When Drug Names Mislead: Auditing Name-Prior Reliance in LLM-Augmented Biomedical Graph Completion

Anonymous Author(s)

Abstract

LLMs are often added to biomedical graph completion as a source of domain knowledge, but their gains can be misleading. The central risk is not simply that LLMs struggle to read graph structure: biomedical names can make a candidate edge appear plausible even without evidence for the target relation. We test this claim on drug–drug interaction prediction with an audit that separates topology, name visibility, semantic scoring, calibration, and the fusion of graph and semantic signals across random, degree-matched, and graph-near candidates. Three results support the diagnosis. A learned graph scorer remains strong under graph-near shift (AUROC/AP = 0.904/0.879), yet a calibrator trained on random candidates assigns high probabilities to plausible negatives (ECE = 0.146). A name-visible LLM scorer reaches AUROC/AP = 0.743/0.760 on graph-near candidates, but a masked-name control yields no ranking signal once identities are removed. Adding this semantic signal to graph-near-trained fusion still reduces AP. These findings suggest that LLM-augmented biomedical graph completion should audit name semantics, topology, and relation evidence separately before claiming complementarity between language and graph signals.

CCS Concepts

• **Computing methodologies** → **Knowledge representation and reasoning**; **Machine learning**; • **Applied computing** → *Bioinformatics*.

Keywords

biomedical knowledge graph completion, large language models, graph learning, name semantics, relation evidence, reliability audit

1 Introduction

LLMs are now commonly used to augment graph learning systems with text-derived priors, instruction-following reasoning, or entity-level semantic scores [13, 17, 28]. Biomedical knowledge graphs are a natural setting for this trend: drug, disease, protein, and gene names often encode biomedical signal, and resources such as DrugBank, Decagon, Hetionet, and PrimeKG make graph completion central to drug interaction and precision-medicine analysis [3, 8, 32, 35].

A large part of recent LLM-for-graph research treats graph structure itself as the primary bottleneck. Prior work asks how graphs should be serialized into language [5], whether LLMs truly use graph structural information in prompts [11], how well they solve explicit graph reasoning tasks [1, 20], and whether graph encoders or graph instruction tuning are needed to interface graphs with LLMs [7, 15, 17, 28, 30]. This diagnosis is reasonable: graph tasks require directionality, neighborhood aggregation, multi-hop constraints, and absence/presence discrimination, all of which can be blurred by a flat text prompt.

We argue that biomedical graph completion introduces a different and under-audited failure mode. The main issue is not always that LLMs cannot read graph structure. Rather, biomedical names can make misleading evidence appear relation-relevant. A drug pair can appear pharmacologically related without forming a clinically relevant interaction; a gene and disease can co-occur without implying causation; protein names can be lexically or functionally close without supporting a specific PPI edge. Biomedical graph completion is relation-grounded discrimination, not topic completion.

This reframes the reliability question. If topology, entity names, and relation evidence disagree, which signal does the system follow? We treat this as an empirical audit rather than a new model proposal, comparing structural graph scores, a DistMult-style embedding scorer, name-only LLM scoring, a biomedical text-encoder proxy, and simple fusion models under increasingly plausible DDI candidate regimes.

This paper makes three contributions:

- We separate three signals that are often conflated in biomedical graph completion: topology use, entity-name semantics, and relation-grounded evidence.
- We define a lightweight audit protocol that compares random, degree-matched, and graph-near candidate regimes, with an additional topology/name-injection control for isolating name-semantics dominance.
- We provide evidence that candidate shift damages calibration, and that even stronger name-visible LLM signals require masking and fusion audits before being treated as relation evidence.

2 Name Semantics vs. Relation Evidence

The topology-centric view is important but incomplete for biomedical graphs. We use *name semantics* to denote surface or corpus-derived associations attached to entity strings: whether two drugs, genes, diseases, or proteins sound related, co-occur in text, or belong to adjacent biomedical topics. We use *relation evidence* to denote support for a specific KG edge under a schema: direction, relation type, mechanism, safety profile, experimental context, or graph-supported interaction pattern.

These two signals can disagree. Drug–drug interaction prediction is not the same as drug-name relatedness. A pair of drugs with similar indications may have no observed interaction, while an interacting pair may be hard to infer from names alone. Thus, an LLM's language advantage can become a source of structured confounding: it can generate plausible biomedical rationales while failing to discriminate whether a specific relation is supported.

Our stronger hypothesis is therefore not merely that LLM–graph complementarity is fragile. It is that the dominant degradation can be induced by semantic exposure rather than topology alone. If an LLM performs above chance on symbolic topology controls but degrades when real or misleading biomedical names are injected

into the same structural context, the failure is better explained as name-semantics dominance than as pure graph unreadability.

3 Related Work and Positioning

LLM–graph evaluation. Recent graph–LLM work studies whether language models can use graph structure from prompts and serialized graph descriptions [5, 11]. Other systems connect LLMs with graph encoders or instruction-tuned graph interfaces [17, 20, 28]. Closest to our diagnostic stance, Huang et al. ask when prompted LLMs exploit graph structure rather than textual context [11]. Workshop-style evaluation papers also emphasize targeted protocol design: LLM-Hype evaluates taxonomy reasoning across datasets, prompts, and models [12]; Mihindukulasooriya et al. vary prompt optimizers, execution models, schema complexity, and datasets for knowledge graph construction [21]. We follow this evaluation-first framing, but focus on biomedical graph completion, where entity names can become a confounder.

Architectural systems such as GraphGPT and LLaGA ask how to make graph information legible to LLMs through encoders, projectors, or graph instruction tuning [17, 28]. Our design is intentionally different: it is not a new graph–LLM pipeline, but a confounding audit. We hold the candidate pair fixed while varying topology difficulty, name visibility, and scorer visibility, so that improvements from name priors are not mistaken for relation-grounded reasoning.

KGC evaluation and negative candidates. Knowledge graph completion has a long history, from surveys of knowledge graphs and embedding methods to learned models such as TransE, DistMult, ComplEx, RotatE, ConvE, R-GCN, and SEAL [2, 4, 9, 22, 25, 26, 29, 31, 33, 34]. At the same time, evaluation protocol choices can distort KGC conclusions: Sun et al. show that inappropriate protocols can produce over-optimistic performance reports [27]. Our audit extends this concern to biomedical candidate regimes by comparing random, degree-matched, and graph-near unobserved pairs, and by treating unobserved biomedical edges as benchmark candidates rather than clinically verified negatives.

Calibration under biomedical candidate shift. Calibration work shows that ranking quality and probability reliability are distinct, especially under dataset shift [6, 23, 24]. This distinction is important for biomedical graph completion: a model may rank observed DDI edges above random non-edges, yet assign high probability to plausible graph-near candidates whose true biomedical status is unknown. Our contribution is not a new KGC model; it is a compact reliability audit that separates topology, name semantics, relation evidence, and fusion behavior.

4 Audit Protocol

4.1 Candidate Regimes

We instantiate the audit on `ogbl-ddi` [10], a standard drug–drug interaction graph. Nodes are drugs and observed edges indicate interactions. Because unobserved biomedical edges are not clinically verified non-interactions, we interpret negatives as benchmark candidates rather than definitive false biomedical claims.

We compare three candidate regimes. **Random** candidates are uniformly sampled unobserved drug pairs and approximate common random-negative evaluation. **Degree-matched** candidates control for endpoint degree, reducing shortcuts based on node popularity. **Graph-near** candidates are unobserved pairs from local graph neighborhoods or high-overlap structural contexts. In a DDI graph, this is a reproducible proxy for shared interaction neighborhoods: drugs that touch similar partners are more likely to share safety profiles, pathway contexts, or pharmacological neighborhoods than uniformly sampled pairs. These candidates are therefore harder and more biomedical-facing because they are structurally close to observed interactions and can represent plausible alternatives. Figure 1 summarizes the resulting audit axes.

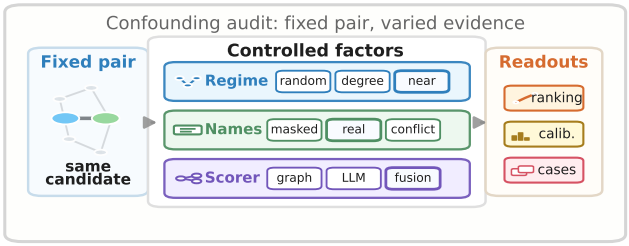

Figure 1: Confounding audit design. The DDI candidate pair is held fixed while candidate regime, name exposure, and scoring view are varied; ranking, calibration, and case-level readouts test whether name priors behave as relation evidence.

4.2 Name-Injection Control

To separate topology use from name-semantics dominance, the completed audit includes a controlled prompt experiment with three matched conditions. In the **symbolic topology** condition, candidate pairs are anonymized and only graph evidence is visible. In the **real-name** condition, the same structural evidence is paired with true biomedical entity names. In the **conflict-name** condition, the same structural evidence is paired with misleading names selected to pull the language prior away from the graph-supported answer.

This control does not attempt to prove that LLMs are strong graph algorithms. Its role is diagnostic: if the largest performance or confidence shift appears after name exposure rather than in the symbolic topology setting, then the main bottleneck in this biomedical audit is not graph unreadability but the dominance of name semantics over relation evidence.

On 60 paired local-topology cases, Qwen2.5-7B perfectly separates high- and low-support candidates under symbolic labels (AUROC/AP = 1.00/1.00; positive/negative mean scores = 0.80/0.30), indicating that the controlled failure is not a total inability to use the provided graph. Replacing symbols with ordinary real drug names reduces AUROC/AP to 0.898/0.951, and conflict-inducing names further reduce it to 0.581/0.731. The key effect is a collapse of the positive–negative score margin from 0.500 to 0.087, showing that name exposure changes the decision rule rather than merely adding useful biomedical knowledge.

4.3 Scorers and Name Visibility

The graph-only audit uses common neighbors, Adamic–Adar, and resource allocation (RA), which are standard local link prediction

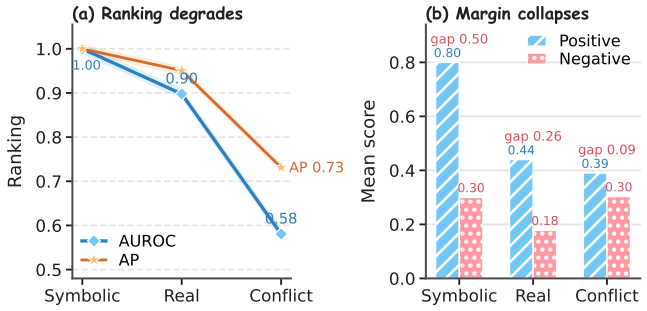

Figure 2: Topology/name-injection control. Qwen2.5-7B uses symbolic topology, but biomedical names reduce ranking performance and compress the positive–negative score margin.

scores [18]. RA is our main structural heuristic because it discounts high-degree shared neighbors. We also include a lightweight DistMult-style embedding scorer: node embeddings are trained with positive DDI edges and sampled unobserved pairs, and candidate pairs are scored by an embedding dot product with endpoint biases. It is not intended as a state-of-the-art KGE model; its role is to check whether candidate-shift effects persist beyond hand-crafted heuristics.

Semantic scoring is performed on the same sampled candidate pairs. To reduce leakage from external DDI descriptions, the current setup uses drug names only. We evaluate two open-source instruction models, Qwen2.5-7B and Qwen3-8B, a stronger closed-source GPT-5.4 scorer, and a biomedical text-encoder proxy based on MedCPT entity-name similarity, following biomedical representation work such as BioBERT and SapBERT [14, 16, 19]. For GPT-5.4, we also run a masked-name control in which drug identifiers are hidden and only entity type plus relation type remain visible. This separates generative prompting behavior, biomedical name exposure, and a non-generative biomedical semantic similarity signal.

4.4 Fusion and Reliability Metrics

We evaluate direct score fusion and learned fusion. Direct fusion combines normalized graph and semantic scores:

$$s_{\text{fusion}} = \alpha s_{\text{graph}} + (1 - \alpha) s_{\text{sem}},$$

where α is selected on a validation candidate regime. Learned fusion uses logistic regression over graph features such as RA, common neighbors, Adamic–Adar, and endpoint degree statistics, optionally adding a normalized semantic score.

We report AUROC and average precision (AP) for ranking behavior. To study reliability, we also report Brier score and expected calibration error (ECE) [6, 23].

4.5 Experimental Setup

For graph-side candidate-shift experiments, all graph metrics are means over four seeds; each seed samples 5,000 test positives and 5,000 validation positives and pairs them with random, degree-matched, or graph-near unobserved candidates. The embedding graph scorer is trained separately per seed and evaluated on the same regimes. Semantic scoring uses a fixed 240-pair audit subset, balanced across regimes with 40 observed and 40 unobserved candidates per regime. Visible prompts expose drug names only, while masked prompts replace identities with anonymized drug identifiers. Calibration and fusion hyperparameters are selected on validation regimes and tested on held-out candidate sets.

Table 1: Graph scorers under candidate shift. Cells report AUROC/AP.

Scorer	Random	Degree-matched	Graph-near
RA	0.965/0.962	0.818/0.818	0.826/0.816
Embedding graph	0.983/0.979	0.929/0.913	0.904/0.879

Table 2: Calibration under shifted candidate regimes. Cells report mean values over seeds.

Test regime	AUROC/AP	Brier	ECE	Neg. prob.
Random	0.969/0.966	0.070	0.027	0.126
Degree-matched	0.851/0.850	0.186	0.136	0.435
Graph-near	0.847/0.833	0.187	0.146	0.450

5 Findings

5.1 F1: Random Negatives Overstate Graph Reliability

Table 1 compares graph scorers across candidate regimes. RA drops substantially from random candidates to degree-matched and graph-near candidates, confirming that random negatives overstate structural reliability. The learned graph scorer is stronger, but it also drops under graph-near candidates. This shows that the candidate shift is not only an artifact of simple heuristics.

The diagnostic point is that the embedding graph scorer retains strong aggregate discrimination. However, structural evidence can still be unreliable in important subregions: when graph-near positives have low common-neighbor support, RA declines to AUROC/AP = 0.639/0.578. This creates a reliability question: if a model preserves ranking discrimination overall, do its probabilities remain meaningful under shifted and low-support candidates?

5.2 F2: Ranking Persists While Calibration Degrades

Table 2 shows calibration results when a calibrator fitted on random validation candidates is evaluated on shifted regimes. On random candidates, calibration appears strong. On degree-matched and graph-near candidates, however, Brier score and ECE increase sharply, and the average probability assigned to negative candidates rises to roughly 0.45.

This is a subtler failure mode than a simple performance drop. A biomedical graph scorer can preserve acceptable ranking while becoming unreliable as a probability estimator on plausible graph-near candidates. The calibration source also matters: when evaluated on graph-near candidates, a degree-matched calibrator reduces ECE to 0.035, while a graph-near calibrator improves AUROC/AP/Brier to 0.882/0.862/0.141. We therefore treat calibration as part of the audit protocol rather than auxiliary post-processing.

5.3 F3: Stronger LLMs Rely on Name Priors

Biomedical entity names are often treated as a source of useful semantic prior. Table 3 suggests a more cautious interpretation.

Table 3: Semantic-only scoring on sampled DDI candidates. Cells report AUROC/AP.

Semantic scorer	Random	Degree-matched	Graph-near
Qwen2.5-7B visible	0.524/0.518	0.540/0.577	0.557/0.523
Qwen3-8B visible	0.500/0.487	0.512/0.553	0.500/0.476
MedCPT entity-cos	0.613/0.582	0.601/0.609	0.614/0.606
GPT-5.4 visible	0.695/0.667	0.725/0.722	0.743/0.760
GPT-5.4 masked	0.500/0.472	0.500/0.530	0.500/0.464

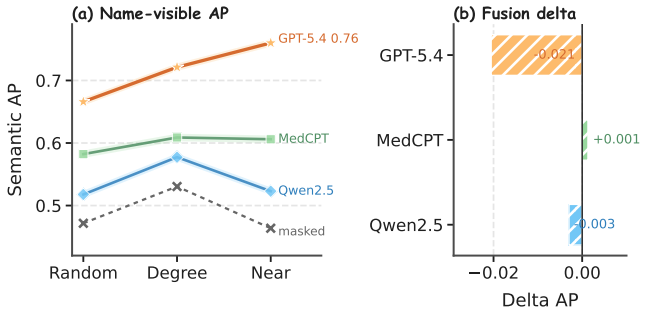

Figure 3: GPT-5.4 semantic and fusion audit. Visible-name semantics improve AP, the masked-name control provides no ranking signal, and logistic fusion still reduces AP.

Qwen2.5-7B and Qwen3-8B provide weak or nearly constant semantic scores, and MedCPT entity-name similarity provides only a modest signal. GPT-5.4 is substantially stronger under visible drug names, reaching AUROC/AP = 0.743/0.760 on graph-near candidates, where semantic benefit is most expected. The masked-name control is intentionally an identity ablation, not a graph-evidence prompt: with only anonymized drug types, GPT-5.4 assigns all samples score 0.50, yielding AUROC = 0.500 by the standard tie rule.

This does not imply that biomedical semantics are useless. Rather, it sharpens the distinction between name semantics and relation evidence. GPT-5.4 can extract a useful name-visible prior, but the masked result shows that this signal is attached to entity names rather than relation-grounded evidence available from the prompt. Name semantics can encode topical or pharmacological relatedness while remaining poorly aligned with whether a specific DDI edge is supported.

5.4 F4: Stronger Semantics Do Not Guarantee Fusion Gains

Graph-near candidates are the regime where semantic priors appear most desirable: they are structurally plausible and difficult for local graph scores. Figure 3 shows graph-near-trained logistic fusion. For weak semantic scorers, AP changes are small. For GPT-5.4, the semantic signal is stronger, but adding it to the logistic graph features still reduces AP on all three test regimes: Qwen2.5 changes AP by +0.008/−0.006/−0.003 on random, degree-matched, and graph-near tests; MedCPT changes AP by −0.009/−0.001/+0.001; GPT-5.4 visible changes AP by −0.025/−0.013/−0.021.

The counterintuitive result is therefore not that graph-near candidates are difficult. It is that stronger name-visible semantics still do not automatically become stable complementary evidence. A direct alpha fusion tuned on graph-near candidates can place most weight on GPT-5.4 semantics ($\alpha = 0.05$) and improve the graph-near sample to AUROC/AP = 0.758/0.772, but that same semantic-heavy mixture does not transfer to random and degree-matched regimes. Learned fusion is even more conservative: it preserves the graph score's shape but reduces AP when the GPT feature is added. LLM-derived semantic scores should therefore be treated as auditable priors rather than automatic relation evidence.

Table 4: Graph-near GPT-5.4 case analysis. Unobserved candidates are benchmark negatives, not verified clinical non-interactions.

Pair	Obs.	RA	GPT	Effect
Cyclosporine–Sulfamoxole	Y	0.093	0.72/0.50	recovers low-support edge
Mestranol–Ulipristal	N	0.450	0.81/0.50	elevates hormonal candidate
Scopolamine–Fenproporex	N	0.076	0.68/0.50	elevates pharmacodynamic candidate
Capecitabine–Levosalbutamol	Y	0.479	0.22/0.50	downweights name-distant edge

5.5 F5: Case Analysis Reveals Semantic Shortcuts

Representative graph-near cases from the GPT-5.4 audit illustrate why aggregate gains from visible names should not be interpreted as direct relation evidence. Table 4 shows that the same name prior can recover a low-support observed edge, raise plausible unobserved candidates, or suppress an observed edge whose names appear mechanistically distant. The masked-name control returns 0.50 in these rows, confirming that the visible-name score is driven by entity-name semantics.

6 Reporting Recommendations

Our audit points to three reporting recommendations for LLM-augmented biomedical graph learning. First, evaluations should report candidate regimes beyond random negatives, because random, degree-controlled, and graph-near candidates answer different reliability questions. Second, calibration should be tested after candidate shift; ranking alone can miss confident errors on plausible non-edges. Third, name-only, masked-name, and evidence-augmented LLM settings should be separated, because biomedical entity names create both opportunity and shortcut risk.

These recommendations are methodological rather than presentational in the biomedical setting. Graph-near drug pairs can share safety profiles, mechanisms, or neighboring interaction patterns, making them more meaningful than generic hard negatives; at the same time, unobserved biomedical edges should not be treated as clinically verified false edges. Because `ogbl-ddi` is a controlled single-relation benchmark, the same protocol can port to multi-relational biomedical KGs, stronger KGE/GNN backbones, and prompts supplied with external evidence such as precision medicine KGs [3]. Graph–LLM systems should therefore avoid collapsing name semantics and relation evidence into one score: entity-name dropout, schema-constrained prompts, evidence-grounded scoring, and conflict-aware fusion should credit semantic gains only after masked-name, graph-near, calibration, and fusion checks. A compact audit checklist should report sampler, name visibility, calibration source, and fusion tuning regime.

References

[1] Garima Agrawal, Tharindu Kumarage, Zeyad Alghamdi, and Huan Liu. 2024. Can LLMs Perform Structured Graph Reasoning? *arXiv preprint arXiv:2402.01805* (2024).

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*.

[3] Payal Chandak, Kexin Huang, and Marinka Zitnik. 2023. Building a Knowledge Graph to Enable Precision Medicine. *Scientific Data* 10, 1 (2023), 67. doi:10.1038/s41597-023-01960-3

[4] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2D Knowledge Graph Embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[5] Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2023. Talk like a Graph: Encoding Graphs for Large Language Models. *arXiv preprint arXiv:2310.04560* (2023).

[6] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*.

[7] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*.

[8] Daniel S. Himmelstein, Antoine Lizee, Christine Hessler, Leo Brueggeman, Sabrina L. Chen, Dexter Hadley, Ari Green, Pouya Khankhanian, and Sergio E. Baranzini. 2017. Systematic Integration of Biomedical Knowledge Prioritizes Drugs for Repurposing. *eLife* 6 (2017), e26726. doi:10.7554/eLife.26726

[9] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, Jose Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge Graphs. *Comput. Surveys* 54, 4 (2021), 71:1–71:37. doi:10.1145/3447772

[10] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *Advances in Neural Information Processing Systems*.

[11] Xuanwen Huang, Kaiqiao Han Li, Xianzhi Liu, Hongxu Qu, Wentao Zou, Yew-Soon Song, Wei Liu, Yuening Zhou, Wenxuan Zhao, and Tong Zhang. 2023. Can LLMs Effectively Leverage Graph Structural Information through Prompts, and Why? *OpenReview preprint* (2023).

[12] Qiu Ji, Pengfei Shen, Haolei Zhu, Guilin Qi, Yang Sheng, Lianlong Wu, Kang Xu, and Yuan Meng. 2025. LLM-Hype: A Targeted Evaluation Framework for Hypernym-Hyponym Identification in Large Language Models. In *VLDB 2025 Workshop: LLM+Graph*.

[13] Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2023. Large Language Models on Graphs: A Comprehensive Survey. *arXiv preprint arXiv:2312.02783* (2023).

[14] Qiao Jin, Robert Leaman, and Zhiyong Lu. 2023. MedCPT: Contrastive Pre-trained Transformers with Large-scale PubMed Search Logs for Zero-shot Biomedical Information Retrieval. *Bioinformatics* (2023).

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[16] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A Pre-trained Biomedical Language Representation Model for Biomedical Text Mining. *Bioinformatics* 36, 4 (2020), 1234–1240. doi:10.1093/bioinformatics/btz682

[17] Yuhan Li, Peisong Li, Yuxuan Bai, Zhen Chen, and Lichao Wang. 2024. LLaGA: Large Language and Graph Assistant. *arXiv preprint arXiv:2402.08170* (2024).

[18] David Liben-Nowell and Jon Kleinberg. 2007. The Link-Prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology* 58, 7 (2007), 1019–1031.

[19] Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-Alignment Pretraining for Biomedical Entity Representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4228–4238. doi:10.18653/v1/2021.naacl-main.334

[20] Linhao Luo, Jiaxin Ju, Bo Xiong, Yuan-Fang Liang, Gholamreza Haffari, and Shirui Pan. 2024. GraphInstruct: Empowering Large Language Models with Graph Understanding and Reasoning Capability. *arXiv preprint arXiv:2403.04483* (2024).

[21] Nandana Mihindukulasooriya, Niharika S. D'Souza, Faisal Chowdhury, and Horst Samulowitz. 2025. Automatic Prompt Optimization for Knowledge Graph Construction: Insights from an Empirical Study. *arXiv preprint arXiv:2506.19773* (2025).

[22] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104, 1 (2016), 11–33. doi:10.1109/JPROC.2015.2483592

[23] Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting Good Probabilities with Supervised Learning. In *International Conference on Machine Learning*.

[24] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. In *Advances in Neural Information Processing Systems*.

[25] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *European Semantic Web Conference*. 593–607.

[26] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.

[27] Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha Talukdar, and Yiming Yang. 2020. A Re-evaluation of Knowledge Graph Completion Methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 5516–5522. doi:10.18653/v1/2020.acl-main.489

[28] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Long Xia, Dawei Yin, Chao Li, and Qianyu Huang. 2023. GraphGPT: Graph Instruction Tuning for Large Language Models. *arXiv preprint arXiv:2310.13023* (2023).

[29] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex Embeddings for Simple Link Prediction. In *International Conference on Machine Learning*.

[30] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[31] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743. doi:10.1109/TKDE.2017.2754499

[32] David S. Wishart, Yannick D. Feunang, An C. Guo, Elvis J. Lo, Ana Marcu, Jason R. Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. 2018. DrugBank 5.0: A Major Update to the DrugBank Database for 2018. *Nucleic Acids Research* 46, D1 (2018), D1074–D1082. doi:10.1093/nar/gkx1037

[33] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *International Conference on Learning Representations*.

[34] Muhan Zhang and Yixin Chen. 2018. Link Prediction Based on Graph Neural Networks. In *Advances in Neural Information Processing Systems*.

[35] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling Polypharmacy Side Effects with Graph Convolutional Networks. *Bioinformatics* 34, 13 (2018), i457–i466. doi:10.1093/bioinformatics/bty294