Appendix

A. Datasets

We evaluate SubGCache on three newly constructed datasets: Scene Graph, OAG and DBLP. Existing GraphQA benchmarks (He et al. 2024) typically associate each textual graph with a single query, overlooking the in-batch query setting. To bridge this gap, we adapt and construct datasets that support in-batch queries for graph-based RAG. Table 1 presents the textual graph details and showcases example queries with their answers from all datasets.

- **Scene Graph.** Based on the original Scene Graph dataset from (He et al. 2024), we select a graph with 22 nodes and 147 edges, representing objects, attributes, and relationships within an image. We manually construct 426 queries targeting specific entities or relations, with answers grounded in node or edge attributes. Many of these queries require multi-hop reasoning. The dataset is split into 113/113/200 queries for training, validation, and testing, respectively.
- **OAG.** The original OAG (Zhang et al. 2019; Zhu et al. 2025) is a textual graph with various types of nodes and edges. To adapt it to our setting, we construct a query set by sampling 3,434 link prediction queries, where each query involves predicting the relation type between two entities. The dataset is split into 1,617/1,617/200 for training, validation, and testing, respectively.
- **DBLP.** The original DBLP (Tang et al. 2008) dataset is also a textual graph containing 21,000 nodes and 22,718 edges. Following the same protocol as OAG, we generate a query set by sampling 1,840 link prediction instances, each requiring the model to infer the relation type between a given pair of entities. The set is divided into 552/552/736 queries for training, validation, and testing, respectively.

B. Setup

Baseline models and LLM backbones. We use G-Retriever (He et al. 2024) and GRAG (Hu et al. 2025) as our baselines, and evaluate SubGCache by integrating it as a plug-and-play module during inference, resulting in the variants G-Retriever+SubGCache and GRAG+SubGCache. We primarily adopt Llama-3.2-3B (Dubey et al. 2024) as the backbone LLM, and further test with Llama-2-7B (Touvron et al. 2023), Mistral-7B (Team et al. 2023), and Falcon-7B (Penedo et al. 2023) to assess SubGCache's scalability and robustness across larger LLMs.

Architecture and Configuration. For graph retrieval, we follow the default pipeline of the original baselines (He et al. 2024; Hu et al. 2025): SentenceBERT (Reimers and Gurevych 2019) is used to encode node and edge attributes, as well as queries, for both G-Retriever and GRAG. For G-Retriever and its SubGCache variant, we select the top-k nodes and edges with $k = 3$ and set the edge cost to 0.5. For GRAG and its SubGCache variant, we select the top-k subgraphs with $k = 3$ and include the top-10 entities within two hops. For graph encoding, G-Retriever and its SubGCache variant use a Graph Transformer (Shi et al. 2021), while GRAG and GRAG+SubGCache adopt GAT (Veličković et al. 2018). Both encoders are configured with 4 layers, 4 attention heads per layer, and a hidden dimension aligned with the LLM backbone. The maximum input sequence length is set to 1024, and the number of generated tokens is capped at 32. For clustering, we adopt agglomerative hierarchical clustering with Euclidean distance and determine cluster assignments by cutting the dendrogram at a predefined number of clusters.

Training and Evaluation Protocol. All models are trained using the AdamW optimizer (Loshchilov and Hutter 2017) with a learning rate of 1e-5 and weight decay of 0.05. Training runs for up to 10 epochs with early stopping: a patience of 2 is used for G-Retriever, and 5 for GRAG. Following both baselines (He et al. 2024; Hu et al. 2025), the LLM backbone remains frozen.

During inference, SubGCache is integrated in a plug-and-play manner without modifying any components of the original models. G-Retriever and G-Retriever+SubGCache share the same pretrained G-Retriever model; similarly, GRAG and GRAG+SubGCache share the same pretrained GRAG model. For the main evaluation, we randomly sample 100 test queries from each dataset. All experiments are conducted on two NVIDIA A100-SXM4-40GB GPUs.

C. Metrics

We evaluate all models in the in-batch query setting using four key metrics that jointly assess generation quality and inference efficiency:

- **Accuracy (ACC).** ACC measures the proportion of correctly answered queries, serving as the primary metric for

Dataset	Textual Graph	Question	Answer
Scene Graph	node id,node attr 0,“name: eye glasses; attribute: black; (x,y,w,h): (330, 125, 25, 7)” 1,“name: laptop; (x,y,w,h): (67, 170, 62, 60)” 2,“name: cords; attribute: blue; (x,y,w,h): (0, 182, 110, 109)” 3,“name: windows; (x,y,w,h): (395, 0, 105, 58)” 4,“name: man; (x,y,w,h): (447, 102, 52, 231)” 5,“name: woman; (x,y,w,h): (304, 109, 78, 224)” 6,“name: jeans; (x,y,w,h): (382, 265, 77, 68)” . . . src,edge attr,dst 0,to the right of,21 0,to the left of,4 0,to the left of,8 0,to the right of,19 0,to the left of,18 0,to the left of,20 . . .	What is the color of the cords?	blue
OAG	node id,node attr 0,“name: a dynamic environment for video surveillance” 1,“name: is the writing on the wall for tabletops” 2,“name: university of castilla la mancha” 3,“name: aalborg university copenhagen” 4,“name: queen mary university of london” 5,“name: panayiotis zaphiris” 6,“name: antonietta grasso” . . . src,edge attr,dst 0,written by,963 0,focuses on,967 1,written by,942 1,focuses on,967 1,cites,455 2,has member,895 . . .	How is ”cross cultural understanding of content and interface in the context of e learning systems” connected to ”computer science”?	focuses on
DBLP	node id,node attr 0,“name: Kernelized Non-Euclidean Relational c-Means Algorithms” 1,“name: Advances in Video-Based Biometrics” 2,“name: Deformation-based augmented reality for hepatic surgery” 3,“name: Immersed finite element method for eigenvalue problem” 4,“name: Efficient linear fusion of partial estimators” 5,“name: Irreducible elementary cellular automata found” . . . src,edge attr,dst 0,written by,15549 0,written by,14200 1,cites,2583 1,written by,15168 1,published by,20000 2,written by,15047 . . .	Describe how ’Machine scheduling with job delivery coordination’ relates to ’Chung Yee Lee’.	written by

Table 1: Datasets.

generation quality.

- **Response Time (RT).** RT denotes the total end-to-end latency for each query, measured from the moment the query is submitted to the completion of the full model response. This includes subgraph retrieval, prompt construction, LLM prefill, and token generation.
- **Time to First Token (TTFT).** TTFT measures the time from query submission to the generation of the first output token. It reflects the system’s responsiveness, which is especially important in latency-sensitive applications.
- **Prefill and First Token Time (PFTT).** PFTT isolates the portion of TTFT that corresponds to the LLM’s prefill computation and first-token generation. It directly reflects the effectiveness of KV cache and prompt reuse strategies.

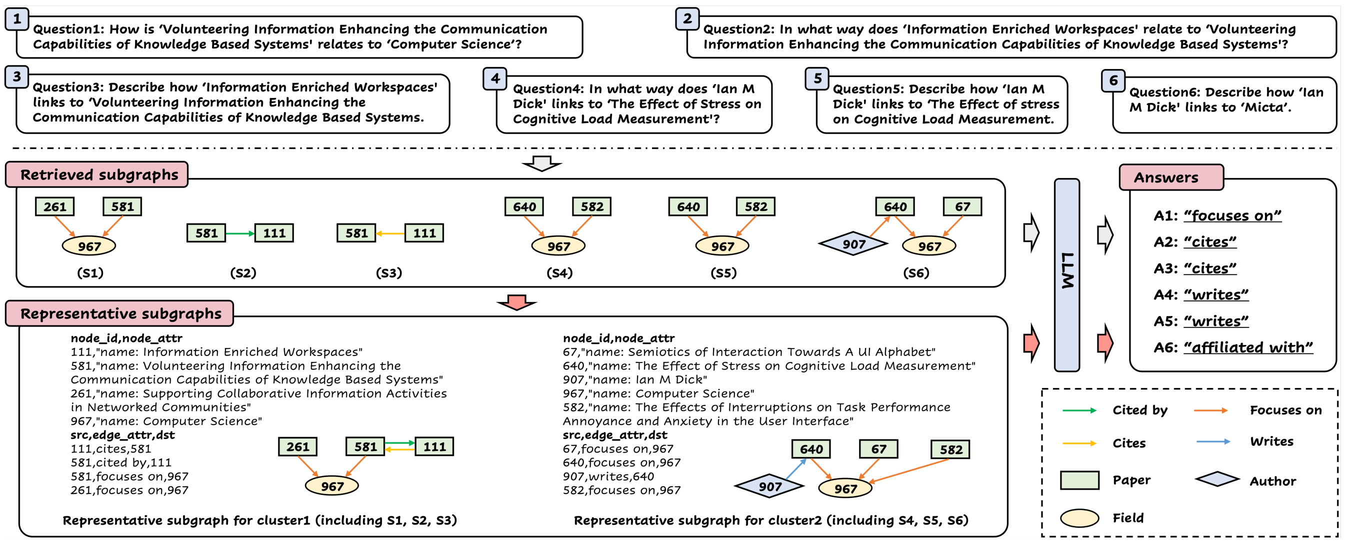
1
Question1: How is 'Volunteering Information Enhancing the Communication Capabilities of Knowledge Based Systems' relates to 'Computer Science'?
2
Question2: In what way does 'Information Enriched Workspaces' relate to 'Volunteering Information Enhancing the Communication Capabilities of Knowledge Based Systems'?
3
Question3: Describe how 'Information Enriched Workspaces' links to 'Volunteering Information Enhancing the Communication Capabilities of Knowledge Based Systems.
4
Question4: In what way does 'Ian M Dick' links to 'The Effect of Stress on Cognitive Load Measurement'?
5
Question5: Describe how 'Ian M Dick' links to 'The Effect of stress on Cognitive Load Measurement.
6
Question6: Describe how 'Ian M Dick' links to 'Micta'.
Retrieved subgraphs
261
581
967
(S1)
581
111
(S2)
581
111
(S3)
640
582
967
(S4)
640
582
967
(S5)
640
67
907
967
(S6)
LLM
Answers
A1: "focuses on"
A2: "cites"
A3: "cites"
A4: "writes"
A5: "writes"
A6: "affiliated with"
Representative subgraphs
node_id,node_attr
111,"name: Information Enriched Workspaces"
581,"name: Volunteering Information Enhancing the Communication Capabilities of Knowledge Based Systems"
261,"name: Supporting Collaborative Information Activities in Networked Communities"
967,"name: Computer Science"
src,edge_attr,dst
111,cites,581
581,cited by,111
581,focuses on,967
261,focuses on,967
261
581
111
967
Representative subgraph for cluster1 (including S1, S2, S3)
node_id,node_attr
67,"name: Semiotics of Interaction Towards A UI Alphabet"
640,"name: The Effect of Stress on Cognitive Load Measurement"
907,"name: Ian M Dick"
967,"name: Computer Science"
582,"name: The Effects of Interruptions on Task Performance Annoyance and Anxiety in the User Interface"
src,edge_attr,dst
67,focuses on,967
640,focuses on,967
907,writes,640
582,focuses on,967
640
67
582
907
967
Representative subgraph for cluster2 (including S4, S5, S6)
Cited by
Focuses on
Cites
Writes
Paper
Author
Field

Figure 1: Case study.

Model	Scene Graph ACC ↑	Scene Graph RT ↓	Scene Graph TTFT ↓	Scene Graph PFTT ↓	OAG ACC ↑	OAG RT ↓	OAG TTFT ↓	OAG PFTT ↓
50 in-batch queries								
G-Retriever	54.00	873.84	843.88	683.25	**96.00**	838.56	716.62	482.61
G-Retriever+SubGCache	**64.00**	**180.98**	**154.04**	**45.95**	**96.00**	**750.40**	**677.68**	**61.64**
$\Delta_{G-Retriever}$	↑ 10.00	↑ 4.83×	↑ 5.48×	↑ 14.87×	0.00	↑ 1.12×	↑ 1.12×	↑ 7.83×
GRAG	**54.00**	1073.04	1041.12	923.91	**100.00**	400.52	327.40	222.16
GRAG+SubGCache	**54.00**	**257.84**	**223.90**	**50.44**	98.00	**225.40**	**187.42**	**59.22**
Δ_{GRAG}	0.00	↑ 4.16×	↑ 4.65×	↑ 18.32×	↓ 2.00	↑ 1.57×	↑ 1.75×	↑ 3.75×
150 in-batch queries								
G-Retriever	57.33	1345.37	1301.90	694.75	**95.33**	773.88	702.56	477.40
G-Retriever+SubGCache	**64.00**	**170.97**	**142.93**	**44.62**	**95.33**	**641.43**	**573.56**	**65.05**
$\Delta_{G-Retriever}$	↑ 6.67	↑ 7.87×	↑ 9.11×	↑ 15.57×	0.00	↑ 1.21×	↑ 1.22×	↑ 7.34×
GRAG	54.00	1199.54	1744.63	923.17	98.67	439.55	374.89	211.83
GRAG+SubGCache	**55.33**	**219.27**	**281.54**	**51.63**	**99.33**	**247.41**	**181.47**	**61.83**
Δ_{GRAG}	↑ 1.33	↑ 5.47×	↑ 6.20×	↑ 17.88×	↑ 0.66	↑ 1.78×	↑ 2.07×	3.43×
200 in-batch queries								
G-Retriever	58.50	827.87	796.95	676.80	**96.50**	699.36	629.36	462.83
G-Retriever+SubGCache	**66.00**	**171.57**	**144.15**	**46.00**	94.50	**631.47**	**562.98**	**62.29**
$\Delta_{G-Retriever}$	↑ 7.50	↑ 4.83×	↑ 5.53×	↑ 14.71×	↓ 2.00	↑ 1.11×	↑ 1.12×	↑ 7.43×
GRAG	**54.50**	1118.74	1085.86	924.04	99.00	420.10	353.33	208.07
GRAG+SubGCache	**54.50**	**216.18**	**182.19**	**51.20**	**99.50**	**240.28**	**172.20**	**61.57**
Δ_{GRAG}	0.00	↑ 5.18×	↑ 5.96×	↑ 18.05×	↑ 0.50	↑ 1.76×	↑ 2.05×	3.38×

Table 2: Effect of different in-batch query size on the Scene Graph and DBLP datasets (Backbone: Llama-2-7B).

D. Case Study

Figure 1 compares how a batch of example queries is processed with and without SubGCache. Without SubGCache, each query is processed separately using its own retrieved subgraph. In contrast, SubGCache clusters similar queries (*i.e.*, q_1–q_3 and q_4–q_6) and constructs a representative subgraph for each cluster, enabling shared KV cache reuse. These representative subgraphs retain all relevant nodes and relations. Both methods generate correct answers, showing that SubGCache significantly accelerates inference without compromising generation quality.

Model	Scene Graph				OAG			
	ACC $\uparrow$	RT $\downarrow$	TTFT $\downarrow$	PFTT $\downarrow$	ACC $\uparrow$	RT $\downarrow$	TTFT $\downarrow$	PFTT $\downarrow$
50 in-batch queries								
G-Retriever	**66.00**	838.20	807.58	716.72	**100.00**	729.71	646.97	511.57
G-Retriever+SubGCache	**66.00**	**210.76**	**181.06**	**49.38**	**100.00**	**317.86**	**229.59**	**62.46**
$\Delta_{G-Retriever}$	0.00	$\uparrow$3.98×	$\uparrow$4.46×	$\uparrow$ 14.51×	0.00	$\uparrow$ 2.30×	$\uparrow$ 2.82×	$\uparrow$8.19×
GRAG	58.00	1113.50	1082.32	967.28	**100.00**	440.92	358.78	248.16
GRAG+SubGCache	**62.00**	**227.32**	**195.08**	**18.80**	**100.00**	**237.78**	**159.26**	**60.43**
Δ_{GRAG}	$\uparrow$ 4.00	$\uparrow$ 4.90×	$\uparrow$ 5.55×	$\uparrow$ 18.80×	0.00	$\uparrow$ 1.85×	$\uparrow$ 2.25×	4.11×
150 in-batch queries								
G-Retriever	**68.00**	1336.57	1290.55	729.47	**99.33**	755.09	677.22	503.59
G-Retriever+SubGCache	**68.00**	**414.58**	**384.83**	**50.00**	**99.33**	**303.50**	**228.99**	**64.13**
$\Delta_{G-Retriever}$	0.00	$\uparrow$ 3.22×	$\uparrow$ 3.35×	$\uparrow$ 14.59×	0.00	$\uparrow$ 2.49×	$\uparrow$ 2.96×	$\uparrow$ 7.85×
GRAG	57.33	1114.03	1623.39	966.18	**99.33**	456.81	379.37	233.84
GRAG+SubGCache	**65.33**	**193.37**	**243.95**	**52.84**	**99.33**	**241.03**	**165.21**	**65.43**
Δ_{GRAG}	$\uparrow$ 8.00	$\uparrow$ 5.76×	$\uparrow$ 6.65×	$\uparrow$ 18.29×	0.00	$\uparrow$ 1.90×	$\uparrow$ 2.30×	3.57×
200 in-batch queries								
G-Retriever	**68.00**	865.71	834.86	712.71	**99.50**	722.10	644.00	489.87
G-Retriever+SubGCache	67.00	**350.64**	**321.06**	**49.57**	**99.50**	**288.64**	**212.21**	**63.24**
$\Delta_{G-Retriever}$	$\downarrow$ 1.00	$\uparrow$ 2.47×	$\uparrow$ .60×	$\uparrow$ 4.38×	0.00	$\uparrow$ 2.50×	$\uparrow$ 3.03×	$\uparrow$ 7.75×
GRAG	54.50	1113.72	1081.63	966.82	**99.50**	442.55	361.28	232.05
GRAG+SubGCache	**65.00**	**199.61**	**169.05**	**18.52**	**99.50**	**244.99**	**167.18**	**63.28**
Δ_{GRAG}	$\uparrow$ 10.50	$\uparrow$ 5.58×	$\uparrow$ 6.40×	$\uparrow$ 18.52×	0.00	$\uparrow$ 1.81×	$\uparrow$ 2.16×	3.67×

Table 3: Effect of different in-batch query size on the Scene Graph and DBLP datasets (Backbone: Mistral-7B).

E. Evaluation Across Different In-batch Query Sizes with Other LLM Backbones

To further assess the scalability of SubGCache across different LLM backbones, we conduct additional experiments using Llama-2-7B, Mistral-7B, and Falcon-7B. Following the same setup as in Table 2 in the main body, we test SubGCache with 50, 100 (from Table 1 in the main body), 150 and 200 in-batch queries on both datasets. The results are presented in Table 2, Table 3 and Table 4, respectively. Consistent trends are observed across different models and in-batch sizes: SubGCache significantly reduces inference latency while maintaining or even improving generation quality. These observations further validate its generalizability and effectiveness across different LLM backbones and in-batch settings, consistent with the findings discussed in the main body.

F. Choice of Linkage Strategy

To assess the sensitivity of SubGCache to clustering choices, we test five standard linkage strategies: Ward, Single, Average, Complete, and Centroid. As shown in Table 5, SubGCache consistently achieves substantial latency reduction in RT, TTFT, and PFTT, while maintaining comparable accuracy across all strategies. This confirms that SubGCache is robust and flexible to the clustering methods and performs reliably across diverse linkage strategies.

G. Evaluation on DBLP Dataset

Table 6 reports the results on the DBLP dataset using four different LLM backbones with a batch size of 100, following the same experimental setup as the main results on the Scene Graph and OAG datasets in the main body. Across all backbones, we observe consistent trends: SubGCache substantially reduces latency while maintaining comparable generation quality, regardless of the architectural type or model scales. Specifically, for G-Retriever, SubGCache achieves up to 4.84× reduction in RT, 6.07× speedup in TTFT, and 11.33× reduction in PFTT on the DBLP dataset. For GRAG, it yields 2.56× reduction in RT, 3.00× speedup in TTFT, and 3.92× reduction in PFTT. These observations demonstrate the robustness and generalizability of SubGCache as a plug-and-play framework.

References

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The Llama 3 Herd of Models.

He, X.; Tian, Y.; Sun, Y.; Chawla, N.; Laurent, T.; LeCun, Y.; Bresson, X.; and Hooi, B. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. volume 37, 132876–132907.

Hu, Y.; Lei, Z.; Zhang, Z.; Pan, B.; Ling, C.; and Zhao, L. 2025. GRAG: Graph Retrieval-Augmented Generation. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Findings*

Model	Scene Graph				OAG			
	ACC ↑	RT ↓	TTFT ↓	PFTT ↓	ACC ↑	RT ↓	TTFT ↓	PFTT ↓
50 in-batch queries								
G-Retriever	**62.00**	913.80	879.24	672.87	**100.00**	709.12	628.12	471.46
G-Retriever+SubGCache	**62.00**	**289.08**	**253.38**	**53.47**	**100.00**	**258.96**	**176.90**	**56.43**
$\Delta_{G-Retriever}$	0.00	↑ 3.16×	↑ 3.47×	↑ 12.58×	0.00	↑ 2.74×	↑ 3.55×	↑ 8.35×
GRAG	**56.00**	1101.98	1065.00	953.93	**100.00**	433.24	346.60	201.20
GRAG+SubGCache	**56.00**	**243.54**	**209.48**	**54.06**	**100.00**	**273.54**	**191.42**	**57.75**
Δ_{GRAG}	0.00	↑ 4.52×	↑ 5.08×	↑ 17.65×	0.00	↑ 1.58×	↑ 1.81×	3.48×
150 in-batch queries								
G-Retriever	65.33	1027.01	1311.02	684.68	**98.00**	710.75	633.24	472.24
G-Retriever+SubGCache	**69.33**	**208.33**	**174.15**	**51.03**	97.33	**253.10**	**173.18**	**59.34**
$\Delta_{G-Retriever}$	↑ 4.00	↑ 4.93×	↑ 7.53×	↑ 13.42×	↓ 0.67	↑ 2.81×	↑ 3.66×	↑ 7.96×
GRAG	56.67	1122.63	1628.99	956.46	**97.33**	473.41	391.93	193.60
GRAG+SubGCache	**60.00**	**250.13**	**324.87**	**52.19**	**97.33**	**258.30**	**179.96**	**60.87**
Δ_{GRAG}	↑ 3.33	↑ 4.49×	↑ 5.01×	↑ 18.33×	0.00	↑ 1.83×	↑ 2.18×	3.18×
200 in-batch queries								
G-Retriever	65.50	825.75	789.16	669.30	**98.50**	687.32	607.52	459.11
G-Retriever+SubGCache	**68.50**	**186.71**	**153.70**	**49.89**	**98.50**	**626.26**	**544.64**	**61.53**
$\Delta_{G-Retriever}$	↑ 3.00	↑ 4.42×	↑ 5.13×	↑ 13.42×	0.00	↑ 1.10×	↑ 1.12×	↑ 7.46×
GRAG	57.50	1121.91	1082.84	958.31	**98.00**	454.89	371.87	192.12
GRAG+SubGCache	**59.50**	**226.08**	**192.64**	**53.22**	**98.00**	**235.13**	**155.99**	**56.96**
Δ_{GRAG}	↑ 2.00	↑ 4.96×	↑ 5.62×	↑ 18.01×	0.00	↑ 1.93×	↑ 2.38×	3.37×

Table 4: Effect of different in-batch query size on the Scene Graph and DBLP datasets (Backbone: Falcon-7B).

	Strategies	Scene Graph				OAG			
		ΔACC	ΔRT	ΔTTFT	ΔPFTT	ΔACC	ΔRT	ΔTTFT	ΔPFTT
$\Delta_{G-Retriever}$	Ward	**↑ 2.00**	**↑ 5.00×**	**↑ 5.69×**	↑ 11.93×	↑ 1.00	**↑ 5.11×**	**↑ 6.52×**	**↑ 8.19×**
	Single	**↑ 2.00**	↑ 4.82×	↑ 5.55×	↑ 12.33×	↑ 1.00	↑ 3.55×	↑ 4.12×	↑ 8.07×
	Average	**↑ 2.00**	↑ 4.86×	↑ 5.60×	↑ 12.56×	**↑ 2.00**	↑ 4.48×	↑ 5.71×	↑ 8.12×
	Complete	**↑ 2.00**	↑ 4.85×	↑ 5.59×	↑ 12.30×	↓ 1.00	↑ 2.64×	↑ 2.88×	↑ 7.29×
	Centroid	**↑ 2.00**	↑ 4.73×	↑ 5.38×	**↑ 12.93×**	↑ 1.00	↑ 2.81×	↑ 3.12×	↑ 7.11×
Δ_{GRAG}	Ward	**↑ 1.00**	↑ 3.37×	↑ 3.84×	↑ 13.77×	**↓ 1.00**	**↑ 1.39×**	**↑ 1.50×**	↑ 2.78×
	Single	**↑ 1.00**	↑ 3.51×	↑ 3.98×	↑ 14.00×	↓ 2.00	↑ 1.30×	↑ 1.39×	**↑ 2.85×**
	Average	**↑ 1.00**	↑ 3.61×	**↑ 4.08×**	**↑ 19.77×**	↓ 4.00	↑ 1.32×	↑ 1.43×	↑ 2.78×
	Complete	**↑ 1.00**	↑ 3.60×	**↑ 4.08×**	↑ 13.41×	**↓ 1.00**	↑ 1.36×	↑ 1.45×	↑ 2.79×
	Centroid	**↑ 1.00**	**↑ 3.64×**	↑ 3.62×	↑ 14.18×	**↓ 1.00**	↑ 1.29×	↑ 1.39×	↑ 2.78×

Table 5: Impact of different linkage strategies.

of the Association for Computational Linguistics: NAACL 2025, 4145–4157. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-195-7.

Loshchilov, I.; and Hutter, F. 2017. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Penedo, G.; Malartic, Q.; Hesslow, D.; Cojocaru, R.; Cappelli, A.; Alobeidli, H.; Pannier, B.; Almazrouei, E.; and Launay, J. 2023. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992.

Shi, Y.; Huang, Z.; Feng, S.; Zhong, H.; Wang, W.; and Sun, Y. 2021. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 1548–1554. International Joint Conferences on Artificial Intelligence Organization.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD interna-*

Model	ACC↑	RT↓	TTFT↓	PFTT↓
Backbone: Llama-3.2-3B				
G-Retriever	48.00	490.63	455.77	339.05
G-Retriever+SubGCache	**51.00**	**170.03**	**131.58**	**38.73**
$\Delta_{G-Retriever}$	↑ 3.00	↑ 2.89×	↑ 3.46×	↑ 8.75×
GRAG	53.00	321.44	285.21	101.64
GRAG+SubGCache	**58.00**	**165.31**	**132.71**	**34.26**
Δ_{GRAG}	↑ 5.00	↑ 1.94×	↑ 2.15×	↑ 2.97×
Backbone: Llama-2-7B				
G-Retriever	45.00	1113.59	1049.74	928.49
G-Retriever+SubGCache	**46.00**	**227.70**	**210.28**	**81.93**
$\Delta_{G-Retriever}$	↑ 1.00	↑ 4.01×	↑ 4.99×	↑ 11.33×
GRAG	48.00	491.43	433.79	280.51
GRAG+SubGCache	**52.00**	**212.31**	**167.49**	**71.53**
Δ_{GRAG}	↑ 4.00	↑ 2.31×	↑ 2.59×	↑ 3.92×
Backbone: Mistral-7B				
G-Retriever	**44.00**	1178.76	1117.27	966.68
G-Retriever+SubGCache	**44.00**	**243.74**	**184.20**	**87.47**
$\Delta_{G-Retriever}$	0.00	↑ 4.84×	↑ 6.07×	↑ 11.05×
GRAG	54.00	475.69	407.32	300.72
GRAG+SubGCache	**55.00**	**255.47**	**188.06**	**81.70**
Δ_{GRAG}	↑ 1.00	↑ 1.86×	↑ 2.17×	↑ 3.68×
Backbone: Falcon-7B				
G-Retriever	58.00	1039.58	971.84	817.97
G-Retriever+SubGCache	**57.00**	**499.12**	**433.48**	**80.50**
$\Delta_{G-Retriever}$	↓ 1.00	↑ 2.08×	↑ 2.24×	↑ 10.16×
GRAG	44.00	564.37	501.86	240.75
GRAG+SubGCache	**43.00**	**220.72**	**167.27**	**71.85**
Δ_{GRAG}	↓ 1.00	↑ 2.56×	↑ 3.00×	↑ 3.35×

Table 6: Evaluation on the DBLP dataset. The best results are highlighted in bold.

tional conference on Knowledge discovery and data mining, 990–998.

Team, M. N.; et al. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Zhang, F.; Liu, X.; Tang, J.; Dong, Y.; Yao, P.; Zhang, J.; Gu, X.; Wang, Y.; Shao, B.; Li, R.; et al. 2019. OAG: Toward linking large-scale heterogeneous entity graphs. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2585–2595.

Zhu, Q.; Zhang, L.; Xu, Q.; and Long, C. 2025. HierPromptLM: A Pure PLM-based Framework for Representation Learning on Heterogeneous Text-rich Networks.