

An Introduction to Neural Algorithmic Reasoning

报告人：刘佳玮

北京邮电大学博士生

本次分享大纲

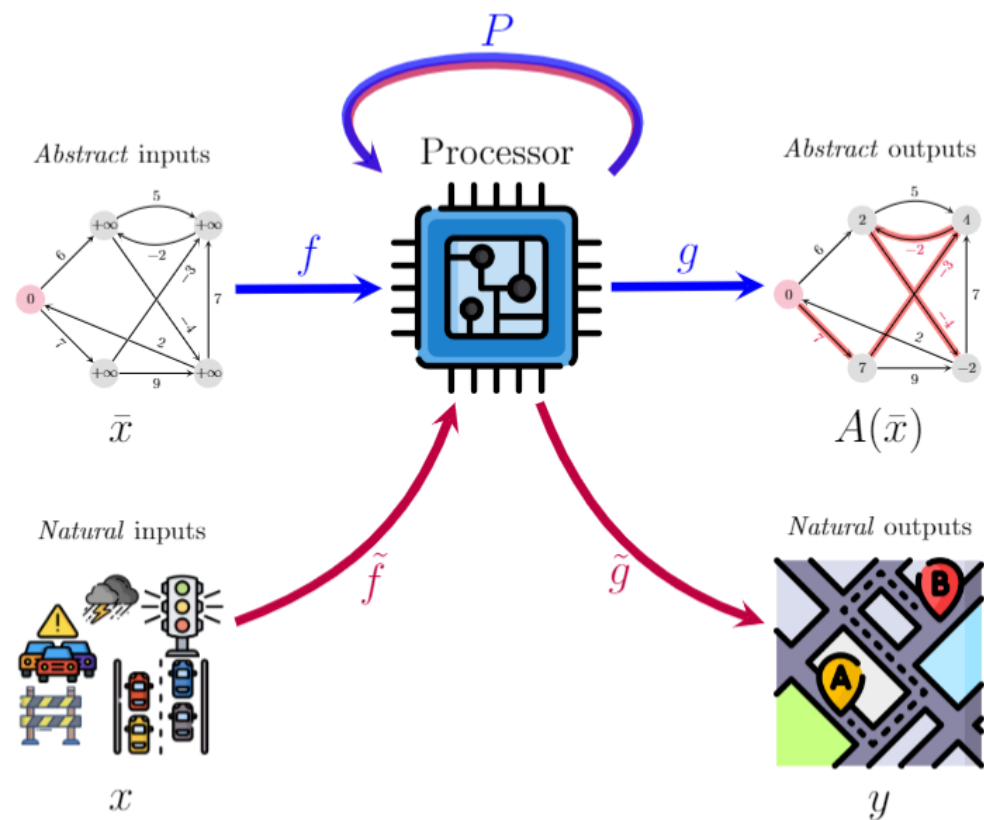
- 动机
 - 为什么要做神经算法推理？
- 核心思想
 - 什么是神经算法推理？
 - 什么样的神经网络适合做神经算法推理？
- 代表性工作
 - *Neural execution of graph algorithms. ICLR 2020.*
 - *The CLRS algorithmic reasoning benchmark. ICML 2022.*
 - *A generalist neural algorithmic learner. LoG 2022.*
- 总结与展望

动机

为什么要做神经算法推理？

- 算法很容易推广到任意大小的输入，并且可以通过可解释的逐步操作进行验证或证明是正确的。它们的缺点是输入必须符合特定的算法规范，并且查看单独的任务通常需要提出全新的算法。 [2]
- 神经网络更加灵活，可以适应和概括原始输入，自动提取适当的特征，但神经网络依赖于大量的训练数据，而且泛化性能较差。 [2]

神经模型是否可以继承一些积极的算法属性，从而在解决具有挑战性的现实世界问题（数据高维、嘈杂、多变）时更具有泛化性？



神经算法推理示意图 [1]

[1] Veličković P, Blundell C. **Neural algorithmic reasoning**[J]. *Patterns*, 2021, 2(7): 100273.

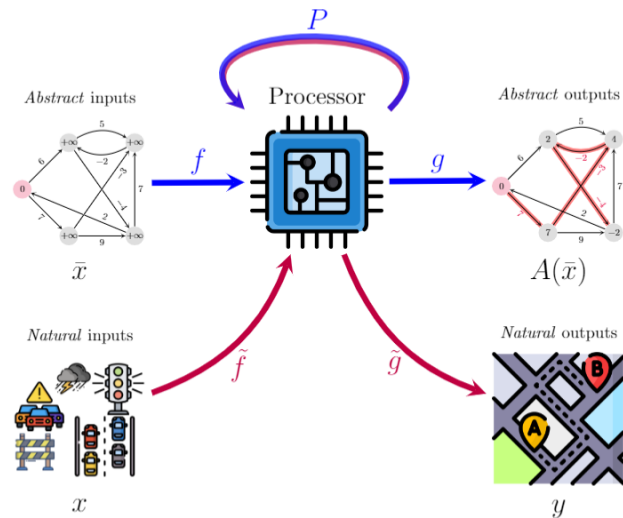
[2] Veličković P, Badia A P, Budden D, et al. **The CLRS algorithmic reasoning benchmark**[C]//ICML. PMLR, 2022: 22084-22102.

本次分享大纲

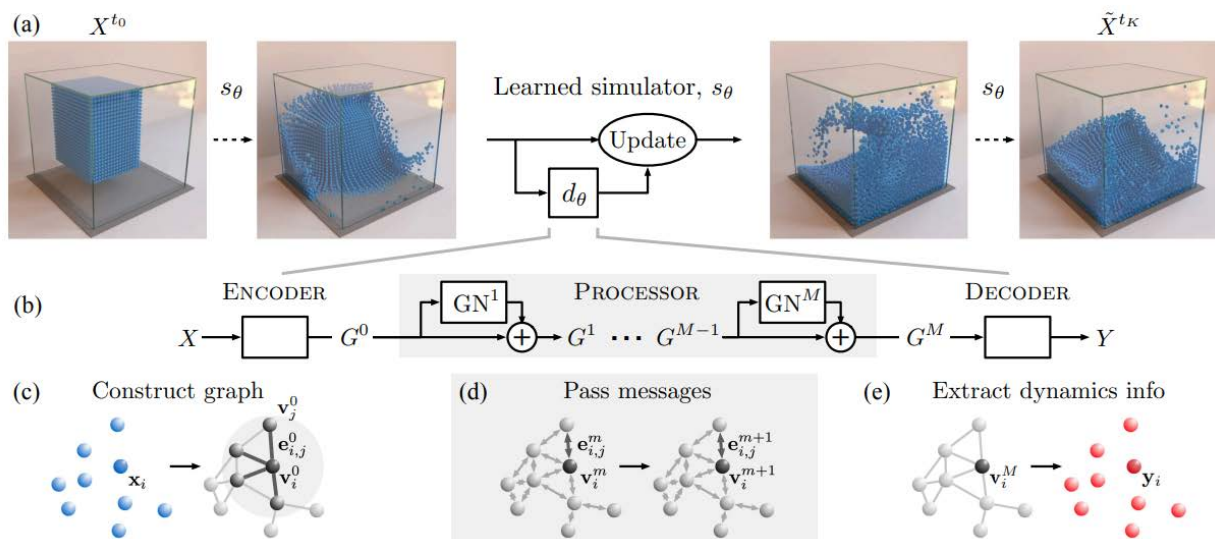
- 动机
 - 为什么要做神经算法推理？
- 核心思想
 - 什么是神经算法推理？
 - 什么样的神经网络适合做神经算法推理？
- 代表性工作
 - *Neural execution of graph algorithms. ICLR 2020.*
 - *The CLRS algorithmic reasoning benchmark. ICML 2022.*
 - *A generalist neural algorithmic learner. LoG 2022.*
- 总结与展望

核心思想

- 什么是神经算法推理？
 - 通过学习执行经典算法来训练神经网络来展示一定程度的算法推理。 [1]
- 神经算法推理的基本步骤 [1]
 - 学习处理器的参数：为算法A设计一个编码器f，处理器P和解码器g，训练P让 $g(P(f(\bar{x}))) \approx A(\bar{x})$
 - 学习编解码器的参数：设计一个神经网络编码器 \tilde{f} 和 \tilde{g} ，并替换f和g，冻结处理器P的参数，让 $g(P(f(\bar{x}))) \approx y$ ，y是真实标签。



神经算法推理示意图 [1]



神经算法推理示意图 [2]

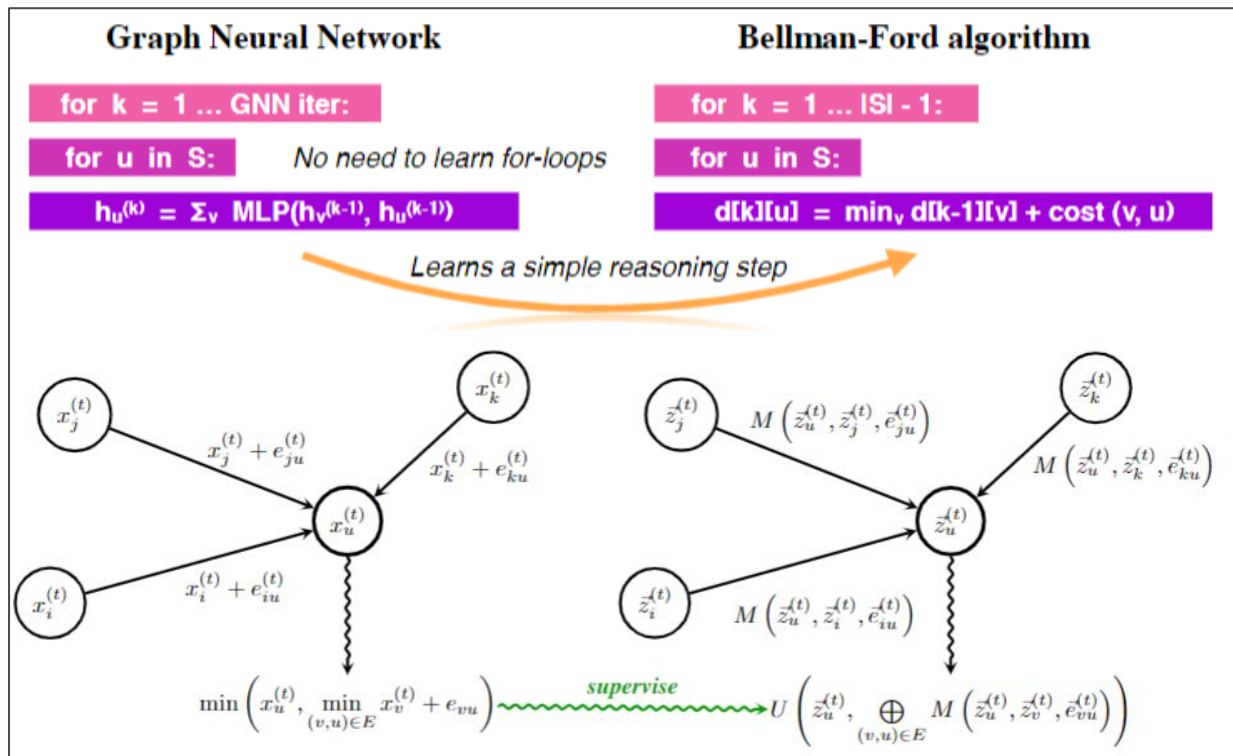
[1] Veličković P, Blundell C. **Neural algorithmic reasoning**[J]. Patterns, 2021, 2(7): 100273.

[2] Sanchez-Gonzalez A, Godwin J, Pfaff T, et al. **Learning to simulate complex physics with graph networks**[C]//ICML.2020

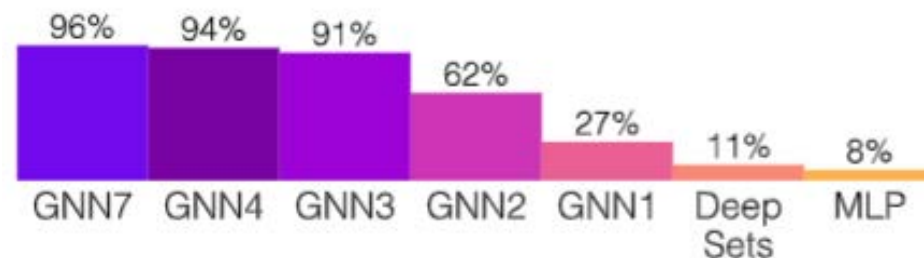
核心思想

什么样的神经网络适合做神经算法推理？

- 算法对齐：学习模型和目标算法之间必须有一个“对齐”，便于优化。
 - 针对不同任务，应该选择和此任务解决算法保持一致的神经网络作为处理器P，使神经网络更好的学习此任务。
- 观察：GNN在和DP(动态规划)问题的算法有很高的相似性，因此能够有效地学习这些任务。
 - 举例：对于Bellman-Ford算法，GNN中只需要MLP去学习一个简单的推理步骤，二者的计算结构就基本相似。
- 实验：对于动态规划任务，只有GNN在足够多的迭代次数下能泛化。[1]



Bellman-Ford算法和GNN的关系[1][2]



GNN与其他模型在动态规划任务上的表现

[1]Keyulu Xu, Jingling Li, Mozhi Zhang, et al. **What can neural networks reason about?** In ICLR 2020.
[2]Petar Velickovic, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. **Neural execution of graph algorithms.** In ICLR 2020.

本次分享大纲

- 动机
 - 为什么要做神经算法推理？
- 核心思想
 - 什么是神经算法推理？
 - 什么样的神经网络适合做神经算法推理？
- 代表性工作
 - *Neural execution of graph algorithms. ICLR 2020.*
 - *The CLRS algorithmic reasoning benchmark. ICML 2022.*
 - *A generalist neural algorithmic learner. LoG 2022.*
- 总结与展望

基础信息

- 作者基础信息

- Petar Veličković (Deepmind)

- 主要研究兴趣

- Geometric Deep Learning
- Graph Representation Learning
- Graph Neural Networks
- Algorithmic Reasoning
- Computational Biology



Petar Veličković

关注

Staff Research Scientist, Google DeepMind | Affiliated Lecturer, University of Cambridge
在 deepmind.com 的电子邮件经过验证 - 首页

[Geometric Deep Learning](#) [Graph Representation Learning](#) [Graph Neural Networks](#)
[Algorithmic Reasoning](#) [Computational Biology](#)

标题	引用次数	年份
Graph Attention Networks P Veličković, G Cucurull, A Casanova, A Romero, P Liò, Y Bengio 6th International Conference on Learning Representations (ICLR 2018)	13821*	2018
Deep Graph Infomax P Veličković, W Fedus, WL Hamilton, P Liò, Y Bengio, RD Hjelm 7th International Conference on Learning Representations (ICLR 2019)	1545*	2019
Geometric deep learning: Grids, groups, graphs, geodesics, and gauges MM Bronstein, J Bruna, T Cohen, P Veličković arXiv preprint arXiv:2104.13478	671	2021
Principal neighbourhood aggregation for graph nets G Corso*, L Cavalleri*, D Beaini, P Liò, P Veličković Advances in Neural Information Processing Systems 33	371	2020
Towards Sparse Hierarchical Graph Classifiers C Cangea*, P Veličković*, N Jovanović, T Kipf, P Liò arXiv preprint arXiv:1811.01287	221*	2018

代表性工作

- 动机：能否用 GNN 架构来模仿经典图算法的各个步骤？
- 核心思想：将算法操作作为操作和查询节点的一部分，使用图神经网络来学习如何执行这些操作。

- P是GNN, f_A, g_A, T_A 都是线性变换

- 主要结论：

- 可以使用图神经网络通过学习算法的中间步骤来逐步得到最终解。
- 多任务学习有优势，因为许多经典算法共享相关的子程序。

GNN

encoder $\vec{z}_i^{(t)} = f_A(\vec{x}_i^{(t)}, \vec{h}_i^{(t-1)})$

processor $\mathbf{H}^{(t)} = P(\mathbf{Z}^{(t)}, \mathbf{E}^{(t)})$

decoder $\vec{y}_i^{(t)} = g_A(\vec{z}_i^{(t)}, \vec{h}_i^{(t)})$

termination $\tau^{(t)} = \sigma(T_A(\mathbf{H}^{(t)}, \overline{\mathbf{H}^{(t)}}))$

Bellman-Ford

$$x_i^{(1)} = \begin{cases} 0 & i = s \\ +\infty & i \neq s \end{cases}$$

$$x_i^{(t+1)} = \min \left(\vec{x}_i^{(t)}, \min_{(j,i) \in E} x_j^{(t)} + e_{ji}^{(t)} \right)$$

$$p_i^{(t)} = \begin{cases} i & i = s \\ \operatorname{argmin}_{j:(j,i) \in E} x_j^{(t)} + e_{ji}^{(t)} & i \neq s \end{cases}$$

$$\vec{y}_i^{(t)} = p_i^{(t)} \parallel x_i^{(t+1)}$$

Model	Predecessor (mean step accuracy / last-step accuracy)		
	20 nodes	50 nodes	100 nodes
LSTM (Hochreiter & Schmidhuber, 1997)	47.20% / 47.04%	36.34% / 35.24%	27.59% / 27.31%
GAT* (Veličković et al., 2018)	64.77% / 60.37%	52.20% / 49.71%	47.23% / 44.90%
GAT-full* (Vaswani et al., 2017)	67.31% / 63.99%	50.54% / 48.51%	43.12% / 41.80%
MPNN-mean (Gilmer et al., 2017)	93.83% / 93.20%	58.60% / 58.02%	44.24% / 43.93%
MPNN-sum (Gilmer et al., 2017)	82.46% / 80.49%	54.78% / 52.06%	37.97% / 37.32%
MPNN-max (Gilmer et al., 2017)	97.13% / 96.84%	94.71% / 93.88%	90.91% / 88.79%
MPNN-max (curriculum)	95.88% / 95.54%	91.00% / 88.74%	84.18% / 83.16%
MPNN-max (no-reach)	82.40% / 78.29%	78.79% / 77.53%	81.04% / 81.06%
MPNN-max (no-algo)	78.97% / 95.56%	83.82% / 85.87%	79.77% / 78.84%

Petar Veličković, Rex Ying, Matilde Padovano, Raia Hadsell, and Charles Blundell. *Neural execution of graph algorithms*. In ICLR 2020.

代表性工作

- 动机：能否对现有的神经算法推理器进行统一评估？
- 核心思想：提出CLRS基准测试，支持30个算法。
- 主要结论：
 - GNN 模型优于 Deep Sets 和 Memory Nets 等模型。PGN 整体效果最好，但在需要远程展开（例如 DFS）或递归推理（如 Quicksort）的任务上表现困难。
 - 可能需要更专业的归纳偏差和训练方案来处理字符串匹配算法。

Table 1. Average test micro-F₁ score of all models on all algorithm classes. The full test results for all 30 algorithms, along with a breakdown of the “win/tie/loss” metric, are given in Appendix C.

Algorithm	Deep Sets	GAT	Memnet	MPNN	PGN
Divide & Conquer	12.48% ± 0.67	24.43% ± 0.74	13.05% ± 0.00	20.30% ± 0.85	65.23% ± 4.44
Dynamic Prog.	66.05% ± 7.79	67.19% ± 5.33	67.94% ± 7.75	65.10% ± 6.44	70.58% ± 6.48
Geometry	64.08% ± 6.60	73.27% ± 11.18	45.14% ± 11.65	73.11% ± 17.19	61.19% ± 7.01
Graphs	37.65% ± 8.09	46.80% ± 8.66	24.12% ± 5.20	62.79% ± 8.75	60.25% ± 8.42
Greedy	75.47% ± 6.81	78.96% ± 4.59	53.42% ± 20.73	82.39% ± 3.01	75.84% ± 6.59
Search	43.79% ± 18.29	37.35% ± 19.81	34.35% ± 21.67	41.20% ± 19.87	56.11% ± 21.56
Sorting	39.60% ± 7.19	14.35% ± 4.64	71.53% ± 1.09	11.83% ± 2.78	15.45% ± 8.46
Strings	2.64% ± 0.68	3.02% ± 1.08	1.51% ± 0.21	3.21% ± 0.94	2.04% ± 0.20
Overall average	42.72%	43.17%	38.88%	44.99%	50.84%
Win/Tie/Loss counts	0/3/27	1/5/24	4/2/24	8/3/19	8/6/16

实验设置：

- 算法：从CLRS教科书中的算法中筛选出30个算法，包括排序、搜索、分治、贪心、动态规划、图算法、字符串算法和几何算法。
- 模型：由处理器、编码器和解码器组成。CLRS提供了以下GNN基线处理器：Deep Sets, Memory Networks, GAT, GATv2, MPNN, PGN。

Veličković P, Badiá A P, Budden D, et al. **The CLRS algorithmic reasoning benchmark**[C]//ICML. PMLR, 2022: 22084-22102.

代表性工作

- 动机：能否设计一个通用的神经算法推理器？
- 核心思想：
 - 处理器：门控机制、三元组推理
 - 编码器、解码器改进：随机化位置标量、置换解码器和Sinkhorn算子
 - 数据集和训练策略改进：去除教师强制、增强训练数据、软提示传播、静态提示消除、通过编码器初始化和梯度裁剪提高训练稳定性
- 主要结论：
 - 可以以多任务的方式学习不同的算法，但需要注意(G)NN的学习动态和稳定性。
 - 如果修改（对 GNN 架构、数据管道或损失函数）在通用性的右侧进行，则可以一次提高大组算法的算法执行性能。

门控机制

$$\mathbf{g}_i^{(t)} = f_g \left(\mathbf{z}_i^{(t)}, \mathbf{m}_i^{(t)} \right)$$
$$\hat{\mathbf{h}}_i^{(t)} = \mathbf{g}_i^{(t)} \odot \mathbf{h}_i^{(t)} + (1 - \mathbf{g}_i^{(t)}) \odot \mathbf{h}_i^{(t-1)}$$
$$\mathbf{z}^{(t)} = \mathbf{x}_i^{(t)} \parallel \hat{\mathbf{h}}_i^{(t-1)}$$

三元组推理

$$t_{ijk} = \psi_t(\mathbf{h}_i, \mathbf{h}_j, \mathbf{h}_k, \mathbf{e}_{ij}, \mathbf{e}_{ik}, \mathbf{e}_{kj}, \mathbf{g}) \quad \mathbf{h}_{ij} = \phi_t(\max_k t_{ijk})$$

Alg. Type	Memnet [5]	MPNN [5]	PGN [5]	Triplet-GMPNN (ours)
Div. & C.	13.05% ± 0.14	20.30% ± 0.85	65.23% ± 4.44	76.36% ± 1.34
DP	67.94% ± 8.20	65.10% ± 6.44	70.58% ± 6.48	81.99% ± 4.98
Geometry	45.14% ± 11.95	73.11% ± 17.19	61.19% ± 7.01	94.09% ± 2.30
Graphs	24.12% ± 5.30	62.79% ± 8.75	60.25% ± 8.42	81.41% ± 6.21
Greedy	53.42% ± 20.82	82.39% ± 3.01	75.84% ± 6.59	91.21% ± 2.95
Search	34.35% ± 21.67	41.20% ± 19.87	56.11% ± 21.56	58.61% ± 24.34
Sorting	71.53% ± 1.41	11.83% ± 2.78	15.45% ± 8.46	60.37% ± 12.16
Strings	1.51% ± 0.46	3.21% ± 0.94	2.04% ± 0.20	49.09% ± 23.49
Overall avg.	38.88%	44.99%	50.84%	74.14%
> 90%	0/30	6/30	3/30	11/30
> 80%	3/30	9/30	7/30	17/30
> 60%	10/30	14/30	15/30	24/30

Ibarz B, Kurin V, Papamakarios G, et al. **A generalist neural algorithmic learner**[C]//Learning on Graphs Conference. PMLR, 2022: 2: 1-2: 23.

本次分享大纲

- 动机
 - 为什么要做神经算法推理？
- 核心思想
 - 什么是神经算法推理？
 - 什么样的神经网络适合做神经算法推理？
- 代表性工作
 - *Neural execution of graph algorithms. ICLR 2020.*
 - *The CLRS algorithmic reasoning benchmark. ICML 2022.*
 - *A generalist neural algorithmic learner. LoG 2022.*
- 总结与展望

总结与展望

- 本次分享基于范长俊老师推荐的论文列表，为大家回顾了神经算法推理的动机、核心思想和代表性工作。潜在的未来研究方向如下：
 - 如何提高神经算法推理器的对齐能力、泛化能力？
 - 改进图神经网络[1]
 - 改进对齐过程：考虑问题的对偶性[2]、因果性[3]
 - 能否将神经算法推理与图预训练模型结合？
 - 基于知识蒸馏[4]把预训练模型的知识注入到神经算法推理器
 - 基于工具学习[5]让预训练模型与外部推理器交互

[1] Dudzik A J, Veličković P. **Graph neural networks are dynamic programmers**[J]. *Advances in Neural Information Processing Systems*, 2022, 35: 20635-20647.

[2] Numeroso D, Bacciu D, Veličković P. **Dual Algorithmic Reasoning**[J] ICLR 2023.

[3] Bevilacqua B, Nikiforou K, Ibarz B, et al. **Neural Algorithmic Reasoning with Causal Regularisation**[J]. *arXiv preprint arXiv:2302.10258*, 2023.

[4] Yang Cheng, Jiawei Liu, and Chuan Shi. **Extract the knowledge of graph neural networks and go beyond it: An effective knowledge distillation framework**. WWW 2021.

[5] Zhang J. **Graph-ToolFormer: To Empower LLMs with Graph Reasoning Ability via Prompt Augmented by ChatGPT**[J]. *arXiv preprint arXiv:2304.11116*, 2023.