Anchored SubgraphRAG: Structurally Gated Message Passing for Multi-Hop KG-RAG Retrieval

Konstantin Sturtzkopf [*1] **Lorenzo Giusti** **Pietro Liò** [1]

Abstract

Answering multi-hop questions over knowledge graphs requires retrieving structurally coherent evidence, yet most Graph-RAG retrievers score triples in isolation or rely on semantic relevance alone. Adding a Graph Neural Network (GNN) to aggregate neighbourhood information is a natural fix, but recent work reports that doing so can actually degrade retrieval. We argue that this failure is not caused by message passing itself but by uncontrolled semantic aggregation that washes out the contrasts needed to identify relevant triples, an over-smoothing-like effect. We introduce *Anchored SubgraphRAG*, a lightweight retriever that uses query-dependent anchors to localise the relevant graph region and gates message passing with structural positional information. On WebQSP, Anchored SubgraphRAG improves `triple recall@100` from 0.883 to 0.905 ± 0.0095 over the SubgraphRAG baseline. Experiments show that both the GNN and the structural controls are required for this gain, identifying structural gating as a design principle for Graph-RAG retrieval.

1. Introduction

Large Language Models (LLMs) generate fluent text, but when a query requires factual context they do not possess, they often produce plausible yet incorrect answers (Kalai et al., 2025). Supplying the right context at inference time mitigates this failure and unlocks strong performance on tasks ranging from question answering to summarisation and dialogue generation (Ji et al., 2023). The dominant approach is *Retrieval-Augmented Generation* (RAG), which retrieves external knowledge and prepends it to the prompt before the model generates an answer (Shuster et al., 2021; Lewis et al., 2020). Vector-based RAG, however, struggles when an answer depends on reasoning over several connected pieces of evidence. Edge et al. (2025) argue that such queries motivate retrieving from graph-structured knowledge, where the relationships between entities are first-class objects rather than incidental features of a passage (Hogan et al., 2021). Tang & Yang (2024) make the empirical case: existing RAG methods drop sharply on *multi-hop* queries, which by definition require chaining evidence across multiple entities to reach an answer. Graph-RAG is therefore a natural direction, but retrieving the relevant subgraph remains a challenge (Li et al., 2025). Li et al. (2025) recently proposed *SubgraphRAG*, a lightweight retriever that performs strongly overall, yet its accuracy on multi-hop queries still lags its accuracy on single-hop ones. A natural next step is to enrich entity representations with information from their graph neighbourhoods through a Graph Neural Network (GNN). Surprisingly, Li et al. (2025) report that adding such a GNN does not help. We revisit this finding and reach a different conclusion. Adding a GNN *can* improve retrieval, but only when its message passing is **structurally controlled**.

We argue that the failure mode reported by prior work is best understood as an *over-smoothing-like* effect: unconstrained message passing averages representations across semantically heterogeneous neighbourhoods and washes out the contrasts needed to identify relevant triples. Conditioning aggregation on graph position, rather than on semantic similarity alone, attenuates this effect and recovers the benefits of neighbourhood information.

Our contributions are as follows: (i) We frame the degradation of GNN-based KG-RAG retrievers as an over-smoothing-like failure mode caused by uncontrolled semantic aggregation over heterogeneous graph neighbourhoods; (ii) we propose **Anchored SubgraphRAG**, a lightweight retrieval architecture that uses query-dependent anchors and structural positional encodings to gate relational message passing; (iii) we provide preliminary evidence on WebQSP that structurally gated message passing improves triple retrieval over a SubgraphRAG baseline, particularly on multi-hop questions; (iv) through ablations, we show that message passing is beneficial only when paired with structural controls, motivating structural gating as a design principle for Graph-RAG retrieval.

[1]University of Cambridge, Cambridge, United Kingdom. Correspondence to: Konstantin Sturtzkopf <kcs41@cam.ac.uk>.

The Workshop on Graph Foundation Models at the 43rd International Conference on Machine Learning (ICML 2026), Seoul, South Korea. Copyright 2026 by the author(s).

2. Background and Related Works

Let $\mathcal{G} = \{(h, r, t) \mid h, t \in \mathcal{V}, r \in \mathcal{R}\}$ be a knowledge graph, with $\mathcal{V}$ the set of entities and $\mathcal{R}$ the set of relations; each triple (h, r, t) encodes the directed fact that head entity h stands in relation r to tail entity t. We write $\mathcal{N}(v)$ for the 1-hop undirected neighbourhood of v, and $\mathcal{N}^{\rightarrow}(v), \mathcal{N}^{\leftarrow}(v)$ for its out- and in-neighbours; the degree of v is $\deg(v) = |\mathcal{N}(v)|$. For $u, v \in \mathcal{V}$, $d(u, v)$ denotes the length of the shortest undirected path between u and v in $\mathcal{G}$, with $d(u, v) = \infty$ when no such path exists. Each entity carries a learned embedding $\mathbf{x}_v \in \mathbb{R}^d$ and each relation a learned embedding $\mathbf{e}_r \in \mathbb{R}^{d_r}$, while a natural-language question q is mapped by a fixed encoder to a vector $\mathbf{q}_{\text{emb}} \in \mathbb{R}^{d_q}$. We write $\|$ for vector concatenation. A KGQA instance is a tuple $(q, \mathcal{T}, \mathcal{Y}_q)$, where q is a natural-language question, $\mathcal{T} \subseteq \mathcal{V}$ is the set of topic entities that anchor q in $\mathcal{G}$, and $\mathcal{Y}_q \subseteq \mathcal{V}$ is the set of answer entities. The retrieval task is to learn a parametrized scoring function $s_\theta\big(q, (h, r, t)\big)$ that ranks candidate triples in $\mathcal{G}$ by their relevance to q. A question is *multi-hop* when reaching any element of $\mathcal{Y}_q$ from $\mathcal{T}$ requires traversing more than one edge in $\mathcal{G}$.

From RAG to KG-RAG. RAG augments language models with non-parametric memory through dense retrievers (Lewis et al., 2020; Karpukhin et al., 2020; Izacard & Grave, 2021). While effective on single-fact queries, retrieval quality degrades on multi-hop questions whose answers require composing several pieces of evidence (Tang & Yang, 2024). KG-RAG retrieves over structured knowledge instead (Hogan et al., 2021; Edge et al., 2025), and a large body of work prompts the LLM itself to traverse or rewrite the graph at inference time (Sun et al., 2024; Luo et al., 2024; Jiang et al., 2023; Baek et al., 2023; Wu et al., 2023; Gutiérrez et al., 2024; He et al., 2024). We instead take the complementary view of retrieval as a learned ranking problem over candidate triples. SubgraphRAG (Li et al., 2025), our closest baseline, scores each triple with an MLP that consumes a directional distance encoding (DDE) of the endpoints relative to $\mathcal{T}$, and reports that adding GraphSAGE (Hamilton et al., 2017) or Personalized PageRank (Jeh & Widom, 2003) for neighbourhood aggregation *degrades* recall.

GNNs for KGQA and over-smoothing phenomena. GNNs have long been used in KGQA, where iterative propagation over the graph supports multi-hop reasoning (Sun et al., 2018; 2019; Saxena et al., 2020; He et al., 2021; Jiang et al., 2022), and recent KG-RAG systems combine GNN-based candidate selection with LLM reasoning (Mavromatis & Karypis, 2025). Standard backbones include GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017), and the relational variants R-GCN (Schlichtkrull et al., 2018) and CompGCN (Vashishth et al., 2019) that condition each message on its edge label. A well-known failure mode of repeated aggregation is *over-smoothing*, in which node representations converge towards a common fixed point and lose the contrasts needed for discrimination (Li et al., 2018; Oono & Suzuki, 2019). Proposed remedies include feature normalisation (Zhao & Akoglu, 2019), edge dropping (Rong et al., 2019), and gated propagation, where learned gates regulate how node states are updated (Li et al., 2016). In KG-RAG retrieval this effect is particularly pronounced: the candidate subgraph spans many topics, and standard message passing mixes them indiscriminately. Our approach lets graph structure, rather than semantic similarity, determine which neighbours contribute to each update.

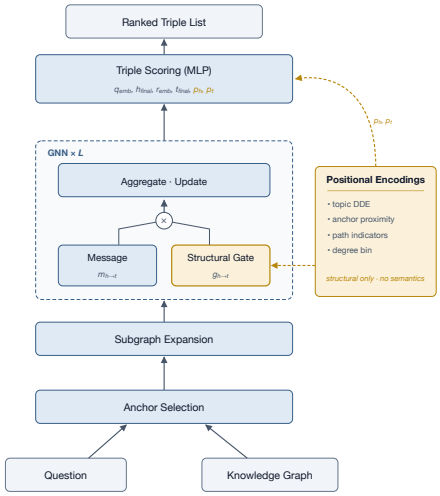

Figure 1. Architecture of Anchored SubgraphRAG. Anchor selection and subgraph expansion narrow the KG to a query-relevant region. For each node we compute structural positional encodings $\mathbf{p}_v$ (amber) from topic distance, anchor proximity, path membership, and degree. These encodings play two roles: inside the GNN they gate message passing through a structure-only gate $g_{h\to t}$, and they feed the final MLP scorer alongside the contextualised entity embeddings. The gate receives no semantic input, which is what lets graph context help rather than diffuse.

3. Structurally Controlled Retrieval

Given a question in natural language q and a knowledge graph $\mathcal{G}$, the task of the retrieval model is to rank candidate triples such that triples $(h, r, t) \in \mathcal{G}$ that are most relevant to answering q receive the highest score. We outline the architecture for structurally controlled retrieval in Figure 1.

Anchor Selection We first select *anchor triples*. We use the same embedding model as SubgraphRAG's reference implementation. We embed all triples as the mean of the component embeddings and calculate the cosine similarity of the triple and query embeddings, to select the top-k triples as anchors. These anchors are, by construction, the triples that are semantically most similar to the question.

Algorithm 1 Structurally gated ASR retrieval pass

Input: question q, topics $\mathcal{T}$, KG $\mathcal{G}$; hops H, anchors k, layers $\mathcal{L}$
Output: candidate triples sorted by relevance
Shorthand: $\mathbf{e}_v := \mathbf{v}_{\text{emb}},\ \bar{\mathbf{x}}_v := \mathbf{x}_v^{(\mathcal{L})}$
$\mathcal{A} \leftarrow$ top-k in $\mathcal{G}$ for cosine-similarity with q
$\mathcal{V}_S \leftarrow \mathrm{BFS}_H(\mathcal{T} \cup \mathrm{ends}(\mathcal{A}))$
$\mathcal{E}_S \leftarrow \{(h,r,t) \in \mathcal{G} : h,t \in \mathcal{V}_S\}$
$\mathbf{p}_v \leftarrow \mathrm{DDE}(v) \,\|\, \mathrm{APE}(v) \,\|\, \mathrm{PI}(v) \,\|\, \mathrm{DB}(v)$
$\mathbf{x}_v^{(0)} \leftarrow \mathbf{e}_v$
for $\ell = 0$ **to** $\mathcal{L}-1$ **do**
  for all $(h,r,t) \in \mathcal{E}_S$ **do**
    $g_{h\to t} \leftarrow \sigma(\mathrm{MLP}([\mathbf{p}_h \,\|\, \mathbf{p}_t]))$ {gate}
    $m_{h\to t}^{(\ell+1)} \leftarrow \sigma(\mathbf{W}_{\text{msg}}[\mathbf{x}_h^{(\ell)} \,\|\, \mathbf{e}_r])$
  end for
  $\bar{\mathbf{m}}_t \leftarrow \frac{1}{|\mathcal{N}(t)|}\sum_{h\in\mathcal{N}(t)} g_{h\to t}\, m_{h\to t}^{(\ell+1)}$
  $\mathbf{x}_t^{(\ell+1)} \leftarrow \sigma(\mathbf{W}_{\text{upd}}[\bar{\mathbf{m}}_t \,\|\, \mathbf{x}_t^{(\ell)}])$
end for
$\mathbf{z} \leftarrow \mathbf{e}_q \,\|\, \bar{\mathbf{x}}_h \,\|\, \mathbf{e}_r \,\|\, \bar{\mathbf{x}}_t \,\|\, \mathbf{p}_h \,\|\, \mathbf{p}_t$
$\mathrm{score}(q,(h,r,t)) \leftarrow \mathrm{MLP}(\mathbf{z})$
return $\mathcal{E}_S$ sorted by score descending

Subgraph Expansion Given the top-k anchor triples and topic entities, we construct a local subgraph in three steps. We form a seed set from all topic entities and the endpoints of anchor triples, expand this set by H-hop BFS (breadth-first-search) over the (undirected) candidate graph, and keep every triple $(h,r,t) \in \mathcal{G}$ whose head and tail both lie in the expanded set.

Positional Encoding: For each node $v \in \mathcal{V}$, we compute a positional encoding $\mathbf{p}_v$ from four structural signals. First, *Topic Directional Distance Encoding (DDE)*, adapted from Li et al. (2025): starting from the one-hot topic indicator $\mathbf{d}_v^{(0)} = \mathbf{1}[v \in \mathcal{T}]$ on the topic set $\mathcal{T}$, we iterate T rounds of mean-aggregation along the directed edges,

$$\mathbf{d}_v^{(\tau+1)} = \frac{1}{|\mathcal{N}^{\to}(v)|} \sum_{u \in \mathcal{N}^{\to}(v)} \mathbf{d}_u^{(\tau)}, \tag{1}$$

and T further rounds along the reversed edges to obtain $\tilde{\mathbf{d}}_v^{(\tau)}$. We concatenate all intermediate states $\{\mathbf{d}_v^{(\tau)}, \tilde{\mathbf{d}}_v^{(\tau)}\}_{\tau=0}^{T}$ to form $\mathrm{DDE}(v)$, which records v's directional distance from the topics in both senses. Next, we compute an *anchor proximity encoding (APE)*: for each anchor $a \in \mathcal{A}$, we take the undirected BFS distance $d(v,a)$, clamp it to a maximum $d_{\max}$, embed it via a learned lookup $\mathbf{E}$, and average across anchors,

$$\mathrm{APE}(v) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \mathbf{E}\big[\min(d(v,a),\, d_{\max})\big]. \tag{2}$$

Lastly, two coarse structural features: a *path indicator* form a binary vector $\mathrm{PI}(v)$ flagging whether v lies on a shortest topic–anchor path, and whether it is a direct neighbour of a topic or an anchor, and a *degree bin* $\mathrm{DB}(v)$ assigns $\deg(v)$ to a one-hot bucket. The full positional encoding concatenates all four signals:

$$\mathbf{p}_v = [\mathrm{DDE}(v) \,\|\, \mathrm{APE}(v) \,\|\, \mathrm{PI}(v) \,\|\, \mathrm{DB}(v)]\,. \tag{3}$$

Graph Neural Network: We refine entity representations of our expanded subgraph with a stack of $\mathcal{L}$ relational gated convolution layers. For each layer $0 \le \ell \le \mathcal{L}-1$ and each directed edge (h,r,t), we compute:

$$m_{h\to t}^{(\ell+1)} = \sigma\Big(\mathbf{W}_{\text{msg}}[\mathbf{x}_h^{(\ell)} \,\|\, \mathbf{e}_r]\Big)\,, \tag{4}$$

where $\mathbf{x}_h$ is the current representation of entity h, σ is a point-wise non-linear activation function (i.e., ReLU), and $\mathbf{e}_r$ is a learned projection of the relation.

Before aggregation, we also scale each message by a scalar gate

$$g_{h\to t} = \sigma\big(\mathrm{MLP}([\mathbf{p}_h \,\|\, \mathbf{p}_t])\big), \tag{5}$$

where $\mathbf{p}_h, \mathbf{p}_t$ are the positional encodings, only encoding structural information calculated for the nodes in the previous step. As such, the gate **receives no semantic input**.

For each layer $0 \le \ell \le \mathcal{L}-1$ in the GNN-Stack, we update each node $\mathbf{t}$

$$\mathbf{x}_t^{(\ell+1)} = \sigma\left(\mathbf{W}_{\text{upd}}\left[\frac{1}{|\mathcal{N}(t)|}\sum_{h\in\mathcal{N}(t)} g_{h\to t}\, m_{h\to t}^{(\ell+1)} \,\Big\|\, \mathbf{x}_t^{(\ell)}\right]\right), \tag{6}$$

where $\mathcal{N}(t)$ is the 1-hop neighbourhood of t, and $\mathbf{x}_t^{(0)} = \mathbf{t}_{\text{emb}}$ is the initial embedding. For each node v, the final embedding we pass to the following layers is $v_{\text{final}} = \mathbf{x}_v^{(\mathcal{L})}$. We refer to this final embedding as the *contextualised embedding* of v.

Triple Scoring: For each candidate triple (h,r,t) and question q, the scorer concatenates the question embedding $\mathbf{q}_{\text{emb}}$, the GNN-contextualised endpoint embeddings $\mathbf{h}_{\text{final}}$ and $\mathbf{t}_{\text{final}}$, the relation embedding $\mathbf{r}_{\text{emb}}$, and the structural positional encodings $\mathbf{p}_h, \mathbf{p}_t$, then passes the result through an MLP:

$$\mathbf{z} = [\mathbf{q}_{\text{emb}} \,\|\, \mathbf{h}_{\text{final}} \,\|\, \mathbf{r}_{\text{emb}} \,\|\, \mathbf{t}_{\text{final}} \,\|\, \mathbf{p}_h \,\|\, \mathbf{p}_t]\,, \tag{7}$$

$$\mathrm{score}\big(q,(h,r,t)\big) = \mathrm{MLP}(\mathbf{z}). \tag{8}$$

Routing $\mathbf{p}_h$ and $\mathbf{p}_t$ to the scorer in addition to the gate gives the model two views of structural position: one filtered through message passing, the other used directly. The full algorithm is detailed in Algorithm 1.

Table 1. Ablation on WebQSP. TR@100 is `triple recall@100` (%); Δ is the change relative to Full ASR.

Ablation	TR@100	Δ	Readout
Full ASR	**90.5**	—	reference
Message passing	85.1	-5.4	graph context load-bearing
Structural PE	87.7	-2.8	structure controls context
Structural gate	89.5	-1.0	gates separate weights, semantics
Path indicators	89.3	-1.2	strongest single cue
Topic DDE	89.8	-0.7	useful with anchors
Anchor proximity	89.8	-0.7	complements topic frame

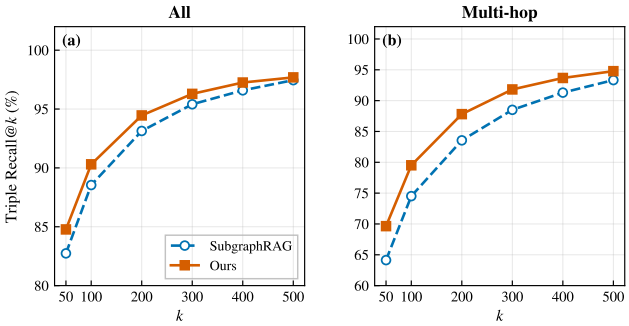

Figure 2. `triple recall@k` on the test set, split by (a) overall and (b) multi-hop performance on the WebQSP (Yih et al., 2016) dataset. Based on one run of SubgraphRAG and our method each.

Training: This model is trained based on weak supervision derived from the graph structure. Triples on any shortest path between a question's topic entities and its known answer entity receive label 1, and we train the model using a binary cross-entropy target.

4. Experiments

We evaluate the performance of our model based on the `triple recall@k` metric, which measures the fraction of ground-truth shortest-path triples that appear among the k triples with the highest score given by the retriever. On three different seeds, our model achieves `triple recall@100` of 0.905 ± 0.0095, which is 2.2 percentage points higher than results reported by Li et al. (2025). Figure 2 shows `triple recall@k` as a function of k and that our method outperforms the baseline for every k and that multi-hop performance benefits the most.

Li et al. (2025) further discuss that multi-hop questions remain difficult for the retrieval model and, as Figure 2b) shows, we see the largest gains for these types of questions. Li et al. (2025) find that inserting a GNN into the retrieval pipeline degrades performance, which they attribute to the diffusion of semantic noise across the graph. Our results suggest that this failure mode is not inherent to graph-based retrieval, but depends on how message passing is controlled.

As shown in Table 1, structurally gated GNN updates are critical to retrieval performance: removing the gate substantially weakens the model.

We select anchors based on semantic similarity to the questions, meaning they might naturally contain answer entities. Li et al. (2025) show that cosine similarity alone is not sufficient for retrieval, but the created path indicators might still correlate with the training labels. However, the anchor selection is dependent on only the question, not the answer entity, making the anchor placement effective.

5. Discussion and Conclusions

A natural reading of prior work is that GNNs simply do not help for knowledge graph retrieval. Our experiments tell a different story. The problem is not that message passing is too weak, but that it is too undiscriminating: when every neighbour gets a vote, the very contrasts that distinguish relevant triples get averaged away. Letting graph structure rather than semantic similarity decide which messages count restores those contrasts, and with them the benefits of neighbourhood information. On WebQSP, Anchored SubgraphRAG raises `triple recall@100` from 0.883 to 0.905. The gains are largest on the multi-hop questions that motivated graph retrieval in the first place. Our ablations show that neither the GNN nor the structural controls do the work alone; they succeed only together, which we take as evidence that structural gating is a useful design principle for Graph-RAG retrieval. Several caveats apply. Our training signal labels only triples on shortest paths as positive, which probably understates the value of message passing on multi-hop questions; a query-aware labelling scheme is a natural next step. We have measured retrieval in isolation, so whether better recall translates into better answers from the downstream language model is still open.

The same question deserves to be asked on harder benchmarks such as CWQ (Talmor & Berant, 2018), with other GNN architectures, and across a wider sweep of hyperparameters before we draw broader conclusions. We finally argue that it is not that graphs solve retrieval, nor that this particular architecture is the right one. It is that graph context becomes useful for retrieval when the graph itself, rather than semantics alone, is allowed to shape how information moves through it. A natural extension is to ask whether the same structural-gating recipe attenuates over-squashing through bottlenecks (Di Giovanni et al., 2023), and how it relates to continuous-time GNN formulations that mitigate over-smoothing by construction. Future advancements of this work will consider provide a quantitative analysis of the dispersion phenomena by applying Algorithm 1 to larger, and possibly synthetic graphs.

Acknowledgements

This work was partially supported by a scholarship from the German Academic Exchange Service (DAAD).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Use of LLMs

To comply with the workshop's policy regarding LLM usage disclosure, we acknowledge partial involvement of LLMs in the following tasks: documentation and literature review, grammar-checking, formatting, and code debugging.

References

Baek, J., Aji, A. F., and Saffari, A. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pp. 78–106, 2023.

Di Giovanni, F., Giusti, L., Barbero, F., Luise, G., Lio, P., and Bronstein, M. M. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *International conference on machine learning*, pp. 7865–7885. PMLR, 2023.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., and Larson, J. From Local to Global: A Graph RAG Approach to Query-Focused Summarization, February 2025. URL http://arxiv.org/abs/2404.16130. arXiv:2404.16130 [cs].

Gutiérrez, B. J., Shu, Y., Gu, Y., Yasunaga, M., and Su, Y. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive Representation Learning on Large Graphs. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

He, G., Lan, Y., Jiang, J., Zhao, W. X., and Wen, J.-R. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 553–561, 2021.

He, X., Tian, Y., Sun, Y., Chawla, N. V., Laurent, T., LeCun, Y., Bresson, X., and Hooi, B. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907, 2024.

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G., Gutierrez, C., Kirrane, S., Gayo, J., Navigli, R., Neumaier, S., Ngomo, A., Polleres, A., Rashid, S., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., and Zimmermann, A. Knowledge graphs. *ACM Computing Surveys*, 54(4), July 2021. ISSN 0360-0300. doi: 10.1145/3447772. Place: New York, NY, USA Publisher: Association for Computing Machinery.

Izacard, G. and Grave, E. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pp. 874–880, 2021.

Jeh, G. and Widom, J. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pp. 271–279, 2003.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Chen, D., Dai, W., Chan, H. S., Madotto, A., and Fung, P. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12): 1–38, December 2023. ISSN 0360-0300, 1557-7341. doi: 10.1145/3571730. URL http://arxiv.org/abs/2202.03629. arXiv:2202.03629 [cs].

Jiang, J., Zhou, K., Zhao, W. X., and Wen, J.-R. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. *arXiv preprint arXiv:2212.00959*, 2022.

Jiang, J., Zhou, K., Dong, Z., Ye, K., Zhao, X., and Wen, J.-R. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9237–9251, 2023.

Kalai, A. T., Nachum, O., Vempala, S. S., and Zhang, E. Why Language Models Hallucinate, September 2025. URL http://arxiv.org/abs/2509.04664. arXiv:2509.04664 [cs].

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. Dense passage retrieval

for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Li, M., Miao, S., and Li, P. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=JvkuZZ04O7.

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. S. Gated Graph Sequence Neural Networks. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.05493.

Luo, L., Li, Y.-F., Haffari, R., and Pan, S. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations*, volume 2024, pp. 14400–14423, 2024.

Mavromatis, C. and Karypis, G. GNN-RAG: Graph neural retrieval for efficient large language model reasoning on knowledge graphs. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 16682–16699, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.856. URL https://aclanthology.org/2025.findings-acl.856/.

Oono, K. and Suzuki, T. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

Rong, Y., Huang, W., Xu, T., and Huang, J. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

Saxena, A., Tripathi, A., and Talukdar, P. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 4498–4507, 2020.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pp. 593–607. Springer, 2018.

Shuster, K., Poff, S., Chen, M., Kiela, D., and Weston, J. Retrieval Augmentation Reduces Hallucination in Conversation, April 2021. URL http://arxiv.org/abs/2104.07567. arXiv:2104.07567 [cs].

Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., and Cohen, W. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 4231–4242, 2018.

Sun, H., Bedrax-Weiss, T., and Cohen, W. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. 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)*, pp. 2380–2390, 2019.

Sun, J., Xu, C., Tang, L., Wang, S., Lin, C., Gong, Y., Ni, L. M., Shum, H.-Y., and Guo, J. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph, March 2024. URL http://arxiv.org/abs/2307.07697. arXiv:2307.07697 [cs].

Talmor, A. and Berant, J. The Web as a Knowledge-base for Answering Complex Questions, March 2018. URL http://arxiv.org/abs/1803.06643. arXiv:1803.06643 [cs].

Tang, Y. and Yang, Y. MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries, January 2024. URL http://arxiv.org/abs/2401.15391. arXiv:2401.15391 [cs].

Vashishth, S., Sanyal, S., Nitin, V., and Talukdar, P. Composition-based multi-relational graph convolutional networks. *arXiv preprint arXiv:1911.03082*, 2019.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Wu, Y., Hu, N., Bi, S., Qi, G., Ren, J., Xie, A., and Song, W. Retrieve-Rewrite-Answer: A KG-to-Text Enhanced LLMs Framework for Knowledge Graph Question Answering, September 2023. URL https://arxiv.org/abs/2309.11206v2.

Yih, W.-t., Richardson, M., Meek, C., Chang, M.-W., and Suh, J. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 201–206, 2016.

Zhao, L. and Akoglu, L. Pairnorm: Tackling oversmoothing in gnns. *arXiv preprint arXiv:1909.12223*, 2019.