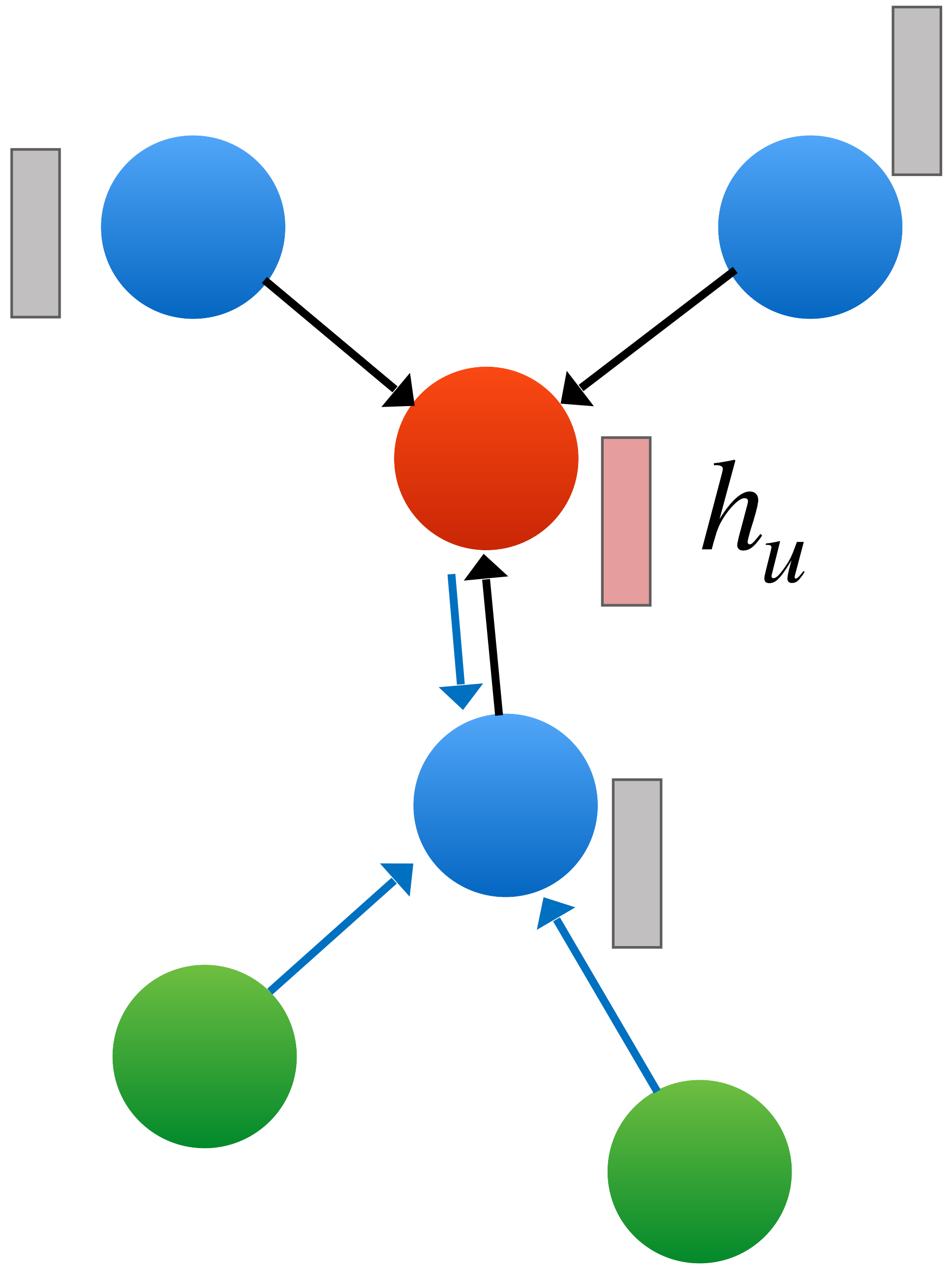
Graph Neural Networks: Generalization and Extrapolation

Keyulu Xu

MIT

Graph Neural Networks (GNNs)



In each round:

For $u \in V$ concurrently:

Aggregate over neighbors

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

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

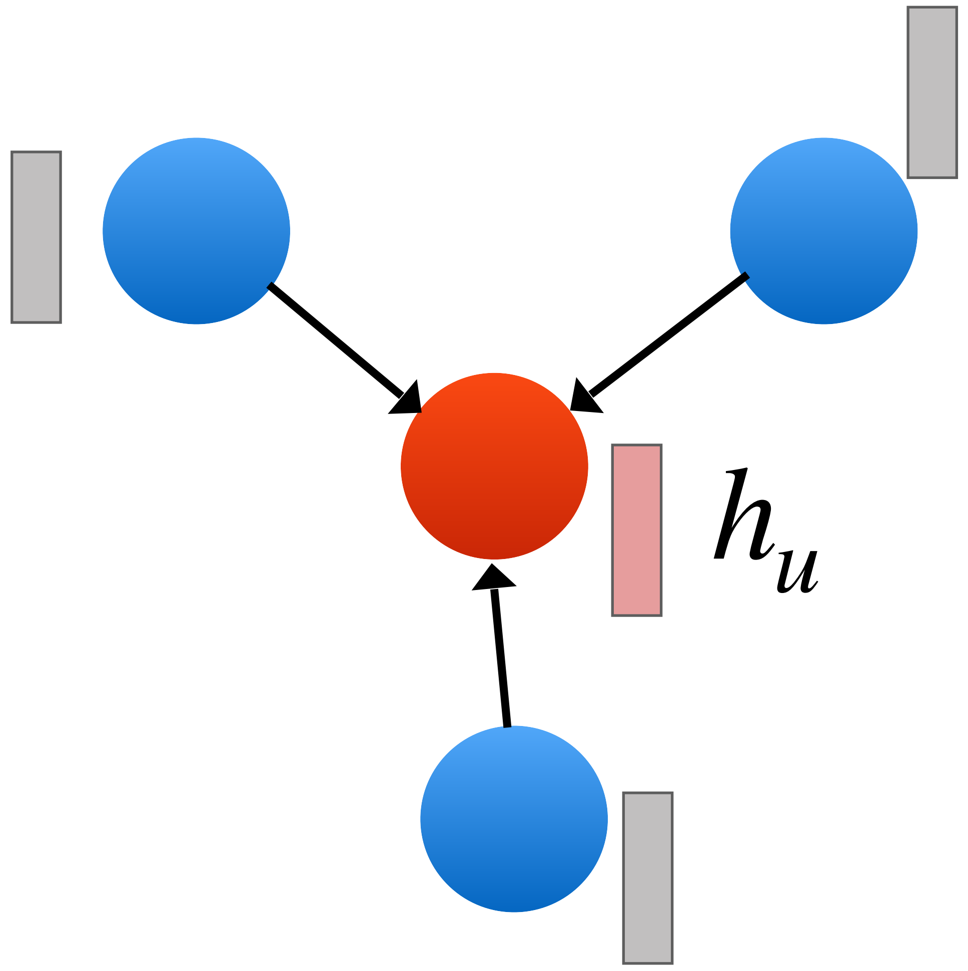
…………

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)

Training



1. Parameterize AGGREGATE$^{(k)}$ and READOUT

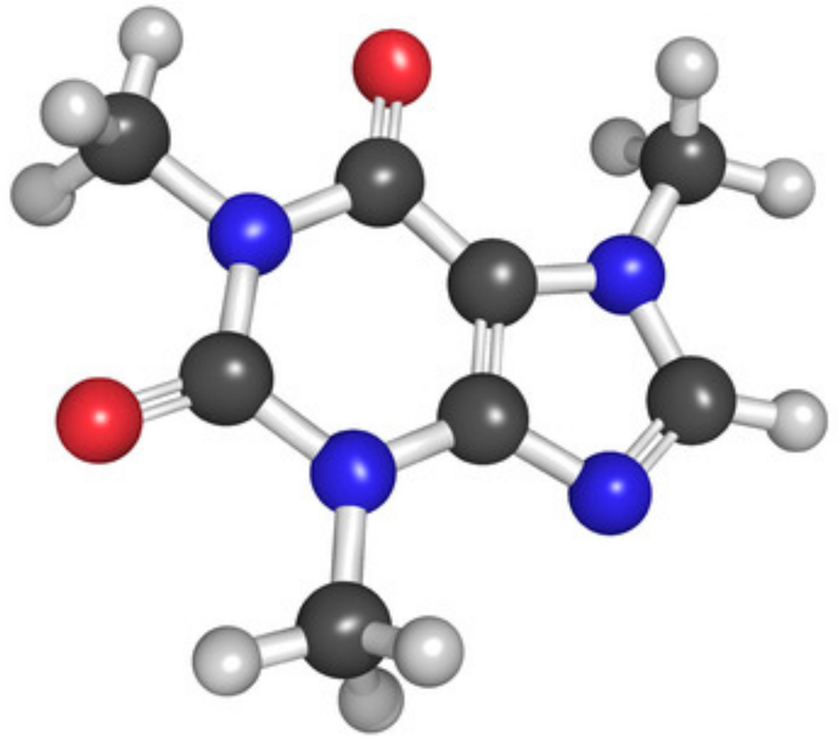
$$\boldsymbol{h}_u^{(k)} = \sum_{v \in \mathcal{N}(u)} \mathrm{MLP}^{(k)}\left(\boldsymbol{h}_u^{(k-1)}, \boldsymbol{h}_v^{(k-1)}, \boldsymbol{w}_{(v,u)}\right), \quad \boldsymbol{h}_G = \mathrm{MLP}^{(K+1)}\left(\sum_{u \in G} \boldsymbol{h}_u^{(K)}\right)$$

Can recover ConvNets, Transformer etc
with appropriate AGGREGATE

2. Specify a loss on node/graph/edge representations

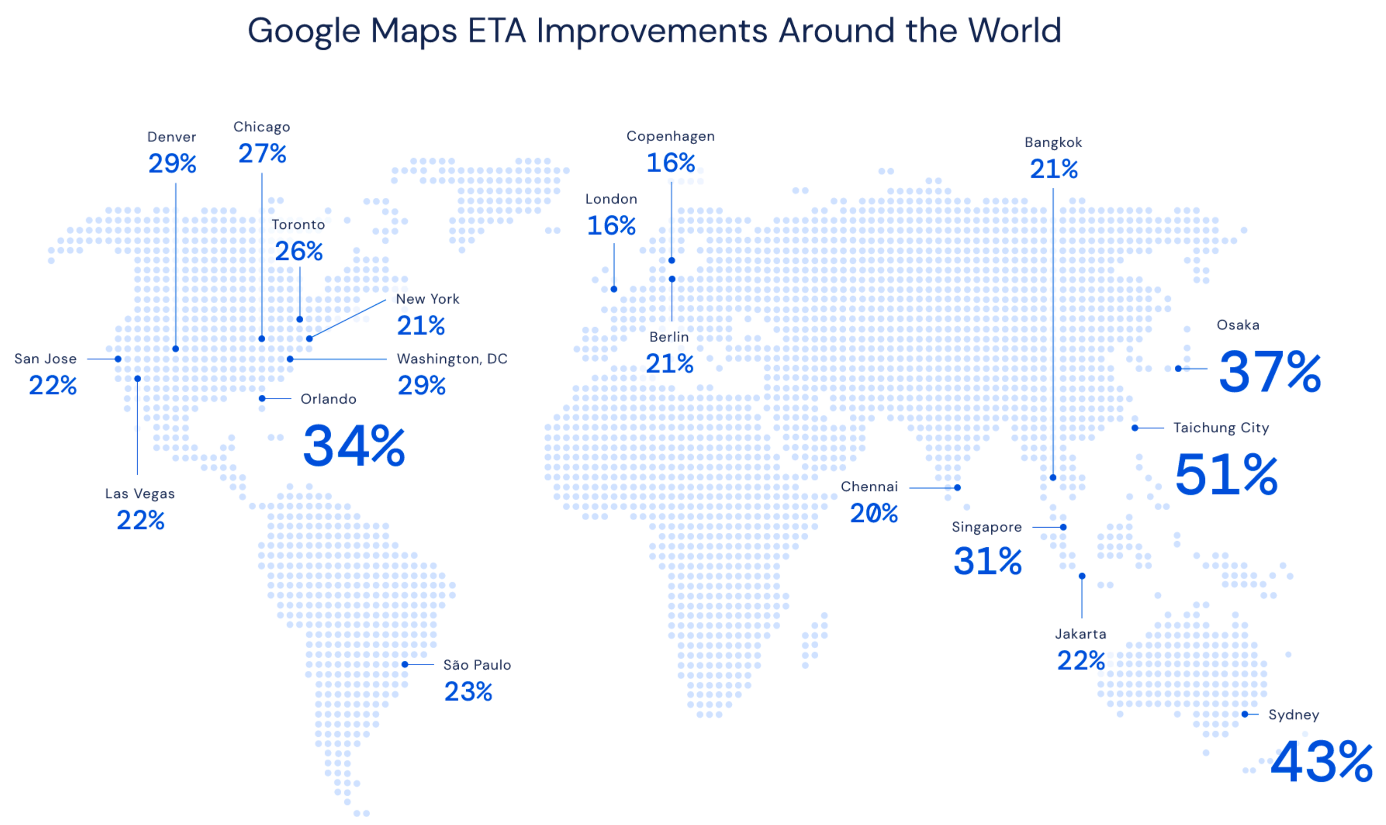
3. Train on data points with SGD

Applications



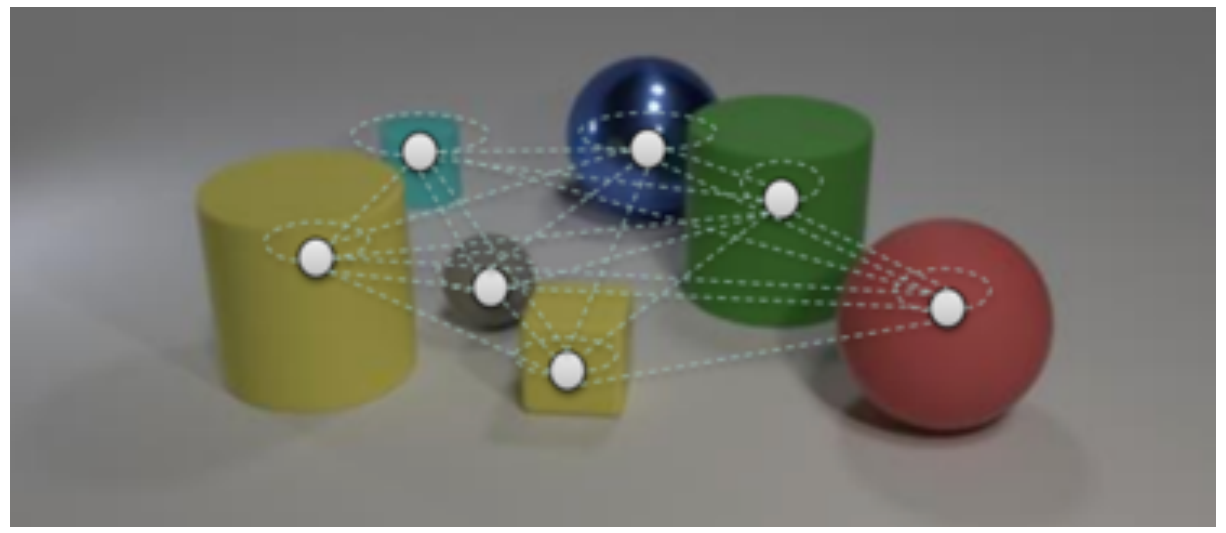
Drug discovery
(Duvenaud et al. 2015)

Recommender system
(Ying et al. 2018)



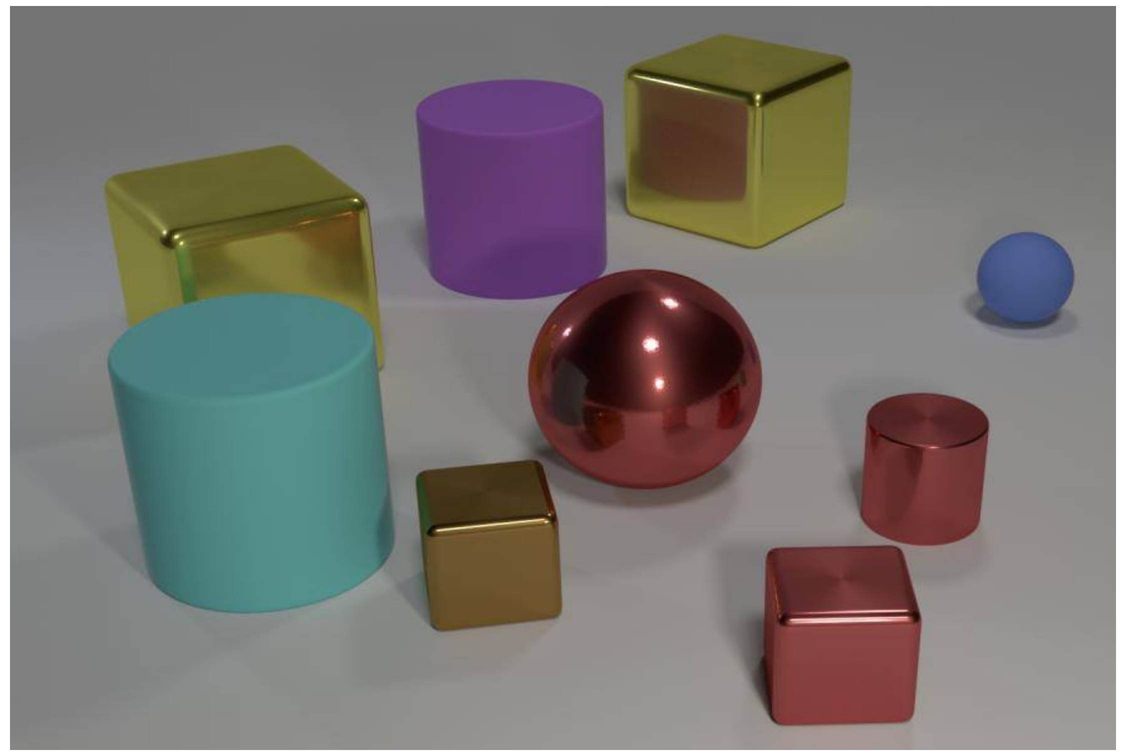
Google Map ETA
(Lange et al. 2020)

Physical reasoning
(Wu et al. 2017)

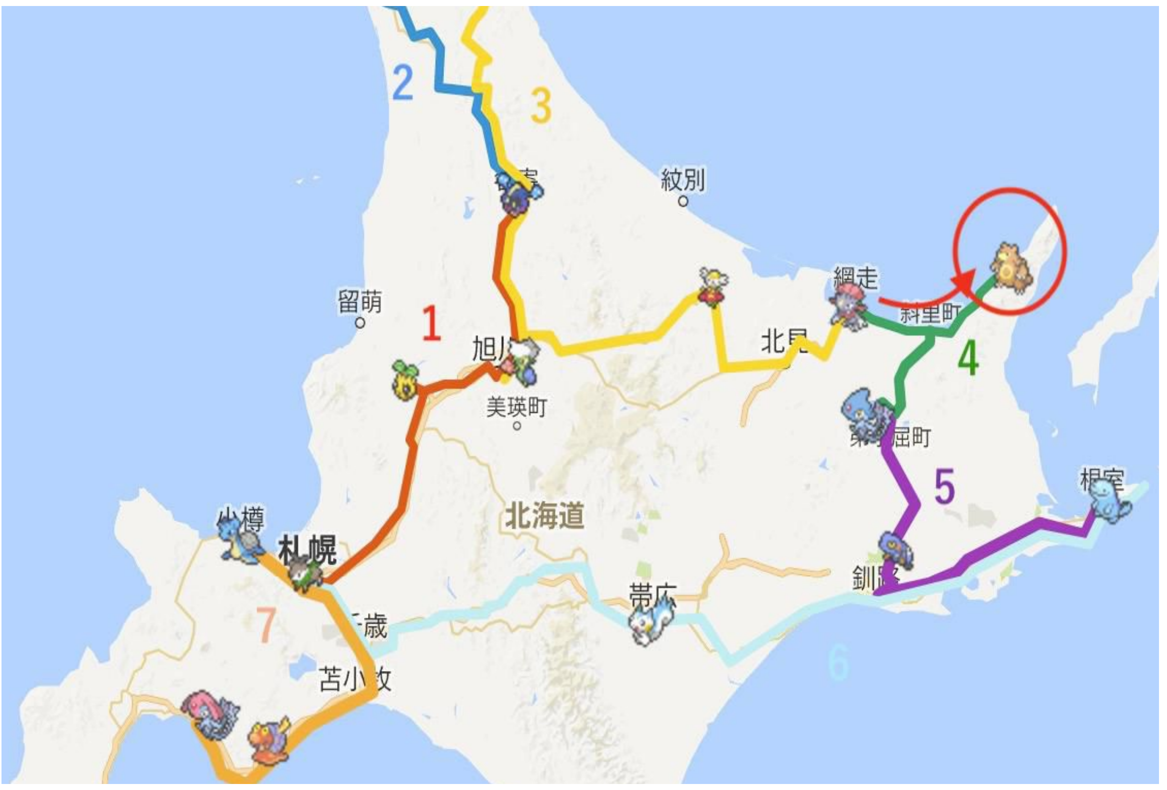


Visual reasoning
(Santoro et al. 2017)

Reasoning tasks



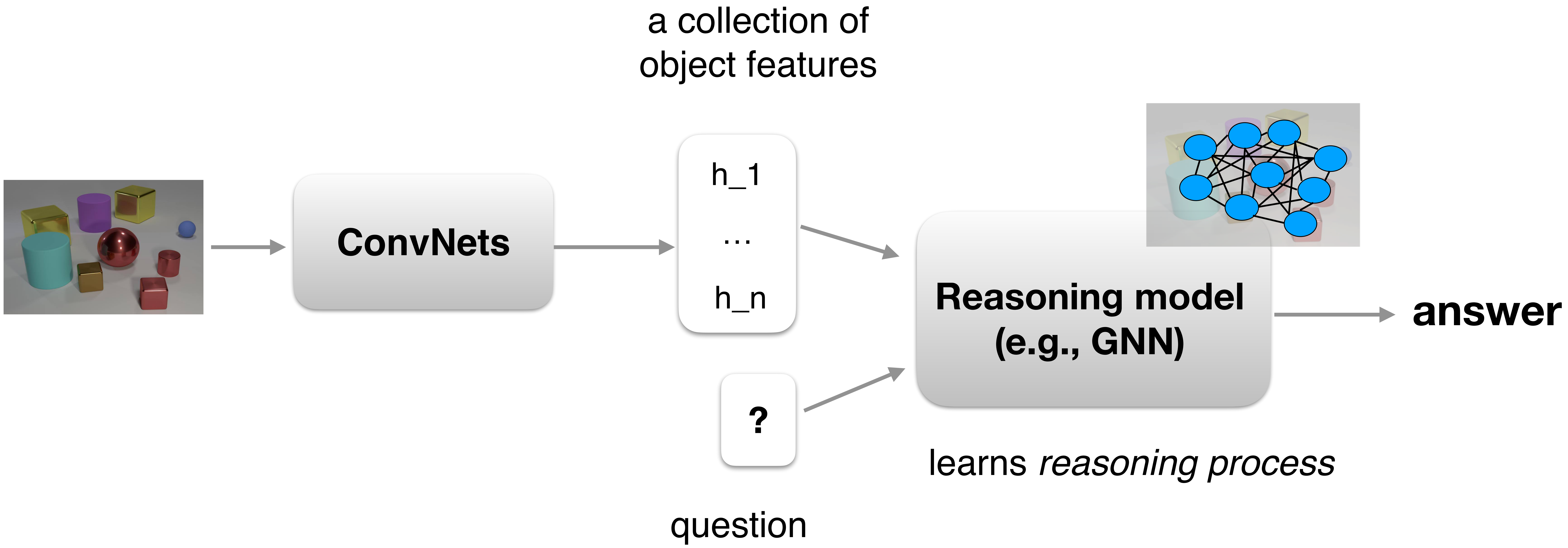
Furthest pair of objects?



Best path for Pokemon Go?

Next position of the blocks?

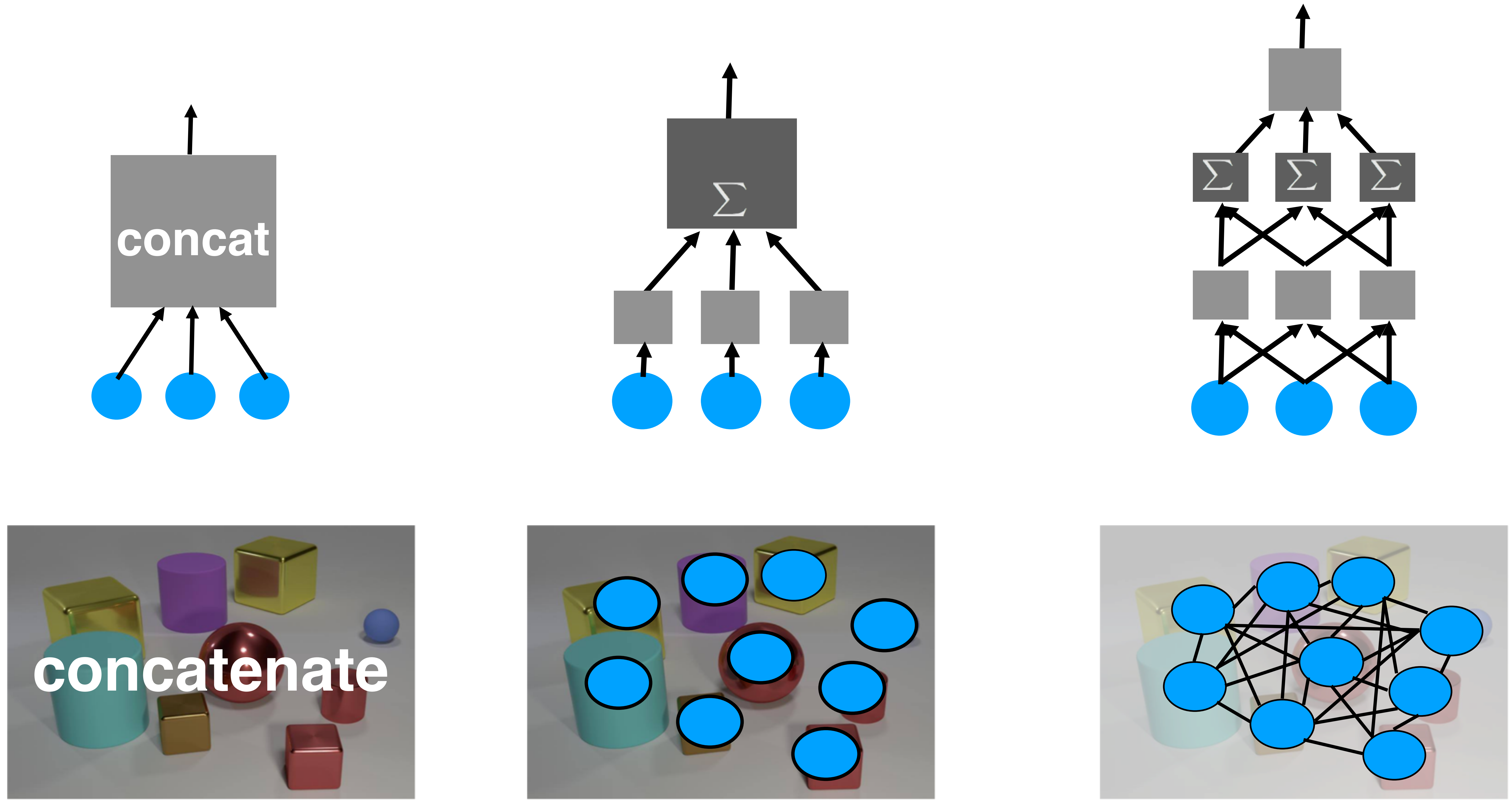
A typical pipeline of object-centric reasoning



(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…)

Architectures: capability of learning to reason

"Equal" expressive power (universal approximators), **big difference in generalization**

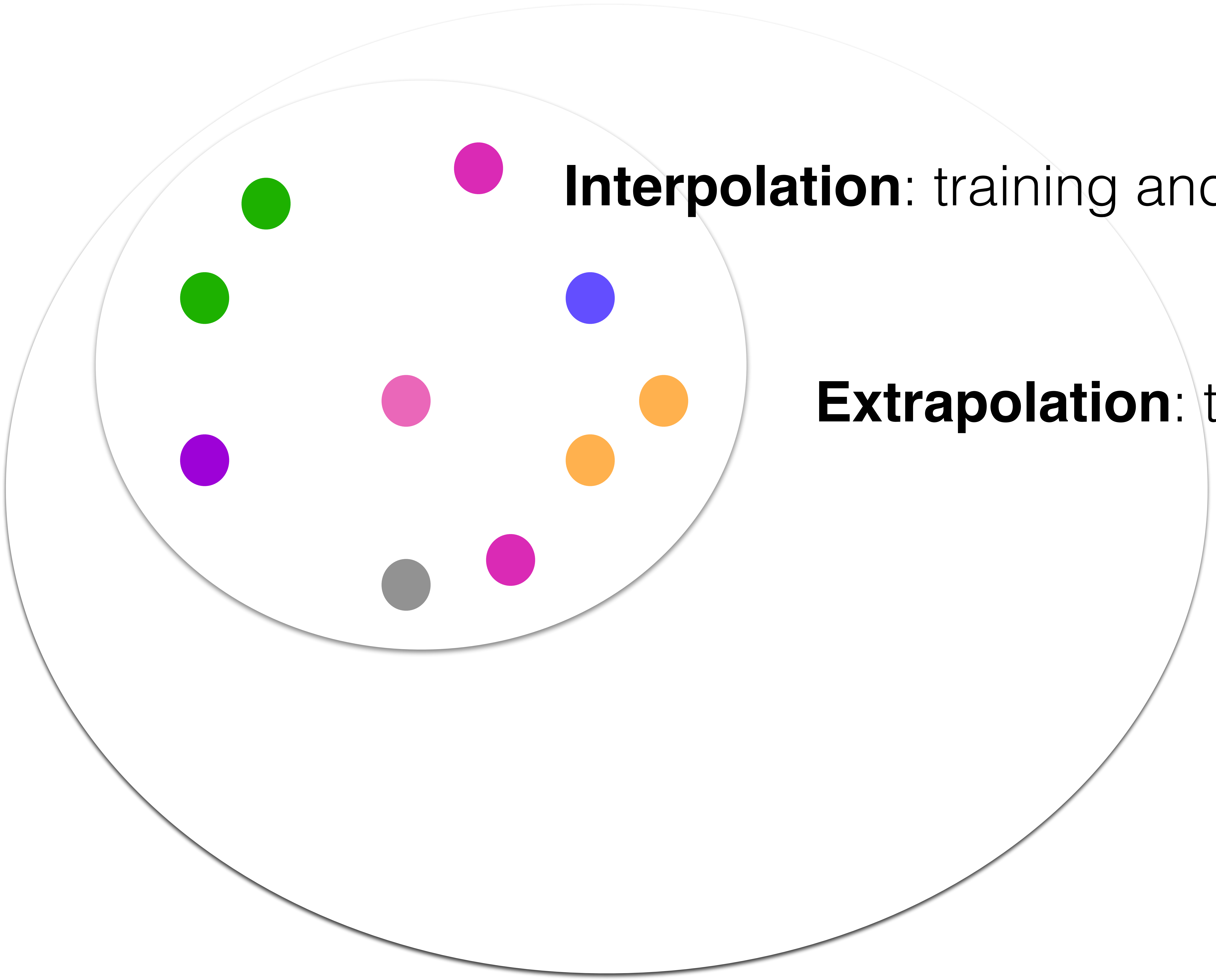


.....

```
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*

Generalization analysis: interpolation and extrapolation



Interpolation: training and test data from the same distribution

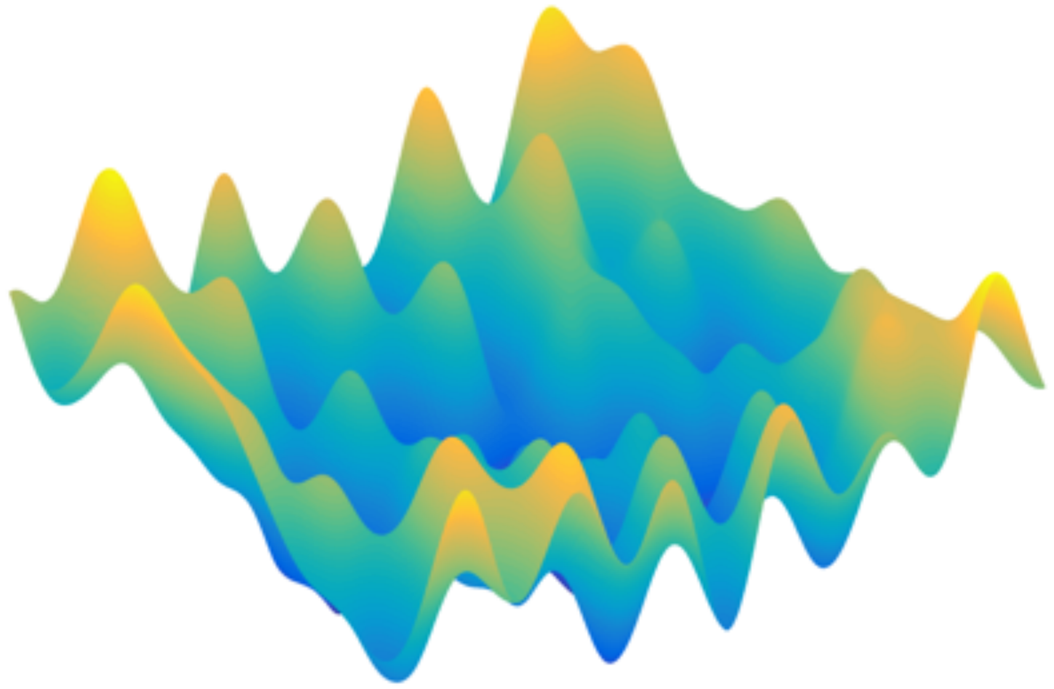
Extrapolation: test data outside the training distribution

$$L(f) \triangleq \mathbb{E}_{(x,y)\sim P}[\ell(f(x), y)]$$

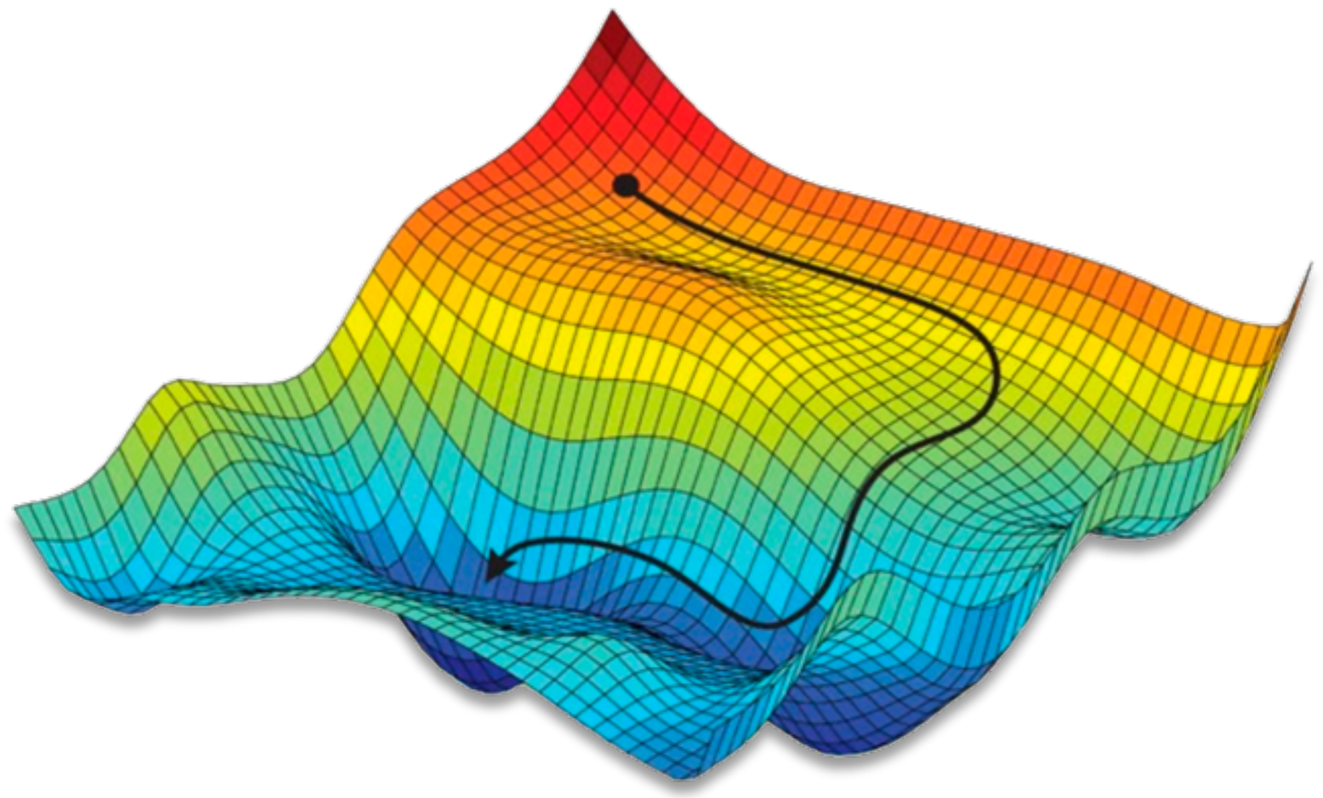
f: function learned by NN

P: test distribution

Approaches of generalization analysis



non-convex landscape



Parameter trajectory

$\theta_{GNN}(t)$

Norm based (covering number)

(Bartlett et al 2017, Golowich et al 2018, Garg et al 2020…)

Trajectory analysis (NTK)

(Jacot et al 2018, Arora et al. 2019, Du et al 2019…)

Inductive bias of network & trajectory analysis

(Xu et al. 2020, 2021…)

more "practical"

more assumptions

Formalizing inductive bias of architectures

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

Claim: Better algo alignment implies better generalization.

Graph Neural Network	**Bellman-Ford algorithm**
for k = 1 … GNN iter:	for k = 1 … |S| - 1:
for u in S:	for u in S:
$h_u^{(k)} = \Sigma_v \; \mathrm{MLP}(h_v^{(k-1)}, h_u^{(k-1)})$	$d[k][u] = \min_v d[k-1][v] + \mathrm{cost}(v, u)$

No need to learn for-loops

Learns a simple reasoning step

Without good alignment -> need to learn complicated functions e.g., for-loop

Alignment measure

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.

$$\mathbb{P}_{x \sim \mathcal{D}} \left[\| f(x) - g(x) \| \leq \epsilon \right] \geq 1 - \delta$$

(Valiant 1984)

learned function

true function (algorithm)

* *Sample complexity of learning simple modules can be estimated via e.g., NTK (Arora et al. 2019)*

Better alignment implies better generalization

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.

- *Lipschitznes and SGD sequential training*
- *Related work experimenting assumptions: Veličković et al 2020*

GNNs sample-efficiently learn dynamic programming

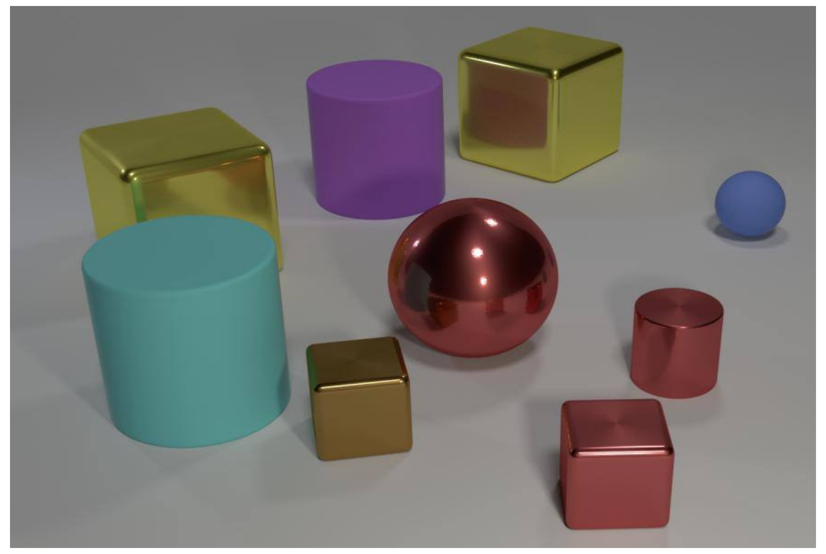
DP-Update: simple module easily learned by GNN's MLP modules

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

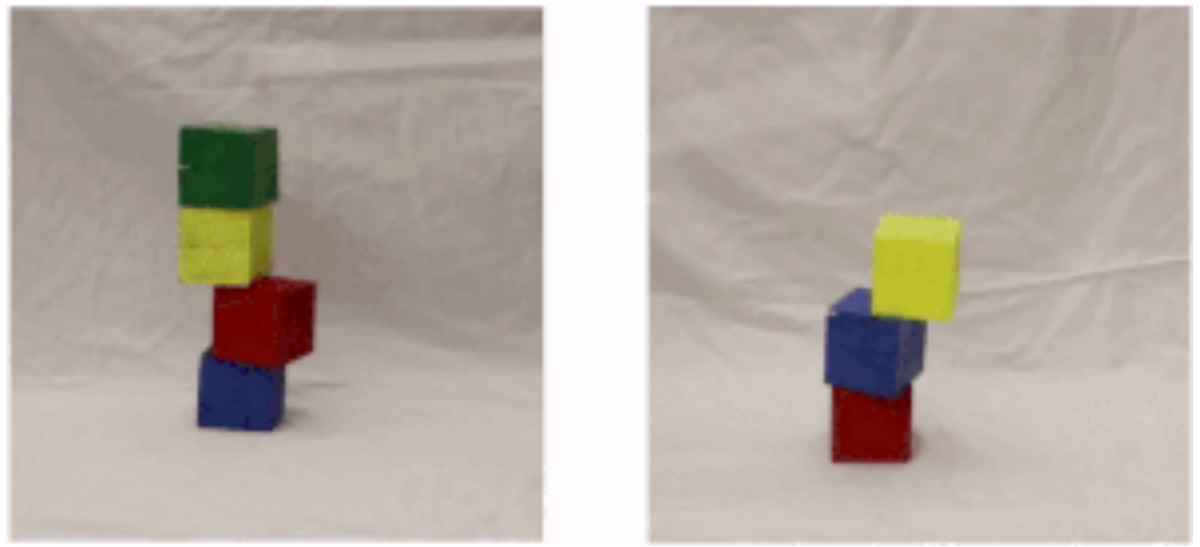
$$h_s^{(k)} = \sum\nolimits_{t \in S} \mathbf{MLP}_1^{(k)}\left(h_s^{(k-1)}, h_t^{(k-1)}\right)$$

Reasoning tasks as DP:

many graph algorithms



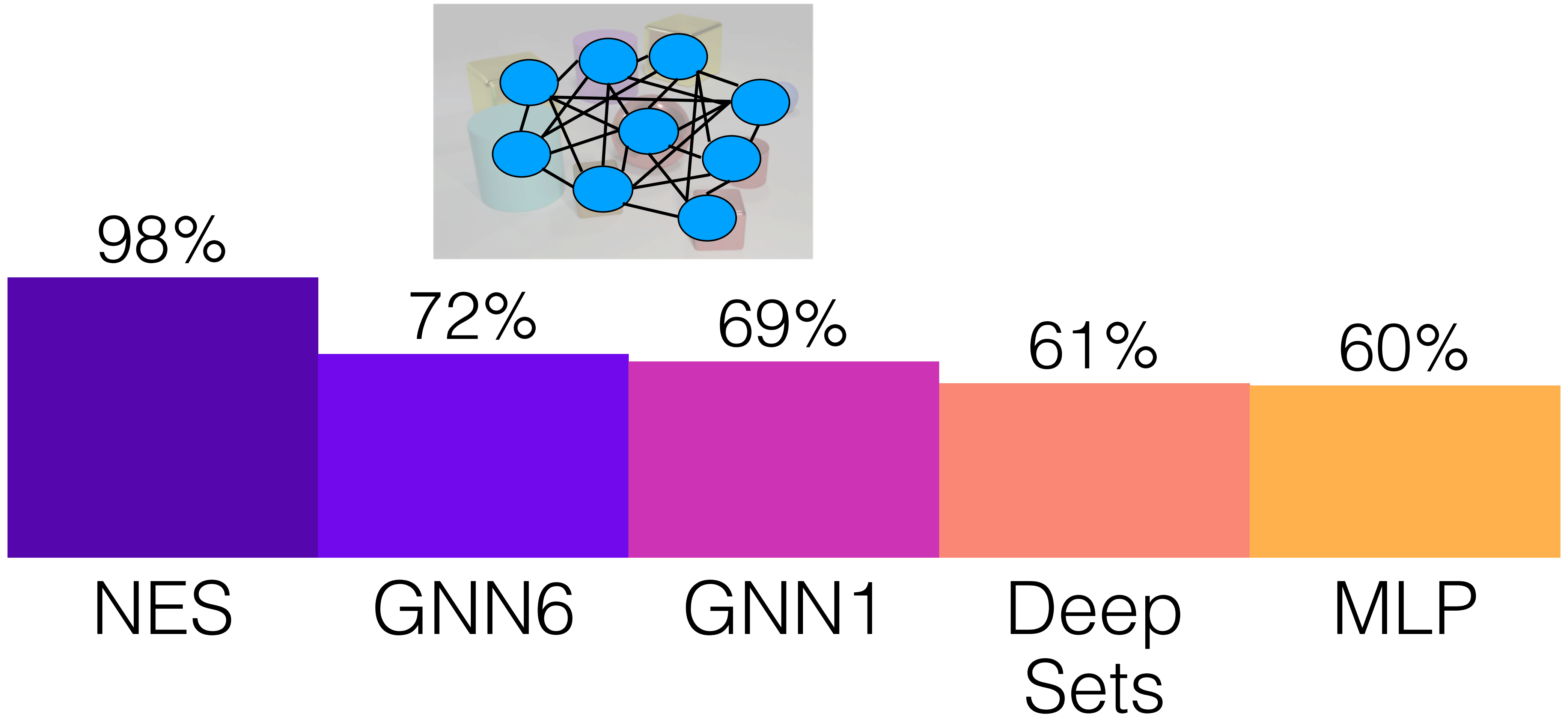
visual question answering



Intuitive physics

Limits of GNN: NP-hard problem

Subset sum: Can any subset of a set of numbers sum to zero?

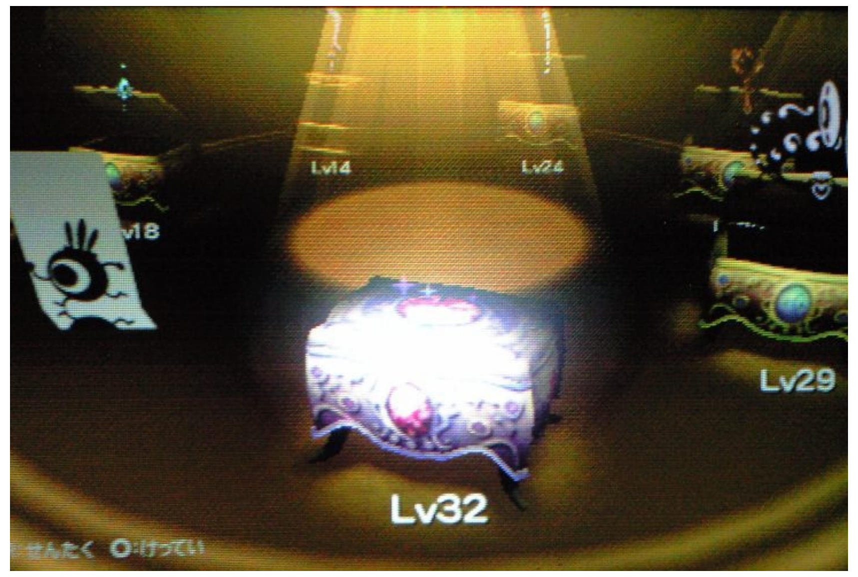


NES (Neural Exhaustive Search) - based on **algo alignment**

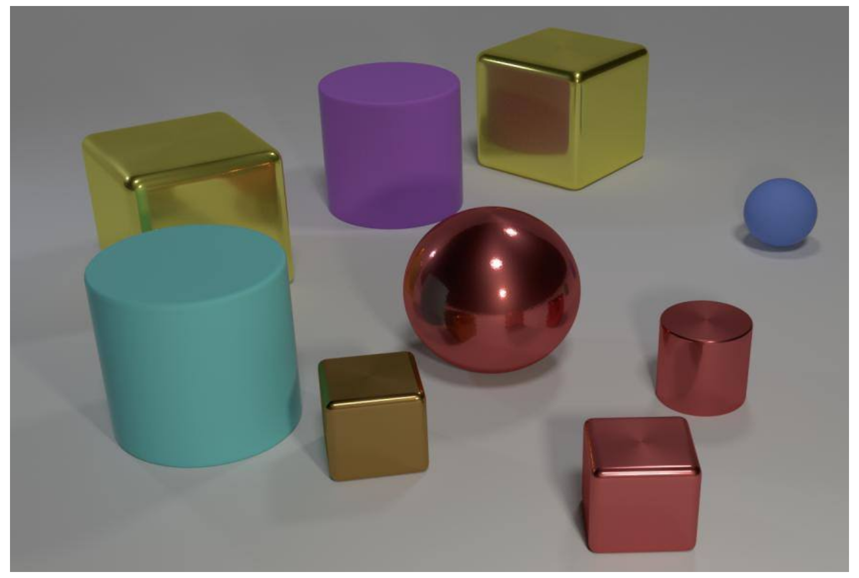
$$\mathrm{MLP}_2(\max_{\tau \subseteq S} \mathrm{MLP}_1 \circ \mathrm{LSTM}(X_1, ..., X_{|\tau|} : X_1, ..., X_{|\tau|} \in \tau))$$

$y = \max_S 1[\, h(S) = 0 \,]$, $\quad h(S) = \Sigma_{x \text{ in } S}\ X$

A hierarchy of tasks

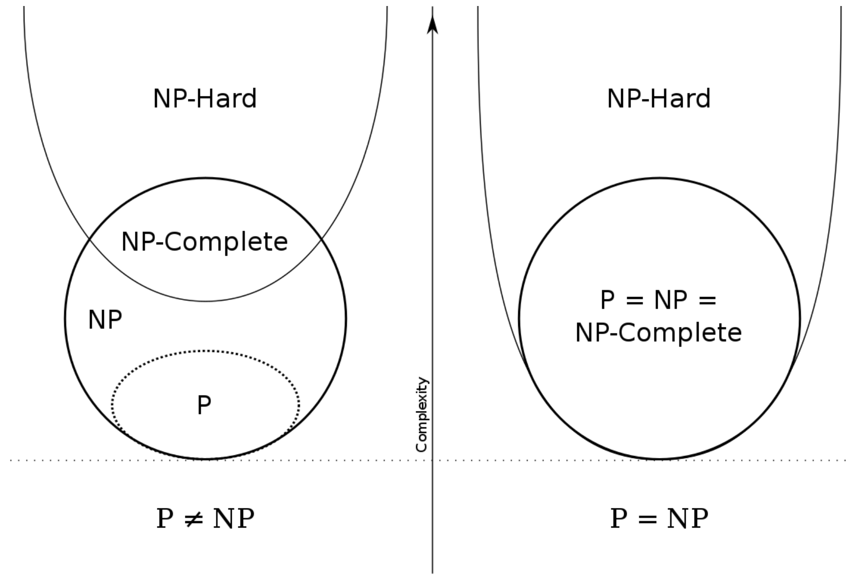


Summary statistics
What is the maximum value difference among treasures?



Relational argmax
What are the colors of the furthest pair of objects?

Dynamic programming
What is the cost to defeat monster X by following the optimal path?

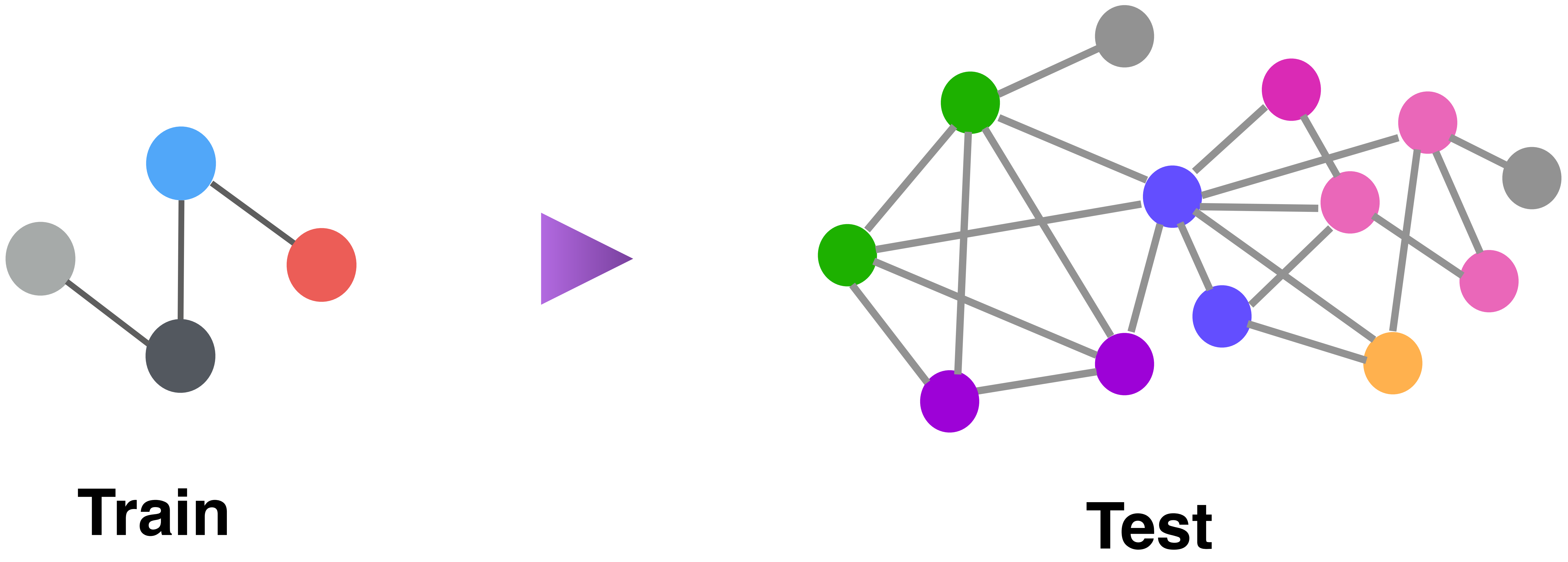


NP-hard problem
Subset sum: Is there a subset that sums to 0?

	Summary statistics	Relational argmax	Dynamic programming	NP-hard problem
Graph Neural Network (GNN)	✓	✓	✓	✗
DeepSets *(Zaheer et al. 2017)*	✓	✗	✗	✗
MLP	✗	✗	✗	✗
Neural Exhaustive Search (NES)				✓

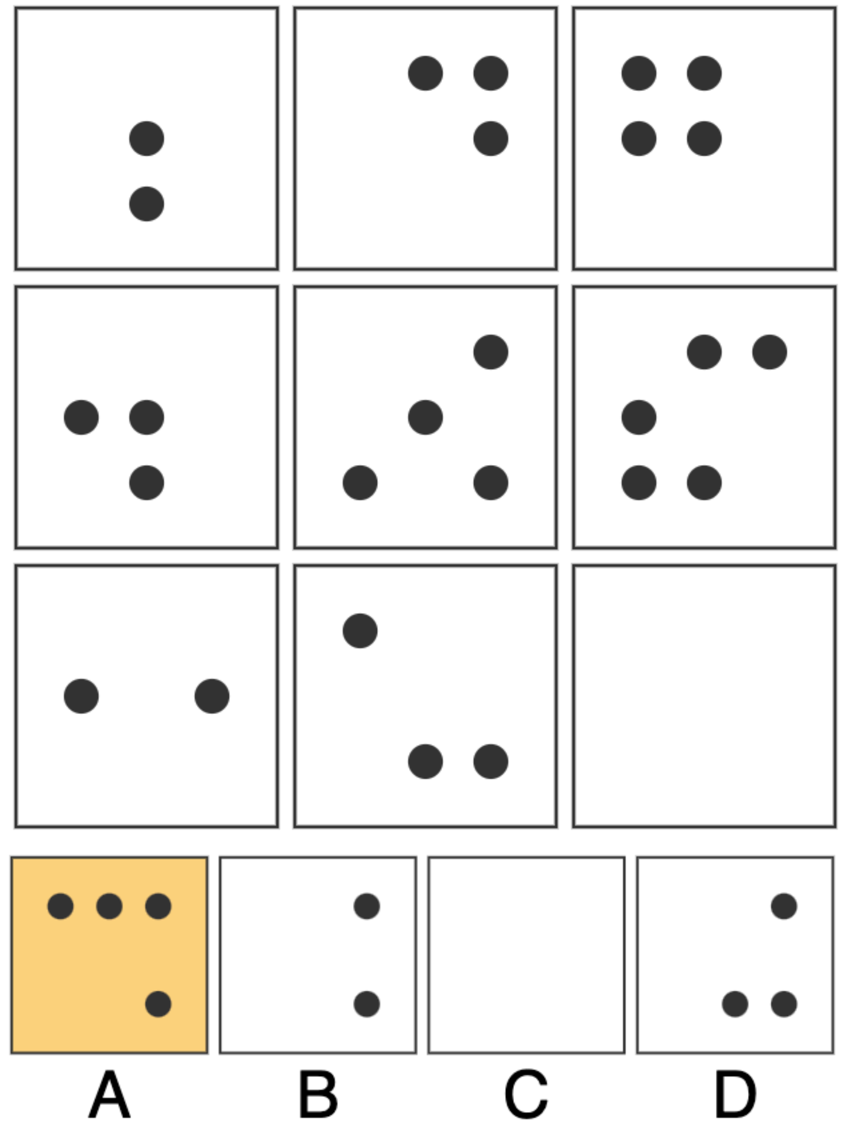
Extrapolation

What function does a neural network trained by GD implement outside the support of the training distribution?



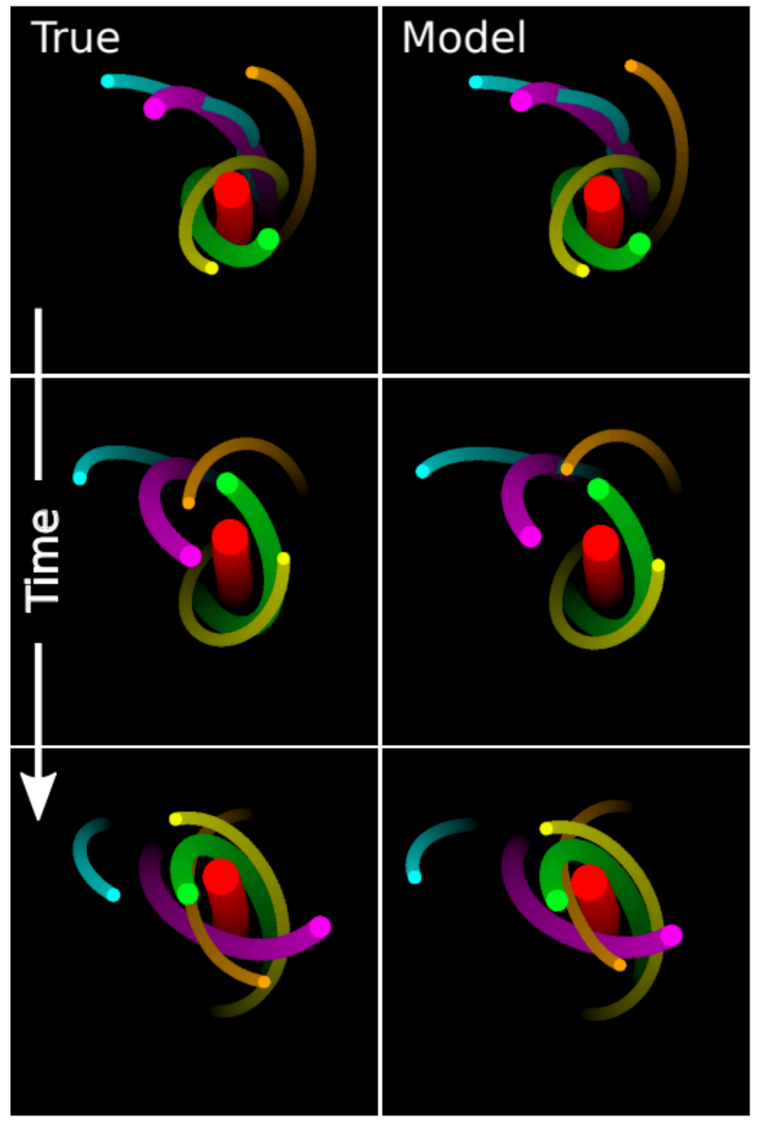
Generalize across graph structure, size, node & edge features?

Evaluation of extrapolation in literature



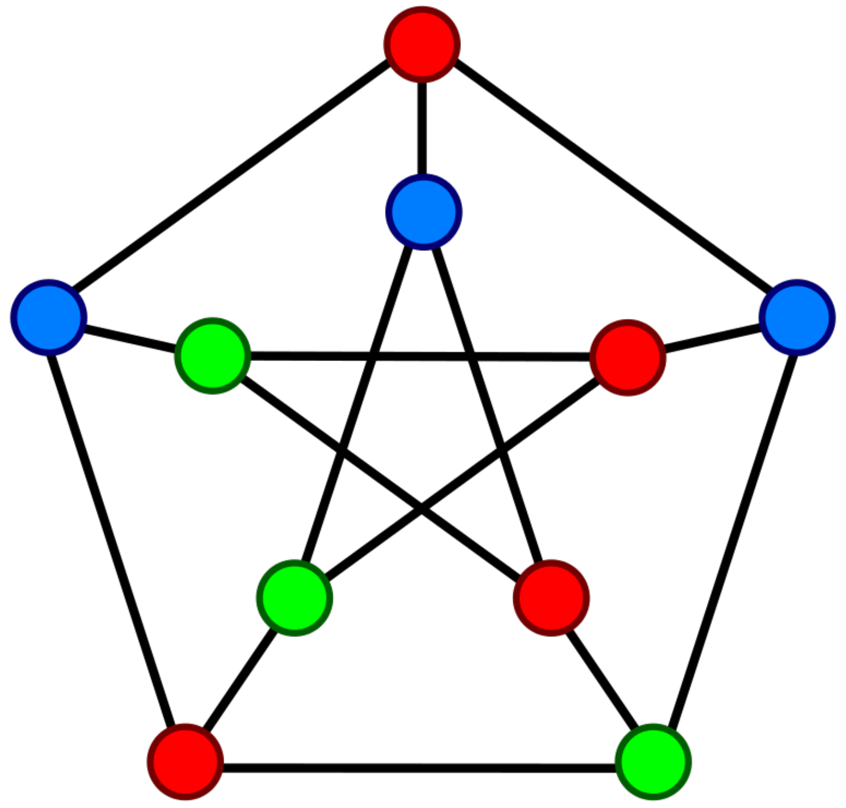
IQ tests

(Santoro et al. 2018)



n-body system

(Battagalia et al. 2016)



Graph algorithms

(Battagalia et al. 2018, Dai et al 2018, Velickovic et al 2020…)

CNN, MLP fail to extrapolate;
CNN+GNN better, still not ideal

GNN “reasonable” accuracy
on larger systems

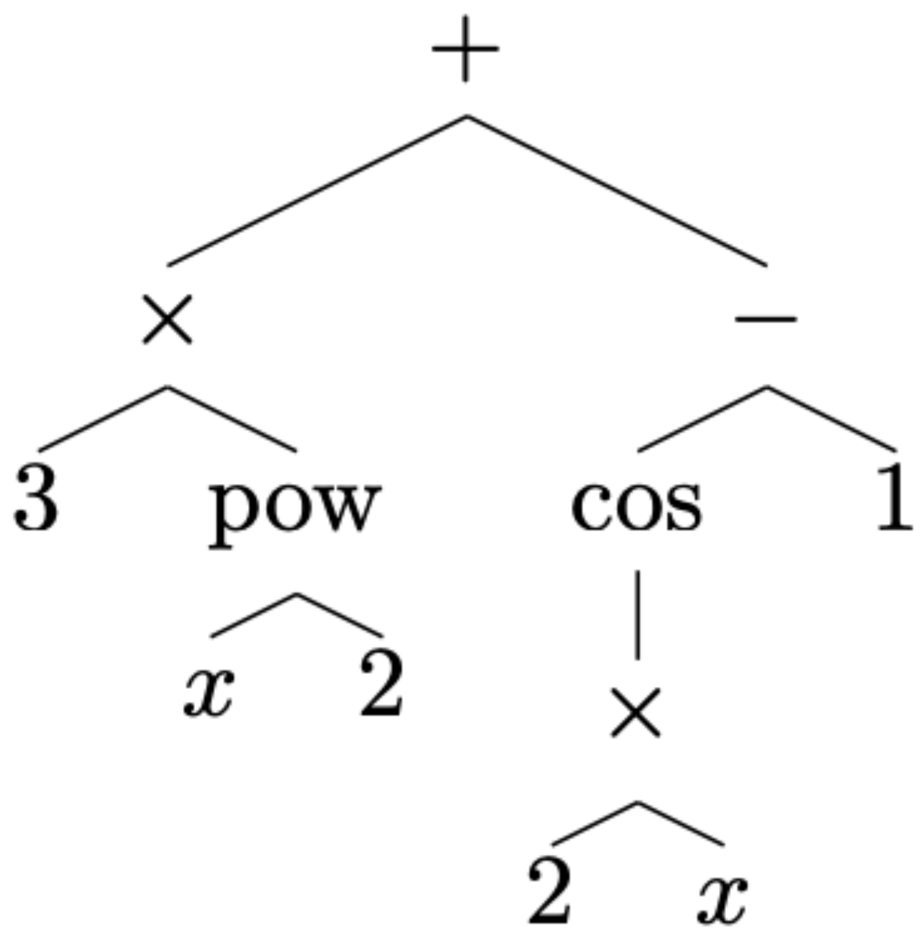
Certain modified GNNs
perform well on larger graphs

Evaluation of extrapolation in literature

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`

Transformer better than LSTM, but performance still not ideal

(Saxton et al. 2019)



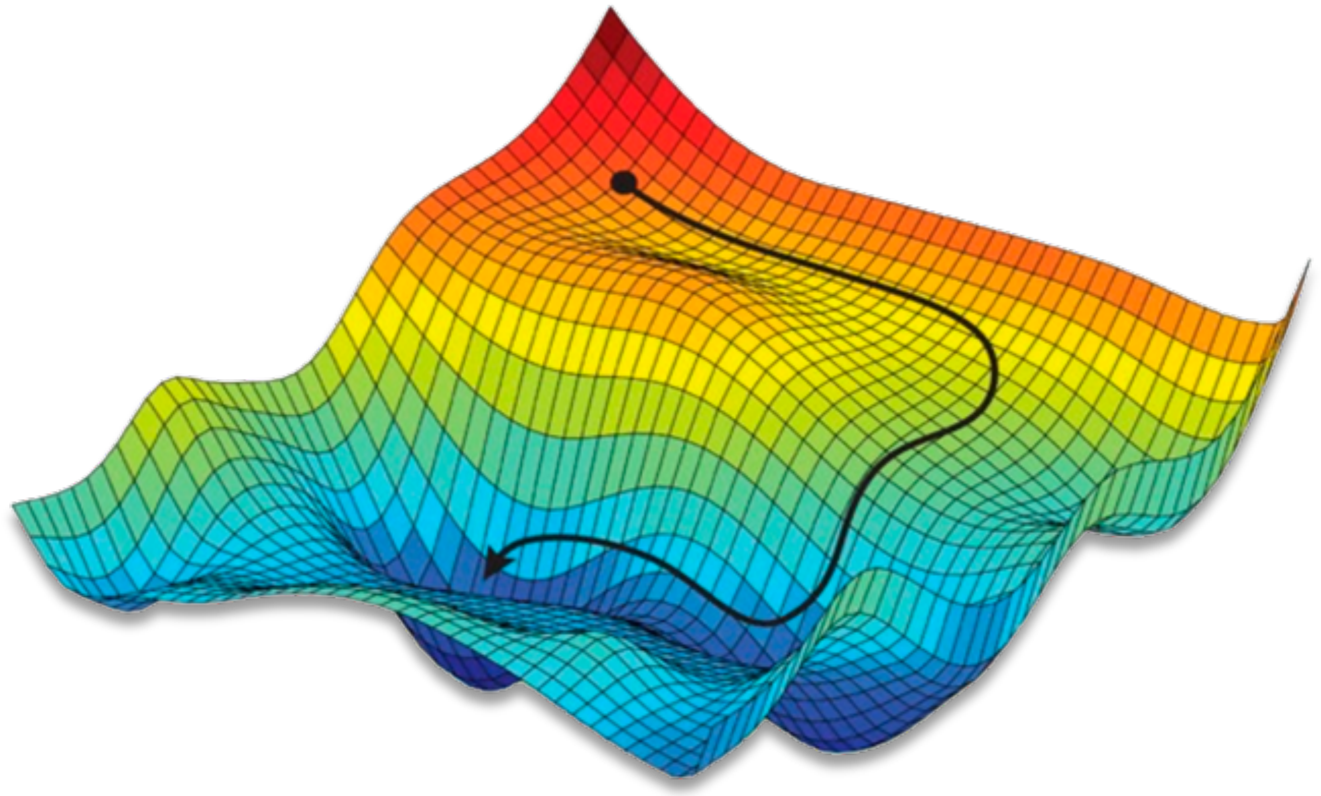
Transformer extrapolates well with specialized symbolic inputs

(Lample et al. 2020)

Puzzle

MLP and CNN usually fail to extrapolate, but GNNs extrapolate well in some cases.

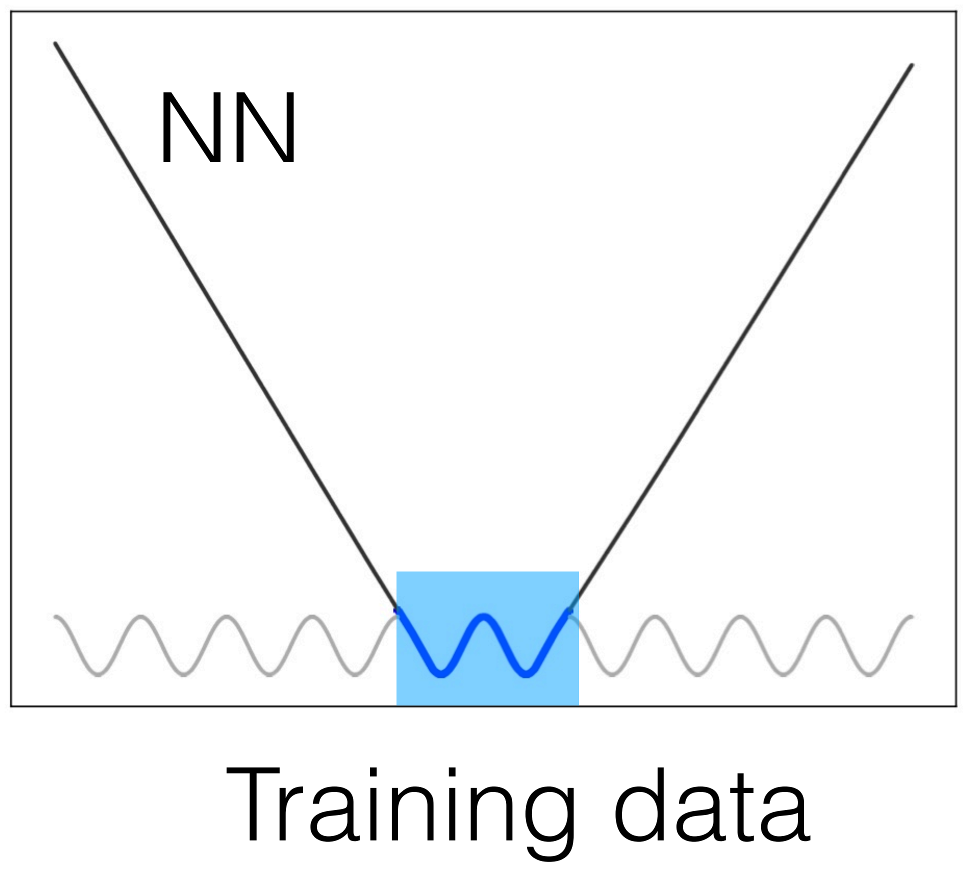
How do neural networks extrapolate?

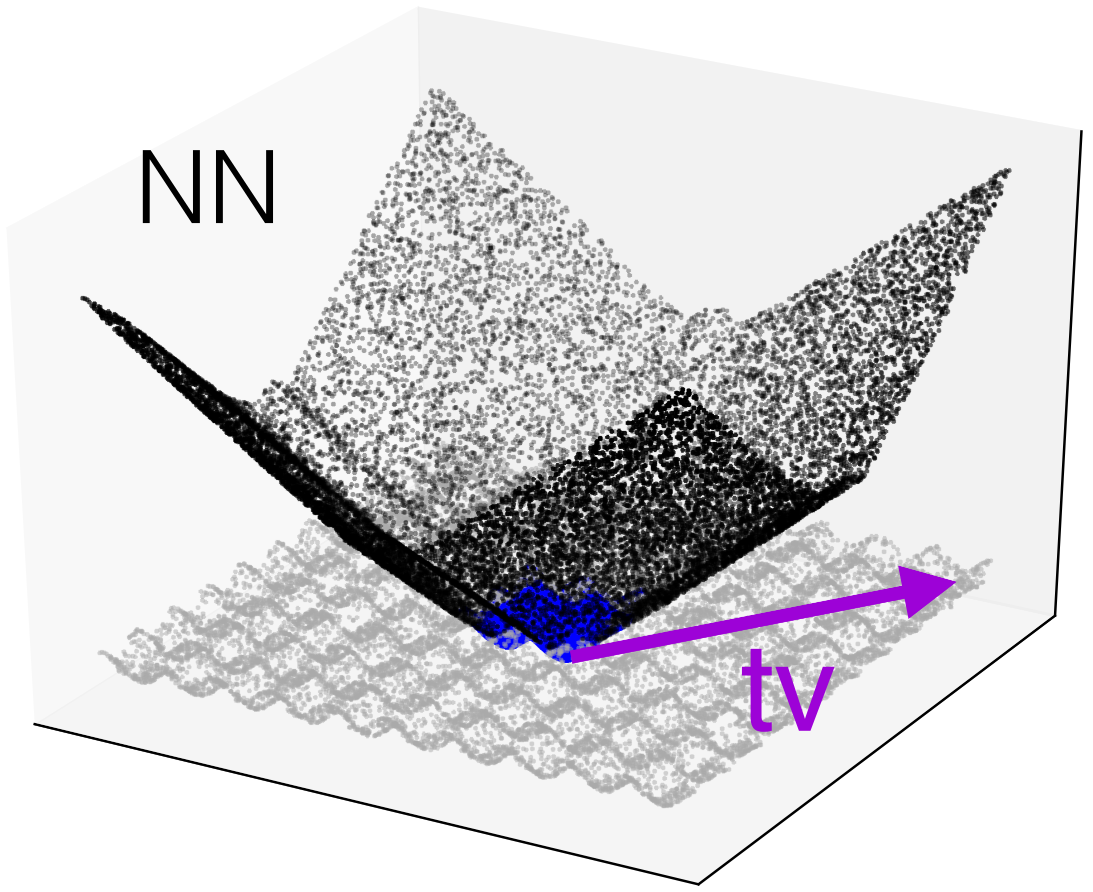


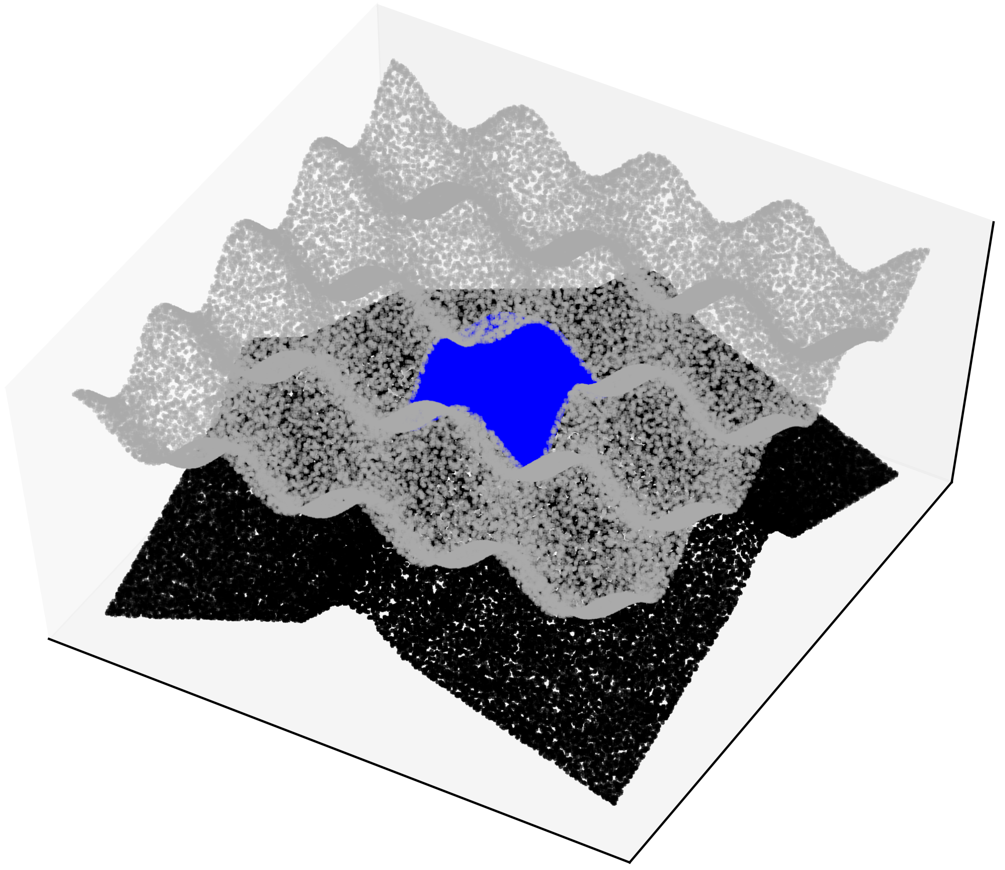
Parameter trajectory
$\theta(t)$

Depends on inductive bias of gradient descent training and neural network.

Linear extrapolation behavior of ReLU MLPs



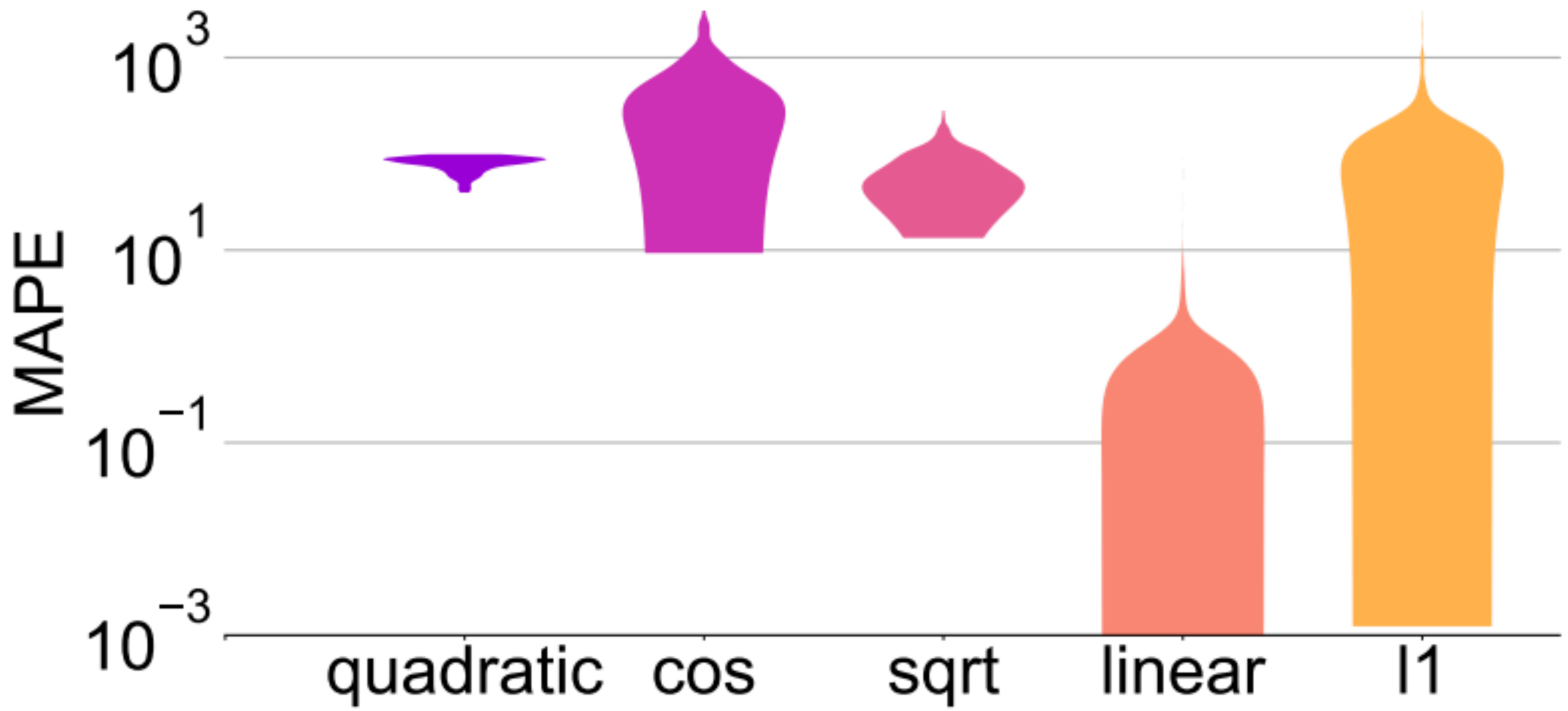




Theorem (XZLDKJ'20)
Let f be a two-layer ReLU MLP trained by GD*. For any direction $v \in \mathbb{R}^d$, let $x = tv$. For any $h > 0$, as $t \to \infty$, $f(x + hv) - f(x) \to \beta_v h$ with rate $O(1/t)$

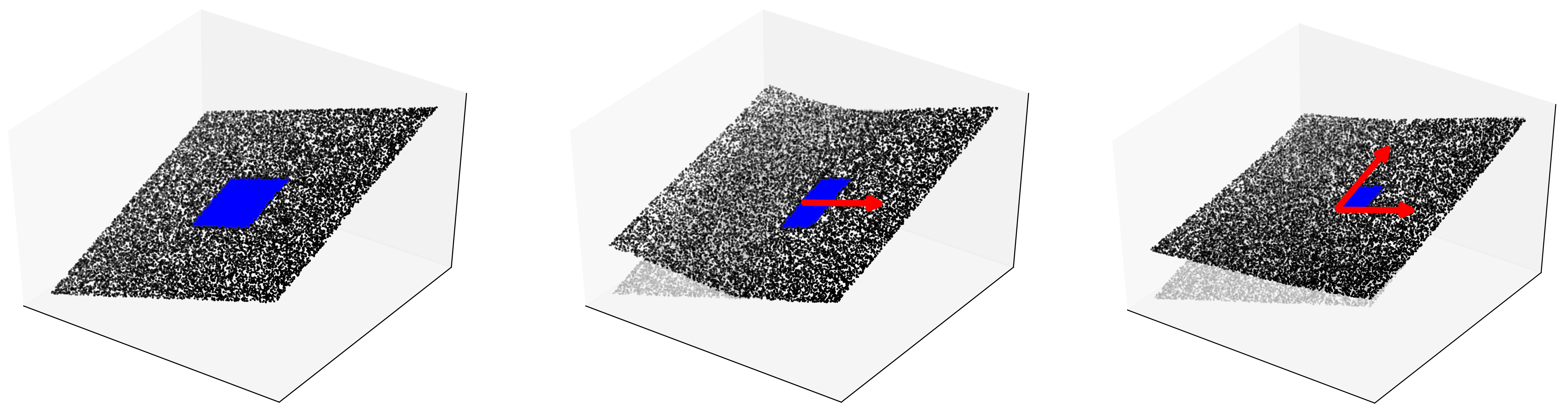
* *Assumption: NTK regime*

Implication of linear extrapolation



MAPE extrapolation error: lower the better

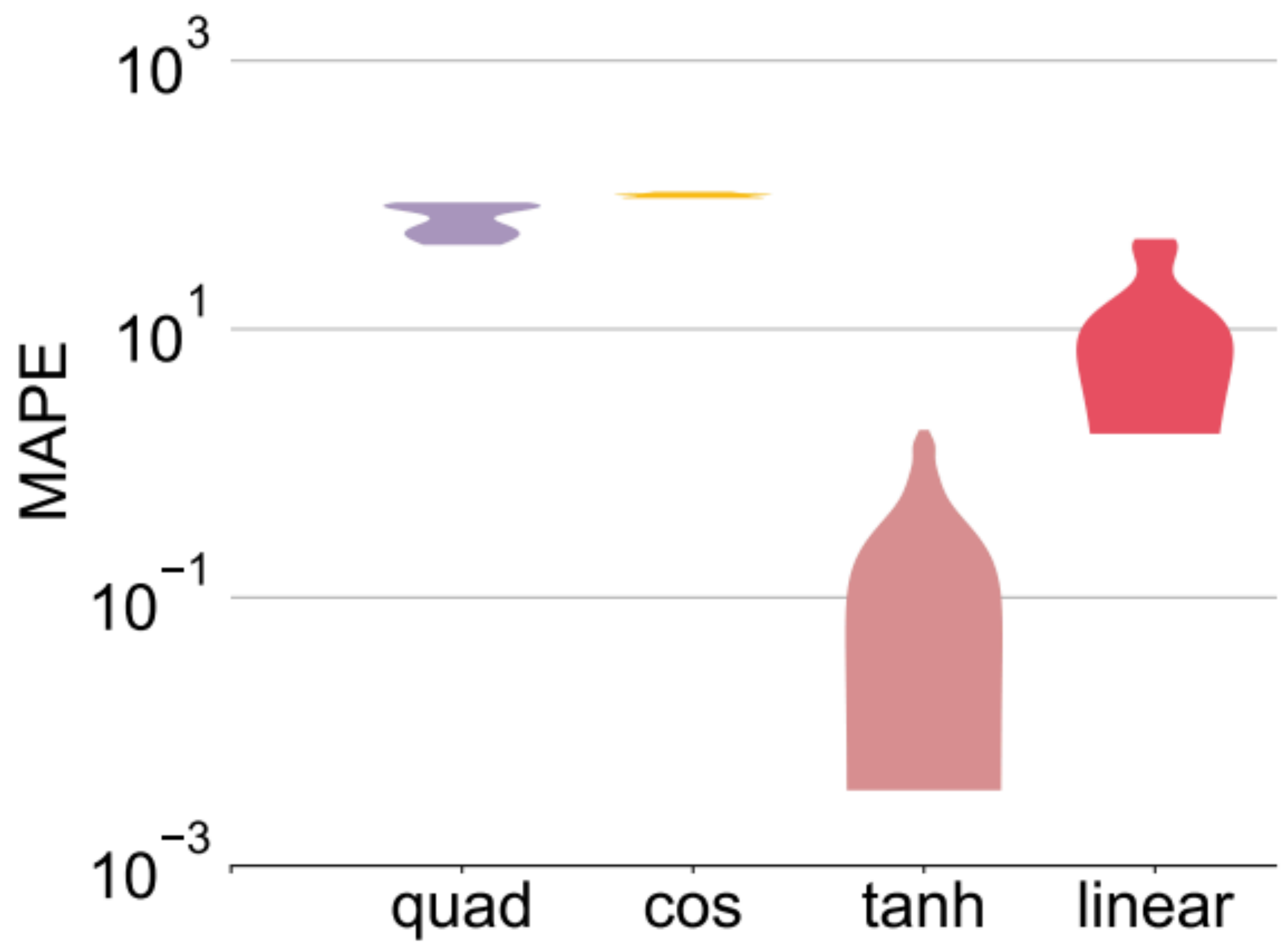
Provable learning of linear functions with diverse training data



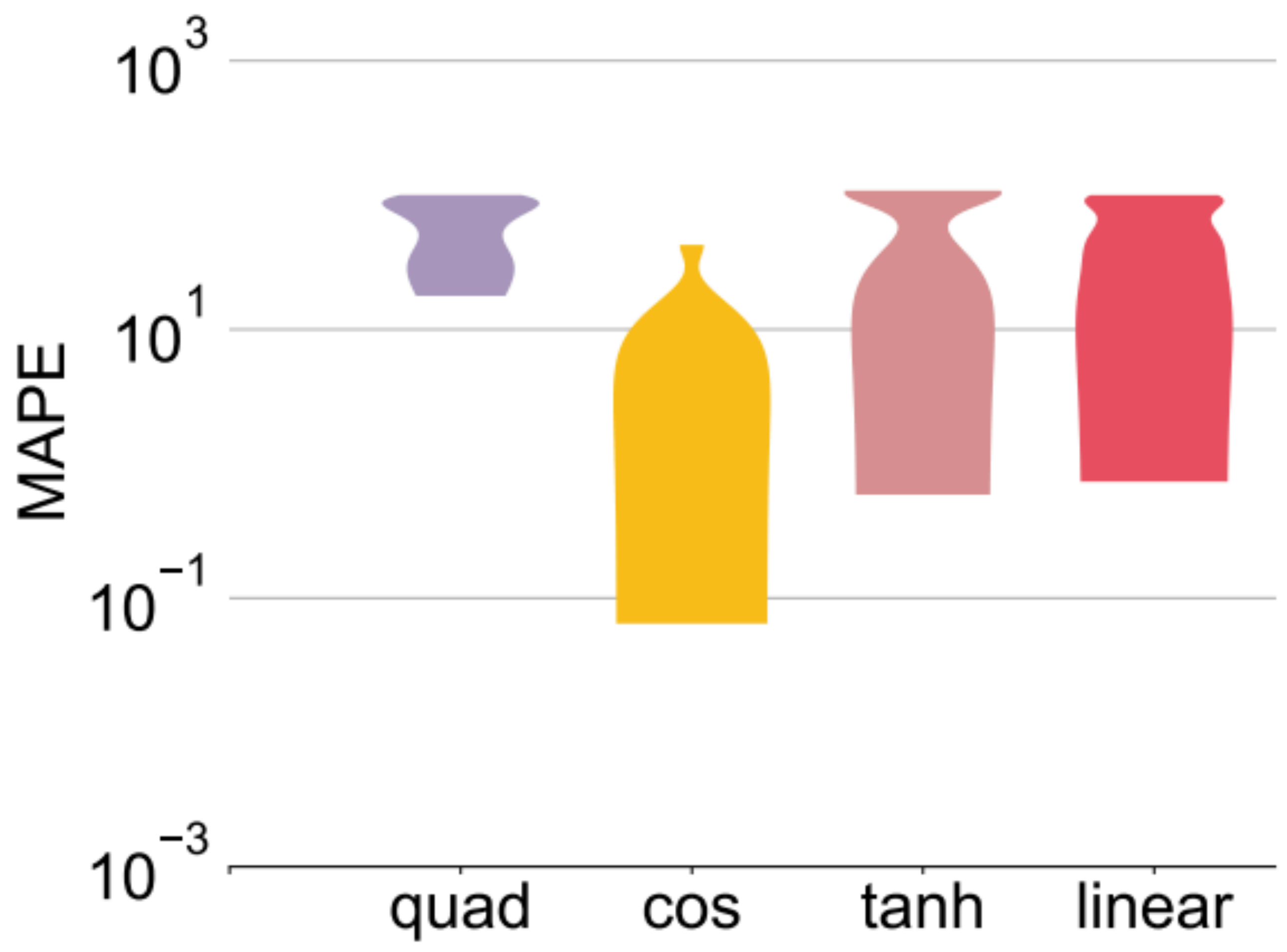
Theorem (XZLDKJ'20)
Let f be a two-layer ReLU MLP trained by GD*. Suppose target function is $\beta^\top x$ and support of training distribution covers all directions. As the number of training examples $n \to \infty$, $f(x) \to \beta^\top x$.

* *Assumption: NTK regime*

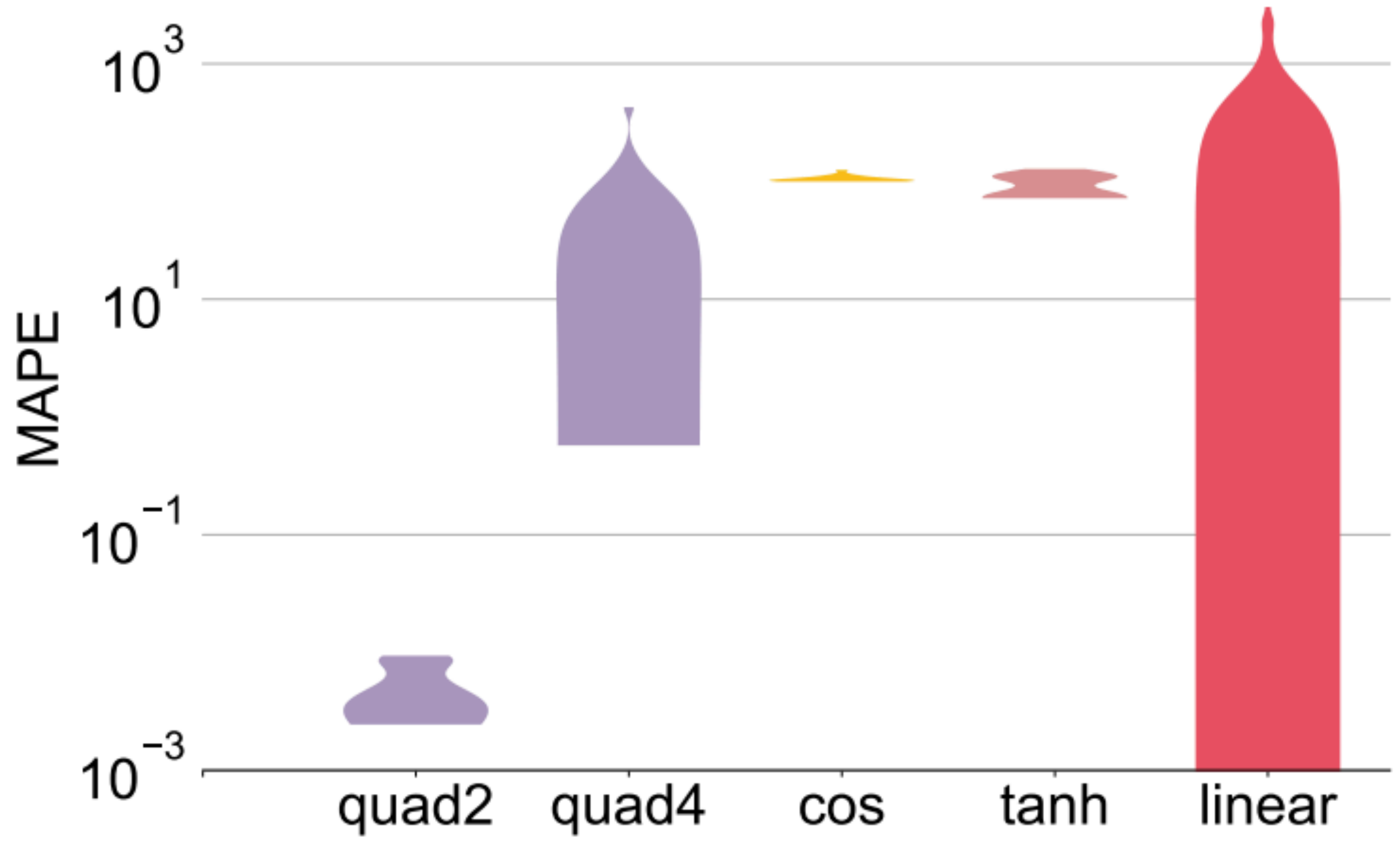
Feedforward networks with other activation



(a) tanh activation



(b) cosine activation



(c) quadratic activation

Extrapolates well if activation is “similar” to target function

Implications for GNNs

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

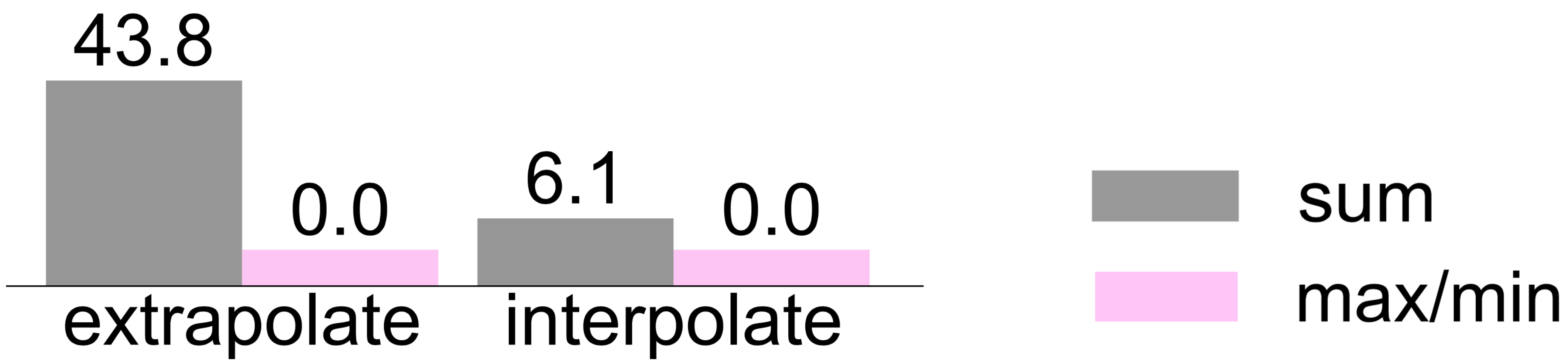
GNN (sum): $$h_u^{(k)} = \boxed{\Sigma_v} \ \mathbf{MLP}^{(k)}\left(h_v^{(k-1)}, h_u^{(k-1)}, w(v,u)\right)$$

✗ *MLP has to learn non-linear steps*

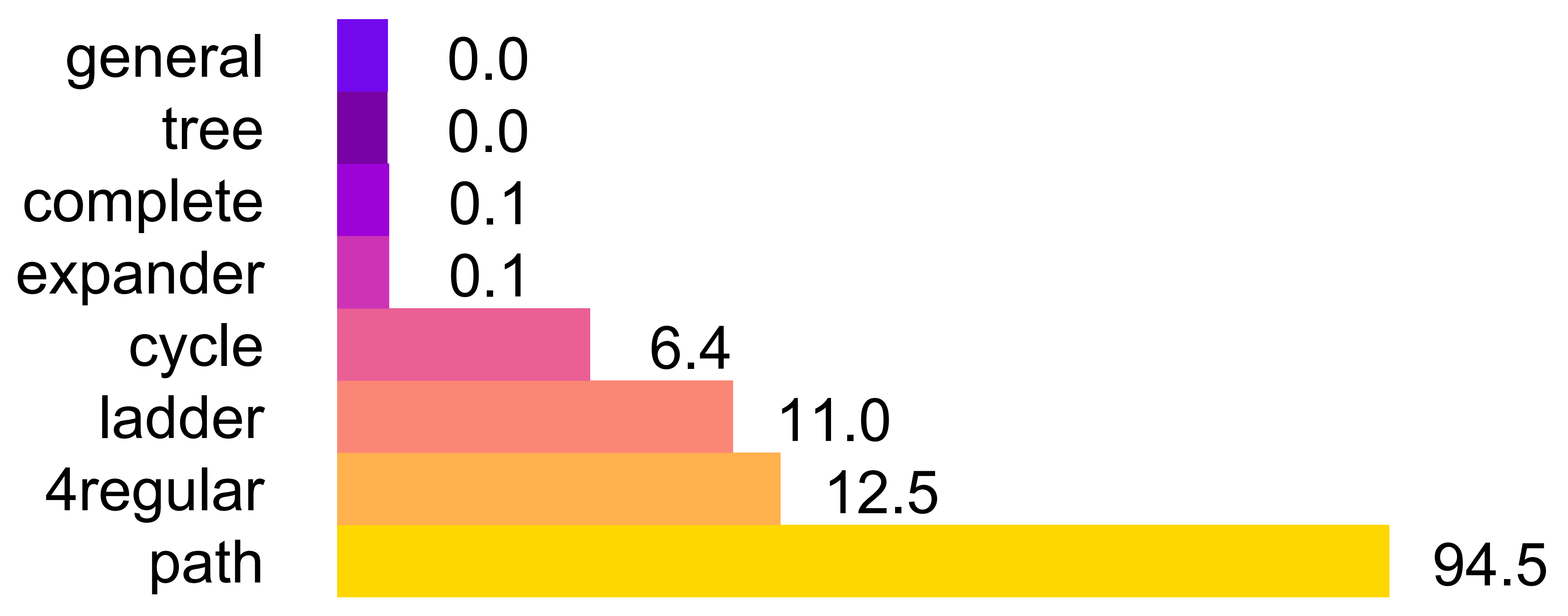
Some works extrapolate with:

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

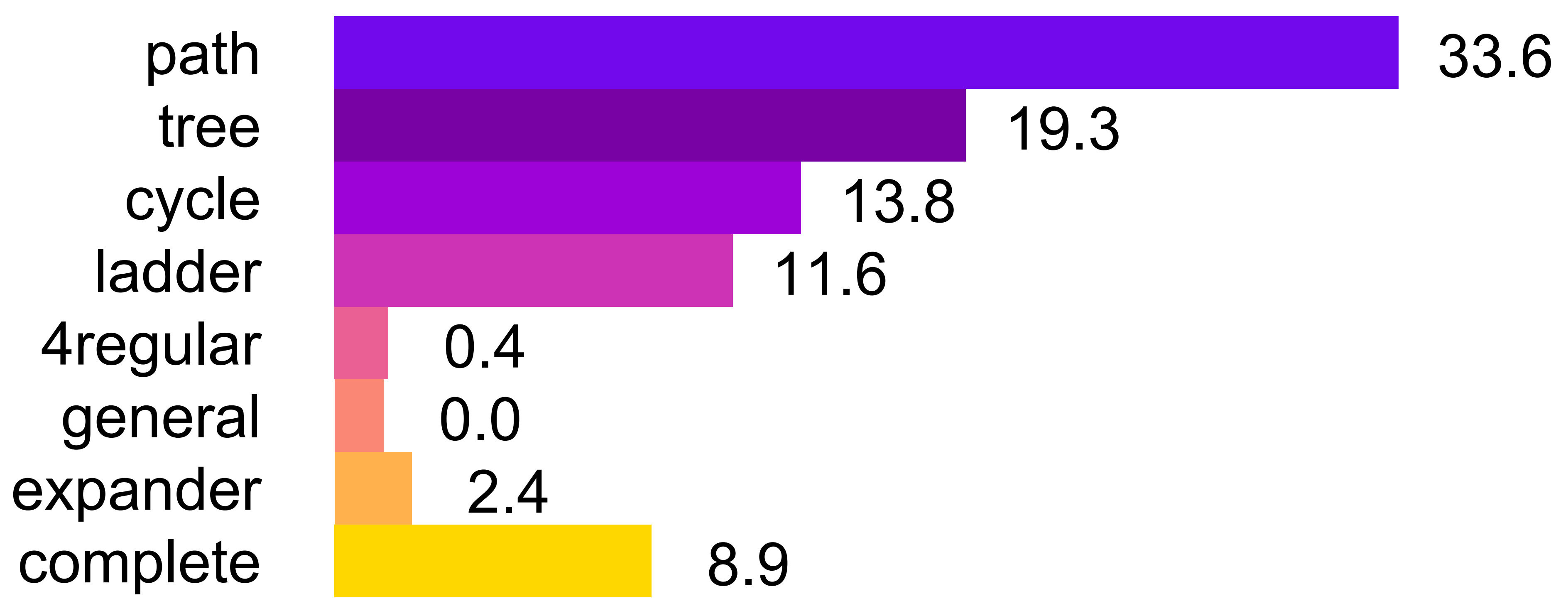
✓ *MLP learns linear steps*



Provable extrapolation: architecture and graph structure



Max Degree



Shortest Path

Proposition (XZLDKJ'20)

A max-aggregation GNN 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

Linear algorithmic alignment (XZLDKJ’20)
Network can simulate algorithm via ~~*easy-to-learn*~~ *linear* “modules”.

Hypothesis: Linear algo alignment helps *extrapolation.*

Encoding nonlinearity in architecture or input representation

Encoding nonlinearities in architecture

Symbolic operation, activation, pooling etc…

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

✔ *MLP learns linear steps*

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

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

Encode exp log for learning multiplication

(Trask et al. 2018)

Q: What direction is the closest creature facing?

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

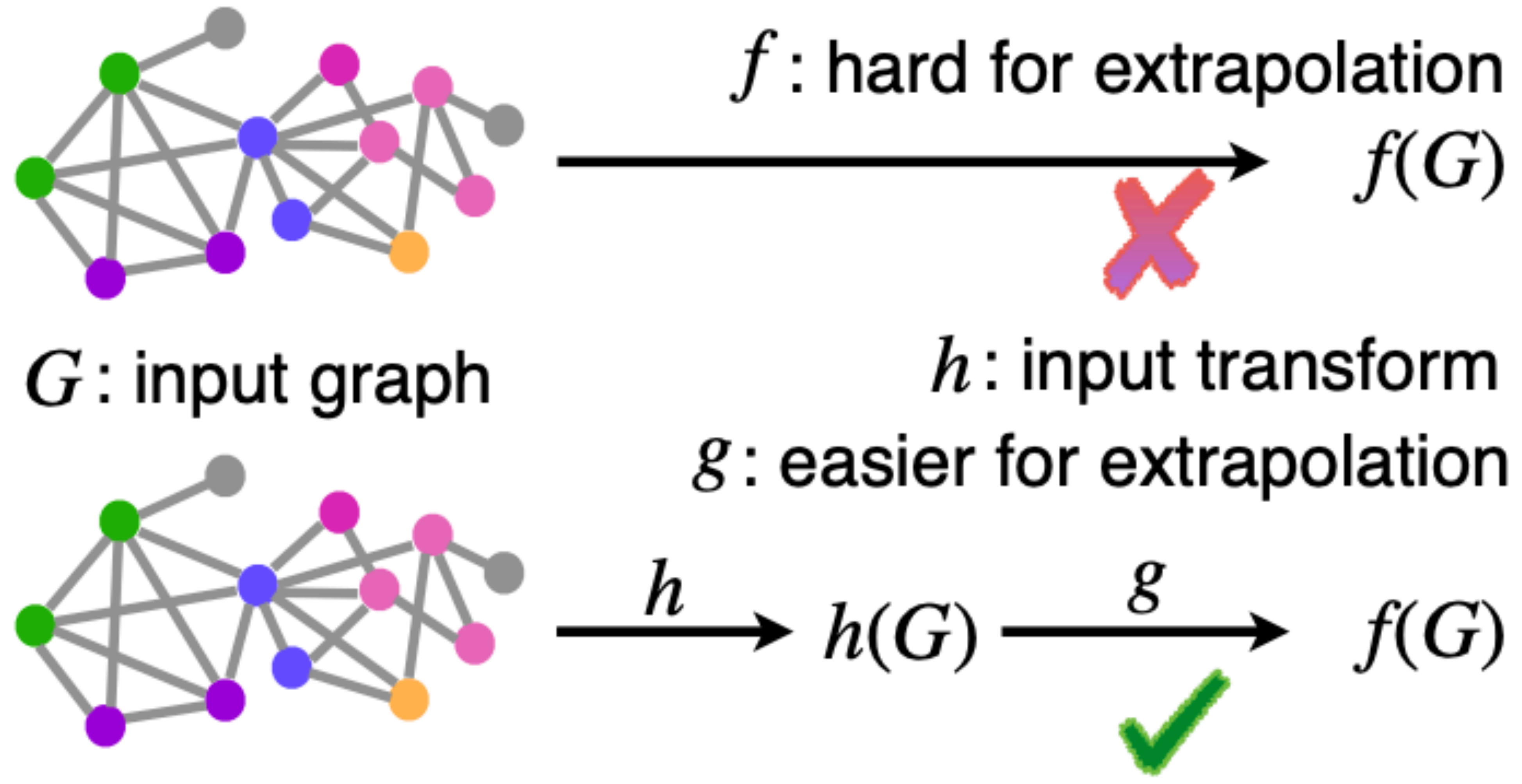
A: left

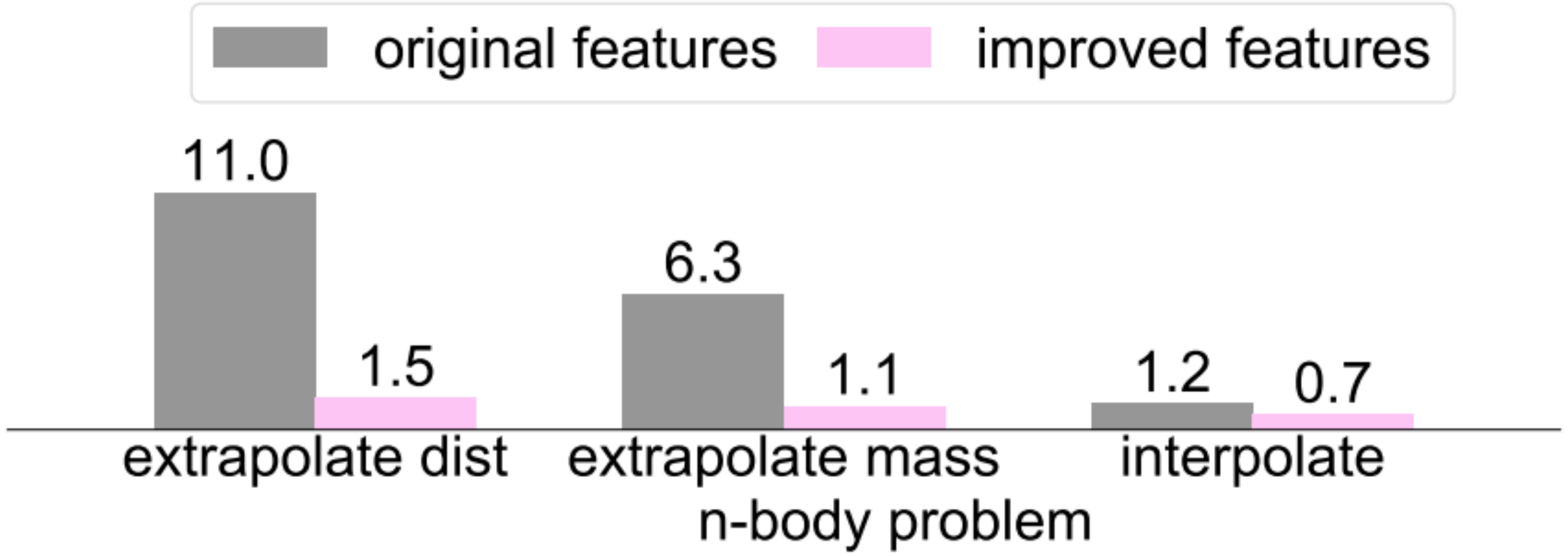
Encode a library of programs (~2K)

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

Encoding nonlinearities in input representation

Feature engineering, representation learning with large-scale out-of-distribution data (e.g., BERT)…





Summary

Generalization: Inductive bias via alignment of architecture and task

Extrapolation: Nonlinearities (network and representation) matter

Graph Neural Tangent Kernel: Fusing Graph Neural Networks with Graph Kernels. S. S. Du, K. Hou, B. Poczos, R. Salakhutdinov, R. Wang, K. Xu. NeurIPS 2019.

What Can Neural Networks Reason About? K. Xu, J. Li, M. Zhang, S. S. Du, K. Kawarabayashi, S. Jegelka. ICLR 2020.

How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. K. Xu, M. Zhang, J. Li, S. S. Du, K. Kawarabayashi, S. Jegelka. ICLR 2021.