Giving a Voice to Your Graph: Representing Structured Data for LLMs

Bryan Perozzi
Google Research

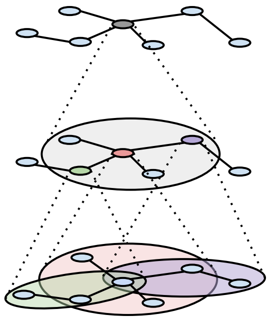
6/17/24
Workshop on Scene Graphs and Graph Representation Learning (CVPR)

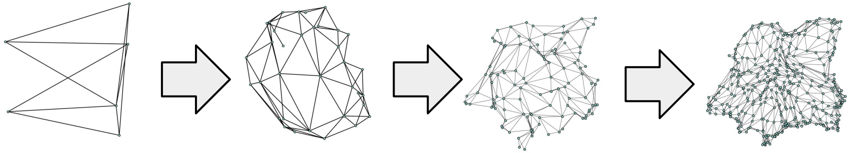
GNN Team

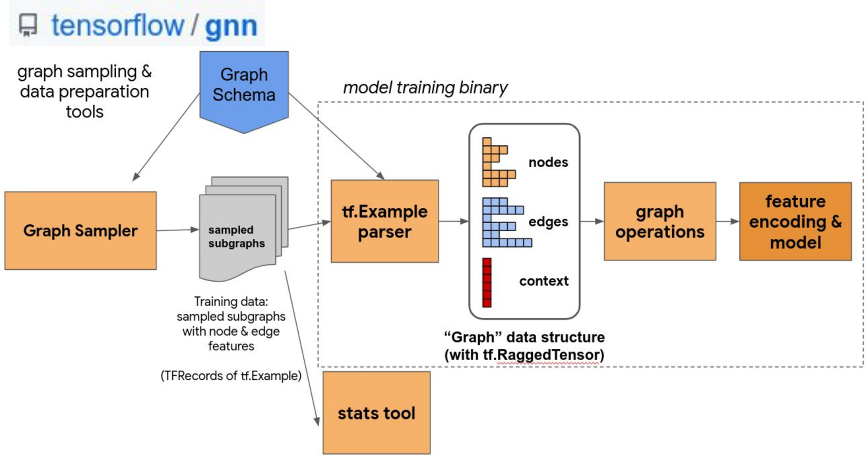
The GNN Team's mission is to make it easy to use graph structured data inside neural networks.

In service of this goal, we produce:

- **Research** in the area (ICML, NeurIPS, KDD, etc)
- **Library** to express these models (TF-GNN)
- **Infra** for training & deploying GNNs
- **Models** for custom applications







Motivation

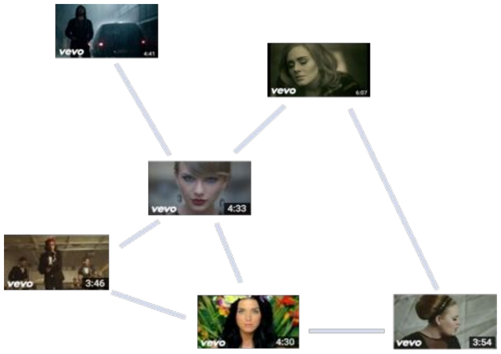
We have a powerful new tool (GenAI models), that allows us incredible new capabilities in language, vision, video, and more. However, there are weaknesses stopping us from applying this tool everywhere.

- LLMs make stuff up (***hallucinations***)
- LLMs have stale pretraining (***freshness***)
- There's a desire for *private* LLMs that can use personal data without leaking it (***privacy***)
- Training and inference is expensive (***cost***)

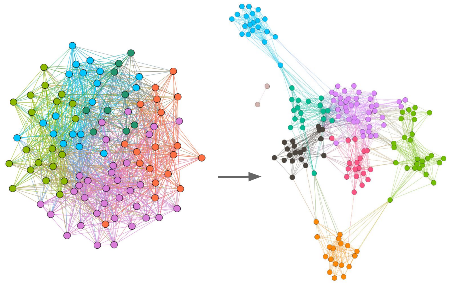
How can Graphs help?

Graphs are tools for modeling the fundamental structure of data. They allow us to

- ***Represent knowledge*** in human-accessible formats (knowledge graphs, relational databases, etc)
  - Grounding
  - Structured Data in prompt
  - Private information in prompt

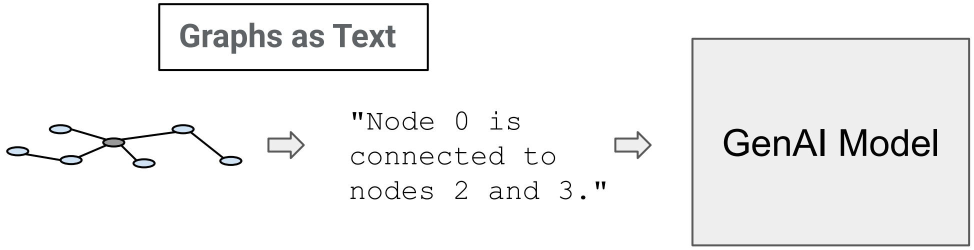


- Use sparsity for efficient algorithms and hierarchical decompositions (***efficiency***)
  - Graph models vs Transformers
    - Graphs: more parameter efficient
    - Graphs: higher sample efficiency

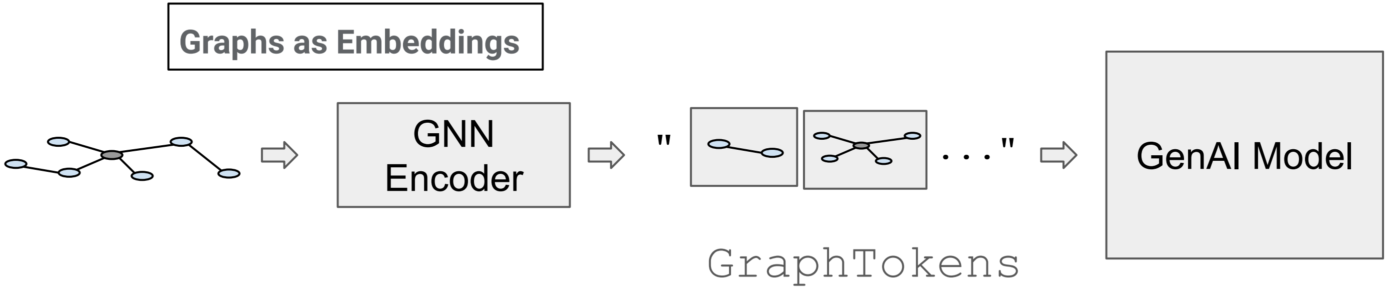


How can we best get graph structure into GenAI models?

Idea 1: Graphs as Text



Idea 2: Graphs as Embedding



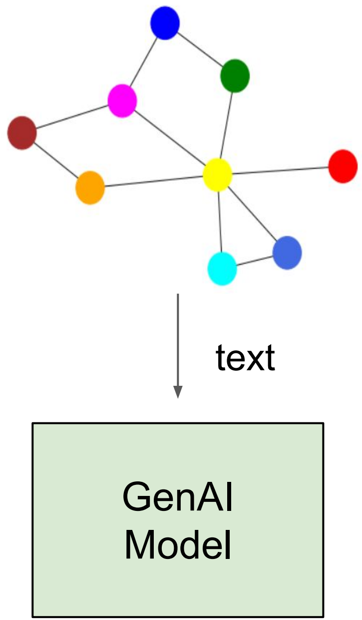
Question 1: How to best encode graphs as Text?

In this work, we seek to find the best way to encode graphs as text for "black box" use in LLMs.

Why ***text***? → Dominant LLM interface

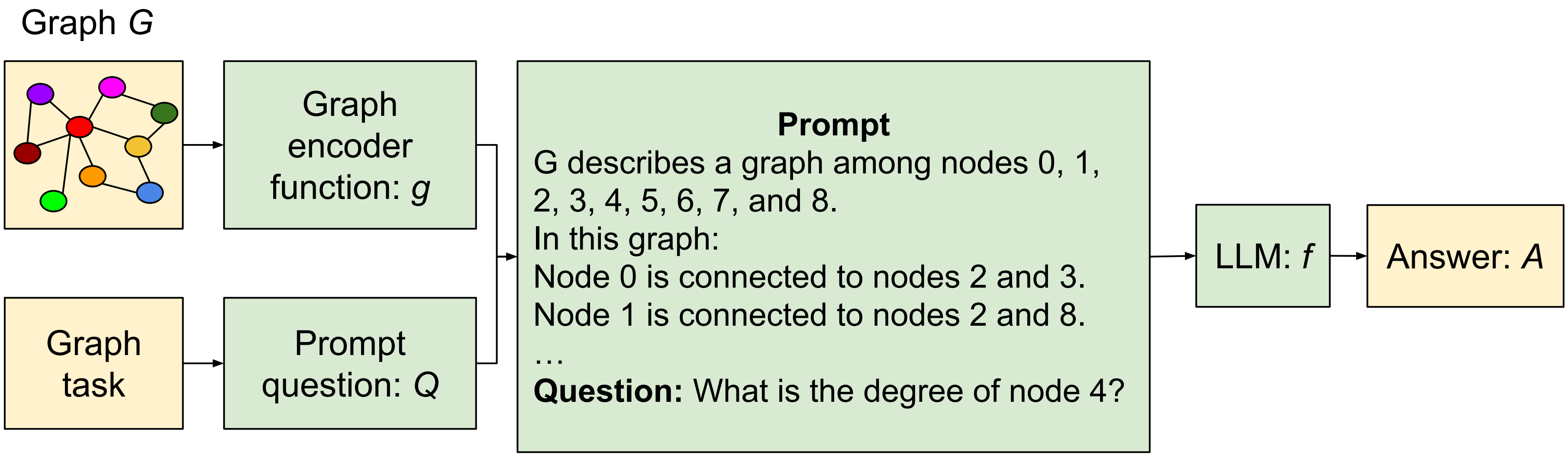
What's novel here?

- ❖ Exploration of graph prompting
- ❖ Evaluation of graph encoding
- ❖ Analysis of the effect of graph structure



TALK LIKE A GRAPH: ENCODING GRAPHS FOR LARGE LANGUAGE MODELS (ICLR'24)
https://arxiv.org/pdf/2310.04560.pdf

Overview of Framework

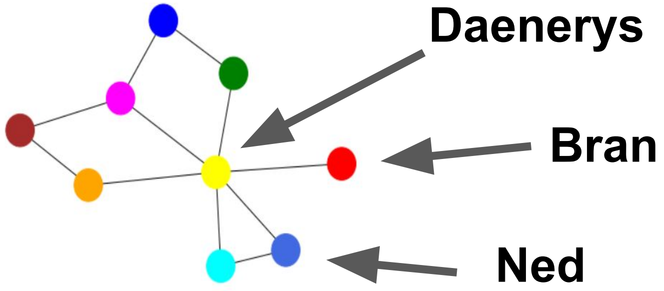


Goal: We seek to find a *graph encoder* and *prompt question* that can maximize the score of the correct answers over the training dataset.

TALK LIKE A GRAPH: ENCODING GRAPHS FOR LARGE LANGUAGE MODELS
https://arxiv.org/pdf/2310.04560.pdf

What's in a node id?

Q: **What if we used Fantasy Characters?**



GOT: G describes a friendship graph among Ned, Cat, Daenerys, Jon, Bran, Sansa, Arya, Cersei, and Jaime.
In this friendship graph: Ned and Cat are friends, Ned and Daenerys are friends, Cat and Daenerys are friends, Daenerys and Jon are friends, Daenerys and Bran are friends, Daenerys and Sansa are friends, Daenerys and Cersei are friends, Jon and Jaime are friends, Sansa and Arya are friends, Arya and Cersei are friends, Cersei and Jaime are friends.

(More encoder descriptions in paper)

Graph Encoding Choice Matters

graph encoding	ZERO-SHOT
Adjacency	4.83
Incident	6.16
Co-authorship	6.08
Friendship	5.16
SP	5.16
GOT	4.33
Social Network	4.58
Politician	**3.50**
Expert	5.16

Table 5: Ranking of graph encodings from experiment in Section 3.1 (lower better).

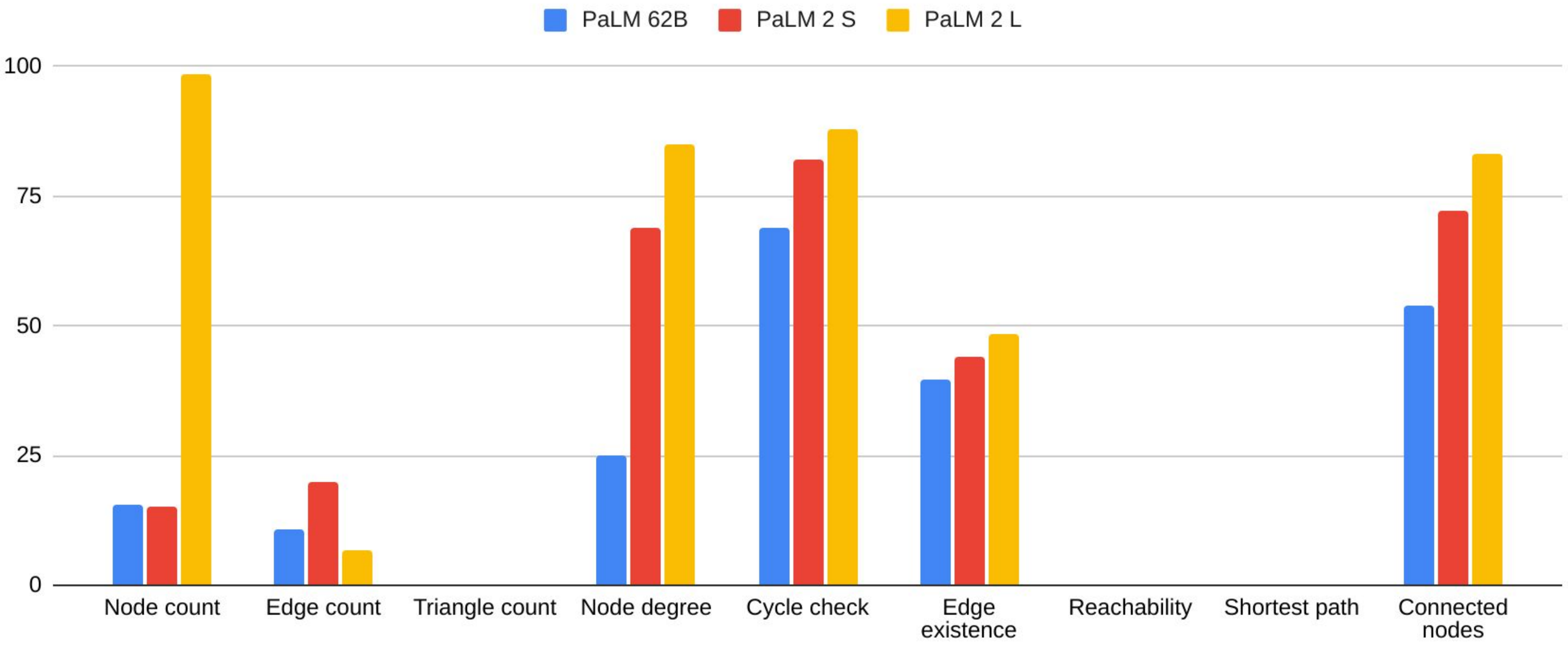
Graph Encoding Choice Matters

graph encoding	ZERO-SHOT	ZERO-COT	FEW-SHOT	COT	COT-BAG
Adjacency	4.83	3.25	2.16	3.00	1.83
Incident	6.16	**2.58**	**2.00**	**2.33**	**1.33**
Co-authorship	6.08	6.33	5.58	6.75	8.83
Friendship	5.16	6.41	6.25	4.66	6.00
SP	5.16	4.50	5.25	5.75	4.66
GOT	4.33	4.08	5.83	5.00	6.25
Social Network	4.58	6.50	5.83	6.16	6.41
Politician	**3.50**	6.33	6.25	5.58	4.00
Expert	5.16	5.00	5.83	5.75	5.66

Table 5: Ranking of graph encodings from experiment in Section 3.1 (lower better).

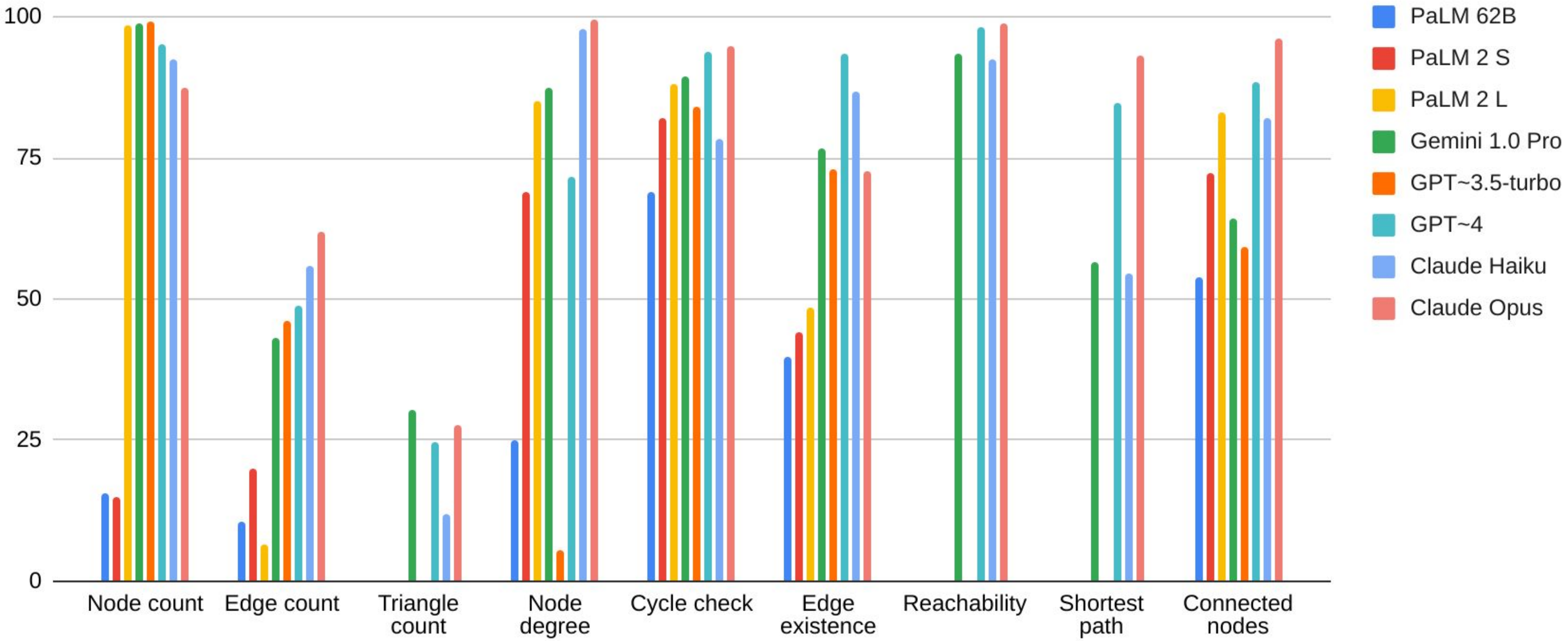
“Talk like a Graph” Leaderboard

Sept 20th, 2023



“Talk like a Graph” Leaderboard

May 8th, 2024



Question 2: How to Best Encode Data for LLMs?

In this work, we seek to find the best way to encode data for LLMs.

Why ***graphs***? → Can encode anything, e.g.

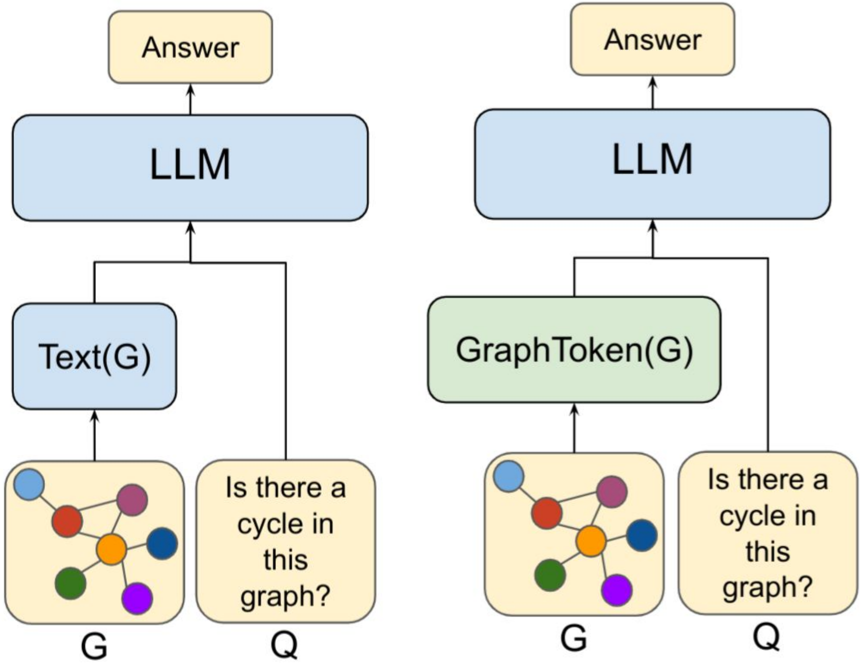
- complex interactions
- multi-scale effects
- multi-modal data

Why ***not text***? →*Talk like a Graph* showed some limitations of text encoding.

What's novel here?

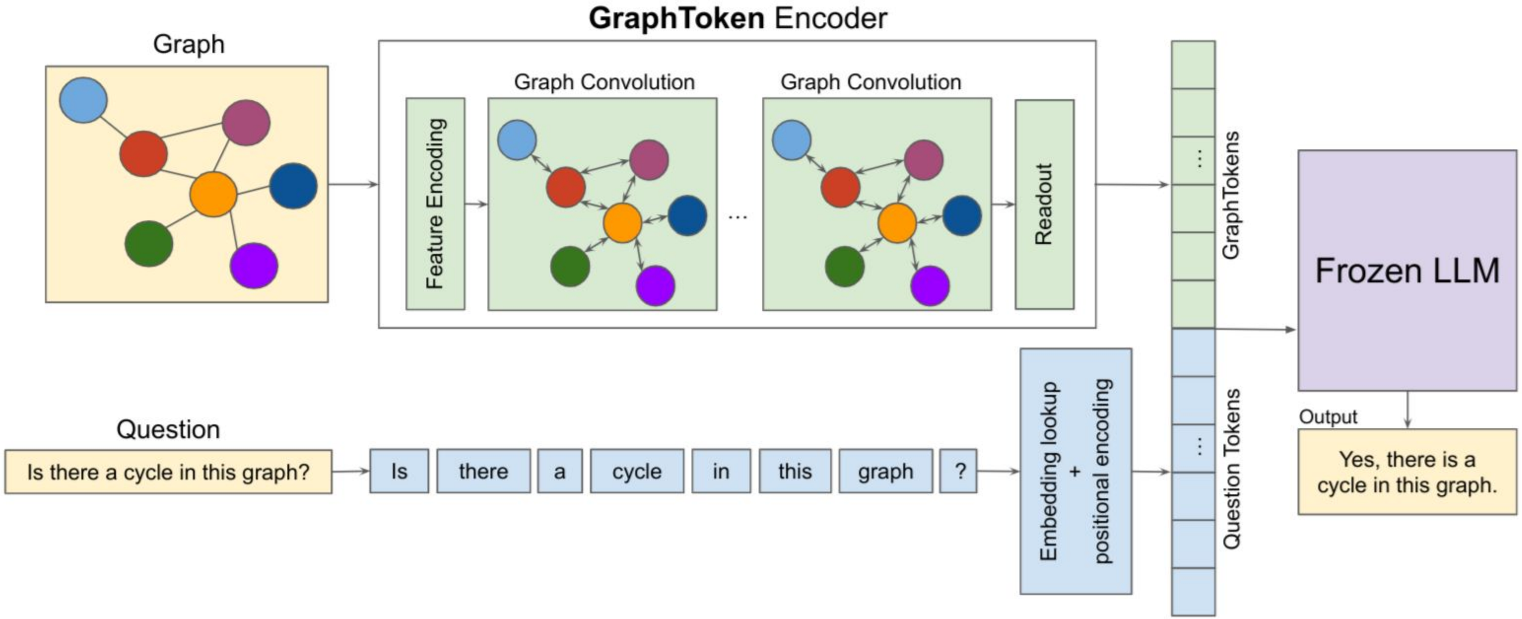
- ❖ GraphToken → Fundamentally new way of learning graph embeddings
- ❖ GNN Soft Prompting Functions
- ❖ Generalization experiments of GraphToken

“Talk like a Graph” ⇨ “GraphToken”



LET YOUR GRAPH DO THE TALKING: ENCODING STRUCTURED DATA FOR LLMS
https://arxiv.org/pdf/2402.05862.pdf

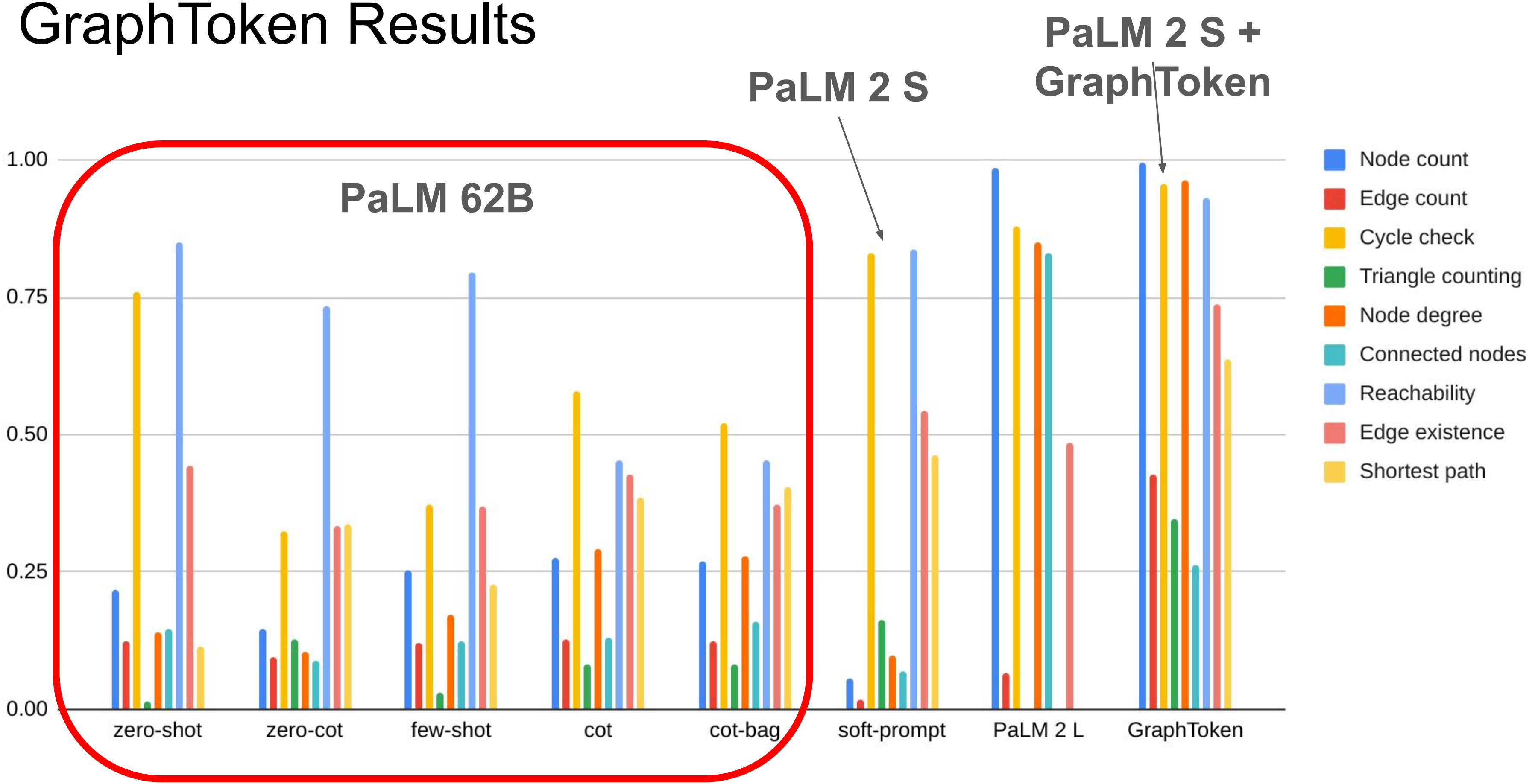
GraphToken: *Graphs that Talk*



Goal: We seek a GNN encoding function that can project structured data into the token space of a frozen model.

LET YOUR GRAPH DO THE TALKING: ENCODING STRUCTURED DATA FOR LLMS
https://arxiv.org/pdf/2402.05862.pdf

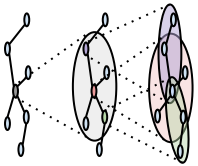
GraphToken Results



What else is at the intersection of Graphs and LLMs?

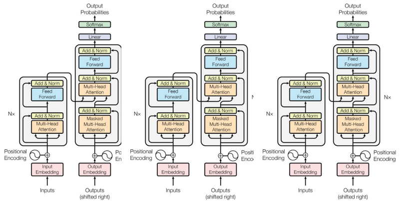
Idea 3: Theory - Graph capabilities of Transformers

GNNs

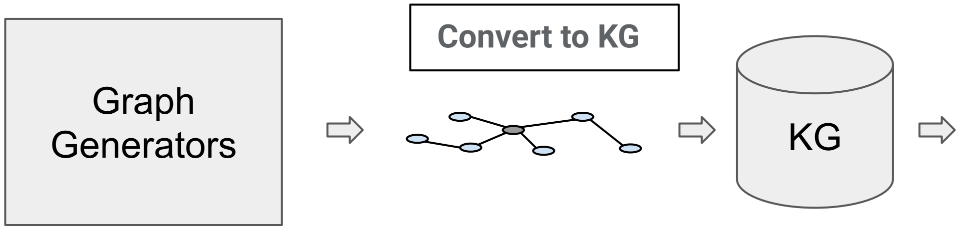


VS

Transformers



Idea 4: Using Graphs to generate unseen synthetic data patterns

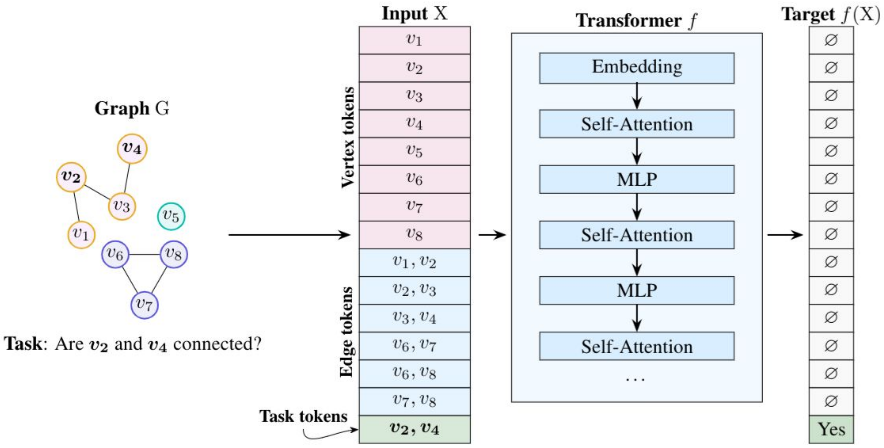


"E11 was the R21 of E23 from 1983 to 1985.
E23 was the R21 of E32 from 2007 to 2013.
E51 was the R17 of E23 from 2004 to 2009.
E32 was the R30 of E4 from 2010 to 2012."

Question 3: What can we say about Graph Algorithms in Transformers?

Our new work explores the theoretical underpinnings of graph algorithms in Transformers!

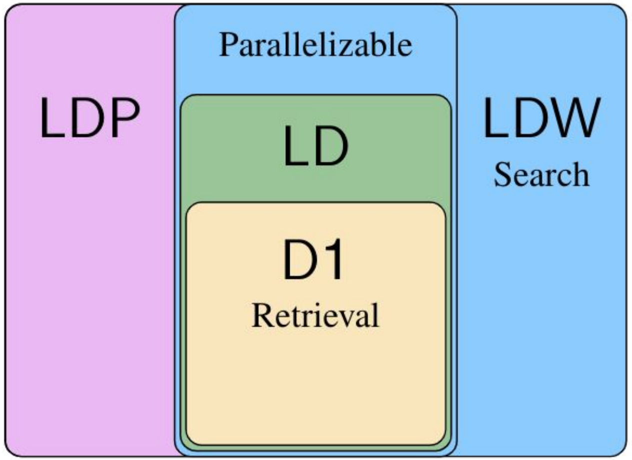
We analyze the “token only” (no text representation) →



Understanding Transformer Reasoning Capabilities via Graph Algorithms

Clayton Sanford[1,2]*, Bahare Fatemi[1], Ethan Hall[3], Anton Tsitsulin[1], Mehran Kazemi[4], Jonathan Halcrow[1], Bryan Perozzi[1], and Vahab Mirrokni[1]
[1]Google Research, [2]Columbia University, [3]Google, [4]Google DeepMind

Hierarchy of Graph Algorithms



(a) The complexity hierarchy

Task class	Example tasks	Complexity
Retrieval (§3.3)	Node count	D1
$L = 1$	Edge count	D1
$m = O(\log N)$	Edge existence	D1
	Node degree	D1
Parallelizable (§3.1)	Connectivity	LD
$L = O(\log N)$	Cycle check	LDP$\cap$ LDW
$m = O(N^{\epsilon})$	Bipartiteness	LDP$\cap$ LDW
Search (§3.2)	Shortest path	LDW
$L = O(\log N)$	Diameter	LDW
$m = O(N^{1/2+\epsilon})$		

(b) Example tasks and their complexity.

Figure 2: A summary of the theoretical hierarchy of Section 3 that visualizes which type of graph reasoning tasks can be solved in which transformer scaling regime (Depth1 (D1), LogDepth (LD), LogDepthWide (LDW) and LogDepthPause (LDP)).

Experimental Results

- Transformers excel at global tasks
  a. Shortest path, connectedness, etc
  b. “Pure GNNs” may have capacity problems with tasks that require

Model	Node Degree 1K		Cycle Check 1K	
GCN [42]	9.8		83.2	
MPNN [26]	99.4		99.0	
GIN [82]	36.2		98.8	
60M transformer	31.6		97.1	

GNN winning on local tasks

- GNN’s graph bias does help them learn with less data, but
  a. With more data, Transformers can be very competitive

Model	# of training samples 1K	
GCN [42]	50.2	
MPNN [26]	66.8	
GIN [82]	54.0	
60M transformer	57.4	

Shortest path results

note: ~100k params

Experimental Results

- Transformers excel at global tasks
  a. Shortest path, connectedness, etc
  b. “Pure GNNs” may have capacity problems with tasks that require

- GNN’s graph bias does help them learn with less data, but
  a. With more data, Transformers can be very competitive

Model	Node Degree		Cycle Check	
	1K	100K	1K	100K
GCN [42]	9.8	9.4	83.2	83.2
MPNN [26]	99.4	**99.8**	99.0	**100.0**
GIN [82]	36.2	37.8	98.8	83.2
60M transformer	31.6	91.7	97.1	98.0

GNN winning on local tasks

Model	# of training samples	
	1K	100K
GCN [42]	50.2	55.0
MPNN [26]	66.8	72.6
GIN [82]	54.0	58.6
60M transformer	57.4	**97.1**

note: ~100k params

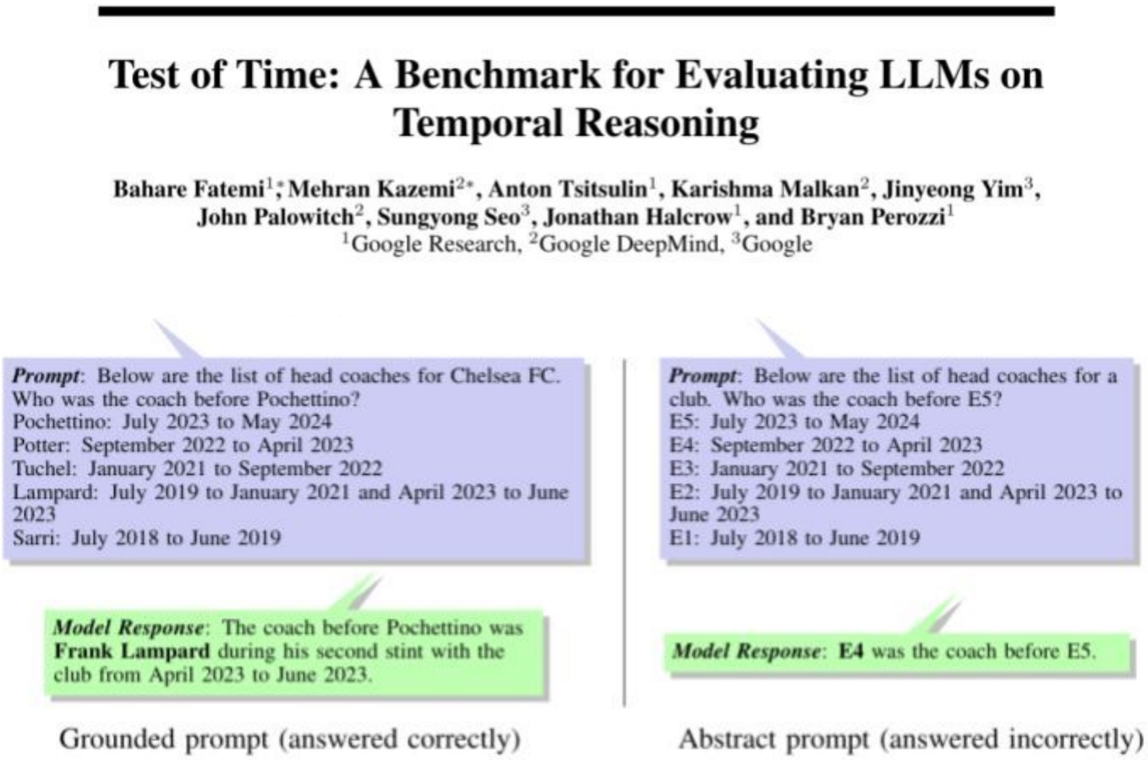
Shortest path results

Idea 4: Graphs for synthetic data generation

Synthetic data generation is becoming more popular for evaluating LLMs.

- Why?
  - Synthetic data is harder to memorize
  - Synthetic data can disentangle “reasoning” from memorized facts (“parametric knowledge”)

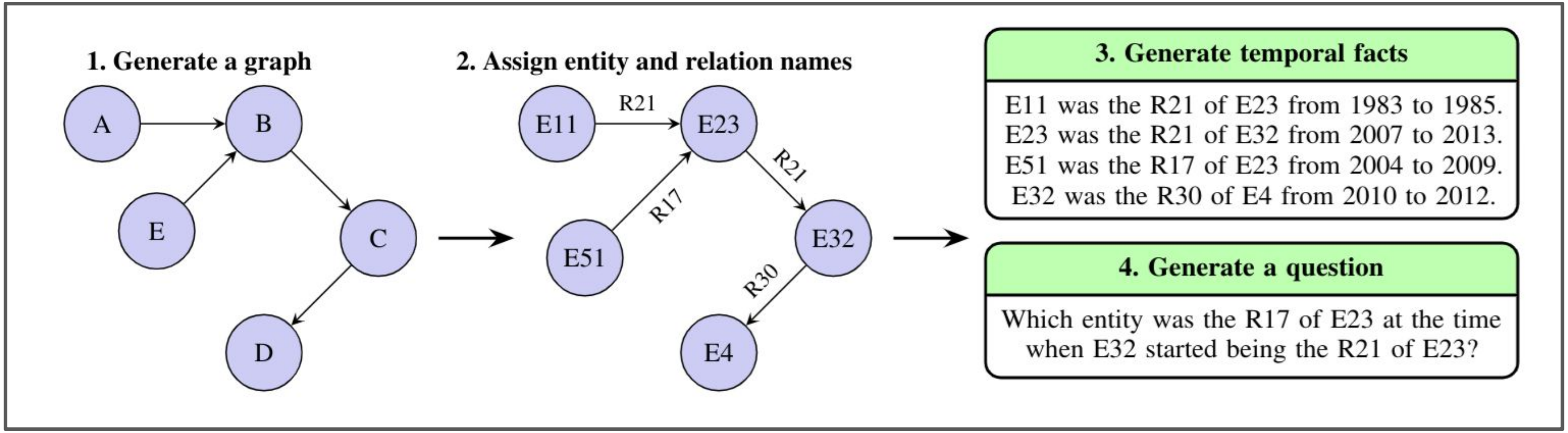
- Graphs can help here!



Test of Time: A Benchmark for Evaluating LLMs on Temporal Reasoning

Bahare Fatemi[1]*, Mehran Kazemi[2*], Anton Tsitsulin[1], Karishma Malkan[2], Jinyeong Yim[3], John Palowitch[2], Sungyong Seo[3], Jonathan Halcrow[1], and Bryan Perozzi[1]
[1]Google Research, [2]Google DeepMind, [3]Google

Turning Random Graphs into Random KGs



Illustrating different synthetic temporal datasets

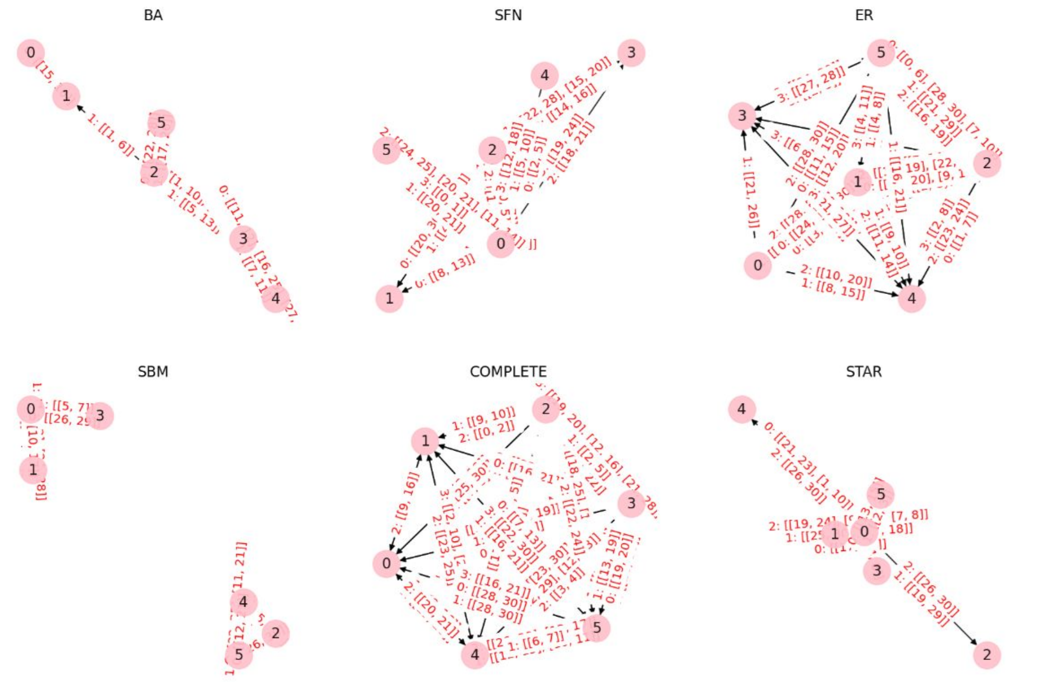


Figure 5: A visualization of a representative graph from each graph generator: Erdős-Rényi (ER), Scale-Free Networks (SFN), Barabási–Albert (BA), Stochastic Block Model (SBM), star-graph, and complete-graph.

Conclusions

1. Adding graph information can help frozen LLMs solve graph tasks.

2. The way you represent your structured data matters!

3. Transformers are less sample efficient than GNNs, but with enough training data they do get a lot better.

4. Graphs are useful for synthetic data generation for LLMs too.

5. … many more details in the respective papers

Understanding Transformer Reasoning Capabilities via Graph Algorithms
Sanford, Fatemi, Hall, Tsitsulin, Kazemi, Halcrow, Perozzi, Mirrokni
https://arxiv.org/pdf/2405.18512

Test of Time: A Benchmark for Evaluating LLMs on Temporal Reasoning
Fatemi, Kazemi, Tsitsulin, Malkan, Yim, Palowitch, Seo, Halcrow, Perozzi
https://arxiv.org/abs/2406.09170

Thanks!

Bryan Perozzi

LinkedIn:

Twitter: @phanein

TALK LIKE A GRAPH: ENCODING GRAPHS FOR LARGE LANGUAGE MODELS (ICLR'24)
Fatemi, Halcrow, Perozzi
https://arxiv.org/pdf/2310.04560.pdf

LET YOUR GRAPH DO THE TALKING: ENCODING STRUCTURED DATA FOR LLMS
Perozzi, Fatemi, Zelle, Tsitsulin, Kazemi, Al-Rfou, Halcrow
https://arxiv.org/pdf/2402.05862.pdf