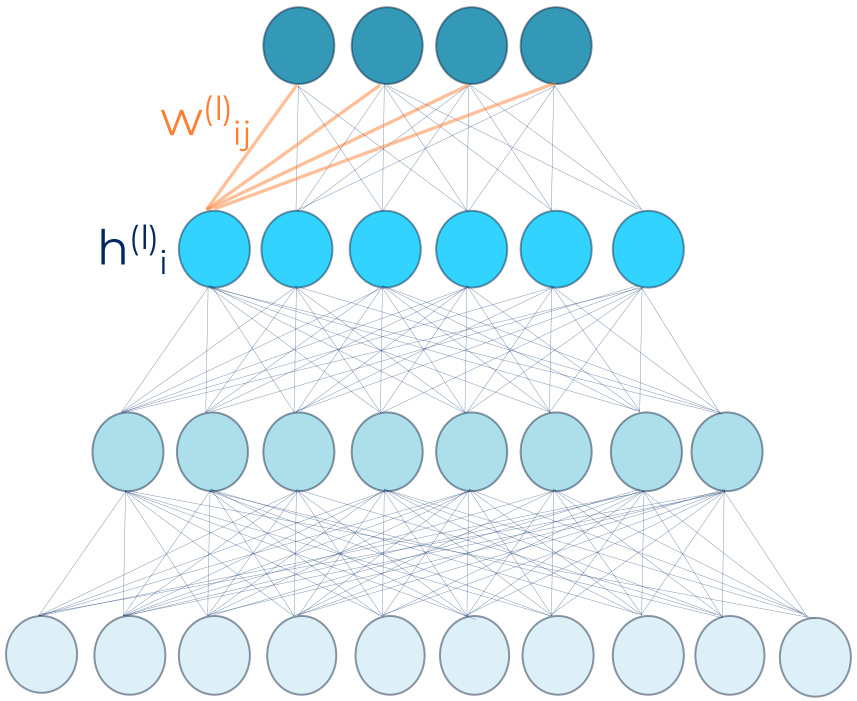
Preactivation:

$h^{(l)}_i = \sum_j x^{(l-1)}_j w^{(l)}_{ij} + b^{(l)}_i$

State: $x^{(l)}_i = \phi(h^{(l)}_i)$

ReLU: $\phi(x) = \max(x, 0)$





Neural network = cascade process

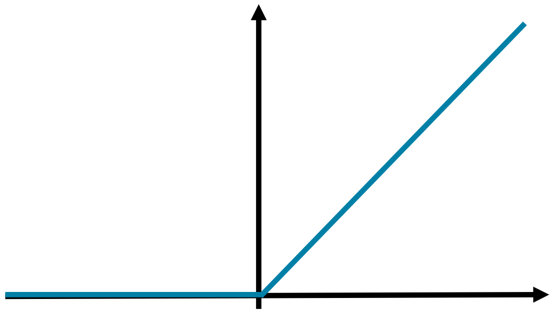
Neural network

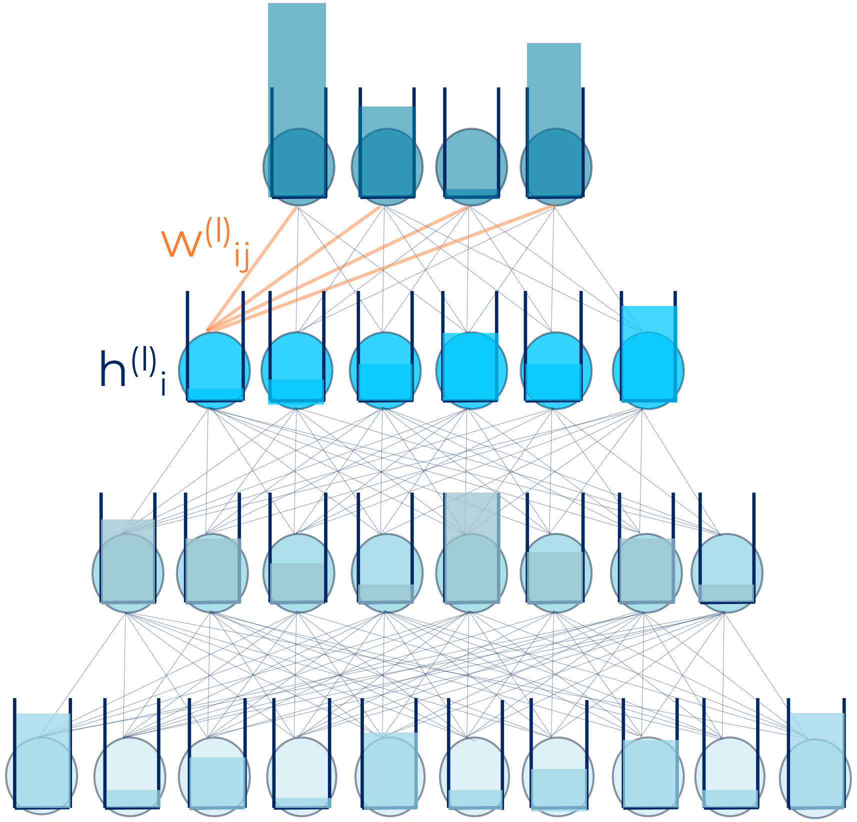
Preactivation:

$h^{(l)}_i = \sum_j x^{(l-1)}_j w^{(l)}_{ij} + b^{(l)}_i$

State: $x^{(l)}_i = \phi(h^{(l)}_i)$

ReLU: $\phi(x) = \max(x, 0)$





Neural network = cascade process

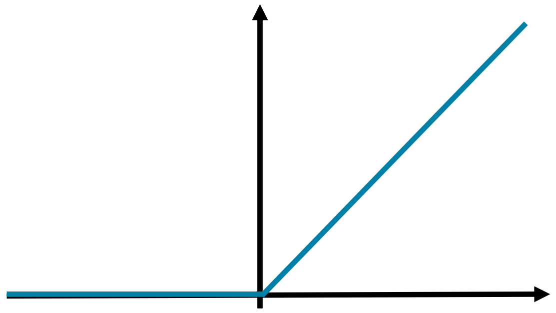
Neural network

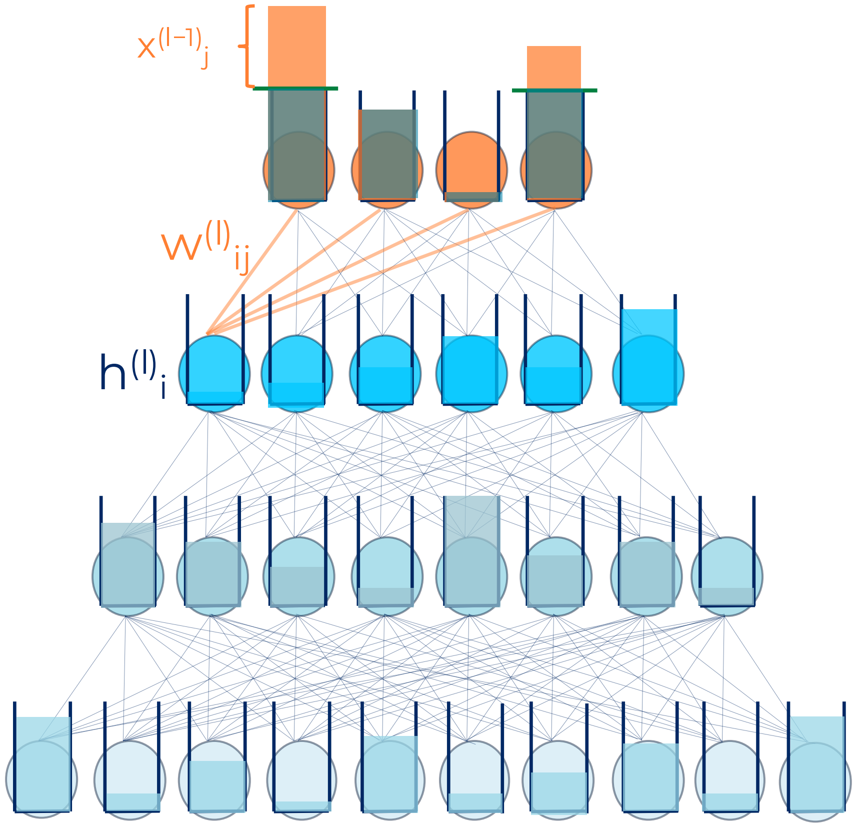
Preactivation:

$h^{(l)}_i = \sum_j x^{(l-1)}_j w^{(l)}_{ij} + b^{(l)}_i$

State: $x^{(l)}_i = \phi(h^{(l)}_i)$

ReLU: $\phi(x) = \max(x, 0)$





Neural network = cascade process

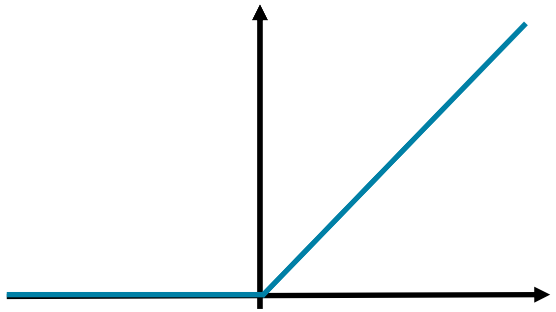
Neural network

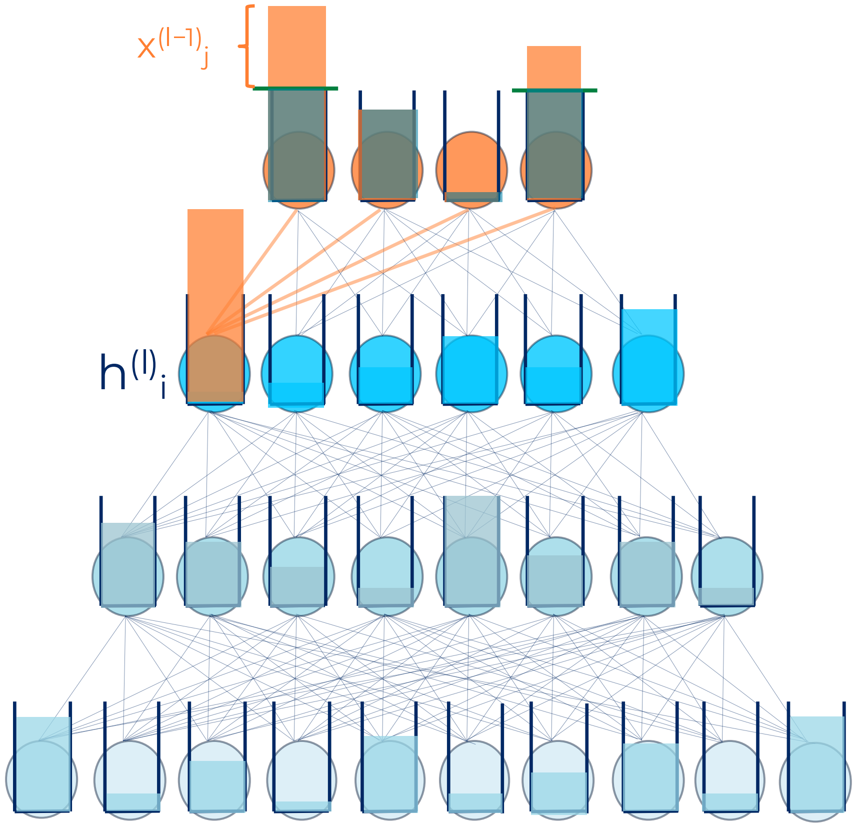
Preactivation:

$h^{(l)}_i = \sum_j x^{(l-1)}_j w^{(l)}_{ij} + b^{(l)}_i$

State: $x^{(l)}_i = \phi(h^{(l)}_i)$

ReLU: $\phi(x) = \max(x, 0)$





Neural network = cascade process

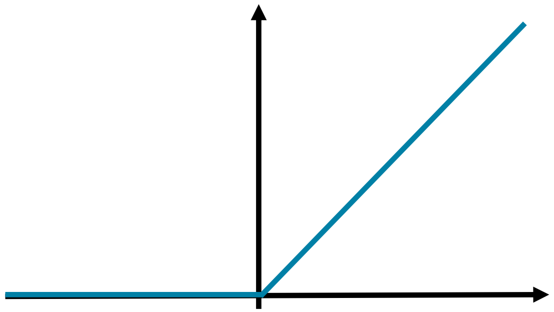
Neural network

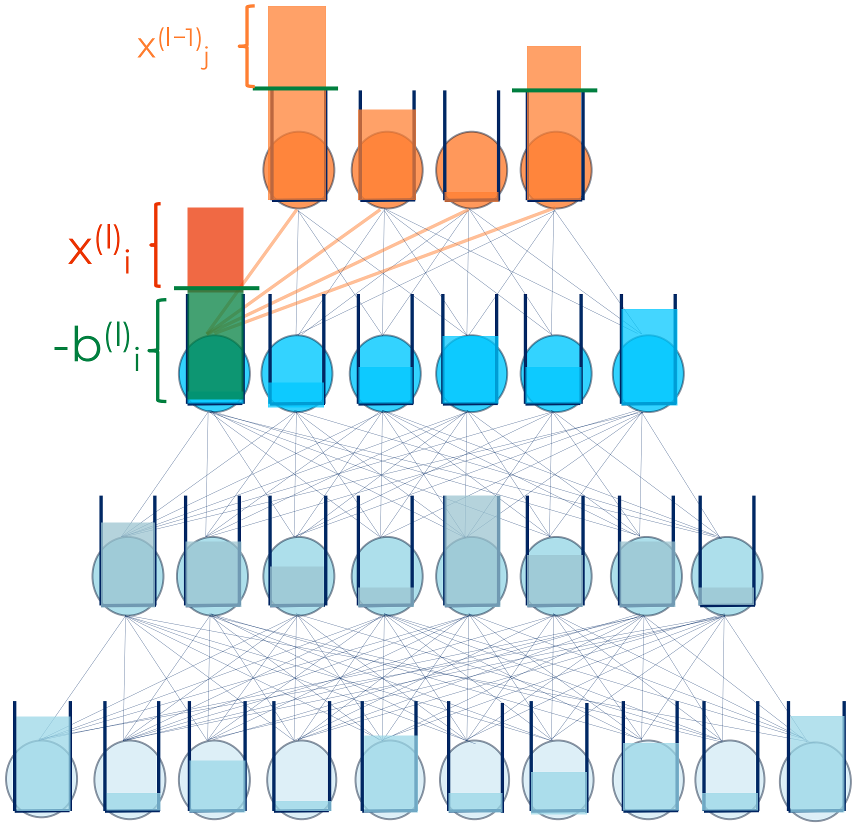
Preactivation:

$h^{(l)}_i = \sum_j x^{(l-1)}_j w^{(l)}_{ij} \; - (-b^{(l)}_i)$

State: $x^{(l)}_i = \phi(h^{(l)}_i)$

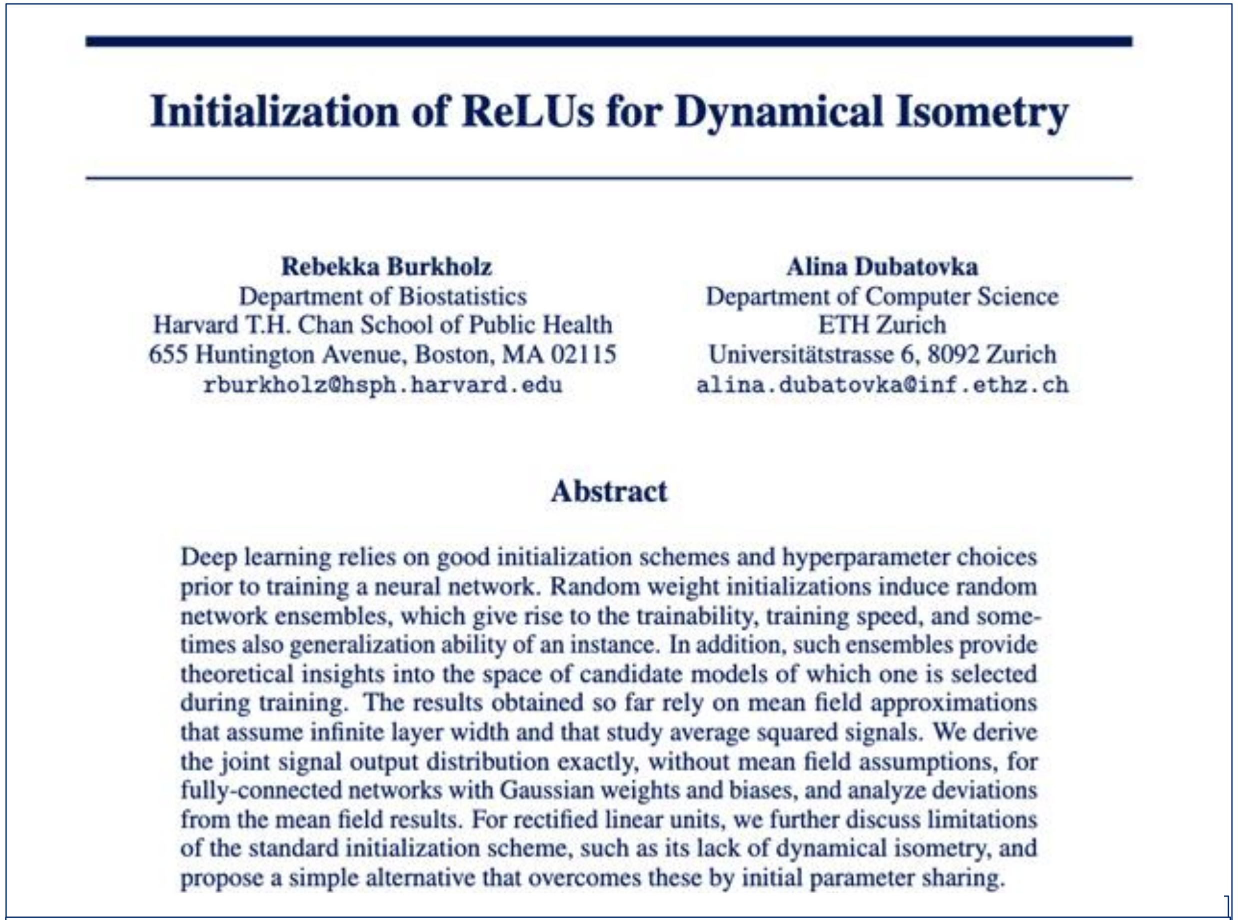
ReLU: $\phi(x) = \max(x, 0)$





Random graph ensemble of neural nets at initialization

Leveraging results from complex network science



Initialization of ReLUs for Dynamical Isometry

Rebekka Burkholz
Department of Biostatistics
Harvard T.H. Chan School of Public Health
655 Huntington Avenue, Boston, MA 02115
rburkholz@hsph.harvard.edu

Alina Dubatovka
Department of Computer Science
ETH Zurich
Universitätstrasse 6, 8092 Zurich
alina.dubatovka@inf.ethz.ch

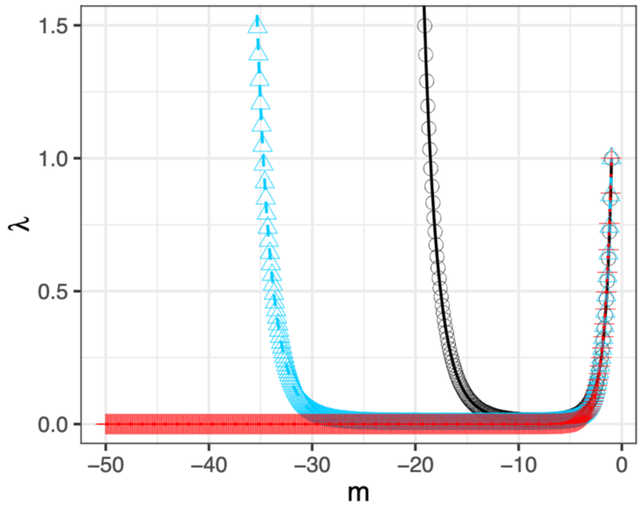
Abstract

Deep learning relies on good initialization schemes and hyperparameter choices prior to training a neural network. Random weight initializations induce random network ensembles, which give rise to the trainability, training speed, and sometimes also generalization ability of an instance. In addition, such ensembles provide theoretical insights into the space of candidate models of which one is selected during training. The results obtained so far rely on mean field approximations that assume infinite layer width and that study average squared signals. We derive the joint signal output distribution exactly, without mean field assumptions, for fully-connected networks with Gaussian weights and biases, and analyze deviations from the mean field results. For rectified linear units, we further discuss limitations of the standard initialization scheme, such as its lack of dynamical isometry, and propose a simple alternative that overcomes these by initial parameter sharing.

➡ Complex network science perspective can improve trainability of neural networks.

Track **signal propagation** at the **edge of stability**:

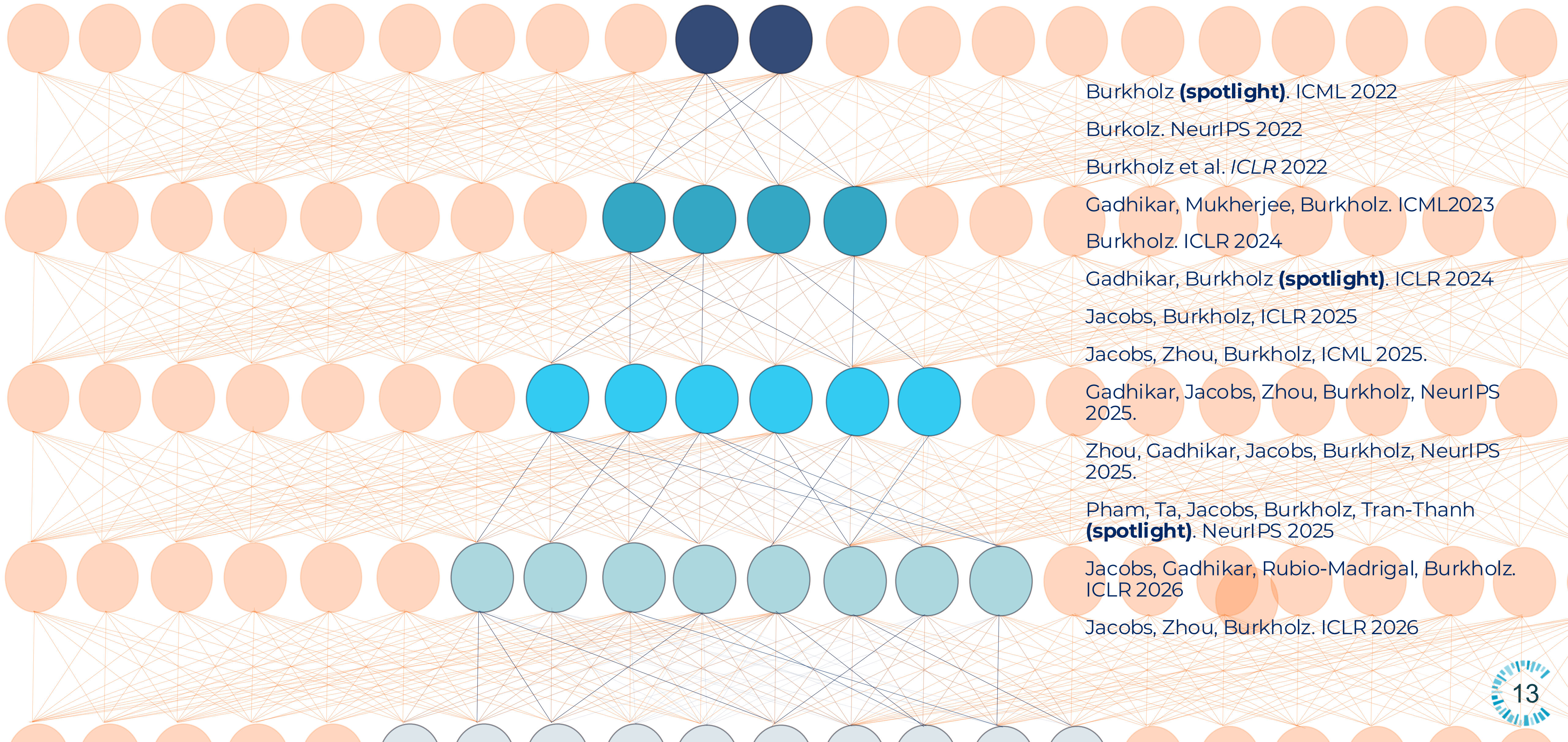
$$F_{x_i^{(l)}}(x) = \int_0^\infty dz\, p_{l-1}(z)\Phi\left(\frac{\phi^{-1}(x)}{\sqrt{\sigma_w^2 z + \sigma_b^2}}\right)$$



Spectrum of layerwise transition operator

Burkholz, Dubatovka. *NeurIPS* 2019.

Neural network sparsification by lottery tickets



Burkholz **(spotlight)**. ICML 2022

Burkolz. NeurIPS 2022

Burkholz et al. *ICLR* 2022

Gadhikar, Mukherjee, Burkholz. ICML2023

Burkholz. ICLR 2024

Gadhikar, Burkholz **(spotlight)**. ICLR 2024

Jacobs, Burkholz, ICLR 2025

Jacobs, Zhou, Burkholz, ICML 2025.

Gadhikar, Jacobs, Zhou, Burkholz, NeurIPS 2025.

Zhou, Gadhikar, Jacobs, Burkholz, NeurIPS 2025.

Pham, Ta, Jacobs, Burkholz, Tran-Thanh **(spotlight)**. NeurIPS 2025

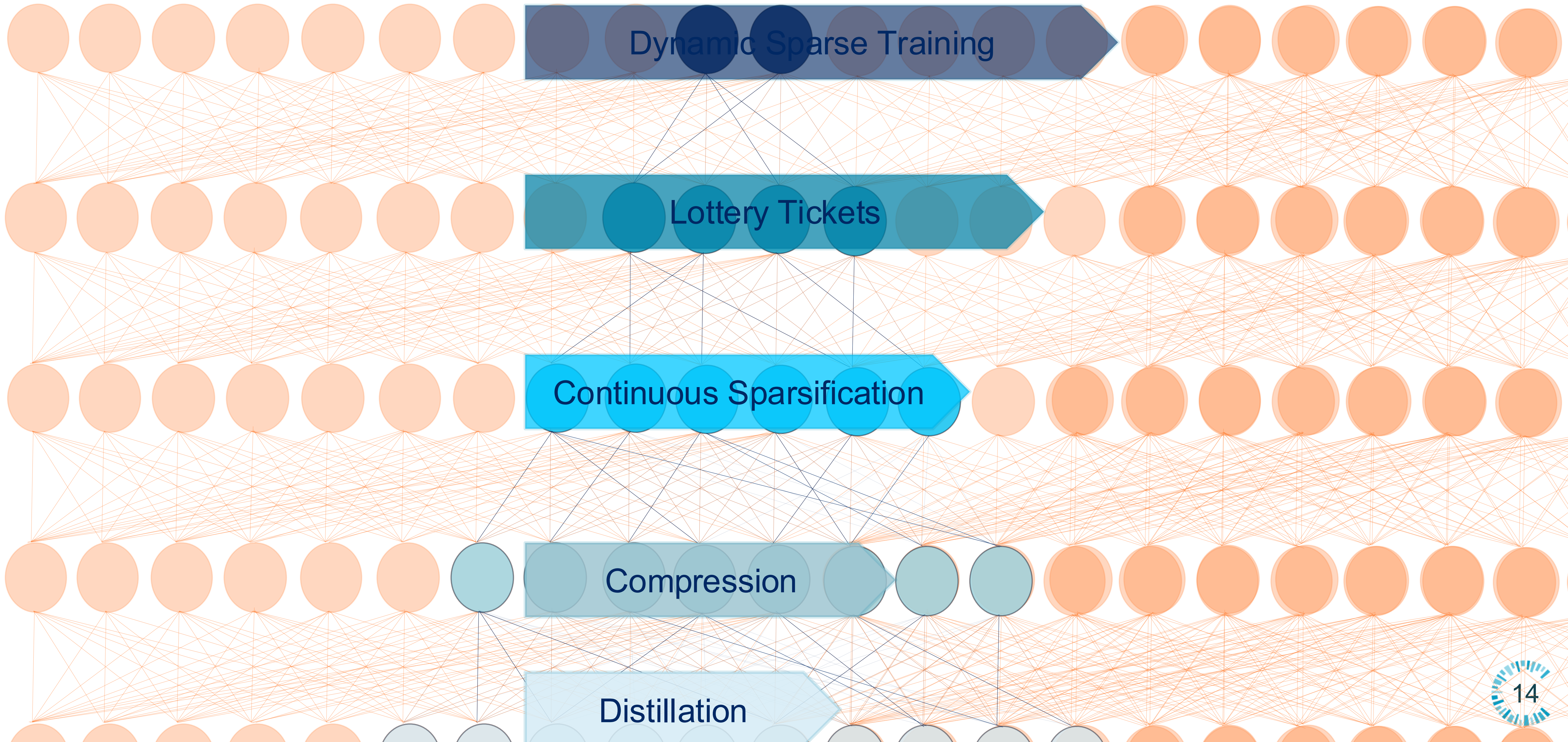
Jacobs, Gadhikar, Rubio-Madrigal, Burkholz. ICLR 2026

Jacobs, Zhou, Burkholz. ICLR 2026

Sparse complex networks

Compression ratios of 1-10%

Fischer, Burkholz. *ICLR* 2022.



Infinite width limit with sparsity

Why is **training sparse** neural networks **hard?**

And how can we **improve** it**?**

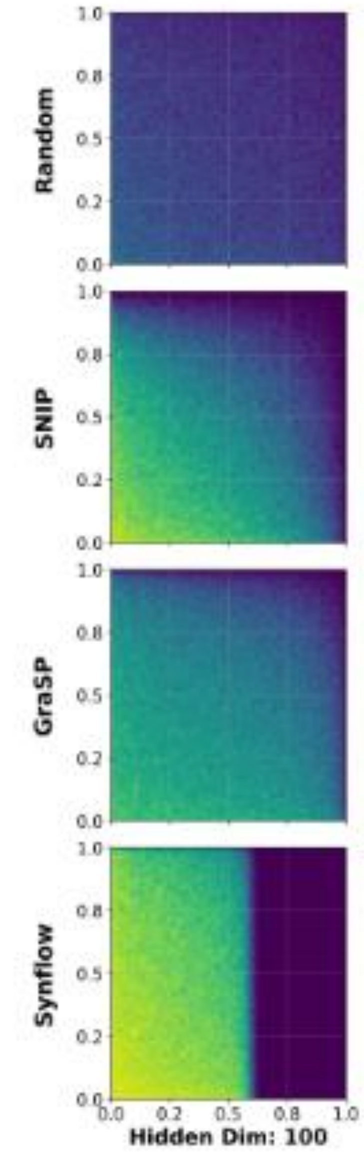
Pham, Ta, Jacobs, Burkholz, Tran-Thanh.
The Graphon Limit Hypothesis: Understanding Neural Network Pruning via Infinite Width Analysis. NeurIPS 2025 **(spotlight)**.

Infinite width limit with sparsity

Why is **training sparse** neural networks **hard?**

And how can we **improve** it**?**

Graphon Neural Tangent Kernel: $\Theta(x, x') = \nabla_\theta f(x, \theta)^\top \nabla_\theta f(x', \theta)$

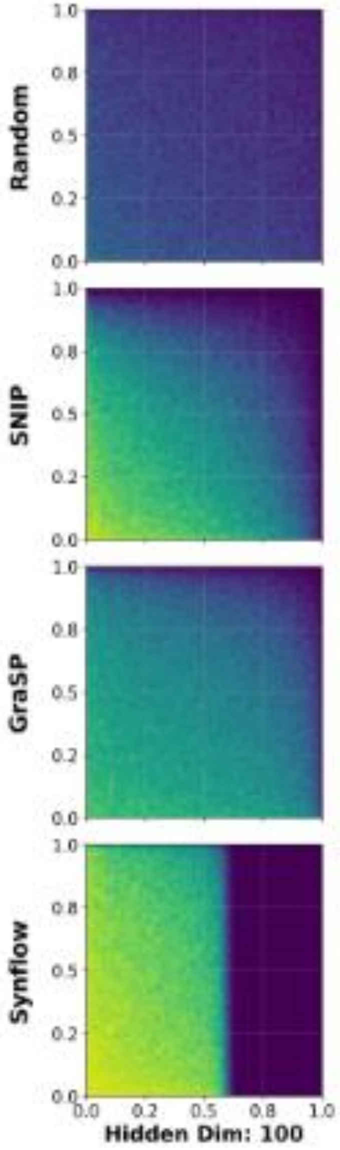


Pham, Ta, Jacobs, Burkholz, Tran-Thanh.
The Graphon Limit Hypothesis: Understanding Neural Network Pruning via Infinite Width Analysis. NeurIPS 2025 **(spotlight)**.

Infinite width limit with sparsity

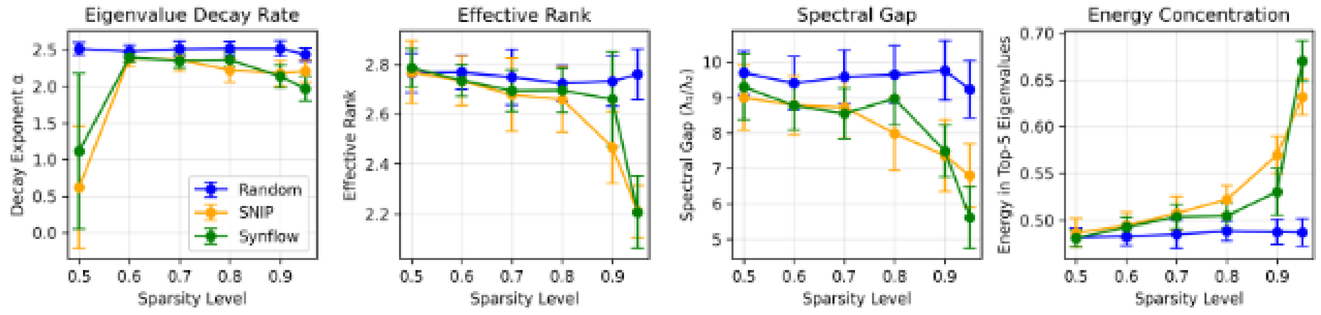
Why is **training sparse** neural networks **hard?**

And how can we **improve** it**?**



Graphon Neural Tangent Kernel: $\Theta(x, x') = \nabla_\theta f(x,\theta)^\top \nabla_\theta f(x',\theta)$

$$\Theta(x,x') = \sum_{l=1}^{L} \int_0^1 \left(\dot{\Sigma}^{(l)}(u_l, u_l, x, x') \int_{[0,1]^{L-l+1}} \prod_{m=l+1}^{L+1} \mathcal{W}^{(m)}(u_m, u_{m-1}) \dot{\Sigma}^{(m)}(u_m, u_m, x, x')\, d\mathbf{u}_{l+1} \right) \cdot \left(\int_0^1 \Sigma^{(l-1)}(u_{l-1}, u_{l-1}, x, x') du_{l-1} \right) du_l, \quad (4)$$

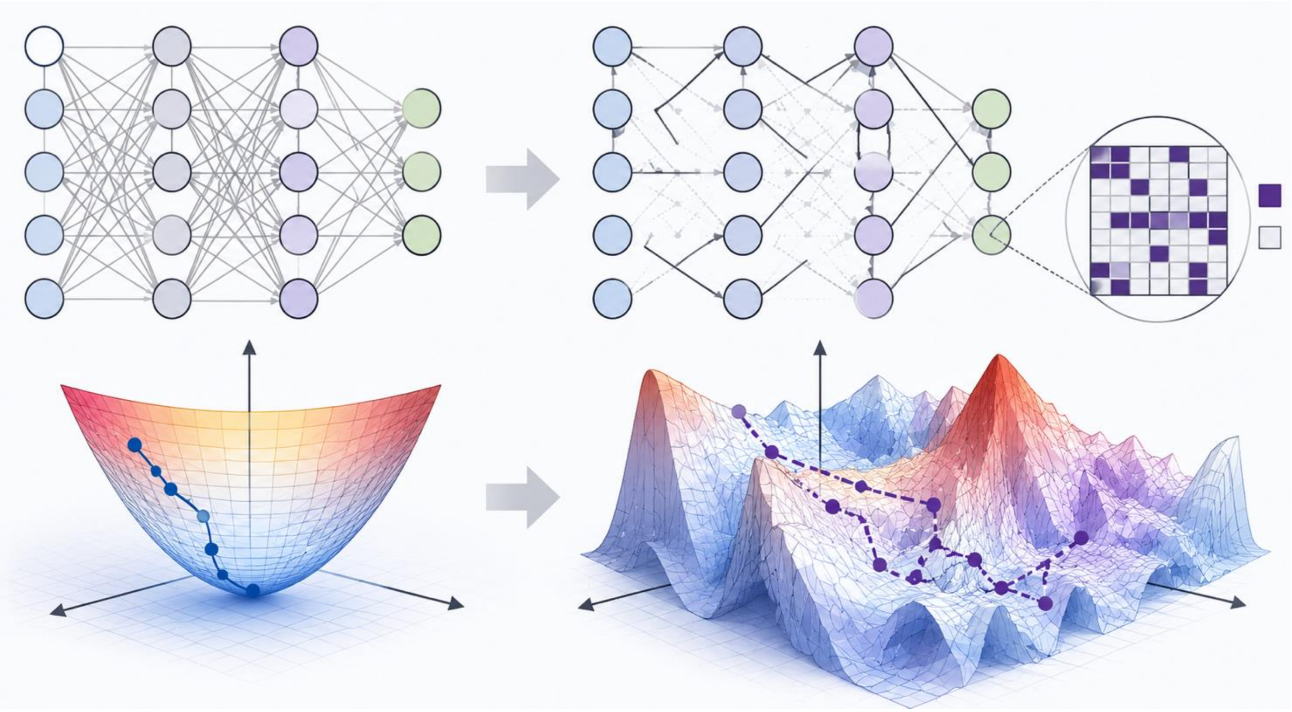


Conclusion: The optimization problem becomes ill conditioned.

Pham, Ta, Jacobs, Burkholz, Tran-Thanh.
The Graphon Limit Hypothesis: Understanding Neural Network Pruning via Infinite Width Analysis. NeurIPS 2025 **(spotlight)**.

How to handle sparsity in deep learning?

Still open problem: **Sparsity** aware **optimization**



Gadhikar, Burkholz. Masks, Signs, and Learning Rate Rewinding. ICLR 2024 **(spotlight)**.

Gadhikar, Jacobs, Zhou, Burkholz. Sign-In to the Lottery: Reparameterizing Sparse Training. NeurIPS 2025

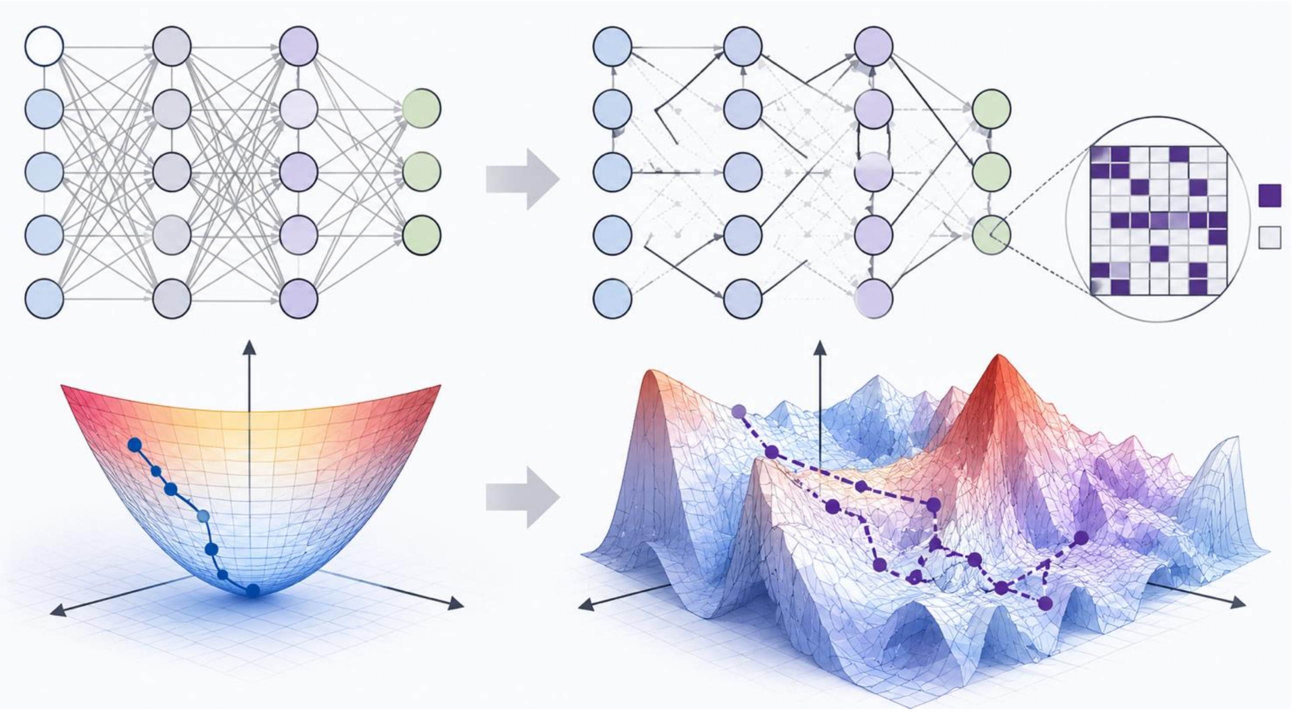
Gadhikar, Jacobs, Rubio-Madrigal, Burkholz. A Hyperbolic Step to Regulate Implicit Bias. ICLR 2026

Jacobs, Jain, Burkholz. HORST: Composing Optimizer Geometries for Sparse Transformer Training. *HiLD*@ICML 2026

Adnan, Jain, Jacobs, Sharma, Krishnan, Burkholz, Ioannou. SparseOpt: Addressing Normalization-induced Gradient Skew in Sparse Training. ICML 2026

How to handle sparsity in deep learning?

Still open problem: **Sparsity** aware **optimization**



Y Zhang, H Bai, H Lin, J Zhao, L Hou, CV Cannistraci. Plug-and-play: An efficient post-training pruning method for large language models. ICLR 2024.

Zhao, Muscoloni, Michieli, Zhang, Cannistraci. Adaptive Cannistraci-Hebb Network Automata Modelling of Complex Networks for Path-based Link Prediction. NeurIPS 2025.

Wu, Zhang, Zhao, Cannistraci. Alignment-Enhanced Integration of Connectivity and Spectral Sparsity in Dynamic Sparse Training of LLM. ICLR 2026

Hua, Zhang, Zhang, Gu, You, Xiong, Cannistraci, Chen. Cannistraci-Hebb Training on Ultra-Sparse Spiking Neural Networks. ICLR 2026

Gadhikar, Burkholz. Masks, Signs, and Learning Rate Rewinding. ICLR 2024 **(spotlight)**.

Gadhikar, Jacobs, Zhou, Burkholz. Sign-In to the Lottery: Reparameterizing Sparse Training. NeurIPS 2025

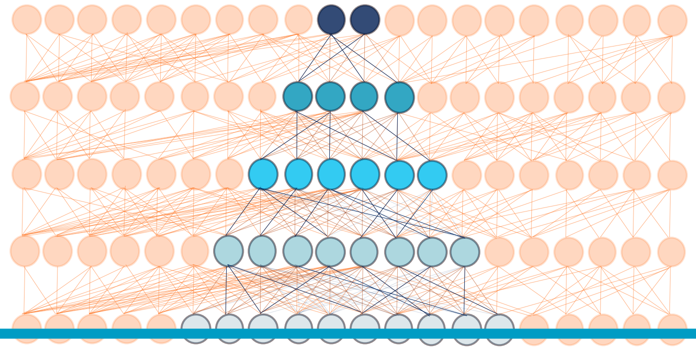
Gadhikar, Jacobs, Rubio-Madrigal, Burkholz. A Hyperbolic Step to Regulate Implicit Bias. ICLR 2026

Jacobs, Jain, Burkholz. HORST: Composing Optimizer Geometries for Sparse Transformer Training. *HiLD*@ICML 2026

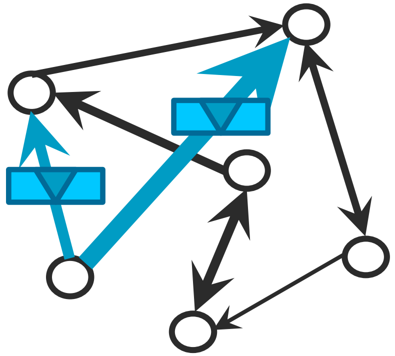
Adnan, Jain, Jacobs, Sharma, Krishnan, Burkholz, Ioannou. SparseOpt: Addressing Normalization-induced Gradient Skew in Sparse Training. ICML 2026

Outline: Complex network science for AI

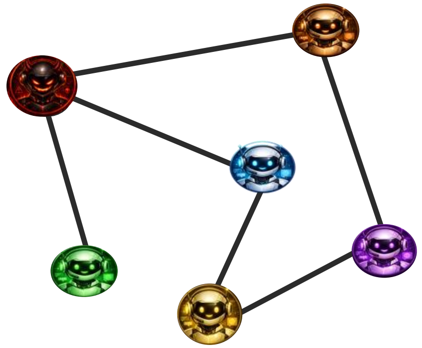
1. **Sparse** deep learning
    - Random graph **ensembles**
    - Learning networks (by sparsification)



2. **Graph** Neural Networks:
    - Spectral rewiring (**Braess** paradox)
    - Features by running cascade **process**
    - Learning **on** networks

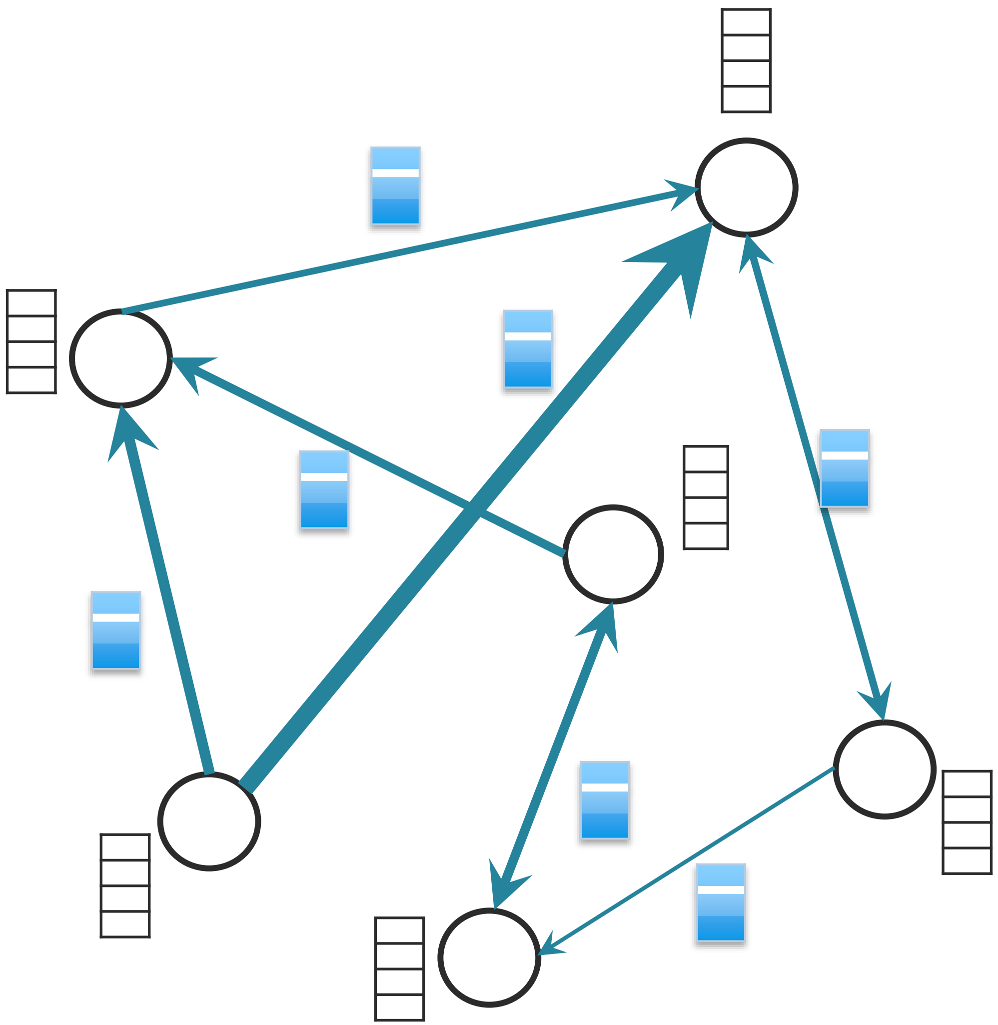


3. **Agentic** networks
    - Robustness of agentic LLM system
    - Friedkin-Johnsen **opinion formation**
    - Process/network weights **emerge**



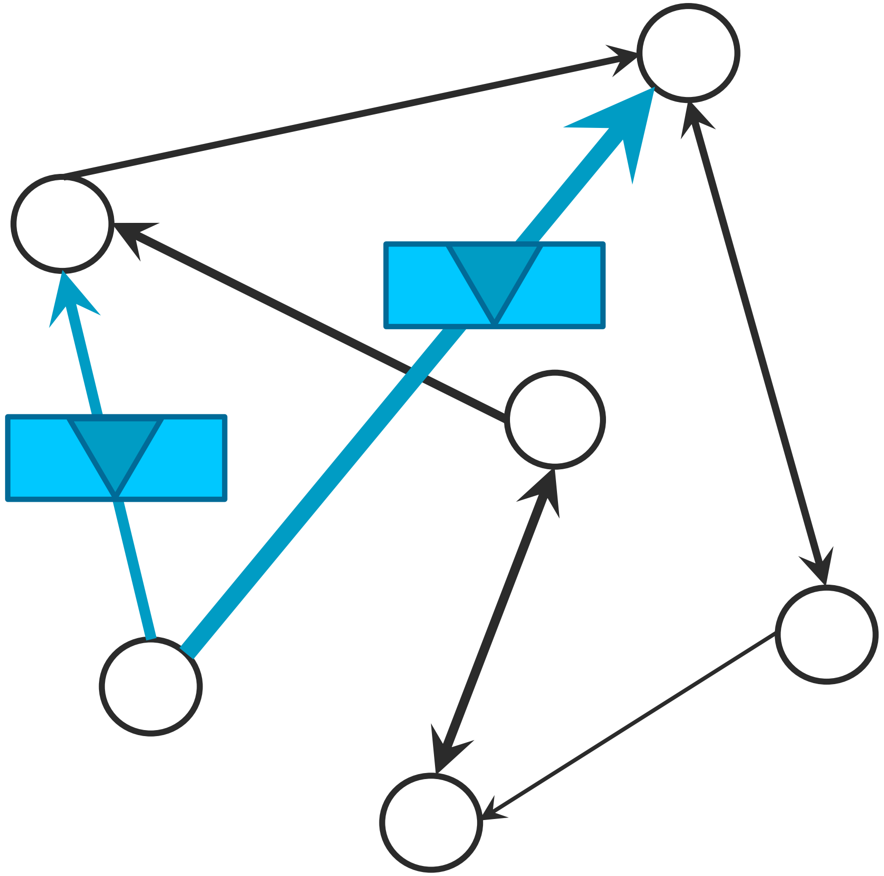
GNNs equate data and computational structure

- GNNs learn from **graph**-structured **data**

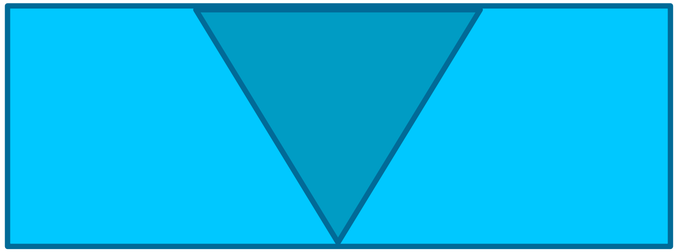


GNNs learn message passing algorithm

- GNNs learn from **graph**-structured **data**
- Training GNNs → message passing on input graph

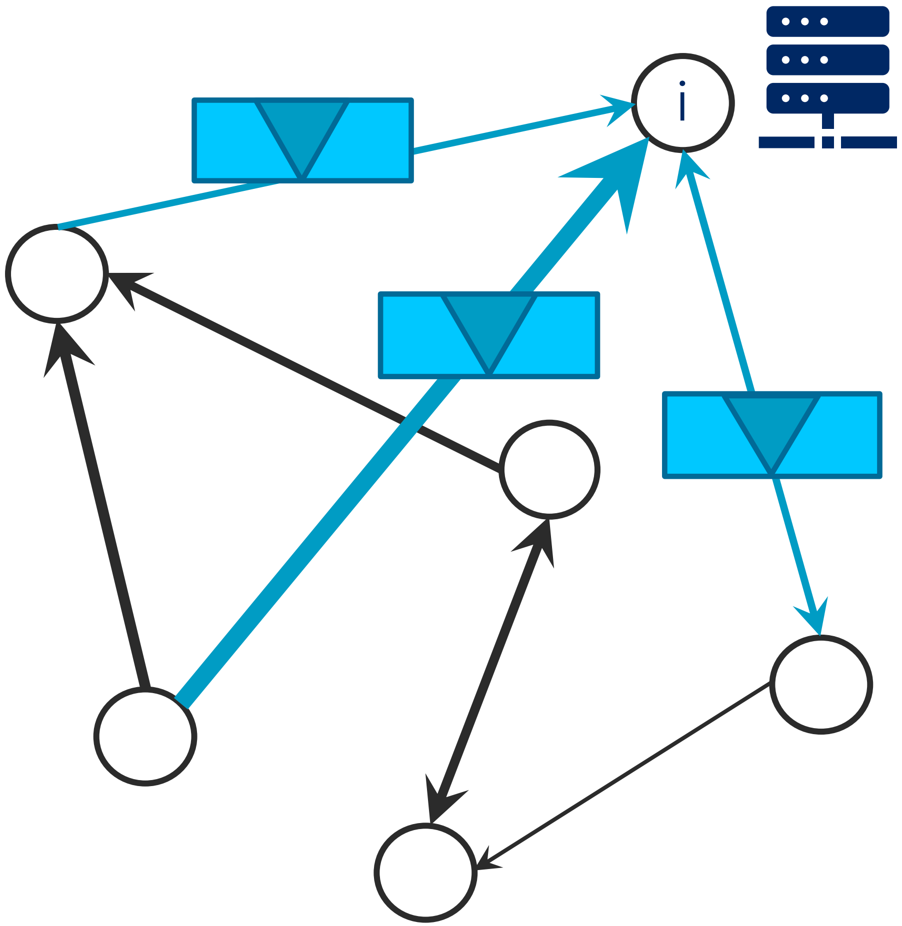


Messages:

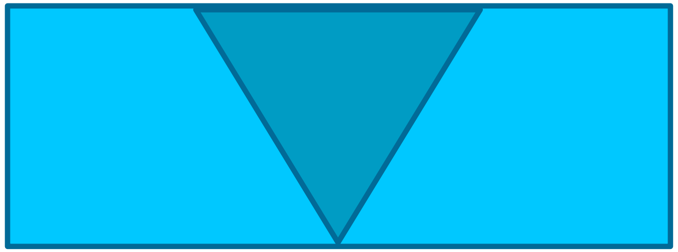


GNNs learn message passing algorithm

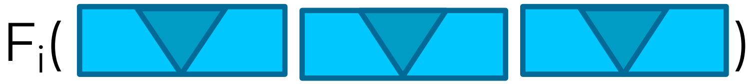
- GNNs learn from **graph**-structured **data**
- Training GNNs → message passing on input graph



Messages:



Aggregation/computation in node:



Rewiring the input graph

Spectral Graph Pruning Against Over-Squashing and Over-Smoothing

Adarsh Jamadandi[*1,2] adarsh.jam@gmail.com

Celia Rubio-Madrigal[*2] celia.rubio-madrigal@cispa.de

Rebekka Burkholz[2] burkholz@cispa.de

[1] Universität des Saarlandes
[2] CISPA Helmholtz Center for Information Security

Abstract

Message Passing Graph Neural Networks are known to suffer from two problems that are sometimes believed to be diametrically opposed: *over-squashing* and *over-smoothing*. The former results from topological bottlenecks that hamper the information flow from distant nodes and are mitigated by spectral gap maximization, primarily, by means of edge additions. However, such additions often promote over-smoothing that renders nodes of different classes less distinguishable. Inspired by the Braess phenomenon, we argue that deleting edges can address over-squashing and over-smoothing simultaneously. This insight explains how edge deletions can improve generalization, thus connecting spectral gap optimization to a seemingly disconnected objective of reducing computational resources by pruning graphs for lottery tickets. To this end, we propose a computationally effective spectral gap optimization framework to add or delete edges and demonstrate its effectiveness on the long range graph benchmark and on larger heterophilous datasets.

NeurIPS 2024

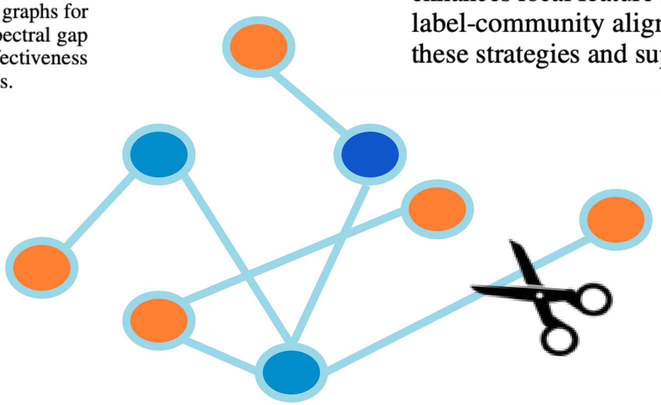
GNNs Getting ComFy: Community and Feature Similarity Guided Rewiring

Anonymous authors
Paper under double-blind review

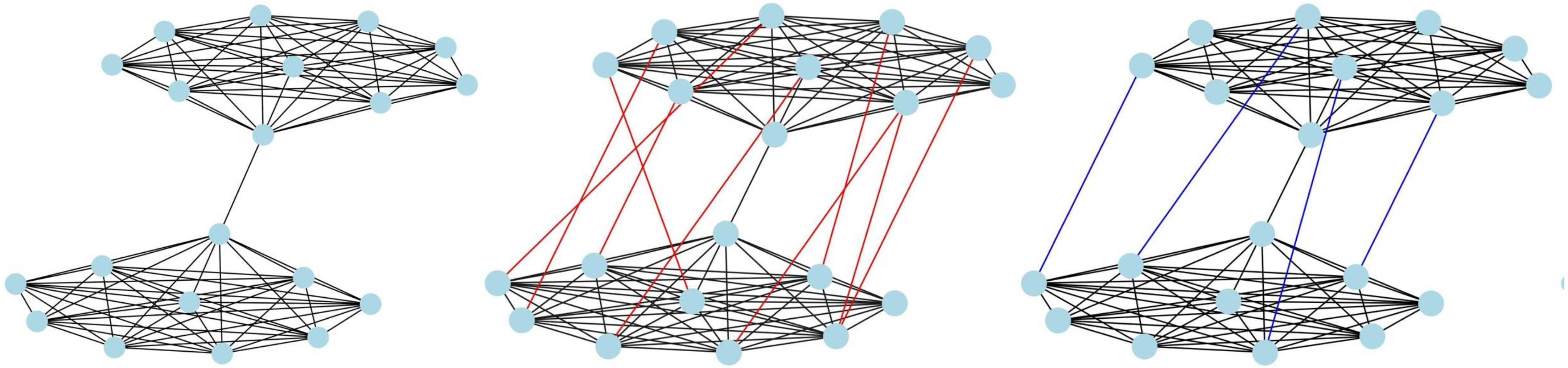
Abstract

Maximizing the spectral gap through graph rewiring has been proposed to enhance the performance of message-passing graph neural networks (GNNs) by addressing over-squashing. However, as we show, minimizing the spectral gap can also improve generalization. To explain this, we analyze how rewiring can benefit GNNs within the context of stochastic block models. Since spectral gap optimization primarily influences community strength, it improves performance when the community structure aligns with node labels. Building on this insight, we propose three distinct rewiring strategies that explicitly target community structure, node labels, and their alignment: (a) community structure-based rewiring (ComMa), a more computationally efficient alternative to spectral gap optimization that achieves similar goals; (b) feature similarity-based rewiring (FeaSt), which focuses on maximizing global homophily; and (c) a hybrid approach (ComFy), which enhances local feature similarity while preserving community structure to optimize label-community alignment. Extensive experiments confirm the effectiveness of these strategies and support our theoretical insights.

ICLR 2025



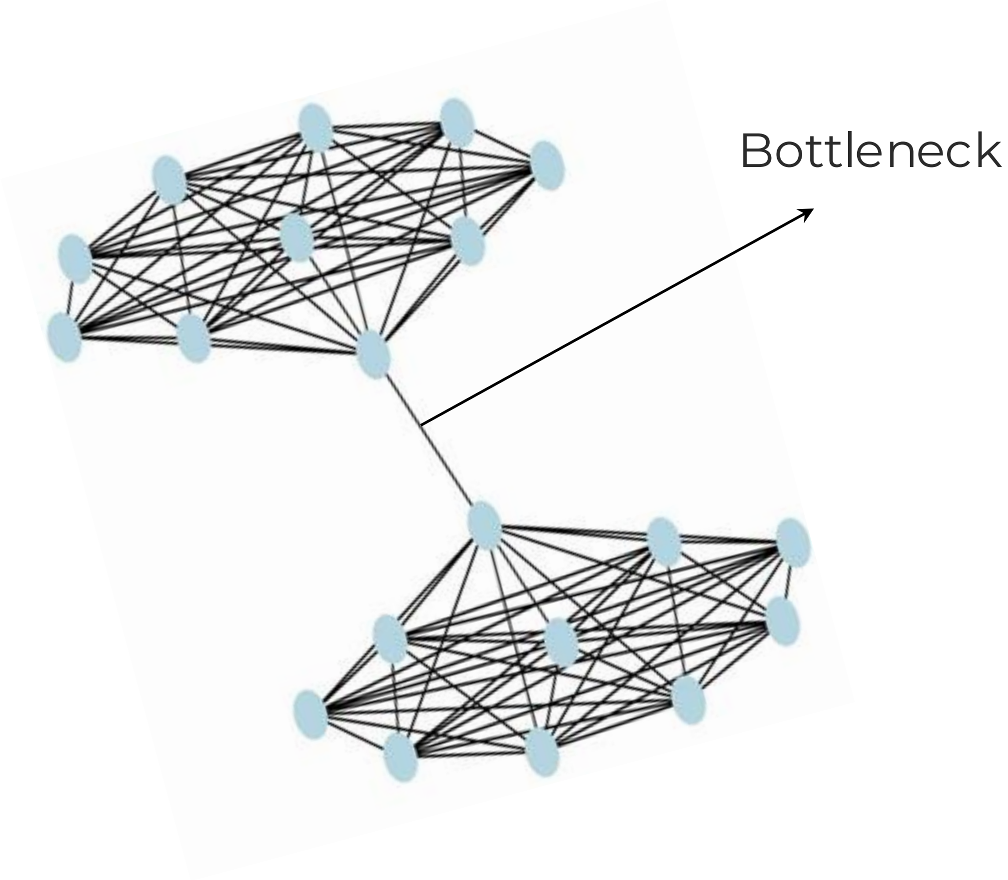
Not all graphs propogate information easily

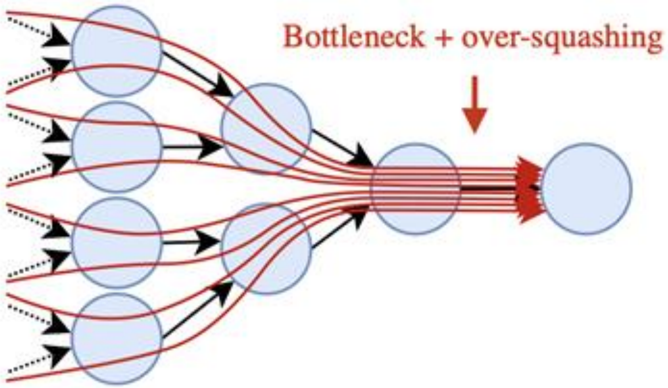


- Bottlenecks ⟶ over-squashing
- Common mitigation: add few edges

Over-squashing

Bottlenecks obstruct the flow of information during message passing





Normalized Laplacian: $L_G = I - D^{-1/2} A\, D^{-1/2}$

Spectral gap: $\lambda_1 - \lambda_0 \ (= \lambda_1)$

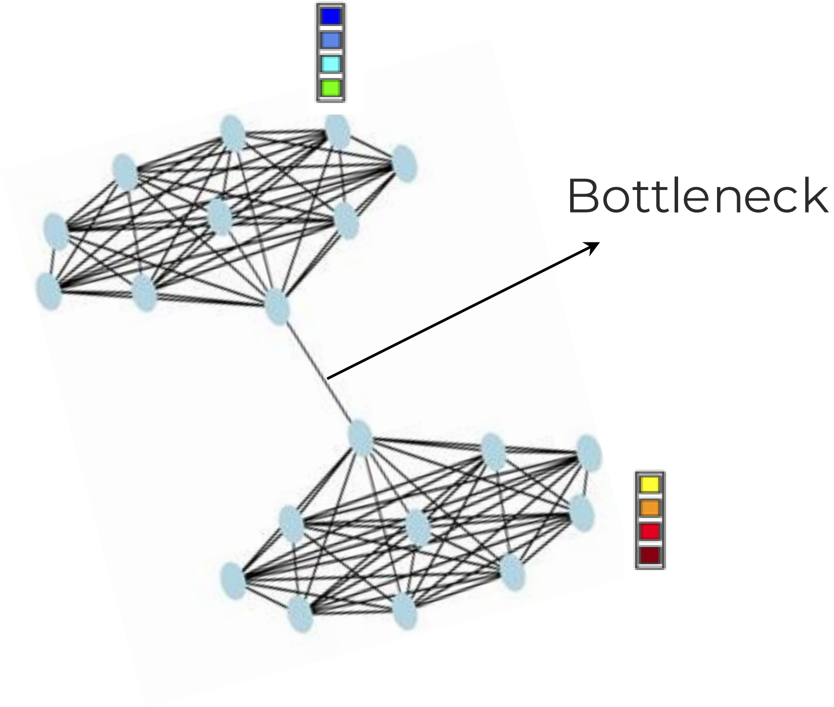
Small spectral gap ≡ bottlenecks

Trade-off?

Do we need to balance over-squashing and over-smoothing?

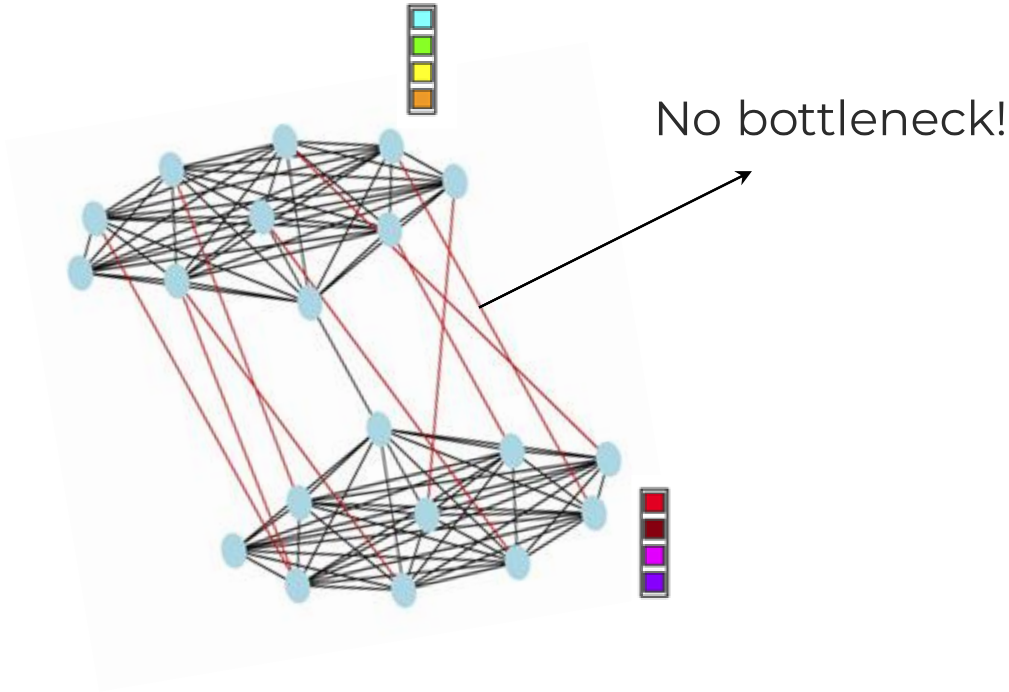
Over-squashing!

(info. prop. hampered)



Over-smoothing?

(nodes too similar)



Braess Paradox

Adding this extra road causes delays (Braess, 1968)

- Not all edge additions increase connectivity
- Not all edge deletions decrease connectivity

(Eldan et al., 2017) ⇒ there is a Braess Paradox for the **spectral gap** of the normalized Laplacian

We can

1. DELETE edges
2. INCREASE spectral gap λ_1

(mitigating over-squashing)

Random walker analogy to rewire graph

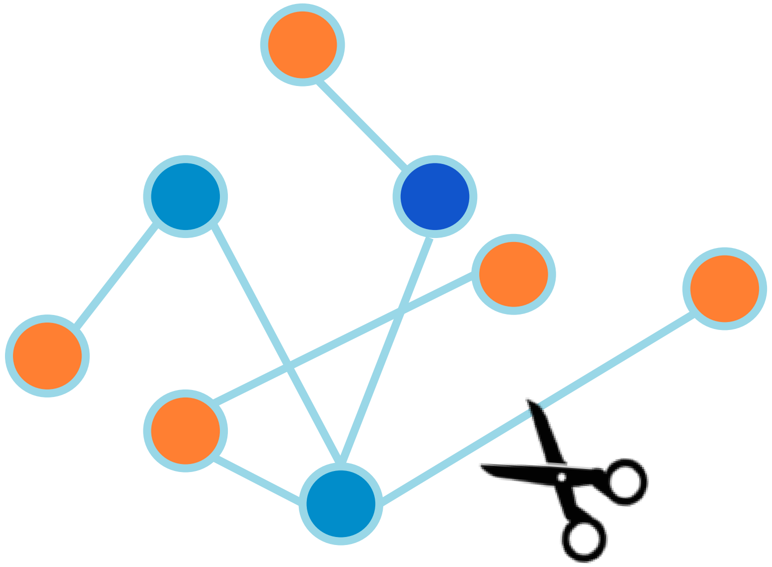
- We can **delete** edges and still fight **over-squashing**
- and, simultaneously, **over-smoothing.**
- **Note:** Random walker still converges quickly (⇒ smoothing is still going on).

Table 3: Node classification on Amazon-Ratings.

Method	#EdgesAdded	Accuracy	#EdgesDeleted	Accuracy	Layers
GCN	-	47.20±0.33	-	47.20±0.33	10
GCN+FoSR	25	49.68±0.73	-	-	10
GCN+Eldan	25	48.71±0.99	100	**50.15±0.50**	10
GCN+ProxyGap	10	49.72±0.41	50	**49.75±0.46**	10
GAT	-	47.43±0.44	-	47.43±0.44	10
GAT+FoSR	25	51.36±0.62	-	-	10
GAT+Eldan	25	51.68±0.60	50	**51.80±0.27**	10
GAT+ProxyGap	20	49.06±0.92	100	**51.72±0.30**	10
GCN	-	47.32±0.59	-	47.32±0.59	20
GCN+FoSR	100	49.57±0.39	-	-	20
GCN+Eldan	50	**49.66±0.31**	20	48.32±0.76	20
GCN+ProxyGap	50	49.48±0.59	500	**49.58±0.59**	20
GAT	-	47.31±0.46	-	47.31±0.46	20
GAT+FoSR	100	51.31±0.44	-	-	20
GAT+Eldan	20	51.40±0.36	20	**51.64±0.44**	20
GAT+ProxyGap	50	47.53±0.90	20	**51.69±0.46**	20

Table 4: Node classification on Minesweeper.

Method	#EdgesAdded	Accuracy	#EdgesDeleted	Test ROC	Layers
GCN	-	88.57± 0.64	-	88.57± 0.64	10
GCN+FoSR	50	90.15±0.55	-	-	10
GCN+Eldan	100	**90.11±0.50**	50	89.49±0.60	10
GCN+ProxyGap	20	**89.59±0.50**	20	89.57±0.49	10
GAT	-	93.60±0.64	-	93.60±0.64	10
GAT+FoSR	100	93.14±0.43	-	-	10
GAT+Eldan	50	93.26±0.48	100	**93.82±0.56**	10
GAT+ProxyGap	20	93.60±0.69	20	**93.65±0.84**	10
GCN	-	87.41±0.65	-	87.41±0.65	20
GCN+FoSR	100	89.64±0.55	-	-	20
GCN+Eldan	72	**89.70±0.57**	10	88.90±0.44	20
GCN+ProxyGap	20	**89.46±0.50**	50	89.35±0.30	20
GAT	-	93.92±0.52	-	93.92±0.52	20
GAT+FoSR	50	93.56±0.64	-	-	20
GAT+Eldan	10	93.92±0.44	20	**95.48±0.64**	20
GAT+ProxyGap	20	**94.89±0.67**	20	94.64±0.81	20



Do we have to learn graph convolutions?

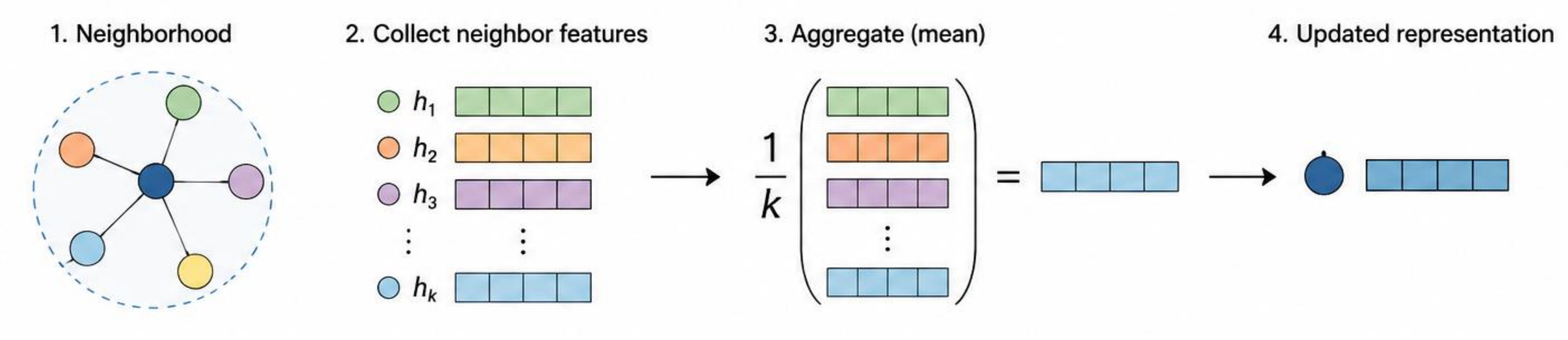
Rubio-Madrigal, Burkholz. Fixed Aggregation Features Can Rival GNNs. ICML 2026.

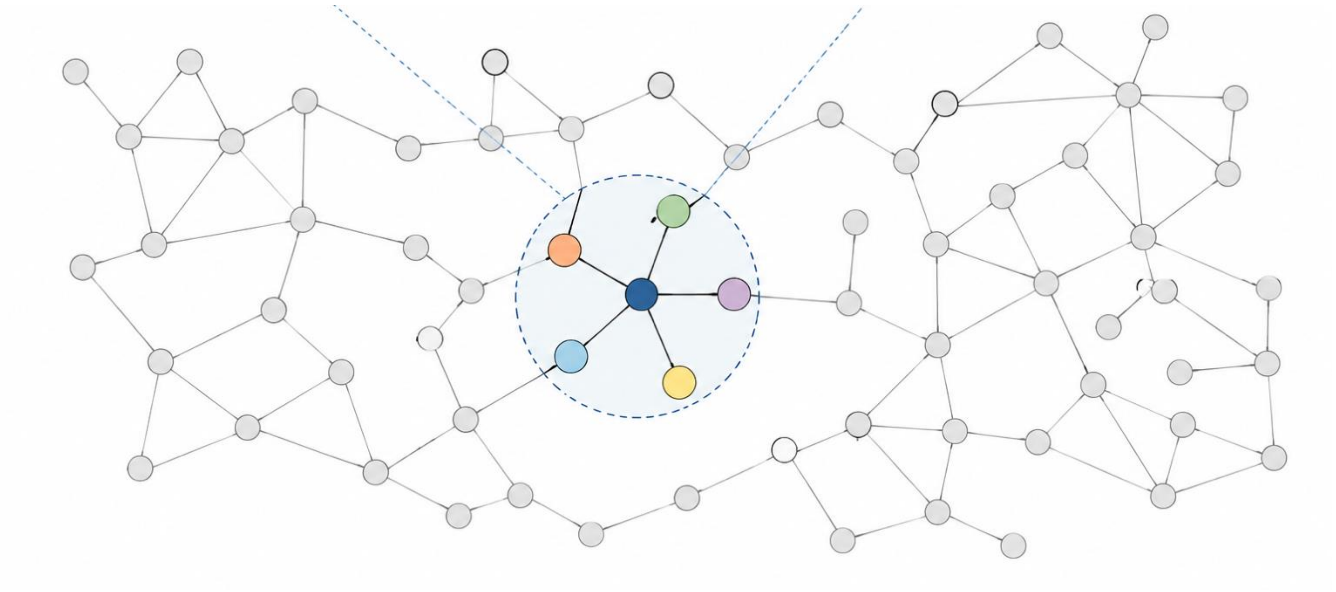
Do we have to learn graph convolutions?

- Anwer: No (mostly)!

Rubio-Madrigal, Burkholz. Fixed Aggregation Features Can Rival GNNs. ICML 2026.

Do we have to learn graph convolutions?





- Anwer: No (mostly)!
- Can use simple process instead.
- No training needed!

Rubio-Madrigal, Burkholz. Fixed Aggregation Features Can Rival GNNs. ICML 2026.

Do we have to learn graph convolutions?

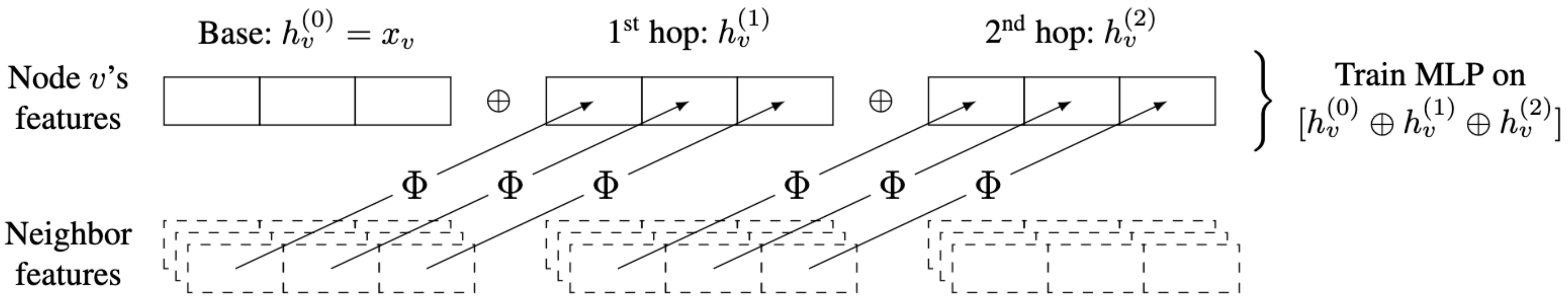


Table 1. Test accuracy on node classification: FAFs against classic GNNs.

Dataset	computer	photo	ratings	chameleon	citeseer	coauthor-cs	coauthor-physics
GCN	93.58 ± 0.44	95.77 ± 0.27	53.86 ± 0.48	**44.62 ± 4.50**	**72.72 ± 0.45**	95.73 ± 0.15	**97.47 ± 0.08**
GAT	93.91 ± 0.22	96.45 ± 0.37	**55.51 ± 0.55**	42.90 ± 5.47	71.82 ± 0.65	96.14 ± 0.08	97.12 ± 0.13
SAGE	93.31 ± 0.17	96.17 ± 0.44	55.26 ± 0.27	43.11 ± 4.73	71.82 ± 0.81	**96.21 ± 0.10**	97.10 ± 0.09
$FAF_{bestval}$	**94.01 ± 0.21**	**96.54 ± 0.13**	55.09 ± 0.24	42.96 ± 2.45	70.48 ± 1.24	95.37 ± 0.17	97.05 ± 0.18

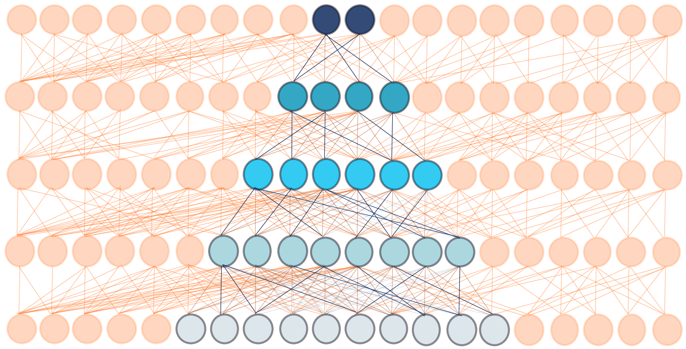
Dataset	cora	minesweeper	pubmed	questions	roman-empire	squirrel	wikics
GCN	**84.38 ± 0.81**	97.48 ± 0.06	80.00 ± 0.77	78.44 ± 0.23	**91.05 ± 0.15**	44.26 ± 1.22	80.06 ± 0.81
GAT	83.02 ± 1.21	97.00 ± 1.02	79.80 ± 0.94	77.72 ± 0.71	90.38 ± 0.49	39.31 ± 2.42	**81.01 ± 0.23**
SAGE	83.18 ± 0.93	**97.72 ± 0.70**	77.42 ± 0.40	76.75 ± 1.07	90.41 ± 0.10	40.22 ± 1.47	80.57 ± 0.42
$FAF_{bestval}$	82.84 ± 0.63	90.00 ± 0.39	**80.96 ± 1.06**	**78.69 ± 0.50**	78.11 ± 0.38	**44.59 ± 1.62**	80.25 ± 0.34

- Fixed Aggregation Features (FAFs) can turn graph struct. data into tabular data
- Sum/mean aggregations have strong inductive bias

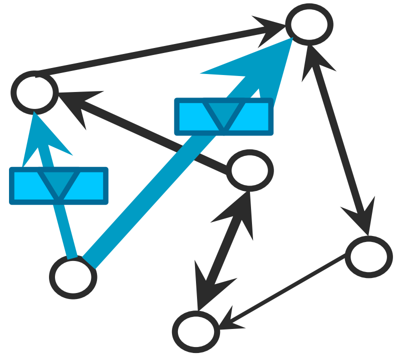
Rubio-Madrigal, Burkholz. Fixed Aggregation Features Can Rival GNNs. ICML 2026.

Outline: Complex network science for AI

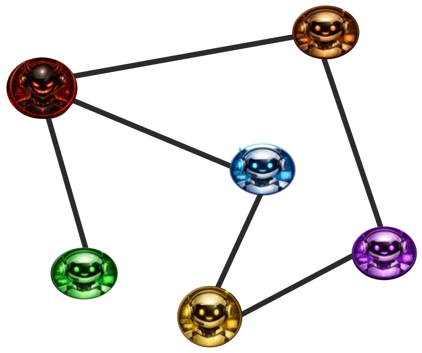
1. **Sparse** deep learning
    - Random graph **ensembles**
    - Learning networks (by sparsification)



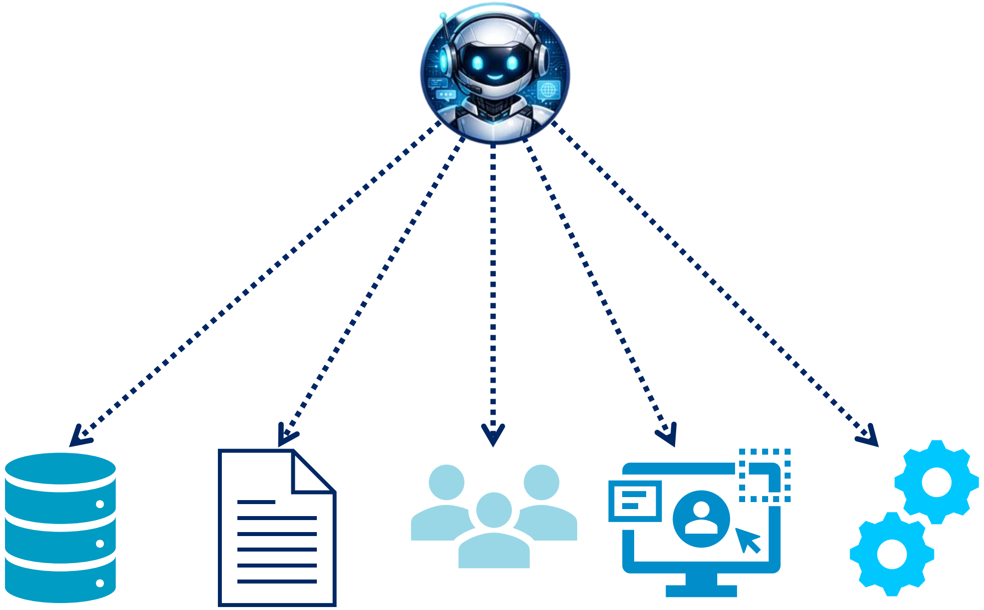
2. **Graph** Neural Networks:
    - Spectral rewiring (**Braess** paradox)
    - Features by running cascade **process**
    - Learning **on** networks



3. **Agentic** networks
    - Robustness of agentic LLM system
    - Friedkin-Johnsen **opinion formation**
    - Process/network weights **emerge**

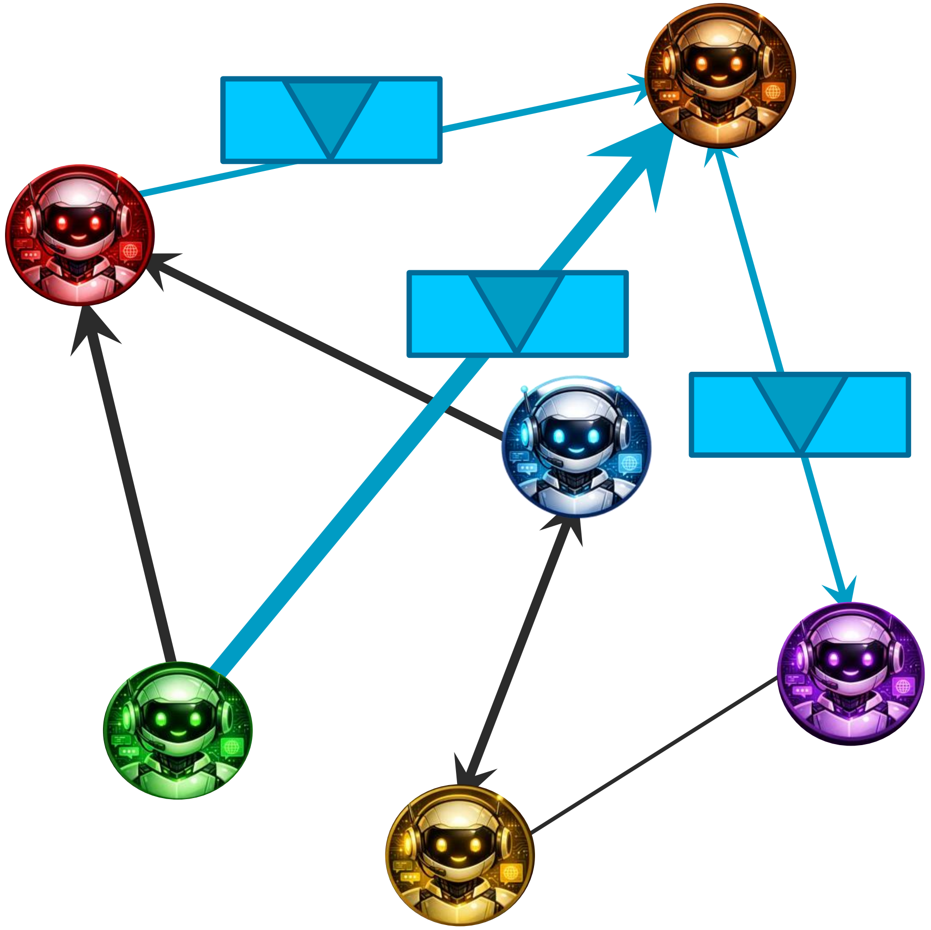


Multi-Agent LLM Systems



- LLM agent:
  - Individual objectives
  - Unique data access
  - Specialized expertise/competency
  - Process information
  - Take actions
  - Use tools
  - Make decisions
- Multi-agent system (LLM-MAS):
  - **Multiple, heterogenous LLM agents**
  - Interact via (dynamic) graph (e.g. send messages)

Multi-Agent LLM Systems

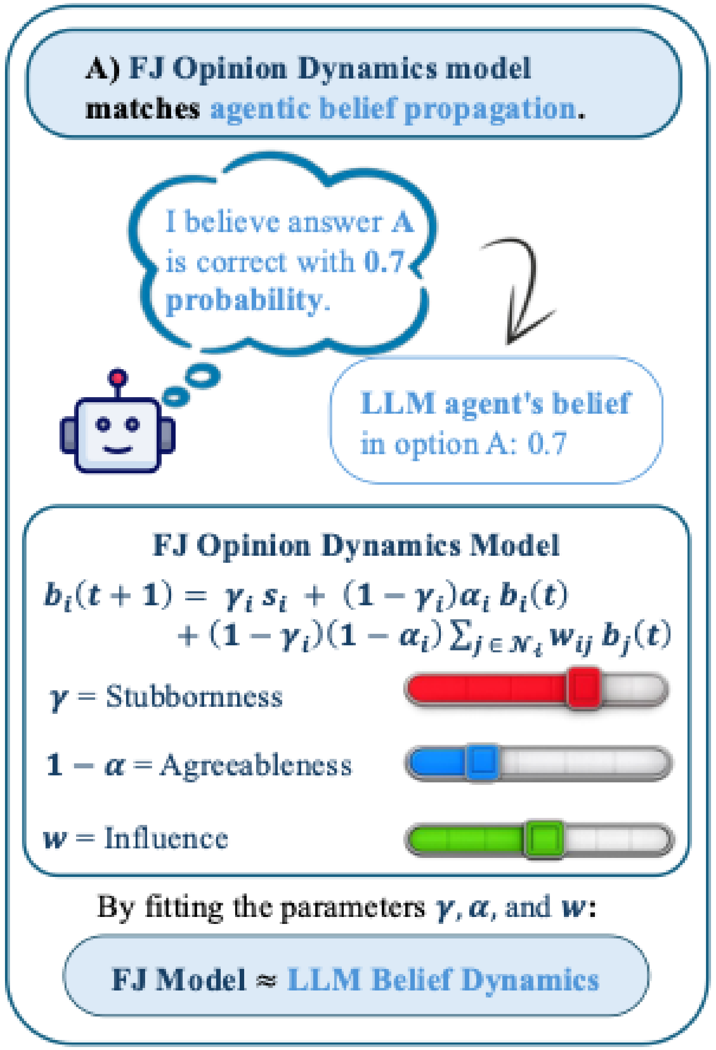


- LLM agent:
  - Individual objectives
  - Unique data access
  - Specialized expertise/competency
  - Process information
  - Take actions
  - Use tools
  - Make decisions

- Multi-agent system (LLM-MAS):
  - **Multiple, heterogenous LLM agents**
  - Interact via (dynamic) graph (e.g. send messages)

⟹ **emergence** of **collective** intelligence & **systemic risk**

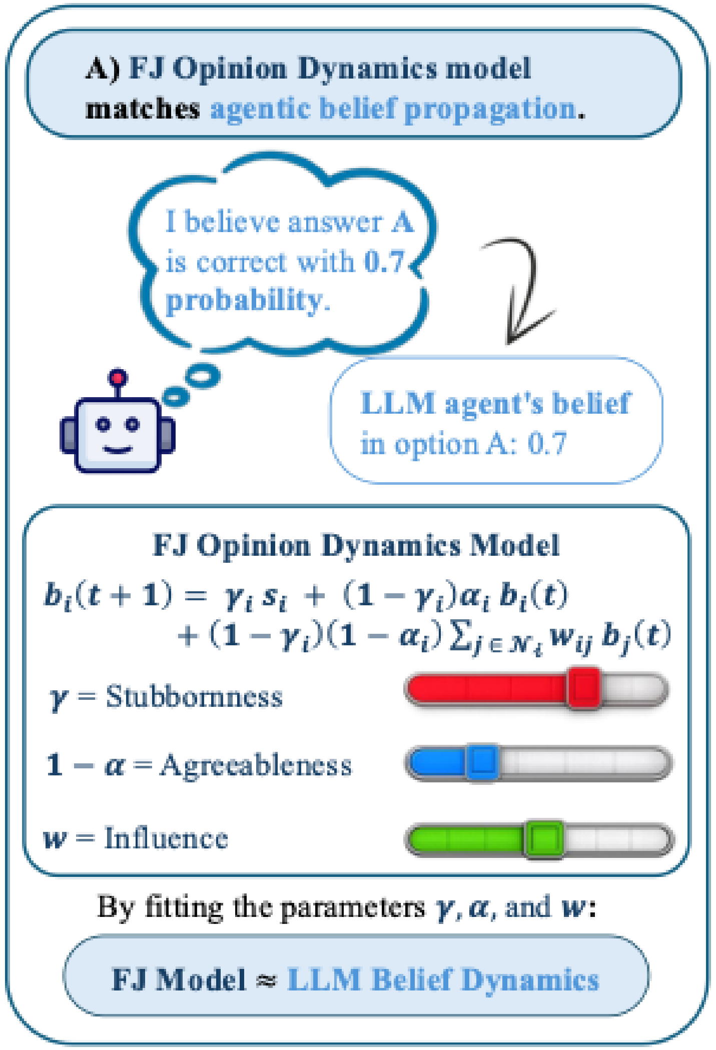
Robustness of opinion formation in agentic network



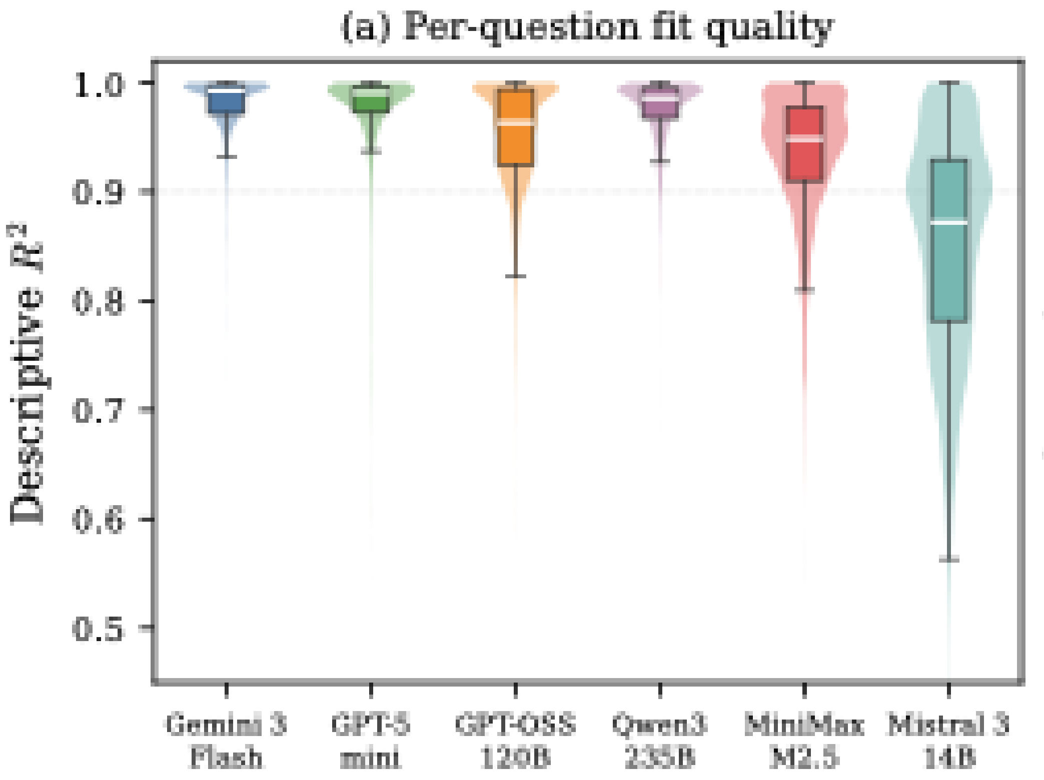
Abedini, Mavali, Schönherr, Pawelczyk, Burkholz. Don't Trust Stubborn Neighbors: A Security Framework for Agentic Networks. *CompLearn @ ICML*, 2026

.

Robustness of opinion formation in agentic network



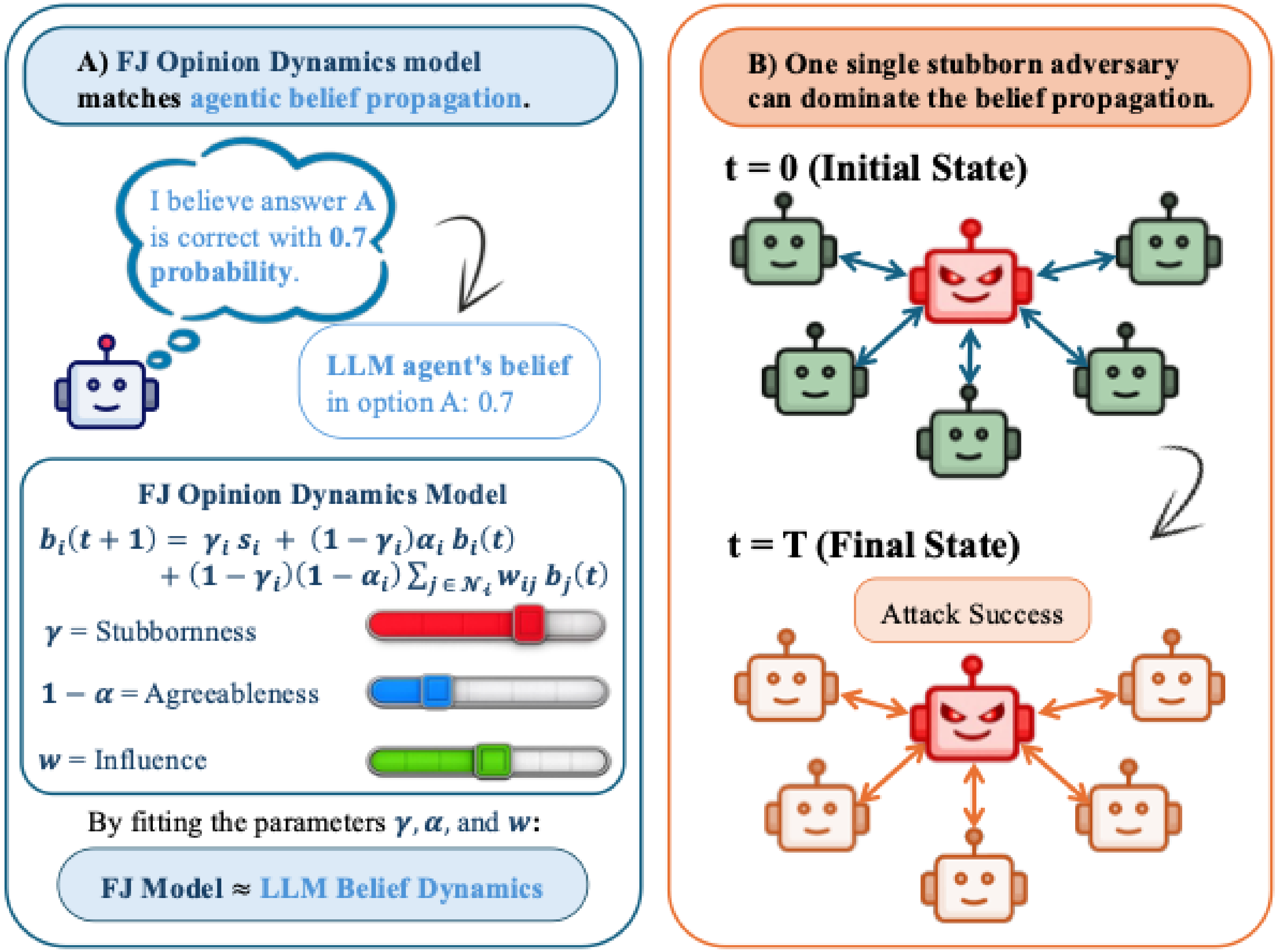
Friedkin-Johnsen model fits LLM opinion formation!



Abedini, Mavali, Schönherr, Pawelczyk, Burkholz. Don't Trust Stubborn Neighbors: A Security Framework for Agentic Networks. *CompLearn*, 2026

Bause, Niederle, Pawelczyk, Burkholz. Multi-Agent Systems are Mixtures of Experts: Who Becomes an Influencer? *CompLearn @ ICML*, 2026

Robustness of opinion formation in agentic network



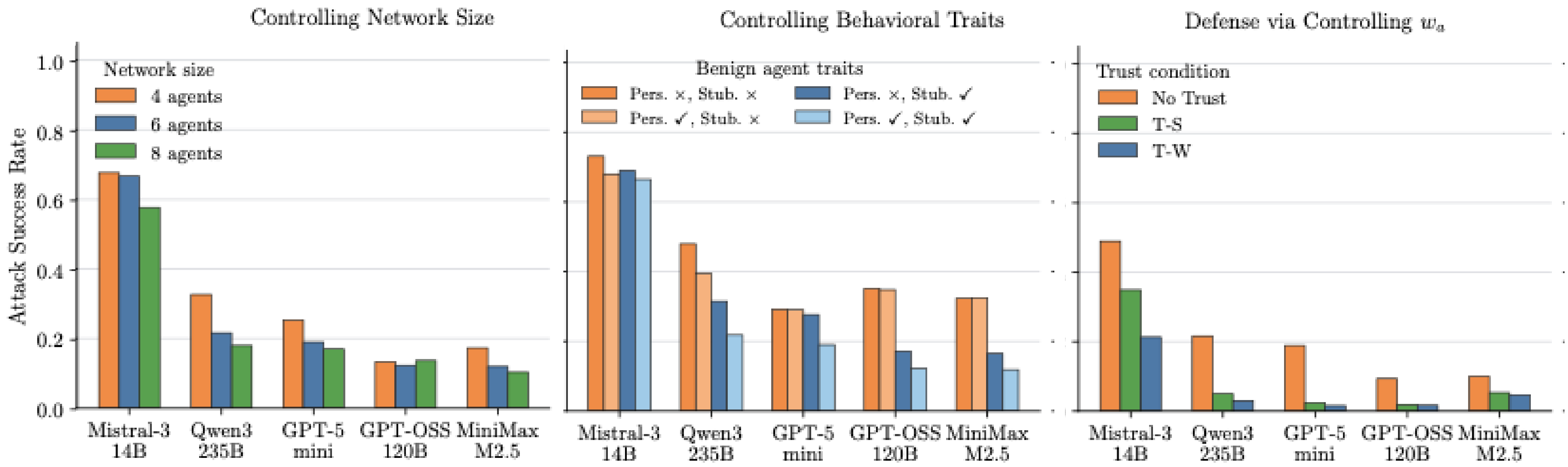
Theory:

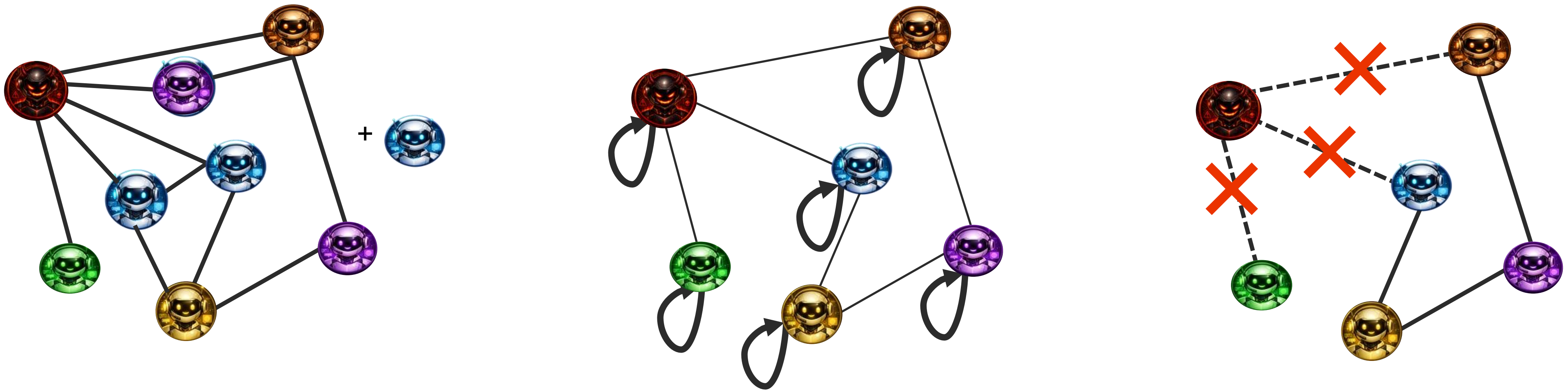
Agreeable agents get **dominated** by **stubborn** agents.

$$B_a(\infty) = \underbrace{(I - W_a)^{-1} W_s}_{\text{Propagation multiplier}} \underbrace{B_s(0)}_{\text{Stubborn belief}} .$$

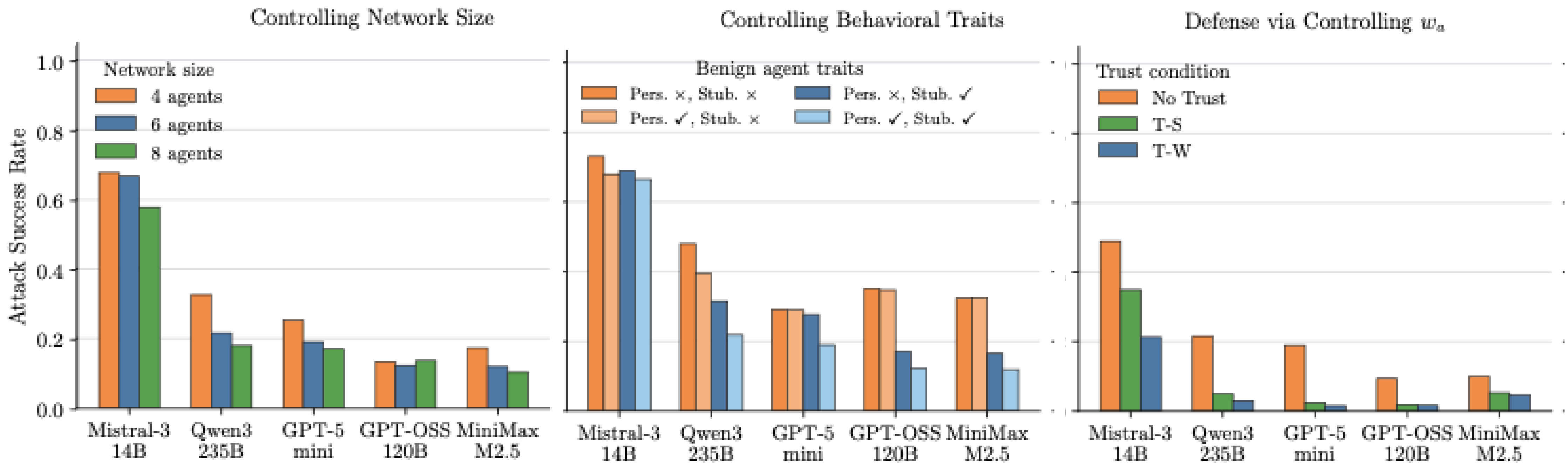
Abedini, Mavali, Schönherr, Pawelczyk, Burkholz. Don’t Trust Stubborn Neighbors: A Security Framework for Agentic Networks. *CompLearn @ ICML*, 2026

How to increase robustness?





How to increase robustness?



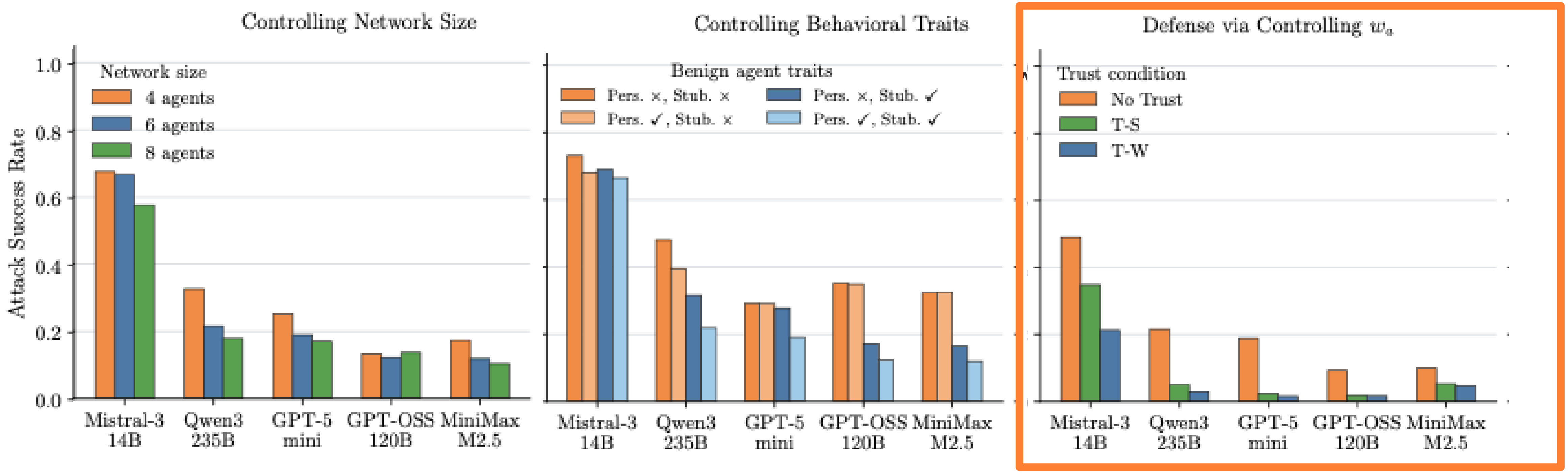
- Downside:

- expensive
- requires high stubbornness/low effective peer pull -> less consensus/utility

- Downside:

less consensus/utility

How to increase robustness? – Control trust/network



- Downside:

- expensive
- requires high stubbornness/low effective peer pull -> less consensus/utility

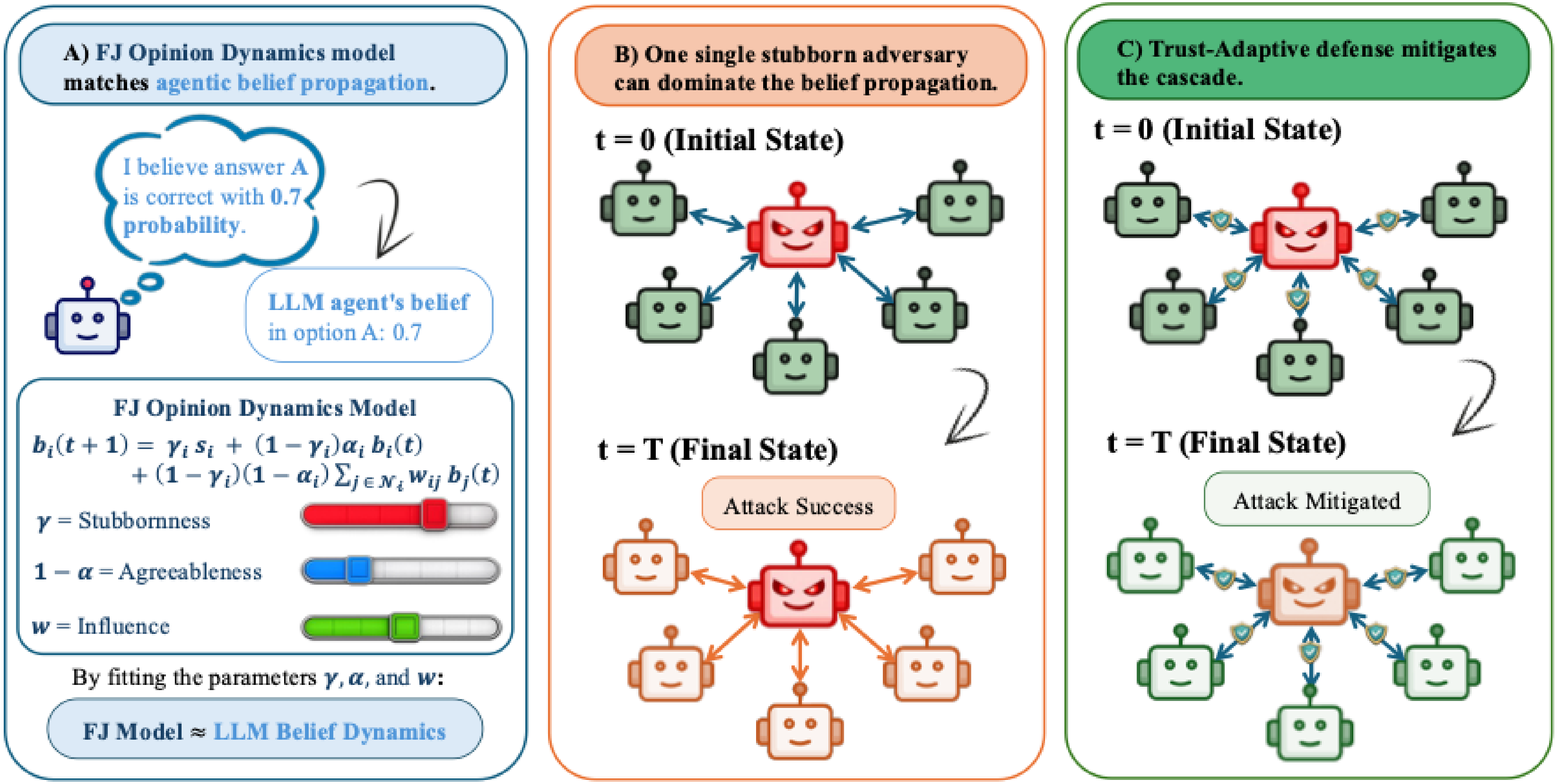
- Downside:

less consensus/utility

+ Advantage:

High utility and robustness

Robustness of opinion formation in agentic network

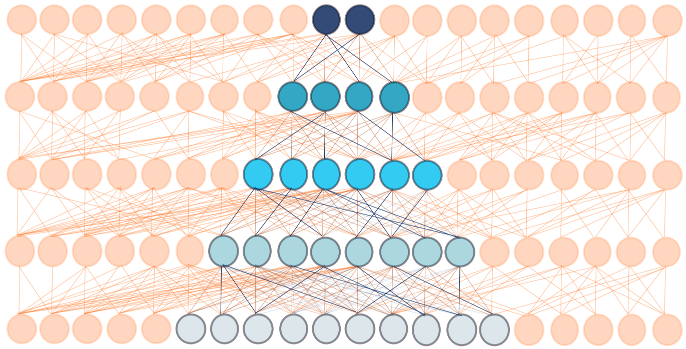


Abedini, Mavali, Schönherr, Pawelczyk, Burkholz. Don't Trust Stubborn Neighbors: A Security Framework for Agentic Networks. *CompLearn @ ICML*, 2026

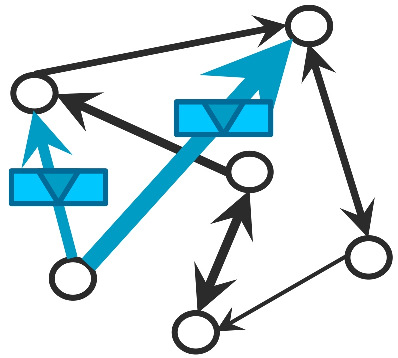
.

Summary: Complex network science for AI

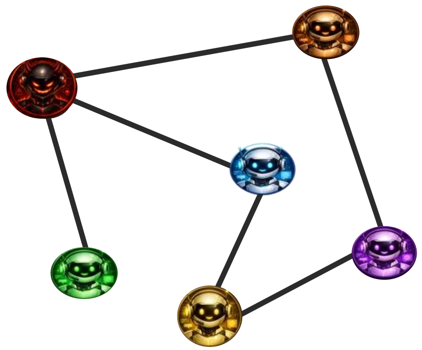
1. **Sparse** deep learning
   - Random graph **ensembles**
   - **Learning networks** (by sparsification)



2. **Graph** Neural Networks:
   - Spectral rewiring (**Braess** paradox)
   - Features by running cascade **process**
   - **Learning on** networks



3. **Agentic** networks
   - Robustness of agentic LLM system
   - Friedkin-Johnsen **opinion formation**
   - Process/network weights **emerge**



Many thanks to my collaborators

Relational Machine Learning Lab

and generous funding by

erc

Sparse-ML

and Apple Research

Machine Learning Research