On the Extrapolation of Graph Neural Networks and Beyond

Yihe Deng, Yu Yang, Qing Li, Shuwen Qiu

Outline

- Introduction
- Background and related work
- Experiment design and results
  - Max-degree and shortest-path
  - N-body
  - Handwritten Arithmetic with Integers
- Conclusion

Extrapolation

- How GNNs extrapolate, i.e. what does a GNN learn outside the support of the training distribution?

Generalization/Interpolation: new, unseen data drawn from the same distribution as the training data.

Extrapolation: test data could be outside the training distribution.

Extrapolation (formal definition)

- Given a distribution of interest $\mathcal{D}$ with domain $\mathcal{X}$ and target domain y, there is an underlying function $g : \mathcal{X} \to y$.
- The goal is then to learn the underlying function g with the support of a given set of training data $\{(\mathrm{x}_i, y_i)_{i \in \mathrm{I}}\} \subset \mathcal{D}'$ with $\mathcal{D}'$ as the distribution that training data draws from.
- The difference between interpolation task and extrapolation task then lies in the following different conditions:
  - Interpolation: $\mathcal{D} = \mathcal{D}'$
  - Extrapolation: $\mathcal{D} \supset \mathcal{D}'$

Related Works

- How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks (Xu et al., ICLR 2021)
  - Provides theoretical evidence to explain why and how GNNs are successful in extrapolating algorithmic tasks.
    - If we encode appropriate non-linearities in the architecture and input representation.
  - Empirically studies the extrapolation ability of GNNs on three tasks across different GNN architectures.
    - Max degree
    - Shortest path
    - n-body problem

Our Project

- We specifically investigate the extrapolation ability of GNNs across different architectures.
- Extend the empirical analysis of Xu et al. and introduce new extrapolation tasks for evaluation.
- Ablation studies and our analysis on the extrapolation ability of GNNs.

Experiment Design and Results

Max degree and shortest path (Xu el al. 2021)

- Max degree: finding the maximum degree on a graph
  - Input graph $G = (V, E)$
  - Output label $y = g(G) = \max_{u \in G} \sum_{v \in N(u)} 1$
- Extrapolation on number of nodes
- Dataset sampled as randomly generated graphs
  - Training and validation
    - |V| sampled uniformly from [20 … 30]
  - Test
    - |V| sample uniformly from [50 … 100]
    - (Interpolation) |V| sampled uniformly from [20 … 30]

Max degree -- Results

Extrapolate / Interpolate (MAPE)		Neighborhood pooling (AGGREGATE)		
		Max-pooling	Mean	Sum
Graph pooling (READOUT)	Max-pooling	51.9 / 12.5	18.0 / 13.7	1.1 / 0.2
	Mean	15.1 / 7.7	50.0 / 47.8	12.7 / 8.7
	Sum	113.4 / 10.7	58.1 / 47.7	159.9 / 10.6

The mean absolute percentage error (MAPE) loss across different AGGREGATE functions and READOUT functions for interpolation and extrapolation.
Experiment settings included in (Xu el al. 2021) is highlighted.

N-body - Task Description

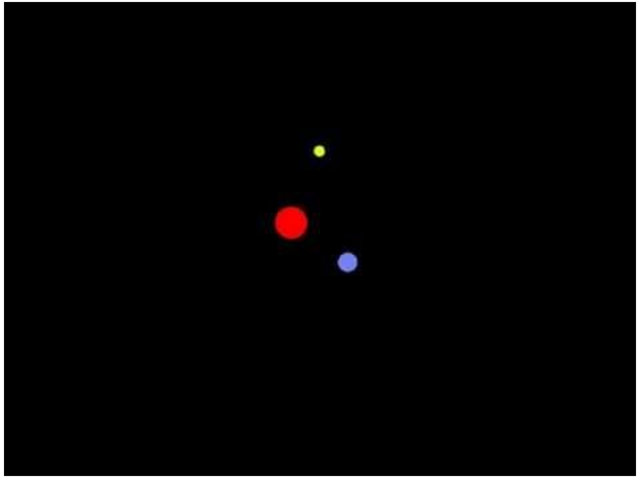
- N bodies with gravitational forces in between (distance and mass dependent)
- The overall force a star X_i receives from other stars

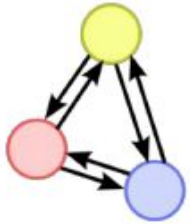
$$\boldsymbol{F}_i^t = G \sum_{j \neq i} \frac{m_i \times m_j}{\left\| \boldsymbol{x}_i^t - \boldsymbol{x}_j^t \right\|_2^3} \cdot (\boldsymbol{x}_j^t - \boldsymbol{x}_i^t)$$

- The acceleration of X_i is $\boldsymbol{a}_i^t = \boldsymbol{F}_i^t / m_i$
- The velocity of X_i at the next time frame is

$$\boldsymbol{v}_i^{t+1} = \boldsymbol{v}_i^t + \boldsymbol{a}_i^t \cdot dt$$

- In our task, given m_i, x_i^t, v_v^t, we want to predict v_i^{t+1}





n(n-1) relations

Gravitational forces

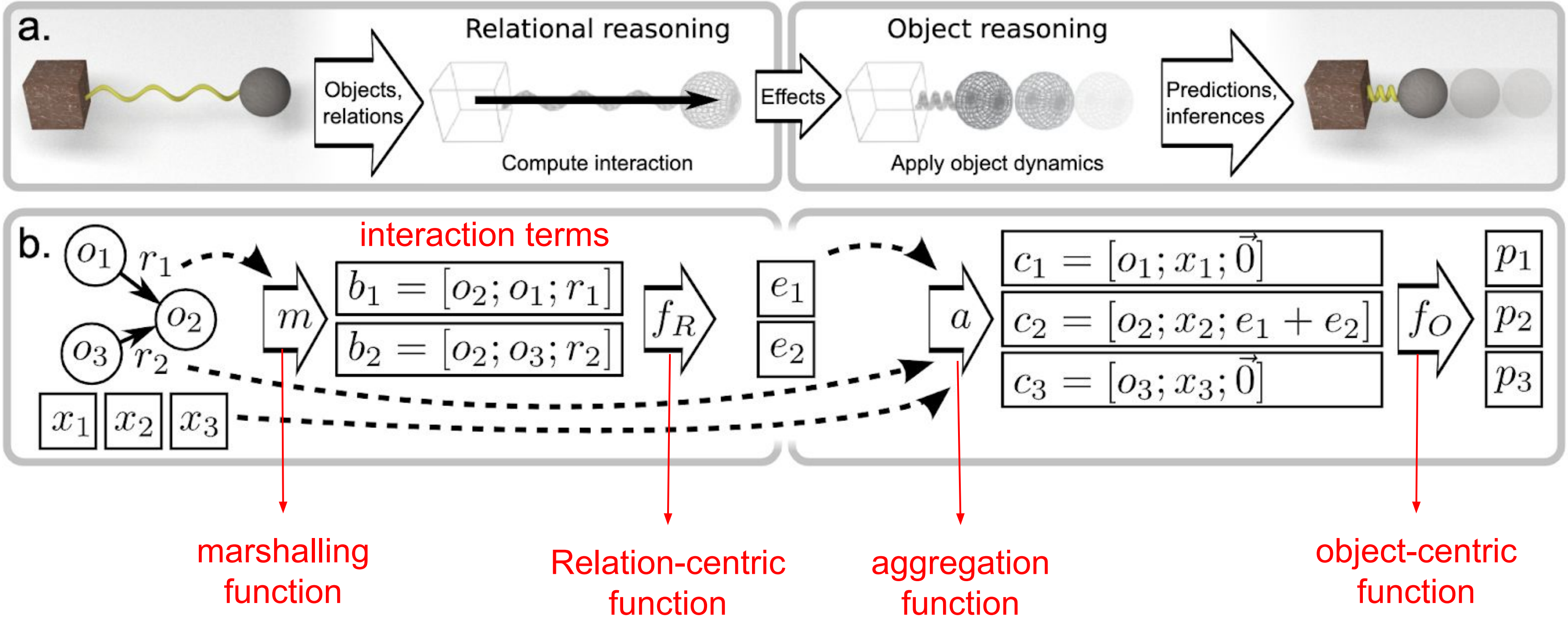
N-body - Parameters

Mass of center star	Masses of other stars	Number of starts	Initial position	Velocity
100kg	[0.02, 9.0]kg	3	Center: (0, 0) Others: samples from all angle, with a distance in [10.0, 100.0]m	Center: **0** Others: perpendicular to the gravitational force between the center star and itself

N-body - Extrapolation Schemes

- The distances between stars are out-of-distribution for test set
  - sample the distance of the test set to ensure all pairwise distances of stars in a frame are from [1..20]m, but distance of the training set is greater than 30m
- The masses of stars are out-of-distribution for test set
  - The mass of the center star is 200kg for testing, 100kg for training
  - The masses of others stars are sampled from [0.04, 18]kg for testing and [0.02, 9]kg for training

N-body - Model Setting



N-body - Input Representation

- We consider two representations of edge/relations
  - $w_{ij} = 0$
  - $w_{ij} = \frac{m_j}{\left\|x_i^t - x_j^t\right\|_2^3} \cdot (x_j^t - x_i^t)$

Can help the MLP
only learn linear

$$\boldsymbol{F}_i^t = G \sum_{j \neq i} \frac{m_i \times m_j}{\left\|\boldsymbol{x}_i^t - \boldsymbol{x}_j^t\right\|_2^3} \cdot (\boldsymbol{x}_j^t - \boldsymbol{x}_i^t)$$

$$\boldsymbol{a}_i^t = \boldsymbol{F}_i^t / m_i$$

$$\boldsymbol{v}_i^{t+1} = \boldsymbol{v}_i^t + \boldsymbol{a}_i^t \cdot dt$$

N-body - Experimental Results

- MAPE: Mean Absolute Percentage Error

Input edge representation	Extrapolate distance	Extrapolate mass	Interpolate
0	8.776	6.632	1.938
w_{ij}	3.448	3.600	1.274

HINT: Handwritten Arithmetic with Integers

HINT: Task Definition

Input: handwritten expressions	**Output**: results
$2 \times 5 \div 9$	2
$5 \times 5 + (9 - 0 - 2)$	32
$4 \times (3 + 9) - 7 - (0 - 5)$	41
$3 \times (7 \times 2) + (8 + 4) + 4 \times 3$	66

HINT: Train and Evaluation

$$
\begin{aligned}
&D_{train} \subset \mathcal{D}_{train} = \{(x, y) : |x| \leqslant 10, \max(v) \leqslant 100\}, \\
&D_{test} = D_{test}^{(1)} \cup D_{test}^{(2)} \cup D_{test}^{(3)} \cup D_{test}^{(4)} \cup D_{test}^{(5)}, \text{where} \\
&D_{test}^{(1)} = D_{train}, \\
&D_{test}^{(2)} \subset \mathcal{D}_{train} \backslash D_{train}, \\
&D_{test}^{(3)} \subset \{(x, y) : |x| \leqslant 10, \max(v) > 100\}, \\
&D_{test}^{(4)} \subset \{(x, y) : |x| > 10, \max(v) \leqslant 100\}, \\
&D_{test}^{(5)} \subset \{(x, y) : |x| > 10, \max(v) > 100\}.
\end{aligned}
$$

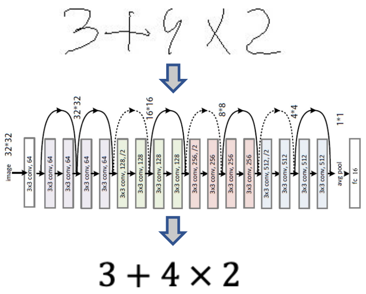
x: the handwritten expression
$|x|$: the number of operators
y: the final result
v: all the intermediate values

HINT: Train and Evaluation

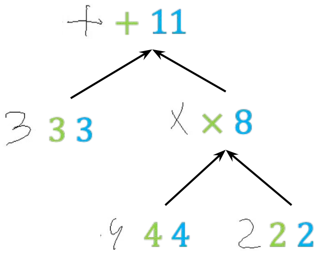
Train		$2\times5\div9$ 2 $(9-9)\times(3-4)-1\times(0+3-(6-(2-2\div2)))$ 0 $5\times5+(9-0-2)$ 32 $4\times(3+9)-7-(0-5)$ 41
Test	**1**	$1\div4$ 1 $1\times(2\div5)\times(8-8-6)$ 0 $6-4+(0-(6+0\div(4\div(6/1)\times1)))+(9+4)$ 15
	2	$1+3\div4$ 2 $3\times(7\times2)+(8+4)+4\times3$ 66 $4+(0-(7+7+6))\times4-0$ 4
	3	$3\times(8\times(8\times1)+0\div9)$ 192 $5\times(3\div1\times9)+(2-5)\times(7\times(6+5))$ 135 $2\times(3\times(3\div6+6\times(3\times4\times6\div(1\times6))))+0\div3)$ 438
	4	$(6\times5-0)\div((4+9+5)\div9)+(3-((2-(2+(9\times7-8\div9))))/4-9))$ 18 $6-3\div(9\times(9\div(4-(4-7))))+(1+1/(6-2))\div(7\times2+6\div8)$ 6 $(7+3)/(6-6\times(0\times(6\div1)))-(3\times1-6-4/(4-3))\times(9\times3)$ 2
	5	$(6+7\times1+2\div4+(1+4-0\div3)\times8-(1+3\times8))\times((0+(2\times9-0)/3)\div(8\div9))$ 174 $(3+(8+(4-7\times(7+8))\times(8\div4-(4-(6+5)+6))))\div(7+5\times1\times0)\div5$ 1 $7\times(8\div(1\times(7\div7))+(1+2))\times10+9-5\div(8+4\div(9\times6)))+(8-(9-8+3))$ 620

Method: Graph Neural Network

- An expression can be seen as a graph $G = (V, E)$
- Node features V :
  - Image input: ResNet-18
  - Symbol input: embedding
- Edges E :
  - Dependency relations generated by a parser (shunting yard algorithm)



ID	Stack Buffer	Transition	Dependency
0	3 + 4 × 2	Shift	
1	3 + 4 × 2	Shift	
2	3 + 4 × 2	Left-Arc	3 ← +
3	+ 4 × 2	Shift	
4	+ 4 × 2	Shift	
5	+ 4 × 2	Left-Arc	4 ←×
6	+ × 2	Shift	
7	+ × 2	Right-Arc	×→ 2
8	+ ×	Right-Arc	+→×



Results

Table 1: The performance comparison of GNN and LSTM

Input	Model	Test Accuracy (%)					
		Overall	**1**	**2**	**3**	**4**	**5**
Symbol	GNN	49.71	97.05	63.67	11.58	52.41	12.57
(Embedding)	LSTM	34.58	98.31	29.79	2.91	26.39	2.76
Image	GNN	39.39	87.02	46.17	6.51	40.44	6.47
(ResNet-18)	LSTM	32.95	87.31	30.74	2.67	31.17	2.55

Conclusion

- Generalization/Extrapolation can span over inputs, graph structure, and outputs.
- Encoding appropriate non-linearity in graph neural networks can help extrapolation over graph structure (size).
- Encoding appropriate non-linearity in the input representation can help extrapolation over unseen inputs.
- How to improve extrapolation over outputs is still unknown.