ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964.
Available from http://www.i6doc.com/en/.

Learning and Reasoning on Knowledge and Heterogeneous Graphs in the era of Graph Foundation and Large Language Models

Matteo Zignani[1], Pasquale Minervini[2], Roberto Interdonato[3] and Manuel Dileo[4]

1- Department of Computer Science, University of Milan, IT
2- School of Informatics, University of Edinburgh, UK and Miniml.AI, UK
3- CIRAD - UMR TETIS, Montpellier, France
4- Center for Computational Biology, Human Technopole, Milan, IT

Abstract. Knowledge Graphs (KGs) and heterogeneous graphs (HGs) offer a principled way to represent multi-entity, multi-relational systems, while also revealing a persistent tension between expressive modeling, scalable learning, and faithful reasoning. Two trends are rapidly reshaping the field: *graph foundation models* (GFMs), which seek transfer across graphs, tasks, and domains via large-scale pretraining, and the growing integration of *large language models* (LLMs) with graph-structured knowledge to improve grounding, interaction, and reasoning. Temporal settings add further challenges, as evolving facts and interactions demand time-consistent modeling and evaluation. This tutorial provides a structured survey of these directions: we introduce a unified background and notation for typed heterogeneous graphs, (temporal) KGs, and event-based temporal heterogeneous graphs; we then formalize the main task families (KG completion, query answering, node/graph prediction, and temporal variants), emphasizing evaluation protocols and leakage pitfalls. Finally, we review recent advances in GFMs and LLM–graph integration, and summarize the state of the art in learning over temporal heterogeneous graphs and temporal KGs.

1 Introduction

Graphs with typed nodes, typed relations, and rich attributes have become the default substrate for representing data where interactions are as important as entities. Knowledge graphs (KGs) instantiate this view with curated or automatically extracted facts, typically expressed as triples (s, r, o) (or temporal quadruples (s, r, o, t)). Heterogeneous graphs generalize beyond strict KG schemas by allowing multiple node and edge types, multimodal attributes, and domain-specific semantics.

The technical objective in this area is twofold. First, we need learning models that can exploit heterogeneity: they must be relation- and type-aware, robust to missing and noisy edges, and scalable to large graphs. Second, we need *reasoning* mechanisms that go beyond pattern matching: models should support multi-hop inference, enable constraints (logical consistency, temporal validity), and provide explanations that are intelligible and auditable in high-stakes applications.

Two recent trends are reshaping the current landscape on knowledge and heterogeneous graphs: (1) *Graph foundation models* (GFMs) aim to learn transferable representations and reasoning behaviors from large-scale pretraining, so that a single model can be adapted (or prompted) for diverse graphs and tasks [1]. In the KG setting, this naturally leads to *fully inductive* generalization: the model should operate on unseen KGs with unseen entity and relation vocabularies [2, 3, 4, 5]; and (2) LLMs enable new interfaces between unstructured corpora and structured relational representations, and they can be combined with graphs both as external memory (grounding) and as a target modality for instruction tuning [6].

A third axis is time. Many real graphs evolve; so in temporal KGs and temporal heterogeneous graphs, the learning goal is inherently forward-looking such as predicting future links/events and explaining them under non-stationarity. This makes evaluation protocols and leakage control decisive [7].

Contributions and scope. This tutorial paper provides: (i) an up-to-date review of GFMs for KGs and heterogeneous graphs, emphasizing transfer regimes and text-aware semantics; (ii) a design-space taxonomy of LLM–graph integration with attention to provenance and faithfulness; and (iii) a tutorial view of temporal heterogeneous learning, including model families and evaluation pitfalls

2 Modeling and task formalization

We adopt a unified formalism that allows us to discuss heterogeneous graphs, knowledge graphs, and their temporal counterparts within a single mathematical language. This is useful because many modern approaches move fluidly across these representations, sometimes treating a KG as a typed graph, sometimes treating a typed graph as a set of relational facts, and often augmenting either with textual metadata.

A (static) heterogeneous graph is a typed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} \subseteq \mathcal{N} \times \mathcal{S}$, with $\mathcal{N}$ the set of nodes and $\mathcal{S}$ the set of node-types, is the set of typed nodes; and and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \times \mathcal{R}$ is the set of typed edges, with $\mathcal{R}$ a set of types for links. This formalization, expressed in [8], represents the extension to the case of a multi-graph of the usual definition of heterogeneous graph [9], allowing an edge (u, v) to have different types and modeling different kinds of relations between the same pair of nodes. In practical settings, nodes and edges are often equipped with attributes and metadata. We denote by x_v the (possibly vector-valued) attributes of node v and by x_e the attributes of edge e. Since many recent pipelines are text-aware, we also allow each node v to carry a set of textual content s_v (e.g., name, abstract, profile, documentation, posts), and similarly, edges may carry textual labels. Moreover, heterogeneous graphs can be fitted with a *schema* (sometimes called a meta-graph), which describes what type-to-type relations are admissible by specifying which triples $((u, s_1), (v, s_2), r)$ are allowed. This way schema may affect both modeling (e.g., relation-specific message passing and attention) and evaluation (e.g., schema-valid negative sampling).

Knowledge graphs can be treated as a particular case of this typed perspective. In fact, we model a knowledge graph as a set of relational facts in the form of triples $\mathcal{K} = \{(h, r, t)\}$ where $h, t \in \mathcal{V}$ are entities and $r \in \mathcal{R}$ is a relation type. Under this view, the KG is a heterogeneous graph whose edge type set coincides with the relation vocabulary $\mathcal{R}$. Many applied KGs also include literals (numeric or string values) and more expressive statement forms (e.g., qualifiers). These can be modeled by augmenting the graph with additional nodes and edges, or by using hyper-relational/reification encodings; the important point for our purposes is that the resulting structure remains representable as a typed heterogeneous graph.

Finally, the notation can be extended to include time within the graph. In the continuous-time perspective a temporal heterogeneous graph can be represented as an event stream $\mathcal{D} = \{(u, r, v, \tau, a)\}$ where $u, v \in \mathcal{V}$ are typed-nodes, $r \in \mathcal{E}$ is an edge type, $\tau \in \mathbb{R}^+$ is the event time, and a represents optional event attributes. This continuous-time perspective is well-suited for interaction logs and transactional data. An alternative is a discrete-time (snapshot) view, where one observes a sequence of graphs $\{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(T)}\}$; this is often convenient for periodic measurements, but can hide fine-grained temporal dynamics. A temporal knowledge graph (tKG) is typically represented as timestamped quadruples (h, r, t, τ), emphasizing that relational validity is time-conditioned: the plausibility of (h, r, t) may change as τ evolves.

From the perspective of learning and reasoning on HGs and KGs through graph representation learning, we denote by f_θ a parametric encoder that maps graph elements to representations. In heterogeneous graphs, this typically yields node embeddings $z_v = f_\theta(v, \mathcal{G})$, while in KGs it yields entity embeddings $z_e = f_\theta(e, \mathcal{K})$ and often a parameterization of relations, which we denote generically by z_r or by relation-specific parameters within the model. Two broad paradigms recur across the literature. The first is message passing (heterogeneous GNNs), where embeddings are computed by aggregating information from typed neighborhoods with relation/type-dependent transformations. The second is knowledge graph embedding (KGE) modeling, where a scoring function $s_\theta(h, r, t)$ measures the plausibility of a triple and learning proceeds by making observed triples score higher than corrupted or sampled negatives. Although these paradigms are historically distinct, many recent foundation-model directions blur the boundary by combining message passing with relational scoring or by unifying them through pretraining objectives.

When nodes and edges are text-attributed, we also introduce a text encoder $g_\omega(\cdot)$, which may be an LLM or an LLM-derived embedding model, and yields text representations from s_v, i.e. $w_v = g_\omega(s_v)$. In LLM–graph integration pipelines, the representations w_v and z_v (or w_e and z_e in the KG setting) can be fused in multiple ways, for example by concatenation and projection, by cross-attention, by adapters, or by retrieval mechanisms that condition generation on subgraphs and node/edge texts.

2.1 Task formalization

To compare methods across graph foundation models, LLM–graph integration, and temporal heterogeneous learning, it is essential to clarify which learning

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964. Available from http://www.i6doc.com/en/.

problems are being solved and how success is measured. We therefore formalize the tasks that recur most frequently in the literature, distinguishing static settings from temporal settings while keeping the underlying operators compatible with the notation in the previous section..

In the static KG setting, the canonical problem is *knowledge graph completion.* One observes a set of triples $\mathcal{K}$ and wishes to infer missing facts. Operationally, this is typically posed as a ranking problem: given a query of the form $(h, r, ?)$ one ranks candidate entities $\tilde{t} \in \mathcal{V}$ by a score $s_\theta(h, r, \tilde{t})$; similarly, for $(?, r, t)$ one ranks candidate heads. Learning is commonly done by maximizing the score of observed triples while minimizing the score of negatives produced by corrupting heads or tails, leading to margin-based or contrastive objectives, sampled-softmax variants, or other noise-contrastive formulations. Evaluation is based on ranking metrics such as mean reciprocal rank (MRR) and Hits@K, frequently under the "filtered" protocol. A closely related static task is *relation prediction*, where the input is $(h, ?, t)$ and the goal is to infer the most plausible relation type $r \in \mathcal{R}$.

Moving to heterogeneous graphs more broadly, a central supervised problem is *node classification.* We provide labels y_v for nodes (typically for a subset of nodes of one or more types), and the goal is learning a predictor from node representations. If $z_v = f_\theta(v, \mathcal{G})$ denotes the learned embedding, training proceeds by minimizing a cross-entropy loss on labeled nodes, often in semi-supervised regimes. Many foundation model pipelines follow a pretrain/adapt pattern in which f_θ is first learned with self-supervised objectives and then adapted through linear probing or fine-tuning for the target classification task; the task definition itself remains the same.

Beyond these supervised tasks, KGs are often used for *query answering and reasoning.* In this setting, the input is a structured query q that imposes constraints over relations and entities, and the output is an answer set (or a ranking of candidate entities) $\mathcal{A}(q) \subseteq \mathcal{V}$. Multi-hop path queries, conjunctive queries, and compositional logical forms are common instances. Methods span embedding-based query operators, neural logical reasoning, rule-based reasoning, and hybrid pipelines in which an LLM decomposes or rewrites the query, while retrieval and a graph model produce grounded candidate answers. Standard evaluation focuses on answer ranking accuracy, but LLM-mediated systems often require additional assessment of faithfulness and traceability, namely whether the returned answer can be justified by explicit evidence (retrieved subgraphs, paths, or documents).

Another frequently encountered static problem, especially when integrating multiple sources, is *entity alignment* (or record linkage) across knowledge graphs. Given two graphs $\mathcal{K}_1$ and $\mathcal{K}_2$, the goal is to identify correspondences $e^{(1)} \leftrightarrow e^{(2)}$ between entities. This task is relevant for foundation models because it stresses cross-graph generalization, representation invariances, and robustness to schema mismatch and textual ambiguity.

Temporal settings introduce additional structure because training and inference must be conditioned on history and constrained by time. In temporal

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964.
Available from http://www.i6doc.com/en/.

knowledge graphs, the analogue of KG completion is *temporal link prediction* (temporal KG completion). One assumes access to facts up to time τ and aims to predict facts at a future time $\tau + \Delta$. The scoring function may be written as $s_\theta(h, r, t, \tau)$ or more explicitly as a conditional score $s_\theta(h, r, t \mid \text{history} \leq \tau)$, reflecting that the model should not use information beyond τ. The evaluation protocol is crucial: time-based splits must be respected, and negative sampling should be both type-consistent and temporally feasible. In continuous-time temporal heterogeneous graphs represented as event streams $\mathcal{D}$, a common formulation is *event prediction.* Given the past events up to time τ, one may predict the next interacting counterpart (e.g., predicting v in an event $(u, \rho, v, \tau, \cdot)$), predict the event type ρ, or model the time of the next event itself. Many models express the timing component through intensity or hazard functions, while the structural component is evaluated through ranking metrics over candidate targets conditioned on the source node and the recent history. Even when the primary objective is likelihood over event times, downstream evaluation frequently returns to ranking-based link prediction metrics to facilitate comparison across model families. Across all temporal tasks, a recurring methodological theme is that protocols matter as much as models. Transductive and inductive splits must be declared explicitly (e.g., whether test nodes or relations were unseen during training), negative sampling must obey schema constraints, and temporal leakage must be systematically prevented.

3 Promising research directions for knowledge and heterogeneous graphs

3.1 Foundation models for KGs and HGs

From relation-aware GNNs to transferable representations. Classical KG learning is dominated by embedding models and relational message passing. Translational/bilinear embeddings provide strong baselines for link prediction, while relational GNNs, such as R-GCN [10], model multi-relational neighborhoods with relation-specific parameters. For heterogeneous graphs, early successes relied on meta-paths and attention mechanisms to capture type-specific context, e.g., HAN [11]. Graph transformers extend attention to the relational setting, with HGT providing type-dependent projections and relation-specific attention for general heterogeneous graphs [12].

Graph Foundation Models (GFMs) are raising the bar and introducing a few interesting challenges since, instead of learning on a single dataset, they attempt to learn reusable primitives for representation and reasoning. Recent surveys on the topic propose a modular view with three components: (a) a backbone (GNN, Transformer, LLM), (b) pretraining objectives, and (c) adaptation mechanisms [1]. This decomposition is useful when comparing seemingly different efforts (KG reasoning models, TAG models, retrieval-augmented GFMs) under a shared lens.

ESANN 2026 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and
Machine Learning. Bruges (Belgium) and online event, 22-24 April 2026, i6doc.com publ., ISBN 9782875870964.
Available from http://www.i6doc.com/en/.

Fully inductive GFM. A specific feature of GF modeling is *vocabulary-agnostic* reasoning: entities and relations in the test graphs may be entirely unseen during training. To address this constraint, ULTRA introduced a conditioning mechanism for relation representations that supports zero-shot KG completion across many graphs [2]. TRIX increased expressivity for zero-shot domain transfer and was evaluated in the fully inductive regime [3]. KG-ICL reframed KG reasoning via prompt-graph generation and in-context learning, targeting "one model for all KGs" without finetuning [5]. SEMMA argues that structural inductive bias is not enough when relation vocabularies shift: it systematically injects textual semantics (via LLM-derived relation embeddings) to improve generalization when structure-only methods collapse [4].

Text-attributed graphs as a unifying interface. Many heterogeneous graphs are naturally text-attributed (papers, products, users, entities). Text can therefore act as a cross-domain interface for transfer. UniGraph proposed a cross-domain foundation model for text-attributed HGs that generalizes to unseen graphs and tasks by leveraging text as a shared semantic space [13]. PromptGFM ("LLM as GNN") highlights an important practical issue: naive node-to-token mapping causes token explosion and graph-specific semantics; it instead learns a graph vocabulary and prompts the LLM to emulate a GNN workflow in text space [14].

Beyond parameter-only knowledge: retrieval and scalability. A recurring theme is that parameterizing all knowledge in a GFM is inefficient and brittle. Recent work explores retrieval augmentation for GFMs: GFM-RAG trains a graph foundation model for retrieval augmented generation, aiming to generalize across unseen retrieval graphs [15]. RAG-GFM instead uses retrieval to externalize knowledge and mitigate in-memory bottlenecks in GFMs, combining semantic retrieval (text) and structural retrieval (motifs) [16]. These directions suggest that future GFMs for KGs and HGs will likely combine a pretrained structure with explicit memory and evidence.

Open problems. Despite rapid progress, the field still lacks a standardized evaluation for transfer: what constitutes a fair "unseen KG" protocol, and how do we control for hidden leakage through text, preprocessing, or shared entities? Additionally, heterogeneous graphs expose schema-shift challenges (new types/relations) and extreme long-tail semantics that are not well captured by current benchmarks. Finally, the interface between neural pretraining and symbolic constraints (temporal validity, ontology constraints) remains underdeveloped: foundation models should not only predict, but also support predictions with human-readable traces.

3.2 Integrating LLMs and Graphs

A further ongoing research direction, which, for many aspects, overlaps with graph foundation models, is the integration of LLM and learning methods on graphs. To this aim, research is converging to a common taxonomy where LLMs can be used on graphs (predictor/encoder/aligner), or graphs can be used for

LLMs (grounding/external memory) [6]. For KGs and HGs, three integration paradigms are especially actionable.

Graph grounding for LLMs (GraphRAG-style). Retrieval-augmented generation improves factuality by conditioning answers on evidence. GraphRAG extends this by building a graph over retrieved units (entities, communities, or document fragments) and performing local-to-global aggregation for question answering and summarization [17]. Recent work moves from heuristically constructed graphs to learned graph retrievers: GFM-RAG trains a graph foundation model that reasons over query/knowledge relationships and generalizes to unseen retrieval graphs without finetuning [15]. A complementary approach is to augment GFMs with retrieval (RAG-GFM), offloading knowledge to external stores while keeping transferable representation learning [16]. For tutorial purposes, this paradigm is attractive because it exposes explicit provenance surfaces (retrieved nodes/edges and multi-hop paths) that can be inspected and audited.

LLMs as graph learners: instruction tuning and verbalization. Instruction tuning can endow LLMs with graph reasoning behaviors, either by verbalizing graphs into code-like formats or by aligning LLMs with graph encoders. GraphGPT introduced graph instruction tuning and a text–graph grounding component to improve zero-shot generalization on graph tasks [18]. InstructGraph unified multiple graph tasks into a universal code-like format and combined instruction tuning with preference alignment to improve both graph reasoning and generation [19]. These approaches are useful when the output must be natural language (explanations, plans) but still adheres to the graph structure.

LLM-empowered KG construction and maintenance. LLMs have also shifted KG construction from brittle extraction pipelines toward prompt-driven workflows for ontology engineering, extraction, fusion, and refinement. A recent survey systematizes this space, distinguishing schema-based and schema-free paradigms and highlighting limitations such as inconsistency, entity resolution, and provenance tracking [20]. In practice, KG construction is where "LLM meets graph" becomes operational: the KG becomes an external memory for agents and RAG systems, while also enabling consistency checks and symbolic constraints.

3.3 Learning on temporal HGs and KGs

Temporal graph representations and continuous-time learning. Temporal graphs can be modeled as snapshot sequences or as event streams. Continuous-time temporal GNNs learn from event histories with time encodings, attention over temporal neighborhoods, and memory updates. TGAT introduced time-aware attention mechanisms over temporal neighborhoods [21], while TGN integrated a memory module to represent evolving node states for link prediction and downstream tasks [22]. These methods focus mostly on homogeneous graphs, but they provide the architectural primitives (time encodings, temporal attention, memory) used in heterogeneous extensions.

Heterogeneous temporal GNNs. Heterogeneous temporal graphs (HTGs) introduce a tight coupling between type-aware aggregation and temporal evolution, because both the neighborhood structure and the semantics of interactions change over time and must be modeled without violating schema constraints. HTGNN proposed a hierarchical aggregation pipeline that separates intra-relation aggregation, inter-relation semantic fusion, and across-time propagation, thereby explicitly decomposing the learning problem into (i) relation-specific message passing, (ii) type-aware combination of relational signals, and (iii) temporal smoothing/memory over snapshots or time steps [23]. SE-HTGNN revisited this design space by arguing that some of the architectural complexity is not always necessary in practice, and introduced a simpler attention-based paradigm that yields substantial speedups while preserving forecasting accuracy on standard HTG benchmarks [24]. On the other side, DURENDAL emphasizes that many heterogeneous graph learners can be repurposed to evolving networks if the temporal update mechanism is made explicit and composable with multirelational message passing [25]. It combines snapshot-based temporal modeling with relation-aware semantic aggregation and introduces two update schemes (*Update-Then-Aggregate* and *Aggregate-Then-Update*) that differ in whether temporal state updates act before or after the relation-wise semantic fusion step.

Temporal knowledge graphs and temporal reasoning. Temporal KGs encode evolving facts and support temporal KG completion, event forecasting, and time-sensitive question answering. One line of work injects temporal signals into embeddings and relational models; another leverages PLMs by converting temporal facts to prompted text sequences. PPT cast temporal KG completion as masked prediction over prompted sequences with time-interval prompts [26]. More recently, the first steps toward *foundation models for temporal KG reasoning* emphasize fully-inductive transfer across TKGs: POSTRA proposes a pretrained model that generalizes zero-shot to unseen temporal KGs with different temporal granularities and vocabularies [27].

LLMs for temporal KG reasoning and rule extraction. LLMs can contribute to temporal reasoning by extracting temporal rules or providing explainable reasoning traces. LLM-DA uses an LLM to analyze historical facts and extract temporal logical rules, then updates these rules via a dynamic adaptation strategy without fine-tuning the LLM [28]. This is illustrative of a broader theme: in temporal settings, "fast updating" is essential, and rule/explanation artifacts can be easier to update than large neural parameters.

4 Conclusion and outlook

We brought together converging work on knowledge graphs, heterogeneous graph learning, graph foundation models, and LLM–graph integration, with a particular focus on the temporal aspects. A key message is that many differences across method families boil down to modeling choices—how we represent types, attributes, text, and time—which is why an explicit and unified data model is useful for fair comparison. Equally important, task definitions and evaluation protocols strongly shape what results mean, especially in temporal settings where leakage and inconsistent splits can quietly inflate performance. Looking ahead, progress hinges on clarifying what "transfer" should mean for graph foundation models, on making LLM–graph pipelines more faithful and traceable, and on standardizing temporal benchmarks and metrics so that claims remain reproducible and practically relevant.

References

[1] Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S. Yu, and Chuan Shi. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[2] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *The Twelfth International Conference on Learning Representations*, 2024.

[3] Yucheng Zhang, Beatrice Bevilacqua, Mikhail Galkin, and Bruno Ribeiro. Trix: A more expressive model for zero-shot domain transfer in knowledge graphs. In Guy Wolf and Smita Krishnaswamy, editors, *Proceedings of the Third Learning on Graphs Conference*, volume 269 of *Proceedings of Machine Learning Research*, pages 12:1–12:28. PMLR, 26–29 Nov 2025.

[4] Arvindh Arun, Sumit Kumar, Mojtaba Nayyeri, Bo Xiong, Ponnurangam Kumaraguru, Antonio Vergari, and Steffen Staab. SEMMA: A semantic aware knowledge graph foundation model. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31825–31848. ACL, November 2025.

[5] Yuanning Cui, Zequn Sun, and Wei Hu. A prompt-based knowledge graph foundation model for universal in-context reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[6] Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh Chawla, and Chao Huang. A survey of large language models for graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pages 6616–6626. ACM, 2024.

[7] Julia Gastinger, Shenyang Huang, Mikhail Galkin, Erfan Loghmani, Ali Parviz, Farimah Poursafaei, Jacob Danovitch, Emanuele Rossi, Ioannis Koutis, Heiner Stuckenschmidt, Reihaneh Rabbany, and Guillaume Rabusseau. Tgb 2.0: A benchmark for learning on temporal knowledge graphs and heterogeneous graphs. In *Advances in Neural Information Processing Systems*, volume 37, pages 140199–140229, 2024.

[8] Josephine Thomas, Alice Moallemy-Oureh, Silvia Beddar-Wiesing, and Clara Holzhüter. Graph neural networks designed for different graph types: A survey. *Transactions on Machine Learning Research*, 2023.

[9] Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and Philip S Yu. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. *IEEE Transactions on Big Data*, 9(2):415–436, 2022.

[10] Michael Schlichtkrull, , Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Euroepan Semantic Web Conference– ESWC 2018*, pages 593–607, 2018.

[11] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The world wide web conference*, pages 2022–2032, 2019.

[12] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710, 2020.

[13] Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, pages 448–459, New York, NY, USA, 2025. ACM.

[14] Xi Zhu, Haochen Xue, Ziwei Zhao, Wujiang Xu, Jingyuan Huang, Minghao Guo, Qifan Wang, Kaixiong Zhou, Imran Razzak, and Yongfeng Zhang. Llm as gnn: Graph vocabulary learning for text-attributed graph foundation models. *arXiv preprint arXiv:2503.03313*, 2025.

[15] Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. GFM-RAG: Graph foundation model for retrieval augmented generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[16] Haonan Yuan, Qingyun Sun, Jiacheng Tao, Xingcheng Fu, and Jianxin Li. Overcoming in-memory bottlenecks in graph foundation models via retrieval-augmented generation. *arXiv preprint arXiv:2601.15124*, 2026.

[17] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[18] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 491–500, 2024.

[19] Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian McAuley. InstructGraph: Boosting large language models via graph-centric instruction tuning and preference alignment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13492–13510, Bangkok, Thailand, 2024. ACL.

[20] Haonan Bian. Llm-empowered knowledge graph construction: A survey. *arXiv preprint arXiv:2510.20345*, 2025.

[21] da Xu, chuanwei ruan, evren korpeoglu, sushant kumar, and kannan achan. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations*, 2020.

[22] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *ICML 2020 Workshop on Graph Representation Learning*, 2020.

[23] Yujie Fan, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. Heterogeneous temporal graph neural network. In *Proceedings of the 2022 SIAM international conference on data mining (SDM)*, pages 657–665. SIAM, 2022.

[24] YiliWang, Tairan Huang, Changlong He, Qiutong Li, and Jianliang Gao. Simple and efficient heterogeneous temporal graph neural network. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[25] Manuel Dileo, Matteo Zignani, and Sabrina Gaito. Durendal: Graph deep learning framework for temporal heterogeneous networks. *arXiv preprint arXiv:2310.00336*, 2023.

[26] Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. Pre-trained language model with prompts for temporal knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7790–7803. ACL, July 2023.

[27] Jiaxin Pan, Mojtaba Nayyeri, Osama Mohammed, Daniel Hernandez, Rongchuan Zhang, Cheng Cheng, and Steffen Staab. Towards foundation model on temporal knowledge graph reasoning. *arXiv preprint arXiv:2506.06367*, 2025.

[28] Jiapu Wang, Kai Sun, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, 2024.