



Yuhan Li, Hanlin Long, DSA Thrust

# Make LLMs Solve Graph Problems Explicitly

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

# Recap - Solve graph problems in natural language

## Goal

We aim at leveraging instruction-tuning to build a powerful **instruction-following LLM** that can map textual descriptions of graphs and structures, and then solve different graph problems **explicitly** in natural language

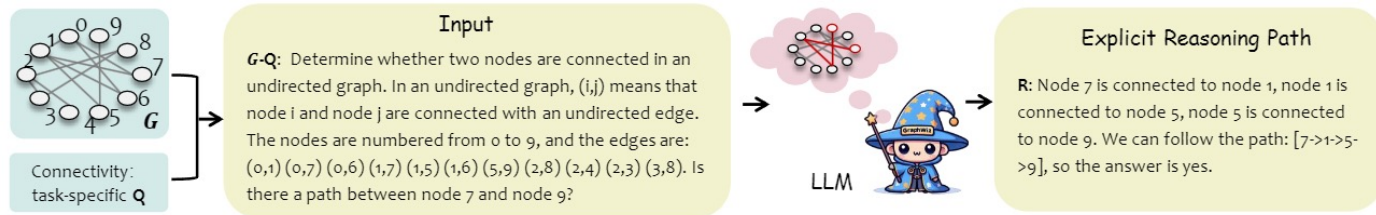


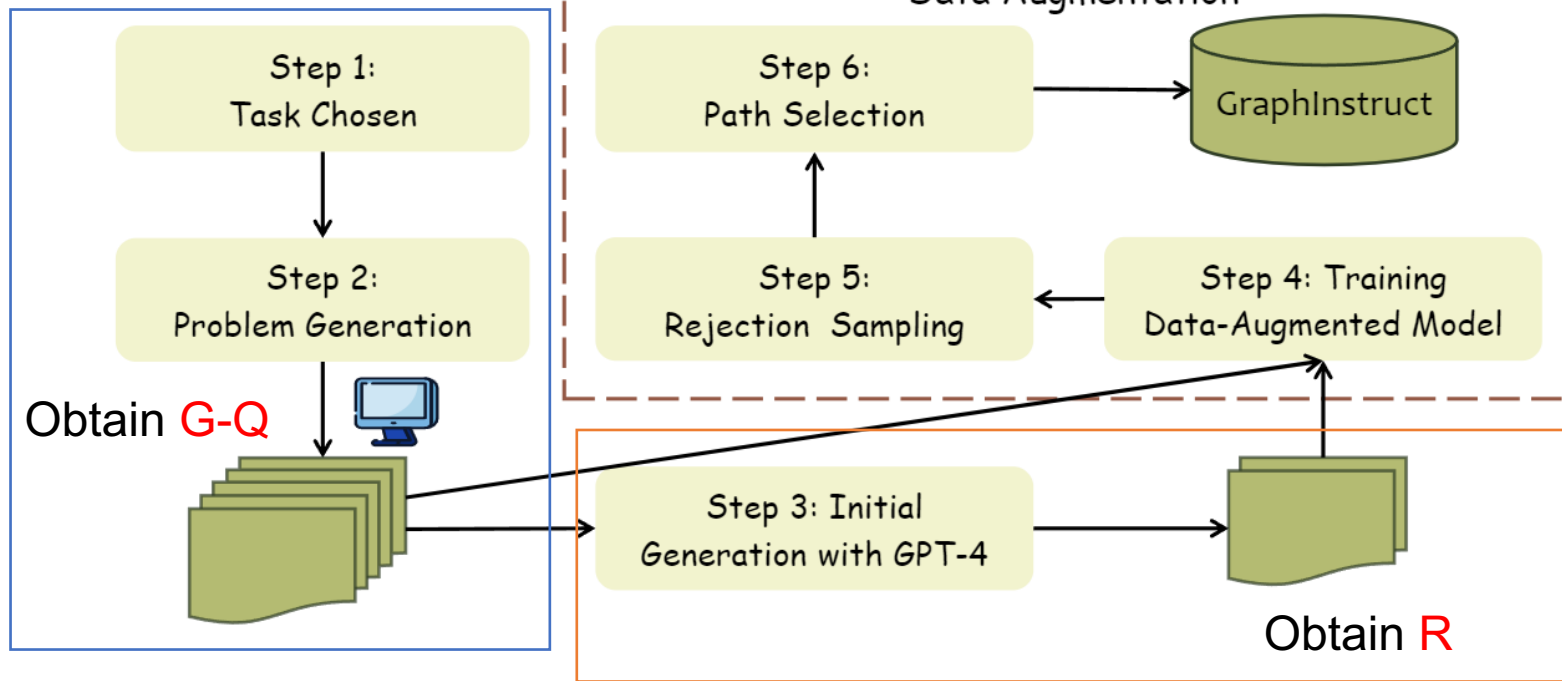
Figure 1: An example of solving the Connectivity task explicitly within natural language via LLMs.

Collection Process

# GraphInstruct

# GraphInstruct

## Self-Augment



# Graph Problem Generation

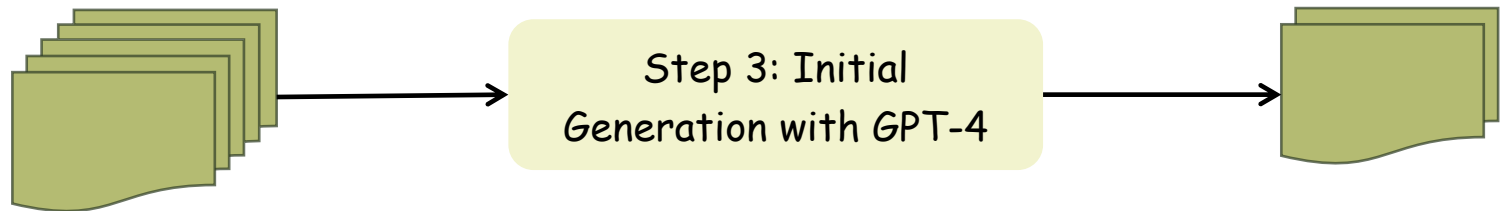
## Strategy

- Diverse Distributions:**  
Node range and edge density
- Length Constraints:**  
No more than 4K
- Unique Instances**
- Scalable Graph Sizes.**

Initial 27k graph problem (G-Q)

Problem	Definition	Time Complexity	Weighted?	Directed?	Node Range	Difficulty
Cycle Detection	Detect if a given graph $\mathcal{G}$ contains any cycles.	$O( E )$	✗	✗	[2, 100]	Easy
Connectivity	Assess if two nodes $u$ and $v$ in a given graph $\mathcal{G}$ are connected via a path.	$O( V  +  E )$	✗	✗	[2, 100]	Easy
Bipartite Graph Check	Judge if a given graph $\mathcal{G}$ is bipartite.	$O( V  +  E )$	✗	✓	[2, 100]	Easy
Topological Sort	Find a topological ordering of vertices in a directed acyclic graph $\mathcal{G}$ .	$O( V  +  E )$	✗	✓	[2, 50]	Easy
Shortest Path	Compute the shortest path between two specific nodes $u$ and $v$ in a given graph $\mathcal{G}$ .	$O( E  +  V \log V )$	✓	✗	[2, 100]	Medium
Maximum Triangle Sum	Find the maximum sum of weights for any connected triplet of vertices in a given graph $\mathcal{G}$ .	$O( V ^3)$	✓	✗	[2, 25]	Medium
Maximum Flow	Calculate the maximum flow from a source node $s$ to a sink node $t$ in a directed graph $\mathcal{G}$ .	$O( V ^2\sqrt{ E })$	✓	✓	[2, 50]	Medium
Hamilton Path	Determine if a given graph $\mathcal{G}$ has a Hamiltonian path that visits each vertex exactly once.	NP-Complete	✗	✗	[2, 50]	Hard
Subgraph Matching	Verify if there exists a subgraph in $\mathcal{G}$ that is isomorphic to a given graph $\mathcal{G}'$ .	NP-Complete	✗	✓	[2, 30]	Hard

# Explicit Reasoning Paths Generation



Initial **27k** graph problem (G-Q)

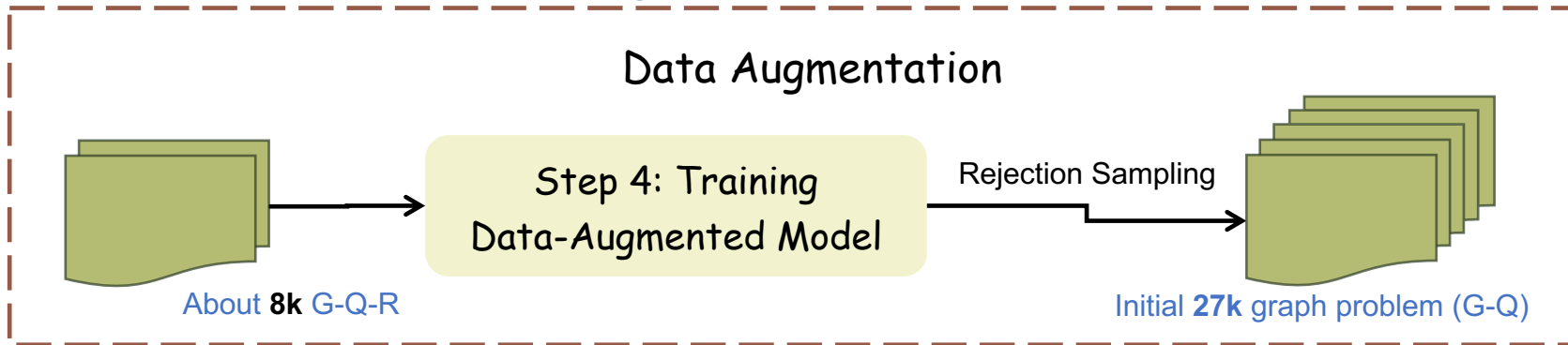
About **8k** G-Q-R

GPT-4 performs poorly on many graph tasks, such as resulting in fewer than **100** correct samples for the Maximum Flow task.

**TODO:** We need data augmentation to get more G-Q-R!



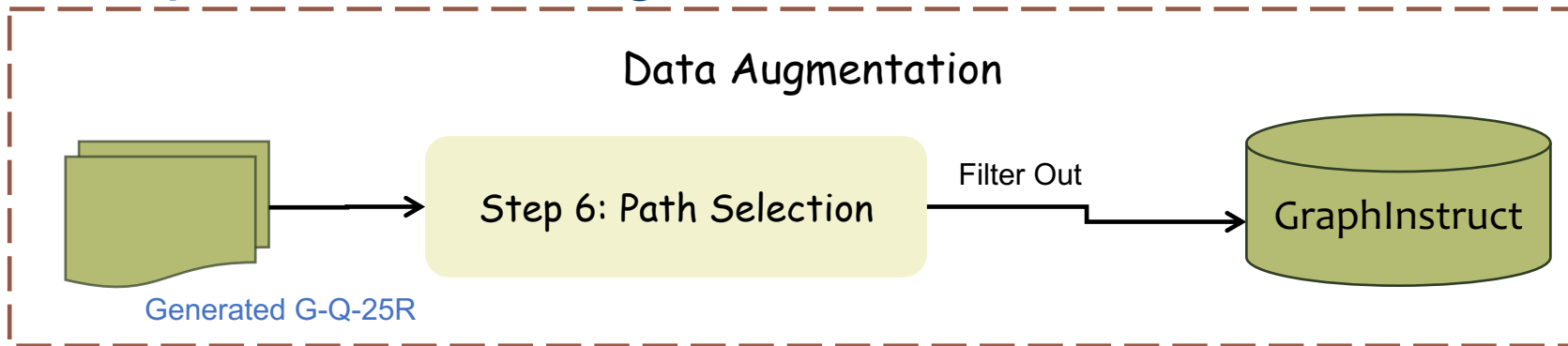
# Explicit Reasoning Paths Selection



Inference data-augmented model 25 times for each G-Q:

- Increasing G-Q quantity
- Increasing R diversity

# Explicit Reasoning Paths Selection



**String-based** strategies:

- Edit distance
- TF-IDF
- Jaccard similarity

The most different and correct one!

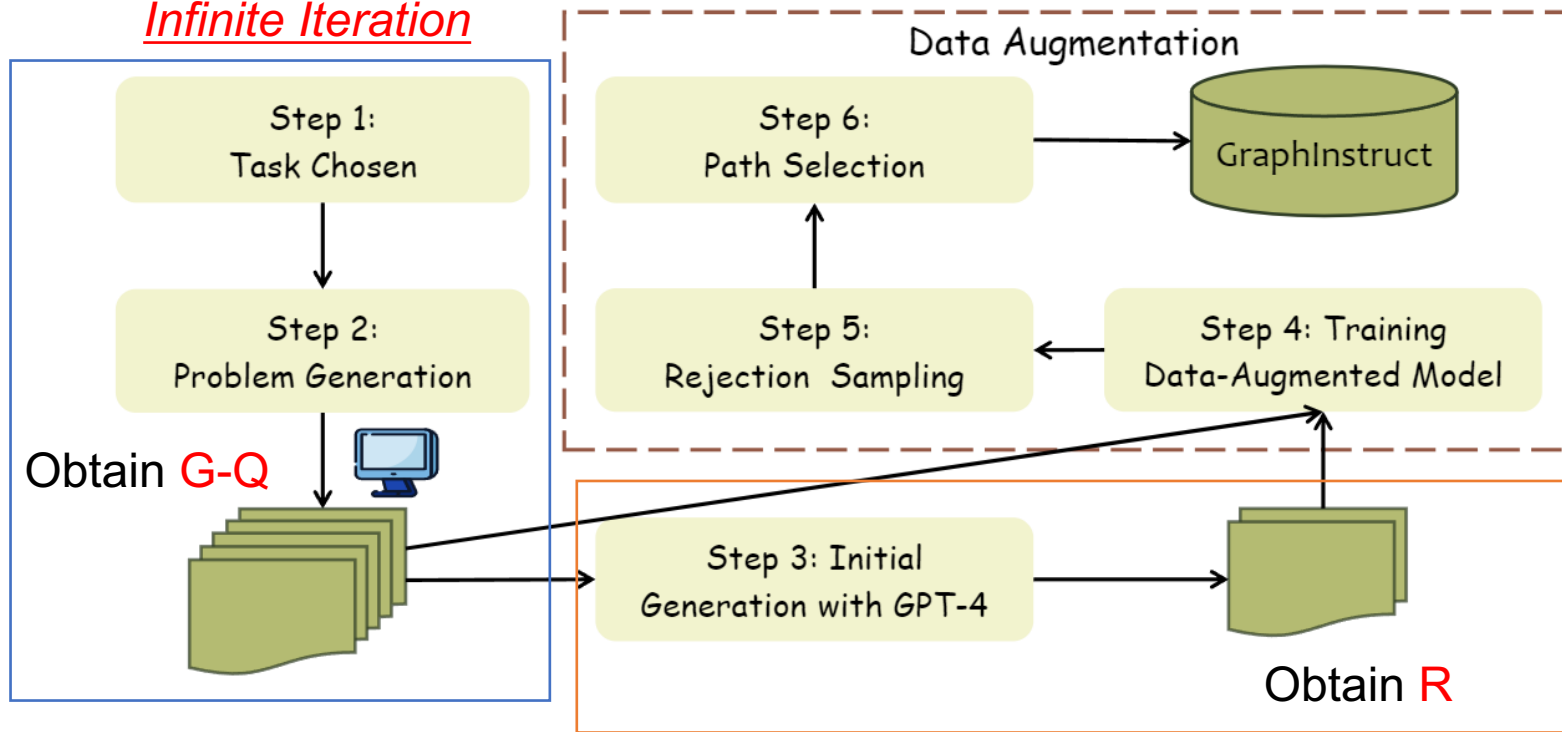
**Semantic-based** strategies:

- Cosine similarity
- K-means



# GraphInstruct

## Infinite Iteration



# GraphInstruct

Tasks		Easy				Medium			Hard		Sum.
		cycle	connect	bipartite	topology	shortest	triangle	flow	hamilton	Subgraph	
Train	Total $\mathcal{G}\text{-}Q$	3,717	2,687	2,013	902	1,392	2,756	405	2,097	1,435	17,158
	Total $\mathcal{V}$	84,980	79,853	58,860	10,146	23,204	14,714	4,333	33,284	7,847	315,051
	Total $\mathcal{R}$	13,122	10,001	9,324	4,481	5,859	13,483	747	8,454	6,274	<b>72,785</b>
Test	Total $\mathcal{G}\text{-}Q$	400	400	400	400	400	400	400	400	400	3,600
	Total $\mathcal{V}$	19,570	19,500	19,515	9,449	19,449	4,990	10,024	9,732	6,594	118,823

Training Strategies

# GraphWiz

# GraphWiz

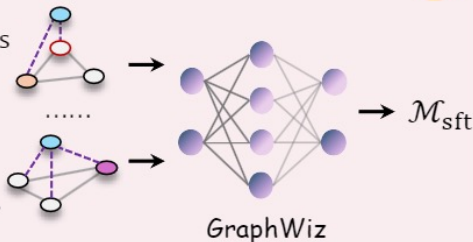
## Phase 1: Mixed-Task Instruction Tuning



Detect if a given graph  $G$  contains any cycles.

.....

Assess if two nodes  $u$  and  $v$  in graph  $G$  are connected via a path.



## Phase 2: DPO Alignment



Preferred Reasoning Path



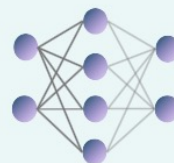
$\mathcal{R}_w$

$\vee$

$\mathcal{R}_l$

Dispreferred Reasoning Path

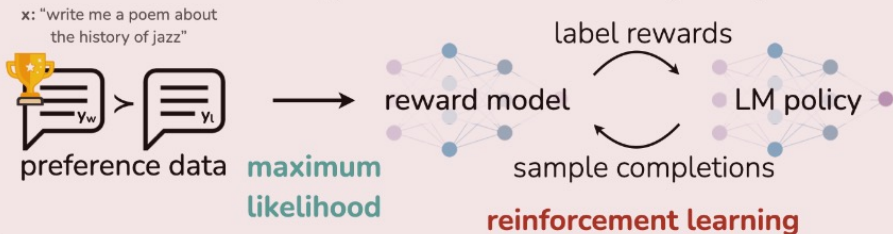
Maximum Likelihood



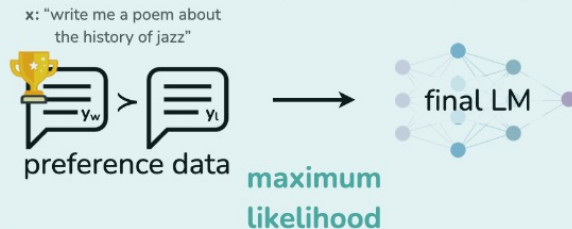
- **Phase 1:** The first hones the model's ability to interpret and solve a wide range of graph problems.
- **Phase 2:** DPO Alignment, further sharpens the model's reasoning by training it to distinguish between more and less effective problem-solving paths.

# DPO: Direct Preference Optimization

## Reinforcement Learning from Human Feedback (RLHF)



## Direct Preference Optimization (DPO)



- DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, fitting an implicit reward model whose corresponding optimal policy can be extracted in closed form.
- Of note, DPO requires a separate training corpus that belongs to a similar domain with SFT model.

Analysis and Ablations

# Experiments

# Results

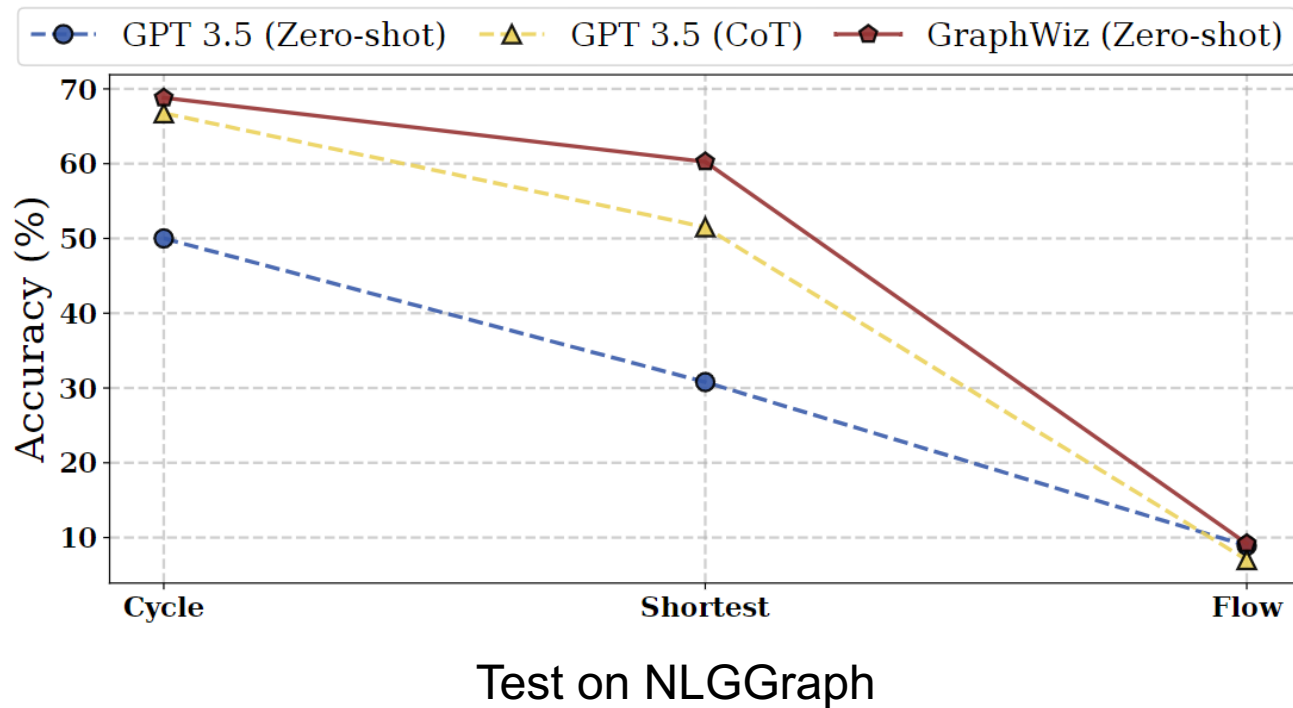
- **Q1:** How does GraphWiz perform on these graph tasks, particularly in comparison to the most powerful closed-source model, GPT-4?
- **Q2:** What impact does the variation in training data volume have on GraphWiz's performance?
- **Q3:** What is the transferability of GraphWiz across different graph tasks?
- **Q4:** How do changes in the number of nodes in a graph affect GraphWiz's performance? Further, what is the maximum complex graph that it can effectively handle?



# Main Results

Models	Easy				Medium			Hard		Average
	cycle	connect	bipartite	topology	shortest	triangle	flow	hamilton	subgraph	
In-context Learning										
GPT-4 (zero-shot)	38.75	17.00	65.25	5.00	9.25	5.75	3.25	59.25	45.50	27.67
GhatGPT (2-shot)	51.25	43.75	70.75	4.50	3.50	17.25	8.50	54.25	43.00	32.97
GPT-4 (2-shot)	52.50	62.75	74.25	25.25	18.25	31.00	7.75	75.75	46.75	43.81
Mistral-7B										
Naive SFT	73.75	83.50	78.50	1.00	23.00	47.00	28.75	31.75	41.25	46.56
GraphWiz	92.00	89.50	72.00	19.00	31.25	38.75	29.25	26.50	85.50	53.75
GraphWiz-DPO	85.50	79.50	85.50	85.25	12.50	29.00	35.50	62.75	48.50	58.22
LLaMA 2-7B										
Naive SFT	73.75	83.50	41.25	4.00	9.50	30.00	16.50	69.00	75.45	44.81
GraphWiz	91.50	87.00	74.00	18.00	28.00	38.25	24.50	52.25	82.25	55.08
GraphWiz-DPO	89.00	82.50	84.75	46.75	24.00	52.75	43.50	81.50	77.25	65.00
LLaMA 2-13B										
Naive SFT	73.75	83.75	59.00	0.50	11.75	34.75	24.25	59.75	54.75	44.69
GraphWiz	94.75	87.00	78.00	28.00	27.75	36.00	24.50	59.00	81.50	57.39
GraphWiz-DPO	87.50	88.50	88.25	72.75	22.00	48.75	43.75	46.50	77.00	63.89

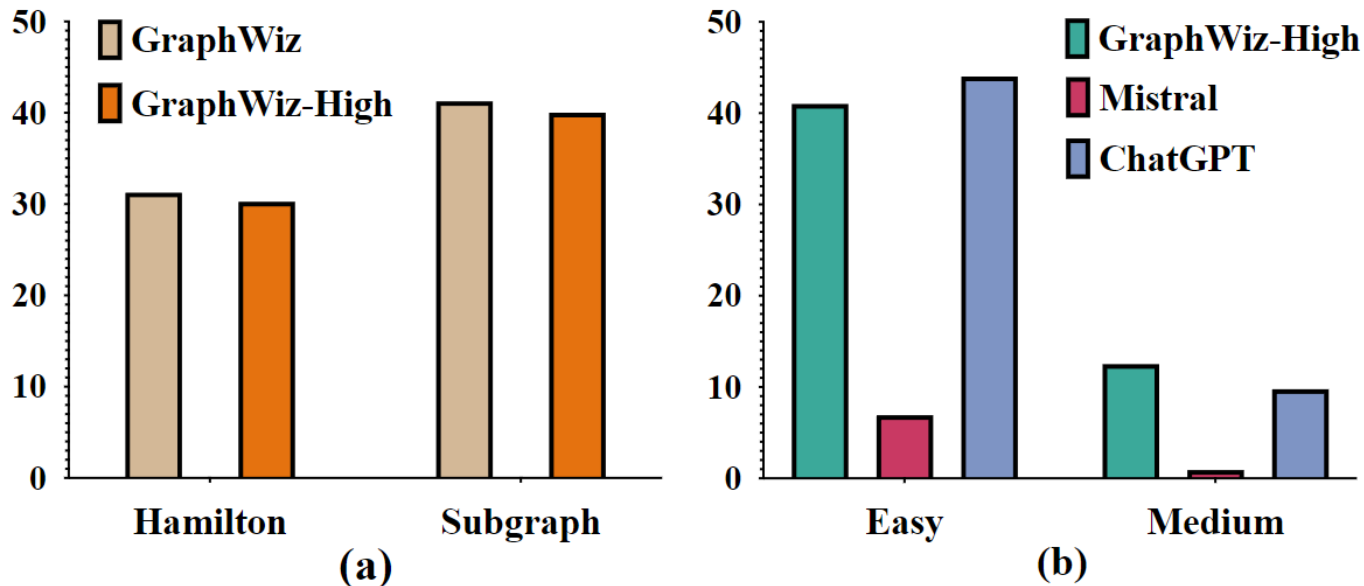
# Main Results



# GraphWiz with Reasoning Path Increasing

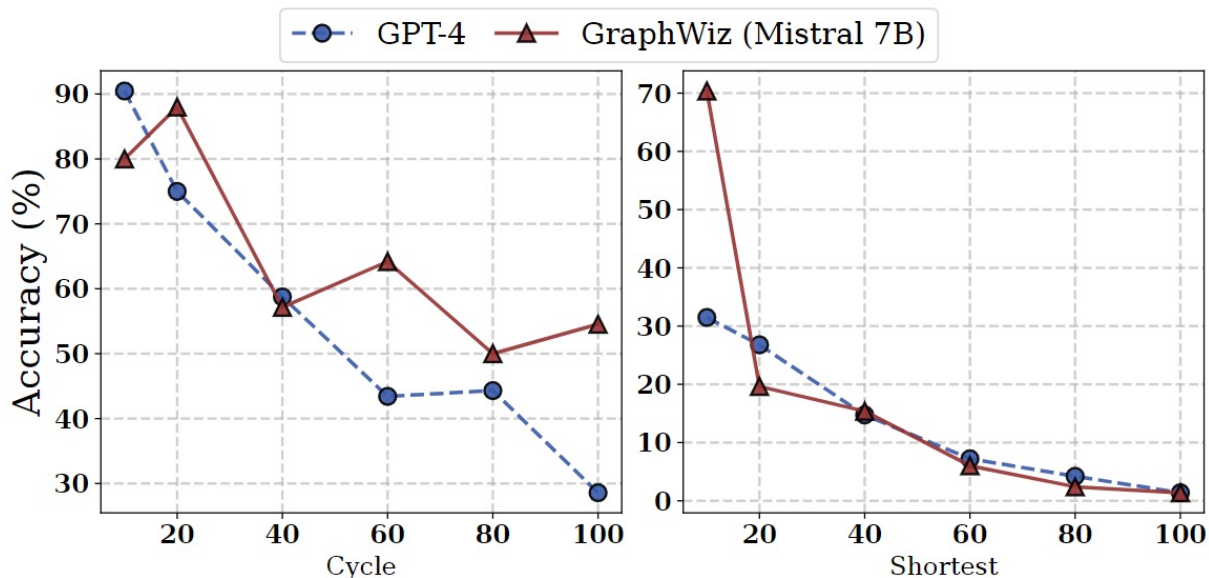
$\mathcal{G}:\mathcal{R}$	Easy				Medium			Hard		Avg.
	cycle	connect	bipartite	topology	shortest	triangle	flow	hamilton	substructure	
GraphWiz (Mistral-7B)										
1:1	80.25	87.50	78.50	1.00	23.00	47.00	28.75	31.75	41.25	46.56
1:2	78.25	85.00	81.75	27.00	28.00	47.00	24.50	34.25	42.75	49.83
1:3	90.75	88.25	62.75	21.75	30.75	39.75	32.25	30.75	84.25	53.47
1:4	90.25	87.00	78.00	19.75	26.00	43.25	26.50	26.50	85.00	53.58
1:5	92.00	89.50	72.00	19.00	31.25	38.75	29.25	26.50	85.50	53.75
GraphWiz (LLaMA 2-7B)										
1:1	77.75	84.00	71.75	1.00	18.50	46.25	22.00	31.25	45.00	44.17
1:2	94.00	85.25	76.25	17.25	23.50	35.75	19.50	40.50	83.00	53.00
1:3	93.00	89.00	72.50	18.25	24.75	40.50	22.50	45.00	83.25	54.00
1:4	92.75	86.00	77.75	21.75	23.25	41.50	25.25	40.25	80.75	54.36
1:5	91.50	87.00	74.00	18.00	28.00	38.25	24.50	52.25	82.25	55.08

# Transferability of GraphWiz



GraphWiz-High: is trained exclusively on two high-complexity (NP-Complete) graph tasks: Hamilton Path and Subgraph Matching

# GraphWiz with Graph Complexity Increasing



# Cases

**G-Q:** Determine whether or not there is a cycle in an undirected graph. In an undirected graph.,the nodes are numbered from **0 to 88**, and the edges are: (**0, 73**) (0, 51) (0, 10) (0, 63) (**0, 28**) (1, 57) (1, 84) (1, 61) (1, 5) (1, 24) (2, 84) (2, 3) (2, 66) (2, 68) (2, 17) (2, 35) (2, 34) (2, 15) (3, 39) (3, 52) (3, 16) (3, 15) (3, 8) (4, 69) (4, 85) (4, 36) (4, 72) (5, 44) (6, 77) (6, 7) (7, 85) (8, 64) (8, 23) (8, 28) (9, 34) (9, 31) (9, 61) (9, 28) (10, 26) (11, 37) (11, 39) (11, 19) (11, 64) (13, 73) (13, 61) (13, 80) (13, 85) (14, 86) (14, 59) (14, 32) (14, 58) (14, 85) (14, 66) (15, 43) (15, 48) (15, 73) (15, 19) (15, 47) (15, 68) (16, 46) (16, 60) (16, 84) (**17, 44**) (17, 72) (**17, 36**) (17, 37) (17, 61) (18, 20) (18, 24) (18, 22) (18, 41) (19, 45) (19, 83) (20, 25) (20, 29) (21, 38) (21, 64) (21, 24) (21, 22) (21, 34) (22, 23) (23, 34) (23, 30) (23, 83) (24, 47) (24, 50) (25, 59) (25, 42) (25, 70) (25, 72) (26, 45) (26, 30) (26, 87) (26, 80) (26, 50) (27, 77) (27, 58) (27, 60) (27, 29) (**28, 36**) (28, 59) (28, 88) (29, 31) (29, 33) (29, 56) (29, 55) (29, 78) (29, 73) (29, 82) (30, 34) (31, 58) (31, 70) (32, 78) (33, 73) (33, 68) (34, 48) (34, 88) (34, 51) (34, 66) (34, 54) (35, 43) (35, 56) (36, 44) (36, 39) (37, 57) (37, 71) (38, 47) (38, 58) (38, 88) (39, 74) (39, 48) (39, 45) (39, 42) (40, 60) (40, 59) (40, 71) (40, 76) (41, 79) (42, 60) (42, 62) (42, 72) (43, 57) (44, 47) (44, 74) (44, 75) (**44, 82**) (44, 70) (45, 68) (45, 67) (45, 71) (46, 84) (47, 56) (48, 75) (49, 71) (49, 58) (50, 76) (51, 55) (51, 87) (51, 79) (52, 78) (56, 69) (56, 83) (56, 79) (57, 76) (57, 70) (57, 72) (57, 78) (59, 70) (60, 80) (60, 76) (60, 77) (60, 75) (61, 74) (61, 77) (62, 84) (62, 66) (66, 69) (68, 88) (70, 88) (70, 76) (71, 81) (71, 72) (71, 86) (72, 75) (**73, 82**) (73, 83) (74, 85). **Is there a cycle in this graph?**

**GPT-4:** No, there is no cycle in this graph. ✕

**GraphWiz:** There are many possible ways to find a cycle in this graph, but one example is: Starting from node 0, we can go to node 73 (via edge 0-73), then to node 82 (via edge 73-82), then to node 44 (via edge 82-44), then to node 17 (via edge 44-17), then to node 36 (via edge 17-36), then to node 28 (via edge 36-28), and back to node 0 (via edge 28-0). This forms a cycle [**0-73-82-44-17-36-28-0**] without revisiting any edge. So, there is a cycle in this graph. ✓

### Case 1 (Connectivity)

**G-Q:** Determine whether two nodes are connected in an undirected graph. In an undirected graph, (i,j) means that node i and node j are connected with an undirected edge. Given a graph and a pair of nodes, you need to output Yes or No, indicating whether the node i and node j are connected. Q: The nodes are numbered from 0 to 12, and the edges are: (0, 12) (0, 8) (0, 4) (1, 12) (1, 5) (1, 7) (1, 2) (1, 6) (2, 8) (2, 12) (2, 3) (2, 5) (3, 12) (3, 6) (4, 7) (4, 6) (5, 12) (6, 7) (6, 12) (6, 10) (7, 11) (7, 10) (**8, 12**) (9, 12) (9, 11) (**10, 12**). **Is there a path between node 8 and node 10?**

---

**GPT-4:** Yes, there is a path.

**GraphWiz:** Node 8 is connected to node 0, node 2, and node 12. Node 10 is connected to node 6, node 7, and node 12.

We can follow the path: **[8->12->10]**, so the answer is yes. ### Yes. **✓**

### Case 2 (Hamilton Path)

**G-Q:** Determine whether or not there is a Hamiltonian path in an undirected graph. In an undirected graph, (i,j) means that node i and node j are connected with an undirected edge. Given a graph, you need to output Yes or No, indicating whether there is a Hamiltonian path in the graph. Q: The nodes are numbered from 0 to 4, and the edges are: (0, 1) (0, 4) (1, 3) (2, 3) (2, 4). **Is there a Hamiltonian path in this graph?**

---

**GPT-4:** Yes, there is a path.

**GraphWiz:** To determine if a Hamiltonian path exists in an undirected graph, we need to check if there's a path that visits each node exactly once. We can start at any node and try to find a path that covers all the nodes.

**For example, starting at node 0, we can go to node 1, then node 3, then node 2, then node 4. This is a valid Hamiltonian path, as it visits each node once and only once.** Another possible Hamiltonian path is **[4, 2, 3, 1, 0]. ✓**



### Case 3 (Shortest Path)

**G-Q:** Find the shortest path between two nodes in an undirected graph. In an undirected graph,  $(i,j,k)$  means that node  $i$  and node  $j$  are connected with an undirected edge with weight  $k$ . Given a graph and a pair of nodes, you need to output the shortest path between the two nodes. Q: The nodes are numbered from 0 to 8, and the edges are:  $(0,1,4)$   $(1,2,7)$   $(1,7,1)$   $(1,3,4)$   $(2,6,2)$   $(2,4,8)$   $(2,7,5)$   $(3,6,1)$   $(4,8,3)$   $(5,6,6)$   $(6,8,8)$   $(7,8,7)$ . **Give the weight of the shortest path from node 0 to node 8.**

---

**GPT-4:** The weight of the shortest path from node 0 to node 8 is 14. ✕

**GraphWiz:** All the paths from node 0 to node 8 are: 0,1,7,8 with a total weight of  $\langle\langle 4 + 1 + 7 = 12 \rangle\rangle$ , 0,1,3,6,8 with a total weight of  $\langle\langle 4 + 4 + 1 + 8 = 17 \rangle\rangle$ , 0,1,2,6,8 with a total weight of  $\langle\langle 4 + 7 + 2 + 8 = 21 \rangle\rangle$ , 0,1,2,4,8 with a total weight of  $\langle\langle 4 + 7 + 8 + 3 = 22 \rangle\rangle$ , 0,1,2,7,8 with a total weight of  $\langle\langle 4 + 7 + 5 + 7 = 23 \rangle\rangle$ . The weight of path 0,1,7,8 is the smallest, so the shortest path from node 0 to node 8 is [0,1,7,8] with a total weight of 12. ✓

#### Case 4 (Subgraph Matching)

**G-Q:** Determine if a smaller graph is present as an exact match within a larger graph. In a directed graph,  $(i \rightarrow j)$  means that node  $i$  and node  $j$  are connected with a directed edge from node  $i$  to node  $j$ . Given a graph  $G$  and a subgraph  $G'$ , you need to output Yes or No, indicating whether subgraph  $G'$  is present within the directed graph  $G$ . **Q:** The nodes of graph  $G$  are numbered from 0 to 8, and the edges are:  $(0 \rightarrow 1)$   $(0 \rightarrow 8)$   $(0 \rightarrow 5)$   $(0 \rightarrow 6)$   $(0 \rightarrow 4)$   $(1 \rightarrow 3)$   $(1 \rightarrow 5)$   $(1 \rightarrow 6)$   $(2 \rightarrow 6)$   $(2 \rightarrow 3)$   $(3 \rightarrow 6)$   $(3 \rightarrow 5)$   $(3 \rightarrow 4)$   $(5 \rightarrow 7)$   $(7 \rightarrow 8)$ . The nodes of subgraph  $G'$  are numbered from  $a$  to  $e$ , and the edges are:  $(a \rightarrow e)$   $(a \rightarrow d)$   $(b \rightarrow c)$   $(c \rightarrow e)$ . **Is subgraph  $G'$  present within graph  $G$  as a direct substructure?**

**GraphWiz:** To find if subgraph  $G'$  is present in graph  $G$ , we need to identify a node in  $G$  that has two outgoing edges to different nodes, just like 'a' in  $G'$ . Then, we need to check if one of those nodes has another outgoing edge to a different node, just like 'e' in  $G'$ . Finally, we need to check if the other node has an incoming edge from a different node, just like 'd' in  $G'$ . Looking at graph  $G$ : Node 0 has five outgoing edges, but none of them lead to a node that has another outgoing edge to a different node, so node 0 cannot be 'a'. Node 1 has three outgoing edges, but none of them lead to a node that has another outgoing edge to a different node, so node 1 cannot be 'a'. Node 2 has two outgoing edges, to nodes 6 and 3. Node 6 has no outgoing edges, so it cannot be 'e'. Node 3 has three outgoing edges, to nodes 6, 5, and 4. Node 6 cannot be 'e', since it has no outgoing edges. Node 5 has one outgoing edge, to node 7, which could be 'e'. Node 4 has no outgoing edges, so it cannot be 'e' or 'd'. Therefore, node 2 cannot be 'a'. Node 3 has four outgoing edges, to nodes 6, 5, 4, and 7. Node 6 cannot be 'e', since it has no outgoing edges. Node 5 has one outgoing edge, to node 7, which could be 'e'. Node 4 has no outgoing edges, so it cannot be 'e' or 'd'. Node 7 has one outgoing edge, to node 8, which could be 'e'. Therefore, node 3 could be 'a', with nodes 5 and 7 being 'e' and 'd' respectively, or vice versa. Checking the connections of node 3's targets: Node 3 to 5 (potential 'e') Node 3 to 7 (potential 'd') Node 5 to 7 (matches 'e' to 'd') Node 7 to 8 (potential 'e') Node 5 to 8 (potential 'd') Node 8 has no outgoing edges, so it cannot be 'e'. Node 5 has one outgoing edge, to node 7, which could be 'e'. Therefore, node 7 cannot be 'd', since it has no incoming edges. Node 5 could be 'd', since it has an incoming edge from node 3. Thus, the sequence  $3 \rightarrow 5 \rightarrow 7$  in  $G$  corresponds to the sequence  $a \rightarrow e \rightarrow d$  in  $G'$ , and the sequence  $3 \rightarrow 7 \rightarrow 8$  in  $G$  corresponds to the sequence  $a \rightarrow d \rightarrow e$  in  $G'$ , which means subgraph  $G'$  is present as a direct substructure in graph  $G$ . ### Yes. ✓

# Takeaways

- High Interpretability
- Superior Performances in Graph Problems
- The framework can be extended to other field.





# Thanks!



香港科技大学(广州)  
THE HONG KONG  
UNIVERSITY OF SCIENCE AND  
TECHNOLOGY (GUANGZHOU)

数据科学与分析学域  
**DATA SCIENCE AND ANALYTICS THRUST**  
信息枢纽  
INFORMATION HUB