Linguistic Initialization for Inductive Reasoning in Heterogeneous Knowledge Graphs

Daniele Pasquini, Danilo Croce and Roberto Basili
Department of Enterprise Engineering, University of Rome Tor Vergata
Via del Politecnico 1, 00133, Rome, Italy
daniele.pasquini@uniroma2.eu,
{croce, basili}@info.uniroma2.it

Abstract

Knowledge Graphs (KGs) provide explicit relational structure, while Large Language Models (LLMs) encode rich semantic knowledge. We propose a lightweight linguistic initialization strategy for heterogeneous link prediction that improves robustness under sparsity and imbalance. For each node, we construct a compact textual view combining intrinsic description and local neighborhood context, encode it with a pre-trained language model, and use the resulting embeddings to initialize a relation-aware GNN. This design preserves standard message passing while providing early semantically meaningful representations. Across multiple imbalance regimes and strict entity-to-entity cold-start settings, the proposed initialization consistently improves over random initialization and reduces degree-dependent degradation. Our results show that semantic grounding can be integrated into heterogeneous GNN pipelines with minimal architectural changes and strong empirical benefits.

Keywords: Knowledge Graph Completion, Graph Neural Networks, Heterogeneous Graph Attention

1. Introduction

Artificial Intelligence increasingly relies on structured representations. Many high-impact problems are naturally graph-shaped: entities interact, constraints are relational, and evidence is often distributed across topological neighborhoods rather than contained in a single feature vector. This is especially true in *Heterogeneous Information Networks* (HINs) (Sun et al., 2022), where different node and edge types carry rich complementary information.

Graph Neural Networks (GNNs) provide a principled mechanism to reason over such structures by iteratively aggregating neighborhood information. In Knowledge Graphs (KGs), topology-driven models learn from recurring connectivity patterns and generalize from local structures. When neighborhoods are informative and sufficiently dense, message passing can effectively propagate relational evidence across multiple hops.

However, this paradigm implicitly assumes that nodes are initialized with meaningful representations. In most existing approaches, semantics must be reconstructed from structural co-occurrence alone. In practice, most graph models rely on identifiers, short labels, or randomly initialized embeddings (Dai et al., 2022). These signals encode node identity but not their semantics. As a result, the first layers of message passing must simultaneously infer both node meaning and relational structure. In sparse and noisy graphs (Choi et al., 2025; He et al., 2024), this becomes problematic: nodes with little or no connectivity history (Cold Start) become statistically invisible (Qian et al., 2023), and topology-only models tend to fall back on global popularity priors or homophily bias (Ma et al., 2025; Patrini et al., 2017).

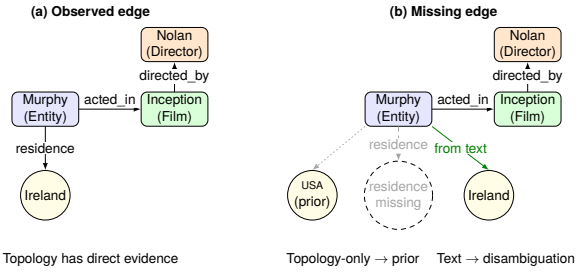

Figure 1: Structural symmetry under missing edges. Two query-centered graphs share identical topology. When `residence` is unobserved, topology-only models default to dominant priors. Text-grounded initialization injects semantic cues that enable correct disambiguation without explicit structural evidence.

This limitation becomes evident when structural evidence is incomplete. A topology-only model can generalize from repeated neighborhood patterns, but it cannot reliably recover attributes whose supporting edges are missing or under-sampled. In these cases, the graph induces plausible but non-factual defaults: the only available signal is global frequency and local structural similarity.

Figure 1 illustrates this issue. Two query-centered graphs share the same high-level structure (actor–film–director). In one case, the `residence` edge is observed; in the other, it is missing. From a purely topological perspective, the neighborhoods are indistinguishable. A structure-only

Proceedings of the Knowledge Graphs and Large Language Models Workshop 2026 (KG-LLM@LREC26), pages 1–10
May 16, 2026.

model therefore tends to prefer just the dominant location prior (e.g., USA). Yet the task would be significantly easier if the node representation already contained semantic cues such as "*Irish actor*". This information is not encoded in the adjacency structure but it can often be induced through natural language inferences. Language provides precisely this missing bridge. Crucially, we do not replace structural reasoning with text-based inference; rather, we reshape the initial state from which relational propagation begins. A textual description can encode intrinsic properties that are hard to infer from connectivity alone, disambiguating nodes even when structural context is partial or missing. Large Language Models (LLMs) offer a natural approach for this integration: the graph supplies relational evidence, while language supplies semantic grounding at the node and relation type level. More broadly, this setting can be interpreted as an alignment between explicit and implicit knowledge. Knowledge Graphs encode structured, symbolic relations with clear semantics and constraints, while LLMs store large amounts of implicit world knowledge acquired during pretraining. A principled integration should allow linguistic priors to inform graph reasoning, while ensuring that these priors are constrained and refined by the relational structure of the KG.

Motivated by this observation, we propose a heterogeneous graph representation framework coupling structure and language at the initialization stage. For each node, we construct a textual counterpart that combines (i) the intrinsic node description and (ii) a compact description of its local neighborhood. This rich textual representation is encoded through a pre-trained language model in order to provide initial embeddings for node embeddings before any message passing occurs. A relation-aware GNN then refines these semantically grounded representations through typed relational propagation.

This design yields a dual behavior. When structural context is sparse or missing, the model can rely on semantic signals inherited from language. When neighborhoods are informative, message passing refines these priors and prevents semantically distinct nodes from collapsing into structurally similar representations. In synthesis, initializing nodes with semantically meaningful representations simplifies learning under sparse supervision and reduces reliance on hub-based structural priors. Nodes no longer start from arbitrary identifiers and reconstruct meaning purely from co-occurrence. Instead, they begin from semantically grounded representations that are subsequently refined through relational propagation.

Empirically, we show that this initialization strategy consistently improves inductive link prediction, particularly under class imbalance and cold-start conditions. On a strict entity-to-entity cold-start subset, the proposed model substantially outperforms topology-only baselines, while preserving strong performance in dense regions.

Our contributions are threefold: *i*) we introduce a dual-view textual initialization strategy that verbalizes both intrinsic identity and local structural context before relational learning is triggered; *ii*) we integrate this initialization into a relation-aware heterogeneous GNN using bounded supervision mechanisms for scalability; *iii*) we provide a rigorous empirical evaluation under multiple imbalance regimes and strict entity-to-entity cold-start settings, demonstrating consistent robustness gains.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed model. Section 4 describes the experimental evaluation. We conclude with limitations and future directions.

2. Related works

Knowledge Graphs (KGs) and Large Language Models (LLMs) represent two complementary paradigms: KGs encode explicit symbolic relations and structural constraints, while neural models learn distributed representations capturing implicit semantics. A central challenge is integrating structured relational reasoning with these semantically grounded representations.

Early Knowledge Graph Completion (KGC) relied on geometric embedding models like TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), and RotatE (Sun et al., 2019). While effective, these transductive methods tie embeddings to node identifiers, failing to generalize to unseen entities. Graph Neural Networks (GNNs) (Gilmer et al., 2017) addressed this by introducing relational message passing. In the KG setting, R-GCN (Schlichtkrull et al., 2018) and HAN (Wang et al., 2019) extended this to multi-relational and heterogeneous data. Nevertheless, pure structural propagation remains vulnerable in sparse regimes: when local subgraphs carry weak signals, message passing alone often fails (Chamberlain et al., 2023).

To mitigate structure-only limitations, textual descriptions were incorporated via models like DKRL (Xie et al., 2016) and pre-trained language models like KG-BERT (Yao et al., 2019). While capturing semantic relatedness, text-only models often violate structural constraints. This motivated hybrid approaches: SimKGC (Wang et al., 2022) aligns language and graph representations via contrastive learning, while GraIL (Teru et al., 2020) learns subgraph patterns but degrades when neighborhoods are disconnected. Other works inject graph structure directly into LLM inputs (Fatemi et al., 2024),

highlighting that LLMs excel at semantics but struggle with precise topological navigation (Fatemi et al., 2024). Moreover, (Church and Bian, 2021) argues that Knowledge Graph Completion cannot be reduced to mere data filling: genuine coverage gaps require external semantic knowledge rather than inference over existing edges.

ecently, Text-Attributed Graphs (TAGs) have enriched node features with LLM-generated text. Methods such as TAPE and KEA (Chen et al., 2024) use LLMs as enhancers to augment node attributes before GNN training. However, they mainly target homogeneous graphs and still rely on standard GNNs, which can fail under strict cold-start settings.

Most existing hybrid models process text and graph structure via separate encoders, combining them only at the end of the pipeline. In contrast, we inject language at the beginning of the GNN training: nodes are initialized as text embeddings before any message passing occurs. Thanks to our residual fallback mechanism, isolated nodes safely retain their original text meaning, while well-connected nodes use the graph structure to refine it. This reframes KG–LLM integration from the *a posteriori* fusion to the dynamic shaping of the inductive bias within the relational reasoning. Recent retrieval-oriented graph-LLM frameworks combine graph-based retrieval with language generation (Peng et al., 2024), but they are mainly designed for query-time QA. In contrast, we address inductive missing-edge prediction in heterogeneous knowledge graphs, where the goal is to learn a reusable scoring function over node pairs and relations. Our contribution is therefore not an alternative to retrieval-based QA, but a lightweight, modular enhancement for predictive GNN pipelines.

3. Semantic Grounding for Heterogeneous Graph Learning

Knowledge Graph reasoning combines two distinct sources of information: (i) explicit relational structure and (ii) implicit semantic content. In most graph neural approaches, node representations are initialized randomly or from structural statistics, and semantic content must be reconstructed solely through message passing. We instead study a different regime: we inject linguistic semantics at initialization, and let relational propagation refine it.

Intuitively, consider the example in Figure 1. Two actors share identical structural neighborhoods (same film, same director). If the `residence` edge is missing, topology alone cannot distinguish them. However, a textual cue such as "*Irish actor*" already provides a semantic prior toward IRELAND. Our model formalizes this idea: language defines the starting state, and structure constrains its evolution.

3.1. Graph Formulation and Linguistic Encoding

We model data as a heterogeneous knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node-type mapping $\phi : \mathcal{V} \to \mathcal{A}$ and edge-type mapping $\psi : \mathcal{E} \to \mathcal{R}$ (Sun et al., 2011, 2022). Each node v may be associated with textual content $\mathcal{T}(v)$ (e.g., name, description, definition). We encode this content using a pre-trained language model f_θ:

$$\mathbf{x}_v = f_\theta(\mathcal{T}(v)), \qquad \mathbf{x}_v \in \mathbb{R}^{d_{\text{in}}}$$

These representations define the initial state of graph learning. Since heterogeneous node types (e.g., PERSON, LOCATION) may occupy different semantic regions, we apply a type-specific projection (DROP stands for Dropout):

$$\mathbf{h}_v^{(0)} = \text{Drop}\Big(\text{ReLU}\big(\mathbf{W}_{\phi(v)}\,\mathbf{x}_v + \mathbf{b}_{\phi(v)}\big)\Big)$$

with $\mathbf{W}_{\phi(v)} \in \mathbb{R}^{d \times d_{\text{in}}}$ and $\mathbf{b}_{\phi(v)} \in \mathbb{R}^d$. We use K attention heads (Velickovic et al., 2017) with per-head dimensionality d_k such that $d = K\,d_k$.

Graph learning then proceeds through L layers of relation-aware attention. For each relation r and attention head k, we first compute projected node representations

$$\tilde{\mathbf{h}}_i^{r,k} = \mathbf{W}_{r,k}\mathbf{h}_i^{(l)}, \qquad \tilde{\mathbf{h}}_j^{r,k} = \mathbf{W}_{r,k}\mathbf{h}_j^{(l)}$$

where $\mathbf{W}_{r,k} \in \mathbb{R}^{d_k \times d}$. The attention coefficient is then defined as

$$\alpha_{ij}^{r,k} = \text{softmax}_{j \in \mathcal{N}_i^r}\Big(\text{LeakyReLU}\big(\mathbf{a}_{r,k}^\top[\tilde{\mathbf{h}}_i^{r,k} \| \tilde{\mathbf{h}}_j^{r,k}]\big)\Big)$$

with $\mathbf{a}_{r,k} \in \mathbb{R}^{2d_k}$. Messages are aggregated per relation and summed across all the relation types $r \in \mathcal{R}_i$ incident to the i-th node:

$$\mathbf{m}_i^{\text{tot}} = \sum_r \text{Concat}_{k=1}^K \sum_{j \in \mathcal{N}_i^r} \alpha_{ij}^{r,k}\,\mathbf{W}_{r,k}\mathbf{h}_j^{(l)}$$

Node states are updated with a residual connection:

$$\mathbf{h}_i^{(l+1)} = \begin{cases} \text{Drop}\big(\text{ReLU}(\mathbf{m}_i^{\text{tot}})\big) + \mathbf{h}_i^{(l)} & \text{if } \mathcal{N}_i \neq \emptyset, \\ \mathbf{h}_i^{(l)}, & \text{otherwise} \end{cases}$$

This formulation yields two limiting behaviors. If a node is structurally isolated across all layers, then $\mathbf{h}_i^{(L)} = \mathbf{h}_i^{(0)}$: the model reduces to a pure semantic matcher. Conversely, when neighborhoods are informative, message passing refines and constrains the initial linguistic prior. Language therefore defines the inductive bias, while topology enforces relational consistency.

For a candidate triple (u, r, v), we compute:

$$s_{uv}^r = \mathbf{W}_{r,2}\ \text{ReLU}\Big(\mathbf{W}_{r,1}[\mathbf{h}_u^{(L)} \| \mathbf{h}_v^{(L)}]\Big)$$
$$\hat{y}_{uv}^r = \sigma(s_{uv}^r)$$

and optimize using binary cross-entropy with typed negative sampling.

From a Knowledge Graph perspective, training amounts to learning a scoring function $P(r \mid u, v)$ over typed triples (u, r, v), where relation-specific parameters ($\mathbf{W}_\phi$, $\mathbf{W}_{r,k}$, $\mathbf{W}_{r,1}$, $\mathbf{W}_{r,2}$) are optimized via supervised edge prediction. Crucially, the optimization dynamics depend on the initialization of node representations $\mathbf{h}_v^{(0)}$. With random initialization, the model must infer both semantic meaning and relational compatibility patterns solely from sparse structural co-occurrence, forcing early message-passing layers to reconstruct semantics from adjacency signals and often inducing slow convergence and hub-driven priors. In contrast, text-grounded initialization places $\mathbf{h}_v^{(0)}$ in a semantically meaningful region of the representation space, so training refines and calibrates an informed prior rather than creating semantics from scratch. Structure thus constrains existing representations instead of generating them, typically yielding more stable convergence and better generalization in long-tail and cold-start regimes.

3.2. Designing Node Linguistic Grounding

The previous section reframed initialization as a mechanism for shaping relational inductive bias. A practical question then arises: how can we construct semantically meaningful representations when explicit node descriptions are unavailable?

In many real-world KGs, nodes do not come with rich textual fields. We propose to derive textual information directly from the graph structure. For each node v, we construct: (i) an intrinsic textual view t_v^{node} for v (when available through v denotations), as well as (ii) the node v *contextual* view t_v^{ctx}. The intuition is that neighborhood structure explicitly encodes relational semantics. The proposed verbalization makes this information accessible to language models. In the example shown in Fig. 1, consider a node v representing an actor whose bounded 1-hop neighborhood, denoted by $\mathcal{N}(v)$, includes the relations `acted_in(Inception)` and `directed_by(Nolan)`. Verbalizing $\mathcal{N}(v)$ yields a compact description such as: *"Actor connected to the film Inception directed by Christopher Nolan"*, which is encoded as t_v^{ctx}. The combined textual representation is then encoded as: $\mathbf{x}_v = f_{\text{text}}\big(t_v^{\text{node}} || t_v^{\text{ctx}}\big)$ where $\|$ denotes the juxtaposition of the node text t_v^{node} and the contextual text t_v^{ctx} into a single input sequence, and f_{text} is any text encoder.

In this way, even structurally sparse nodes inherit semantic context from their local topology. Rather than injecting external knowledge, the graph itself becomes a source of linguistic grounding. The framework is schema-agnostic: it requires only typed nodes and labeled edges, without assumptions about a specific ontology, domain, or dataset. While graphs may differ in textual fields or type systems, the initialization–projection–propagation pipeline remains unchanged.

3.3. Complexity and Optimization

Real-world knowledge graphs are typically characterized by (i) highly skewed relation distributions and (ii) hub nodes with very large neighborhoods. In naive heterogeneous GNN implementations, both the number of relation-specific parameters and the supervision cost scale with $|\mathcal{R}|$ and the total number of edges M, which quickly becomes prohibitive (Song et al., 2024; Serafini and Guan, 2021; Liu et al., 2023).

Our approach keeps full-topology message passing, but strictly bounds the supervision phase.

Encoder cost. For a batch with N nodes and M edges, the heterogeneous encoder has per-layer complexity $\mathcal{O}_{\text{enc}} = \mathcal{O}\big(L\,(Nd^2 + Md)\big)$, which is standard for attention-based GNNs and necessary to preserve structural fidelity.

Bounded supervision. Let M_r denote the number of available supervised positive edges for relation r. We limit the number of positives used per relation as $\hat{M}_r = \min(M_r, K_{\text{cap}})$ where K_{cap} is the maximum number of supervised positive edges retained for any single relation, and activate only a random subset of relations $\mathcal{R}_{\text{active}}$ with $|\mathcal{R}_{\text{active}}| \le R_{\max}$ where $R_{\max}$ is the maximum number of distinct relations considered per step. For each positive edge, we further sample ρ negative edges. Additionally, we partition the relation space into a set of frequent head relations $\mathcal{R}_{\text{head}}$ and rare tail relations $\mathcal{R}_{\text{tail}}$, grouping the latter into semantic buckets $\mathcal{B}_{\text{tail}}$ (e.g. by their domain). This reduces parameter growth from $|\mathcal{R}|$ to $|\mathcal{R}_{\text{head}}| + |\mathcal{B}_{\text{tail}}| \ll |\mathcal{R}|$.

The resulting per-step complexity is

$$\mathcal{O}\big(L\,(Nd^2 + Md)\big) + \mathcal{O}\big(|\mathcal{R}_{\text{active}}| K_{\text{cap}}(1+\rho)d^2\big)$$

so that supervision cost is bounded by $R_{\max}$ and K_{cap}, independent of the global graph size.

Beyond efficiency, this design addresses three structural failure modes of topology-only GNNs. First, it mitigates the cold start by allowing isolated nodes retain semantic embeddings through textual initialization. Second, it resolves the structural symmetry by disambiguating nodes that are topologically similar but linguistically distinct. Third, it reduces long-tail sparsity, as parameter sharing across rare relations stabilizes learning. The model therefore exhibits hybrid behavior: structural reasoning in dense regions, semantic matching in sparse ones.

Metric	Train	Val	Test	Full
Counts				
# Graphs	350	75	75	500
# Nodes	479,126	119,374	105,422	703,922
# Edges	2,551,151	624,008	551,791	3,726,950
Averages				
Avg Nodes / Graph	1,368.93	1,591.65	1,405.63	1,407.84
Avg Entities / Graph	1,154.46	1,362.23	1,190.07	1,190.97
Avg Types / Graph	214.47	229.43	215.56	216.88
Avg Edges / Graph	7,289.00	8,320.11	7,357.21	7,453.90

Table 1: General Dataset Statistics across the training, validation and test splits.

Category	In	Out	Total	Conn.%
Global	5.29	5.29	10.59	**100**
Entity	4.10	5.26	9.36	**100**
Type	**11.85**	5.49	17.34	**100**

Table 2: Average degree and connectivity by node category.

4. Experimental Evaluation

We evaluate heterogeneous link prediction under structural sparsity and class imbalance, with a focus on inductive initialization. Using 500 query-centered subgraphs from the WebQSP-linked collection (He et al., 2021), we test (i) robustness to increasing imbalance, (ii) strict cold-start generalization, and (iii) degree-stratified recall. This setup shows whether text-grounded initialization solely improves aggregate performance, or fundamentally alters the inductive bias of heterogeneous graph learning in sparse regimes. To ensure rigorous reproducibility, the complete preprocessing workflow, dataset splits, evaluation protocols, and comprehensive implementation details are publicly accessible [1].

Experimental Setup. Rather than training on a single monolithic knowledge graph, we operate on a collection of bounded, query-centered subgraphs. This choice is both practical and methodological. Full-graph training is computationally expensive when running multiple controlled experiments, while evaluation over many heterogeneous subgraphs provides a more stable estimate of performance across diverse local topologies, avoiding domination by a few large hubs (Bajaj et al., 2024). This setting also mirrors realistic retrieval pipelines, where inference is performed over local neighborhoods rather than the entire graph. Each instance is constructed by expanding a bounded-hop neighborhood from one or more seed nodes. Node and relation identifiers are normalized into canonical strings and processed through three deterministic stages: (i) base initialization, (ii) structural resolution, and (iii) contextual optimization. The preprocessing workflow is publicly available[2].

[1] https://github.com/crux82/SemInit4GNN

[2] https://github.com/RichardHGL/WSDM20_NSM/tree/main/preprocessing/Freebase

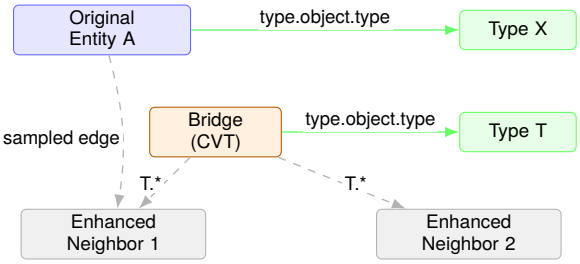

Figure 2: Pipeline: base entity-type construction, sampled enrichment, CVT resolution (sampled/CVT edges are in grey) and pruning.

During base construction, triples are ingested while literal nodes are discarded and stored as `name` attributes to avoid structural inflation. Targets of `type.object.type` edges are explicitly marked as TYPE nodes. Each non-type node is assigned a semantic category through a strict four-level priority scheme that favors informative shared types over generic meta-types.

To enrich structural context, we sample a bounded number of neighbors for original entities, filtering irrelevant predicates. A key step is the resolution of Compound Value Types (CVTs), i.e., unlabeled mediator nodes common in Freebase. We identify such bridges by detecting nodes without display labels whose incident predicates share a common prefix (e.g., `education.education.*`). As illustrated in Figure 2, bridges are resolved by explicitly linking them to their inferred type ID while preserving internal structure. If no display label is available, the canonical type ID is used as node name to ensure interpretability. After enrichment, auxiliary nodes are pruned except enhanced type nodes that remain structurally informative, yielding a compact, type-enriched graph. To provide semantic grounding, we generate concise natural-language descriptions for all nodes using `Gemma-7B-it` (Thomas Mesnard, 2024). Type nodes receive one-sentence conceptual definitions, as shown below

Type-node description example

Input node type: `film.film`
Prompt: "Below is the name of a conceptual type from a knowledge graph. Your task is to write a clear, one-sentence definition of this concept for a general audience. [...]"
Generated description: "A Film is a physical or digital medium used for recording, storing, and projecting moving images."

Entity and bridge nodes are described through their sampled local connectivity, as exemplified in Section 3.2. We adopt a binary node-type schema with two categories: ENTITY and TYPE. All non-type nodes are mapped to ENTITY, while category nodes are mapped to TYPE. Node descriptions are encoded using `ModernBERT-base` (Warner et al.,

2025), applying mean pooling over token representations to obtain a 768-dimensional feature vector. To isolate the contribution of semantic grounding, we also evaluate a random-initialization variant using Xavier uniform initialization. Data splitting follows a two-level protocol. At the graph level, subgraphs are partitioned into training (70%), validation (15%), and test (15%) sets, as summarized in Table 1, totaling over 703k nodes and 3.7M edges. On average, each graph contains approximately 1,400 nodes and 7,450 edges. Entities dominate the distribution (≈1,190 per graph) compared to types (≈217 per graph), reflecting the schema design where a limited set of ontological categories organizes many entities.

Topologically, all nodes have non-zero degree prior to training. As shown in Table 2, TYPE nodes act as structural hubs (average in-degree 11.85), while ENTITY nodes exhibit lower connectivity (average in-degree 4.10). This asymmetry induces a natural hub bias in message passing, making the setting suitable for analyzing degree-dependent effects. Within each graph, edges are split relation-wise: 10% are reserved for validation and 10% for testing. To prevent message-passing leakage, 20% of training edges are masked and used exclusively as supervision targets, while the remaining 80% remain in the adjacency matrix. Relation remapping is learned only on the training split, retaining frequent relations explicitly and grouping long-tail relations into semantic buckets (as defined in Section 3.3). The encoder is a 2-layer `HeteroGAT` (Velickovic et al., 2017; Schlichtkrull et al., 2018) with input dimension $d_{in} = 768$ and hidden dimension $d_{hid} = 256$. The decoder is a relation-specific MLP over concatenated endpoint embeddings. Models are trained for a maximum of 100 epochs using Adam ($\mathrm{lr} = 5 \times 10^{-4}$), gradient clipping (0.5), and Binary Cross-Entropy with Logits. To prevent overfitting, we evaluate the checkpoint that achieved the lowest validation loss. To bound supervision cost, we employ stochastic relation activation (maximum 8 active relations per step) and positive edge capping ($K_{\text{cap}} = 256$). Negatives are sampled from schema-consistent typed pairs. For relation r, negatives are drawn from $\Omega_r \setminus \mathcal{E}_r$, where $\Omega_r = \mathcal{V}_{\mathrm{src}(r)} \times \mathcal{V}_{\mathrm{dst}(r)}$. The number of negatives is $|\mathcal{N}_r| = \min(\rho|\hat{\mathcal{P}}_r|,\ |\Omega_r \setminus \mathcal{E}_r|)$ with $\rho = 19$, corresponding to approximately a 1:20 positive-to-negative ratio during training and validation. To disentangle structural and semantic contributions, we evaluate three topological configurations: (i) **Heterogeneous**, the full model with node types and relation-specific prediction (u, r, v); (ii) **Edge Existence**, where node types are preserved but relations are collapsed into a single connectivity label $(u, \exists, v)$; and (iii) **Homogeneous**, where both node and edge types are collapsed.

Baselines include Majority Class, Cosine Similarity over initial node embeddings, and a supervised Multi-Layer Perceptron (MLP) that predicts links using only concatenated node features, which is useful as a topology-agnostic baseline to evaluate performance against our architecture that incorporates the graph structure through the message passing mechanism. Robustness to imbalance is evaluated under three positive-to-negative ratios: **1:1** (Balanced), **1:9** (Weak), and **1:100** (Realistic). For GNN-based models, thresholds are tuned on validation data, and we report micro-averaged metrics aggregated across graphs. Across all configurations, architectural components remain fixed; only the initialization strategy varies, ensuring that observed differences can be attributed to semantic grounding rather than structural changes. Importantly, the generated textual descriptions are fixed prior to training and are not fine-tuned during graph optimization. Thus, improvements cannot be attributed to joint LLM–GNN training, but solely to semantically grounded initialization. Generated descriptions are conceptual and do not include instance-level relational facts from the target subgraph, preventing potential leakage of prediction targets.

Results and discussion. We evaluate the impact of linguistic grounding across three test settings: homogeneous link prediction, heterogeneous edge existence, and full heterogeneous relation prediction (Table 3). When heterogeneity is removed, the comparison isolates the effect of initialization. Under balanced supervision, both models perform well (Random F1=.84, Text F1=.98), indicating that dense positives allow topology to compensate for weak priors. As imbalance increases, the difference becomes structural. At 1:100, the Random model collapses (F1=.22), whereas the Text-grounded model remains strong (F1=.77). This supports our claim that linguistic features reshape the optimization regime: instead of learning semantics and relational compatibility simultaneously from sparse positives, the GNN refines pre-existing semantic structure. Text alone is insufficient. The MLP baseline reaches F1=.49 at 1:100, well below HomoGNN (.77), showing that message passing is required to transform semantic priors into relational decisions. Conversely, raw cosine similarity remains ineffective (F1=.03), confirming that link prediction requires structured learning beyond embedding proximity. Collapsing relation types tests whether typing and structure alone improve binary connectivity. With Random initialization, ExistGNN (F1=.30 at 1:100) outperforms the homogeneous random baseline (.22), likely due to explicit node typing that highlights hub patterns. With text grounding, ExistGNN (F1=.77) matches the homogeneous Text baseline, suggesting that textual de-

		Random init				Text-grounded init			
Ratio	**Method**	**Acc**	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**
a) Homogeneous setup (HomoGNN)									
Balanced	Majority (All 0)	.50	.00	.00	.00	.50	.00	.00	.00
	Cosine	.50	.50	1.00	.67	.58	.55	.89	.68
	MLP	.50	.50	1.00	.67	.89	.87	.93	.90
	HomoGNN	.83	.79	.90	.84	.98	.97	.98	**.98**
Weak	Majority (All 0)	.90	.00	.00	.00	.90	.00	.00	.00
	Cosine	.10	.10	1.00	.18	.68	.15	.45	.22
	MLP	.10	.10	1.00	.18	.94	.72	.71	.71
	HomoGNN	.91	.53	.59	.56	.98	.91	.93	**.92**
Realistic	Majority (All 0)	.99	.00	.00	.00	.99	.00	.00	.00
	Cosine	.01	.01	1.00	.02	.83	.02	.29	.03
	MLP	.01	.01	1.00	.02	.99	.54	.45	.49
	HomoGNN	.98	.17	.31	.22	.99	.76	.78	**.77**
b) Heterogeneous setup with collapsed relations (ExistGNN)									
Balanced	Majority (All 0)	.50	.00	.00	.00	.50	.00	.00	.00
	Cosine	.50	.50	1.00	.67	.58	.55	.89	.68
	MLP	.50	.50	1.00	.67	.90	.89	.92	.90
	ExistGNN	.85	.83	.88	.86	.97	.97	.98	**.97**
Weak	Majority (All 0)	.90	.00	.00	.00	.90	.00	.00	.00
	Cosine	.10	.10	1.00	.18	.80	.22	.37	.27
	MLP	.10	.10	1.00	.18	.95	.74	.73	.73
	ExistGNN	.92	.58	.65	.62	.98	.91	.93	**.92**
Realistic	Majority (All 0)	.99	.00	.00	.00	.99	.00	.00	.00
	Cosine	.01	.01	1.00	.02	.92	.03	.21	.05
	MLP	.01	.01	1.00	.02	.99	.52	.44	.48
	ExistGNN	.99	.28	.32	.30	.99	.80	.74	**.77**
c) Heterogeneous setup (HeteroGNN)									
Balanced	Majority (All 0)	.50	.00	.00	.00	.50	.00	.00	.00
	Cosine	.50	.50	1.00	.67	.58	.55	.90	.68
	MLP	.50	.50	1.00	.67	.85	.81	.91	.85
	HeteroGNN	.95	.94	.95	.95	.97	.97	.98	**.97**
Weak	Majority (All 0)	.90	.00	.00	.00	.90	.00	.00	.00
	Cosine	.10	.10	1.00	.18	.81	.22	.36	.27
	MLP	.10	.10	1.00	.18	.92	.57	.63	.60
	HeteroGNN	.97	.84	.85	.84	.98	.90	.92	**.91**
Realistic	Majority (All 0)	.99	.00	.00	.00	.99	.00	.00	.00
	Cosine	.03	.01	.98	.02	.93	.03	.20	.05
	MLP	.01	.01	1.00	.02	.98	.27	.31	.29
	HeteroGNN	.99	.55	.67	.60	.99	.70	.75	**.72**

Table 3: Comparison of random vs. text-grounded initialization across three setups (homogeneous, heterogeneous with collapsed relations, and heterogeneous).

scriptions already separate conceptual categories, making explicit type tags less critical for this simplified task. Again, structure matters: MLP remains substantially weaker (.48).

Predicting relation labels in the original schema is the most demanding setting. Under balanced supervision both models perform strongly (Random F1=.95, Text F1=.97), indicating that dense topology provides rich relational signal. Under realistic imbalance (1:100), Random drops to F1=.60, while Text-grounded maintains F1=.72. The advantage emerges particularly when relations share similar structural footprints (e.g., multiple predicates between the same entity types). Topology captures compatibility, but textual semantics resolves label-level ambiguity. The MLP baseline (F1=.29) confirms that text without relational propagation cannot recover fine-grained predicates. Overall, results consistently show hybrid behavior: in dense regimes, relational structure dominates; in sparse regimes, linguistic grounding provides a stable semantic fallback that prevents cold-start collapse.

Ratio	**Init**	$\lvert S_+ \rvert$	$\lvert S_- \rvert$	**Acc**	**P**	**R**	**F1**
1:1	**Random**	6,340	6,340	.81	.98	.64	.77
	Text-grounded	6,265	6,265	.92	.99	.85	.92
1:9	**Random**	6,340	57,060	.96	.88	.64	.74
	Text-grounded	6,265	56,385	.98	.92	.85	.89
1:100	**Random**	6,340	634,000	.99	.40	.64	.49
	Text-grounded	6,265	626,500	.99	.53	.85	.65

Table 4: Performance on the **Entity-to-Entity** subset.

Error Analysis. To move beyond aggregate metrics, we design a structured error analysis targeting regimes where graph topology is known to be fragile. We begin with a global hard subset that simulates cold start conditions by retaining only edges (u, r, v) where at least one endpoint has observed degree ≤ 1 in the training graph. This isolates the model's ability to generalize when structural history is minimal.

Cold-start evaluation in knowledge graphs can be artificially inflated by hub shortcuts: models often default to high-degree TYPE nodes (e.g., linking to popular entities such as "USA" for `nationality`). To remove this effect and probe genuine inductive behavior, we construct a stricter *entity-to-entity* subset where (i) the source has degree ≤ 1, (ii) the destination is an ENTITY (excluding TYPE hubs), and (iii) the relation is specific (excluding generic `Bucket` labels). Results are reported in Table 4. Under Random initialization, recall stabilizes around $.64$ across imbalance ratios, revealing a topology-driven blind spot: when the source has almost no neighborhood evidence, structural propagation alone cannot reliably recover the target. Text grounding raises recall to approximately $.85$, indicating that semantic initialization provides usable signal when structural context is absent. Under heavy imbalance $(1{:}100)$, both models lose precision, but degradation is sharper for Random ($P = .40$, $F1 = .49$) than for text-grounded ($P = .53$, $F1 = .65$). This suggests that linguistic grounding not only improves recall but also reduces spurious matches in sparse regimes. For KG practitioners, this result highlights that semantic priors become critical once hub shortcuts and type-based heuristics are removed. We further decompose the cold-start subset by relation category. We distinguish HEAD relations, frequent and semantically specific predicates requiring fine-grained discrimination among entities of similar types, from BUCKET relations, which approximate coarse domain membership. Table 5 shows that BUCKET relations are already near ceiling under Random initialization, confirming that structural compatibility and typing are sufficient for coarse associations. The bottle-

Init	Cat.	P	R	F1	N
Random	Head	0.99	0.68	0.81	3,412
	Bucket	0.99	0.97	0.98	251
Text	Head	0.99	0.88	0.93	3,278
	Bucket	0.99	0.92	0.96	244

Table 5: Analysis on the Hard Set (Deg≤ 1). Precision (P), Recall (R), and F1 for HEAD vs. BUCKET relations.

Init	Degree	Recall	N
Random	**Low**	0.67	3,853
	Med	0.72	17,293
	High	0.86	30,517
Text	**Low**	0.88	3,699
	Med	0.87	17,225
	High	0.90	30,739

Table 6: Topological bias analysis via Recall stratified by node degree.

neck emerges for HEAD relations: Random recall is $.68$ (F1=.81), reflecting ambiguity when multiple predicates share similar structural footprints. Text grounding resolves this last-mile ambiguity, increasing recall to $.88$ and F1 to $.93$. In other words, topology suggests *plausible* connectivity, but semantic grounding enables selection of the *correct* predicate under minimal structural evidence.

To quantify structural dependence more systematically, we stratify recall over the full test set by node degree into Low (≤ 2), Medium ($3-10$), and High (> 10) regimes (Table 6). With Random initialization, performance is strongly degree-dependent: recall rises from $.67$ (Low) to $.86$ (High), revealing reliance on dense neighborhoods. This pattern is characteristic of topology-driven inference, where reliable signals emerge primarily from hubs and well-connected regions. With Text-grounded initialization, the dependence on degree largely disappears: recall remains high across regimes (Low=.88, Medium=.87, High=.90), and the Low–High gap shrinks to roughly $.02$. On low-degree nodes alone, text grounding provides an absolute gain of about $+.20$ over Random, confirming that semantic priors compensate when structural evidence is insufficient.

Qualitative inspection of ranking behavior supports these quantitative patterns. For a cold-start node with only two training neighbors under the relation `medicine.disease.risk_factors`, the Text-grounded model ranks the correct target first with maximal confidence (P=1.00). In contrast, the Random model fails to place the true target in the top-10 and assigns similar low-margin scores (around $.24$) to unrelated candidates, reflecting uncertainty in the absence of structural cues. For a hub node with 28 neighbors under relations such as *"has topic primary type"* (`common.topic.notable_types`) and *"authored by"* (`book.written_work.author`), the Random model ranks correct targets at the top with probabilities above $.99$, showing that dense relational context alone can drive accurate predictions. Together, these diagnostics reveal a clear regime separation. In dense regions of the graph, relational message passing dominates, and even random initialization converges to strong structural predictors. In sparse or cold-start regions (particularly for specific predicates where structural footprints overlap) textual grounding acts as a stabilizing semantic prior that prevents collapse onto hub heuristics and improves predicate-level discrimination.

5. Conclusions

We investigated how semantically grounded initialization influences inductive reasoning in heterogeneous knowledge graphs. Rather than treating textual features as auxiliary inputs, we framed initialization itself as a mechanism for shaping the inductive bias of graph neural models. Our results show that text-grounded initialization consistently improves performance in entity-to-entity cold-start settings, particularly under strict splits that eliminate topological shortcuts and type leakage. Beyond aggregate gains, degree-stratified analyses reveal a structural effect: while random initialization induces popularity-driven bias toward high-degree hubs, semantically grounded initialization reduces this imbalance, improving recall for structurally sparse entities without sacrificing head performance. Predicate-level evaluation further indicates that structure captures coarse compatibility patterns, whereas language-derived priors enhance fine-grained relation discrimination.

Overall, these findings suggest that semantically grounded initialization provides a simple yet effective interface between implicit knowledge encoded in language models and explicit relational structure in knowledge graphs. Unlike recent Text-Attributed Graph paradigms that assume dense homogeneous neighborhoods, our approach achieves robustness through architectural design. By shaping the optimization landscape from the outset and employing a residual fallback, it offers a lightweight mechanism to mitigate structural bias and prevent representation collapse in strictly sparse regimes. Although demonstrated here with a specific heterogeneous GNN and fixed encoder, the approach is modular and readily transferable to alternative graph architectures or textual encoders, offering a reproducible framework for studying semantic–structural interaction in inductive link prediction and related knowledge graph reasoning tasks.

6. References

Saurabh Bajaj, Hojae Son, Juelin Liu, Hui Guan, and Marco Serafini. 2024. Graph neural network training systems: A performance comparison of full-graph and mini-batch. *Proc. VLDB Endow.*, 18(4):1196–1209.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Benjamin Paul Chamberlain, Sergey Shirobokov, Emanuele Rossi, Fabrizio Frasca, Thomas Markovich, Nils Hammerla, Michael M. Bronstein, and Max Hansmire. 2023. Graph neural networks for link prediction with subgraph sketching.

Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. 2024. Exploring the potential of large language models (llms)in learning on graphs. *SIGKDD Explor. Newsl.*, 25(2):42–61.

Yoonhyuk Choi, Jiho Choi, Taewook Ko, and Chong-Kwon Kim. 2025. Mitigating overfitting in graph neural networks via feature and hyperplane perturbation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, WSDM '25, page 50–59, New York, NY, USA. Association for Computing Machinery.

Kenneth Church and Yuchen Bian. 2021. Data collection vs. knowledge graph completion: What is needed to improve coverage? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6210–6215, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Enyan Dai, Wei Jin, Hui Liu, and Suhang Wang. 2022. Towards robust graph neural networks for noisy graphs with sparse labels. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 181–191, New York, NY, USA. Association for Computing Machinery.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. 2024. Talk like a graph: Encoding graphs for large language models. In *The Twelfth International Conference on Learning Representations*.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1263–1272. JMLR.org.

Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *WSDM*.

Tao He, Ming Liu, Yixin Cao, Zekun Wang, Zihao Zheng, and Bing Qin. 2024. Exploring & exploiting high-order graph structure for sparse knowledge graph completion. *Front. Comput. Sci.*, 19(2).

Zirui Liu, Chen Shengyuan, Kaixiong Zhou, Daochen Zha, Xiao Huang, and Xia Hu. 2023. RSC: Accelerate graph neural networks training via randomized sparse computations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 21951–21968. PMLR.

Yihong Ma, Yijun Tian, Nuno Moniz, and Nitesh V. Chawla. 2025. Class-imbalanced learning on graphs: A survey. *ACM Comput. Surv.*, 57(8).

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach . In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2233–2241, Los Alamitos, CA, USA. IEEE Computer Society.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey.

Tieyun Qian, Yile Liang, Qing Li, and Hui Xiong. 2023. Attribute Graph Neural Networks for Strict Cold Start Recommendation : Extended Abstract . In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 3783–3784, Los Alamitos, CA, USA. IEEE Computer Society.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Marco Serafini and Hui Guan. 2021. Scalable graph neural network training: The case for sampling. *SIGOPS Oper. Syst. Rev.*, 55(1):68–76.

Jaeyong Song, Hongsun Jang, Hunseong Lim, Jaewon Jung, Youngsok Kim, and Jinho Lee. 2024.

Granndis: Fast distributed graph neural network training framework for multi-server clusters. In *Proceedings of the 2024 International Conference on Parallel Architectures and Compilation Techniques*, PACT '24, page 91–107, New York, NY, USA. Association for Computing Machinery.

Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. Pathsim: Meta path-based top-K similarity search in heterogeneous information networks. *Proceedings of the VLDB Endowment*, 4(11):992–1003.

Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2022. Heterogeneous information networks: the past, the present, and the future. In *Proceedings of the 48th International Conference on Very Large Data Bases (VLDB)*, volume 15, pages 3807–3811.

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

Komal Teru, Etienne Denis, and Will Hamilton. 2020. Inductive relation prediction by subgraph reasoning. In *International conference on machine learning*, pages 9448–9457. PMLR.

Robert Dadashi et al. Thomas Mesnard, Cassidy Hardin. 2024. Gemma: Open models based on gemini research and technology.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat*, 1050(20):10–48550.

Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. 2022. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland. Association for Computational Linguistics.

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

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases.

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