SEMMA: A Semantic Aware Knowledge Graph Foundation Model

Arvindh Arun [°1], **Sumit Kumar** [°2], **Mojtaba Nayyeri** [1], **Bo Xiong** [3]
Ponnurangam Kumaraguru [2], **Antonio Vergari** [4], **Steffen Staab** [15]

[1]Institute for AI, University of Stuttgart, [2]IIIT Hyderabad,
[3]Stanford University, [4]University of Edinburgh, [5]University of Southampton
arvindh.arun@ki.uni-stuttgart.de

Abstract

Knowledge Graph Foundation Models (KGFMs) have shown promise in enabling zero-shot reasoning over unseen graphs by learning transferable patterns. However, most existing KGFMs rely solely on graph structure, overlooking the rich semantic signals encoded in textual attributes. We introduce SEMMA, a dual-module KGFM that systematically integrates transferable textual semantics alongside structure. SEMMA leverages Large Language Models (LLMs) to enrich relation identifiers, generating semantic embeddings that subsequently form a textual relation graph, which is fused with the structural component. Across 54 diverse KGs, SEMMA outperforms purely structural baselines like ULTRA in fully inductive link prediction. Crucially, we show that in more challenging generalization settings, where the test-time relation vocabulary is entirely unseen, structural methods collapse while SEMMA is 2x more effective. Our findings demonstrate that textual semantics are critical for generalization in settings where structure alone fails, highlighting the need for foundation models that unify structural and linguistic signals in knowledge reasoning.

1 Introduction

Knowledge Graphs (KGs) are used to store data and knowledge as triples *(subject entity, relation, object entity)*, which form graphs (Hogan et al., 2021). Knowledge can be expressed in its ontologies that enable logical reasoning over data. Knowledge graph embedding methods add reasoning by similarity and analogy, allowing for, e.g., recommending relationships between entities that have not been asserted explicitly and cannot be deduced deductively (Ji et al., 2022). Such link prediction has been applied to recommender systems (Zhang et al., 2025a), entity linking (Kolitsas et al., 2018), and question answering over knowledge graphs (Perevalov et al., 2022).

In analogy to Large Language Models (LLMs) that learn complex correlations between tokens from a large corpus and apply them in previously unseen contexts (Zhao et al., 2025), related work has started to investigate Knowledge Graph Foundation Models (KGFMs) to learn complex reasoning capabilities from *training knowledge graphs* and apply them to previously unseen *test knowledge graphs* (Galkin et al., 2024; Zhang et al., 2025b; Huang et al., 2025).

These KGFMs exhibit several important advantages (Mao et al., 2024) like *(i) Broad applicability:* Zero-shot reasoning can be performed on unseen knowledge graphs. *(ii) Efficiency:* While the training of knowledge-graph embedding methods is computationally highly challenging, performing a single round of inferences with zero-shot reasoning scales well to huge knowledge graphs. *(iii) Effectiveness:* Reasoning patterns can be learned from rich knowledge graphs and applied to knowledge graphs that lack a rich ontology and densely linked entities.

Knowledge graph triples are expressed using symbols that denote entities and relations. While logical and similarity-based reasoning do not require human-understandable symbols, de Rooij et al. (2016) have shown that symbols are "often constructed mnemonically, in order to be meaningful to a human interpreter" and have established for over 500k datasets that there is significant mutual information between the symbols and their formal meaning.

Though related work has used word embeddings of knowledge graph symbols and other attributed texts (e.g. comments) when embedding individual knowledge graphs (Nayyeri et al., 2023; Yuan et al., 2025; Yao et al., 2019; Wei et al., 2023; Xu et al., 2021), KGFMs have so far neglected the ca-

°Core Contributors.

pabilities of symbols and other text attributions to generalize across knowledge graphs.

Generalizing within-knowledge graph embedding methods to transfer embeddings from one or multiple knowledge graphs with millions of nodes to an unseen knowledge graph with largely distinct nodes can be hindered by a low signal-to-noise ratio (Zhao et al., 2021). However, we observe that knowledge graphs tend to have a huge number of nodes, but a much smaller number of highly informative relations. E.g., Wikidata (Vrandecic and Krötzsch, 2014) contains over 110 million nodes but only 12,681 relations as of May 19, 2025.

Therefore, we introduce SEMMA, which extends the capability of KGFMs to learn foundational knowledge graph semantics from the training knowledge graphs and apply it for reasoning on test knowledge graphs by analyzing (i) graph patterns **and** (ii) word embeddings of relation symbols.

As illustrated in Fig. 1, our approach constructs: (i) a *structural relation graph*, which represents patterns that generalize from graph structures of the training graphs, similar to ULTRA (Galkin et al., 2024), and (ii) a *textual relation graph*, which represents patterns that are found by prompting an LLM to generate semantically rich descriptions for each relation identifier and/or attributed labels. Together, the two relation graphs allow for predicting high-quality links that would not be found by existing KGFM methods. In summary, our contributions are:

- We introduce SEMMA, a novel KGFM designed to perform zero-shot link prediction by learning from graph structures and the word embeddings of relation identifiers (and/or related text attributions).
- Extensive experiments on 54 diverse knowledge graphs demonstrating SEMMA's superior performance over KGFM baselines like ULTRA in fully inductive link prediction. Importantly, after identifying and mitigating data leakages, SEMMA's advantage remains, highlighting its robust generalization capabilities.

2 Preliminaries

Knowledge Graphs. KGs represent knowledge in a directed edge-labeled graph.

Definition 1 (Knowledge Graphs)**.** *A Knowledge Graph $\mathcal{G}$ is a structure $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{F})$, where $\mathcal{E}$ is a set of entities, $\mathcal{R}$ is a set of relation identifiers, and $\mathcal{F} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is a set of facts.*

A fact $(h, r, t) \in \mathcal{F}$ (also referred to as a *triple*) connects its head entity h with its tail entity t via the relation identifier r.

Link prediction. We aim at link prediction, a fundamental task for KG reasoning. We assume that one knowledge graph $\mathcal{G}_{\text{T}} = (\mathcal{E}_{\text{T}}, \mathcal{R}_{\text{T}}, \mathcal{F}_{\text{T}})$ describes the set of facts that are true, another knowledge graph $\mathcal{G}_{\text{O}} = (\mathcal{E}_{\text{O}}, \mathcal{R}_{\text{O}}, \mathcal{F}_{\text{O}})$ represents the set of facts that have been observed and are known to be true, and $\mathcal{E}_{\text{O}} \subseteq \mathcal{E}_{\text{T}}, \mathcal{R}_{\text{O}} \subseteq \mathcal{R}_{\text{T}}, \mathcal{F}_{\text{O}} \subseteq \mathcal{F}_{\text{T}}$.

Given the observed graph $\mathcal{G}_{\text{O}}$, link prediction aims to infer missing true triples $\mathcal{F}_{\text{T}} \setminus \mathcal{F}_{\text{O}}$ (called queries) using triples with either the head or tail entity masked, denoted as $(h, r, ?)$ and $(?, r, t)$. We assume that for every relation, denoted by r, there is an inverse relation, denoted by r^{-1}. Let $\mathcal{G}_{\text{TRAIN}} = (\mathcal{E}_{\text{TRAIN}}, \mathcal{R}_{\text{TRAIN}}, \mathcal{F}_{\text{TRAIN}})$ correspond to the training graph and $\mathcal{G}_{\text{TEST}} = (\mathcal{E}_{\text{TEST}}, \mathcal{R}_{\text{TEST}}, \mathcal{F}_{\text{TEST}})$ denote the graph during inference, where $\mathcal{G}_{\text{TRAIN}} \cup \mathcal{G}_{\text{TEST}} = \mathcal{G}_{\text{O}}$. The task falls into three regimes, based on the overlap between $\mathcal{G}_{\text{TRAIN}}$ and $\mathcal{G}_{\text{TEST}}$:

- **Transductive**, in which the sets of entities and relation identifiers are fully shared ($\mathcal{G}_{\text{TRAIN}} = \mathcal{G}_{\text{TEST}}$). Traditional KG embedding models like TransE (Bordes et al., 2013), RotatE (Sun et al., 2019), and ComplEx (Trouillon et al., 2016) learn embeddings for all entities and relations under this assumption. These models cannot predict links that involve entities or relations not observed and used for generating the embedding.
- **Partially Inductive.** In this regime, all relation identifiers are known at training time ($\mathcal{R}_{\text{TRAIN}} = \mathcal{R}_{\text{TEST}}$), but not all the entities ($\mathcal{E}_{\text{TRAIN}} \subset \mathcal{E}_{\text{TEST}}$). Inductive Graph Neural Networks (GNNs) or rule mining methods like NBFNet (Zhu et al., 2022) and GraIL (Teru et al., 2020) learn to generalize in order to predict links involving entities from $\mathcal{E}_{\text{T}} \setminus \mathcal{E}_{\text{O}}$. However, they cannot predict links involving unseen relation identifiers.
- **Fully Inductive.** In this most general regime, neither entities nor relations are fully known during training ($\mathcal{E}_{\text{TRAIN}} \subset \mathcal{E}_{\text{TEST}}$ and $\mathcal{R}_{\text{TRAIN}} \subset \mathcal{R}_{\text{TEST}}$). This regime best reflects the link prediction capabilities expected from a foundation model, namely, the ability to apply its pre-trained knowledge to entirely new, previously unseen KGs at test time. KGFMs, such as ULTRA (Galkin et al., 2024), TRIX (Zhang et al., 2025b) and MOTIF

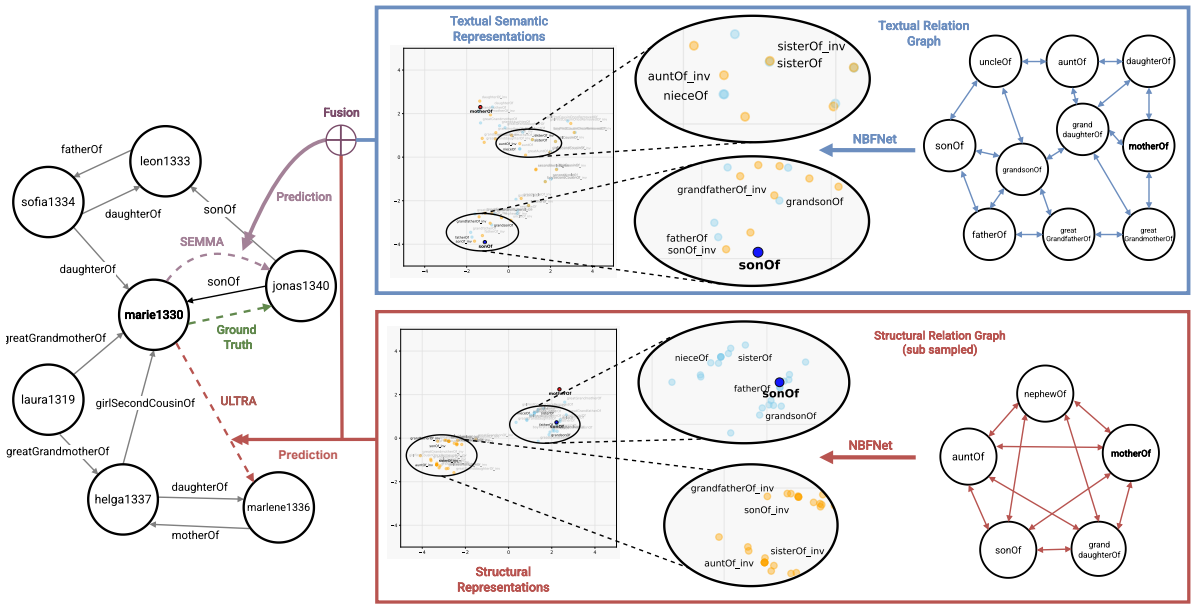

Figure 1: **SEMMA's advantage in link prediction. (Left)** SEMMA correctly predicts *(marie1330, motherOf, jonas1340)* from Metafam (Zhou et al., 2023) where ULTRA fails. Blue nodes correspond to relations, and orange nodes to their inverses. **(Right)** The Textual Relation Graph (top) is more ontologically coherent than the Structural Relation Graph (bottom), which is a clique, leading to different embeddings. **(Middle)** Textual semantic embeddings (top) show *fatherOf* is near *sonOf_inv* (equivalence) and *sisterOf* overlaps with *sisterOf_inv* (symmetry and inversion leading to equivalence), reflecting semantic understanding. Meanwhile, structural embeddings lack this clear organization for these pairs.

(Huang et al., 2025) learn widely transferable structural patterns.

Text-Attributed Knowledge Graphs. Virtually all real-world knowledge graphs associate textual attributions with entities and relation identifiers.

Definition 2 (Text-Attributed Knowledge Graphs)**.** *A Text-Attributed Knowledge Graph (TAKG) is a structure* $\mathcal{G} = (\mathcal{E}, \mathcal{R}, l, \mathcal{F})$*, with a knowledge graph* $(\mathcal{E}, \mathcal{R}, \mathcal{F})$ *and a labeling function* $l : \mathcal{E} \cup \mathcal{R} \to \Sigma^*$*, where* Σ^* *denotes strings of arbitrary finite length over an alphabet* Σ*. Let* $\mathcal{T}_\mathcal{E} = \{l(e) \mid e \in \mathcal{E}\}$ *and* $\mathcal{T}_\mathcal{R} = \{l(r) \mid r \in \mathcal{R}\}$ *be the set of entity and relation labels.*

For example, Wikidata (Vrandecic and Krötzsch, 2014) contains the triple (`Q42 P27 Q145`), with l(`Q42`) = "Douglas Adams", l(`P27`) = "country of citizenship", and l(`Q145`) = "United Kingdom". All of the 57 datasets from the benchmark for this task (see Section 5.1) attribute textual information. While some real-world TAKGs, like Wikidata, exhibit even more complex graph structures (e.g., involving time) and text attributes (e.g., multi-lingual), our definitions allow for a sufficiently expressive investigation of link prediction models in all of them.

To the best of our knowledge, the state-of-the-art in link prediction has either considered textual attributions but ignored the fully inductive regime (Yuan et al., 2025; Yao et al., 2019; Wei et al., 2023; Xu et al., 2021), or it has considered the fully inductive regime but has ignored text information (Lee et al., 2023; Zhu et al., 2022; Teru et al., 2020). Our main hypothesis is that leveraging textual attributes can enhance fully inductive link prediction. The closest work to ours is PROLINK (Wang et al., 2024), focused on working in the low-resource setting where the query relations occur sparsely in $\mathcal{G}_\text{O}$. PROLINK prompts LLMs to predict potential entity connections for unseen relations during testing, aiming to mitigate data sparsity. However, both their motivation and final evaluation setup differ from ours.

3 Foundation Models for KGs

Unlike traditional KGE models trained on a single graph, a KG foundation model is pre-trained on a large and diverse collection of KGs, collected in $\mathcal{G}_\text{TRAIN}$. The objective is to learn transferable patterns that enable zero-shot generalization to new, unseen KGs at inference time without requiring fine-tuning on $\mathcal{G}_\text{TEST}$.

ULTRA. ULTRA (Galkin et al., 2024) is a KGFM

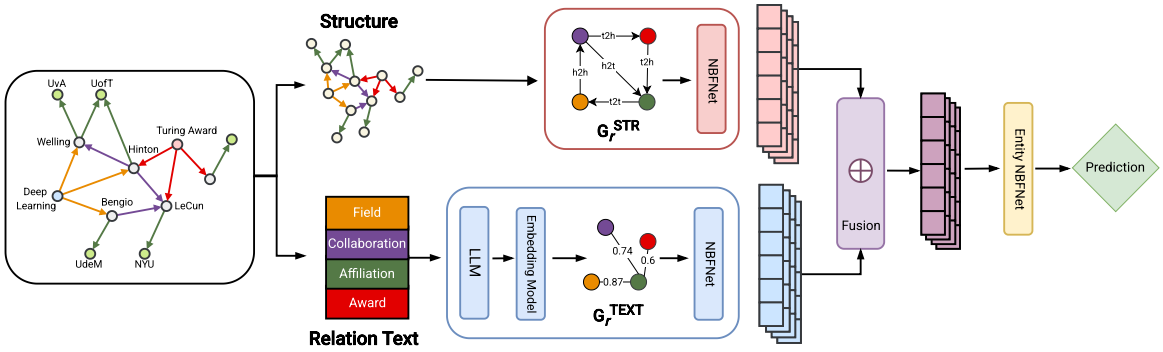

Figure 2: **Parallel Architecture of SEMMA.** The *structure processing module* (red) utilizes $\mathcal{G}_R^{\text{STR}}$ and NBFNet to derive structure-based representations, similar to ULTRA. Concurrently, the *text processing module* (blue) leverages LLM enrichment of relation text and $\mathcal{G}_R^{\text{TEXT}}$ with its own NBFNet to produce textual semantic representations. These two representations are fused and fed into an entity-level NBFNet to perform link prediction based on the input query.

designed for fully inductive link prediction, capable of zero-shot generalization to unseen KGs. However, it does not exploit textual attributions. ULTRA learns transferable structural patterns based on how relations interact within a graph, but it does not learn embeddings for specific relation types. Theoretically grounded by the principle of double equivariance (Zhou et al., 2025), ULTRA lifts the training and test graph $\mathcal{G}_{\text{O}}$ to a higher-level structure, which we refer to as $\mathcal{G}_R^{\text{STR}}$. In $\mathcal{G}_R^{\text{STR}}$, each node corresponds to a unique relation type or its inverse from $\mathcal{G}_{\text{O}}$. Edges in $\mathcal{G}_R^{\text{STR}}$ capture fundamental structural interactions between pairs of relations in $\mathcal{G}_{\text{O}}$, categorized into head-to-head (h2h), tail-to-head (t2h), head-to-tail (h2t), tail-to-tail (t2t). $\mathcal{G}_R^{\text{STR}}$ represents the transferable structural aspect, independent of the specific entity or relation vocabulary. ULTRA utilizes a GNN, specifically NBFNet (Zhu et al., 2022), over $\mathcal{G}_R^{\text{STR}}$, leveraging a labeling trick (Zhang et al., 2021) to obtain conditional relation representations. These conditional relation representations are used as input features for a second NBFNet operating on the original graph $\mathcal{G}_{\text{O}}$ to perform the final link prediction. By learning relative structural patterns via $\mathcal{G}_R^{\text{STR}}$ and conditioning representations on the query, ULTRA avoids learning fixed relation embeddings, enabling its zero-shot transfer capabilities. TRIX (Zhang et al., 2025b) and MOTIF (Huang et al., 2025) are follow-up works that enhanced ULTRA's representation power by leveraging extra memory and compute.

What do they lack? A key limitation of current KGFMs like ULTRA, as discussed in Section 1, is their sole reliance on graph structure, often neglecting or underutilizing the textual attributions commonly associated with entities and relations in most real-world KGs. These textual attributions, especially the semantics encoded in relation names or descriptions, can be crucial for accurate and generalizable reasoning but are often overlooked by structure-focused methods (Cornell et al., 2022; Alam et al., 2024). The generalization capabilities of LLMs have been widely recognized and leveraged in other domains (Li et al., 2024a), yet their potential to enrich KGFMs by interpreting textual attributes remains largely unexplored.

4 SEMMA: Integrating Structure and Textual Semantics

What do we want from a KGFM for TAKGs? This gap motivates the need for KGFMs specifically designed for TAKGs. In response, we propose SEMMA and establish the following desiderata:

1 **Explicitly Leverage Textual Attributions.** The model should effectively utilize the textual attributions present in KGs, when it is available.

2 **Maintain Robustness.** When textual attributions are not available, are noisy, or provide little semantic value, the model's performance should gracefully degrade to be at least as good as strong structure-only baselines like ULTRA.

Transferability of Ontological information. There are some key considerations to keep in mind while using the textual attributions in TAKGs,

1. The usefulness of text attributions for link prediction varies across different datasets and even within a single dataset. Relation names in $\mathcal{T}_{\mathcal{R}}$ might range from highly descriptive natural language phrases (e.g., *country of citizenship*) to

Model	Variant	Inductive e, r (23 graphs)		Inductive e (18 graphs)		Transductive (13 graphs)		Total Avg (54 graphs)	
		MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10
ULTRA		0.344	0.511	0.428	0.570	**0.316**	0.464	$0.365_{\pm 0.003}$	$0.519_{\pm 0.004}$
SEMMA		**0.350**	**0.514**	**0.447**	**0.584**	**0.316**	**0.467**	$\mathbf{0.374_{\pm 0.003}}$	$\mathbf{0.526_{\pm 0.004}}$
SEMMA	HYBRID	0.353	0.519	0.448	0.585	0.318	0.470	$0.376_{\pm 0.003}$	$0.529_{\pm 0.003}$

Table 1: **Zero-shot results of SEMMA.** Zero-shot link prediction MRR and Hits@10 reported over 54 KGs averaged over 5 runs. SEMMA outperforms ULTRA by considerable margins and SEMMA HYBRID increases the gap further.

database paths (e.g., */film/film/genre*). Consequently, the semantic value inferable from this text can vary significantly.

2. While entity text ($\mathcal{T}_\mathcal{E}$) may sometimes be limited to identifiers or obfuscated for privacy, relations ($\mathcal{R}$) fundamentally require some semantic basis to be meaningful within a knowledge graph. Hence, we constrain our approach to only leverage the relation text/labels ($\mathcal{T}_\mathcal{R}$). We assume that some textual representation for relations ($\mathcal{T}_\mathcal{R}$) generally exists, even if it's merely an ID.

Our approach. To address the established desiderata, we propose the SEMMA architecture (Fig. 2), which explicitly separates structural processing (as in ULTRA) from a module dedicated to processing relation's textual semantics derived from LLMs in parallel. This dual-module approach provides modularity, allowing the text processing module to be deactivated when the textual attributions are noisy or unhelpful, thus maintaining robust performance (2). When suitable, the modules merge to effectively leverage structural and textual modalities (1).

We modularize our pipeline into three independent components: utilizing LLMs to generate semantically rich text for each relation (Section 4.1), embedding the extracted text attributions into a vector space (Section 4.2), and using the embeddings to incorporate the semantics of text attributions into the prediction pipeline (Section 4.3).

4.1 Extracting textual semantic information from LLMs

The text processing module utilizes LLMs to process raw relation labels ($\mathcal{T}_\mathcal{R}$) and generate richer textual representations suitable for subsequent embedding. We leverage the inherent world knowledge and zero-shot capability of LLMs (Li et al., 2024b; Petroni et al., 2019) to interpret relation semantics even for previously unseen KGs.

What information can we extract from LLMs? For each relation identifier in $\mathcal{T}_\mathcal{R}$, we prompt an LLM to generate two outputs:

1. **Cleaned Relation Name.** A cleaned name without any special characters for semantically accurate tokenization (Kudo and Richardson, 2018). E.g., *greatGranddaughterOf* to *Great Granddaughter Of*.

2. **Concise Textual Description.** A short definition explaining the core meaning or function of the relation and its inverse (Ding et al., 2024). E.g., *greatGranddaughterOf* to *Female great-grandchild of*, providing deeper contextual understanding.

How do we extract this information from LLMs? We employ zero-shot prompting with illustrative examples and strict output constraints. This guides the LLM to produce consistent, reliable, and semantically meaningful text suitable for embedding across diverse relation inputs. More details about the prompt are provided in Appendix A.

4.2 Extracting embeddings from LLMs

Once we have the enriched textual representations from Section 4.1, we encode them into vector embeddings using transformers that capture their semantic meaning. These embeddings are the primary input for the text processing module.

Which text representation yields the most suitable embeddings? The process in Section 4.1 yields multiple textual candidates for each relation: (1) the original identifier (REL_NAME), (2) the LLM-generated cleaned name (LLM_REL_NAME), and (3) the LLM-generated textual description (LLM_REL_DESC). Backed by the theory of concepts in the representation space (Park et al., 2025), we also create COMBINED_SUM, where the embeddings of all the above are combined through vector addition, and COMBINED_AVG, where they are averaged in the vector space. We denote the text embedding of a relation r by τ_r, which can be obtained by one of the above-mentioned approaches.

We evaluate SEMMA with all of the mentioned five variants, with more details in Section 5.3.

How do we embed inverse relations? For each relation r, ULTRA generates its virtual inverse relation, denoted by r^{-1}, without its ground-truth text. Since we cannot embed the inverse relations without text (except in LLM_REL_DESC), we require a method to derive their semantic embeddings. Intuitively, if the original relation embedding signifies a transformation in a certain conceptual direction in the embedding space, its inverse should logically point the other way (Sun et al., 2019). So, given a relation's embedding τ_r, we generate the inverse relation's embedding $\tau_{r^{-1}}$ by rotating the original vector by 180 degrees to address this, i.e., $\tau_{r^{-1}} = (\mathbf{I} - 2\frac{\tau_r \tau_r^T}{\|\tau_r\|^2})\tau_r$ which simplifies to $\tau_{r^{-1}} = -\tau_r$. While theoretically imperfect for symmetric relations (which we observe to be rare in practice), this method is a reasonable heuristic and proves empirically effective (seen in Table 5).

4.3 Utilizing the embeddings

The final stage integrates the textual relation embeddings (from Section 4.2) with the structural information processed by SEMMA in parallel.

How do we represent inter-relation semantics? We construct a Textual Relation Graph ($\mathcal{G}_R^{\text{TEXT}}$) where nodes are relations and weighted edges (with weights ω) represent the cosine similarity between their textual embeddings, to explicitly model semantic proximity between relations. An edge in $\mathcal{G}_R^{\text{TEXT}}$ is denoted by a triple (u, ω, v) where u, v are two relations with $\omega = \cos(\tau_u, \tau_v)$. For efficiency and to avoid oversmoothing of representations, we filter the edges $\mathcal{G}_R^{\text{TEXT}}$ based on edge weights (ω) using sampling methods like choosing the Top-x% and thresholding (Putra and Tokunaga, 2017).

How are textual and structural information processed? We utilize NBFNet (Zhu et al., 2022) to process both the structural relation graph $\mathcal{G}_R^{\text{STR}}$ and the textual relation graph $\mathcal{G}_R^{\text{TEXT}}$ in parallel. Importantly, the text processing module leverages NBFNet's support for weighted edges, using the cosine similarities (w) in $\mathcal{G}_R^{\text{TEXT}}$ to modulate message passing based on semantic relatedness.

Formally, let $\mathbf{h}_u^{(t)}, \mathbf{z}_u^{(t)}$ denote the state of relation node u in $\mathcal{G}_R^{\text{STR}}$ and $\mathcal{G}_R^{\text{TEXT}}$ respectively at iteration t (conditioned on query relation r). Let UPD, MSG, $\bigoplus$ denote the NBFNet update, message, and aggregation functions. Let $\mathbf{e}_{r'}$ denote the embedding of the edge type $r' \in \{h2h, t2h, h2t, t2t\}$ in $\mathcal{G}_R^{\text{STR}}$. $\mathbf{h}_u^{(0)}$ is initialized as a $\mathbb{1}_d$ for r and 0_d for the rest of the relations. $\mathbf{z}_u^{(0)}$ are initialized with the respective embeddings after LLM enrichment, i.e., $\mathbf{z}_u^{(0)} = \tau_u$. Then the node embeddings are updated as follows,

$$\mathbf{h}_u^{(t+1)} = \text{UPD}\left(\mathbf{h}_u^{(t)}, \bigoplus_{\substack{v \in \mathcal{N}(u) \\ (u,r',v) \in \mathcal{G}_R^{\text{STR}}}} \text{MSG}(\mathbf{h}_v^{(t)}, \mathbf{e}_{r'})\right)$$

$$\mathbf{z}_u^{(t+1)} = \text{UPD}\left(\mathbf{z}_u^{(t)}, \bigoplus_{\substack{v \in \mathcal{N}(u) \\ (u,\omega,v) \in \mathcal{G}_R^{\text{TEXT}}}} \text{MSG}(\mathbf{z}_v^{(t)}, \omega)\right)$$

How are structural and textual signals fused? The outputs from the parallel structural ($\mathcal{G}_R^{\text{STR}}$) and textual ($\mathcal{G}_R^{\text{TEXT}}$) NBFNet modules are combined to produce the final embeddings. We initially explored simple concatenation for merging the structural ($\mathbf{h}_u$) and textual signals ($\mathbf{z}_u$), but this more than doubled the model parameters to handle the extra dimensions and also resulted in significantly longer convergence times. Hence, we evaluate standard fusion techniques (F) like MLP and Attention to reduce them back to the original dimensions. Formally, let n denote the final iteration,

$$\mathbf{H}_u^{(n)} = F(\mathbf{h}_u^{(n)} \oplus (\alpha \mathbf{z}_u^{(n)} + 0_d)),\ \alpha \in \{0, 1\}$$

where $\oplus$ is the concatenation operator and α is a hyperparameter that disables the text processing module if set to zero ($\alpha = 1$ by default). This is then passed onto the Entity NBFNet for final prediction. We use the same training objective as ULTRA for the whole pipeline.

5 Experiments

Through our experiments, we address the following questions: **RQ1:** Where does textual semantics matter? **RQ2:** Does adding textual semantics increase average performance? **RQ3:** Can textual semantics help generalize to newer, harder settings? **RQ4:** Do the existing benchmarks suffice?

5.1 Experimental Setup

Datasets, Baseline, and Setup. We use the same setup as ULTRA, pretraining on 3 datasets and testing on 54. They are categorized into three categories: Transductive, Partially Inductive (Inductive e), and Fully Inductive (Inductive e, r), depending on the overlap between $\mathcal{G}_{\text{TRAIN}}$ and $\mathcal{G}_{\text{TEST}}$ as discussed in Section 2. We use ULTRA as the main baseline for fair comparison. We leave it for future work to extend this to TRIX and MOTIF, where our framework can be adapted to fit in their pipeline.

Model	Variant	Inductive e, r (8 graphs)		Inductive e (7 graphs)		Transductive (7 graphs)		Total Avg (22 graphs)	
		MRR	**H@10**	**MRR**	**H@10**	**MRR**	**H@10**	**MRR**	**H@10**
ULTRA		0.388	0.585	0.347	0.475	**0.300**	0.438	$0.347_{\pm 0.006}$	$0.503_{\pm 0.010}$
SEMMA		**0.399**	**0.589**	**0.357**	**0.482**	0.295	**0.441**	$\mathbf{0.353_{\pm 0.006}}$	$\mathbf{0.508_{\pm 0.009}}$
SEMMA	HYBRID	0.406	0.600	0.357	0.484	0.298	0.445	$0.356_{\pm 0.007}$	$0.514_{\pm 0.010}$

Table 2: **Zero-shot results of SEMMA on unleaked datasets.** Zero-shot link prediction MRR and Hits@10 reported over 22 KGs averaged over 5 runs after removing leaked datasets from testing. SEMMA still outperforms ULTRA.

Dataset	MRR	H@10
Metafam	+0.3303	+0.2935
WN18RRInductive:v1	+0.1058	+0.0550
ConceptNet100k	+0.0620	+0.1188
YAGO310	−0.0879	−0.0626
Hetionet	−0.0556	−0.0487
WikiTopicsMT4:sci	−0.0364	−0.0709

Table 3: **Maximum and Minimum improvements of SEMMA.** SEMMA has significant increases on some datasets while there are drops in a few others.

SEMMA is relatively small and has around 227k parameters. All experiments were run on 2 NVIDIA A100 GPUs. More details in Appendices B and D.

Metrics. We report the Mean Reciprocal Rank (**MRR**) and Hits@10 (**H@10**) averaged across the categories and over all the datasets. The reported values are averaged across 5 runs, along with the standard deviation values.

Design Choices. We use gpt-4o-2024-11-20 (OpenAI et al., 2024) as the LLM, jina-embeddings-v3 (Sturua et al., 2024) as the text embedding model, COMBINED_SUM for combining the embeddings and a threshold of 0.8 to construct $\mathcal{G}_R^{\text{TEXT}}$ based on insights from (Putra and Tokunaga, 2017) and MLP for fusing $\mathcal{G}_R^{\text{STR}}$ and $\mathcal{G}_R^{\text{TEXT}}$. Ablations of these choices are discussed in Section 5.3.

5.2 Where does textual semantics matter?

Before looking at the average numbers across categories, we first analyse the impact of SEMMA on the dataset level. From Table 3, we can observe that for datasets like Metafam, the performance increase is almost 2x, while there are some datasets like YAGO310, where there is a considerable drop in MRR and H@10. We provide more insights on this in Appendix E. This hints at how textual semantics can be helpful for some datasets, subject to how rich the relation texts are. To leverage this insight, we introduce the SEMMA HYBRID variant, where we switch off the text processing module based on the validation set's performance.

5.3 Does adding textual semantics increase average performance?

From Table 1, we can see that SEMMA clearly outperforms ULTRA averaged across all 54 datasets. MRR has an increase of 9 points, and H@10 has an increase of 7 points on average. Note that in the domain of KGFMs, increases of this magnitude are significant, as the tasks are quite demanding (Zhang et al., 2025b; Huang et al., 2025). We also perform the Mann-Whitney U test to confirm the statistical significance of the performance gains of SEMMA. Total Avg MRR has a p-value of 0.0040 (< 0.05) across the runs, and Total Avg Hits@10 has a p-value of 0.0278 (< 0.05). We provide dataset-wise results in Appendix G.

> **Takeaway 1.** Textual attributions can add value to link prediction compared to purely structural information. Therefore, SEMMA can outperform ULTRA by a considerable margin.

How robust is SEMMA to design choices? We ablate all the components of SEMMA to study the impact of the design choices we made. We experiment with other cheaper open-weight LLMs: DeepSeek V3 (DeepSeek-AI et al., 2025) and Qwen3 32b (Yang et al., 2025), SentenceBERT (Reimers and Gurevych, 2019) as the embedding model, and other choices described in Section 4.3 and Section 4.2. Results from Table 5 reinforce our final design choices, with regime-wise results in Table 10.

5.4 Can textual semantics help generalize to newer, harder settings?

Similar to the low-resource setting proposed by Wang et al. (2024), we create a harder evaluation setting where the relation vocabulary of the queries is disjoint from $\mathcal{R}_{\text{TEST}}$, inspired by time-evolving KGs (Cai et al., 2024) where new relations are po-

Dataset	Split Statistics		MRR		Hits@3		Hits@10	
	$\lvert\mathcal{F}_{\text{TEST}}\rvert$	$\lvert\mathcal{F}_{\text{T}} \setminus \mathcal{F}_{\text{O}}\rvert$	ULTRA	SEMMA	ULTRA	SEMMA	ULTRA	SEMMA
FBNELL	5750	335	0.025	**0.058**$_{(+128\%)}$	0.035	**0.067**$_{(+87\%)}$	0.064	**0.135**$_{(+112\%)}$
Metafam	5900	177	0.098	**0.180**$_{(+83\%)}$	0.079	**0.206**$_{(+161\%)}$	0.138	**0.338**$_{(+145\%)}$

Table 4: **SEMMA vs ULTRA evaluated on a harder setting with inductive relation vocabulary.** SEMMA, being more robust to new relation vocabulary, outperforms ULTRA by almost 2x on all the metrics.

	Design choices	Total Avg (54 graphs)	
		MRR	H@10
LLMs	gpt-4o-2024-11-20	**0.377**	0.529
	deepseek-chat-v3-0324	0.375	**0.530**
	qwen3-32b	0.368	0.519
LM	jina-embeddings-v3	**0.377**	**0.529**
	Sentence-BERT	0.368	0.528
F	MLP	**0.377**	**0.529**
	Attention	0.369	0.525
Text	REL_NAME	0.369	0.528
	LLM_REL_NAME	0.375	0.529
	LLM_REL_DESC	0.373	0.524
	COMBINED_SUM	**0.377**	**0.529**
	COMBINED_AVG	0.374	0.523
$\mathcal{G}_R^{\text{TEXT}}$	Threshold (0.8)	**0.377**	**0.529**
	Top-x% (20%)	0.371	0.525

Table 5: **Ablations of Design Choices.** We conduct a rigorous search over settings for SEMMA.

tentially introduced over time. We discuss this setting in detail in Appendix F. We adapt ULTRA and SEMMA to work in this setting by generating a new $\mathcal{G}_R^{\text{STR}}$ and $\mathcal{G}_R^{\text{TEXT}}$ for each query. For queries that share the same head but have different relations, the corresponding $\mathcal{G}_R^{\text{STR}}$ will be identical, causing ULTRA to produce the same predictions for all such queries, whereas $\mathcal{G}_R^{\text{TEXT}}$ will be able to distinguish them based on the relation identifier. For a working example, refer to Fig. 8. As observed in Table 4, ULTRA's performance drops substantially in the new setting where SEMMA is 2x better.

Takeaway 2. The limitation of purely structural approaches like ULTRA becomes evident when query relations are disjoint from $\mathcal{R}_{\text{TEST}}$, where SEMMA still performs competently.

5.5 Do the existing benchmarks suffice?

In some datasets, we identified an overlap where triples from the pretraining data appeared in the test data. We call this common set of triples, the "leaked set". To assess the impact of this, we conduct a study where we remove all datasets exhibiting any leakage from our evaluation and report these "unleaked" results in Table 2, with more dataset-wise leakage statistics reported in Appendix C. From the results in Table 2, SEMMA maintains its superior performance over ULTRA on this cleaner, more challenging subset of 22 KGs. However, this analysis also highlights a broader issue: the current KGFM benchmarks, while diverse in some structural aspects, often draw from a limited set of popular, large-scale KGs (such as Freebase and WordNet) for both pretraining and testing. This homogeneity can lead to an overestimation of true generalization to genuinely novel domains and ontologies not represented in these common sources.

Takeaway 3. We need new benchmarks in the field that are from diverse domains and not derived from the same popular KGs to evaluate the actual effectiveness of KGFMs.

6 Conclusion and Future Directions

We introduce SEMMA, a KGFM that utilizes LLM-derived textual semantics for relations alongside structure. By incorporating a textual relation graph, SEMMA outperforms purely structural baselines such as ULTRA across 54 diverse KGs and shows strong generalization in harder settings with unseen relation vocabularies. This work is a first step toward semantically grounded KGFMs. While we focused on relation-level semantics, future works need to explore extending SEMMA to entity-level text, richer multilingual inputs, and integration into more expressive models like TRIX and MOTIF. Building on our findings regarding current dataset limitations, the domain also needs new benchmarks with KGs from genuinely diverse and unseen domains to rigorously evaluate true generalization.

Limitations

While SEMMA demonstrates a promising direction for KGFMs, several limitations and avenues for future work remain. Firstly, our current approach exclusively focuses on relation text, leaving the semantic information often present in entity names and descriptions untapped; future iterations should explore integration of both. Secondly, the quality of the textual representations is inherently tied to the capabilities and outputs of the upstream LLM used for enrichment. While there are no direct risks of our work, there is a possibility of biases present in LLMs leaking into our pipeline. Next, while SEMMA advances upon ULTRA, future works should investigate its semantic enrichment pipeline with more recent, expressive KGFMs (such as TRIX or MOTIF). There is also scope to explore more adaptive fusion mechanisms that dynamically weigh structural versus textual signals, potentially improving upon the current SEMMA HYBRID's validation-dependent switch, especially when relation text quality varies significantly. Finally, SEMMA only explores ontological concepts that can be modeled with textual semantic similarity; there is scope for future work to broaden the horizon.

Acknowledgments

The authors would like to thank Akshit Sinha for helpful feedback and help with the figures. AA was funded by the CHIPS Joint Undertaking (JU) under grant agreement No. 101140087 (SMARTY), and by the German Federal Ministry of Education and Research (BMBF) under the sub-project with the funding number 16MEE0444. AV was supported by the "UNREAL: Unified Reasoning Layer for Trustworthy ML" project (EP/Y023838/1) selected by the ERC and funded by UKRI EPSRC. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting AA. AA also thanks the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program for support. The authors gratefully acknowledge compute time on HoreKa HPC (NHR@KIT), funded by the BMBF and Baden-Württemberg's MWK through the NHR program, with additional support from the DFG; and on the Artificial Intelligence Software Academy (AISA) cluster funded by the Ministry of Science, Research and Arts of Baden-Württemberg.

Author contributions

AA conceived the project based on discussions with SS and led the overall design of SEMMA. AA led the setup and experiments of RQ1 (Section 5.2), RQ2 (Section 5.3) with the help of SK. AA and SK wrote the necessary code and ran experiments for all the RQs. AA and MN discovered the data leakage problem and, with the help of AV, proposed and ran experiments for RQ4 (Section 5.5) with help from SK. Based on an initial idea by BX, SK led the setup and experiments for RQ3 (Section 5.4) with the help of AA. AA wrote the initial draft with help from SK, where all other authors actively contributed to refining it. MN, BX, and AV actively advised on the design of all the experiments. SS and PK provided feedback and advice throughout the project.

References

Mehwish Alam, Frank van Harmelen, and Maribel Acosta. 2024. Towards semantically enriched embeddings for knowledge graph completion. *Preprint*, arXiv:2308.00081.

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*, volume 26. Curran Associates, Inc.

Li Cai, Xin Mao, Yuhao Zhou, Zhaoguang Long, Changxu Wu, and Man Lan. 2024. A survey on temporal knowledge graph: Representation learning and applications. *Preprint*, arXiv:2403.04782.

Yihong Chen, Pasquale Minervini, Sebastian Riedel, and Pontus Stenetorp. 2021. Relation prediction as an auxiliary training objective for improving multi-relational graph representations. *Preprint*, arXiv:2110.02834.

Filip Cornell, Chenda Zhang, Jussi Karlgren, and Sarunas Girdzijauskas. 2022. Challenging the assumption of structure-based embeddings in few- and zero-shot knowledge graph completion. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6300–6309, Marseille, France. European Language Resources Association.

Steven de Rooij, Wouter Beek, Peter Bloem, Frank van Harmelen, and Stefan Schlobach. 2016. Are names meaningful? quantifying social meaning on the semantic web. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15*, pages 184–199. Springer.

DeepSeek-AI, Aixin Liu, Bei Feng, et al. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. *Preprint*, arXiv:1707.01476.

Boyang Ding, Quan Wang, Bin Wang, and Li Guo. 2018. Improving knowledge graph embedding using simple constraints. *Preprint*, arXiv:1805.02408.

Zifeng Ding, Heling Cai, Jingpei Wu, Yunpu Ma, Ruotong Liao, Bo Xiong, and Volker Tresp. 2024. zr-llm: Zero-shot relational learning on temporal knowledge graphs with large language models. *Preprint*, arXiv:2311.10112.

Matthias Fey, Vid Kocijan, Francisco Lopez, and Jure Leskovec. 2025. Introducing kumorfm: A foundation model for in-context learning on relational data. Accessed: 2025-05-24.

Mikhail Galkin, Max Berrendorf, and Charles Tapley Hoyt. 2022. An open challenge for inductive link prediction on knowledge graphs. *Preprint*, arXiv:2203.01520.

Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2024. Towards foundation models for knowledge graph reasoning. *Preprint*, arXiv:2310.04562.

Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. 2018. Knowledge base completion with out-of-knowledge-base entities: A graph neural network approach. *Transactions of the Japanese Society for Artificial Intelligence*, 33(2):F–H72_1–10.

Daniel Scott 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:e26726.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José 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. *ACM Computing Surveys*, 54(4):1–37.

Xingyue Huang, Pablo Barceló, Michael M. Bronstein, İsmail İlkan Ceylan, Mikhail Galkin, Juan L Reutter, and Miguel Romero Orth. 2025. How expressive are knowledge graph foundation models? *Preprint*, arXiv:2502.13339.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. 2022. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jaejun Lee, Chanyoung Chung, and Joyce Jiyoung Whang. 2023. Ingram: inductive knowledge graph embedding via relation graphs. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, Bin Xu, Ren Bowen, Chong Feng, Yang Gao, and Heyan Huang. 2024a. Fundamental capabilities of large language models and their applications in domain scenarios: A survey. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11116–11141, Bangkok, Thailand. Association for Computational Linguistics.

Mingchen Li, Chen Ling, Rui Zhang, and Liang Zhao. 2024b. Zero-shot link prediction in knowledge graphs with large language models. In *2024 IEEE International Conference on Data Mining (ICDM)*, pages 753–760. IEEE.

Shuwen Liu, Bernardo Grau, Ian Horrocks, and Egor Kostylev. 2021. Indigo: Gnn-based inductive knowledge graph completion using pair-wise encoding. In *Advances in Neural Information Processing Systems*, volume 34, pages 2034–2045. Curran Associates, Inc.

Xin Lv, Xu Han, Lei Hou, Juanzi Li, Zhiyuan Liu, Wei Zhang, Yichi Zhang, Hao Kong, and Suhui Wu. 2020. Dynamic anticipation and completion for multi-hop reasoning over sparse knowledge graph. *Preprint*, arXiv:2010.01899.

Farzaneh Mahdisoltani, Joanna Asia Biega, and Fabian M. Suchanek. 2015. Yago3: A knowledge base from multilingual wikipedias. In *Conference on Innovative Data Systems Research*.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2019. Commonsense knowledge base completion with structural and semantic context. *Preprint*, arXiv:1910.02915.

Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and

Jiliang Tang. 2024. Position: graph foundation models are already here. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Mojtaba Nayyeri, Zihao Wang, Mst. Mahfuja Akter, Mirza Mohtashim Alam, Md. Rashad Al Hasan Rony, Jens Lehmann, and Steffen Staab. 2023. Integrating knowledge graph embeddings and pre-trained language models in hypercomplex spaces. In *ISWC*, pages 388–407.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrowand, et al. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. 2025. The geometry of categorical and hierarchical concepts in large language models. In *The Thirteenth International Conference on Learning Representations*.

Aleksandr Perevalov, Xi Yan, Liubov Kovriguina, Longquan Jiang, Andreas Both, and Ricardo Usbeck. 2022. Knowledge graph question answering leaderboard: A community resource to prevent a replication crisis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2998–3007, Marseille, France. European Language Resources Association.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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)*, pages 2463–2473.

Jan Wira Gotama Putra and Takenobu Tokunaga. 2017. Evaluating text coherence based on semantic similarity graph. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pages 76–85, Vancouver, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Tara Safavi and Danai Koutra. 2020. Codex: A comprehensive knowledge graph completion benchmark. *Preprint*, arXiv:2009.07810.

Harry Shomer, Jay Revolinsky, and Jiliang Tang. 2024. Towards better benchmark datasets for inductive knowledge graph completion. *Preprint*, arXiv:2406.11898.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *Preprint*, arXiv:2409.10173.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *Preprint*, arXiv:1902.10197.

Komal K. Teru, Etienne Denis, and William L. Hamilton. 2020. Inductive relation prediction by subgraph reasoning. *Preprint*, arXiv:1911.06962.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*, pages 57–66, Beijing, China. Association for Computational Linguistics.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. *Preprint*, arXiv:1606.06357.

Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Kai Wang, Siqiang Luo, Caihua Shan, and Yifei Shen. 2025. Towards graph foundation models: Training on knowledge graphs enables transferability to general graphs. *Preprint*, arXiv:2410.12609.

Kai Wang, Yuwei Xu, Zhiyong Wu, and Siqiang Luo. 2024. LLM as prompter: Low-resource inductive reasoning on arbitrary knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3742–3759, Bangkok, Thailand. Association for Computational Linguistics.

Yanbin Wei, Qiushi Huang, Yu Zhang, and James Kwok. 2023. KICGPT: Large language model with knowledge in context for knowledge graph completion. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8667–8683, Singapore. Association for Computational Linguistics.

Wenhan Xiong, Thien Hoang, and William Yang Wang. 2017. DeepPath: A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573, Copenhagen, Denmark. Association for Computational Linguistics.

Yichong Xu, Chenguang Zhu, Ruochen Xu, Yang Liu, Michael Zeng, and Xuedong Huang. 2021. Fusing context into knowledge graph for commonsense question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1201–1207, Online. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, et al. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *Preprint*, arXiv:1909.03193.

Jianxiang Yu, Jiapeng Zhu, Hao Qian, Ziqi Liu, Zhiqiang Zhang, and Xiang Li. 2025. Relation-aware graph foundation model. *Preprint*, arXiv:2505.12027.

Duanyang Yuan, Sihang Zhou, Xiaoshu Chen, Dong Wang, Ke Liang, Xinwang Liu, and Jian Huang. 2025. Knowledge graph completion with relation-aware anchor enhancement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(14):15239–15247.

Haonan Zhang, Dongxia Wang, Zhu Sun, Yanhui Li, Youcheng Sun, Huizhi Liang, and Wenhai Wang. 2025a. Kg4receval: Does knowledge graph really matter for recommender systems? *ACM Transactions on Information Systems*, 43(3):1–36.

Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. 2021. Labeling trick: a theory of using graph neural networks for multi-node representation learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Yucheng Zhang, Beatrice Bevilacqua, Mikhail Galkin, and Bruno Ribeiro. 2025b. Trix: A more expressive model for zero-shot domain transfer in knowledge graphs. *Preprint*, arXiv:2502.19512.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

Yu Zhao, Zhiquan Li, Wei Deng, Ruobing Xie, and Qing Li. 2021. Learning entity type structured embeddings with trustworthiness on noisy knowledge graphs. *Knowledge-Based Systems*, 215:106630.

Jincheng Zhou, Beatrice Bevilacqua, and Bruno Ribeiro. 2023. A multi-task perspective for link prediction with new relation types and nodes. *Preprint*, arXiv:2307.06046.

Jincheng Zhou, Yucheng Zhang, Jianfei Gao, Yangze Zhou, and Bruno Ribeiro. 2025. Double equivariance for inductive link prediction for both new nodes and new relation types. *Preprint*, arXiv:2302.01313.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. 2022. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Preprint*, arXiv:2106.06935.

A Prompt

We decompose the prompt into two steps - one to extract the cleaned relation names and the second to obtain the relation descriptions. We define clear rules for each of them and specify the output format. The exact prompt is listed in Fig. 5 and the system instruction in Fig. 3.

System Instruction

Provide exactly two separate JSON objects in your response, corresponding to each step, strictly in the order presented above. Do not include additional explanations or metadata beyond the specified JSON objects. Always provide both JSON objects and ensure they contain all the original relation names as keys.

Figure 3: System Instruction used to ensure output consistency.

We use the OpenRouter[1] API to query the LLMs. In total, the experiments were pretty cost-effective, we spent $8 on openai/gpt-4o-2024-11-20, $0.8 on deepseek/deepseek-chat-v3-0324, and $0.67 on qwen/qwen3-32b-04-28, totalling to less than $10 for all the 57 datasets cumulatively. We also attach a sample output from openai/gpt-4o-2024-11-20, which can be found at Fig. 4.

B Datasets

We follow the same evaluation strategy as ULTRA and conduct experiments on 57 publicly available KGs spanning a variety of sizes and domains. These datasets are grouped into three generalisation regimes: transductive, inductive with new entities (e), and inductive with new entities and relations (e, r) at test time. The detailed statistics for transductive datasets is presented in Table 6, inductive (e) in Table 7 and inductive (e, r) in Table 8. Consistent with ULTRA, we perform tail-only evaluation on FB15k237_10, FB15k237_20, and FB15k237_50, where predictions are restricted to the form $(h, r, ?)$ during test time.

C Leakage

During our evaluation of ULTRA, we discovered a notable source of information leakage arising from

[1] https://openrouter.ai/

Sample Output

```
"cleaned_relations": {
    "Causes": "Causes",
    "/organization/organization/headquarters./location/mailing_address/citytown":
        "Headquarters City",
    "GpMF": "Gene participates Molecular Function"
    ...
}

"relation_descriptions": {
    "Causes": ["leads to effect", "effect caused by"],
    "/organization/organization/headquarters./location/mailing_address/citytown":
        ["Headquarters located in city", "City has headquarters of"],
    "GpMF": ["Gene contributes to molecular function", "Molecular function involves gene"]
    ...
}
```

Figure 4: Sample outputs from GPT 4o.

the overlap between test sets and the pretraining corpus corroborating with findings from Shomer et al. (2024). Specifically, we found that 30 out of the 54 datasets used in the evaluation of ULTRA contain at least one triple from their test graph that is already present in the pretraining corpus. This overlap stems from the fact that many of the benchmark datasets in the 57-dataset corpus are derived from widely-used and well-known knowledge graphs such as WordNet, Freebase, Wikidata, and NELL, some of which also contribute to the pretraining data. We categorize this leakage into two distinct types:

1. **Test Graph Leakage (Indirect):** This occurs when one or more triples from $\mathcal{G}_{\text{TEST}}$ are already present in the pretraining corpus. Although the specific query triple may not be included, the model can potentially exploit these known graph facts to better predict test queries, thus providing an unfair advantage. This represents an indirect form of information leakage.

2. **Query Leakage (Direct):** This refers to a more explicit form of leakage where the exact test query triples themselves are present in the pretraining corpus. In such cases, the model may have already been trained to predict the ground truth, effectively reducing the evaluation to a memorization check rather than a generalization task.

We conduct a detailed analysis of both types of leakage and quantify their extent across each dataset. The results are presented in Fig. 6 and Fig. 7, which respectively depict the proportion of datasets affected by indirect (test graph) and direct (query triple) leakage. This analysis highlights the importance of careful dataset curation and motivates the need for evaluating models in settings where such overlaps are explicitly mitigated.

D Implementation Details

Codebase. Our implementation is based on the official ULTRA codebase,[2] which we extended to integrate the proposed dual-module architecture, including the LLM-based relation enrichment, $\mathcal{G}_R^{\text{TEXT}}$ construction, and the structural-textual fusion module. The full source code for SEMMA and a pretrained model checkpoint are available in the supplementary material.

Hyperparameters. For the structure processing module (NBFNet operating on $\mathcal{G}_R^{\text{STR}}$) and the final entity-level NBFNet, we largely retain the hyperparameter configurations reported by ULTRA (see Table 9 for a summary). Hyperparameters specific to SEMMA's novel components were determined as follows: the $\mathcal{G}_R^{\text{TEXT}}$ construction threshold was set to 0.8 based on ablation studies and Putra and Tokunaga (2017); the MLP for fusion (F) consists of one hidden layer of size 128 with ReLU activation, chosen based on preliminary experiments on a validation subset of the pre-training data. The choice of LLM (gpt-4o) and text embedding model (Jina-embeddings-v3) was also informed by our ab-

[2] https://github.com/DeepGraphLearning/ULTRA

Dataset	Reference	Entities	$\lvert\mathcal{R}\rvert$	Train	Valid	Test	Task
CoDExSmall	Safavi and Koutra (2020)	2034	42	32888	1827	1828	h/t
WDsinger	Lv et al. (2020)	10282	135	16142	2163	2203	h/t
FB15k237_10	Lv et al. (2020)	11512	237	27211	15624	18150	tails
FB15k237_20	Lv et al. (2020)	13166	237	54423	16963	19776	tails
FB15k237_50	Lv et al. (2020)	14149	237	136057	17449	20324	tails
FB15k237	Toutanova and Chen (2015)	14541	237	272115	17535	20466	h/t
CoDExMedium	Safavi and Koutra (2020)	17050	51	185584	10310	10311	h/t
NELL23k	Lv et al. (2020)	22925	200	25445	4961	4952	h/t
WN18RR	Dettmers et al. (2018)	40943	11	86835	3034	3134	h/t
AristoV4	Chen et al. (2021)	44949	1605	242567	20000	20000	h/t
Hetionet	Himmelstein et al. (2017)	45158	24	2025177	112510	112510	h/t
NELL995	Xiong et al. (2017)	74536	200	149678	543	2818	h/t
CoDExLarge	Safavi and Koutra (2020)	77113	69	511359	30622	30622	h/t
ConceptNet100k	Malaviya et al. (2019)	78334	34	100000	1200	1200	h/t
DBpedia100k	Ding et al. (2018)	99604	470	597572	50000	50000	h/t
YAGO310	Mahdisoltani et al. (2015)	123182	37	1079040	5000	5000	h/t

Table 6: Dataset-wise statistics for the 16 KGs evaluated under the transductive regime. **Entities** and $\lvert\mathcal{R}\rvert$ indicate the vocabulary sizes of entities and relations, respectively. **Train**, **Valid**, and **Test** represent the number of triples in each split. The **Task** column specifies whether the model is evaluated on both head and tail prediction (`h/t`) or only on tail prediction (`tails`), consistent with ULTRA's evaluation setup.

lations (Table 5). We only explore late fusion techniques as we want to reserve the option to switch off the text processing module.

Training & Complexity. SEMMA has a total of approximately 227k trainable parameters. The pre-training phase, conducted on the combined pre-training dataset of 500K samples, required approximately 9 hours on a single NVIDIA A100 GPU for 10 epochs. At inference time, the generation of LLM-enriched representations for the relation vocabulary $\mathcal{R}$ introduces an additional computational step with complexity proportional to $O(|\mathcal{R}| \times L_{\text{LLM}})$, where L_{LLM} is the cost of LLM processing per relation. The subsequent GNN inference on $\mathcal{G}_R^{\text{TEXT}}$ and $\mathcal{G}_R^{\text{STR}}$ remains efficient.

We also utilized AI assistants during the development cycle for tasks such as code debugging and suggesting alternative phrasing for manuscript clarity, ensuring the core contributions and final methodology remained our own.

E Performance variance in SEMMA

The observed performance variations of our approach across different KGs may be attributed to their intrinsic structural and textual properties. KGs such as Hetionet and YAGO310 seem to be carefully curated with standardized relational schemas, where each relation possesses precise semantics and consistent usage. In such highly regularized environments, the introduction of textual semantic similarity mechanisms, if not carefully calibrated, could potentially disrupt this precision, leading to an over-smoothing effect and a degradation in reasoning performance by failing to preserve the subtle semantic differences that distinguish various relations.

Conversely, datasets like ConceptNet, which incorporates commonsense knowledge, and Metafam, exhibit relations with comparatively looser standardization and greater linguistic variability. In these contexts, our textual semantic similarity approach appears to effectively identify important semantic parallels, thereby bridging linguistic variations and compensating for less formal relational definitions.

The usefulness of textual semantic similarity is also dependent upon the threshold used. A high threshold predominantly captures high-certainty similarities (e.g., exact synonymy, close paraphrases), while lower thresholds may encompass broader contextual or domain-specific similarities. This leads to a hypothesis: highly standardized KGs might benefit from a stringent threshold to preserve relational specificity, whereas KGs with more flexible relational definitions could achieve optimal performance with a more lenient threshold that accommodates linguistic diversity. Based on preliminary experiments, we also observed a correlation between the sparsity of the KG and the value of textual semantics. While this provides a high-level rationale, we acknowledge that other factors, such as the distribution of relation types and graph topology, likely influence the observed outcomes.

So, from these preliminary analyses, we hypothe-

Dataset	$\lvert\mathcal{R}\rvert$	Train Graph		Validation Graph			Test Graph		
		$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{F}\rvert$	**Test**	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{F}\rvert$	**Test**
FB v1	180	1594	4245	1594	4245	489	1093	1993	411
FB v2	200	2608	9739	2608	9739	1166	1660	4145	947
FB v3	215	3668	17986	3668	17986	2194	2501	7406	1731
FB v4	219	4707	27203	4707	27203	3352	3051	11714	2840
WN v1	9	2746	5410	2746	5410	630	922	1618	373
WN v2	10	6954	15262	6954	15262	1838	2757	4011	852
WN v3	11	12078	25901	12078	25901	3097	5084	6327	1143
WN v4	9	3861	7940	3861	7940	934	7084	12334	2823
NELL v1	14	3103	4687	3103	4687	414	225	833	201
NELL v2	88	2564	8219	2564	8219	922	2086	4586	935
NELL v3	142	4647	16393	4647	16393	1851	3566	8048	1620
NELL v4	76	2092	7546	2092	7546	876	2795	7073	1447
ILPC Small	48	10230	78616	6653	20960	2908	6653	20960	2902
ILPC Large	65	46626	202446	29246	77044	10179	29246	77044	10184
HM 1k	11	36237	93364	36311	93364	1771	9899	18638	476
HM 3k	11	32118	71097	32250	71097	1201	19218	38285	1349
HM 5k	11	28601	57601	28744	57601	900	23792	48425	2124
IndigoBM	229	12721	121601	12797	121601	14121	14775	250195	14904

Table 7: Dataset-wise statistics for the 18 datasets used under the inductive entity (e) generalization regime. $|\mathcal{R}|$ and $|\mathcal{E}|$ indicate vocabulary sizes of relations and entities present in each split, respectively. **Train**, **Validation**, and **Test Graphs** include the number of entities and triples present in each split. The **Test** columns denote the number of link prediction queries evaluated in each corresponding graph. The first part of the datasets is from Teru et al. (2020), the second from Galkin et al. (2022), the next from Hamaguchi et al. (2018), and IndigoBM from Liu et al. (2021).

size SEMMA is especially better for datasets where there is a lot of deviance between the structure and the textual semantics, or if the relation text is rich, or if the KG is sparse.

F Harder setting

We extend our evaluation to a more challenging setting, wherein the relation vocabulary of queries is entirely disjoint from the relation vocabulary present in the test graph. Formally, given $\mathcal{G}_{\text{TEST}}$ and queries $Q = \mathcal{F}_{\text{T}} \setminus \mathcal{F}_{\text{O}}$, we define a new evaluation setting:

$$\mathcal{R}_Q \cap \mathcal{R}_{\mathcal{G}_{\text{TEST}}} = \emptyset$$

where $\mathcal{R}_Q$ denotes the set of relations used in queries, and $\mathcal{R}_{\mathcal{G}_{\text{TEST}}}$ denotes the set of relations present in the test graph. This scenario is motivated by the practical case of temporally evolving knowledge graphs, wherein new relations frequently emerge over time and must be incorporated dynamically into existing inference frameworks (Cai et al., 2024). To construct datasets compliant with this harder setting, we perform the following procedure:

1. Start with the original dataset split, combining the test graph and test triples to create a unified set of facts.
2. Randomly select and mask out a subset of relations from the relation vocabulary of the test graph, based on a predefined split ratio.
3. Remove all triples involving these masked relations from the combined set, effectively filtering out these relations from the test graph.
4. From the filtered combined set, we define a subset to serve as our new test graph.
5. From the masked-out set of triples (associated with the masked relations), we select a representative subset as new test triples, ensuring the ratio of test graph to test triples matches the original dataset split and that entities in the test triples appear in the test graph.

In this more stringent setting, purely structural approaches such as ULTRA encounter significant difficulties. Specifically, when multiple distinct relations in the test triples share the same head or tail entity, ULTRA fails to differentiate among these relations. This limitation arises because ULTRA relies solely on structural identifiers (IDs), assigning identical temporary IDs to unseen relations at inference time due to their absence from the test graph.

In contrast, SEMMA effectively addresses this limitation. By incorporating semantic embeddings

Dataset	Train Graph			Validation Graph				Test Graph			
	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{F}\rvert$	**Valid**	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{F}\rvert$	**Test**
FB-25	5190	163	91571	4097	216	17147	5716	4097	216	17147	5716
FB-50	5190	153	85375	4445	205	11636	3879	4445	205	11636	3879
FB-75	4659	134	62809	2792	186	9316	3106	2792	186	9316	3106
FB-100	4659	134	62809	2624	77	6987	2329	2624	77	6987	2329
WK-25	12659	47	41873	3228	74	3391	1130	3228	74	3391	1131
WK-50	12022	72	82481	9328	93	9672	3224	9328	93	9672	3225
WK-75	6853	52	28741	2722	65	3430	1143	2722	65	3430	1144
WK-100	9784	67	49875	12136	37	13487	4496	12136	37	13487	4496
NL-0	1814	134	7796	2026	112	2287	763	2026	112	2287	763
NL-25	4396	106	17578	2146	120	2230	743	2146	120	2230	744
NL-50	4396	106	17578	2335	119	2576	859	2335	119	2576	859
NL-75	2607	96	11058	1578	116	1818	606	1578	116	1818	607
NL-100	1258	55	7832	1709	53	2378	793	1709	53	2378	793
Metafam	1316	28	13821	1316	28	13821	590	656	28	7257	184
FBNELL	4636	100	10275	4636	100	10275	1055	4752	183	10685	597
MT1 tax	10000	10	17178	10000	10	17178	1908	10000	9	16526	1834
MT1 health	10000	7	14371	10000	7	14371	1596	10000	7	14110	1566
MT2 org	10000	10	23233	10000	10	23233	2581	10000	11	21976	2441
MT2 sci	10000	16	16471	10000	16	16471	1830	10000	16	14852	1650
MT3 art	10000	45	27262	10000	45	27262	3026	10000	45	28023	3113
MT3 infra	10000	24	21990	10000	24	21990	2443	10000	27	21646	2405
MT4 sci	10000	42	12576	10000	42	12576	1397	10000	42	12516	1388
MT4 health	10000	21	15539	10000	21	15539	1725	10000	20	15337	1703

Table 8: Dataset-wise statistics for the 23 datasets used under the inductive entity and relation (e, r) generalization regime. $|\mathcal{R}|$ and $|\mathcal{E}|$ indicate vocabulary sizes of relations and entities present in each split, respectively. **Train**, **Validation**, and **Test Graphs** include the triples present in each split. The **Test** and **Valid** columns denote the number of link prediction queries evaluated in each corresponding graph. The first half of the datasets are from Lee et al. (2023) and the second half from Zhou et al. (2023).

derived from textual descriptions, SEMMA distinguishes between novel, previously unseen relations, leveraging the rich semantic signals inherent in relation texts. Consequently, SEMMA achieves significantly better prediction performance compared to purely structural methods as shown in Table 4.

Consider the example shown in Fig. 8, where ULTRA is unable to distinguish between `agentcollaborateswithagent` and `competeswith`, resulting in identical and incorrect top-10 predictions for both relations. In contrast, SEMMA successfully differentiates these two relations, as evident from the clearly ordered top-10 predictions that include the correct ground truth. This clearly illustrates SEMMA's ability to leverage textual understanding to overcome structural ambiguities.

G Full results

We provide the regime-wise averages of the dataset for our ablation studies in Table 10. The complete results per data set reporting MRR and Hits@10 of the zero-shot inference of the pre-trained ULTRA and SEMMA model are presented in Table 11 and Table 12.

H Discussion on Parallel works

REEF (Yu et al., 2025) leverages "relation tokens" from textual descriptions to adaptively generate parameters for GNN components via hypernetworks, aiming for effective pre-training and transfer, while SEMMA fuses distinct structural and textual relation graph representations for link prediction.

SCR (Wang et al., 2025) trains on KGs and reformulates general graph tasks into an inductive KG reasoning format to enable transfer, proposing Semantic Conditional Message Passing. SCR focuses on transferring KG reasoning to general graph tasks by reformatting them, whereas SEMMA focuses on inductive link prediction *within* KGs using relation semantics.

KumoRFM (Fey et al., 2025) is a pre-trained model for in-context learning on general relational databases, designed for zero-shot predictions across diverse enterprise tasks without specific training, using a Predictive Query Language. Insights like the helpfulness of textual semantics (Section 5.3) can potentially help improve such relational foundation models by leveraging the textual information in the database key and table names.

Component	Hyperparameter	Value
GNN_{STR}	# layers	6
	hidden dim	64
	message	DistMult
	aggeregation	sum
GNN_{TEXT}	# layers	6
	hidden dim	64
	message	DistMult
	aggeregation	sum
GNN_{ENT}	# layers	6
	hidden dim	64
	message	DistMult
	aggregation	sum
	$g(\cdot)$	2-layer MLP
Learning	optimizer	AdamW
	learning rate	0.0005
	training steps	200,000
	adv temperature	1
	# negatives	128
	batch size	64
	Training graph mixture	FB15k237, WN18RR, CoDExMedium

Table 9: **SEMMA hyperparameters for pre-training.** GNN_{STR} corresponds to the NBFNet that operates on $\mathcal{G}_R^{\text{STR}}$, GNN_{SEM} to the NBFNet that operates on $\mathcal{G}_R^{\text{TEXT}}$ and GNN_{ENT} to the NBFNet that operates on entity level.

		Inductive e, r (23 graphs)		**Inductive** e (18 graphs)		**Transductive** (13 graphs)		**Total Avg** (57 graphs)	
		MRR	**H@10**	**MRR**	**H@10**	**MRR**	**H@10**	**MRR**	**H@10**
LLMs	gpt-4o-2024-11-20	**0.355**	0.515	0.448	0.585	**0.322**	**0.473**	**0.377**	0.529
	deepseek-chat-v3-0324	**0.355**	**0.524**	**0.45**	**0.586**	0.307	0.461	0.375	**0.53**
	qwen3-32b	0.343	0.513	0.44	0.57	0.313	0.461	0.368	0.519
LM	jina-embeddings-v3	**0.355**	0.515	**0.448**	**0.585**	**0.322**	**0.473**	**0.377**	**0.529**
	Sentence-BERT	0.339	**0.521**	**0.448**	0.584	0.308	0.461	0.368	0.528
F	MLP	**0.355**	0.515	**0.448**	**0.585**	**0.322**	**0.473**	**0.377**	**0.529**
	Attention	0.343	**0.518**	0.441	0.577	0.314	0.467	0.369	0.525
Text	REL_NAME	0.338	0.517	**0.449**	**0.588**	0.314	0.465	0.369	0.528
	LLM_REL_NAME	**0.358**	**0.528**	0.447	0.585	0.308	0.456	0.375	**0.529**
	LLM_REL_DESC	**0.358**	0.525	0.436	0.566	0.312	0.4625	0.373	0.524
	COMBINED_SUM	0.355	0.515	0.448	0.585	**0.322**	**0.473**	**0.377**	**0.529**
	COMBINED_AVG	0.348	0.507	0.45	0.584	0.316	0.466	0.374	0.523
$\mathcal{G}_R^{\text{TEXT}}$	Threshold (0.8)	**0.355**	**0.515**	0.448	0.585	**0.322**	**0.473**	**0.377**	**0.529**
	Top-x% (20%)	0.341	0.509	**0.453**	**0.591**	0.31	0.462	0.371	0.525

Table 10: **Ablation Studies.** Evaluating the impact of different design choices in SEMMA. We report MRR and Hits@10 across the three evaluation regimes. The table compares (i) different large language models (LLMs), (ii) language encoders for deriving relation embeddings, (iii) fusion mechanisms (MLP vs. attention), (iv) variations in relation textual input, and (v) different strategies for constructing the textual relation graph ($\mathcal{G}_R^{\text{TEXT}}$).

LLM Prompt for Relation Text Enrichment

You will be provided with a list of relation names, each accompanied by exactly one example triple from a knowledge graph. Follow the instructions below carefully, strictly adhering to the output formats specified.

Step 1: *Convert Relation Names to Human-Readable Form.*
Clean each provided relation name, converting it into plaintext, human-readable form.

Output Format (JSON Dictionary): {{
"original_relation_name1": "Clean Human-Readable Form",
"original_relation_name2": "Clean Human-Readable Form",
...
}}

Step 2: *Generate Short Descriptions*
For each provided relation, generate a concise description (3-4 words) that clearly captures its semantic meaning based on the given example triple as context. Also, for each relation, generate a description of its supposed inverse relation. These descriptions will be converted into embeddings using jinaai/jina-embeddings-v3 to uniquely identify relations and to measure semantic similarities. So, avoid using common or generic words excessively, and do NOT reuse other relation names, to prevent false semantic similarities. Follow the rules below,

Be Concise and Precise: Use as few words as possible while clearly conveying the core meaning. Avoid filler words, unnecessary adjectives, and overly generic language.

Emphasize Key Semantics: Focus on the distinctive action or relationship the relation name implies. Ensure that the description highlights the unique aspects that differentiate it from similar relations.

Handle Negation Carefully: If the relation involves negation (e.g., “is not part of”), state the negation explicitly and unambiguously. Ensure that the description for a negated relation is clearly distinguishable from its affirmative counterpart.

Avoid Common Stopwords as Filler: Do not use common stopwords or phrases that add little semantic content. Every word should contribute meaning. Do not use repetitive words to avoid creating false semantic similarities.

Take care of symmetry: Ensure that for relations that are symmetric, the description does not change for its inverse relation.

Output Format (JSON Dictionary): {{
"original_relation_name1": ["concise description", "concise inverse relation description"],
"original_relation_name2": ["concise description", "concise inverse relation description"],
...
}}

List of Relations:
relation_name: “exist as” ; example: (“chloride”, “exist as”, “crystal”)
relation_name: “concept:statehascapital” ; example: (“concept:stateorprovince:mn”, “concept:statehascapital”, “concept:city:st_paul”)
...

Figure 5: LLM Prompt for relation text enrichment

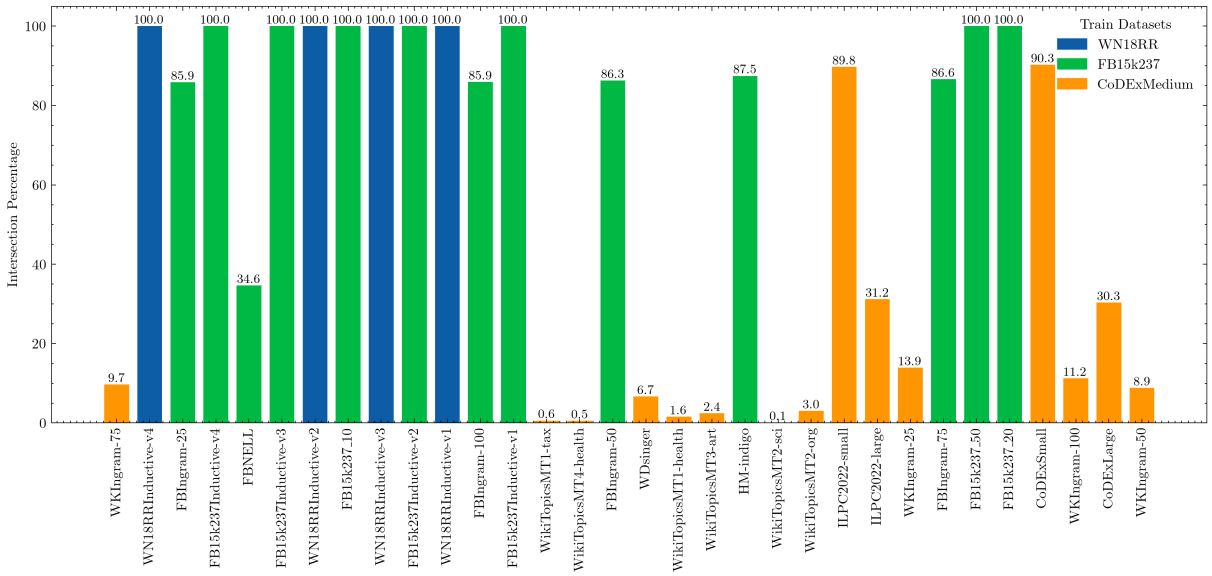

Figure 6: **Test graph Leak.** Percentage of test graph triples found in the pretraining corpus, indicating indirect leakage across datasets. Colors represent the corresponding training datasets in which leakage was found.

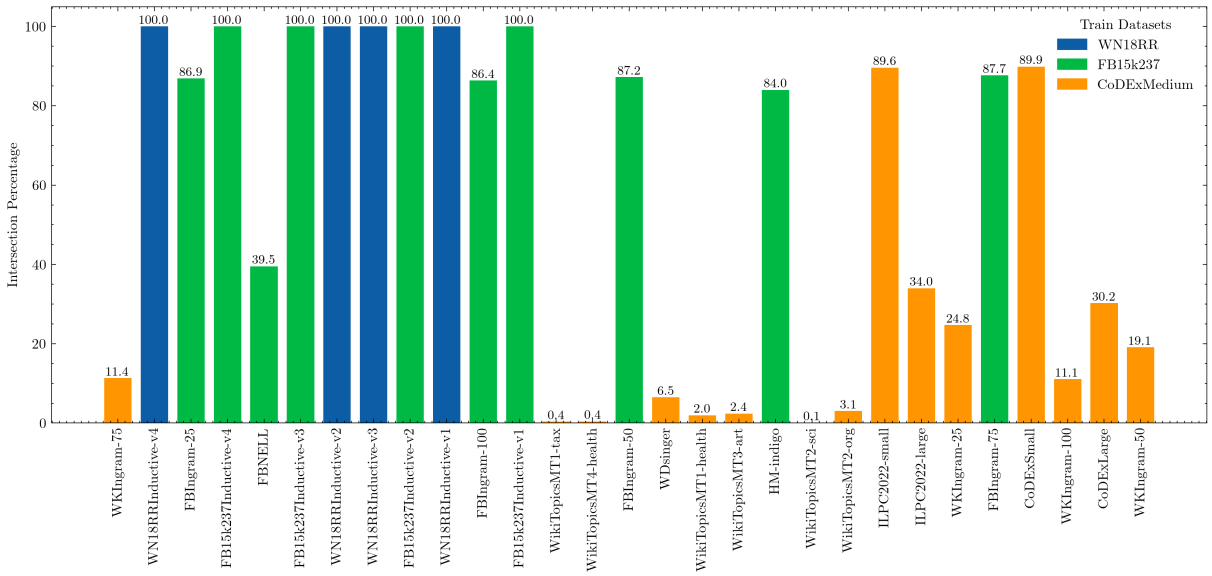

Figure 7: **Query triple Leak.** Percentage of query triples found in the pretraining corpus, indicating direct leakage across datasets. Colors represent the corresponding training datasets in which leakage was found.

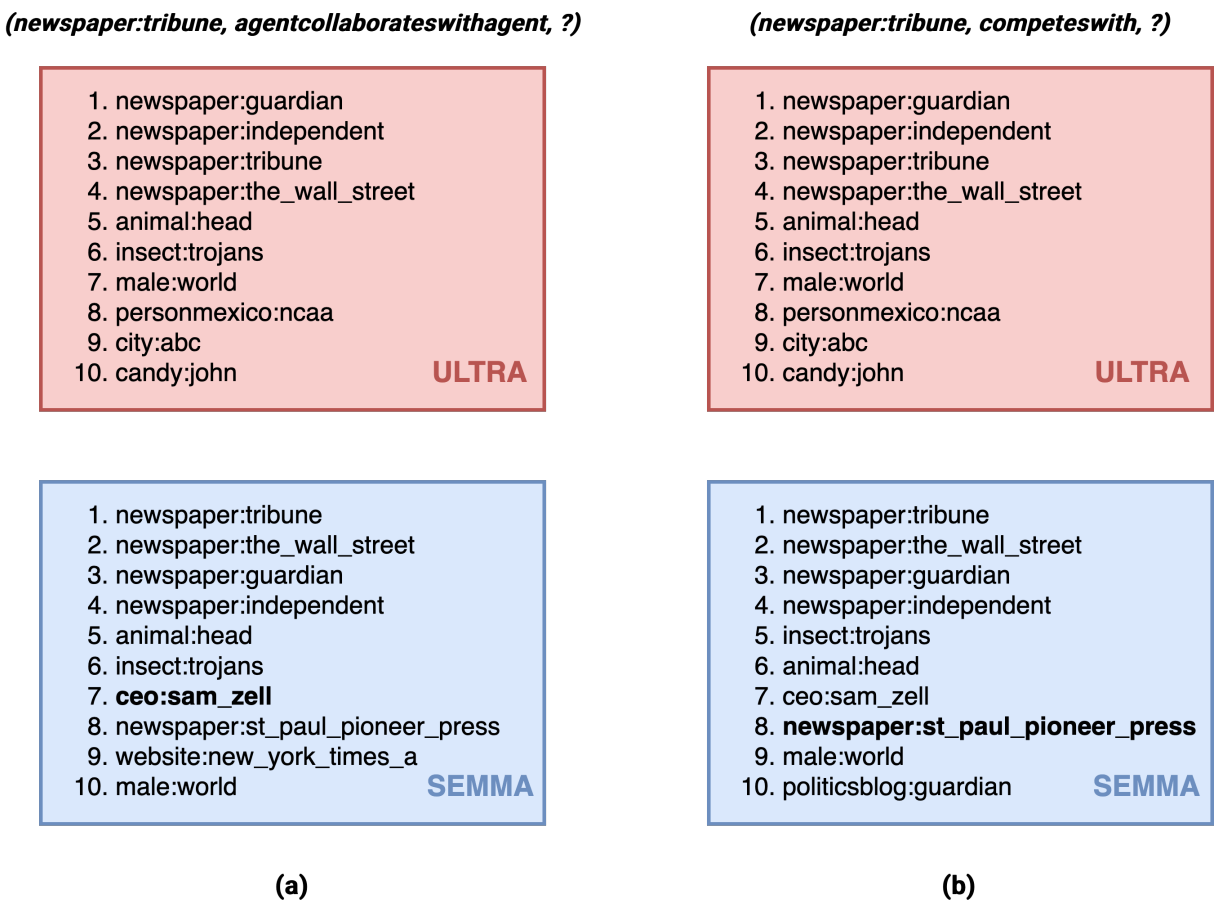

Figure 8: Comparison of ULTRA and SEMMA in the challenging setting where the query triples relation vocabulary is disjoint from the test graph relation vocabulary. ULTRA fails to differentiate between distinct relations (`agentcollaborateswithagent` vs. `competeswith`), producing identical incorrect predictions. In contrast, SEMMA distinguishes between the two relations and correctly predicts the ground truth within its top-10 predictions.

Dataset	ULTRA		SEMMA	
	MRR	**H@10**	**MRR**	**H@10**
Pre-training datasets				
FB15k237	0.372	0.569	**0.377**	**0.574**
WN18RR	0.499	0.622	**0.548**	**0.656**
CoDExMedium	**0.377**	**0.531**	0.376	0.529
Transductive datasets				
CoDExSmall	0.470	**0.674**	**0.479**	0.672
CoDExLarge	0.339	0.470	**0.349**	**0.478**
NELL995	0.402	0.534	**0.442**	**0.577**
DBpedia100k	0.387	0.561	**0.408**	**0.576**
ConceptNet100k	0.109	0.207	**0.162**	**0.311**
NELL23k	0.234	0.390	**0.242**	**0.413**
YAGO310	**0.467**	**0.634**	0.394	0.567
Hetionet	**0.289**	**0.412**	0.249	0.361
WDsinger	0.365	0.480	**0.386**	**0.496**
AristoV4	0.223	0.322	**0.232**	**0.346**
FB15k237_10	0.243	0.391	**0.244**	**0.393**
FB15k237_20	0.269	**0.434**	**0.269**	0.430
FB15k237_50	0.324	**0.525**	**0.324**	0.525
Inductive (e) datasets				
FB15k237Inductive:v1	0.481	0.647	**0.486**	**0.655**
FB15k237Inductive:v2	0.493	0.682	**0.503**	**0.691**
FB15k237Inductive:v3	0.480	0.641	**0.494**	**0.651**
FB15k237Inductive:v4	0.476	0.664	**0.492**	**0.677**
WN18RRInductive:v1	0.615	0.767	**0.724**	**0.816**
WN18RRInductive:v2	0.658	0.766	**0.705**	**0.803**
WN18RRInductive:v3	0.379	0.491	**0.442**	**0.577**
WN18RRInductive:v4	0.598	0.712	**0.664**	**0.741**
NELLInductive:v1	0.732	0.863	**0.798**	**0.935**
NELLInductive:v2	0.501	0.698	**0.543**	**0.730**
NELLInductive:v3	0.509	0.680	**0.530**	**0.720**
NELLInductive:v4	0.476	0.703	**0.496**	**0.729**
ILPC2022:small	**0.299**	0.447	0.298	**0.449**
ILPC2022:large	0.297	0.422	**0.307**	**0.429**
HM:1k	**0.063**	0.105	0.062	**0.109**
HM:3k	0.052	0.098	**0.056**	**0.102**
HM:5k	0.047	0.088	**0.055**	**0.102**
HM:indigo	**0.439**	**0.651**	0.435	0.645

Table 11: Comparison of ULTRA and SEMMA across transductive and partially inductive regime. The bold indicates the highest value of that metric for a specific dataset.

Dataset	ULTRA		SEMMA	
	MRR	**H@10**	**MRR**	**H@10**
Inductive (e, r) datasets				
FBIngram:25	0.383	0.636	**0.400**	**0.642**
FBIngram:50	0.332	0.536	**0.344**	**0.546**
FBIngram:75	0.395	0.598	**0.404**	**0.600**
FBIngram:100	0.436	0.634	**0.445**	**0.635**
WKIngram:25	0.288	0.481	**0.303**	**0.509**
WKIngram:50	0.152	0.304	**0.174**	**0.318**
WKIngram:75	0.372	**0.534**	**0.387**	0.525
WKIngram:100	0.178	0.295	**0.179**	**0.301**
NLIngram:0	0.327	0.491	**0.367**	**0.567**
NLIngram:25	0.381	0.534	**0.387**	**0.548**
NLIngram:50	0.365	0.531	**0.409**	**0.574**
NLIngram:75	0.333	0.494	**0.353**	**0.544**
NLIngram:100	0.444	0.631	**0.465**	**0.676**
WikiTopicsMT1:tax	**0.232**	**0.302**	0.229	0.302
WikiTopicsMT1:health	0.308	0.419	**0.336**	**0.435**
WikiTopicsMT2:org	0.086	0.146	**0.096**	**0.158**
WikiTopicsMT2:sci	**0.270**	**0.427**	0.258	0.388
WikiTopicsMT3:art	0.270	**0.418**	**0.278**	0.416
WikiTopicsMT3:infra	0.637	0.777	**0.650**	**0.784**
WikiTopicsMT4:sci	**0.288**	0.449	0.287	**0.454**
WikiTopicsMT4:health	0.580	0.735	**0.615**	**0.739**
Metafam	0.155	**0.565**	**0.258**	0.530
FBNELL	0.474	0.638	**0.482**	**0.655**

Table 12: Comparison of ULTRA and SEMMA across fully inductive regime. The bold indicates the highest value of that metric for a specific dataset.