EFP-KGRI: Expert-in-the-Loop Few-Shot LLM-Assisted Knowledge Graph Relationship Inference

Anonymous ACL submission

Abstract

Inferring relationships between entities in a knowledge graph (KG) is vital for numerous downstream applications such as semantic search, ontology construction, and personalized learning. However, many domains lack sufficient labeled data to train robust relationship inference models. In this paper, we present an Expert-in-the-Loop Few-Shot Prompting approach (EFP-KGRI) to perform LLM-Assisted relationship inference in KGs. Our framework leverages a large language model (LLM) to generate pseudo-labeled entity pairs, creating an initial set of positive (relationship present) and negative (no relationship) examples even in the absence of ground truth. We then fine-tune or calibrate these initial labels using embedding-based similarity scores and an active learning loop where expert feedback resolves uncertain cases. Experiments on both general-purpose encyclopedic KGs and specialized educational KGs demonstrate that EFP-KGRI significantly outperforms unsupervised baselines and naive LLM classification. By combining few-shot prompts, LLM self-consistency checks, and expert validation, we achieve more accurate and scalable relationship inference, effectively addressing the cold-start problem in knowledge graph completion.

1 Introduction

Knowledge Graphs (KGs) serve as critical infrastructures for structuring entities and their relationships across diverse domains, including search, recommendation, and education (Nickel et al., 2015; Hogan et al., 2021; Wang et al., 2024b). Despite the utility of KGs, relationship inference, identifying whether a specific relation (e.g., *precedes*, *subclass of*, *requires*) holds between two entities, often struggles when insufficient labeled data exists to train or calibrate models (Lin et al., 2015). Classic supervised methods and purely unsupervised embedding-based approaches both face limitations in cold-start scenarios: the former cannot learn without ground-truth labels, while the latter may over- or under-predict relationships without reliable guidance (Bordes et al., 2013; Nickel et al., 2015).

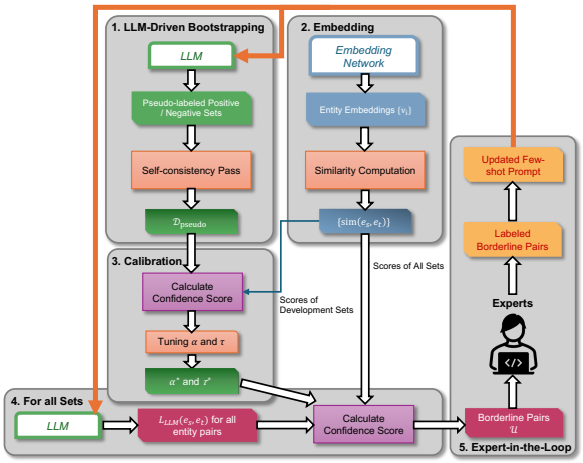

Figure 1: An overview of EFP-KGRI. (1) The LLM bootstraps an initial set of pseudo-labeled positives and negatives, then performs a self-consistency pass. (2) Entities are mapped to embeddings, and pairwise similarity scores are computed. (3) A calibration step fuses LLM labels with embedding scores to tune α and τ. (4) The resulting confidence function is applied to all entity pairs, and (5) borderline cases are submitted to experts for validation or correction, updating the few-shot prompt in an active learning loop.

Recent advances in Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2023), such as GPT-3.5 or GPT-4, offer new avenues for zero-shot and few-shot knowledge capabilities, using prompts that encode minimal examples for a target relation (Petroni et al., 2019; Schick and Schütze, 2020). However, LLMs alone can produce inconsistent or hallucinated outputs, particularly outside mainstream domains (Maynez et al., 2020; Ji et al., 2023). Merely querying an LLM for each entity pair also lacks a systematic check on correctness, potentially generating noisy or biased relationship labels.

To address these challenges, we propose an **E**xpert-in-the-Loop **F**ew-Shot **P**rompting pipeline for LLM-Assisted **K**nowledge **G**raph **R**elationship **I**nference (EFP-KGRI) as shown in Figure 1. First, we exploit the LLM's generative capacity to bootstrap initial sets of positive (relation-holding) and negative (no relation) pairs, thereby creating a pseudo-ground truth even in a cold-start scenario. We then integrate embedding-based similarity to calibrate thresholds, and employ an active learning loop (Settles, 2009) where domain experts or refined LLM queries resolve uncertain or conflicting cases, iteratively enhancing label quality.

In experiments on several KGs, one encyclopedic and multiple specialized graphs, our method consistently outperforms unsupervised baselines and naive LLM classification, demonstrating that few-shot prompts, LLM self-consistency checks, and expert validation together yield robust relationship inference. Crucially, the system minimizes expert workload by targeting only high-uncertainty or high-disagreement edges, addressing the cold-start problem with minimal reliance on existing labels. By bridging LLM reasoning, data-driven evidence, and focused human feedback, EFP-KGRI offers a practical and scalable blueprint for building accurate knowledge graphs even when labeled data are scarce. The contribution of this work can be summarized as:

1. We propose EFP-KGRI, a novel approach that integrates LLMs, embedding-based similarity metrics, and expert-in-the-loop feedback, to iteratively infer and validate relationships in knowledge graphs from various domains.

2. The proposed approach enlists domain experts only for uncertain or conflicting cases, optimizing annotation effort and steadily improving the accuracy of inferred relationships.

3. We present results on both encyclopedic and specialized KGs, demonstrating that our method outperforms unsupervised baselines and naive LLM classification in multiple settings.

2 Related Work

Knowledge Graph Relationship Inference. KGs have become prominent for organizing large-scale structured information (Nickel et al., 2015; Hogan et al., 2021). Classical methods for relationship inference of KGs range from embedding-based approaches, such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2014), or CompIEX (Trouillon et al., 2016), to more recent graph neural networks (GNNs) (Schlichtkrull et al., 2018) capturing rich structural patterns. However, these methods typically presume the existence of some labeled edges for training or tuning hyperparameters (Lin et al., 2015). In truly cold-start scenarios, where no ground-truth relationships are available, purely unsupervised strategies can produce noisy or biased predictions (Nickel et al., 2015). In the absence of labeled data, purely unsupervised strategies often produce suboptimal results.

Large Language Models for Knowledge Extraction. Transformer-based LLMs (Brown et al., 2020; Chowdhery et al., 2023) have demonstrated strong performance on zero-shot or few-shot tasks, including text classification and knowledge-based question answering (Radford, 2018; Schick and Schütze, 2020). GPT-style models can often be prompted to identify relationships between entities, leveraging their internal knowledge. Prior work has explored using LLMs for tasks like named entity recognition or relation extraction from text (Petroni et al., 2019). While LLMs can be surprisingly accurate, they can also hallucinate or supply spurious answers, particularly in domain-specific contexts (Maynez et al., 2020; Ji et al., 2023). Prior work has examined ways to reduce hallucinations via chain-of-thought reasoning (Wei et al., 2022) or calibration metrics (Kadavath et al., 2022). Our method extends these ideas by applying few-shot prompts to generate both positive and negative entity pairs for a target relationship. In contrast to prior work that typically focuses on extracting known facts from text (Levy et al., 2017; Bosselut et al., 2019), we use the LLM to bootstrap entirely new labeled pairs in the absence of any initial ground truth, then apply embedding-based checks to mitigate errors.

Active Learning and Expert-in-the-Loop Systems. Active learning (Settles, 2009; Zhang et al., 2022) aims to reduce labeling costs by selectively querying a human annotator (or an oracle) for the most informative samples. In knowledge graph contexts, this can mean focusing on edges near the decision boundary or areas where different models disagree. Expert-in-the-loop pipelines (Ratner et al., 2019) have proven especially valuable in specialized domains (e.g., medical, legal, and educational KGs) where domain expertise is essential. Our framework seamlessly integrates an expert step to validate or correct uncertain cases suggested by either the LLM or our similarity-based model.

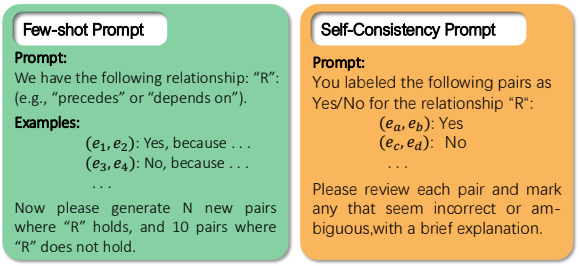

Figure 2: Illustration of the few-shot prompt and the self-consistency prompt used in our pipeline.

3 Method

In this section, we elaborate on our EFP-KGRI (Expert-in-the-Loop Few-Shot pipeline for LLM-Assisted Knowledge Graph Relationship Inference). The overall framework aims to address cold-start labeling by harnessing an LLM to generate pseudo-labeled pairs (both positive and negative), integrate embedding-based similarity for threshold calibration, and iteratively refine uncertain cases with expert or additional LLM feedback. Figure 1 illustrates the pipeline of EFP-KGRI.

3.1 LLM-Driven Bootstrapping

Our system begins with an unlabeled set of entities $\{e_1, e_2, \ldots, e_n\}$. We aim to generate an initial labeled dataset $\mathcal{D}_{\text{pseudo}}$ indicating whether each entity pair (e_s, e_t) holds a target relationship r. This step is critical for cold-start scenarios, where no ground truth is available.

Few-Shot Prompt Construction We first handcraft a small few-shot prompt that demonstrates how to label relationships. This prompt includes a handful of positive examples (where r indeed holds) and negative examples (where r definitely does not). Each example includes: (1) A short description of the entities involved. (2) A label: "Yes" (relationship holds) or "No" (relationship does not hold). (3) A brief rationale.

After listing these seed examples, we instruct the LLM to generate new pairs of entities, labeling each as "Yes" or "No" and optionally providing brief reasoning. We use DeepSeek-R1 (DeepSeek-AI et al., 2025). A simplified example is shown in the left of Fig. 2. This approach bootstraps a labeled set of both positive and negative examples, using the LLM's internal world knowledge and language understanding.

Self-Consistency Verification We collect the newly generated pairs $\{(e_s, e_t)\}$ and their labels$\{y\}$, and present them back to the LLM with a verification prompt shown in the right of Fig. 2. The LLM may identify contradictory or dubious relationships (especially in domain-specific contexts) and mark them for exclusion or reevaluation. Then, we remove pairs flagged by the LLM as incorrect or highly uncertain. The remaining examples constitute our pseudo-labeled dataset $\mathcal{D}_{\text{pseudo}}$. Each pair (e_s, e_t) has a label $y \in \{\text{Yes}, \text{No}\}$, reflecting the LLM's best guess.

Through this procedure, we obtain an initial set of positive and negative pairs—despite having no original ground truth. In subsequent stages, these pseudo-labels are refined and validated using embedding-based thresholds and expert-in-the-loop checks, as described in Sections 3.2 and 3.3.

3.2 Embedding-Based Calibration

While the LLM-generated pseudo-labels offer a valuable initial signal for relationship inference, relying solely on LLM outputs can lead to hallucinations or overgeneralizations—especially in domain-specific contexts. To address this, we integrate embedding-based entity representations and threshold calibration to (1) align the LLM's textual reasoning with data-driven evidence, and (2) filter out implausible or contradictory cases.

Entity Embeddings. We begin by converting each entity e_i into a vector representation $\mathbf{v}_i \in \mathbb{R}^d$. The choice of embedding model depends on the available data: If each entity e_i is accompanied by a textual descriptor (e.g., a short Wikipedia paragraph, an abstract, or a definition), we can use a pretrained transformer (e.g., BERT, RoBERTa, SciBERT if domain-specific) or embedding models (e.g., text-embedding-3-small or voyage-3) to generate a sentence-level or document-level embedding. We use the embedding of DeepSeek-V3 (DeepSeek-AI et al., 2024). Formally,

$$\mathbf{v}_i = \text{Encoder}(\text{Text}(e_i)), \tag{1}$$

where Encoder could be a transformer with a [CLS] token output or mean pooling of contextual embeddings. This approach leverages distributional semantics, capturing relevant domain knowledge from large-scale pretraining corpora.

If partial KG or adjacency information (excluding the target relationship) is available, we can learn graph embeddings (e.g., node2vec, DeepWalk, or a GNN-based approach). Each node e_i is represented by a vector $\mathbf{v}_i$ that encodes structural proximity to other nodes already in the graph. Such methods are

beneficial when textual descriptions are sparse or inconsistent. Depending on data availability, one can also combine text-based and graph-based features (e.g., by concatenating the two embedding vectors). In all cases, the result is a consistent representation $\mathbf{v}_i$ for each entity e_i.

Similarity Computation. Once we have embeddings, we define a similarity function $\mathrm{sim}(\cdot, \cdot)$ for each pair (e_s, e_t). As there are many choices such as Cosine Similarity, Dot Product and Distance-Based Measures, for simplicity, we use Cosine Similarity in our experiments (Section 4), as it handles varying embedding magnitudes and is widely employed in semantic tasks:

$$\mathrm{sim}(e_s, e_t) = \frac{\mathbf{v}_s \cdot \mathbf{v}_t}{\|\mathbf{v}_s\|\|\mathbf{v}_t\|}. \tag{2}$$

Fusing LLM Outputs with Embedding Scores. The target is to combine LLM-generated pseudo-labels, particularly the "Yes" or "No" assignments from Section 3.1, with embedding-based similarity to produce a confidence value for each pair. This helps reduce false positives from LLM hallucinations and false negatives from purely unsupervised embeddings.

Given a pair (e_s, e_t) labeled Yes by the LLM, define the binary indicator:

$$L_{\mathrm{LLM}}(e_s, e_t) = \begin{cases} 1, & \text{if LLM says Yes,} \\ 0, & \text{if LLM says No.} \end{cases} \tag{3}$$

Then, we get the confidence score by blending the LLM label with the embedding similarity via a weighted sum:

$$\mathrm{Conf}(e_s, e_t) = \alpha \cdot L_{\mathrm{LLM}}(e_s, e_t) + (1 - \alpha)\,\mathrm{sim}(e_s, e_t), \tag{4}$$

where $\alpha \in [0, 1]$ is a hyperparameter controlling how much trust we place in the LLM's classification. Intuitively, if the LLM says "Yes" ($= 1$) and the similarity is also high, Conf becomes large. If the LLM says "No," Conf is primarily determined by $\mathrm{sim}(e_s, e_t)$. There may also be alternative approaches. In specialized domains, one may replace L_{LLM} with a confidence score from the LLM or use a logistic/softmax transform on the similarity. The key is ensuring alignment between textual reasoning and the numeric evidence from embeddings, rather than relying on a single source.

Threshold Tuning. To binarize relationships deciding whether a pair (e_s, e_t) truly holds the target relation r, we define a threshold τ on the confidence score:

$$\widehat{y}(e_s, e_t) = \begin{cases} \text{Yes}, & \text{if } \mathrm{Conf}(e_s, e_t) \geq \tau, \\ \text{No}, & \text{otherwise.} \end{cases} \tag{5}$$

Recall from Section 3.1 that we have a pseudo-labeled dataset $\mathcal{D}_{\text{pseudo}}$. Each pair has an LLM-assigned label $\{\text{Yes, No}\}$. We treat $\mathcal{D}_{\text{pseudo}}$ as a development set for tuning α and τ. Specifically, we iterate over candidate values (e.g., a grid search) and pick the pair (α, τ) that optimizes a chosen metric, commonly AUROC or balanced accuracy. The procedure is the following:

1. For each candidate α, compute $\mathrm{Conf}(e_s, e_t)$ for all pairs in $\mathcal{D}_{\text{pseudo}}$.
2. Sweep τ from 0 to 1 (or from min to max of the confidence distribution). At each τ, calculate AUROC.
3. Choose τ^* that yields the best AUROC for α.
4. Finally, select the α^* that leads to the highest performance.

We then apply the chosen α and τ to all entity pairs (not just the ones in $\mathcal{D}_{\text{pseudo}}$), producing a preliminary classification "Yes" or "No" for each pair. This calibration step is crucial: it grounds the LLM's guesses in a numeric measure of semantic similarity and systematically filters improbable relationships.

To conclude, the embedding-based calibration step serves two main functions: mitigating LLM hallucinations and handling conflicting signals. If the LLM labels an entity pair as "Yes" but their embeddings are extremely dissimilar, the fused confidence score may remain below the threshold τ, thus overriding or downgrading a likely incorrect assignment. In cases where the LLM says "No" but the similarity is very high, there may be a mismatch between the LLM's textual reasoning and domain-specific knowledge. Such pairs could be flagged as potentially uncertain for expert review (see Section 3.3).

3.3 Expert-in-the-Loop Active Learning

While LLM bootstrapping (Section 3.1) and embedding-based calibration (Section 3.2) provide a solid starting point, some entity pairs remain uncertain or in conflict with domain knowledge. This is especially true in specialized areas where the LLM's training data may be sparse or where embedding models yield ambiguous signals. To address these gaps, we adopt an active learning framework

in which expert feedback (or further refined LLM queries) is solicited for pairs most likely to improve our overall system performance.

Identifying Uncertain or Disputed Pairs. After computing a confidence score $\mathrm{Conf}(e_s, e_t)$ and deciding a threshold τ, we measure each pair's distance from τ. Pairs with $\mathrm{Conf}(e_s, e_t)$ close to τ are deemed "borderline" and potentially high impact for labeling. If the LLM confidently says "Yes" but the embedding similarity is extremely low (or vice versa), we flag this discrepancy. Such mismatches often indicate incoherent or hallucinated relationships, or overlooked domain nuances. We collect a batch of these uncertain or disputed pairs $\mathcal{U}$. Rather than verifying all pairs (which may be in the tens or hundreds of thousands), we focus only on the small subset where additional labeling is most valuable.

Expert Validation. In many domains, domain experts can give reliable judgments on whether the target relation r holds for a pair (e_s, e_t). We present these pairs in a concise interface, including minimal context (e.g., textual descriptors, any partial KG links, or the LLM's rationale), so the expert can quickly determine if the pair is "Yes" or "No." After done by an expert or a refined LLM query, each pair in $\mathcal{U}$ receives a final label ("Yes" or "No") and, ideally, a brief explanation. These validated pairs then form a new increment of ground truth.

Updating the Model and Few-Shot Prompt. Newly validated pairs $\mathcal{U}$ are added to our labeled set $\mathcal{D}_{\text{pseudo}}$, effectively correcting or reinforcing earlier LLM outputs. If certain pairs contradict the LLM's original label, we override them with the new expert label, thus cleaning up false positives/negatives. Moreover, if the newly confirmed pairs differ significantly from the pseudo-labeled distribution, we may re-optimize the threshold τ and weight α. In practice, re-tuning may be performed only after a substantial batch of updated labels to avoid overfitting. Particularly interesting or representative corner cases (e.g., relationships that consistently confuse the LLM) can be integrated into the few-shot prompt. By showing the LLM exactly how certain tricky pairs were resolved, we guide it toward improved consistency in subsequent iterations. To summarize, we iterate the process of:

1. Classifying all entity pairs using the current best threshold τ.
2. Selecting the most uncertain or contradictory pairs for review.
3. Validating them via expert.
4. Updating the labeled set and optionally re-tuning τ and α.

We repeat this cycle until diminishing returns make further active learning unnecessary. This approach maximizes the utility of limited human intervention, where experts only evaluate pairs that are truly unclear, rather than exhaustively labeling the entire graph. This synergy between human validation and machine learning ensures a more robust, iteratively improving approach to knowledge graph relationship inference.

3.4 Summary of the EFP-KGRI Pipeline

We have described the EFP-KGRI pipeline in three key stages: **1. LLM-Driven Bootstrapping (Section 3.1)** We use a carefully designed few-shot prompt to have the LLM generate pseudo-labeled pairs of entities, both positive (relationship present) and negative (no relationship). A self-consistency pass filters contradictory or low-confidence pairs, resulting in an initial labeled set $\mathcal{D}_{\text{pseudo}}$.
2. Embedding-Based Calibration (Section 3.2) We convert entities to embeddings (using text or structural features) and define a similarity measure. We then fuse the LLM labels with similarity scores into a confidence value, and calibrate a threshold τ to separate "Yes" from "No." This helps mitigate LLM hallucinations and aligns textual reasoning with data-driven signals.
3. Expert-in-the-Loop Active Learning (Section 3.3) We identify uncertain or disputed pairs, typically near the threshold or where LLM and embeddings disagree. Experts or refined LLM prompts validate these pairs, updating the labeled set and (optionally) re-tuning thresholds. This iterative feedback loop maximizes the impact of limited human annotations, steadily improving precision and recall.

Having thus outlined our method, we now move on to Section 4, where we detail our experimental setup, baseline comparisons, and empirical results on two real-world knowledge graphs.

4 Experimental Results

We evaluate EFP-KGRI on six datasets to demonstrate its ability to infer relationships in KGs. We begin by describing the datasets and the baselines (Section 4.1). We then present quantitative and qualitative results (Section 4.2) and ablation studies (Sections 4.3 - 4.4 and Appendix F).

4.1 General Settings

Datasets. We evaluate EFP-KGRI on six datasets, namely IntelliGraphs-syn-types (Thanapalasingam et al., 2023), CoDEx-S (Safavi and Koutra, 2020), DBE-KT22 (Wang et al., 2024a), Junyi (Wang et al., 2024a), WDKG-Course (Wang et al., 2024a), and WDKG-KnowledgePoints (Wang et al., 2024a), covering both synthetic/general and educational domains. IntelliGraphs-syn-types provides a controlled synthetic environment with typed entities, while CoDEx-S is a curated subset of Wikidata reflecting diverse real-world concepts. DBE-KT22, Junyi, WDKG-Course, and WDKG-KnowledgePoints all center on educational knowledge, emphasizing prerequisite or dependency relations at varying granularity from broad course dependencies to fine-grained knowledge. For each dataset, we identify a target relationship r (e.g., "precedes," "depends on"). During experiments, we sampled subgraphs of 50 nodes with connected structure, then repeated this for ten rounds, and reported the average results to cut cost. Negative examples are generated via the LLM-based approach (Section 3.1) or standard random sampling, ensuring balanced sets of positive and negative pairs. These datasets allow us to gauge the effectiveness of our pipeline across synthetic, general-purpose, and education-focused KGs.

Baselines. We compare EFP-KGRI against a set of both classic and recent KG inference methods, ensuring coverage of purely embedding-based approaches and more LLM-centric solutions. Classic embedding-based methods:

- TransE (Bordes et al., 2013): A translational embedding approach, interpreting each relation as a vector $\mathbf{r}$ such that $\mathbf{v}_s + \mathbf{r} \approx \mathbf{v}_t$.
- DistMult (Yang et al., 2014): A bilinear method representing a relation with a diagonal matrix, balancing simplicity and effectiveness.
- ComplEx (Trouillon et al., 2016): Extends DistMult by using complex-valued embeddings, better modeling asymmetric relations.

LLM-based methods:

- BLP (Daza et al., 2021): Adapts BERT to predict missing edges by encoding entity descriptions and training a binary classifier on embeddings.
- KGT5 (Saxena et al., 2022): A text-to-text transformer approach that frames link prediction as a sequence-to-sequence task.
- SimKGC (Wang et al., 2022): Employs contrastive learning over text-enhanced entity representations, using a PLM to encode entity descriptions and a similarity-based objective to rank candidate entities for a given (subject, relation) query.

We also include a direct prompting baseline, without iterative thresholding or expert feedback, to highlight the difference between naïvely using an LLM vs. our integrated pipeline. For each entity pair (e_s, e_t), we prompt an LLM (GPT-4o, o1, DeepSeek-R1) with: "Does relation r hold between e_s and e_t? Yes or No?" No additional calibration or active learning is applied, and any rationale from the LLM is ignored beyond the final yes/no label.

More details about datasets, models, and hyperparameter settings are left in Appendix A, Appendix B, and Appendix C, respectively.

4.2 Overall Performance

Across all six datasets, IntelliGraphs, CoDEx-S, DBE-KT22, Junyi, WDKG-Course, and WDKG-KnowledgePoints, our EFP-KGRI attains the highest AUROC values and outperforms both classic and modern baselines, as shown in Fig. 3. Traditional embedding-based techniques (e.g., TransE, DistMult, and ComplEx) capture broad structural patterns but frequently struggle with specialized domain nuances, particularly in the educational KGs like DBE-KT22, Junyi, and WDKG series. By comparison, PLM- or LLM-driven models such as BLP, KGT5, SimKGC, GPT-4o, o1, and DeepSeek-R1 benefit from large-scale textual pretraining, yet typically lack a mechanism to resolve ambiguous or borderline cases via targeted feedback. In contrast, our method combines LLM-based pseudo-label generation with embedding calibration and actively solicits expert insight for uncertain edges, which yields consistently higher AUROC scores.

A closer inspection of the domain-specific datasets reveals how essential human review can be for clarifying subtle prerequisite relationships, such as those in WDKG-Course and WDKG-KnowledgePoints, where slight conceptual differences may drive significant variations in performance. In these contexts, the Expert-in-the-Loop step systematically corrects errors that either purely data-driven or purely LLM-based approaches fail to catch. Even in the synthetic IntelliGraphs dataset, which presents a more controlled scenario, our pipeline's strategy of fusing LLM reasoning with numerical checks continues to produce a competitive edge, illustrating that iterative threshold tuning and refinement offer advantages regardless of whether the domain is synthetic or real-world.

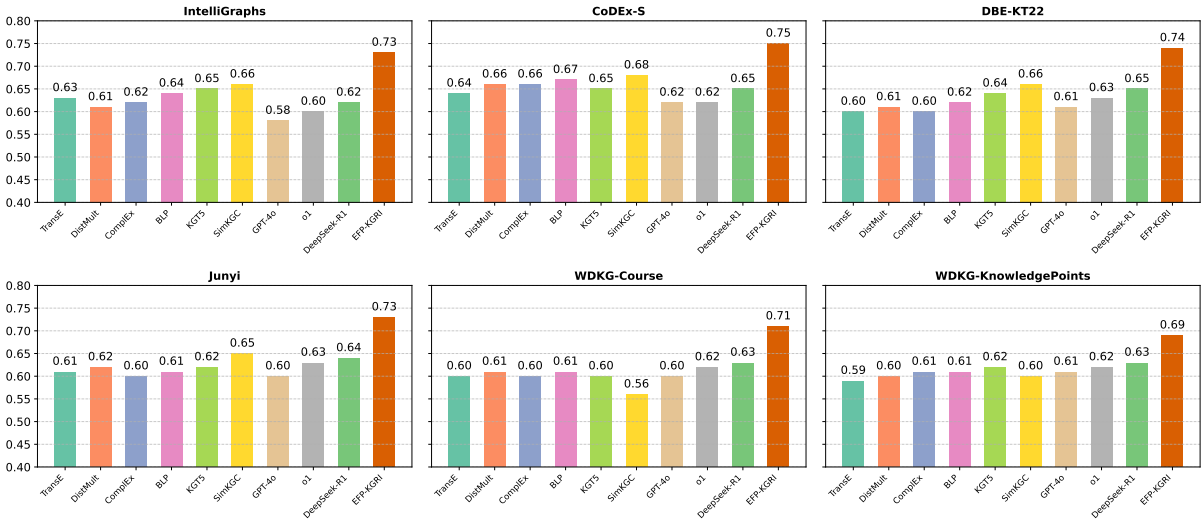

Figure 3: AUROC values on six datasets for baselines, and EFP-KGRI. Higher bars indicate better performance.

Overall, these AUROC results underscore the importance of uniting textual reasoning and numerical validation. The synergy between LLM-generated labels, embedding-based thresholds, and human oversight creates a robust and adaptable framework, ensuring that both general-purpose and domain-specific graphs benefit from iterative error correction. By aligning LLMs' strengths with traditional embedding methods and selective expert guidance, our pipeline provides a reliable means of overcoming data scarcity and contextual ambiguity in KG completion tasks.

4.3 Embedding Model and LLM Combination

While our primary experiments employ DeepSeek-V3 for entity embeddings and DeepSeek-R1 for pseudo-labeling, we also investigate alternative approaches to evaluate their impact on the Junyi dataset. Table 1 reports the AUROC for each (Embedding, LLM) pairing in Junyi.

Overall, DeepSeek-V3 coupled with DeepSeek-R1 attains the highest AUROC (0.73), indicating a strong alignment between this embedding model and LLM. In contrast, BERT and node2vec typically yield AUROCs around $0.60 \sim 0.65$ for most language models, indicating that relying solely on general textual or structural signals may be insufficient for capturing fine-grained relationships.

Among the LLMs, DeepSeek-R1 generally outperforms Claude-3.5-sonnet, GPT-4o, and o1, often by a margin of $4 \sim 5$ points in AUROC. Nevertheless, o1 achieves respectable scores (e.g., 0.70 with DeepSeek-V3), showing that a moderate synergy still emerges when embeddings and LLMs offer complementary strengths. Meanwhile, Claude-3.5-sonnet and GPT-4o hover around $0.60 \sim 0.66$, suggesting they perform adequately but may benefit from additional calibration or prompts to excel.

Table 1: AUROC values of various (Embedding, LLM) combinations on the Junyi dataset.

Embedding	LLM	AUROC
BERT	Claude-3.5-sonnet	0.60
BERT	GPT-4o	0.60
BERT	o1	0.62
BERT	DeepSeek-R1	0.64
node2vec	Claude-3.5-sonnet	0.60
node2vec	GPT-4o	0.60
node2vec	o1	0.65
node2vec	DeepSeek-R1	0.65
text-embedding-3-small	GPT-4o	0.62
text-embedding-3-small	o1	0.66
text-embedding-3-small	DeepSeek-R1	0.71
voyage-3	Claude-3.5-sonnet	0.66
voyage-3	DeepSeek-R1	0.70
DeepSeek-V3	Claude-3.5-sonnet	0.63
DeepSeek-V3	GPT-4o	0.64
DeepSeek-V3	o1	0.70
DeepSeek-V3	**DeepSeek-R1**	**0.73**

In summary, these results illustrate the value of well-aligned embeddings and LLMs: pairings that closely match each other's strengths achieve higher AUROCs, while general-purpose combinations remain competitive but less optimal.

4.4 Ablation: Without Expert Feedback

To assess the impact of our expert-in-the-loop mechanism (Section 3.3), we conduct an ablation experiment removing human feedback entirely. In this "No Expert" variant, the pipeline relies solely on LLM-generated pseudo-labels and embedding-based threshold calibration, bypassing any selec-

Table 2: AUROC of Full vs. No Expert Feedback on two educational datasets. Removing expert checks lowers AUROC by 6–7 points.

	Junyi	**WDKG-KP.**
Full Pipeline	0.73	0.69
No Expert	0.66	0.62
Δ(avg.)	-7.1 pts.	-7.0 pts.

tive review of uncertain pairs. Table 2 compares AUROC scores for the full pipeline versus the No Expert setting across two representative datasets, Junyi and WDKG-KnowledgePoints.

Quantitatively, removing expert checks leads to a $7.0 \sim 7.1$ point drop in AUROC. Junyi experiences a 7-point decrease (from 0.73 to 0.66), while WDKG-KnowledgePoints decreases by 7 points. This pattern underscores the system's reliance on targeted expert validation for resolving ambiguous or borderline cases. For instance, in Junyi, fine-grained skill relationships (e.g., "Linear vs. Quadratic Functions") often confuse the LLM alone, resulting in false positives for closely related concepts and false negatives for slightly more distant ones. Embedding scores help filter some errors, but the pipeline still struggles to recognize overlaps or partial dependencies when experts do not intervene.

Qualitatively, analyzing misclassifications reveals that, in the No Expert condition, nearly half of the borderline pairs remain mislabeled, especially those near the similarity threshold or where the LLM's textual reasoning conflicts with the embedding-based signals. In the Full Pipeline, these pairs are escalated to expert review, where human oracle confirm or deny the relationship. Such selective validation consistently corrects errors that purely automated methods fail to catch.

Additionally, iteration matters: in the Full Pipeline, each confirmed or corrected label is folded back into our pseudo-labeled set, refining future prompts and embedding thresholds. Without this feedback loop, the pipeline stagnates at a lower plateau, never improving upon early misclassifications. These findings highlight that expert checks are not merely a post-processing step but a core driver of iterative accuracy gains.

Overall, the ablation demonstrates that expert involvement is pivotal for handling nuanced relationships, reducing false positives among closely related concepts and false negatives among conceptually adjacent topics. While LLM bootstrapping and embedding calibration provide a strong baseline, omitting selective human oversight can drop final AUROC by 5 points on average, affirming that expert-in-the-loop feedback is crucial for high-fidelity relationship inference. Extra ablation studies are left in Appendix F.

5 Conclusion

In this paper, we introduced an Expert-in-the-Loop Few-Shot pipeline for LLM-Assisted Knowledge Graph Relationship Inference, designed to address the cold-start problem that arises when little or no ground-truth data are available. Our approach combines LLM-driven bootstrapping, embedding-based threshold calibration, and selective human feedback, thereby aligning textual reasoning with data-driven signals while leveraging expert validation to resolve ambiguous or conflicting pairs.

A key novelty of our method is its iterative design, in which LLM outputs (pseudo-labeled pairs) are continuously refined through embedding checks and an active learning loop. Unlike purely unsupervised or purely LLM-based strategies, our pipeline leverages both semantic embeddings and targeted expertise without relying on extensive initial annotations. Experimental results across multiple datasets, ranging from synthetic to educational and general-purpose knowledge graphs, demonstrate that our framework consistently outperforms classic embedding-based baselines and more recent PLM/LLM-driven approaches.

The significance of this work stems from its practical and versatile solution to cold-start challenges in knowledge graph completion. By synthesizing LLM pseudo-labeling with expert checks, we can bootstrap relationship inference in new or sparsely labeled domains. Expert input is directed precisely where it has the greatest impact, on borderline or uncertain cases, thereby minimizing annotation costs while improving inference quality. Although we focused on prerequisite-like relations, the pipeline itself can be applied to other hierarchical or semantic dependencies in diverse knowledge graphs, making it broadly adaptable.

In future work, we aim to extend the pipeline to incorporate multi-hop reasoning, richer domain constraints, and more advanced forms of LLM self-consistency. Our results indicate a promising synergy between large language model reasoning and targeted human oversight, offering a feasible blueprint for constructing and maintaining accurate knowledge graphs even under severe data scarcity.

6 Limitations

Despite its promising results, EFP-KGRI has several limitations that merit further investigation. First, the reliance on LLM bootstrapping can introduce biases or hallucinations when the model encounters unfamiliar domain concepts, especially if its training data are limited in scope. While the active learning loop partially mitigates these issues through expert validation, large-scale or highly specialized knowledge graphs may still require more rigorous filtering or dedicated domain-tuned LLMs. Second, our method's efficiency depends on the cost and availability of expert reviewers. Although we minimize their workload by focusing on uncertain or disputed edges, domains with extensive complexity or frequent updates might strain this resource, suggesting a need for further automation or incremental learning strategies. Third, we currently assume that each knowledge graph entity has at least some textual descriptor or partial structural connectivity; truly sparse or text-scarce domains may demand more sophisticated representation or data enrichment methods. Lastly, our evaluation primarily centers on binary relations (e.g., "Yes" or "No" for a prerequisite edge), leaving open the question of how this pipeline handles multi-relational or multi-hop reasoning scenarios. Exploring these aspects will be crucial in extending the applicability and robustness of EFP-KGRI.

7 Ethical Considerations

EFP-KGRI relies on LLMs for pseudo-label generation, which raises potential bias and fairness issues. LLMs can exhibit systematic biases learned from their training data, such as skewed associations between certain entities or domains, and these biases may inadvertently influence the pseudo-labeled edges in a knowledge graph. While the active learning loop and expert validation help to identify and correct spurious or harmful edges, there is no guarantee that all problematic outputs will be caught, particularly when human reviewers have limited time or domain familiarity.

Additionally, EFP-KGRI emphasizes minimized expert workload by focusing on uncertain or borderline pairs. In practice, experts need to be aware that LLM-provided suggestions might contain subtle biases or stereotypical assumptions about certain topics. This can be especially relevant if the target domain involves sensitive data such as educational records, personal health information, or demographic attributes. We encourage implementers to adopt transparent labeling practices, including clear provenance for each pseudo-labeled edge, and to implement routine bias-checking audits, where domain experts or independent reviewers systematically assess potential misrepresentations or omissions introduced through the LLM.

Lastly, any system applying this pipeline must respect privacy and data protection regulations when dealing with sensitive or personally identifiable information (PII). If textual entity descriptions include PII or other confidential details, embedding and inference processes need secure handling to prevent inadvertent data leaks. As LLM usage evolves, new or updated privacy protocols (e.g., encryption, differential privacy) may become essential. Ensuring compliance with relevant data protection frameworks (e.g., GDPR) is paramount when deploying LLM-driven methods in real-world environments.

References

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.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Daniel Daza, Michael Cochez, and Paul Groth. 2021. Inductive entity representations from text via link prediction. In *Proceedings of The Web Conference 2021*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, et al. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54(4):1–37.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4763–4771.

Tara Safavi and Danai Koutra. 2020. CoDEx: A Comprehensive Knowledge Graph Completion Benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8328–8350. Association for Computational Linguistics.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. 2022. Sequence-to-sequence knowledge graph completion and question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

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 *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer.

Burr Settles. 2009. Active learning literature survey.

Thiviyan Thanapalasingam, Emile van Krieken, Peter Bloem, and Paul Groth. 2023. Intelligraphs: Datasets for benchmarking knowledge graph generation. *arXiv preprint arXiv:2307.06698*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR.

Aoran Wang, Chaoli Zhang, Jun Pang, and Qingsong Wen. 2024a. Evaluating large language models with educational knowledge graphs on prerequisite relationships. Accessed: 2025-02-03.

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)*. Association for Computational Linguistics.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024b. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. A survey of active learning for natural language processing. *arXiv preprint arXiv:2210.10109*.

A More Details about Datasets

We evaluate our method on six distinct datasets, IntelliGraphs-syn-types, CoDEx-S, DBE-KT22, Junyi, WDKG-Course, and WDKG-KnowledgePoints, which collectively cover both general-purpose and educational knowledge graph (KG) scenarios. The usage of these datasets consent to their licenses, respectively. The final four are sourced from a publicly available repository on evaluating large language models (LLMs) with educational KGs. Below, we briefly describe each dataset's domain, size, and key characteristics relevant to our relationship inference task.

IntelliGraphs-syn-types (with CC-BY 4.0 License) is a synthetic dataset designed to test knowledge graph methods under controlled conditions. It includes various typed entities (such as "concept," "topic," or "skill") that exhibit a range of potential relationships, though we focus on the target relation r for consistent evaluation. This dataset helps us observe how our method performs when the underlying graph structure and entity types are systematically generated or manipulated, offering insights into model robustness in low-noise, semi-artificial settings.

CoDEx-S (with MIT License) stems from the CoDEx benchmarks, which extract subsets of Wikidata for link prediction research. CoDEx-S is the smallest variant, curated to ensure adequate diversity in relation types while remaining computationally manageable. Entities in CoDEx-S represent real-world objects (e.g., historical figures, places, scientific concepts), and edges include relationships such as "part of," "subclass of," or "instance of." CoDEx-S covers a broad semantic range, making it well-suited for testing baseline performance and the general applicability of our pipeline.

DBE-KT22 (with CC-BY 4.0 License) focuses on a specialized educational or domain-based environment, where entities correspond to discrete learning units, and edges capture hierarchical or prerequisite-like dependencies. Though not as large as some open-domain KGs, DBE-KT22 features nuanced relationships that mirror real-world knowledge progression in specific subject areas. This characteristic underscores the importance of domain context when evaluating relationship inference, especially for step-by-step or hierarchical knowledge.

Junyi (with CC-BY 4.0 License) provides another educational knowledge graph, emphasizing skill-based or concept-based prerequisites within a K-12 tutoring context. The relationships frequently reflect "topic A must precede topic B," aligning closely with the notion of "depends on" or "precedes" at finer granularity. This dataset exposes how standard link prediction approaches—often tuned to broad, open-domain data—may struggle with the subtle dependencies in an educational domain.

Similarly, WDKG-Course and WDKG-KnowledgePoints (both with CC-BY 4.0 License) represent course-level and knowledge-point-level graphs, respectively, from the same repository. In WDKG-Course, each node is an entire course or module, and edges denote large-scale prerequisite pathways. In contrast, WDKG-KnowledgePoints drills down into individual concepts or competencies within those courses, offering a more granular perspective. These two datasets thus allow us to test whether our method can handle both coarse-grained and fine-grained learning structures, providing a comprehensive view of how well relationship inference performs across different levels of educational detail.

In all cases, the target relationship r aligns with a notion of "prerequisite," "precedes," or "depends on." Where official train/validation/test splits are available, we follow the original partitioning to maintain comparability. Otherwise, we randomly sample around 70% of the labeled edges for training (or pseudo-labeling), 10% for validation, and 20% for final testing. Negative examples are either derived via LLM-based generation (Section 3.1) or conventional negative sampling, ensuring each dataset offers balanced coverage of both positive and negative pairs. By combining synthetic data, general-domain subsets, and multiple education-focused graphs, we obtain a robust evaluation of our pipeline's ability to infer relationships in diverse knowledge graph contexts.

B More Details about Models

In our experiments, we leveraged five different large language model (LLM) configurations, GPT-4o, o1, Claude-3.5-sonnet, DeepSeek-V3(embedding variant) and DeepSeek-R1, to reflect a range of deployment methods and model specializations. Below, we clarify each model's versioning, usage environment, and key differences.

GPT-4o (gpt-4o-2024-08-06). We designate "gpt-4o" as a particular "optimized" GPT-4 re-

lease dated 2024-08-06, accessed via OpenAI API. This variant preserves GPT-4's advanced reasoning capabilities while prioritizing inference cost-effectiveness and slightly reduced context window. In practice, it balances robust zero-/few-shot performance with moderate throughput for large-scale knowledge graph tasks.

o1 (o1-2024-12-17). The "o1" label references a build dated 2024-12-17 that focuses on more detailed chain-of-thought or multi-step reasoning. Despite its smaller parameter count relative to GPT-4, o1 can excel in tasks demanding logical consistency. However, it may require more careful prompt engineering or domain adaptation to match GPT-4's broad topical coverage.

Claude-3.5-sonnet (claude-3-5-sonnet-20241022). Claude is an LLM developed under a different ecosystem (e.g., Anthropic), offering strong multi-turn reasoning and interpretability. The "3.5-sonnet" version (dated 2024-10-22) emphasizes structured outputs and coherent chain-of-thought. Despite robust performance in general dialogues, it occasionally underperforms in specialized "prerequisite" inference unless guided by domain-tailored prompts, possibly due to training corpus coverage.

DeepSeek-R1 A reasoning-oriented LLM tuned explicitly for tasks such as "prerequisite" detection or multi-step logical consistency. Through a domain-aligned training set, DeepSeek-R1 often demonstrated strong results on nuanced edge cases. In our pipeline, it served as one of the top performers for cold-start knowledge graph completion, especially when the domain required more in-depth interpretative reasoning.

DeepSeek-V3 (for embeddings) In addition to the above reasoning models, we rely on DeepSeek-V3 for entity embedding via Ollama. Unlike a general-purpose LLM, this version is tailored for vector encoding—mapping textual or partial adjacency data to dense representations. By integrating these embeddings with the LLM's textual classifications (e.g., from GPT-4o or DeepSeek-R1), we leverage both semantic similarity and chain-of-thought reasoning to refine relationship inference.

B.1 Practical Considerations

Performance vs. Cost. GPT-4 variants (GPT-4o) and Claude-3.5-sonnet tend to excel at zero/few-shot tasks but may have higher token usage fees. Meanwhile, DeepSeek-V3 and DeepSeek-R1 can be more cost-effective or easier to host locally, though they may need specialized prompts to handle complex queries.

Reasoning Depth and Domain Coverage. Models like o1 and DeepSeek-R1 prioritize multi-step logical consistency, which can be vital for hierarchical or prerequisite inferences. In contrast, a more general model like GPT-4o might not specialize in domain intricacies unless carefully prompted. The specific composition of the KG domain ultimately drives which model yields the best synergy.

Security and Data Privacy. For large or sensitive knowledge graphs, shipping data off to external APIs (GPT-4o, Claude) can pose privacy concerns. Local or on-premise solutions (e.g., DeepSeek-R1 with local implementation as it is open-sourced, and DeepSeek-V3 embeddings with Ollama) keep all data in-house at the possible cost of greater hardware requirements.

Iteration and Fine-Tuning. Each model is versioned by date (e.g., 2024-12-17 for o1, 2024-10-22 for Claude-3.5-sonnet). Substantial updates may alter their chain-of-thought policies or system instructions, affecting performance stability. In multi-iteration pipelines, minor parameter shifts can change how pseudo-labeled data evolves, emphasizing the need for periodic re-tuning.

C Hyperparameter Settings

In this section, we present additional details on the hyperparameters used throughout our experiments, covering both embedding-based methods and LLM-based pipelines.

C.1 Embedding Models and Threshold Tuning

Embedding Dimensionality. For *DeepSeek-V3* (with Ollama), we set the embedding dimension to $d = 768$. When using node2vec or other graph-based approaches, we set $d = 200$ unless noted otherwise. In preliminary trials, we found that increasing the dimension above 768 did not yield significant performance gains for text-based embeddings, while dimension sizes above 300 for node2vec sometimes led to overfitting on smaller datasets.

Training Epochs (Graph Embeddings). For node2vec, we ran 20–30 epochs on each dataset to ensure stable convergence, using a negative sampling ratio of $1 : 5$. We used SGD with an initial learning rate of 0.01 and decayed it by 0.9 every 5 epochs. In practice, the best final embeddings

emerged by epoch 20 for most graphs.

Threshold Calibration (α, τ). As described in Section 3.2 of the main paper, we combine LLM pseudo-labeling (L_{LLM}) with similarity-based scores (sim) using a hyperparameter α. We search α over $\{0, 0.25, 0.5, 0.75, 1.0\}$ or a similar small grid, and similarly sweep τ in the range $\{0, 0.1, 0.2, \ldots, 1.0\}$. The chosen (α^*, τ^*) maximizes AUROC (or occasionally F1 if the dataset is imbalanced) on a development set $\mathcal{D}_{\text{pseudo}}$.

Distance to Threshold (δ) for Borderline Cases. When identifying uncertain or borderline pairs for expert review, we set $\delta = 0.05$ unless otherwise stated. Empirically, values $\delta \in \{0.03, 0.05, 0.07\}$ yielded similar results; we selected 0.05 as a balance between capturing subtle disagreements and avoiding excessive labeling overhead.

Few-Shot Examples. In all "LLM Bootstrapping" phases, we typically provide 3-5 demonstration examples (with roughly equal "Yes" and "No" cases) to illustrate the target relationship. We observed diminishing returns after 5 examples, likely due to context overhead.

Self-Consistency Pass. We allocate a maximum of 1024 tokens for the LLM to re-check previously generated pairs. In practice, the self-consistency prompt rarely exceeds 300–400 tokens, even on the largest sets.

Active Learning Batch Size. In iterative pipelines, we typically review 10–20 borderline pairs at each step to minimize expert workload. Larger batch sizes (e.g., 20–50) may speed up final convergence but risk overburdening domain experts.

D Detailed Prompt Examples

Here, we provide verbatim samples of the few-shot prompts and self-consistency prompts used in our pipeline. These examples demonstrate how we instruct each LLM to output new pairs (positive or negative) and how we ask it to verify or flag contradictory pairs.

D.1 Few-Shot Prompt Snippet

> `[User Message]`
> **We have the following relationship:** "precedes" **(e.g., for hierarchical topics).**
> **Examples:**
> (Fractions, Decimals): Yes, because one must understand fractions before decimals.
> (Shakespeare, Calculus): No, because these are unrelated subjects.
> (Quadratic Equations, Polynomial Theorems): Yes, because advanced polynomial analysis needs quadratics.
> **Now please generate 5 new pairs where precedes" holds, and 5 pairs where precedes" does not hold.**

In the above prompt, we provide three demonstration pairs (two "Yes" and one "No") to illustrate the desired structure. The LLM is then asked to generate additional pairs, explicitly separating "Yes" from "No" outcomes.

D.2 Self-Consistency Verification Prompt

> `[User Message]`
> **You labeled the following pairs as Yes/No for the relationship "precedes":**
> (Counting Integers, Basic Probability): Yes
> (Shakespeare, French Grammar): No
> (Exponential Functions, Logarithms): Yes
> (Food Safety, Pythagorean Theorem): No
> **Please review each pair and mark any that seem incorrect or ambiguous, with a short explanation. If correct, say nothing.**

The LLM might respond with lines like:

```
INCORRECT: (Exponential Functions, Logarithms) because log is often taught alongside exponentials
AMBIGUOUS: (Food Safety, Pythagorean Theorem) because they are indeed unrelated, but there's no direct reason for a prerequisite
```

We parse this output and remove or re-check any pairs flagged as `INCORRECT` or `AMBIGUOUS`. The remaining examples become our $\mathcal{D}_{\text{pseudo}}$.

D.3 Domain-Specific Example (DBE-KT22)

Below is a short snippet adapted for a specialized dataset like **DBE-KT22**, which focuses on hierarchical or prerequisite-like dependencies between learning units:

> `[User Message]`
> **We have the following relationship:** "depends on" (e.g., a topic or concept must come before another in a database curriculum).
>
> **Examples:**
> (**Entity-Relationship Model**, **Enhanced Entity-Relationship**): **Yes**, because extended EER diagrams typically build on basic ER concepts.
> (**Locking**, **Enhanced Entity-Relationship**): **No**, because concurrency control is unrelated to advanced ER modeling techniques.
> (**Data Model**, **Data integrity**): **Yes**, because one needs to understand how data is structured (data model) before grasping the consistency/integrity constraints.
>
> **Now please generate 5 new pairs where "depends on" holds, and 5 new pairs where**

"depends on" does not hold.
Include short rationales for each decision.

And the potential output structure:

Yes: (Many-To-Many, Participation constraints), because ...
Yes: (Relational data model, Schema), because ...
No: (Authentication, Two-Phase Locking (2PL) Protocol), because ...
No: (EXISTS, Data integrity), because ...

And here is an example of self-consistency verification prompt:

`[User Message]`
You labeled the following pairs as Yes/No for the relationship "depends on":
(Subclass, Enhanced Entity-Relationship): **Yes**
(Locking, Database Security): **No**
(DDL, ALTER TABLE): **Yes**
Please review each pair and mark any that seem incorrect or ambiguous, with a short explanation. If correct, say nothing.

Here, we reference DBE-KT22 node names directly, using the same short references as in the `.graphml` (e.g., "Subclass" or "DDL"). The LLM's self-consistency check might result in responses like:

INCORRECT: (Subclass, Enhanced Entity-Relationship)
because "Subclass" is introduced after "Enhanced Entity-Relationship," not before.
AMBIGUOUS: (Locking, Database Security)
because concurrency control is sometimes covered alongside overall security measures.

We can then remove or re-label these flagged pairs, updating your $\mathcal{D}_{\text{pseudo}}$ set accordingly.

E More Experimental Settings

The running of the proposed method leverages a MacPro M3pro notebook with 96 GB memory and 8 NVIDIA A100 80GB GPUs.

F Further Ablation Studies

Table 3 compares our **full pipeline** (DeepSeek-V3 embeddings + DeepSeek-R1 LLM + expert feedback) against two ablated variants that remove either the embedding-based similarity ("No Similarity") or the LLM pseudo-labeling ("No LLM"). On the Junyi dataset, the full pipeline attains an AUROC of 0.73, whereas removing embeddings drops performance to 0.69, and discarding the LLM drives it to 0.62.

Table 3: Average AUROC on the Junyi dataset for ablation studies removing either similarity or LLM.

Method	AUROC
Full Pipeline	0.73
No Similarity (LLM + Expert Only)	0.69
No LLM (Embedding + Expert Only)	0.62

No Similarity (LLM + Expert) In this setting, the system relies entirely on LLM-proposed pairs and expert validation, foregoing the embedding-based confidence calibration. The resulting 0.69 AUROC indicates that expert intervention can partially mitigate LLM hallucinations, but the absence of numeric similarity cues hinders consistent filtering of borderline or domain-ambiguous cases.

No LLM (Embedding + Expert) Here, pseudo-label generation is omitted, and the system depends only on embedding-based similarity scores and expert checks. The 0.62 AUROC suggests that while embeddings capture some semantic closeness among knowledge points, they struggle to propose sufficiently diverse or context-specific candidate pairs. Without the LLM's generative insight, the active learning loop focuses too narrowly on already-known relationships, limiting the pipeline's recall.

Overall, these ablations confirm that **both** LLM-driven and embedding-based components are crucial. Removing either yields significantly lower AUROC, even when experts remain in the loop.

G Usage of AI Assistant

Throughout the development of this EFP-KGRI, we employed an AI Assistant, in particular, a LLM such as ChatGPT-4o or DeepSeek-R1 in a "developer assistance" capacity.

Although expert oversight ultimately governs the pipeline's final design, the AI Assistant proved valuable for draft refinement and code scaffolding, supplementing our manual process. All AI-generated outputs underwent human validation for correctness, relevance, and style. By adopting this hybrid approach, we capitalized on the AI Assistant's speed and flexibility without compromising on the accuracy or transparency required in a critical research environment.