Exploring the Potential of Large Language Models (LLMs) in Learning on Graphs

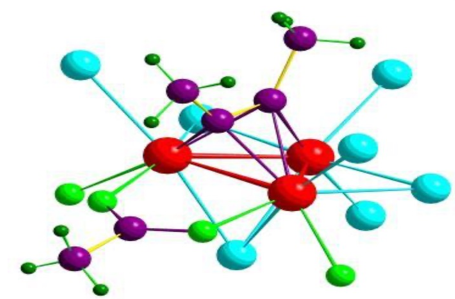
Jiliang Tang

University Foundation Professor
Michigan State University
tangjili@msu.edu

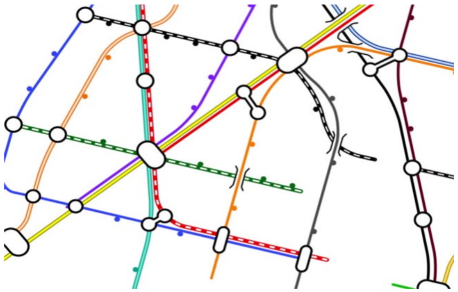
Exploring the Potential of Large Language Models (LLMs) in Learning on Graphs, arXiv:2307.03393

Graph data are everywhere

Social Graphs

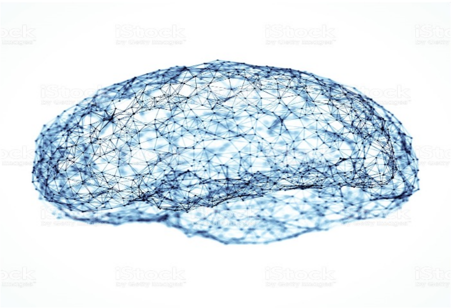


Molecular Graphs

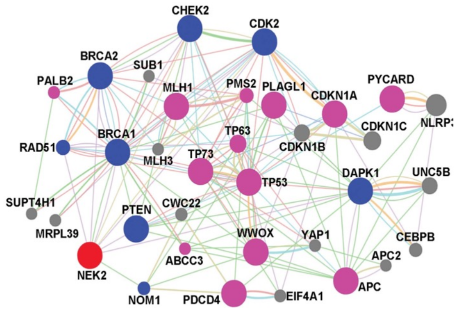


Transportation Graphs

Web Graphs



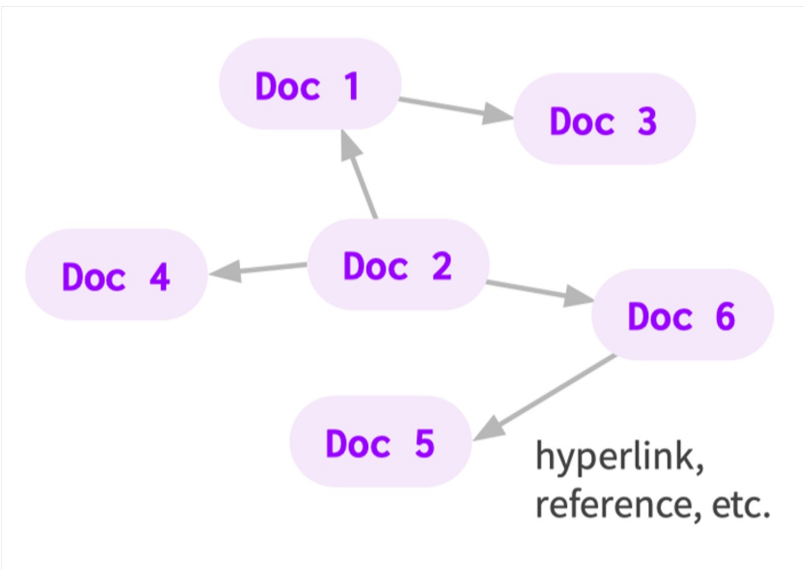
Brain Graphs



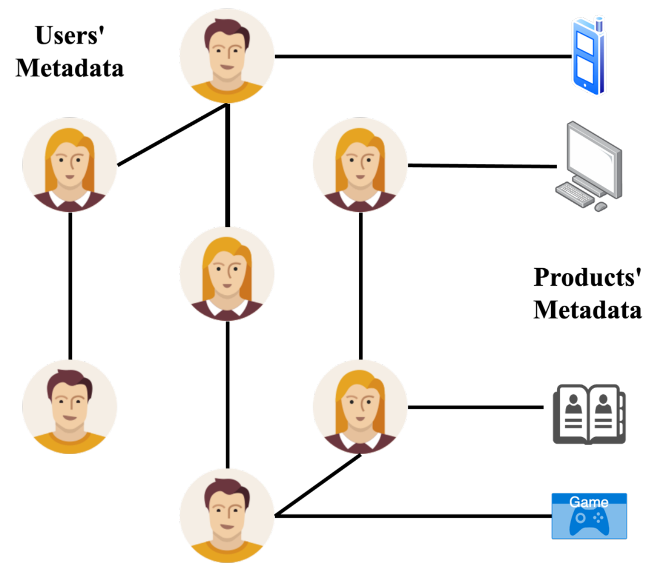
Gene Graphs

Text-attributed graphs (TAGs)

Nodes in graphs are usually associated with text attributes

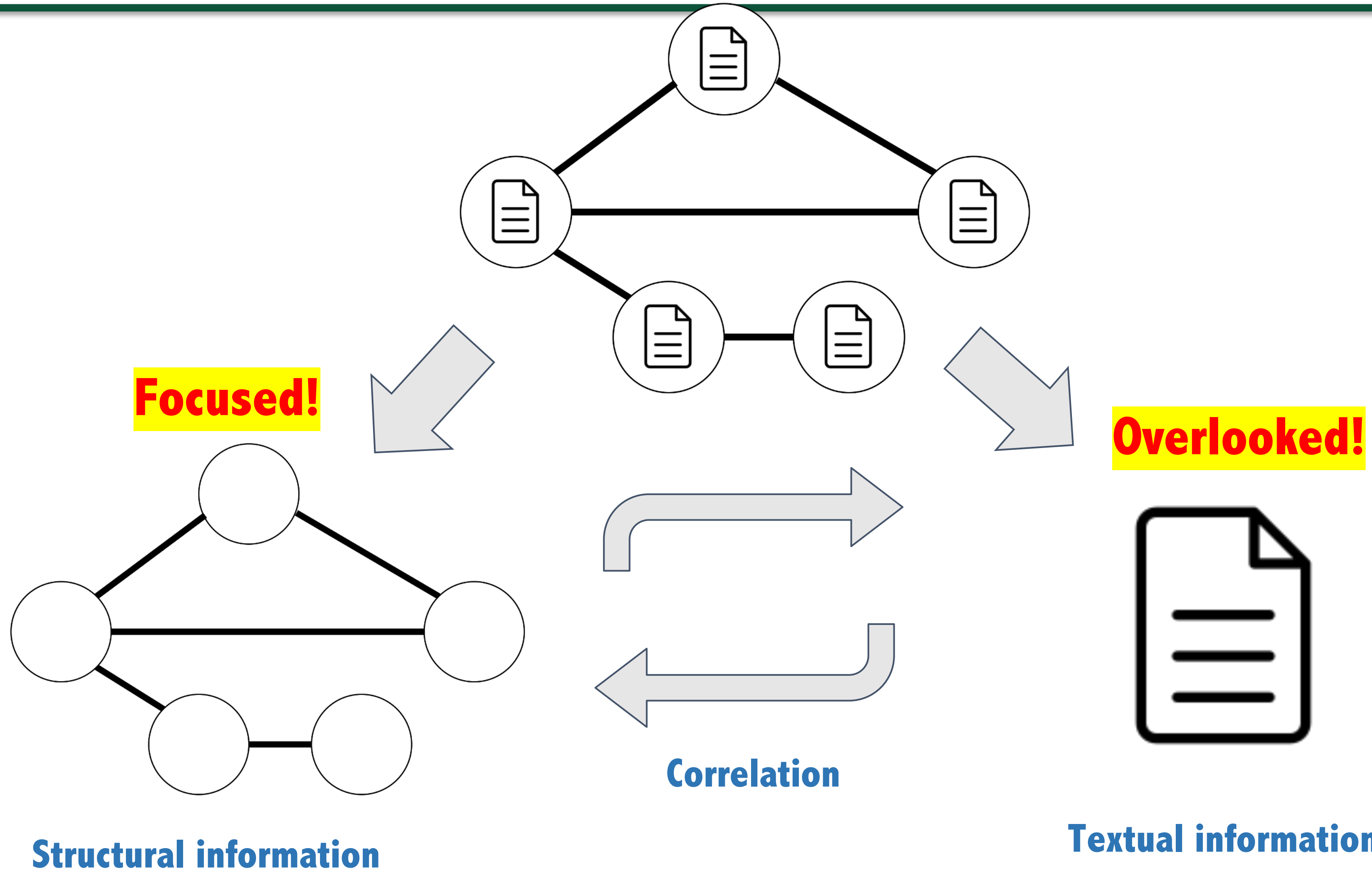


Citation Graphs



E-commerce Graphs

How to effectively process TAGs?



Popular benchmarks majorly use shallow embeddings

Cora,Citeseer,Pubmed,Products	Bag of Words
Arxiv	Word2vec

- **The impact of different embeddings on downstream tasks is often overlooked**
- **These shallow embeddings present potential limitations**

Potential limitations of shallow embeddings

Non-contextualized representations

"machine learning"

"machine gun"

Shallow Embeddings

Same representation for the word "machine"

Potential limitations of shallow embeddings

Limited semantic comprehension capability

I love cats but not dogs

=

I love dogs but not cats

Bag of Words

Understand sentence/document level meanings poorly

Word2vec

Potential limitations of shallow embeddings

Domain-specific feature engineering

Code2vec

Doc2vec

Cell2vec

...

Large Language Models (LLMs)

LLMs' capability can help us mitigate these limitations

Contextualized representations

Superior semantic comprehension capability

Better generalization across different tasks

New challenges

How to effectively leverage various types of LLMs？

Pretrained LMs	BERT	Embeddings are accessible
Sentence Embedding Models	S-BERT	
Open-source LLMs	LLaMA	
Closed-source LLMs	GPT3.5	Embeddings aren't accessible

most powerful

Design pipelines for different models

Embeddings are accessible → Generate embeddings, then combine with GNNs

Embeddings are not accessible → 

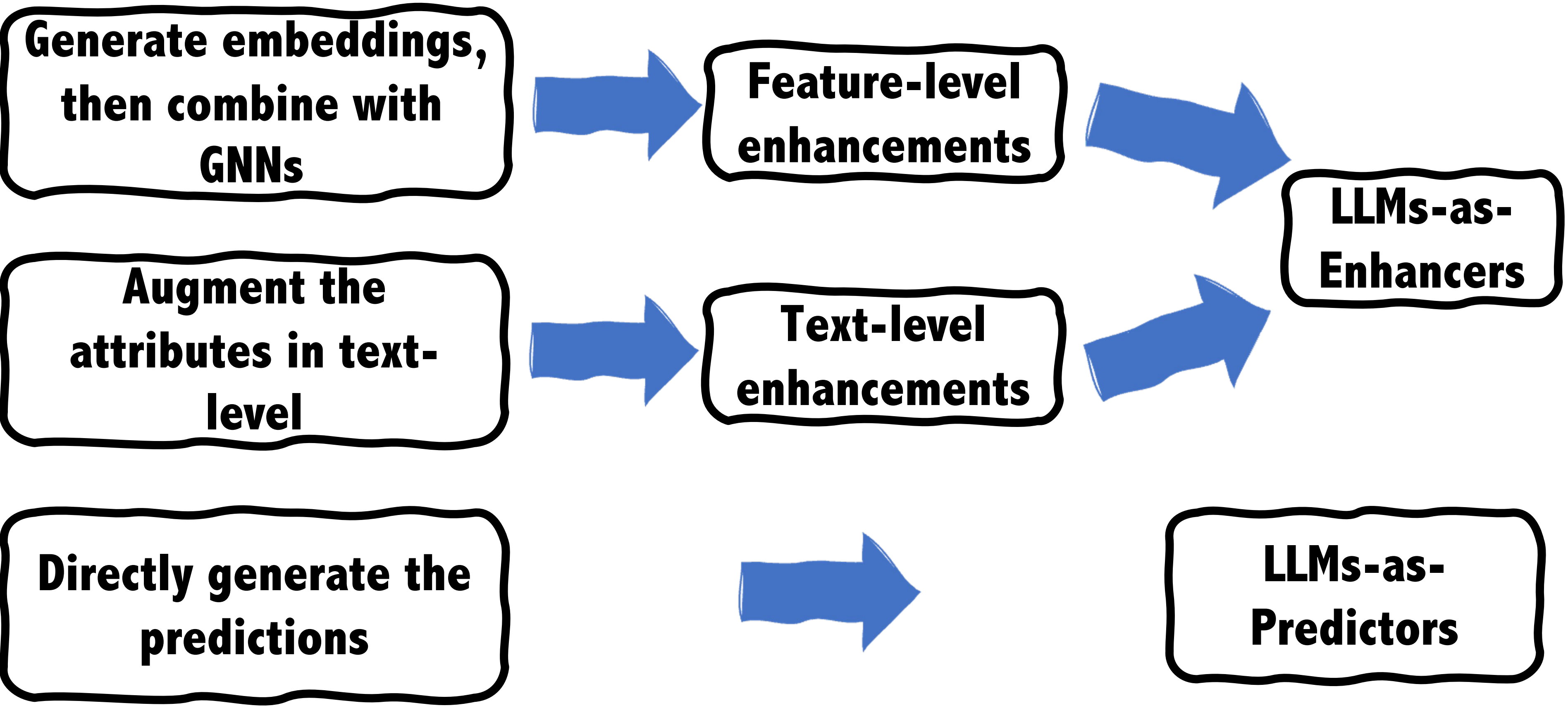
Design pipelines for different models

Embeddings are not accessible

Augment the attributes in text-level

Directly generate the text-formatted predictions

Design pipelines for different models



LLMs-as-Enhancers (feature-level)

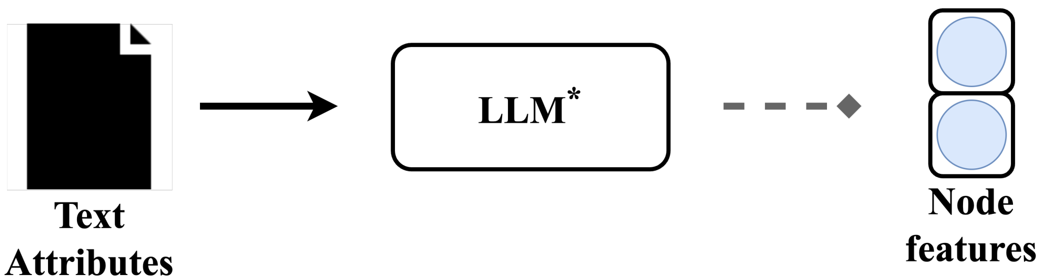
LLMs-as-Enhancers (text-level)

LLMs-as-Predictors

Takeaways and Future Directions

LLMs-as-Enhancers

Feature-level enhancements



LLM* : LLM with accessible embeddings

Settings

Cora, Pubmed

Low labeling rate

20 nodes per class for training, 500 for validation, and 1000 for test

High Labeling rate 60%/20%/20% split

OGB-Arxiv, OGB-Products → Official data splits

We adopt node classification as the downstream tasks to evaluate different strategies

Feature-level enhancements

Selection of GNNs

Selection of LLMs

Selection of integration strategies

Selection of GNNs

For Cora and Pubmed

GCN

GAT

MLP

For OGB-Arxiv

GCN

SOTA
RevGAT

MLP

For OGB-Products

SAGE

SOTA
SAGN

MLP

Selection of LLMs

We aim to check the influence of different textual embeddings

Shallow text embeddings

TF-IDF, Word2vec

Pretrained LM embeddings

- No tuning
- Tuning on attributes
- Tuning on structures

Deberta-base

GIANT

(ICLR 2022)

Deep sentence embeddings

S-BERT

e5

Text-ada-embedding-002 (OpenAI)

Palm Cortex 001 (Google)

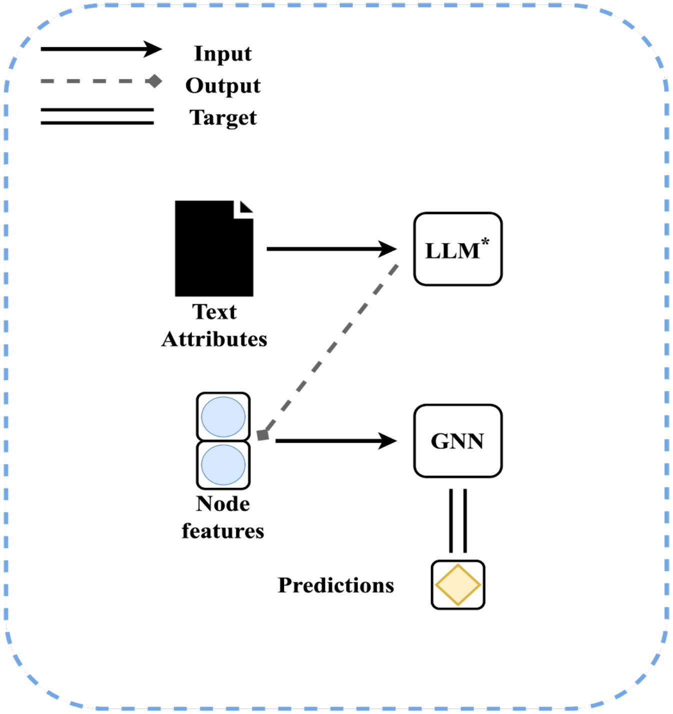
Open-source LLM embeddings

LLaMA

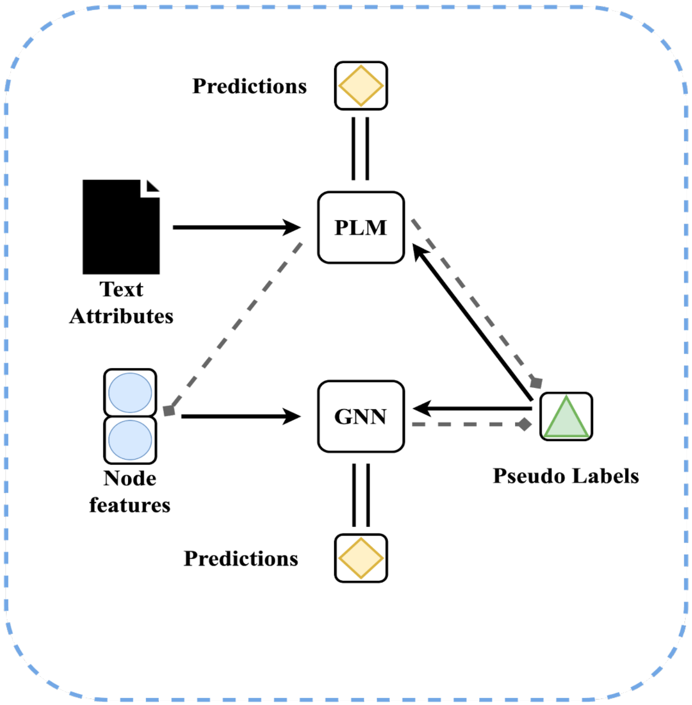
Selection of integration strategies

LLM* : LLM with accessible embeddings

PLM: small-scale PLMs that can be fine-tuned on downstream tasks



Cascading structures



Iterative structures
(GLEM, ICLR 2023)

Observation 1

Table 3: Experimental results for feature-level *LLMs-as-Enhancers* on OGBN-ARXIV and OGBN-PRODUCTS dataset. MLPs do not provide structural information so it's meaningless to co-train it with PLM, thus we don't show the performance. We use yellow to denote the best performance under a specific GNN/MLP model, green the second best one, and pink the third best one.

		OGBN-ARXIV				OGBN-PRODUCTS			
		GCN	MLP	RevGAT	Rank	SAGE	SAGN	MLP	Rank
Cascading Structure	**Non-contextualized Shallow Embeddings**								
	TF-IDF	72.23 ± 0.21	66.60 ± 0.25	75.16 ± 0.14	8	79.73 ± 0.48	84.40 ± 0.07	64.42 ± 0.18	7
	Word2Vec	71.74 ± 0.29	55.50 ± 0.23	73.78 ± 0.19	9	81.33 ± 0.79	84.12 ± 0.18	69.27 ± 0.54	8
	PLM/LLM Embeddings without Fine-tuning								
	Deberta-base	45.70 ± 5.59	40.33 ± 4.53	71.20 ± 0.48	10	62.03 ± 8.82	74.90 ± 0.48	7.18 ± 1.09	10
	Local Sentence Embedding Models								
	Sentence-BERT(MiniLM)	73.10 ± 0.25	71.62 ± 0.10	76.94 ± 0.11	2	82.51 ± 0.53	84.79 ± 0.23	72.73 ± 0.34	6
	e5-large	73.74 ± 0.12	72.75 ± 0.00	76.59 ± 0.44	4	82.46 ± 0.91	85.47 ± 0.21	77.49 ± 0.29	3
	Online Sentence Embedding Models								
	text-ada-embedding-002	72.76 ± 0.23	72.17 ± 0.00	76.64 ± 0.20	3	82.90 ± 0.42	85.20 ± 0.19	76.42 ± 0.31	4
	Fine-tuned PLM Embeddings								
	Fine-tuned Deberta-base	74.65 ± 0.12	72.90 ± 0.11	75.80 ± 0.39	6	82.15 ± 0.16	84.01 ± 0.05	79.08 ± 0.23	9
	Others								

From shallow embeddings to PLM embeddings, the gain for MLPs is significant while it is limited for GNNs

Observation 2

Sentence embeddings are surprisingly effective

Sentence embedding and GNNs with cascading structures can achieve similar performance to GIANT (require task-specific SSL) and GLEM (require LM-GNN co-training)

		GCN	MLP	RevGAT	Rank	SAGE	SAGN	MLP	Rank
Cascading Structure	**Non-contextualized Shallow Embeddings**								
	TF-IDF	72.23 ± 0.21	66.60 ± 0.25	75.16 ± 0.14	8	79.73 ± 0.48	84.40 ± 0.07	64.42 ± 0.18	7
	Word2Vec	71.74 ± 0.29	55.50 ± 0.23	73.78 ± 0.19	9	81.33 ± 0.79	84.12 ± 0.18	69.27 ± 0.54	8
	PLM/LLM Embeddings without Fine-tuning								
	Deberta-base	45.70 ± 5.59	40.33 ± 4.53	71.20 ± 0.48	10	62.03 ± 8.82	74.90 ± 0.48	7.18 ± 1.09	10
	Local Sentence Embedding Models								
	Sentence-BERT(MiniLM)	73.10 ± 0.25	71.62 ± 0.10	76.94 ± 0.11	2	82.51 ± 0.53	84.79 ± 0.23	72.73 ± 0.34	6
	e5-large	73.74 ± 0.12	72.75 ± 0.00	76.59 ± 0.44	4	82.46 ± 0.91	85.47 ± 0.21	77.49 ± 0.29	3
	Online Sentence Embedding Models								
	text-ada-embedding-002	72.76 ± 0.23	72.17 ± 0.00	76.64 ± 0.20	3	82.90 ± 0.42	85.20 ± 0.19	76.42 ± 0.31	4
	Fine-tuned PLM Embeddings								
	Fine-tuned Deberta-base	74.65 ± 0.12	72.90 ± 0.11	75.80 ± 0.39	6	82.15 ± 0.16	84.01 ± 0.05	79.08 ± 0.23	9
	Others								
	GIANT	73.29 ± 0.10	73.06 ± 0.11	75.90 ± 0.19	5	83.16 ± 0.19	86.67 ± 0.09	79.82 ± 0.07	2
Iterative Structure	GLEM-GNN	75.93 ± 0.19	N/A	76.97 ± 0.19	1	83.16 ± 0.09	87.36 ± 0.07	N/A	1
	GLEM-LM	75.71 ± 0.24	N/A	75.45 ± 0.12	7	81.25 ± 0.15	84.83 ± 0.04	N/A	5

Observation 3

Table 1: Experimental results for feature-level *LLMs-as-Enhancer* on CORA and PUBMED with a low labeling ratio. Since MLPs do not provide structural information, it is meaningless to co-train it with PLM (with their performance shown as N/A). We use yellow to denote the best performance under a specific GNN/MLP model, green the second best one, and pink the third best one.

		CORA				PUBMED			
		GCN	GAT	MLP	Rank	GCN	GAT	MLP	Rank
Cascading Structure	**Non-contextualized Shallow Embeddings**								
	TF-IDF	81.99 ± 0.63	82.30 ± 0.65	67.18 ± 1.01	4	78.86 ± 2.00	77.65 ± 0.91	71.07 ± 0.78	5
	Word2Vec	74.01 ± 1.24	72.32 ± 0.17	55.34 ± 1.31	6	70.10 ± 1.80	69.30 ± 0.66	63.48 ± 0.54	7
	PLM/LLM Embeddings without Fine-tuning								
	Deberta-base	48.49 ± 1.86	51.02 ± 1.22	30.40 ± 0.57	10	62.08 ± 0.06	62.63 ± 0.27	53.50 ± 0.43	10
	LLama 7B	66.80 ± 2.20	59.74 ± 1.53	52.88 ± 1.96	7	73.53 ± 0.06	67.52 ± 0.07	66.07 ± 0.56	6
	Local Sentence Embedding Models								
	Sentence-BERT(MiniLM)	82.20 ± 0.49	82.77 ± 0.59	74.26 ± 1.44	2	81.01 ± 1.32	79.08 ± 0.07	76.66 ± 0.50	2
	e5-large	82.56 ± 0.73	81.62 ± 1.09	74.26 ± 0.93	4	82.63 ± 1.13	79.67 ± 0.80	80.38 ± 1.94	1
	Online Sentence Embedding Models								
	text-ada-embedding-002	82.72 ± 0.69	82.51 ± 0.86	73.15 ± 0.89	3	79.09 ± 1.51	80.27 ± 0.41	78.03 ± 1.02	4
	Google Palm Cortex 001	81.15 ± 1.01	82.79 ± 0.41	69.51 ± 0.83	1	80.91 ± 0.19	80.72 ± 0.33	78.93 ± 0.90	3
	Fine-tuned PLM Embeddings								
	Fine-tuned Deberta-base	59.23 ± 1.16	57.38 ± 2.01	30.98 ± 0.68	8	62.12 ± 0.07	61.57 ± 0.07	53.65 ± 0.26	8
Iterative Structure	GLEM-GNN	48.49 ± 1.86	51.02 ± 1.22	N/A	11	62.08 ± 0.06	62.63 ± 0.27	N/A	11
	GLEM-LM	59.23 ± 1.16	57.38 ± 2.01	N/A	9	62.12 ± 0.07	61.57 ± 0.07	N/A	9

Vanilla fine-tuning approaches may not work well in low labeling rates

Observation 4

Sentence embeddings are also effective in low labeling rate

Table 1: Experimental results for feature-level *LLMs-as-Enhancer* on CORA and PUBMED with a low labeling ratio. Since MLPs do not provide structural information, it is meaningless to co-train it with PLM (with their performance shown as N/A). We use yellow to denote the best performance under a specific GNN/MLP model, green the second best one, and pink the third best one.

		CORA				PUBMED			
		GCN	GAT	MLP	Rank	GCN	GAT	MLP	Rank
Cascading Structure	**Non-contextualized Shallow Embeddings**								
	TF-IDF	81.99 ± 0.63	82.30 ± 0.65	67.18 ± 1.01	4	78.86 ± 2.00	77.65 ± 0.91	71.07 ± 0.78	5
	Word2Vec	74.01 ± 1.24	72.32 ± 0.17	55.34 ± 1.31	6	70.10 ± 1.80	69.30 ± 0.66	63.48 ± 0.54	7
	PLM/LLM Embeddings without Fine-tuning								
	Deberta-base	48.49 ± 1.86	51.02 ± 1.22	30.40 ± 0.57	10	62.08 ± 0.06	62.63 ± 0.27	53.50 ± 0.43	10
	LLama 7B	66.80 ± 2.20	59.74 ± 1.53	52.88 ± 1.96	7	73.53 ± 0.06	67.52 ± 0.07	66.07 ± 0.56	6
	Local Sentence Embedding Models								
	Sentence-BERT(MiniLM)	82.20 ± 0.49	82.77 ± 0.59	74.26 ± 1.44	2	81.01 ± 1.32	79.08 ± 0.07	76.66 ± 0.50	2
	e5-large	82.56 ± 0.73	81.62 ± 1.09	74.26 ± 0.93	4	82.63 ± 1.13	79.67 ± 0.80	80.38 ± 1.94	1
	Online Sentence Embedding Models								
	text-ada-embedding-002	82.72 ± 0.69	82.51 ± 0.86	73.15 ± 0.89	3	79.09 ± 1.51	80.27 ± 0.41	78.03 ± 1.02	4
	Google Palm Cortex 001	81.15 ± 1.01	82.79 ± 0.41	69.51 ± 0.83	1	80.91 ± 0.19	80.72 ± 0.33	78.93 ± 0.90	3
	Fine-tuned PLM Embeddings								
	Fine-tuned Deberta-base	59.23 ± 1.16	57.38 ± 2.01	30.98 ± 0.68	8	62.12 ± 0.07	61.57 ± 0.07	53.65 ± 0.26	8
Iterative Structure	GLEM-GNN	48.49 ± 1.86	51.02 ± 1.22	N/A	11	62.08 ± 0.06	62.63 ± 0.27	N/A	11
	GLEM-LM	59.23 ± 1.16	57.38 ± 2.01	N/A	9	62.12 ± 0.07	61.57 ± 0.07	N/A	9

Observation 5

Table 1: Experimental results for feature-level *LLMs-as-Enhancer* on CORA and PUBMED with a low labeling ratio. Since MLPs do not provide structural information, it is meaningless to co-train it with PLM (with their performance shown as N/A). We use yellow to denote the best performance under a specific GNN/MLP model, green the second best one, and pink the third best one.

		CORA				PUBMED			
		GCN	GAT	MLP	Rank	GCN	GAT	MLP	Rank
	Non-contextualized Shallow Embeddings								
	TF-IDF	81.99 ± 0.63	82.30 ± 0.65	67.18 ± 1.01	4	78.86 ± 2.00	77.65 ± 0.91	71.07 ± 0.78	5
	Word2Vec	74.01 ± 1.24	72.32 ± 0.17	55.34 ± 1.31	6	70.10 ± 1.80	69.30 ± 0.66	63.48 ± 0.54	7
	PLM/LLM Embeddings without Fine-tuning								
	Deberta-base	48.49 ± 1.86	51.02 ± 1.22	30.40 ± 0.57	10	62.08 ± 0.06	62.63 ± 0.27	53.50 ± 0.43	10
	LLama 7B	66.80 ± 2.20	59.74 ± 1.53	52.88 ± 1.96	7	73.53 ± 0.06	67.52 ± 0.07	66.07 ± 0.56	6
Cascading Structure	**Local Sentence Embedding Models**								
	Sentence-BERT(MiniLM)	82.20 ± 0.49	82.77 ± 0.59	74.26 ± 1.44	2	81.01 ± 1.32	79.08 ± 0.07	76.66 ± 0.50	2
	e5-large	82.56 ± 0.73	81.62 ± 1.09	74.26 ± 0.93	4	82.63 ± 1.13	79.67 ± 0.80	80.38 ± 1.94	1
	Online Sentence Embedding Models								
	text-ada-embedding-002	82.72 ± 0.69	82.51 ± 0.86	73.15 ± 0.89	3	79.09 ± 1.51	80.27 ± 0.41	78.03 ± 1.02	4
	Google Palm Cortex 001	81.15 ± 1.01	82.79 ± 0.41	69.51 ± 0.83	1	80.91 ± 0.19	80.72 ± 0.33	78.93 ± 0.90	3
	Fine-tuned PLM Embeddings								
	Fine-tuned Deberta-base	59.23 ± 1.16	57.38 ± 2.01	30.98 ± 0.68	8	62.12 ± 0.07	61.57 ± 0.07	53.65 ± 0.26	8
Iterative Structure	GLEM-GNN	48.49 ± 1.86	51.02 ± 1.22	N/A	11	62.08 ± 0.06	62.63 ± 0.27	N/A	11
	GLEM-LM	59.23 ± 1.16	57.38 ± 2.01	N/A	9	62.12 ± 0.07	61.57 ± 0.07	N/A	9

Increasing model size can help, but types of LMs may matter more

Observation 6

Table 3: Experimental results for feature-level *LLMs-as-Enhancers* on OGBN-ARXIV and OGBN-PRODUCTS dataset. MLPs do not provide structural information so it's meaningless to co-train it with PLM, thus we don't show the performance. We use yellow to denote the best performance under a specific GNN/MLP model, green the second best one, and pink the third best one.

		OGBN-ARXIV				OGBN-PRODUCTS			
		GCN	MLP	RevGAT	Rank	SAGE	SAGN	MLP	Rank
Cascading Structure	**Non-contextualized Shallow Embeddings**								
	TF-IDF	72.23 ± 0.21	66.60 ± 0.25	75.16 ± 0.14	8	79.73 ± 0.48	84.40 ± 0.07	64.42 ± 0.18	7
	Word2Vec	71.74 ± 0.29	55.50 ± 0.23	73.78 ± 0.19	9	81.33 ± 0.79	84.12 ± 0.18	69.27 ± 0.54	8
	PLM/LLM Embeddings without Fine-tuning								
	Deberta-base	45.70 ± 5.59	40.33 ± 4.53	71.20 ± 0.48	10	62.03 ± 8.82	74.90 ± 0.48	7.18 ± 1.09	10
	Local Sentence Embedding Models								
	Sentence-BERT(MiniLM)	73.10 ± 0.25	71.62 ± 0.10	76.94 ± 0.11	2	82.51 ± 0.53	84.79 ± 0.23	72.73 ± 0.34	6
	e5-large	73.74 ± 0.12	72.75 ± 0.00	76.59 ± 0.44	4	82.46 ± 0.91	85.47 ± 0.21	77.49 ± 0.29	3
	Online Sentence Embedding Models								
	text-ada-embedding-002	72.76 ± 0.23	72.17 ± 0.00	76.64 ± 0.20	3	82.90 ± 0.42	85.20 ± 0.19	76.42 ± 0.31	4
	Fine-tuned PLM Embeddings								
	Fine-tuned Deberta-base	74.65 ± 0.12	72.90 ± 0.11	75.80 ± 0.39	6	82.15 ± 0.16	84.01 ± 0.05	79.08 ± 0.23	9
	Others								
	GIANT	73.29 ± 0.10	73.06 ± 0.11	75.90 ± 0.19	5	83.16 ± 0.19	86.67 ± 0.09	79.82 ± 0.07	2
Iterative Structure	GLEM-GNN	75.93 ± 0.19	N/A	76.97 ± 0.19	1	83.16 ± 0.09	87.36 ± 0.07	N/A	1
	GLEM-LM	75.71 ± 0.24	N/A	75.45 ± 0.12	7	81.25 ± 0.15	84.83 ± 0.04	N/A	5

OpenAI's embedding models present limited performance gain compared to open-source alternatives

LLMs-as-Enhancers (feature-level)

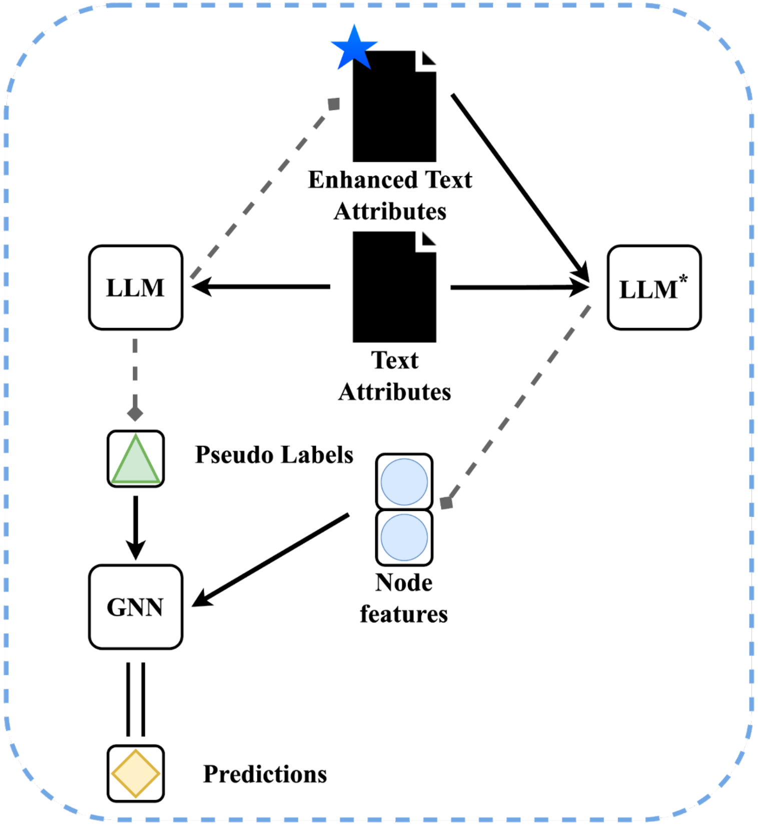
LLMs-as-Enhancers (text-level)

LLMs-as-Predictors

Takeaways and Future Directions

When the embeddings of LLMs are not accessible

Explore them to augment the attributes in the text level.



LLM* : LLM with accessible embeddings

LLM: powerful LLM used to augment the attributes

After augmentation, we further encode the augmented attributes into augmented features

LLMs-as-Enhancers (text-level)

Why may it be effective?

LLMs present a "higher" level of intelligence which may help smaller language models better understand texts

Complex Reasoning

Scenarios need expert knowledge

LLMs-as-Enhancers (text-level)

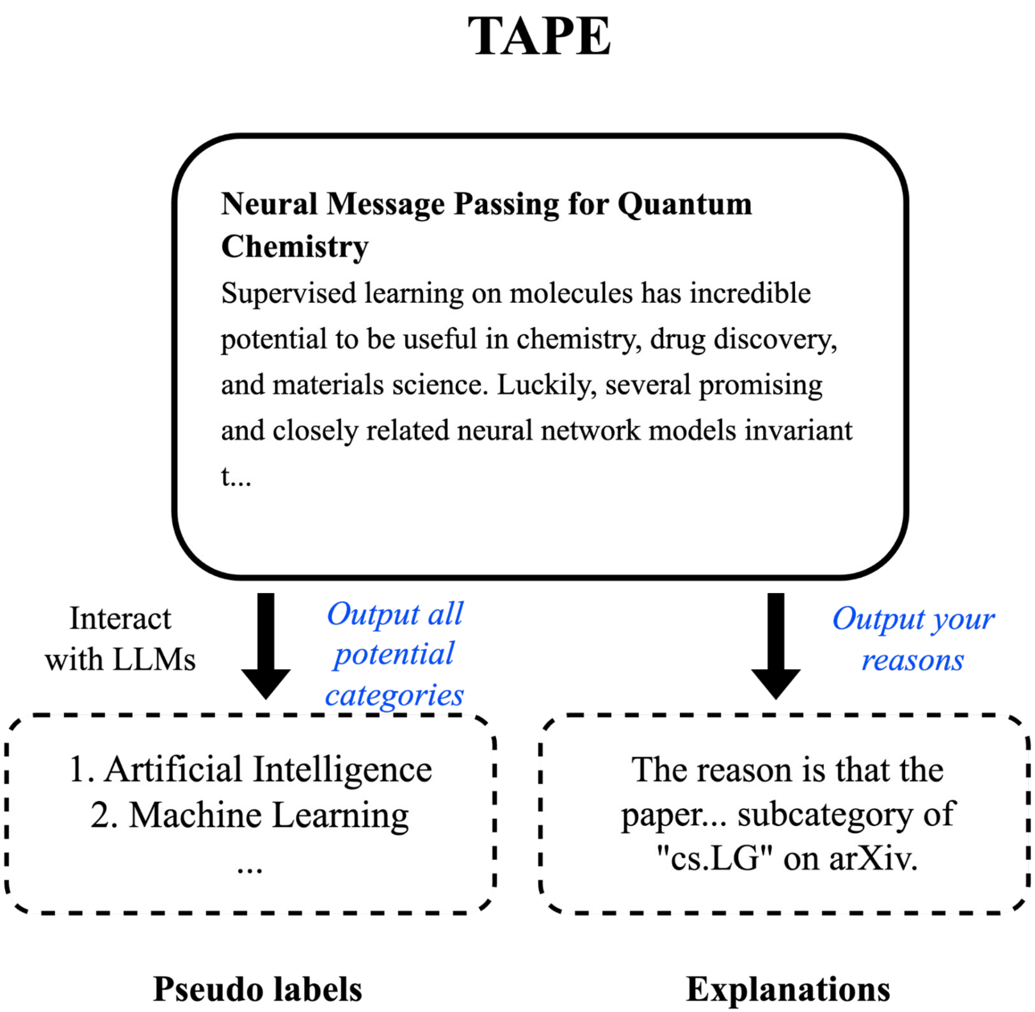
TAPE

Leveraging the knowledge of LLMs to generate predictions and explanations as augmented attributes.

KEA

Leveraging the knowledge of LLMs to extract keywords and generate descriptions as augmented attributes.

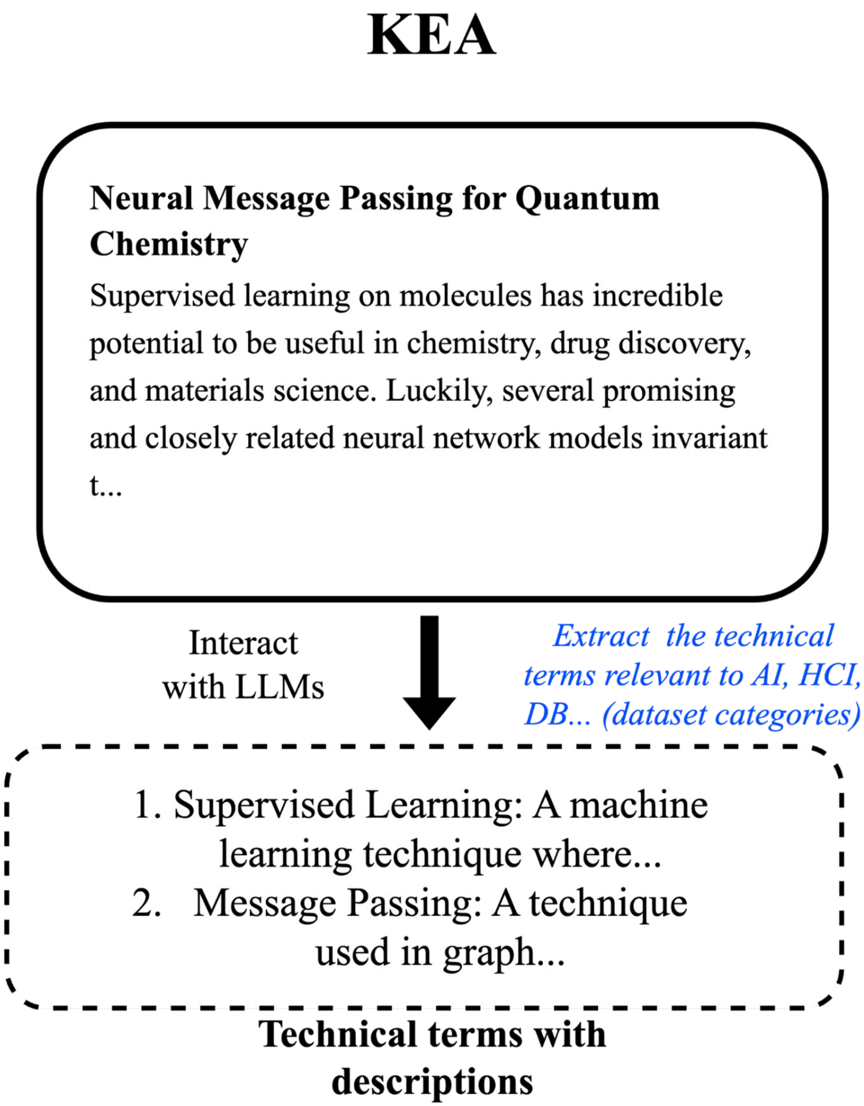
TAPE



1. Prompt the LLMs to generate zero-shot predictions and explanations

2. Encode the augmented attributes info features and do ensembling

KEA



1. Prompt the LLMs to extract domain-specific keywords and generate descriptions

2. Encode the augmented attributes into augmented features and do ensembling

KEA-I: Insert the augmented texts into original attributes, and then encode them together

KEA-S: encode the augmented and original attributes separately

Observation 7

We adopt Cora and Pubmed, and also low/high labeling rate.
We adopt e5 as the encoder, with a cascading structure.
TA refers to the performance of original features

Both methods can achieve performance gain compared to the original attributes

	Cora (low)			Pubmed (low)		
	GCN	**GAT**	**MLP**	**GCN**	**GAT**	**MLP**
TA	82.56 ± 0.73	81.62 ± 1.09	74.26 ± 0.93	82.63 ± 1.13	79.67 ± 0.80	80.38 ± 1.94
KEA-I + TA	83.20 ± 0.56	83.38 ± 0.63	74.34 ± 0.97	83.30 ± 1.75	81.16 ± 0.87	80.74 ± 2.44
KEA-S + TA	84.63 ± 0.58	85.02 ± 0.40	76.11 ± 2.66	82.93 ± 2.38	81.34 ± 1.51	80.74 ± 2.44
TA+E	83.38 ± 0.42	84.00 ± 0.09	75.73 ± 0.53	87.44 ± 0.49	86.71 ± 0.92	90.25 ± 1.56
	Cora (high)			**Pubmed (high)**		
	GCN	**GAT**	**MLP**	**GCN**	**GAT**	**MLP**
TA	90.53 ± 2.33	89.10 ± 3.22	86.19 ± 4.38	89.65 ± 0.85	89.55 ± 1.16	91.39 ± 0.47
KEA-I + TA	91.12 ± 1.76	90.24 ± 2.93	87.88 ± 4.44	90.19 ± 0.83	90.60 ± 1.22	92.12 ± 0.74
KEA-S + TA	91.09 ± 1.78	92.30 ± 1.69	88.95 ± 4.96	90.40 ± 0.92	90.82 ± 1.30	91.78 ± 0.56
TA+E	90.68 ± 2.12	91.86 ± 1.36	87.00 ± 4.83	92.64 ± 1.00	93.35 ± 1.24	94.34 ± 0.86

Observation 8

Best

	Cora (low)		
	GCN	**GAT**	**MLP**
TA	82.56 ± 0.73	81.62 ± 1.09	74.26 ± 0.93
KEA-I + TA	83.20 ± 0.56	83.38 ± 0.63	74.34 ± 0.97
KEA-S + TA	84.63 ± 0.58	85.02 ± 0.40	76.11 ± 2.66
TA+E	83.38 ± 0.42	84.00 ± 0.09	75.73 ± 0.53
	Cora (high)		
	GCN	**GAT**	**MLP**
TA	90.53 ± 2.33	89.10 ± 3.22	86.19 ± 4.38
KEA-I + TA	91.12 ± 1.76	90.24 ± 2.93	87.88 ± 4.44
KEA-S + TA	91.09 ± 1.78	92.30 ± 1.69	88.95 ± 4.96
TA+E	90.68 ± 2.12	91.86 ± 1.36	87.00 ± 4.83

	PubMed (low)		
	GCN	**GAT**	**MLP**
TA	82.63 ± 1.13	79.67 ± 0.80	80.38 ± 1.94
KEA-I + TA	83.30 ± 1.75	81.16 ± 0.87	80.74 ± 2.44
KEA-S + TA	82.93 ± 2.38	81.34 ± 1.51	80.74 ± 2.44
TA+E	87.44 ± 0.49	86.71 ± 0.92	90.25 ± 1.56
	PubMed (high)		
	GCN	**GAT**	**MLP**
TA	89.65 ± 0.85	89.55 ± 1.16	91.39 ± 0.47
KEA-I + TA	90.19 ± 0.83	90.60 ± 1.22	92.12 ± 0.74
KEA-S + TA	90.40 ± 0.92	90.82 ± 1.30	91.78 ± 0.56
TA+E	92.64 ± 1.00	93.35 ± 1.24	94.34 ± 0.86

For different datasets, the most effective enhancement methods may vary

This may be related to LLMs' zero-shot performance on datasets since TAPE generates predictions in the augmented attributes.

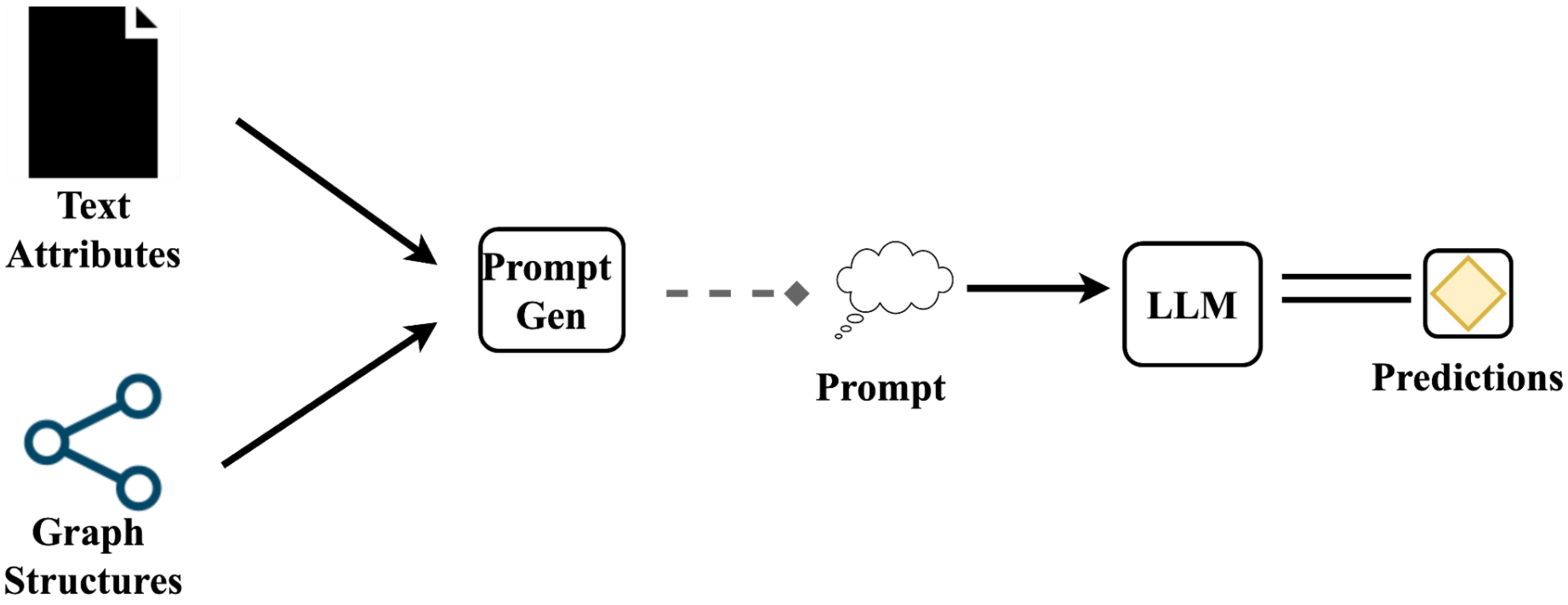
LLMs-as-Enhancers (feature-level)

LLMs-as-Enhancers (text-level)

LLMs-as-Predictors

Takeaways and Future Directions

LLMs-as-Predictors



It's possible to do zero-shot predictions with this pipeline!

Starting point: text classification

By ignoring graph structures, we can view node classification as text classification

Zero-shot Prompts	**Paper:** \n <paper content> \n **Task:** \n There are following categories: \n <list of categories> \n Which category does this paper belong to? \n Output the most 1 possible category of this paper as a python list, like ['XX']
Few-shot Prompts	# Information for the first few-shot samples **Paper:** ... as a python list, like ['XX'] \n [<Ground truth 1>] \n . . . (more few shot samples). . . # Information for the current paper **Paper:** ... category of this paper as a python list, like ['XX']

Does CoT help node classification?

CoT is helpful for reasoning-involved tasks, will it help classification?

Zero-shot prompts with CoT	**Paper:** ... category of this paper as a python list, like ['XX'] \n Think it step by step and output your reason in one sentence.
Few-shot prompts with CoT	# first use zero-shot cot to generate the reasoning process and get CoT process for each few-shot samples # Information for the first few-shot samples **Paper:** ... \n [<Ground truth 1>] \n <CoT process 1> . . . (more few shot samples). . . # Information for this paper **Paper:** ...Think it step by step and output your reason in one sentence.

Experimental Settings

- **Datasets: Cora, Citeseer, Pubmed, OGB-Arxiv, and OGB-Products**
- **We randomly sample 200 nodes from each dataset and repeat the experiment twice.**
- **For LLMs, we adopt either a zero-shot or few-shot setting.**

Observation 9

On some datasets, LLMs' zero-shot performance is close to or even surpasses GNNs'

CoT doesn't show promising gain in this task

	Cora	Citeseer	Pubmed	Ogbn-arxiv	Ogbn-products
Zero-shot	67.00 ± 1.41	65.50 ± 3.53	90.75 ± 5.30	51.75 ± 3.89	70.75 ± 2.48
Few-shot	67.75 ± 3.53	66.00 ± 5.66	85.50 ± 2.80	50.25 ± 1.06	77.75 ± 1.06
Zero-shot with COT	64.00 ± 0.71	66.50 ± 2.82	86.25 ± 3.29	50.50 ± 1.41	71.25 ± 1.06
Few-shot with COT	64.00 ± 1.41	60.50 ± 4.94	85.50 ± 4.94	47.25 ± 2.47	73.25 ± 1.77
GCN/SAGE	82.20 ± 0.49	71.19 ± 1.10	81.01 ± 1.32	73.10 ± 0.25	82.51 ± 0.53

For Cora and Pubmed, we set the performance of GCN in the low labeling rate(20 nodes per class for training, 500 for validation, and 1000 for test) as the baseline.

Observation 10

For some samples, multiple labels seem reasonable from commonsense knowledge

Paper: The Neural Network House: An overview; Typical home comfort systems utilize only rudimentary forms of energy management and conservation. The most sophisticated technology in common use today is an automatic setback thermostat. Tremendous potential remains for improving the efficiency of electric and gas usage...
Ground Truth: Reinforcement Learning
LLM's Prediction: Neural Networks

For these datasets, there's semantic overlap between different labels (many papers are interdisciplinary)

Is the widely adopted single-label setting reasonable here?

Observation 11

40 Arxiv categories

TAPE achieves superior performance on Arxiv with a special

Strategy 1

Abstract: <abstract text> \n Title: <title text>\n There are following categories: "arxiv cs NA, arxiv cs CV, …" What category does this paper belong to …

The difference between this prompt and normal ones are "label names"

Natural languages

Strategy 2

Abstract: <abstract text> \n Title: <title text>\n There are following categories: "Numerical Analysis, Computer vision…" What category does this paper belong to …

TAPE

Strategy 3

Abstract: <abstract text> \n Title: <title text> \n Question: Which arXiv CS sub-category does this paper belong to? Give 5 likely arXiv CS sub-categories as a comma-separated list ordered from most to least likely, in the form "cs.XX", and provide your reasoning. \n \n Answer:

Table 14: Performance of LLMs on OGB-Arxiv dataset, with three different label designs.

	Strategy 1	Strategy 2	Strategy 3
OGBN-ARXIV	48.5	51.8	74.5

What's reason of this phenomenon? Probably different prompts have different effects on the memorization of LLMs

Incorporating neighboring information

How to include neighborhood information in the prompt?

Prompts used to summarize the neighboring information
The following list records some papers related to the current one.
Lists of samples neighboring nodes
The "category" column is optional, and we find it presents little influence on the generated summary
[{ "content": "Cadabra a field theory motivated ...", "category": "computer vision"... }, ...]
 # Instruction
Please summarize the information above with a short paragraph, find some common points which can reflect the category of this paper

One potential solution: <u>Summarization</u>
Trying to simulate the aggregation operation of GNNs

Observation 12

	Cora	Citeseer	Pubmed	Ogbn-arxiv	Ogbn-products
Zero-shot	67.00 ± 1.41	65.50 ± 3.53	90.75 ± 5.30	51.75 ± 3.89	70.75 ± 2.48
Few-shot	67.75 ± 3.53	66.00 ± 5.66	85.50 ± 2.80	50.25 ± 1.06	77.75 ± 1.06
Zero-Shot with 2-hop info	71.75 ± 0.35	62.00 ± 1.41	88.00 ± 1.41	55.00 ± 2.83	75.25 ± 3.53
Few-Shot with 2-hop info	74.00 ± 4.24	67.00 ± 4.94	79.25 ± 6.71	52.25 ± 3.18	76.00 ± 2.82
GCN/SAGE	82.20 ± 0.49	71.19 ± 1.10	81.01 ± 1.32	73.10 ± 0.25	82.51 ± 0.53

By incorporating neighborhood information, we can get performance gain on most datasets

Why is Pubmed an exception?

Why is Pubmed special?

Table 24: An illustrative example for PUBMED

Title: Predictive power of sequential measures of albuminuria for progression to ESRD or death in Pima Indians with **type 2 diabetes**. ... (content omitted here) **Ground truth label:** Diabetes Mellitus Type 2

For Pubmed, it's common that ground truth directly appears in the text attributes

Observation 13

LLMs with a structure-aware prompt may also suffer from heterophilous neighboring nodes.

Table 18: GNNs and LLMs with structure-aware prompts are both wrong

Paper: Title: C-reactive protein and incident cardiovascular events among men with diabetes.
Abstract: OBJECTIVE: Several large prospective studies have shown that baseline levels of C-reactive protein (CRP) are an independent predictor of cardiovascular events among apparently healthy individuals. However, prospective data on whether CRP predicts cardiovascular events in diabetic patients are limited so far. RESEARCH DESIGN AND METHODS ...
Neighbor Summary: This paper focuses on different aspects of **type 2 diabetes** mellitus. It explores the levels of various markers such as tumor necrosis factor-alpha, interleukin-2 ...
Ground truth: "Diabetes Mellitus Type 1"
Structure-ignorant prompts: "Diabetes Mellitus Type 1"
Structure-aware prompt: "Diabetes Mellitus Type 2"
GNN: "Diabetes Mellitus Type 2"

LLMs as Annotators

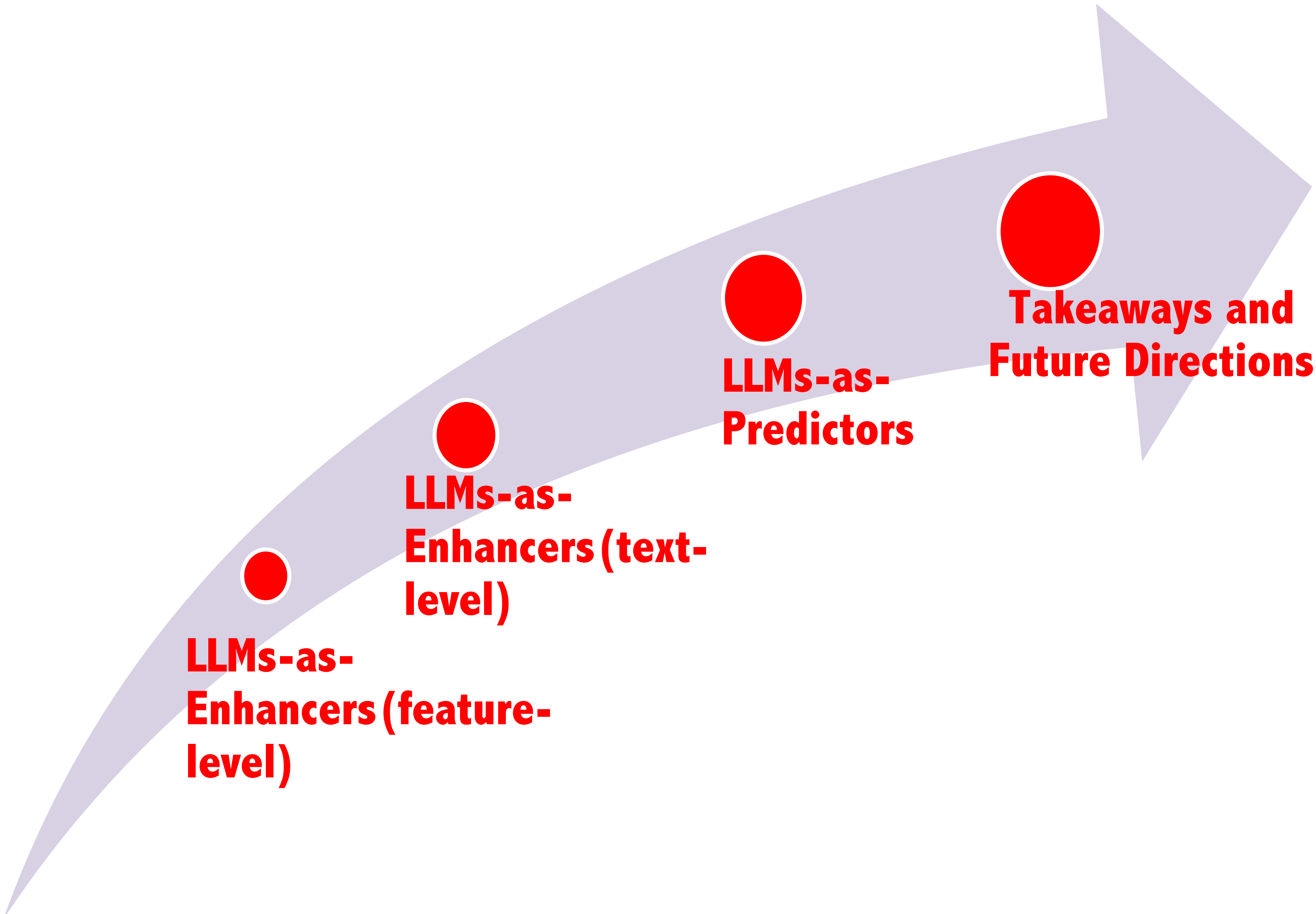
LLMs' effectiveness on zero-shot learning inspire their potential as annotators!

	CORA	PUBMED
Using pseudo labels		
20 shots × #class	64.95 ± 0.98	71.70 ± 1.06
Using ground truth		
3 shots per class	52.63 ± 1.46	59.35 ± 2.67
5 shots per class	58.97 ± 1.41	65.98 ± 0.74
10 shots per class	69.87 ± 2.27	71.51 ± 0.77

Setting: initially all unlabeled nodes, randomly select some nodes to be annotated. 75% for train, 25% for validation.

This presents two novel challenges

1. **How to select informative nodes based on the graph's information**
2. **How to select confident nodes of LLMs to generate high-quality annotations?**


LLMs-as-Enhancers (feature-level)
LLMs-as-Enhancers (text-level)
LLMs-as-Predictors
Takeaways and Future Directions

Takeaway messages

1. For LLMs-as-Enhancers, using deep sentence embedding models to generate embeddings for node attributes presents both effectiveness and efficiency.

2. For LLMs-as-Enhancers, utilizing LLMs to augment node attributes at the text level leads to improvements in downstream performance.

3. For LLMs-as-Predictors, LLMs present preliminary effectiveness but we should be careful about their inaccurate predictions and the potential test data leakage problem.

4. LLMs demonstrate the potential to serve as good annotators for labeling nodes given its zero-shot performance.

Future directions

1. Extending the current pipelines to more tasks, such as link prediction and graph classification

Link prediction

How to represent structural features like common neighborhood and Katz index

Graph classification

How to incorporate whole graph information within limited input context length

2.How to improve the efficiency of LLM-involved pipelines, and scale it to larger graphs?

Inference speed

Inference costs

In this paper, we only test on a few sampled nodes because of LLMs' high usage cost

Future directions

3. How to evaluate the performance of LLMs in a more reasonable approach?

Data Contamination

Most datasets may already be included in the pre-training text corpora of LLMs

Single label setting

For datasets like papers, single label setting seems not reasonable to evaluate LLMs

4. Design novel strategies to use LLMs as a more effective annotators

Informative nodes

We should select those nodes which pose larger influence on the graph

Confident nodes

We should select “confident” nodes of LLMs to generate high-quality annotations

Future directions

5. Large models for the graph domain

In this paper, we mainly consider taking the capability of LLMs to solve graph learning problems

uture directions, we can explore more paradigms

How to design? Let's first have a look at the NLP domain

Foundation models specific for the graph domain

Development of foundation models in NLP

What's special about foundation model?

Pre-training on
massive text

Bert (2018)

tasks

Massive training data

T5 (2019)

Unified into
generation tasks

In-c

Capabilities to unify and generalize across tasks

FLAN (2021)

Instruction
tuning

In-context learning ability

RLHF, better
alignment

InstructGPT (2022)

Graph is more difficult

How to define the transferrable unit in the graph and resolve different structural semantics?

How to unify different tasks and make them help with each other?

Massive graph datasets for pre-training, like MAG240M for the paper domain

We don't even have a pre-trained model like BERT yet, which can achieve good performance on various downstream tasks through a unified pre-training task. We may take a different development path from NLP.

Acknowledgements

Thanks for this great opportunity and our funding support from NSF, DARPA, ARO, AERA, and industrial collaborators.