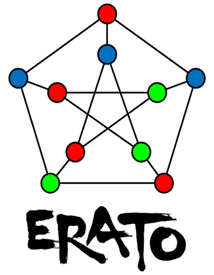
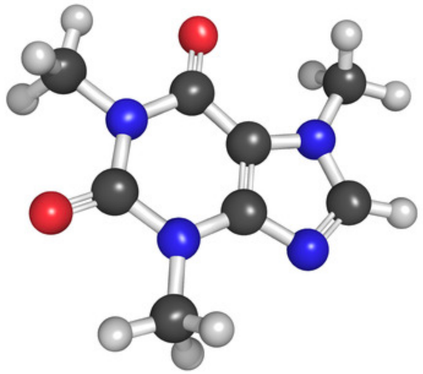


Modeling Intelligence via Graph Neural Networks

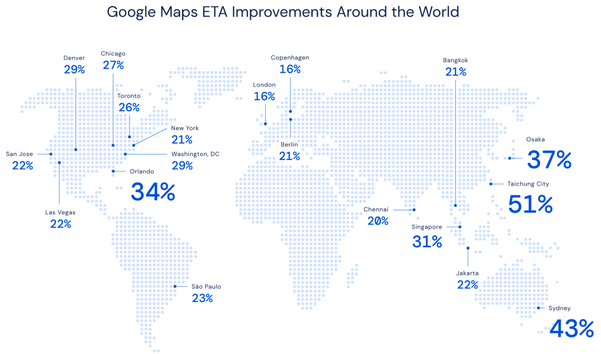
Keyulu Xu

MIT

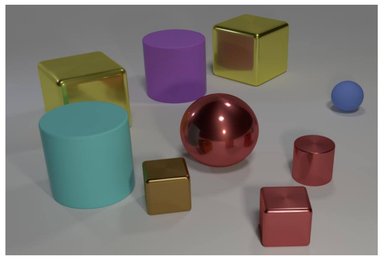
Representation: objects in the world as graphs



$\longrightarrow \quad \mathbb{R}^n \quad \longrightarrow$ prediction



Reasoning: learning to implement an algorithm



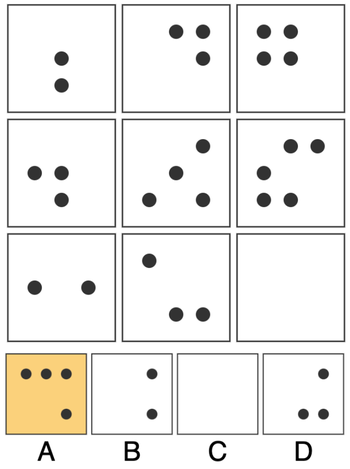
VQA

(Santoro et al 2016)

Question: `Calculate -841880142.544 + 411127.`
Answer: `-841469015.544`
Question: `Let x(g) = 9*g + 1. Let q(c) = 2*c + 1. Let f(i) = 3*i - 39. Let w(j) = q(x(j)). Calculate f(w(a)).`
Answer: `54*a - 30`
Question: `Let e(l) = l - 6. Is 2 a factor of both e(9) and 2?`
Answer: `False`

Mathematical reasoning

(Saxton et al. 2019, Lample et al 2020)



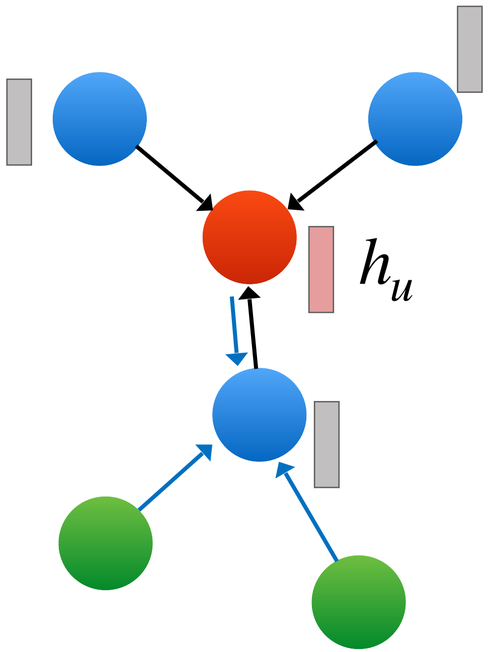
IQ tests

(Santoro et al. 2018, Zhang et al 2019)

Physical reasoning

(Wu et al. 2017, Battagalia et al 2016, Janner et al 2019)

Graph Neural Networks (GNNs)



In each round:

For $u \in V$ concurrently:

Aggregate over neighbors

Representation of neighbor node v in round $k-1$

$$h_u^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{ \left(h_v^{(k-1)}, h_u^{(k-1)} \right) \right\} \middle| v \in \mathcal{N}(u) \right)$$

.....

Graph-level **readout**

$$h_G = \text{READOUT}\left(\left\{ h_u^{(K)} \right\} \middle| u \in V \right)$$

(Gori et al. 2005, Merkwirth & Lengauer 2005, Scarselli et al 2009, Duvenaud et al., 2015, Battaglia et al., 2016, Dai et al., 2016, Defferrard et al., 2016, Kearnes et al., 2016, Li et al., 2016, Gilmer et al., 2017, Hamilton et al., 2017, Kipf & Welling, 2017, Velickovic et al., 2018, Xu et al., 2018)

I. Representation: Expressive Power

ICML’18 (long talk)

ICLR’19 (oral)

II. Reasoning: Generalization

NeurIPS’19

ICLR’20 (spotlight)

III. Reasoning: Extrapolation

ICLR’21 (oral)

IV. Optimization

ICML’21

ICML’21

I. Representation: Expressive power

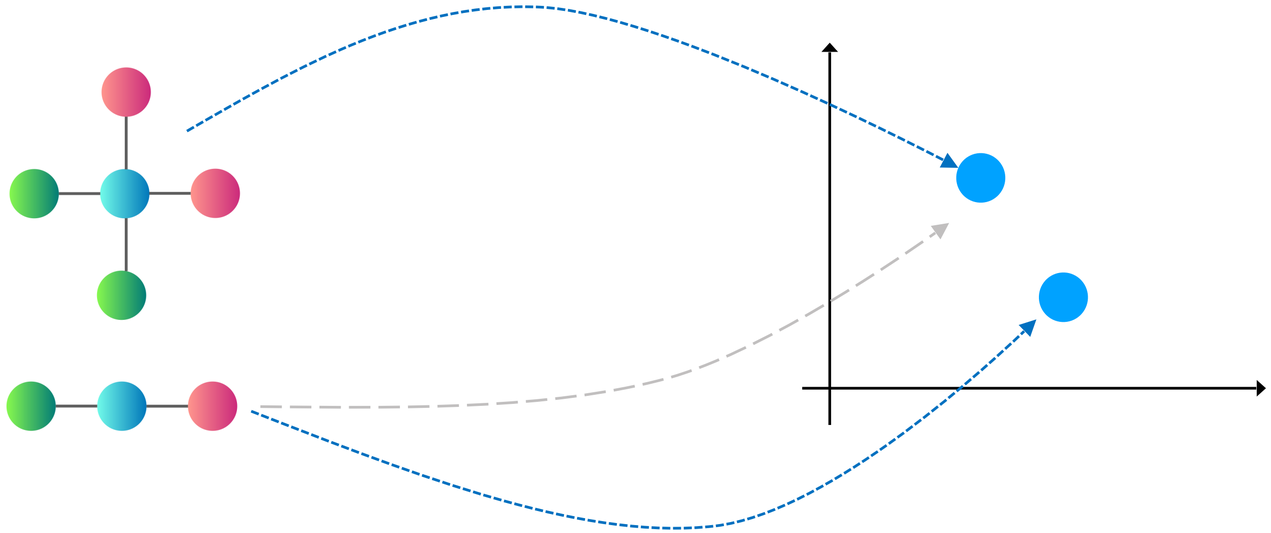
How Powerful are Graph Neural Networks?

K. Xu, W. Hu, J. Leskovec, S. Jegelka
ICLR'19 (oral)

Representation Lerning on Graphs with Jumping Knowledge Networks

K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, S. Jegelka
ICML'18 (long talk)

Expressive power as graph isomorphism test



Which graphs can a GNN distinguish?

How powerful are GNNs?

Theorem (XHLJ'19)
GNNs are at most as powerful as a Weisfeiler-Lehman graph isomorphism test*.

This upper bound is achieved if AGGREGATE and READOUT are injective multiset functions.

neighborhood - multiset

*(Weisfeiler & Lehman 1968, Babai, Erdös, Selkow 1980, Babai & Kucera 1979, Cai, Furer, Immerman 1992, Evdokimov & Ponomarenko 1999, Douglas 2011)

failure cases: certain regular graphs

A maximally powerful GNN

Lemma (XHLJ'19)
Any (injective) multi-set function g can be decomposed as

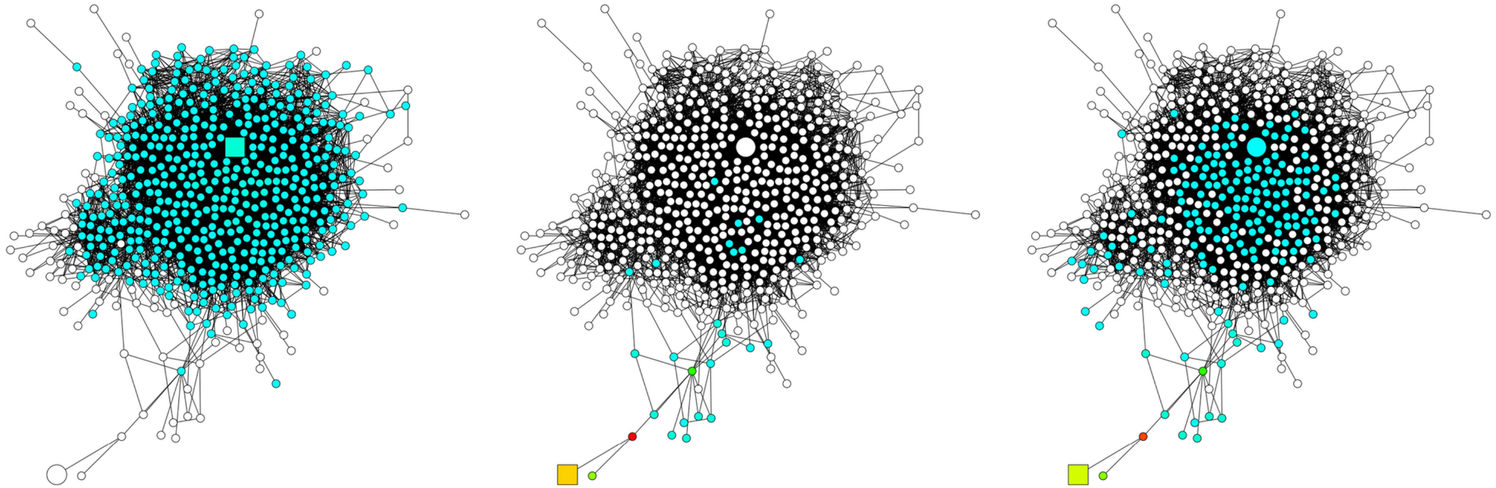
$$g(X) = \phi\left(\sum_{x \in X} f(x)\right)$$

(generalizing Zaheer et al 2017, Ravanbakhsh et al 2016, Qi et al 2017,…)

- Graph Isomorphism Network (GIN):

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1+\epsilon^{(k)}\right)\cdot h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}\right)$$

Puzzle of the underperformance of deeper GNNs



(a) 4 steps at core (b) 4 steps at tree (c) 5 steps at tree

XLTSKJ'18

Optimal depth depends on the subgraph structure (expander vs. tree).

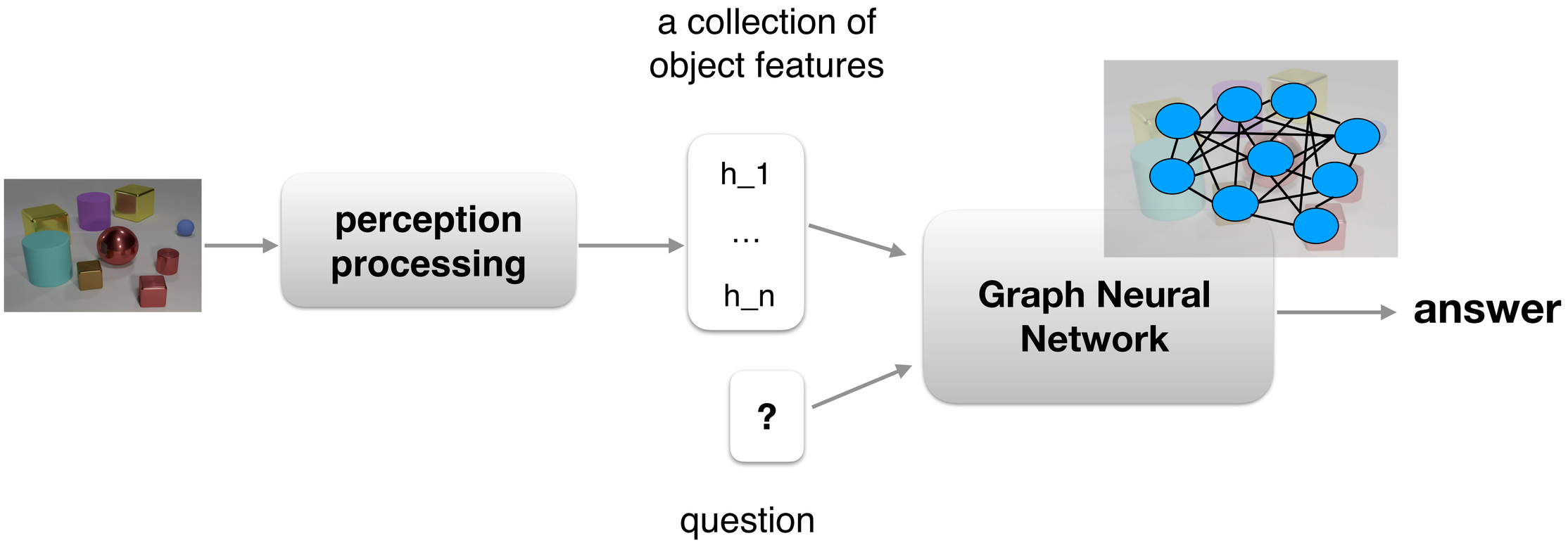
JK-Net: adaptively select the depth via skip connections.

II. Reasoning: Generalization

Graph Neural Tangent Kernel
S. Du, K. Hou, B. Poczos, R. Salakhutdinov, R. Wang, K. Xu
NeurIPS'19

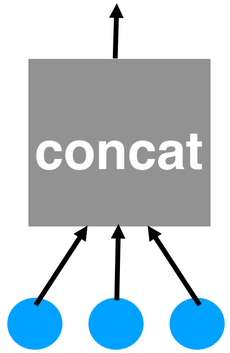
What Can Neural Networks Reason About?
K. Xu, J. Li, M. Zhang, S. Du, K. Kawarabayashi, S. Jegelka
ICLR'20 (spotlight)

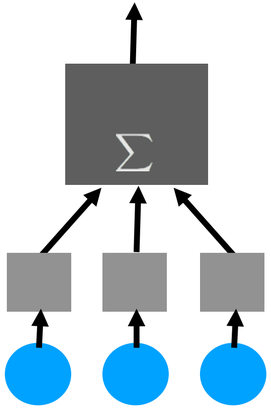
Reasoning with object representations

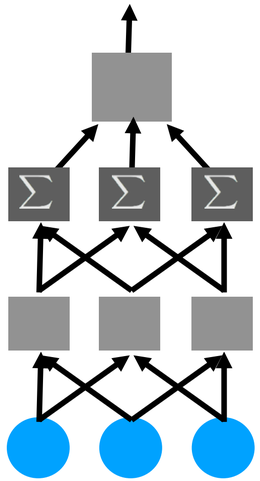


(Weston et al., 2015; Johnson et al., 2017a; Wu et al. 2017, Fleuret et al., 2011; Antol et al., 2015; Battaglia et al., 2016, 2018; Watters et al., 2017; Fragkiadaki et al., 2016; Chang et al., 2017, 2019; Saxton et al., 2019; Santoro et al., 2018…)

Equal expressive power, different generalization

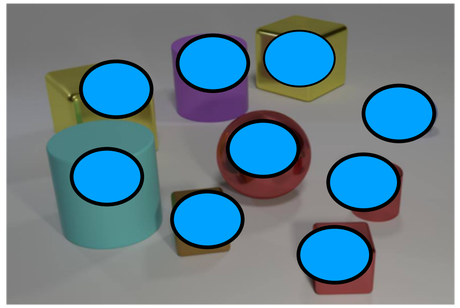


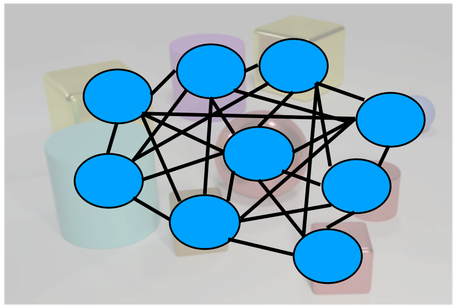




.....

Input: a set of objects





```
1. filter_shape(scene, cylinder)
2. relate(behind)
3. filter_shape(scene, cube)
4. filter_size(scene, large)
5. count(scene)
```

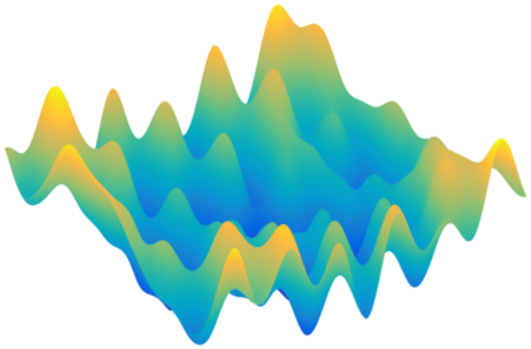
feedforward network

Deep Set

GNN

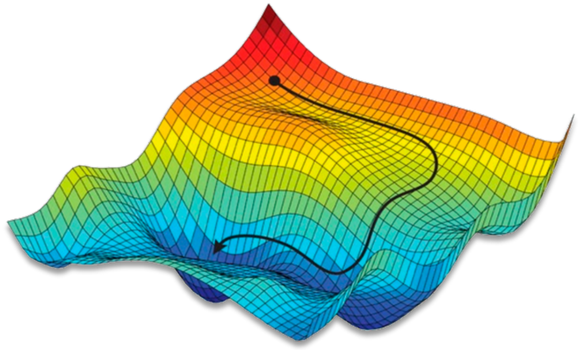
e.g., neural programs

Approaches of generalization analysis



Complexity based

(Scarselli et al 2018, Garg et al 2020)



+ Training algorithm

*(Du, Hou, Poczos, Salakhutdinov, Wang, **X.** 2019)*

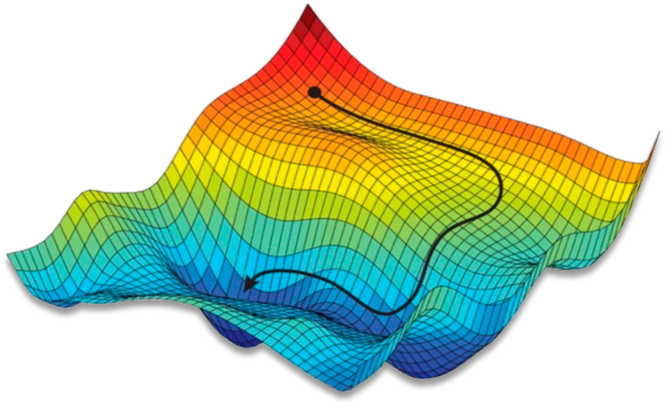
+ Network & task structure

*(**X.**, Zhang, Li, Du, Kawarabayashi, Jegelka 2020, 2021)*

more "practical"

more assumptions

Learning dynamics: Graph NTK



Parameter trajectory

$\theta_{GNN}(t)$

GNN output $f(\theta_{GNN}, G)$

DHPSWX'19

Over-parameterized GNNs trained by GD is equivalent to that of kernel regression with Graph NTK:

$$k(G, G') = \mathbb{E}_{\theta_{GNN} \sim \mathscr{W}} \left[\left\langle \frac{\partial f(\theta_{GNN}, G)}{\partial \theta_{GNN}}, \frac{\partial f(\theta_{GNN}, G')}{\partial \theta_{GNN}} \right\rangle \right]$$

(NTK theory: Jacot et al 2018, Li and Liang 2018, Allen-Zhu et al 2019, Arora et al 2019ab, Cao and Gu 2019, Du et al 2019ab…)

Generalization for simple functions on graphs

Generalization error $\dfrac{\sqrt{\boldsymbol{y}^\top \boldsymbol{H}^{-1}\boldsymbol{y} \cdot \mathrm{tr}\left(\boldsymbol{H}\right)}}{n}$.

(Bartlett and Mendelson 2002)

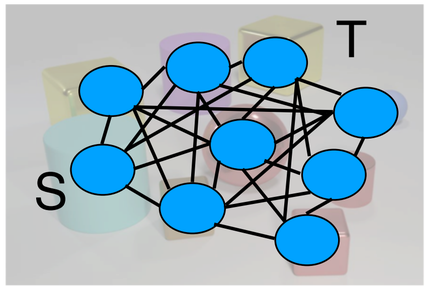
$\boldsymbol{H}$ - ***graph NTK matrix***

$\boldsymbol{y}$ - *training labels*

n - *number of training data*

How task and NN structure affect sample efficiency

Example



shortest path distance?

Graph Neural Network

for k = 1 … GNN iter:

for u in S:

$h_u^{(k)} = \Sigma_v \; MLP(h_v^{(k-1)}, h_u^{(k-1)})$

No need to learn for-loops

Bellman-Ford algorithm

for k = 1 … |S| - 1:

for u in S:

$d[k][u] = \min_v d[k-1][v] + cost(v, u)$

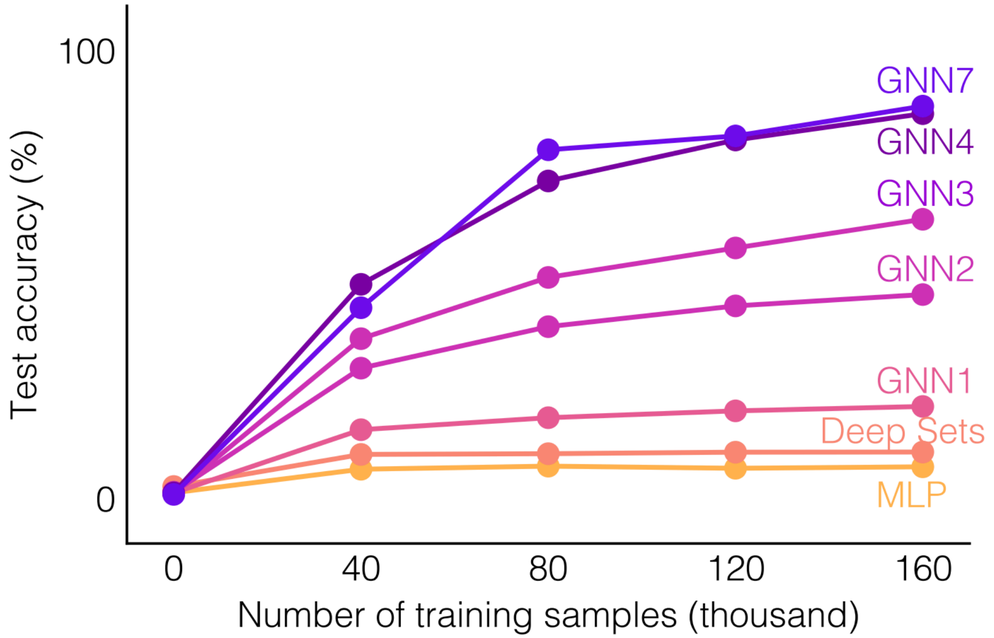
Learns a simple reasoning step

Other architectures need to learn functions with higher complexity, e.g., for-loops

Algorithmic alignment: formalizing inductive biases

Algorithmic alignment (XLZDKJ'20)
Network can simulate algorithm via *few, easy-to-learn* "modules".

Claim: Better algo alignment implies better generalization.



Better alignment implies better generalization

Algorithmic alignment (XLZDKJ'20)

A neural network (M, ϵ, δ)-aligns with an algorithm if it can simulate the algorithm via n weight-shared modules, each of which is (ϵ, δ) PAC-learnable with M/n samples.

* *Sample complexity of modules by e.g., NTK*

Theorem (XLZDKJ'20)

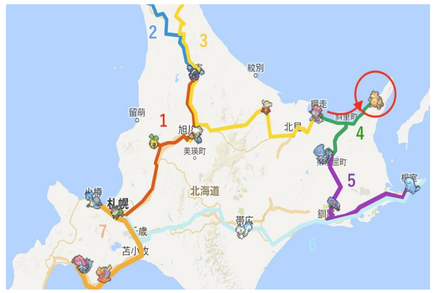
If a neural network and a task algorithm (M, ϵ, δ)-align, then, under assumptions*, the task is $(O(\epsilon), O(\delta))$ PAC-learnable by the network with M examples.

GNNs can sample-efficiently learn DP

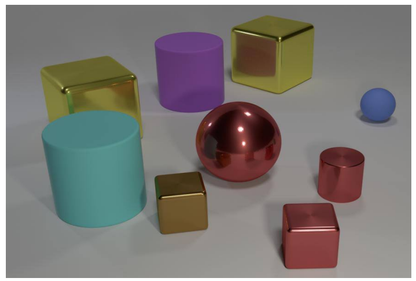
$$\text{Answer}[k][i] = \text{DP-Update}(\{\text{Answer}[k-1][j],\ j = 1 \dots n\})$$

$$h_s^{(k)} = \sum_{t \in S} \mathbf{MLP}_1^{(k)} \left(h_s^{(k-1)}, h_t^{(k-1)} \right)$$

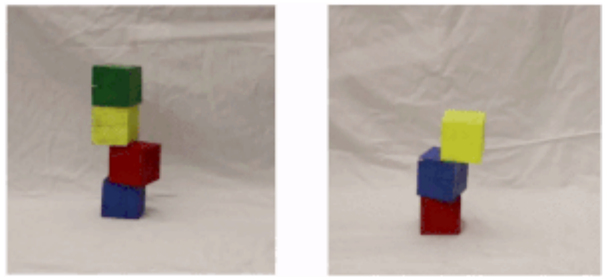
Reasoning tasks as dynamic programming (DP):



graph algorithms



visual question answering



Intuitive physics

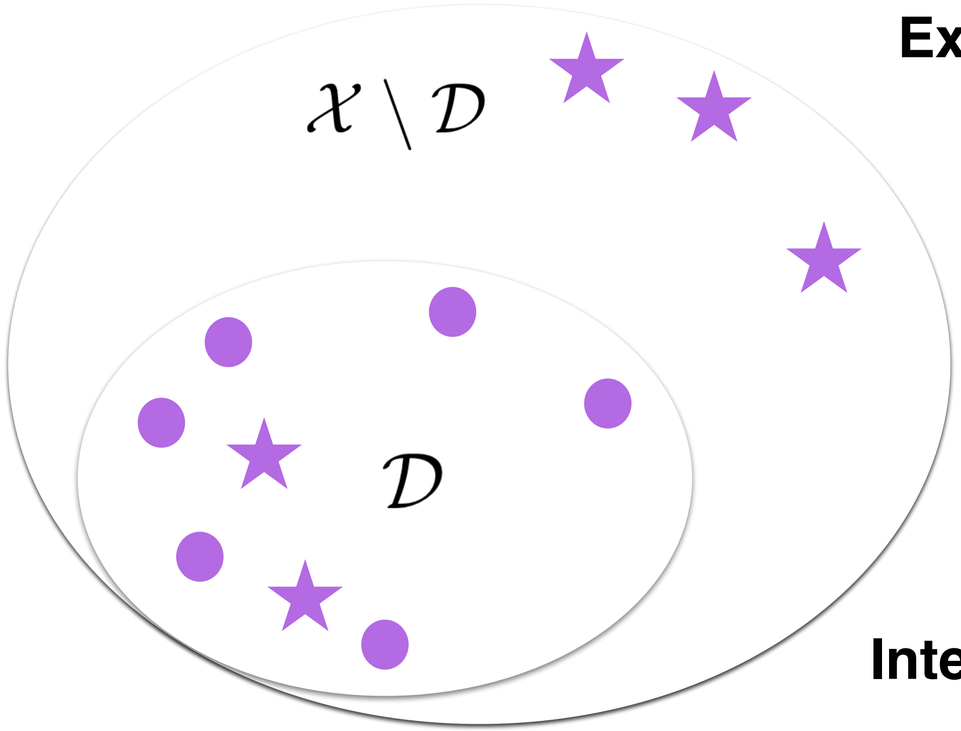
III. Reasoning: Extrapolation

How Neural Networks Extrapolate:
From Feedforward to Graph Neural Networks
K. Xu, M. Zhang, J. Li, S. Du, K. Kawarabayashi, S. Jegelka
ICLR'21 (oral)

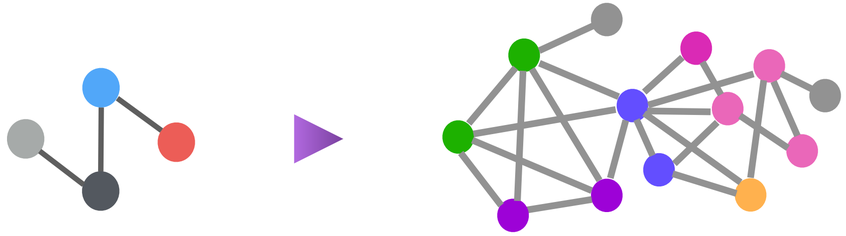
Extrapolation vs. interpolation

Train NN f to learn underlying function $g : \mathcal{X} \to \mathbb{R}$ with training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathcal{D}$

$$\mathbb{E}_{\boldsymbol{x} \sim \mathcal{P}_{\text{test}}} [\ell(f(\boldsymbol{x}), g(\boldsymbol{x}))]$$



Extrapolation

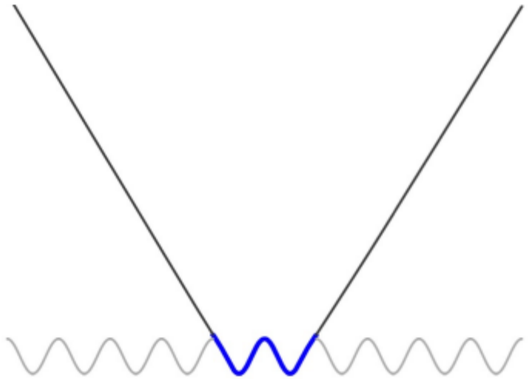


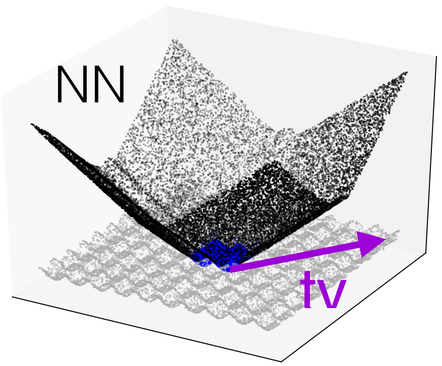
Train

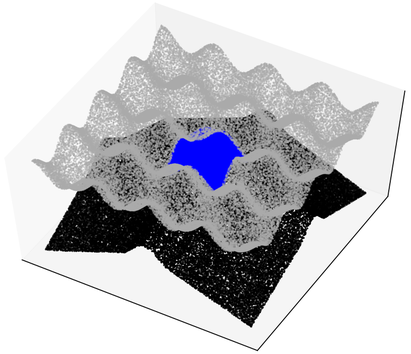
Test

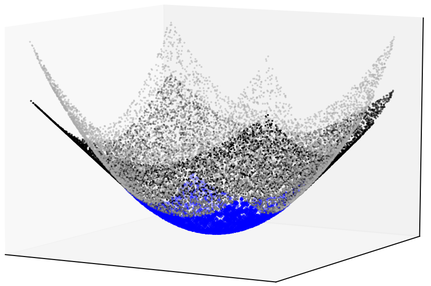
Interpolation $\mathcal{P}_{\text{train}} = \mathcal{P}_{\text{test}}$

Linear extrapolation behavior of ReLU MLPs







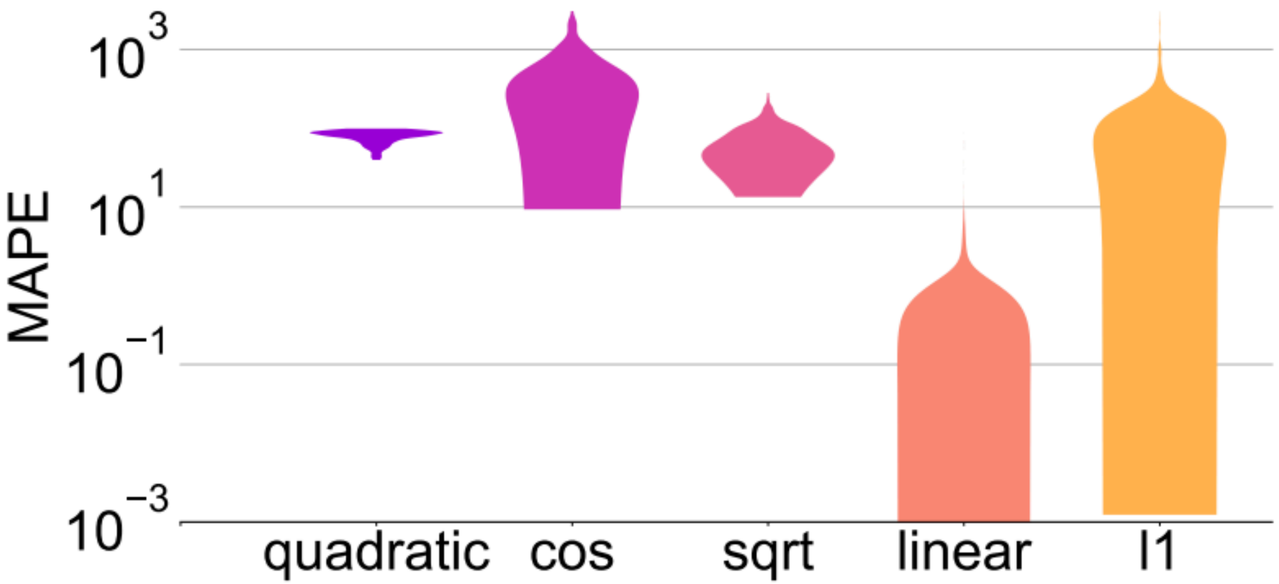


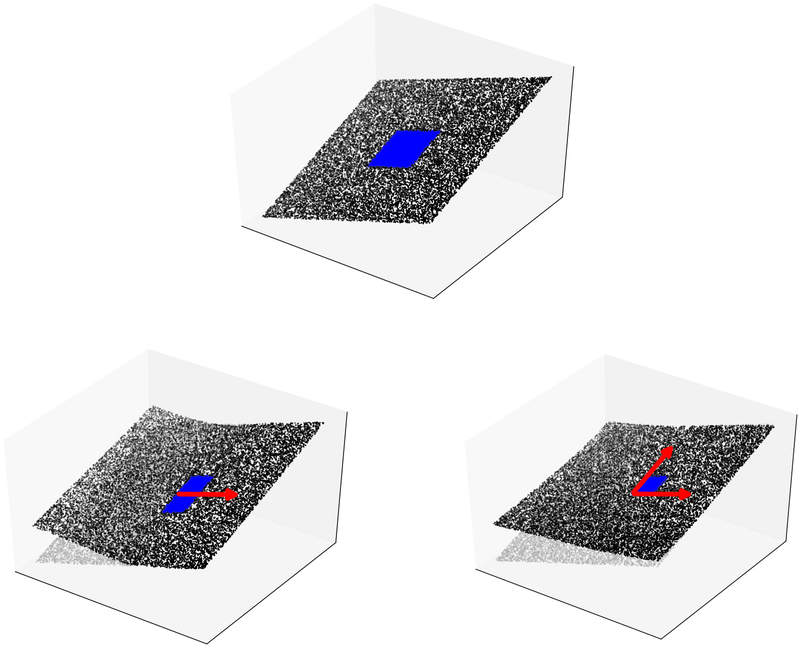
Theorem (XZLDKJ'21)

Let f be a two-layer ReLU MLP trained by GD*. For any direction $v \in \mathbb{R}^d$, let $x = tv$. For any $h > 0$, as $t \to \infty$, $f(x + hv) - f(x) \to \beta_v h$ with rate $O(1/t)$

* *Assumption: NTK regime*

Conditions for ReLU MLPs to extrapolate well





Theorem (XZLDKJ'21)

Let f be a two-layer ReLU MLP trained by GD*. Suppose target function is $\beta^{\intercal}x$ and support of training distribution covers all directions. As the number of training examples $n \to \infty$, $f(x) \to \beta^{\intercal}x$.

Implications for GNNs

Shortest Path: $d[k][u] = \min_{v \in \mathcal{N}(u)} d[k-1][v] + \boldsymbol{w}(v,u)$

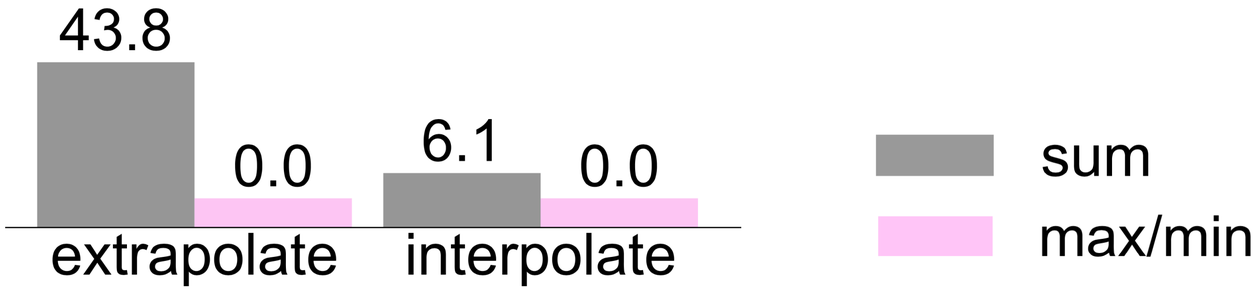
GNN (sum): $h_u^{(k)} = \Sigma_v \ \mathbf{MLP}^{(k)}\left(h_v^{(k-1)}, h_u^{(k-1)}, w(v,u)\right)$

✗ *MLP has to learn non-linear steps*

GNN that encodes the nonlinearity min

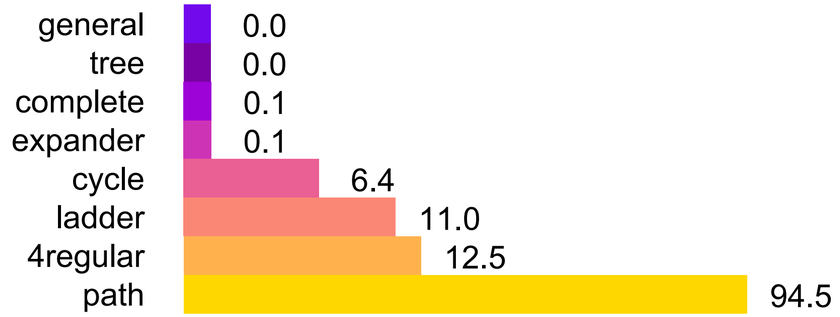
$h_u^{(k)} = \min_v \mathbf{MLP}^{(k)}\left(h_v^{(k-1)}, h_u^{(k-1)}, w(v,u)\right)$

✓ *MLP learns linear steps*

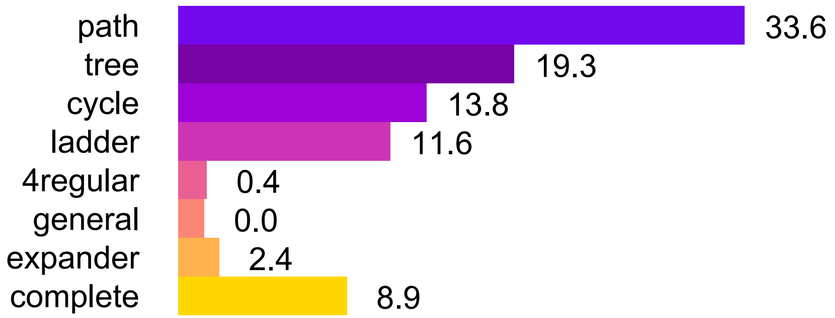


Training distribution for GNNs to extrapolate well

feature direction (MLP) & graph structure (GNN)



Max Degree



Shortest Path

Theorem (XZLDKJ'21)

A GNN encoding max in aggregation trained by GD* learns max degree if training data $\{\deg_{\max}(G_i), \deg_{\min}(G_i), N_i^{\max} \deg_{\max}(G_i), N_i^{\min} \deg_{\min}(G_i)\}_{i=1}^{n}$ spans $\mathbb{R}^4$.

* *Assumption: NTK regime*

Linear algorithmic alignment improves extrapolation

Encoding non-linearities in architecture & representations

Q: What direction is the closest creature facing?

P: `scene, filter_creature, filter_closest, unique, query_direction`

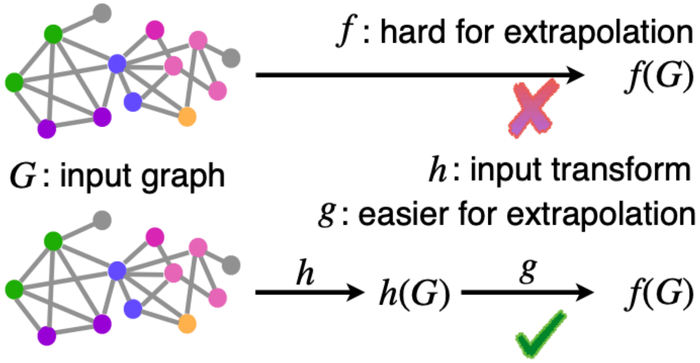
A: left

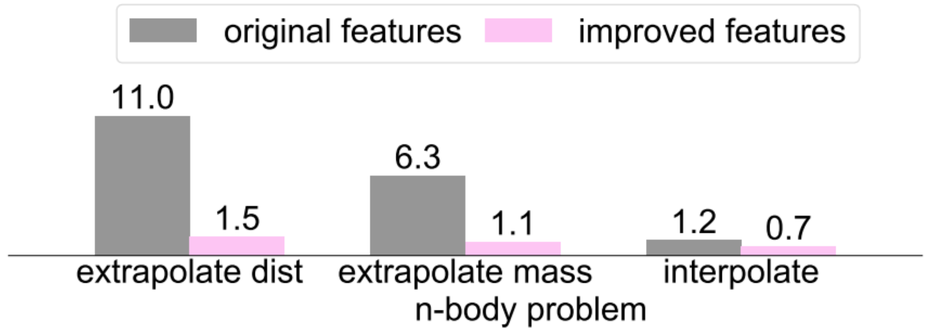
(Johnson et al 2017, Yi et al. 2018, Mao et al 2019…)

$$\texttt{NALU:} \quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$$

$$\mathbf{m} = \exp \mathbf{W}(\log(|\mathbf{x}| + \epsilon)), \ \mathbf{g} = \sigma(\mathbf{G}\mathbf{x})$$

(Trask et al. 2018, Madsen & Johansen 2020)





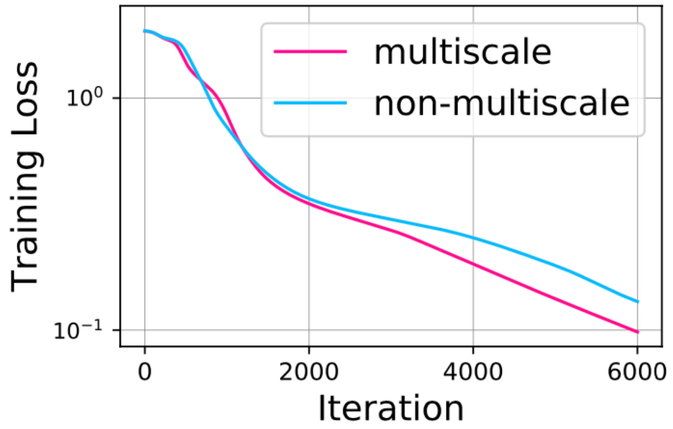
IV. Optimization

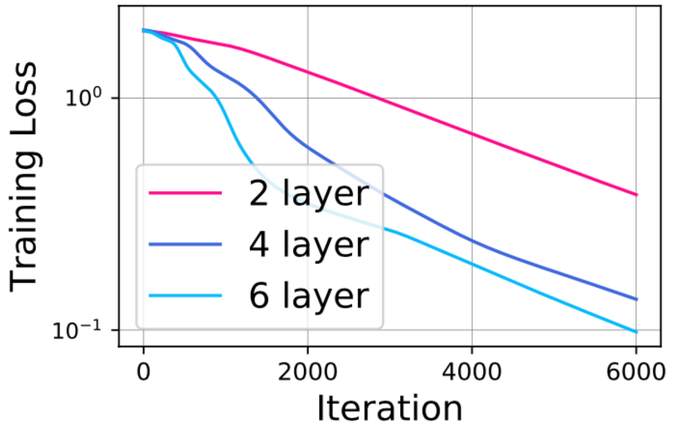
Optimization of GNNs: Implicit Accleration by Skip Connections and More Depth
K. Xu, M. Zhang, S. Jegelka, K. Kawaguchi
ICML'21

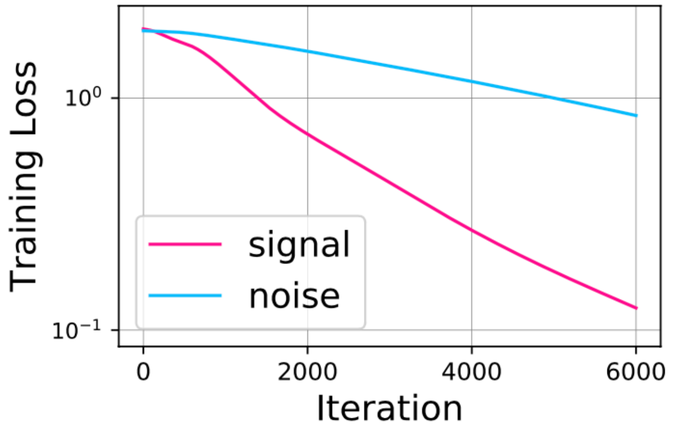
GraphNorm: A Principled Approach to Accelerating GNN Training
T. Cai, S. Luo, K. Xu, D. He, T. Liu, L.Wang
ICML'21

Global convergence and implicit acceleration

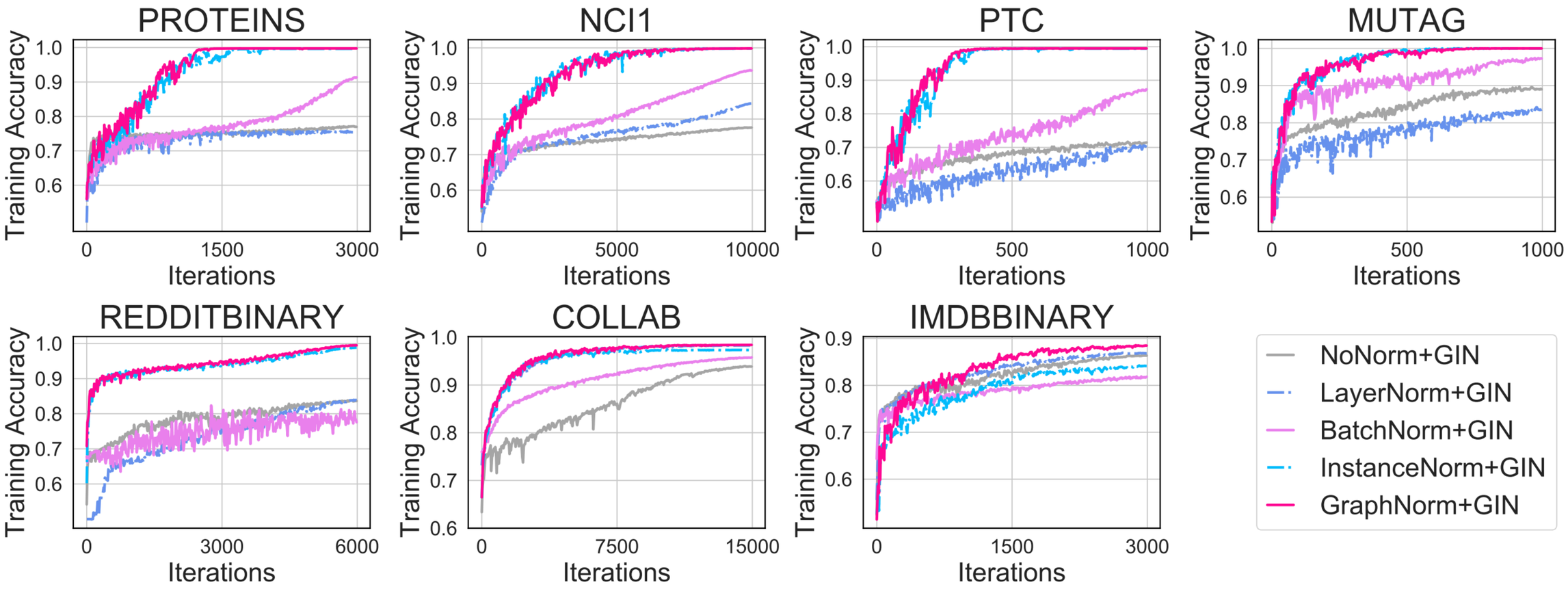
Theorem (XZJK'21)
Gradient descent training a linearized GNN, with or without skip connections, converges to a global minimum at a linear rate.







GraphNorm accelerates training



I. Representation: Expressive Power

ICML’18 (long talk)

ICLR’19 (oral)

II. Reasoning: Generalization

NeurIPS’19

ICLR’20 (spotlight)

III. Reasoning: Extrapolation

ICLR’21 (oral)

IV. Optimization

ICML’21

ICML’21

Other Research

Are Girls Neko or Shōjo?
M. Zhang, K. Xu, K. Kawarabayashi, S. Jegelka, J. Boyd-Graber. ACL'19

Information Obfuscation of Graph Neural Networks
P. Liao, H. Zhao, K. Xu, T. Jaakkola, G. Gordon, S. Jegelka, R. Salakhutdinov. ICML'21

Distributional Adversarial Networks
C. Li, D. Alvarez-Melis, K. Xu, S. Jegelka, S. Sra
ICLR'18 workshop

Noisy Labels Can Induce Good Representations
J. Li, M.Zhang, K. Xu, J. Dickerson, J. Ba

Ackowledgements