Learning to Think on Hypergraph: HyperCoT for Structure-Guided N-ary Knowledge Graph Completion

Mengxue Yang[1], **Jinming Li**[1], **Chun Yang**[1], **Jiaqi Zhu**[1,2*], **Jiafan Li**[1,2], **Guanhua Zhang**[1,2], **Ying Li**[1]
[1]University of Chinese Academy of Sciences, Beijing, China
[2]Institute of Software, Chinese Academy of Sciences, Beijing, China
{yangmengxue20@mails.ucas.ac.cn, zhujq@ios.ac.cn, liying21@ucas.ac.cn}

Abstract

N-ary knowledge graph completion (KGC) aims to infer missing components in facts with multiple entities under distinct semantic roles, commonly formulated as a knowledge hypergraph link prediction task. Most embedding-based approaches score individual hyperedges relying on enriched structural representations, but overlook intermediate propagation states containing complementary local and global structural evidence. Despite their capability to generate chain-of-thought (CoT) representations for the classical KGC task, large language models (LLMs) struggle with hypergraph structure involving multiple facts, while current hypergraph QA methods only provide LLMs with a single query signal rather than path-level evidence. These limitations hinder the transferability of existing methods, especially those leveraging LLMs, to solve the knowledge hypergraph link prediction problem. To bridge this gap, we propose **HyperCoT**, a structure-aware approach that models multi-hop structural reasoning as a *depth-sensitive progressive evidence accumulation* process. It constructs a *Graphical Chain-of-Thought (Graph-CoT)* by aggregating role-aware hyperedge states along strongly correlated reasoning paths, and injects the resulting path-level structural evidence into each token in query and candidate entities to prompt LLMs. Experiments on three real-world datasets demonstrate that HyperCoT consistently outperforms strong n-ary KGC baselines, particularly in high arity and structural sparsity scenarios, meanwhile yielding interpretable multi-hop reasoning traces.

1 Introduction

Knowledge hypergraphs, also known as n-ary knowledge graphs, represent facts governed by high-order interactions and role-dependent structural constraints. For instance, a manufacturing event may involve distinct entities, such as a vehicle model, brand, engine type, factory, and target market, each playing a specific role. Such multi-arity facts are prevalent in various applications including question answering (Yani and Krisnadhi, 2021), recommendation (Chicaiza and Díaz, 2021), and decision making (Wang et al., 2023b). However, real-world knowledge hypergraphs are often incomplete, motivating the task of *knowledge hypergraph completion* (also called n-ary KGC), which aims to infer missing entities or relations within these multi-argument facts (Chen et al., 2020).

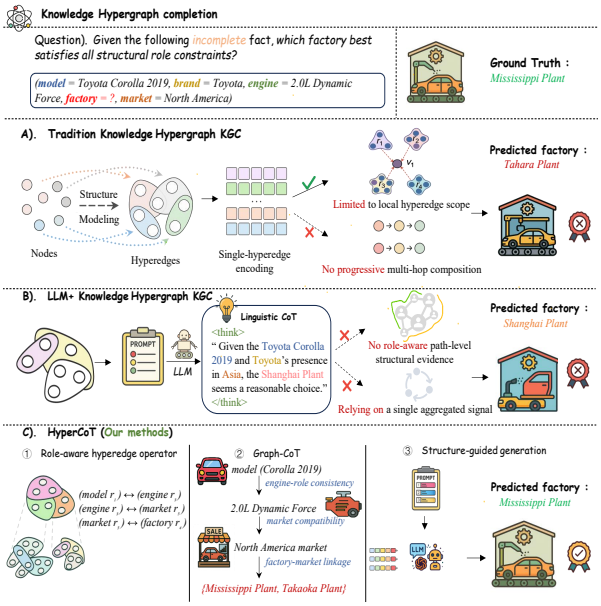

Figure 1: Motivating comparison of n-ary knowledge graph completion paradigms. Existing hypergraph models rely on single-hyperedge or final aggregated representations without progressive multi-hop evidence composition, while LLM-based approaches lack role-aware and path-level structural evidence.

Early approaches extend embedding-based models for classical KGC tasks or adopt reification-based transformations to score the plausibility of each hyperedge as a whole (Wen et al., 2016; Zhang et al., 2020). More recent GNN-based hy-

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 25562–25579
July 2-7, 2026

pergraph encoders establish interactions among the entities and their roles with incidence structures (Zhou et al., 2023; Wang et al., 2023a; Li et al., 2025). Although architecturally different, these methods largely follow a *local hyperedge encoding* paradigm: scoring each hyperedge via learning aggregated representations from its constituent nodes and edges, without explicitly modeling step-by-step correlations between hyperedges to support progressive reasoning. Thus, they fail to effectively integrate the diverse high-order constraints embedded within hypergraphs. As shown in Figure 1(a), this paradigm can lead to incorrect yet locally plausible predictions (e.g., *Tahara Plant*).

In parallel, large language models (LLMs) have reformulated knowledge graph completion as a text generation task and achieved promising results, but just in classical binary or triple-based settings that lack hyperedge structures (Wei et al., 2024; Yang et al., 2025a). While LLMs exhibit a strong ability to associate multiple facts at the linguistic level through chain-of-thought (CoT) prompting in hypergraph QA tasks (Feng et al., 2024; Luo et al., 2025), they are typically guided by the isolated query hyperedge or a single aggregated signal, rather than explicit *path-level structural evidence* for multi-hop reasoning. This often results in predictions that are linguistically coherent but structurally inconsistent (e.g. *Shanghai Plant* in Figure 1(b)).

These observations highlight a core challenge in n-ary KGC: accurate prediction requires not only expressive node/edge representations, but also *explicit and role-aware reasoning rationales that preserve and compose structural evidence across multiple hops of hyperedges*. Neither enriched hyperedge embeddings nor language-driven generation alone can fulfil this requirement.

To address this challenge, we propose **HyperCoT**, a structure-aware approach for n-ary knowledge hypergraph completion. HyperCoT encodes hyperedges as propagatable structural states and performs attention-driven multi-hop reasoning to extract depth-sensitive and path-level structural evidence. By preserving and aggregating intermediate hyperedge states along reasoning paths, it constructs an explicit *Graphical Chain-of-Thought* (Graph-CoT), capturing both local (direct, low-order) and global (indirect, high-order) correlations with varying degrees of contextual enhancement. This structured rationale is then injected into tokens of the query and candidate entities to prompt LLMs for structure-guided prediction.

Our contributions are summarized as follows:

- We introduce **HyperCoT**, the first LLM-based structure-aware approach for n-ary knowledge hypergraph completion, which stimulates the potential of LLMs on role-aware multi-hop structural reasoning.
- We propose a Graphical Chain-of-Thought construction mechanism that preserves and aggregates intermediate structural evidence along reasoning paths, facilitating depth-sensitive feature enhancement.
- We develop a structure-guided generation operator that injects the resulting path-level evidence into individual tokens to prompt LLMs, thereby improving both accuracy and interpretability of link prediction in hypergraphs.

Experiments on JF17K, WikiPeople, and FB-AUTO datasets show that HyperCoT consistently outperforms strong n-ary KGC baselines, especially in high arity and structural sparsity scenarios.

2 Related Work

We review related work on knowledge hypergraph completion and generation-based knowledge graph completion.

Knowledge Hypergraph Completion. Early knowledge graph completion methods focus on binary relations using embedding-based scoring functions, such as TransE (Bordes et al., 2013), DistMult (Yang et al., 2015), and RotatE (Sun et al., 2019). To extend these models to n-ary settings, prior work introduces reification, tuple representations, or role-aware projections, including m-TransH, NaLP, RAM, and ReAlE (Yang et al., 2015; Guan et al., 2019; Liu et al., 2021; Fatemi et al., 2023). More recent GNN-based methods model n-ary facts as hyperedges and perform message passing over incidence structures, such as HyperMLN, RD-MPNN, HyConvE, tNaLP+, and HyCubE (Chen et al., 2022; Zhou et al., 2023; Wang et al., 2023a; Guan et al., 2023; Li et al., 2025). Despite improved structural expressiveness, these approaches largely follow a *local hyperedge encoding* paradigm, scoring each hyperedge individually or based on a single aggregated representation, without preserving intermediate propagation states that encode depth-specific structural evidence across multiple hops.

Generation-based KGC. Another line of work formulates knowledge graph completion as a generation task, including sequence-to-sequence models such as KGT5 (Saxena et al., 2022), GenKGC (Xie et al., 2022), as well as recent LLM-based approaches leveraging instruction tuning or prompting with structural cues, e.g., DIFT (Liu et al., 2024), KICGPT (Wei et al., 2024), GS-KGC (Yang et al., 2025b), and SLiNT (Yang et al., 2025a). These methods demonstrate that LLMs are able to associate multiple facts during generation in classical binary or triple-based KGC task. However, in n-ary or hypergraph settings with multiple interdependent facts, existing approaches often degenerate to exposing only isolated hyperedges or a single aggregated structural signal, as commonly observed in hypergraph QA tasks, rather than explicit path-level structural evidence capturing complicated dependencies. In contrast, HyperCoT performs multi-hop structural reasoning over role-aware hyperedges, preserves intermediate propagation states, and injects the resulting path-level structural evidence into LLMs for structure-guided prediction.

3 Problem Formulation

3.1 Knowledge Hypergraph

Given a finite set of entities $\mathcal{V}$, relations $\mathcal{R}$, and n-ary facts $\mathcal{T}$, a *knowledge hypergraph* is defined as $\mathcal{H} = (\mathcal{V}, \mathcal{R}, \mathcal{T})$. Each fact $e \in \mathcal{T}$ is represented as an ordered tuple $e = (r; u_1, u_2, \ldots, u_n)$, where $r \in \mathcal{R}$ denotes an n-ary relation and $u_i \in \mathcal{V}$ represents the entity participating in the i-th semantic role of relation r. We denote the entity at position i of hyperedge e by $e(i) = u_i$. Notice that a knowledge graph is a special case of a knowledge hypergraph with $n = 2$, i.e., all relations are binary.

3.2 Knowledge Hypergraph Link Prediction

Let $\mathcal{T}_O \subseteq \mathcal{T}$ denote the observed hyperedges. We consider the n-ary knowledge hypergraph link prediction task under the *single missing entity* setting. Given an incomplete query hyperedge $e_q = (r; u_1, \ldots, u_{i-1}, ?, u_{i+1}, \ldots, u_n)$, the task is to predict the missing entity $u_i \in \mathcal{V}$ such that the completed hyperedge belongs to the unobserved set $\mathcal{T} - \mathcal{T}_O$. Following prior work, this task is formulated as ranking candidate entities from $\mathcal{V}$ conditioned on the known context $\tilde{u} = (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_n)$ and the relation r.

4 Method

We propose **HyperCoT**, a structure-aware approach for knowledge hypergraph link prediction built upon LLMs. HyperCoT models a query hyperedge as a propagatable high-order structural state, performs multi-hop reasoning over the hypergraph, and aggregates structural evidence into a *Graphical Chain-of-Thought (Graph-CoT)* for structure-guided prediction. An overview of the approach is shown in Figure 2.

4.1 High-Order Structural State Learning

For an n-ary hyperedge, semantics arise from interactions among entities playing different roles under a given relation. To preserve such role-dependent interactions, HyperCoT constructs a *high-order structural state* by projecting each entity instance into a relation- and role-aware subspace:

$$h_{e,u_i} = W_{r(e),i}\, \hat{h}_{e,u_i}, \tag{1}$$

where $\hat{h}_{e,u_i}$ denotes the raw feature of entity u_i when it appears at the i-th role of hyperedge e, and $W_{r(e),i} \in \mathbb{R}^{d\times d}$ is a learnable projection matrix specific to the relation $r(e)$ and the role index i. In practice, each $W_{r(e),i}$ is parameterized in a low-rank form $W_{r(e),i} = A_{r(e)} B_i^\top$, where $A_{r(e)}, B_i \in \mathbb{R}^{d\times d_r}$ and $d_r \ll d$.

To capture explicit interactions among different roles under a given relation, we define a learnable role interaction operator that jointly encodes the corresponding role-aware entity embeddings and the relation embedding:

$$\psi_{i,j}(e) = \sigma\big(W_\psi \big[\, h_{e,u_i} \,\|\, h_{e,u_j} \,\|\, h_r \,\big]\big), \tag{2}$$

where $\|$ denotes concatenation, h_r is the embedding of relation r, $W_\psi \in \mathbb{R}^{d'\times 3d}$ is a learnable parameter matrix, and $\sigma(\cdot)$ is instantiated as the ReLU activation function (Nair and Hinton, 2010). The resulting representation $\psi_{i,j}(e)$ captures second-order and relation-conditioned interactions between the role pair (i, j), and serves as a compact structural feature for subsequent aggregation.

Then, all of these interaction features for role pairs are aggregated using structural attention:

$$\alpha_{ij} = \mathrm{softmax}_{i,j}\Big(u^\top \psi_{i,j}(e)\Big), \tag{3}$$

where $u \in \mathbb{R}^{d'}$ is a learnable attention vector. The initial hyperedge structural state is computed as

$$h_e^{(0)} = W_s \sum_{i,j} \alpha_{ij}\, \psi_{i,j}(e), \tag{4}$$

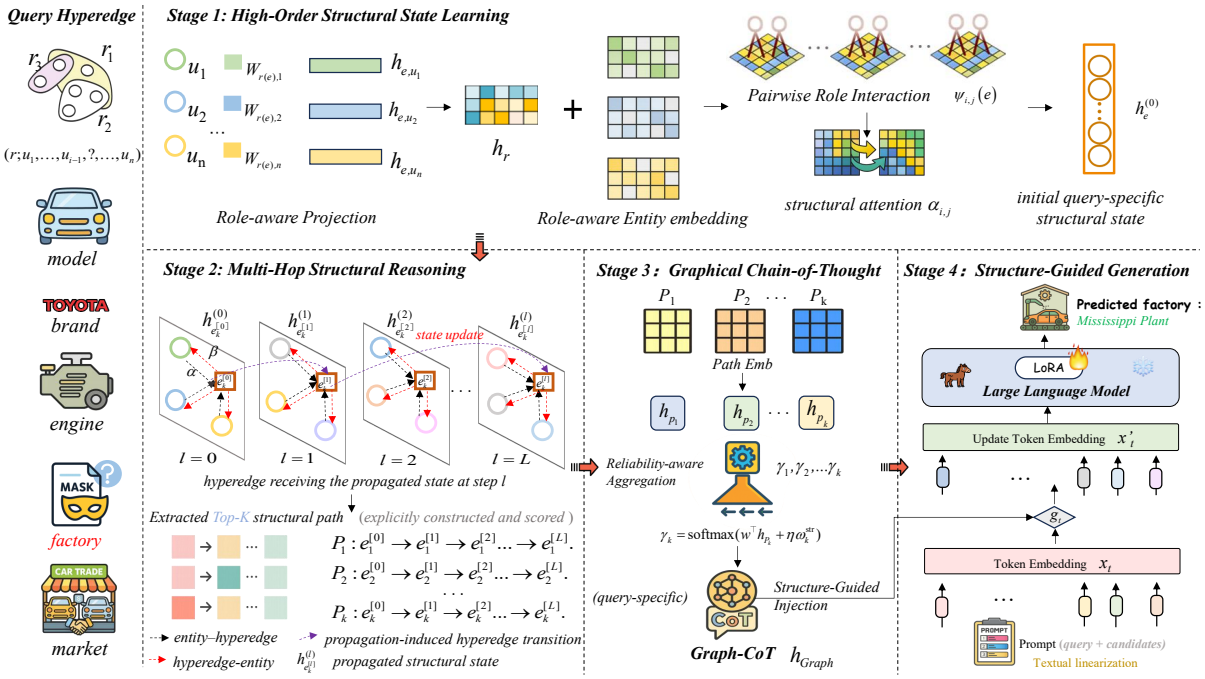

Figure 2: **Overview of HyperCoT.** Given an n-ary query hyperedge $e_q = (r; u_1, \ldots, u_{i-1}, ?, u_{i+1}, \ldots, u_n)$, HyperCoT proceeds in four stages: (1) high-order structural state learning, (2) progressive multi-hop propagation, (3) Graph-CoT construction via structural path extraction and aggregation, and (4) structure-guided generation. The model learns an initial structural state, performs multi-hop reasoning to extract top-K structural paths, aggregates them into a Graph-CoT representation, and injects the resulting path-level evidence into the LLM for prediction.

where $W_s \in \mathbb{R}^{d \times d'}$ maps interaction features to the structural state space.

4.2 Multi-Hop Structural Reasoning

Multi-hop structural reasoning in HyperCoT consists of two sub-steps: (1) progressive propagation for obtaining depth-specific structural states, and (2) structural path extraction with depth-sensitive path encoding.

Step 1: Progressive multi-hop propagation. HyperCoT iteratively propagates a query-conditioned structural state over the incidence structure of the knowledge hypergraph. Each entity v is initialized with embeddings $h_v^{(0)}$, and each hyperedge e starts from a structural state $h_e^{(0)}$ (Eq. 4), which are mutually updated in a shared space.

Let $h_v^{(l)}$ and $h_e^{(l)}$ denote the structural states after the l-th aggregation step, respectively. We denote by $\mathcal{N}_{\mathrm{E}}(v) = \{\, e \mid v \in e \,\}$ the set of hyperedges incident to entity v. At each reasoning step, HyperCoT performs bidirectional propagation to exchange semantic information between entities and hyperedges, updating hyperedge states from participating entities and entity states from incident hyperedges:

$$
\begin{aligned}
h_e^{(l+1)} &= \sum_{v \in e} \alpha_{v,e}^{(l)}\, h_v^{(l)}, \\
h_v^{(l+1)} &= \sum_{e \in \mathcal{N}_{\mathrm{E}}(v)} \beta_{e,v}^{(l)}\, h_e^{(l)},
\end{aligned}
\tag{5}
$$

where $\alpha_{v,e}^{(l)}$ and $\beta_{e,v}^{(l)}$ are normalized attention weights computed via dot-product between entity and hyperedge states, followed by softmax normalization over the corresponding neighborhoods.

Step 2: Structural path extraction and depth-sensitive encoding. Based on the propagated structural states, HyperCoT explicitly extracts structural paths with depth as a key factor. Early hops near the query mainly capture local role compatibility, while later hops encode more global and indirect structural constraints. Here, L denotes the depth of multi-hop reasoning, i.e., the number of propagation steps used to extract L-hop structural paths. Let $\mathcal{N}_{\mathrm{H}}(e)$ denote the set of hyperedges that share at least one entity with hyperedge e, and the set of candidate L-hop paths can be defined as:

$$
\widetilde{\mathcal{P}}_L = \left\{ \langle e^{[0]}, \ldots, e^{[l]} \rangle \mid e^{[0]} = e_q,\ e^{[l+1]} \in \mathcal{N}_{\mathrm{H}}(e^{[l]}) \right\}. \tag{6}
$$

To preserve depth-specific evidence, we synchronize the propagation process with path expansion here, such that the l-th hyperedge along path P_k corresponds to its propagated state $h^{(l)}_{e_k^{[l]}}$ after the l-th propagation step. Accordingly, for two consecutive hyperedges $e^{[l]}$ and $e^{[l+1]} \in \mathcal{N}_{\mathrm{H}}(e^{[l]})$ along a candidate path, we define the propagation-induced transition score at step l as:

$$\pi^{[l]}\left(e^{[l]} \to e^{[l+1]}\right) = \sum_{v \in e^{[l]} \cap e^{[l+1]}} \alpha^{(l)}_{v,e^{[l]}}\, \beta^{(l)}_{e^{[l+1]},v}. \tag{7}$$

Each candidate structural path $\langle e^{[0]}, \ldots, e^{[l]} \rangle$ is then scored by its cumulative transition strength:

$$\mathrm{Score}(\langle e^{[0]}, \ldots, e^{[l]} \rangle) = \sum_{l=0}^{L-1} \log \pi^{[l]}\left(e^{[l]} \to e^{[l+1]}\right). \tag{8}$$

Since $|\widetilde{\mathcal{P}}_L|$ grows exponentially with L, we apply beam-style pruning based on the path scores above to retain the top-K structural paths $\{P_k\}_{k=1}^{K}$. The complete procedure is provided in Appendix B.

Each path is embedded via depth-sensitive aggregation:

$$h_{P_k} = \sum_{l=1}^{L} \omega_l^{[k]}\, h^{(l)}_{e_k^{[l]}}, \tag{9}$$

where the hop-specific weights are calculated by

$$\omega_l^{[k]} = \frac{\pi^{[l-1]}\left(e_k^{[l-1]} \to e_k^{[l]}\right)}{\sum_{j=1}^{L} \pi^{[j-1]}\left(e_k^{[j-1]} \to e_k^{[j]}\right)}. \tag{10}$$

4.3 Graphical Chain-of-Thought Construction

Given the extracted structural path embeddings $\{h_{P_k}\}_{k=1}^{K}$, HyperCoT aggregates them into a single query-specific structural representation, termed *Graphical Chain-of-Thought (Graph-CoT)*.

Formally, the Graph-CoT for a query hyperedge e_q is constructed as

$$h_{\text{Graph}}(e_q) = \sum_{k=1}^{K} \gamma_k\, g(h_{P_k}), \tag{11}$$

where $g(\cdot)$ is a ReLU-based projection function that maps path embeddings into the shared structural representation space. The aggregation weight γ_k reflects the relative contribution of each structural path and is computed by combining learned path relevance and structure-aware reliability:

$$\gamma_k = \mathrm{softmax}_k\left(w^\top h_{P_k} + \eta\, \pi_k^{\text{str}}\right), \tag{12}$$

where w is a learnable scoring vector, η is a scalar hyperparameter, and π_k^{str} is a path-level structural reliability score that summarizes the propagation-induced transition scores along path P_k. The first term $w^\top h_{P_k}$ measures the learned relevance of the path embedding, while the second term $\eta\pi_k^{\text{str}}$ reflects how reliably structural constraints are preserved across multiple hops. Theoretical and empirical support for this design is provided in Appendix A, where we present explanatory analysis of Graph-CoT aggregation together with a controlled pairwise preference experiment to verify the role of intermediate states in reasoning.

4.4 Structure-Guided Generation

HyperCoT incorporates the constructed Graph-CoT into prediction via a structure-guided generation operator, where structural reasoning is performed entirely in the hypergraph space and injected to guide language-based scoring.

Given a query hyperedge e_q, we linearize it into a structured natural language prompt using a fixed template (see Appendix E), which specifies the incomplete n-ary fact and a constrained candidate entity set. In the implementation, we further introduce special anchor tokens `[QUERY]` and `[ENTITY]` to mark the missing argument position and the candidate entity positions, respectively. Let x_t denote the embedding of the t-th token in the resulting prompt. For each token position corresponding to these anchor tokens, Graph-CoT is injected through the following gating mechanism:

$$\begin{aligned} x'_t &= g_t \odot h_{\text{Graph}}(e_q) + (1 - g_t) \odot x_t, \\ g_t &= \sigma\big(W_g \left(h_{\text{Graph}}(e_q) \parallel x_t\right)\big), \end{aligned} \tag{13}$$

where $g_t \in \mathbb{R}^d$ is a learnable gating vector conditioned on the original token embedding and the Graph-CoT representation, $W_g \in \mathbb{R}^{d \times 2d}$ is a learnable projection matrix, and $\odot$ denotes element-wise multiplication. All the token embeddings outside the anchor positions remain unchanged. Since HyperCoT preserves intermediate states to retain depth-specific structural evidence across hops, we adopt gating rather than direct concatenation to inject structural evidence selectively, thereby reducing the impact of noisy or conflicting path-level signals.

Then, the structure-modulated embeddings $\{x'_t\}_{t=1}^{T}$ are fed into an LLM to obtain a conditional distribution:

$$p(\cdot \mid e_q, h_{\text{Graph}}(e_q)) = \mathrm{LLM}\big(\{x'_t\}_{t=1}^{T}\big). \tag{14}$$

Since the LLM is autoregressive, this conditional distribution is factorized over output tokens. For scalability, the prediction is performed under a candidate-constrained setting. An embedding-based n-ary link prediction model NaLP (Guan et al., 2019) is employed to retrieve the top-20 candidates, yielding the candidate set $\mathcal{C}(e_q)$. The language model scores only this reduced set in a discriminative manner.

Specifically, each candidate entity $v \in \mathcal{C}(e_q)$ is scored by the conditional log-likelihood of its tokenized name:

$$s(v) = \sum_{m=1}^{|v|} \log p\big(t_m^v \mid e_q, h_{\text{Graph}}(e_q), t_{<m}^v\big)\,, \tag{15}$$

where $\{t_m^v\}_{m=1}^{|v|}$ denotes the token sequence of entity v. Note that it is not a separate scoring function, but the token-level realization of the distribution above. In this way, the final prediction result can be expressed as

$$\hat{v} = \arg\max_{v \in \mathcal{C}(e_q)} s(v). \tag{16}$$

4.5 Training Objective

HyperCoT is trained by jointly optimizing entity prediction accuracy and structural alignment:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda \mathcal{L}_{\text{align}}, \tag{17}$$

where $\mathcal{L}_{\text{pred}}$ is the negative log-likelihood of the ground-truth entity under the candidate-constrained scoring distribution (Eq. 15), and $\mathcal{L}_{\text{align}}$ regularizes the consistency between the constructed Graph-CoT representation and the correct entity in the structural embedding space. The scalar λ controls the strength of structural regularization. Details of $\mathcal{L}_{\text{align}}$ are provided in Appendix F.

5 Experiments

Our experiments are designed to answer the following four research questions: (RQ1) Does HyperCoT outperform state-of-the-art n-ary KGC methods across diverse benchmarks? (RQ2) Are the performance gains mainly attributable to explicit multi-hop structural reasoning rather than stronger encoders or larger model capacity? (RQ3) How robust is HyperCoT under challenging structural conditions, such as structural sparsity and incompleteness? (RQ4) How sensitive is HyperCoT to its hyperparameter settings?

5.1 Experimental Setup

Datasets. We evaluate on three n-ary knowledge hypergraph benchmarks: JF17K (Liu et al., 2021), FB-AUTO (Fatemi et al., 2020), and WikiPeople (Guan et al., 2019), which vary in arity, sparsity, and structural complexity. Detailed descriptions and statistics are provided in Appendix D.1.

Baselines. We compare HyperCoT with representative n-ary KGC methods from four categories: (1) **Embedding-based models**, including RAE (Zhang et al., 2018) and NaLP (Guan et al., 2019); (2) **Semantic matching models**, including HypE (Fatemi et al., 2020), RAM (Liu et al., 2021), PosKHG (Chen et al., 2023), and ReAlE (Fatemi et al., 2023); (3) **GNN-based models**, including HyperMLN (Chen et al., 2022), tNaLP+ (Guan et al., 2023), RD-MPNN (Zhou et al., 2023), HyConvE (Wang et al., 2023a), and the cubical structural encoders HyCubE/HyCubE+ (Li et al., 2025); and (4) **Generation-based models**,including KICGPT (Wei et al., 2024), DIFT (Liu et al., 2024), and SLiNT (Yang et al., 2025a), which are adapted to our n-ary setting for comparison. In addition, we compare HyperCoT with several LLM baselines under both fine-tuned and zero-shot settings, focusing on models with comparable capacity (e.g., 7B–10B). Detailed descriptions and configurations are provided in Appendix D.2.

Evaluation Metrics. Following standard practice, we report Mean Reciprocal Rank (MRR) and Hits@K ($K = 1, 3, 10$) under the filtered evaluation protocol for single missing entity assumption.

Training Setup. HyperCoT is trained on LLaMA-3-8B-Instruct using a parameter-efficient fine-tuning strategy. Detailed implementation and hyperparameter settings are provided in Appendix D.2.

5.2 Main Results (RQ1)

Table 1 summarizes the performance of HyperCoT on three representative benchmarks.

Overall performance across datasets. Overall, HyperCoT achieves consistently strong results across all datasets, demonstrating robust generalization under various structural settings. On JF17K, which exhibits complex relational structures and diverse role configurations, HyperCoT achieves the best overall performance, improving MRR from 0.584 (HyCubE) to 0.599 and consistently outper-

Table 1: Link prediction results on three n-ary knowledge hypergraph benchmarks. We compare HyperCoT with embedding-based, semantic matching, GNN-based, and generation-based baselines. Best results are shown in **bold**, and second-best ones are underlined. Results not reported in the original papers and obtained locally are marked with †.

Model	JF17K				WikiPeople				FB-AUTO			
	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10	MRR	Hits@1	Hits@3	Hits@10
RAE	0.392	0.312	0.433	0.561	0.253	0.118	0.343	0.463	0.703	0.614	0.764	0.854
NaLP	0.310	0.239	0.334	0.450	0.338	0.272	0.362	0.466	0.672	0.611	0.712	0.774
HypE	0.494	0.399	0.532	0.650	0.263	0.127	0.355	0.486	0.804	0.774	0.824	0.856
RAM	0.539	0.463	0.573	0.690	0.363	0.271	0.405	0.500	0.830	0.803	0.851	0.876
PosKHG	0.545	0.469	0.582	0.706	0.315†	0.214†	0.377†	0.475†	0.856	0.821	0.876	0.895
ReAlE	0.530	0.454	0.563	0.677	0.332†	0.207†	0.417†	0.514†	0.861	0.836	0.877	0.908
HyperMLN	0.556	0.482	0.597	0.717	0.351†	0.270†	0.394†	0.497†	0.831	0.803	0.851	0.877
tNaLP+	0.449	0.370	0.484	0.598	0.339	0.269	0.369	0.473	0.729	0.645	0.748	0.826
RD-MPNN	0.512	0.445	0.573	0.685	–	–	–	–	0.810	0.714	0.880	0.888
HyConvE	0.580	0.478	0.610	0.729	0.362	0.275	0.388	0.501	0.847	0.820	0.872	0.901
HyCubE	0.584	0.508	0.616	0.730	0.448	**0.368**	0.490	0.592	0.881	0.860	0.894	0.918
HyCubE+	0.582	0.511	0.611	0.720	0.433	0.347	0.478	0.591	0.891	0.872	0.901	0.923
KICGPT	0.416	0.362	0.489	0.501	0.287	0.236	0.322	0.413	0.586	0.512	0.597	0.623
DIFT	0.491	0.403	0.533	0.657	0.314	0.257	0.354	0.453	0.632	0.597	0.679	0.701
SLiNT	0.497	0.416	0.547	0.669	0.319	0.263	0.362	0.465	0.645	0.609	0.673	0.717
HyperCoT (Ours)	**0.599**	**0.527**	**0.634**	**0.743**	**0.451**	0.366	**0.495**	**0.612**	**0.904**	**0.883**	**0.912**	**0.927**

forming all baselines across evaluation metrics. On the more sparse and noisy WikiPeople dataset, HyperCoT attains the highest MRR (0.451) and competitive Hits@1/10, indicating improved robustness under incomplete and noisy structural contexts. On FB-AUTO, which features strong type constraints and well-structured relations, HyperCoT achieves the best overall ranking performance, improving Hits@10 by $+0.012$ over HyCubE+. We further analyze the computational efficiency and overhead of HyperCoT in Appendix C.

Comparison with structural and semantic baselines. HyperCoT consistently outperforms embedding-based and semantic matching models, which rely on local or role-level scoring without explicit multi-hop composition. For example, HyperCoT surpasses HyperMLN by $+0.043$ MRR on JF17K and $+0.100$ on WikiPeople, validating the benefit of preserving intermediate structural states in the reasoning process.

Comparison with GNN-based models. GNN-based models propagate information via message passing, but they often summarize structural cues from different depths into a single representation, which may blur the distinction between early-hop role compatibility and later-hop global correlations. By explicitly constructing and aggregating multi-hop reasoning paths, HyperCoT consistently outperforms these models, e.g., by $+0.087$ MRR over RD-MPNN and $+0.019$ over HyConvE on JF17K, and $+0.057$ over HyConvE on FB-AUTO.

Comparison with generation-based models. We further compare HyperCoT with recent generation-based KGC models, including KICGPT, DIFT, and SLiNT, which are adapted to the n-ary setting in our experiments. We can see that HyperCoT performs consistently better than these methods across all three datasets, with particularly clear gains on JF17K and WikiPeople. This suggests that the advantage of HyperCoT comes not merely from adopting an LLM-based generation framework, but from explicitly modeling intermediate structural evidence during multi-hop n-ary reasoning.

Comparison with language models in comparable scale. We evaluate several language models with comparable capacity (7B–10B parameters) under consistent fine-tuning or zero-shot protocols to form fair and reasonable baselines. As shown in Table 2, LLaMA-3-8B-Instruct performs best among these models and is thus adopted as the backbone for HyperCoT. Under the same setting, HyperCoT yields $+0.044$ MRR improvement on JF17K, demonstrating the benefit of injecting explicit structural evidence.

5.3 Ablation Study (RQ2)

To assess the contribution of each key component in HyperCoT, we conduct ablation studies by removing or simplifying one component at a time while keeping all others fixed. The evaluated variants include: **(1) w/o High-order Structural State Learning**, removing role-aware structural encoding; **(2) w/o Progressive Multi-hop Reasoning**,

Table 2: Comparison with same-scale LLMs on JF17K.

Model (7B–10B)	MRR	Hits@1	Hits@10
LLaMA-2-7B (FT)	0.489	0.401	0.655
Mistral-7B-Instruct (FT)	0.512	0.419	0.671
Qwen-2-7B-Chat (FT)	0.524	0.436	0.682
Gemma-2-9B-Instruct (FT)	0.538	0.447	0.695
GPT-4o-mini (Zero-shot)	0.521	0.432	0.681
LLaMA-3-8B-Instruct (FT)	0.551	0.463	0.708
HyperCoT (Ours)	**0.599**	**0.527**	**0.743**

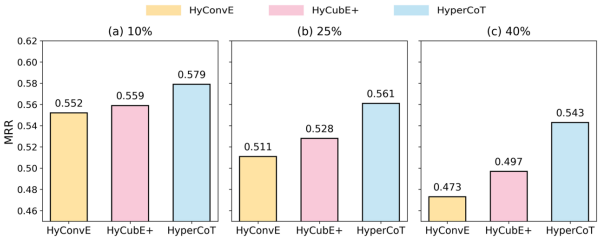

Figure 3: MRR degradation under random hyperedge deletion on JF17K.

disabling iterative structural propagation and utilizing only the initial query state; **(3) w/o Intermediate State Aggregation**, retaining multi-hop propagation but using only the final structural state; **(4) w/o Path-level Adaptive Aggregation**, replacing path-level weighting with uniform averaging; **(5) w/o Structural Alignment**, removing the structural–semantic alignment loss; **(6) w/o Structural Injection**, disabling all structure-guided information in the LLM input; **(7) w/o Graph-CoT Representation**, converting the extracted reasoning paths into textual descriptions and providing them to the same LLM (LLaMA-3-8B-Instruct) without the intermediate state injection of Graph-CoT representation.

Ablation Analysis. Table 3 reports the ablation results of HyperCoT across three benchmarks. Removing any major component consistently degrades performance, confirming that all modules contribute to the effectiveness of the full framework. Among all variants, disabling *progressive multi-hop reasoning* and using only the initial query state leads to the largest performance drop across datasets, showing that local hyperedge representations alone are insufficient for n-ary knowledge graph completion. Even when multi-hop propagation is retained, removing *intermediate state aggregation* and relying solely on the final structural state still reduces prediction accuracy. This verifies that intermediate hyperedge states encode complementary structural evidence beyond a single final signal. We further compare HyperCoT with a *Structural Prompt Only* variant, where reasoning paths are verbalized as textual structural prompts to the same LLM (LLaMA-3-8B-Instruct) without embedding-level intermediate state injection. Although this variant performs better than removing structure-guided information entirely, it consistently underperforms full HyperCoT on all three datasets. This result indicates that the gains of HyperCoT do not mainly come from textualized structural prompting alone. Instead, they should be attributed to explicitly preserving and injecting intermediate reasoning states through Graph-CoT, which provides richer path-level structural evidence than surface-form prompt descriptions.

The contribution of other components is somewhat dataset-dependent. On JF17K and WikiPeople, high-order structural state learning and path-level adaptive aggregation are more important, while on FB-AUTO the model is relatively more sensitive to structure-guided injection.

5.4 Structural Robustness Analysis (RQ3)

We evaluate the robustness of HyperCoT in the scenarios of structural sparsity and graph incompleteness.

(1) Degree Sparsity. We group entities by degree and evaluate performance on the bottom 20%. As shown in Table 4, HyperCoT consistently outperforms strong baselines in this long-tail regime, demonstrating stronger generalization under sparse structural context.

(2) Random Edge Drop. We randomly remove 10–40% of hyperedges to simulate incomplete hypergraphs. Figure 3 shows the resulting MRR degradation on JF17K, demonstrating improved robustness of HyperCoT to structural incompleteness. Qualitative case studies are provided in Appendix G.

5.5 Hyperparameter Sensitivity Analysis (RQ4)

We analyze the sensitivity of HyperCoT on key hyperparameters that control its structural reasoning behaviors.

Effect of reasoning depth L. We vary the reasoning depth $L \in \{1, 2, 3, 4\}$. As shown in Fig. 4(a), performance improves from $L = 1$ to $L = 3$ across all datasets, reflecting the significance of progressive multi-hop structural reasoning. However, further increasing the depth to $L = 4$ yields marginal

Table 3: Ablation results across three benchmarks. Best results are shown in **bold**, and second-best underlined.

Variant	JF17K				WikiPeople				FB-AUTO			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
w/o High-order Structural State Learning	0.574	0.501	0.616	0.733	0.432	0.346	0.474	0.595	0.889	0.868	0.900	0.926
w/o Progressive Multi-hop Reasoning	0.564	0.487	0.602	0.720	0.421	0.331	0.460	0.574	0.882	0.861	0.896	0.919
w/o Intermediate State Aggregation	0.581	0.513	0.621	0.731	0.436	0.349	0.480	0.600	0.893	0.871	0.904	0.924
w/o Path-level Adaptive Aggregation	0.579	0.504	0.618	0.734	0.437	0.351	0.482	0.603	0.896	0.874	0.906	0.923
w/o Structural Alignment	0.583	0.511	0.623	0.741	0.438	0.353	0.484	0.604	0.894	0.872	0.904	0.922
w/o Structural Injection	0.587	0.514	0.627	0.740	0.441	0.357	0.488	0.607	0.891	0.870	0.902	0.923
w/o Graph-CoT Representation	0.592	0.518	0.628	0.737	0.443	0.359	0.490	0.609	0.897	0.875	0.907	0.924
Full HyperCoT	**0.599**	**0.527**	**0.634**	**0.743**	**0.451**	**0.366**	**0.495**	**0.612**	**0.904**	**0.883**	**0.912**	**0.927**

Table 4: MRR on low-degree entities (bottom 20%).

Model	JF17K	WikiPeople	FB-AUTO
HyConvE	0.421	0.298	0.762
HyCubE+	0.458	0.331	0.787
HyperCoT	**0.482**	**0.354**	**0.801**

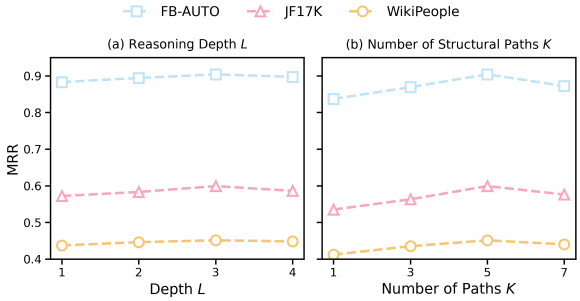

Figure 4: Sensitivity analysis of key hyperparameters. (a) Effect of reasoning depth L. (b) Effect of structural path number K.

or degraded performance, indicating diminishing returns from overly deep reasoning, which would introduce noise. Accordingly, we set $L = 3$ as the default depth.

Effect of structural path number K. We further examine the impact of the number of extracted Graph-CoT paths $K \in \{1, 3, 5, 7\}$. As shown in Fig. 4(b), performance improves consistently as K increases up to 5, confirming the benefit of aggregating multiple high-quality reasoning paths. When K is further increased to 7, performance saturates or slightly degrades, indicating the negative effect of redundant paths. Thus, we adopt $K = 5$ in all experiments.

Sensitivity analyses of additional hyperparameters, including the structural alignment weight λ and the path reliability coefficient η, are reported in Appendix D.3.

6 Conclusion

We presented **HyperCoT**, a structure-aware LLM-based approach for n-ary knowledge graph completion that integrates multi-hop structural reasoning with structure-guided generation. By modeling and aggregating *intermediate hyperedge states* along reasoning paths, HyperCoT constructs a Graphical Chain-of-Thought that captures depth-sensitive structural evidence beyond a single final representation, and effectively stimulates the potential of LLMs in the underexplored field, knowledge hypergraph link prediction. Experiments across multiple benchmarks show that HyperCoT consistently improves performance under high-arity and structurally sparse settings, highlighting the importance of progressive structural evidence accumulation for reliable n-ary knowledge graph completion.

Limitations

HyperCoT is designed to operate on structured knowledge hypergraphs and focuses on reasoning over relational structures. While this work does not explicitly incorporate multimodal signals, extending the approach to support multimodal or weakly structured inputs is a natural direction for future research. Moreover, HyperCoT assumes that the input hypergraph is already available, and does not address the upstream problem of automatic structure construction. Handling noise introduced during automatic construction or extraction remains an important research topic for future work.

Ethical Considerations

All datasets used are public and contain no sensitive information. No human subjects are involved.

Acknowledgments

We would like to thank the reviewers and area chairs for their valuable feedback and constructive suggestions, which helped improve this work. This

research was supported in part by the MIIT Project on Industrial Real-Time Database Based on Next-Generation Information Technology (TC210804D), the CAS Project for Young Scientists in Basic Research (YSBR-040), and the National Key R&D Program of China (2023YFC3010700).

References

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

Zhe Chen, Yuehan Wang, Bin Zhao, Jing Cheng, Xin Zhao, and Zongtao Duan. 2020. Knowledge graph completion: A review. *IEEE Access*, 8:192435–192456.

Zirui Chen, Xin Wang, Chenxu Wang, and Jianxin Li. 2022. Explainable link prediction in knowledge hypergraphs. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 262–271. ACM.

Zirui Chen, Xin Wang, Chenxu Wang, and Zhao Li. 2023. Poskhg: A position-aware knowledge hypergraph model for link prediction. *Data Sci. Eng.*, 8(2):135–145.

Janneth Chicaiza and Priscila Valdiviezo Díaz. 2021. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Inf.*, 12(6):232.

Bahare Fatemi, Perouz Taslakian, David Vázquez, and David Poole. 2020. Knowledge hypergraphs: Prediction beyond binary relations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2191–2197. ijcai.org.

Bahare Fatemi, Perouz Taslakian, David Vázquez, and David Poole. 2023. Knowledge hypergraph embedding meets relational algebra. *J. Mach. Learn. Res.*, 24:105:1–105:34.

Yifan Feng, Chengwu Yang, Xingliang Hou, Shaoyi Du, Shihui Ying, Zongze Wu, and Yue Gao. 2024. Beyond graphs: Can large language models comprehend hypergraphs? *Preprint*, arXiv:2410.10083.

Google DeepMind. 2024. Gemma 2 technical report. Technical report, Google DeepMind.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Saiping Guan, Xiaolong Jin, Jiafeng Guo, Yuanzhuo Wang, and Xueqi Cheng. 2023. Link prediction on n-ary relational data based on relatedness evaluation. *IEEE Trans. Knowl. Data Eng.*, 35(1):672–685.

Saiping Guan, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2019. Link prediction on n-ary relational data. In *The World Wide Web Conference*, WWW '19, page 583–593, New York, NY, USA. Association for Computing Machinery.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Zhao Li, Xin Wang, Jun Zhao, Wenbin Guo, and Jianxin Li. 2025. Hycube: Efficient knowledge hypergraph 3d circular convolutional embedding. *IEEE Trans. Knowl. Data Eng.*, 37(4):1902–1914.

Yang Liu, Xiaobin Tian, Zequn Sun, and Wei Hu. 2024. Finetuning generative large language models with discrimination instructions for knowledge graph completion. In *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part I*, volume 15231 of *Lecture Notes in Computer Science*, pages 199–217. Springer.

Yu Liu, Quanming Yao, and Yong Li. 2021. Role-aware modeling for n-ary relational knowledge bases. In *Proceedings of the Web Conference 2021*, WWW '21, page 2660–2671, New York, NY, USA. Association for Computing Machinery.

Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Ze-min Kuang, Meina Song, Yifan Zhu, and Luu Anh Tuan. 2025. Hypergraphrag: Retrieval-augmented generation with hypergraph-structured knowledge representation. *CoRR*, abs/2503.21322.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814. Omnipress.

OpenAI. 2024. GPT-4o mini: Advancing cost-efficient intelligence.

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 (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2814–2828. Association for Computational Linguistics.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Chenxu Wang, Xin Wang, Zhao Li, Zirui Chen, and Jianxin Li. 2023a. Hyconve: A novel embedding model for knowledge hypergraph link prediction with convolutional neural networks. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 188–198. ACM.

Yu Wang, Feng Ye, Binquan Li, Gaoyang Jin, Dong Xu, and Fengsheng Li. 2023b. Urbanfloodkg: An urban flood knowledge graph system for risk assessment. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 2574–2584, New York, NY, USA. Association for Computing Machinery.

Yanbin Wei, Qiushi Huang, James T. Kwok, and Yu Zhang. 2024. KICGPT: large language model with knowledge in context for knowledge graph completion. *CoRR*, abs/2402.02389.

Jianfeng Wen, Jianxin Li, Yongyi Mao, Shini Chen, and Richong Zhang. 2016. On the representation and embedding of knowledge bases beyond binary relations. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 1300–1307. AAAI Press.

Xin Xie, Ningyu Zhang, Zhoubo Li, Shumin Deng, Hui Chen, Feiyu Xiong, Mosha Chen, and Huajun Chen. 2022. From discrimination to generation: Knowledge graph completion with generative transformer. In *Companion of The Web Conference 2022, Virtual Event / Lyon, France, April 25 - 29, 2022*, pages 162–165. ACM.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mengxue Yang, Chun Yang, Jiaqi Zhu, Jiafan Li, Jingqi Zhang, Yuyang Li, and Ying Li. 2025a. SLiNT: Structure-aware language model with injection and contrastive training for knowledge graph completion. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13658–13671, Suzhou, China. Association for Computational Linguistics.

Rui Yang, Jiahao Zhu, Jianping Man, Hongze Liu, Li Fang, and Yi Zhou. 2025b. GS-KGC: A generative subgraph-based framework for knowledge graph completion with large language models. *Inf. Fusion*, 117:102868.

Mohammad Yani and Adila Alfa Krisnadhi. 2021. Challenges, techniques, and trends of simple knowledge graph question answering: A survey. *Inf.*, 12(7):271.

Fuxiang Zhang, Xin Wang, Zhao Li, and Jianxin Li. 2020. Transrhs: A representation learning method for knowledge graphs with relation hierarchical structure. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 2987–2993. ijcai.org.

Richong Zhang, Junpeng Li, Jiajie Mei, and Yongyi Mao. 2018. Scalable instance reconstruction in knowledge bases via relatedness affiliated embedding. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1185–1194. ACM.

Xue Zhou, Bei Hui, Ilana Zeira, Hao Wu, and Ling Tian. 2023. Dynamic relation learning for link prediction in knowledge hypergraphs. *Appl. Intell.*, 53(22):26580–26591.

A Theoretical Justification of Graph-CoT Reasoning

A.1 Path-Level Aggregation vs. Local Hyperedge Encoding

Setup. Let $\mathcal{H} = (\mathcal{V}, \mathcal{R}, \mathcal{T})$ be a knowledge hypergraph. Given a query hyperedge $e_q = (r; u_1, \ldots, u_{i-1}, ?, u_{i+1}, \ldots, u_n)$, HyperCoT first constructs an initial *local* structural state $h_{e_q}^{(0)} \in \mathbb{R}^d$ using the high-order representation operator in Section 4.1. For brevity, we denote this local representation by $h_0 := h_{e_q}^{(0)}$.

HyperCoT then applies the L-step structural propagation operator described in Section 4.2, producing hyperedge states $\{h_e^{(l)}\}_{l=0}^{L}$ in the shared space $\mathbb{R}^d$, where (l) denotes the l-th propagation step. Let $\widetilde{\mathcal{P}}_L$ be the candidate length-L path space defined in Section 4.2, and let $\mathrm{TopK}(\widetilde{\mathcal{P}}_L)$ return the top-K paths after beam pruning.

For each selected path $P_k = (e_k^{[0]}, \ldots, e_k^{[l]})$, where $e_k^{[0]} = e_q$, its embedding h_{P_k} is computed by the depth-aware aggregation in Section 4.2. Finally, Graph-CoT is obtained by the path aggregation in Eq. (11).

We show that the proposed Graph-CoT representation strictly generalizes a local hyperedge encoding.

Proposition 1 (Local Encoding as a Special Case)**.** *When $L = 0$, no multi-hop propagation is performed, and Graph-CoT reduces to the initial structural state of the query hyperedge. In this case, the model degenerates to a local hyperedge encoding.*

Proof. Choose $L = 0$ so that no propagation is applied and the path space contains the single zero-hop path $P_1 = (e_1^{[0]})$ with $e_1^{[0]} = e_q$. By definition, $h_{P_1} = h^{(0)}_{e_1^{[0]}} = h^{(0)}_{e_q} = h_0$. Set $K = 1$, $\gamma_1 = 1$, and let g be the identity map. Then $h_{\text{Graph}}(e_q) = g(h_{P_1}) = h_0$. □

Implication. Proposition 1 implies that Graph-CoT is at least as expressive as local hyperedge encoding. Allowing $L > 0$ and $K > 1$ enables incorporating multi-hop structural dependencies that cannot be captured by local representations alone.

A.2 Stability Motivation of Top-K Path Aggregation

Aggregating all reachable paths can be undesirable, as weakly supported paths may introduce noisy or conflicting evidence, so HyperCoT aggregates only the Top-K structurally reliable paths. We provide a variance-based justification under a standard estimator view.

Path-level evidence as random variables. For a fixed candidate entity $y \in \mathcal{V}$, treat the evidence contributed by a structural path P_k as a random variable

$$X_{P_k}(y) = \left\langle h_{P_k},\, h_y^{[0]} \right\rangle, \tag{18}$$

where $h_y^{[0]} \in \mathbb{R}^d$ is the structural embedding of entity y in the shared space, consistent with the initialization in Section 4.2. Randomness may arise from attention estimation, incomplete neighborhoods, approximate propagation or beam pruning.

Given a subset of paths $\mathcal{S}$, define the aggregated evidence estimator as

$$\widehat{C}_{\mathcal{S}}(y) = \frac{1}{|\mathcal{S}|} \sum_{P_k \in \mathcal{S}} X_{P_k}(y). \tag{19}$$

Proposition 2 (Top-K Aggregation Reduces Variance under Reliability Ordering)**.** *Under the assumption that the variance of path evidence increases with decreasing path reliability, Top-K aggregation can reduce variance by focusing on the most reliable paths. This offers a stability mechanism that suppresses noise in weakly supported paths and enhances the final aggregated representation.*

Proof. By independence,

$$\mathrm{Var}\Big[\widehat{C}_{\mathcal{S}}(y)\Big] = \frac{1}{|\mathcal{S}|^2} \sum_{P_k \in \mathcal{S}} \mathrm{Var}[X_{P_k}(y)]. \tag{20}$$

Under the stated reliability ordering, enlarging $\mathcal{S}$ by adding less reliable (higher-variance) paths increases the average variance term, yielding a larger (or no smaller) estimator variance. □

A.3 Empirical Support for Intermediate State Reasoning

Controlled Pairwise Preference Experiment. To directly test whether intermediate states in reasoning affect the ranking rule, we conduct a controlled pairwise preference experiment. For each query, we select candidate pairs with similar embedding similarity but different structural consistency, where one candidate satisfies the multi-hop structural constraints and the other does not. We then measured how often the model ranks the structurally consistent entity higher than the inconsistent one. Local encoding (final-state only) achieved 56%, while HyperCoT (with intermediate states) achieved 68%. This shift away from random preference is consistent with our interpretation that intermediate states in reasoning influence the ranking rule, rather than serving only as a theoretical description.

Algorithm 1: Attention-Induced Structural Path Extraction

Input: Query hyperedge e_q; entity–hyperedge attention weights $\{\alpha_{v,e}^{(l)}, \beta_{e,v}^{(l)}\}_{l=0}^{L-1}$; reasoning depth L; beam size K

Output: Top-K structural paths $\{P_k\}_{k=1}^{K}$

Initialize beam $\mathcal{B}_0 = \{((e_1^{[0]}),\ 0)\}$ with $e_1^{[0]} = e_q$, where each element is a (path, score) pair;

for $l = 0$ **to** $L-1$ **do**

  Initialize empty beam $\mathcal{B}_{l+1}$;

  foreach $(P_k, s) \in \mathcal{B}_l$ **do**

    Let $e_k^{[l]}$ be the last hyperedge in path P_k;

    foreach $e_k^{[l+1]} \in \mathcal{N}_H(e_k^{[l]})$ **do**

      Compute transition score: $\pi^{[l]}(e_k^{[l]} \to e_k^{[l+1]}) = \sum_{v \in e_k^{[l]} \cap e_k^{[l+1]}} \alpha_{v,e_k^{[l]}}^{(l)} \beta_{e_k^{[l+1]},v}^{(l)}$.

      Add $\big(P_k \cup \{e_k^{[l+1]}\},\ s + \log \pi^{[l]}(e_k^{[l]} \to e_k^{[l+1]})\big)$ to $\mathcal{B}_{l+1}$;

  Keep the top-K elements in $\mathcal{B}_{l+1}$ ranked by score;

return $\mathcal{B}_L$

B Algorithmic Details

B.1 Attention-Induced Structural Path Extraction

Algorithm 1 presents the detailed procedure for extracting *attention-induced structural paths* described in Section 4.2. The algorithm explicitly follows the multi-hop propagation and path scoring formulation in the main text.

Remarks. The algorithm performs a beam search over hyperedges starting from the query hyperedge $e_k^{[0]} = e_q$. At each step l, transitions between hyperedges are induced by the entity–hyperedge attention weights $\alpha_{v,e_k^{[l]}}^{(l)}$ and $\beta_{e_k^{[l+1]},v}^{(l)}$ obtained from the l-th propagation step described in Section 4.2. The algorithm thus provides an explicit procedural realization of the attention-induced structural paths used in the main text.

C Computational Efficiency Analysis

We analyze the computational cost of HyperCoT from two perspectives: structural parameter size and end-to-end inference latency. Table 5 compares the structural parameter size (measured in millions of parameters) of HyperCoT with representative hypergraph-based KGC models, where the statistics for prior methods are taken from (Li et al., 2025) under identical experimental settings. Although HyperCoT introduces additional cost due to explicit multi-hop structural reasoning, this overhead remains manageable in practice because the reasoning depth L and the number of selected paths K are both kept small.

Table 5: Structural parameter comparison between hypergraph-based KGC models and HyperCoT. Best results are shown in **bold**, and second-best results are underlined.

Model	Structural Parameters (M)		
	JF17K	WikiPeople	FB-AUTO
PosKHG	14.34	27.53	1.65
HyConvE	12.80	21.44	4.80
ReAlE	14.88	29.61	1.64
HyCubE	**1.28**	**2.24**	**0.96**
HyCubE+	5.77	13.46	1.28
HyperCoT	11.52	19.20	3.84

Table 6: End-to-end latency breakdown of HyperCoT per query.

Stage	Time (s)	Ratio
Retrieval (NaLP)	0.005	2%
Graph-CoT reasoning	0.020	8%
LLM scoring	0.225	90%
Total	0.250	100%

To further clarify the runtime cost, Table 6 reports the end-to-end latency breakdown per query. As shown in the table, the dominant inference cost comes from LLM scoring, while Graph-CoT reasoning contributes only a modest additional overhead. Compared with LLM scoring alone (approximately 0.230s per query), HyperCoT adds about 9% extra latency while yielding substantially better ranking performance.

D Experimental Details and Hyperparameter Analysis

D.1 Dataset Statistics and Structural Properties

We evaluate HyperCoT on three widely used benchmarks for n-ary knowledge graph completion: JF17K (Liu et al., 2021), FB-AUTO (Fatemi et al., 2020), and WikiPeople (Guan et al., 2019). Table 7 summarizes their dataset statistics. Beyond scale, we emphasize their structural characteristics, which

are critical for assessing multi-hop and structure-aware reasoning.

JF17K. JF17K is a large-scale n-ary knowledge graph derived from Freebase, containing 28,645 entities and 322 relations. Its facts possess arities ranging from 2 to 6, with a substantial portion of higher-arity relations. As shown in Table 7, more than 45% of the facts involve arity ≥ 3, making JF17K a challenging benchmark that requires modeling role-dependent interactions and compositional constraints across multiple arguments.

WikiPeople. WikiPeople is constructed from Wikipedia tables and infoboxes, representing one of the largest benchmarks for n-ary reasoning. It contains 47,765 entities and 707 relations, with arities ranging from 2 to 9. The dataset is dominated by binary facts, but still includes a non-trivial number of ternary and higher-arity instances. This mixture of low- and high-arity facts makes WikiPeople suitable for evaluating the robustness of models under heterogeneous structural complexity.

FB-AUTO. FB-AUTO is a domain-specific n-ary knowledge graph in the automotive domain, consisting of 3,388 entities and 8 relations. Although smaller in scale, FB-AUTO exhibits a high proportion of higher-arity facts (arity ≥ 4), reflecting complex real-world relational structures such as manufacturing, ownership, and production chains. This dataset is structurally sparse and requires leveraging indirect evidence across multiple hops, making it particularly suitable for evaluating explicit structural reasoning and interpretability.

Structural Implications. Across all datasets, higher-arity facts account for a significant fraction of the data, highlighting the limitations of methods designed primarily for binary relations. Moreover, the presence of structurally sparse entities and long-tail role combinations motivates the necessity for path-level reasoning mechanisms such as Graph-CoT, which can aggregate indirect structural evidence beyond local hyperedges.

D.2 LLM-based Baseline Models and Training Configuration

LLM-based Baseline Models. We compare HyperCoT against a diverse set of strong large language model (LLM) baselines, including both open-source and proprietary models, under fine-tuned and zero-shot settings.

LLaMA-2-7B (Touvron et al., 2023) is a 7B-parameter decoder-only LLM pretrained on large-scale general-domain corpora and fine-tuned on each benchmark using the same supervision as HyperCoT.

Mistral-7B-Instruct (Jiang et al., 2023) is a 7B instruction-tuned LLM optimized for following natural language prompts and fine-tuned on the target tasks.

Qwen-2-7B-chat (Yang et al., 2024) is a 7B instruction-tuned model from the Qwen2 series designed for efficient reasoning under limited computational budgets.

Gemma-2-9B-Instruct (Google DeepMind, 2024) is an instruction-tuned model from the Gemma-2 series with 9B parameters, focusing on improved reasoning and generation quality.

GPT-4o-mini (OpenAI, 2024) is a proprietary LLM evaluated in zero-shot settings using task-specific prompts.

LLaMA-3-8B-Instruct (Grattafiori et al., 2024) is a fine-tuned baseline sharing the same backbone as HyperCoT, but without structure-aware reasoning.

Implementation Details. HyperCoT is implemented in PyTorch. The decoder is instantiated with a LLaMA-3-8B-Instruct model and fine-tuned using Low-Rank Adaptation (LoRA) (Hu et al., 2022), while the backbone parameters remain frozen. LoRA is applied with rank $r = 128$, scaling factor $\alpha = 64$, and dropout rate of 0.1.

Unless otherwise specified, we use $L = 3$ for structural reasoning layers and select the $K = 5$ for top-K structural paths in Graph-CoT aggregation. The structural aggregation coefficient η in Eq. 12 controls the contribution of individual structural paths and is fixed to 0.5 across all experiments.

Training is performed using the AdamW optimizer with a learning rate of 2×10^{-4}. Model checkpoints are selected based on validation MRR. All experiments are conducted on up to two NVIDIA H100 GPUs (80GB) under a consistent hardware environment.

D.3 Sensitivity to Additional Structural Hyperparameters

We analyze the sensitivity of HyperCoT to two additional hyperparameters that control structure-aware learning and aggregation: the structural alignment weight λ in the training objective (Eq. 17) and the path reliability coefficient η in

Table 7: Statistics of the three datasets.

Dataset	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	Arity	#Train	#Valid	#Test	#Arity=2	#Arity=3	#Arity=4	#Arity≥5
JF17K	28,645	322	2–6	61,104	15,275	24,568	54,627	34,544	9,509	2,267
WikiPeople	47,765	707	2–9	305,725	38,223	38,281	337,914	25,820	15,188	3,307
FB-AUTO	3,388	8	2,4,5	6,778	2,255	2,180	3,786	–	215	7,212

Table 8: Effect of structural alignment weight λ on MRR.

λ	JF17K (MRR)	WikiPeople (MRR)	FB-AUTO (MRR)
0.1	0.576	0.437	**0.904**
0.3	**0.599**	0.440	0.883
0.5	0.551	**0.451**	0.870
0.7	0.532	0.427	0.866

Table 9: Effect of path reliability coefficient η on MRR.

η	JF17K(MRR)	WikiPeople(MRR)	FB-AUTO(MRR)
0.0	0.581	0.438	0.892
0.3	0.593	0.446	0.899
0.5	**0.599**	**0.451**	**0.904**
0.7	0.592	0.443	0.897

Graph-CoT aggregation (Eq. 12).

D.3.1 Structural Alignment Weight λ

We first examine the effect of the structural alignment weight λ, which balances entity prediction accuracy and structural regularization. Table 8 reports the MRR of HyperCoT under different λ values across three benchmarks.

Overall, HyperCoT exhibits stable performance over a broad range of λ values, but with clear dataset-dependent preferences. On JF17K, moderate alignment weights (e.g., $\lambda = 0.3$) yield the best performance, suggesting a balanced contribution between entity prediction and structural regularization. On WikiPeople, slightly larger λ values are favored, indicating a stronger reliance on structural alignment under sparse and heterogeneous settings. In contrast, FB-AUTO achieves optimal performance with smaller λ, reflecting its stronger type constraints and more regular relational structure. Across all datasets, performance degrades only when structural regularization is overly emphasized, demonstrating that HyperCoT is robust to moderate variations in λ.

D.3.2 Path Reliability Coefficient η

We further analyze the sensitivity to the path reliability coefficient η, which controls the contribution of structure-aware path reliability in Graph-CoT aggregation (Eq. 12). Table 9 reports the MRR under different η values.

In summary, HyperCoT performs stably when the coefficient η changes. Moderate values (e.g., $\eta = 0.5$) consistently achieve the best or near-best performance across datasets. Very small η underweights structural reliability, while overly large η overemphasizes path consistency and slightly degrades performance. Unless otherwise specified, we set $\eta = 0.5$ in all experiments.

E Prompt Template

HyperCoT linearizes each query hyperedge into a structured natural language prompt that specifies the incomplete n-ary fact and a constrained candidate entity set. In the implementation, we introduce two special anchor tokens, `[QUERY]` and `[ENTITY]`, to mark the missing argument position and the candidate entity positions, respectively. Graph-CoT is injected only at these anchor-token positions using the gating mechanism defined in Eq. 13, while all other token embeddings remain unchanged.

Figure 5 shows the human-readable prompt template augmented with anchor tokens, and Figure 6 gives a concrete example used for candidate-constrained decoding.

F Structural Alignment Loss

The structural alignment loss $\mathcal{L}_{\text{align}}$ enforces the consistency between the Graph-CoT representation and the ground-truth entity in the shared structural embedding space.

Formally, given a query hyperedge e_q and its ground-truth entity v^+, the alignment loss is defined as:

$$\mathcal{L}_{\text{align}} = -\log \frac{\exp\big(\cos\big(h_{\text{Graph}}(e_q),\, h_{v^+}\big)\big)}{\sum_{v \in \mathcal{C}(e_q)} \exp\big(\cos\big(h_{\text{Graph}}(e_q),\, h_v\big)\big)}, \tag{21}$$

where $h_{\text{Graph}}(e_q)$ denotes the Graph-CoT representation, h_v is the structural embedding of entity v, and $\mathcal{C}(e_q)$ is the candidate entity set. This formulation encourages the aggregated structural evidence to be aligned with the correct entity while discriminating against competing candidates.

Prompt Template for an Incomplete n-ary Fact

Instruction:
You are given an incomplete n-ary fact with one missing entity. Your task is to infer the missing entity based on the given relation, known arguments, and structural consistency.

`[Fact:]`
Relation type: `relation`
`[Arguments:]`

- $Role_1$: $entity\_name_1$
- $Role_2$: $entity\_name_2$
- $Role_i$: **[MISSING]** `[QUERY]`
- ...
- $Role_n$: $entity\_name_n$

`[Candidate Entities:]`
The missing entity must be selected from the following candidates:

- $Candidate_1$ `[ENTITY]`
- $Candidate_2$ `[ENTITY]`
- $Candidate_3$ `[ENTITY]`
- ...

`[Task:]`
Select the entity that best satisfies the relation and is most structurally consistent with the given arguments. You must choose exactly one entity from the candidate list.
Output format: Output only the entity name, without explanation.

Figure 5: Prompt template for structure-guided link prediction.

Concrete Prompt Example for an Incomplete n-ary Fact

Instruction:
You are given an incomplete n-ary fact with one missing entity. Your task is to infer the missing entity based on the given relation, known arguments, and structural consistency.

`[Fact:]`
Relation type: `award_nomination`
`[Arguments:]`

- nominee : Leonardo DiCaprio
- award : Academy Award
- year : 2016
- category : **[MISSING]** `[QUERY]`
- ...

`[Candidate Entities:]`
The missing entity must be selected from the following candidates:

- Best Actor `[ENTITY]`
- Best Supporting Actor `[ENTITY]`
- Best Director `[ENTITY]`
- ...

`[Task:]`
Select the entity that best satisfies the relation and is most structurally consistent with the given arguments. You must choose exactly one entity from the candidate list.
Output format: Output only the entity name, without explanation.

Figure 6: Concrete prompt example for structure-guided link prediction.

G Case Studies and Interpretability Analysis

G.1 Case Studies Across Datasets

We present representative case studies from three datasets with distinct structural characteristics, including both successful and failure cases. For each dataset, we analyze an incomplete query hyperedge together with the top-ranked structural paths induced by HyperCoT. For clarity, we display the top-3 paths among the top-5 used during inference. Concrete entity names are used for readability.

G.1.1 Case Study on FB-AUTO (Successful Case)

This case involves a high-arity automotive manufacturing relation with one missing argument, where the goal is to identify the manufacturing plant of a specific vehicle configuration. As shown in Table 10, HyperCoT retrieves multiple structural paths that connect the vehicle model to candidate plants through complementary production-related relations. While each path captures only partial evidence, their aggregation jointly enforces consistency across model, brand, engine, and market information. Consequently, HyperCoT correctly ranks *Mississippi Plant* as the most plausible completion.

G.1.2 Case Study on JF17K (Failure Case)

We next present a failure case from JF17K that highlights the limitation of structure-guided reasoning under structural ambiguity. The query corresponds to the relation `cvg.musical_game_song_relationship`, where the task is to predict the artist of a game soundtrack given the game and song. As shown in Table 11, the extracted structural paths connect *Final Fantasy VII* and *One-Winged Angel* to multiple candidate artists via contribution and group membership relations. However, these paths provide comparable structural support for different artists and fail to impose a role-specific constraint that uniquely identifies the ground-truth answer. As a result, the aggregated Graph-CoT ranks *Masashi Hamauzu* above the correct artist *Nobuo Uematsu*.

G.1.3 Case Study on WikiPeople (Successful Case)

This case is drawn from the WikiPeople dataset and illustrates a qualifier-aware role completion scenario with temporal constraints. The query requires predicting a position held by a person during a specific time interval. As shown in Table 12, HyperCoT extracts multiple query-conditioned structural paths that explicitly associate candidate positions with their corresponding starting and ending times. Only the paths corresponding to *President of the United States* satisfy the temporal constraints between 2009 and 2017. By aggregating these paths, HyperCoT suppresses temporally incompatible roles and correctly identifies the ground-truth position.

G.2 Discussion: Interpretability of Graph-CoT

Across all datasets, these case studies demonstrate that Graph-CoT provides explicit and verifiable structural evidence supporting the model's predictions. By exposing ranked multi-hop reasoning traces, HyperCoT enables transparent analysis of how structural information is utilized, in contrast to embedding-based or purely generative approaches that do not offer explicit reasoning paths.

Table 10: Graph-CoT reasoning example for a successful case on FB-AUTO. Entity identifiers are mapped to surface names for readability.

Incomplete Query Hyperedge	(model; *Toyota Corolla 2019*, *2.0L Engine*, *Toyota*, **?**, *North America*)
Top-3 Structural Paths	**Path 1 (direct model–plant evidence):** (model; Toyota Corolla 2019, 2.0L Engine, Toyota, ?, North America) → (manufacturing_plant_model_relationship; Toyota Corolla 2019, Mississippi Plant) **Path 2 (brand-mediated structural evidence):** (model; Toyota Corolla 2019, 2.0L Engine, Toyota, ?, North America) → (make; Toyota Corolla 2019, Toyota) → (manufacturing_plant_model_relationship; Toyota, Mississippi Plant) **Path 3 (engine-consistent production evidence):** (model; Toyota Corolla 2019, 2.0L Engine, Toyota, ?, North America) → (engine; Toyota Corolla 2019, 2.0L Engine) → (manufacturing_plant_engine_relationship; 2.0L Engine, Mississippi Plant)

Table 11: Graph-CoT reasoning example for a failure case on JF17K. Entity identifiers are mapped to surface names for readability.

Incomplete Query Hyperedge	(cvg.musical_game_song_relationship; *Final Fantasy VII*, **?**, *One-Winged Angel*)
Top-3 Structural Paths	**Path 1 (direct contribution evidence):** (cvg.musical_game_song_relationship; Final Fantasy VII, ?, One-Winged Angel) → (music.recording_contribution; One-Winged Angel, Masashi Hamauzu) **Path 2 (group-mediated participation evidence):** (cvg.musical_game_song_relationship; Final Fantasy VII, ?, One-Winged Angel) → (music.recording_contribution; One-Winged Angel, Nobuo Uematsu) → (music.group_membership; Nobuo Uematsu, Square Enix Music) **Path 3 (game-level performance evidence):** (cvg.musical_game_song_relationship; Final Fantasy VII, ?, One-Winged Angel) → (cvg.game_performance; Square Enix Music, Final Fantasy VII)

Table 12: Graph-CoT reasoning example for a successful case on WikiPeople. Entity identifiers are mapped to surface names for readability.

Incomplete Query Hyperedge	(P39 / P580 / P582; *Barack Obama*, **?**, 2009, 2017)
Top-3 Structural Paths	**Path 1 (position–time consistency evidence):** (P39 / P580 / P582; Barack Obama, ?, 2009, 2017) → (P39; Barack Obama, President of the United States) → (P580; President of the United States, 2009) → (P582; President of the United States, 2017) **Path 2 (citizenship-constrained role evidence):** (P39 / P580 / P582; Barack Obama, ?, 2009, 2017) → (P27; Barack Obama, United States) → (P39; United States, President of the United States) **Path 3 (temporal conflict evidence for an alternative role):** (P39 / P580 / P582; Barack Obama, ?, 2009, 2017) → (P39; Barack Obama, Senator of the United States) → (P580; Senator of the United States, 2005) → (P582; Senator of the United States, 2008)