CoG: Controllable Graph Reasoning via Relational Blueprints and Failure-Aware Refinement over Knowledge Graphs

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities but often grapple with reliability challenges like hallucinations. While Knowledge Graphs (KGs) offer explicit grounding, existing paradigms of KG-augmented LLMs typically exhibit **cognitive rigidity**—applying homogeneous search strategies that render them vulnerable to instability under neighborhood noise and structural misalignment leading to reasoning stagnation. To address these challenges, we propose **CoG**, a training-free framework inspired by **Dual-Process Theory** that mimics the interplay between intuition and deliberation. First, functioning as the fast, intuitive process, the Relational Blueprint Guidance module leverages relational blueprints as interpretable soft structural constraints to rapidly stabilize the search direction against noise. Second, functioning as the prudent, analytical process, the Failure-Aware Refinement module intervenes upon encountering reasoning impasses. It triggers evidence-conditioned reflection and executes controlled backtracking to overcome reasoning stagnation. Experimental results on three benchmarks demonstrate that CoG significantly outperforms state-of-the-art approaches in both accuracy and efficiency.

1 Introduction

Large Language Models (LLMs) (Brown et al., 2020a; Ouyang et al., 2022; Achiam et al., 2023; Dubey et al., 2024; Guo et al., 2025) have demonstrated strong generalization across natural language tasks (Bang et al., 2023; Zhao et al., 2023; Huang and Chang, 2023; Qiao et al., 2023). However, in knowledge-intensive and multi-hop reasoning scenarios, they face substantial reliability challenges, including hallucinations and inconsistent evidence chains (Hu et al., 2023; Wang et al., 2023a; Huang et al., 2024). These issues are exacerbated by reliance on parametric knowledge, which is difficult to update and lacks verifiability. When external evidence is incomplete, models often fall back on language priors, producing answers that are fluent yet weakly grounded. Consequently, integrating knowledge graphs (KGs)—which offer structured, retrievable, and verifiable facts—has become a critical pathway to ground complex reasoning (Pan et al., 2024).

Among KG-augmented LLMs approaches, the **LLM-driven graph agent paradigm** (Sun et al., 2024; Chen et al., 2024; Ma et al., 2025) has been widely adopted for its flexibility. Such methods typically execute a plan–retrieve–generate loop to iteratively extend evidence chains (Sun et al., 2024; Chen et al., 2024; Luo et al., 2024). Despite their utility, these approaches often exhibit instability in complex settings, with performance fluctuating heavily under neighborhood noise. We attribute this instability not merely to knowledge deficiency (Ji et al., 2023), but to a lack of adaptive strategy regulation—a limitation we term **cognitive rigidity**. In practice, many existing systems (Sun et al., 2024; Chen et al., 2024) apply homogeneous search strategies regardless of task uncertainty, leading to reasoning trajectories that oscillate or deviate. Specifically, cognitive rigidity manifests in two reinforcing challenges (Figure 1):

(1) Error Cascading from Indiscriminate Exploration. Indiscriminate exploration fails to distinguish high-value signals from noise. A minor early selection error (e.g., selecting *contains* instead of *adjoins*) exposes the agent to substantially larger candidate sets. This noise dominates the reasoning branch, causing irreversible trajectory deviation (Figure 1(a)) that is difficult to recover from.

(2) Structural Misalignment from Myopic Decisions. Relying heavily on local semantic matching often neglects global logical constraints. This myopia traps models in local optima: selecting relations that appear relevant but utilize the wrong

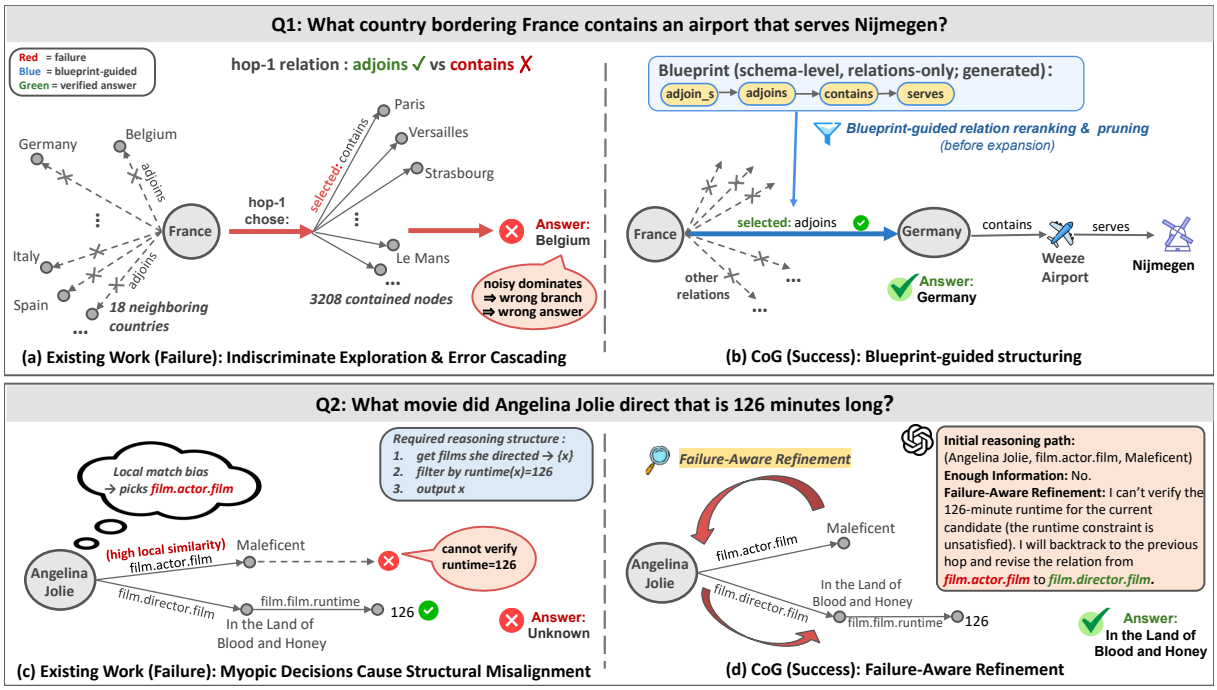

Figure 1: Illustration of cognitive rigidity in existing works and how CoG addresses it: (I) **Error cascading** from indiscriminate exploration (top), mitigated by **Relational Blueprints** for blueprint-guided relation reranking and pruning (a vs. b); (II) **structural misalignment** from myopic decisions (bottom), corrected by **Failure-Aware Refinement** via backtracking and controlled fallback (c vs. d).

structure (e.g., prioritizing *actor* over *director* in Figure 1(c)). Consequently, retrieved candidates fail to satisfy downstream constraints (e.g., runtime checks), forcing premature termination. This highlights that local semantic relevance does not guarantee cross-hop structural consistency.

To address these challenges, we propose **CoG**, a novel training-free framework for **controllable reasoning on KGs**. Inspired by **Dual-Process Theory** (Kahneman, 2011), CoG emulates the cognitive interplay between intuitive pattern recognition and deliberate analysis: **First, Relational Blueprint Guidance (System 1):** Acting as the *fast, intuitive* process driven by experience and structural patterns, this module is responsible for rapidly determining the search direction, particularly when structures are similar or information is sufficient. It leverages relational blueprints as heuristic priors to efficiently filter noise without heavy reasoning overhead (Figure 1(b)). **Second, Failure-Aware Refinement (System 2):** Acting as the *prudent, analytical* process, this module intervenes upon detecting failure signals, such as insufficient evidence or unverifiable constraints. It explicitly diagnoses anomalies (e.g., stagnation due to KG incompleteness (Xu et al., 2024)) and executes corrective measures through controlled backtracking to overcome reasoning dead-ends (Figure 1(d)).

Overall, CoG harmonizes intuitive guidance with analytical refinement to improve robustness while preserving verifiability. Our main contributions are:

- We propose **CoG**, a training-free framework that synergizes blueprint-guided planning (for stability) and failure-aware refinement (for robustness) to mitigate error cascading and structural mismatch.
- We introduce a relational blueprint mechanism, injecting interpretable soft structural priors into graph exploration to balance efficiency and controllability.
- We evaluate CoG on multiple KGQA datasets. Results on both **open-source and closed-source LLMs** validate that our framework outperforms state-of-the-art baselines and generalizes robustly across diverse backbones.

2 Preliminaries

Knowledge Graph (KG). Let $\mathcal{G} = (\mathcal{E}, \mathcal{R})$ be a knowledge graph, where $\mathcal{E}$ and $\mathcal{R}$ denote entities and relations, respectively. It consists of factual triplets (e, r, e'), where $e, e' \in \mathcal{E}$ and $r \in \mathcal{R}$.

Relation & Reasoning Paths. A **relation path** $z = (r_1, \ldots, r_L)$ captures an abstract query pattern (e.g., *born_in* $\rightarrow$ *capital_of*) independent of

concrete entities. A **reasoning path** p_z instantiates z in $\mathcal{G}$ as $e_0 \xrightarrow{r_1} e_1 \dots \xrightarrow{r_L} e_L$, where each step (e_{i-1}, r_i, e_i) is a valid triplet in $\mathcal{G}$.

Problem Statement (KGQA). Given a question Q and a KG $\mathcal{G}$, the goal is to identify the answer set $\mathcal{A} \subseteq \mathcal{E}$. Following prior agent-based paradigms (Sun et al., 2024; Chen et al., 2024), we adopt an iterative retrieve-and-reason approach: starting from topic entities $\mathcal{E}_q$ extracted from Q, the agent expands the subgraph step-by-step to construct a reasoning path pointing to the answer.

3 Methodology

In this section, we introduce CoG, a training-free framework that couples blueprint-guided planning with failure-aware correction. As shown in Figure 2, CoG instantiates Dual-Process Theory with three components. Specifically, (i) Offline blueprint construction abstracts relational sequences from training paths into a searchable template library. (ii) System 1 (Online blueprint-guided exploration) retrieves and adapts a query-specific blueprint as a soft structural constraint for candidate-relation reranking and pruning. (iii) System 2 (Failure-aware refinement) detects failure signals (e.g., search stagnation) and triggers evidence-conditioned reflection with targeted backtracking to revise earlier decisions and recover a verifiable evidence chain. Overall, CoG promotes globally consistent exploration beyond purely local semantic matching.

3.1 Offline Relational Blueprint Construction

In the offline stage, we distill structural priors from training data by abstracting entity-specific reasoning paths into reusable relation-only patterns. This one-time preprocessing yields a *blueprint template library* that provides stable structural cues for online reasoning with negligible computational overhead.

Extraction and Abstraction of Relation Paths. Given a training instance $(q, p) \in \mathcal{D}_{\text{train}}$, where q is the question and p is its reasoning-path representation (e.g., an executable SPARQL query), we aim to extract the underlying logical structure. Since p is typically instantiated with concrete entities, it is not directly transferable. We therefore adopt a deterministic, rule-based de-instantiation strategy: we parse p and apply pattern filters (e.g., regular expressions) to remove entity identifiers (e.g., Freebase IDs) and other non-structural elements, retaining only relation predicates. The remaining relations, ordered by occurrence, form a relational blueprint template:

$$\mathcal{S}(q) = \langle r_1, r_2, \dots, r_L \rangle, \quad r_j \in \mathcal{R}, \qquad (1)$$

where $\mathcal{R}$ is the set of KG relations, r_j is the j-th predicate, and L is the path length. This abstraction converts grounded facts into reusable structural priors that transfer across questions.

Relational Blueprint Template Library Construction. Multiple training questions may share the same logical structure. To ensure a compact yet representative library, we deduplicate templates using their string representations as unique keys. For each unique $\mathcal{S}$, we aggregate its associated questions and select a *semantic anchor* q^* to facilitate online matching. We employ a simple yet effective heuristic by selecting the longest question as the anchor to preserve maximal contextual semantics:

$$q^* = \arg\max_{q \in \{q' | \text{map}(q') = \mathcal{S}\}} \text{len}(q), \qquad (2)$$

where $\text{map}(\cdot)$ is the abstraction mapping and $\text{len}(\cdot)$ denotes text length. The final library is defined as $\mathcal{B} = \{(q_i, \mathcal{S}_i)\}_{i=1}^{N}$.

Semantic Indexing. For efficient retrieval, we encode each anchor question q_i with a pretrained sentence encoder (e.g., SentenceTransformer) to obtain $\mathbf{v}_{q_i}$, and build a vector index over $\{\mathbf{v}_{q_i}\}_{i=1}^{N}$ for nearest-neighbor search at test time. This offline procedure involves no task-specific training: we do not fine-tune the encoder or perform gradient updates, relying only on rule-based extraction and encoder forward passes.

3.2 Online Blueprint-Guided KG Exploration

The online stage incrementally constructs a traceable and verifiable evidence chain on the KG, while keeping reasoning stable and controllable under neighborhood noise and varying candidate scales. We instantiate CoG as a planning agent that performs dynamic navigation on the KG. For a question Q, following prior work (Sun et al., 2024; Chen et al., 2024; Jiang et al., 2025) the agent follows an iterative loop: at step t, it identifies a candidate relation set $\mathcal{R}_{\text{cand}}$ reachable from the entity frontier E_{t-1}, executes a selection to expand the frontier to E_t, and verifies new evidence against constraints. Verified triples and intermediate conclusions are stored in a working memory $\mathcal{M}$, providing context for subsequent decisions. On top of this loop, CoG retrieves and adapts a

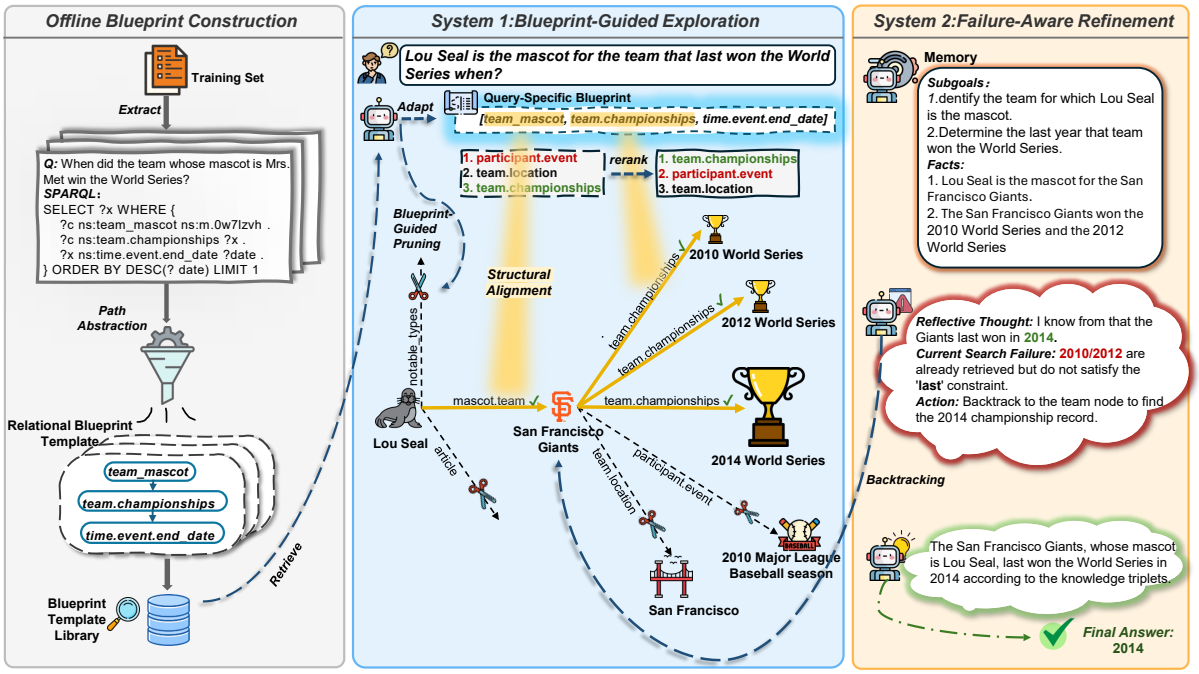

Figure 2: **Overview of the CoG framework.** CoG instantiates Dual-Process Theory as a cooperative reasoning loop. **Left:** Offline blueprint construction abstracts relational sequences from training paths into a searchable template library. **Middle (System 1):** Online blueprint-guided exploration adapts query-specific blueprints as soft structural constraints to rerank and prune candidate relations. **Right (System 2):** Failure-aware refinement monitors the reasoning process; upon failure signals (e.g., search stagnation), it performs evidence-conditioned reflection and targeted backtracking to recover a verifiable answer (e.g., enforcing the temporal constraint "last won").

query-specific relational blueprint S_{BP} and uses it as an interpretable soft structural constraint to guide candidate-relation reranking and pruning at each step, mitigating structural mismatch and error cascading from purely local semantic matching.

3.2.1 Initialization and Structural Planning

Initialization and Task Decomposition. Given Q, we perform entity linking to identify topic entities and obtain the initial frontier E_0. A working memory $\mathcal{M}$ is maintained to store verified evidence triples, historical decisions, and constraint states, supporting conditional decision-making and failure correction. We then decompose Q into an ordered subgoal sequence $\mathcal{O} = [o_1, \ldots, o_T]$, where each o_t specifies the focus at step t to mitigate context drift. While subgoals guide local actions, they do not define the global relational structure, which can lead to the structural misalignment and myopic decisions that CoG aims to resolve.

Entity-masked blueprint retrieval and adaptation. To provide global guidance, we generate a query-specific blueprint S_{BP}. We first mask topic entities in Q to obtain $Q_m = \text{Mask}(Q, E_0)$, shifting retrieval toward compositional structures and predicate patterns. We encode Q_m with a pre-trained sentence encoder $f(\cdot)$ and retrieve Top-K nearest exemplars from an offline semantic index, where each exemplar is an *anchor-question–relational-blueprint-template* pair. We adopt a hybrid *copy–adapt* strategy: if the top similarity exceeds τ_{copy}, we copy the retrieved Top-1 template; otherwise, we feed the most similar exemplars (e.g., Top-2) together with Q into an LLM to generate or lightly rewrite a blueprint under exemplar structural constraints. Crucially, this adaptation mechanism ensures that CoG is not confined to the training distribution. By treating retrieved blueprints as flexible structural exemplars rather than rigid rules, the LLM can synthesize novel reasoning plans for unseen query topologies, effectively bridging the gap between historical priors and new scenarios. Specific implementation details are provided in Appendix **??**. The resulting blueprint is

$$S_{\text{BP}} = \langle r_1^{\text{BP}}, \ldots, r_L^{\text{BP}} \rangle, \tag{3}$$

where r_j^{BP} is the j-th relation slot and L is the blueprint length. Importantly, S_{BP} is not a hard executable plan but an interpretable soft structural prior: it sketches the approximate relation types to follow and continuously constrains candidate

selection. L provides a depth prior (not a hard limit); the actual number of steps is determined by constraint satisfaction and termination criteria.

3.2.2 Blueprint-Guided KG Exploration

Under subgoals and the blueprint, CoG performs stepwise exploration by expanding the frontier, verifying evidence, and updating memory under the blueprint soft constraint.

Candidate relation collection and blueprint-guided reranking. From the current frontier E_{t-1}, we collect all reachable relations to form the candidate set $\mathcal{R}_{\text{cand}}$. To control scale and suppress noise propagation, we apply lightweight rule-based filtering (e.g., removing obviously uninformative generic relations). In noisy neighborhoods, $\mathcal{R}_{\text{cand}}$ may still contain many distractors, and relying only on local semantic matching can cause early mis-selections and error cascading. To stabilize decisions, CoG injects the blueprint S_{BP} into the candidate-relation layer and reranks candidates by three complementary signals. We first define a slot-alignment index

$$\pi(t) = \arg\max_{j\in[1,L]} \text{sim}\big(h(o_t), h(r_j^{\text{BP}})\big), \quad (4)$$

where $h(\cdot)$ is a text encoder and $\text{sim}(\cdot,\cdot)$ is cosine similarity. To prevent structural drift caused by out-of-order slot alignment in multi-hop reasoning, we enforce a monotone alignment constraint in implementation:

- **Initialization:** $\pi(0) = 1$.
- **Monotone update:** restrict $j \in [\pi(t-1), L]$ so $\pi(t)$ is non-decreasing.
- **Clamping:** when steps exceed L, clamp $\pi(t) = L$.

Based on $\pi(t)$, we define three complementary scoring signals:

$$\phi_{\text{loc}}(o_t, r) = \text{sim}\big(h(o_t), h(r)\big), \quad (5a)$$

$$\phi_{\text{step}}(r, r^{\text{BP}}_{\pi(t)}) = \text{sim}\big(h(r), h(r^{\text{BP}}_{\pi(t)})\big), \quad (5b)$$

$$\phi_{\text{glob}}(S_{\text{BP}}, r) = \max_{j\in[1,L]} \text{sim}\big(h(r), h(r_j^{\text{BP}})\big), \quad (5c)$$

where ϕ_{loc} (*local relevance*) captures subgoal–relation relevance, ϕ_{step} (*step-wise alignment*) measures alignment to the current blueprint slot, and ϕ_{glob} (*global compatibility*) evaluates compatibility with the overall blueprint, helping mitigate structural drift over long-horizon reasoning. For each $r \in \mathcal{R}_{\text{cand}}$, we compute a fused score

$$\begin{aligned}\text{Score}(r) = \lambda_{\text{loc}}\phi_{\text{loc}}(o_t, r) \\ + \lambda_{\text{step}}\phi_{\text{step}}(r, r^{\text{BP}}_{\pi(t)}) \\ + \lambda_{\text{glob}}\phi_{\text{glob}}(S_{\text{BP}}, r),\end{aligned} \quad (6)$$

where $\lambda_{\text{loc}}, \lambda_{\text{step}}, \lambda_{\text{glob}} \geq 0$ and $\lambda_{\text{loc}} + \lambda_{\text{step}} + \lambda_{\text{glob}} = 1$. we set weights λ_{loc}=0.6, λ_{step}=0.25, and λ_{glob}=0.15 in all experiments; additional sensitivity details are deferred to Appendix E.3. We then rerank $\mathcal{R}_{\text{cand}}$ by $\text{Score}(\cdot)$ and retain the top-scoring relations to form a compact, structure-aligned shortlist $\tilde{\mathcal{R}}_{\text{cand}}$.

Blueprint-guarded pruning. We further refine $\tilde{\mathcal{R}}_{\text{cand}}$ via an LLM conditioned on $(Q, o_t, \mathcal{M})$ and the shortlist. To reduce the risk of missing structurally correct relations, we enforce a Structure-Consistency Safeguard: the final candidate set is the **union** of the LLM-selected relations and the top candidate by step-wise alignment ϕ_{step}. This dual-source selection balances semantic nuances and structural consistency, mitigating single-view bias.

State update and answer generation. After selecting r_t, CoG expands the KG to obtain the next frontier E_t, filters reached entities using subgoal constraints and the working memory $\mathcal{M}$ (via an LLM), and writes the chosen relation, key evidence triples, and verified intermediate conclusions into $\mathcal{M}$ for traceability and termination. CoG then performs an LLM-based sufficiency check conditioned on the current subgoal state o_t and verified evidence in $\mathcal{M}$. If sufficient, the LLM synthesizes the verified traces and subgoal states to produce the final answer. If insufficient, the failure stems from either missing evidence (addressable by further expansion) or an early wrong decision that derailed the trajectory; in the latter case, CoG invokes Failure-Aware Refinement to backtrack and revise suspicious transitions, recovering a verifiable evidence chain while maintaining controllability.

3.3 Failure-Aware Refinement

While blueprint guidance provides strong structural priors, KG incompleteness and noise can still impede exploration. To counteract this, CoG implements Failure-Aware Refinement. Upon detecting failure signals (e.g., stagnation or insufficient evidence), CoG switches from forward exploration to a controlled correction mode. This process prioritizes minimal interventions on high-risk transitions, ensuring the recovery of verifiability without discarding valid sub-paths.

Diagnosis and Targeted Re-routing. Upon triggering a failure signal, CoG invokes an LLM for evidence-conditioned reflection. Guided by the working memory $\mathcal{M}$, the the LLM reviews the current trajectory $\mathcal{T} = [e_0, r_1, e_1, \ldots]$ alongside

compact summaries of pruned branches to pinpoint the decision point t_{err} responsible for the deviation (e.g., a step biased by local semantics). Subsequently, the agent executes targeted backtracking to re-route the search: it reverts the frontier to the state preceding t_{err}, recalls structurally relevant candidates that were prematurely pruned, and resumes expansion. This re-routing mechanism rectifies myopic errors while enforcing global constraints, allowing CoG to reconstruct a verifiable evidence chain.

Grounded Inference. In extreme cases, missing edges can render verifiable evidence unreachable, such that even re-routing fails to restore a checkable evidence chain. As a fallback, CoG aggregates verified path segments and unmet constraints into a concise summary, prompting the LLM to synthesize a final answer conditioned strictly on this valid context. Unlike free-form generation, this inference is explicitly grounded in the verified semantic space, effectively mitigating parametric hallucination while preventing premature termination.

4 Experiments

4.1 Experiment Setup

Datasets & Evaluation Metrics. To verify CoG's effectiveness in complex reasoning over knowledge graphs, **aligning with the standard evaluation protocols of prior state-of-the-art methods** (Sun et al., 2024; Chen et al., 2024), we evaluate on three representative multi-hop KGQA benchmarks: **CWQ** (Talmor and Berant, 2018), **WebQSP** (Yih et al., 2016), and **GrailQA** (Gu et al., 2021). These datasets are grounded on Freebase (Bollacker et al., 2008), which contains approximately 88 million entities, 20K relations, and 126 million triples, serving as one of the most comprehensive knowledge bases for standardized KGQA evaluation. Following previous studies (Sun et al., 2024; Chen et al., 2024; Jiang et al., 2023a), we use Exact Match accuracy (**Hits@1**) as the primary metric, i.e., the percentage of questions whose predicted answer exactly matches the ground-truth answer, thus emphasizing exact answers over partially correct ones.

Compared Methods. We compare CoG with strong existing LLM-based baselines from two groups: LLM-only methods and KG-augmented LLM methods, covering both fine-tuned and prompting methods. Detailed descriptions of the compared methods are deferred to Appendix D.

Implementations. We construct dataset-specific blueprint libraries by parsing gold SPARQL queries from the training splits of WebQSP, GrailQA, and CWQ. Semantic anchors are encoded into dense vectors using a pre-trained Sentence Transformer (Sanh et al., 2019) for retrieval. Detailed implementation settings and template library statistics are deferred to the Appendix E.2.1.

Method	CWQ	WebQSP
LLM-Only (GPT-3.5)		
IO Prompt (Brown et al., 2020b)	37.6	63.3
CoT (Wei et al., 2022)	38.8	62.2
SC (Wang et al., 2023b)	45.4	61.1
Fine-Tuned KG-Augmented LLM		
RE-KBQA (Cao et al., 2023)	50.3	74.6
UniKGQA (Jiang et al., 2023b)	51.2	79.1
RoG (Luo et al., 2024)	62.6	85.7
DECAF (Yu et al., 2022)	70.4	82.1
KG-Agent (Jiang et al., 2025)	72.2	83.3
Prompting KG-Augmented LLM w/ Qwen2.5-7B		
ToG (Sun et al., 2024)	42.5	56.0
PoG (Chen et al., 2024)	46.0	58.5
CoG (Ours)	**54.0**	**74.5**
Prompting KG-Augmented LLM w/ GPT-3.5 or others		
KD-CoT (Wang et al., 2023a)	50.5	73.7
KB-BINDER (Li et al., 2023)	–	74.4
StructGPT (Jiang et al., 2023a)	54.3	72.6
ToG (Sun et al., 2024)	57.1	76.2
PoG (Chen et al., 2024)	63.2	82.0
CoG (Ours)	**66.9**	**86.8**
Prompting KG-Augmented LLM w/ GPT-4		
ToG (Sun et al., 2024)	67.6	82.6
PoG (Chen et al., 2024)	75.0	87.3
CoG (Ours)	**77.8**	**89.7**

Table 1: Performance comparison of different methods on CWQ and WebQSP.

4.2 Main Results

Table 1 and Table 2 present the comparisons between CoG and representative state-of-the-art (SOTA) baselines on WebQSP, CWQ, and GrailQA. Overall, CoG delivers consistently strong performance across all evaluation settings. First, compared with prompting-based KG-augmented LLM baselines, CoG shows clear advantages: under both GPT-3.5 and GPT-4 backbones, CoG consistently outperforms the strongest baseline *Planning-on-Graph* (PoG). While PoG introduces adaptive planning, its online search is still largely driven by local exploration signals, making it more susceptible to structural ambiguity in complex neighborhoods. CoG mitigates this limitation through the synergy of relational blueprint guidance and failure-aware refinement, which jointly encourages global

Method	GrailQA			
	Overall	I.I.D.	Compositional	Zero-shot
LLM-Only				
IO Prompt (Brown et al., 2020b)	29.4	–	–	–
CoT (Wei et al., 2022)	28.1	–	–	–
SC (Wang et al., 2023b)	29.6	–	–	–
Fine-Tuned KG-Augmented LLM				
RnG-KBQA (Ye et al., 2022)	68.8	86.2	63.8	63.0
TIARA (Shu et al., 2022)	73.0	87.8	69.2	68.0
FC-KBQA (Zhang et al., 2023)	73.2	88.5	70.0	67.6
Pangu (Gu et al., 2023)	75.4	84.4	74.6	71.6
FlexKBQA (Li et al., 2024)	62.8	71.3	59.1	60.6
GAIN (Shu and Yu, 2024)	76.3	88.5	73.7	71.8
KG-Agent (Jiang et al., 2025)	86.1	92.0	80.0	86.3
Prompting KG-Augmented LLM w/ Qwen2.5-7B				
ToG (Sun et al., 2024)	62.6	60.8	47.0	68.9
PoG (Chen et al., 2024)	68.9	67.9	51.5	75.4
CoG (Ours)	**72.0**	**72.5**	**57.6**	**76.9**
Prompting KG-Augmented LLM w/ GPT-3.5 or others				
KB-BINDER (Li et al., 2023)	53.2	72.5	51.8	45.0
ToG (Sun et al., 2024)	68.7	70.1	56.1	72.7
PoG (Chen et al., 2024)	76.5	76.3	62.1	81.7
CoG (Ours)	**79.2**	**80.4**	**65.2**	**83.6**
Prompting KG-Augmented LLM w/ GPT-4				
ToG (Sun et al., 2024)	81.4	79.4	67.3	86.5
PoG (Chen et al., 2024)	84.7	87.9	69.7	88.6
CoG (Ours)	**86.4**	**88.3**	**76.3**	**89.1**

Table 2: Performance comparison of different methods on GrailQA.

structural alignment and enables controlled correction under failure cases, leading to higher accuracy. Second, although CoG is a training-free prompting framework, it remains highly competitive against fine-tuned KG-augmented LLM baselines. On CWQ and WebQSP, CoG (with GPT-4) surpasses all included fine-tuned baselines; on GrailQA, CoG still outperforms all fine-tuned methods even when using the weaker GPT-3.5 backbone. A potential concern with blueprint-guided exploration is generalization to unseen structures. However, results on the GrailQA Zero-shot split (Table 2) using GPT-3.5 show that CoG (83.6%) achieves substantial gains over baselines, significantly outperforming both the exploration-based ToG (72.7%) and the adaptive planning method PoG (81.7%). This empirically confirms that our approach leverages training data as abstract logical priors to aid generalization, rather than merely memorizing paths, thereby offering superior robustness even with less capable backbone models. Moreover, these findings strongly support the effectiveness of combining structural guidance with failure awareness for complex reasoning, and further suggest that explicit structural priors coupled with dynamic self-correction can generalize more robustly to unseen query structures and compositional patterns than relying solely on implicit

Method	CWQ	WebQSP	GrailQA
CoG (Ours)	**66.9**	**86.8**	**79.2**
Component Removal			
w/o Blueprint Adaptation	62.4	83.5	77.5
w/o Blueprint-guided Reranking	63.5	84.0	76.8
w/o Failure-Aware Refinement	58.5	79.9	75.3
Reranking Variants			
Local relevance only	64.6	84.4	76.2
w/o Global compatibility	65.8	85.4	77.2
w/o Step-wise alignment	65.9	85.7	77.6

Table 3: Ablation study of core components and blueprint-guided reranking signals.

parameter updates. Furthermore, compared with LLM-only baselines, CoG benefits substantially from incorporating structured evidence from external knowledge graphs, underscoring the value of KG grounding for complex queries that are difficult to handle with parametric knowledge alone. To assess generalization across model scales, we further evaluate CoG with a smaller backbone LLM (e.g., Qwen2.5-7B (Team, 2024)); even with substantially reduced model capacity, CoG still consistently surpasses baseline agents, indicating that the gains primarily stem from methodological advances rather than merely scaling up the underlying LLM. To further intuitively demonstrate how CoG navigates complex reasoning paths and corrects errors, we provide detailed case studies in Appendix J.

4.3 Ablation Study

To evaluate the effectiveness of individual components and adaptive exploration in CoG, we conducted ablation studies on CWQ, WebQSP, and GrailQA, progressively removing key components. The results are shown in Table 3. Specifically, *w/o Blueprint Adaptation* refers to using retrieved relations directly from the database, without allowing the LLM to modify them to better align with the query. *w/o Blueprint-guided Reranking* refers to removing the step where the retrieved blueprint serves as an interpretable structural prior, guiding the ranking of relations based on its structural constraints. *w/o Failure-Aware Refinement* refers to the removal of the failure recovery mechanism, which corrects reasoning when search stalls or evidence is insufficient. Empirical results indicate that removing *Failure-Aware Refinement* causes the most significant performance drop, underscoring its critical role in resolving search impasses. Moreover, removing either *Blueprint Adaptation* or *Blueprint-guided Reranking* consistently degrades

Data	Method	Calls	Input	Output	Total	H@1
CWQ	ToG	22.6	8,182.9	1,486.4	9,669.4	57.1
	PoG	13.3	7,803.0	**353.2**	8,156.2	63.2
	CoG	**11.7**	**6,589.0**	486.8	**7,075.8**	**66.9**
WebQSP	ToG	15.9	6,031.2	987.7	7,018.9	76.2
	PoG	9.0	5,234.8	282.9	5,517.7	82.0
	CoG	**8.3**	**4,693.6**	**206.0**	**4,899.6**	**86.8**
GrailQA	ToG	11.1	4,066.0	774.6	4,840.6	68.7
	PoG	6.5	3,372.8	202.8	3,575.6	76.5
	CoG	**5.5**	**3,122.0**	**166.1**	**3,288.1**	**79.2**

Table 4: Efficiency comparison. Metrics include average LLM calls and token usage per query.

performance, indicating that CoG benefits from both adapting the retrieved relational templates to the specific query and leveraging the blueprint as an interpretable guide for relation selection. Notably, while the blueprint offers implicit global context (e.g., depth cues) even without reranking, its efficacy is substantially diminished when decoupled from the candidate ranking process. Regarding reranking variants, relying solely on *local relevance* yields suboptimal gains. Incorporating *Stepwise Alignment* mitigates myopic decision-making, while *Global Compatibility* ensures long-term consistency. Peak performance is achieved by integrating all three signals, confirming the synergy of our fused ranking design.

4.4 Efficiency Analysis

We evaluate the efficiency of different methods in terms of LLM calls and token usage, as shown in Table 4. CoG achieves a superior accuracy-cost trade-off across all datasets. Unlike traditional expansion-based methods, which rely on exhaustive searches over neighboring nodes, CoG uses the relational blueprint to constrain the search space. This reduces irrelevant reasoning paths by pruning them in advance, eliminating the need for exhaustive exploration and avoiding redundant intermediate queries. As a result, CoG significantly reduces computational overhead, setting a new performance benchmark while minimizing costs, making it highly cost-effective and practical for real-world deployment.

5 Related Work

LLM Reasoning. LLMs have transitioned from heuristic prompting to structured reasoning frameworks (Brown et al., 2020a; Wei et al., 2022; Zhou et al., 2022; Zhao et al., 2023; Huang and Chang, 2023; Qiao et al., 2023). Beyond Chain-of-Thought (CoT) (Wei et al., 2022), recent works utilize nonlinear topologies (Yao et al., 2023; Besta et al., 2024; Li and Qiu, 2023; Ning et al., 2023) to navigate complex spaces (Yao et al., 2023; Besta et al., 2024). However, purely parametric reasoning remains vulnerable to hallucinations and logical inconsistencies (Ji et al., 2023; Hu et al., 2023). While ensemble strategies like majority voting (Wang et al., 2023b) bolster stability, they cannot rectify knowledge staleness or decision opacity (Huang and Chang, 2023; Pan et al., 2024).

KG-Augmented LLM. Knowledge Graphs (KGs) provide verifiable grounding (Ji et al., 2023; Pan et al., 2024) via implicit integration (Jiang et al., 2023b; Luo et al., 2024) or explicit retrieval (Jiang et al., 2023a; Li et al., 2023). Yet, implicit methods struggle with schema evolution, while explicit retrieval often decouples search from reasoning (Jiang et al., 2023a; Li et al., 2023). Agent-based paradigms (Sun et al., 2024; Chen et al., 2024) enable interactive navigation but suffer from open-loop rigidity and irreversible semantic drifts (Chen et al., 2024). Similarly, specialized navigation or exemplar-based priors (Luo et al., 2024; Zhang et al., 2024; He et al., 2024; Xu et al., 2025) apply guidance in a static, open-loop manner, lacking robustness against structural misalignment. In contrast, CoG introduces a closed-loop paradigm. While training-free methods like ToG (Sun et al., 2024) and GoG (Xu et al., 2024) suffer from inefficient indiscriminate exploration, CoG utilizes training data as a reference library—a "compass" for navigation. This design strikes a superior balance: it avoids parameter fine-tuning costs while preventing the aimless wandering of zero-shot exploration, without sacrificing the flexibility to handle novel cases via adaptation.

6 Conclusion

In this paper, we present CoG, a training-free framework for controllable reasoning over knowledge graphs. CoG introduces two key innovations: relational blueprint-guided exploration and failure-aware refinement. These mechanisms mitigate critical challenges such as error cascading and structural misalignment in complex multi-hop reasoning tasks. Extensive experiments on WebQSP, CWQ, and GrailQA demonstrate that CoG outperforms existing state-of-the-art methods in both performance and efficiency, making it a robust and scalable solution for knowledge graph question answering.

Limitations

Despite the effectiveness of CoG in enhancing multi-hop reasoning, several limitations remain. First, its performance is constrained by the coverage and accuracy of the underlying knowledge graph. While failure-aware refinement can help mitigate gaps in evidence through grounded synthesis, it cannot fully compensate for the absence of key relational paths in the graph. Second, the quality and coverage of the pre-constructed relational blueprint library directly impact the effectiveness of structural guidance. In specialized or niche domains where such templates are scarce, the model's ability to retrieve and adapt appropriate structural constraints may be hindered. Moreover, while CoG achieves a favorable accuracy-cost trade-off compared to unstructured search methods, the process of resolving complex cascading failures through iterative backtracking and reflection can introduce additional computational latency. Finally, since blueprints are currently generated in a static offline manner, exploring the dynamic online evolution and adaptive refinement of these templates remains a promising avenue for future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *Preprint*, arXiv:2302.04023.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 17682–17690.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020b. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yong Cao, Xianzhi Li, Huiwen Liu, Wen Dai, Shuai Chen, Bin Wang, Min Chen, and Daniel Hershcovich. 2023. Pay more attention to relation exploration for knowledge base question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2119–2136.

Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. 2024. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. In *Proceedings of the 38th Conference on Neural Information Processing Systems*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yu Gu, Xiang Deng, and Yu Su. 2023. Don't generate, discriminate: A proposal for grounding language models to real-world environments. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 4928–4949.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the web conference 2021*, pages 3477–3488.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.

Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S Yu, and Zhijiang Guo. 2023. Do large language models know about facts? *arXiv preprint arXiv:2310.05177*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey.

In *Findings of the association for computational linguistics: ACL 2023*, pages 1049–1065.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

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.

Jinhao Jiang, Kun Zhou, KeMing Ye, Xin Zhao, and Ji-Rong Wen. 2023a. StructGPT: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251.

Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang Song, Chen Zhu, Hengshu Zhu, and Ji-Rong Wen. 2025. Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9505–9523.

Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023b. Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations*.

Daniel Kahneman. 2011. *Thinking, fast and slow*. macmillan.

Tianle Li, Xueguang Ma, Alex Zhuang, Yu Gu, Yu Su, and Wenhu Chen. 2023. Few-shot in-context learning on knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6966–6980.

Xiaonan Li and Xipeng Qiu. 2023. Mot: Memory-of-thought enables chatgpt to self-improve. *arXiv preprint arXiv:2305.05181*.

Zhenyu Li, Sunqi Fan, Yu Gu, Xiuxing Li, Zhichao Duan, Bowen Dong, Ning Liu, and Jianyong Wang. 2024. Flexkbqa: A flexible llm-powered framework for few-shot knowledge base question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 18608–18616.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations*.

Jie Ma, Zhitao Gao, Qi Chai, Wangchun Sun, Pinghui Wang, Hongbin Pei, Jing Tao, Lingyun Song, Jun Liu, Chen Zhang, and 1 others. 2025. Debate on graph: a flexible and reliable reasoning framework for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24768–24776.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-thought: Large language models can do parallel decoding. *Proceedings ENLSP-III*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 5368–5393.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Yiheng Shu and Zhiwei Yu. 2024. Distribution shifts are bottlenecks: Extensive evaluation for grounding language models to knowledge bases. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 71–88.

Yiheng Shu, Zhiwei Yu, Yuhan Li, Börje Karlsson, Tingting Ma, Yuzhong Qu, and Chin-Yew Lin. 2022. TIARA: Multi-grained retrieval for robust question answering over large knowledge base. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8108–8121.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651.

Qwen Team. 2024. Qwen2. 5: A party of foundation models, september 2024. *URL https://qwenlm. github. io/blog/qwen2*, 5(4).

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023a. Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jingao Xu, Shuoyoucheng Ma, Xin Song, Rong Jiang, Hongkui Tu, and Bin Zhou. 2025. Exemplar-guided planing: Enhanced llm agent for kgqa. *Preprint*, arXiv:2510.15283.

Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. 2024. Generate-on-graph: Treat LLM as both agent and KG for incomplete knowledge graph question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18410–18430.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. 2022. Rng-kbqa: Generation augmented iterative ranking for knowledge base question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6032–6043.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206.

Donghan Yu, Sheng Zhang, Patrick Ng, Henghui Zhu, Alexander Hanbo Li, Jun Wang, Yiqun Hu, William Yang Wang, Zhiguo Wang, and Bing Xiang. 2022. Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. In *The Eleventh International Conference on Learning Representations*.

Lingxi Zhang, Jing Zhang, Yanling Wang, Shulin Cao, Xinmei Huang, Cuiping Li, Hong Chen, and Juanzi Li. 2023. FC-KBQA: A fine-to-coarse composition framework for knowledge base question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1006–1020.

Qinggang Zhang, Junnan Dong, Hao Chen, Daochen Zha, Zailiang Yu, and Xiao Huang. 2024. Knowgpt: Knowledge graph based prompting for large language models. *Advances in Neural Information Processing Systems*, 37:6052–6080.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Prompt Templates for LLM Agents

In this section, we provide the prompt templates used in CoG for LLM agents. To ensure proper parsing of the LLM output, we require the LLM to provide answers using specific data structures, such as lists and JSON. This structured output facilitates easier processing and integration with the system. Additionally, we explicitly instruct the LLM not to include any irrelevant information in its responses. Some of the prompt templates are based on the design of PoG (Plan-on-Graph) (Chen et al., 2024) prompts, and have been adjusted and optimized to better suit the reasoning process in CoG.

A.1 Task Decomposition

> Please break down the process of answering the question into as few subgoals as possible based on semantic analysis.
>
> *In-Context Few-Shot*
>
> Now you need to directly output subgoals of the following question in list format without other information or notes.
> Q: {}

A.2 KG Exploration

A.2.1 Relation Pruning

Please provide as few highly relevant relations as possible to the question and its subgoals from the following relations (separated by semicolons).

In-Context Few-Shot

Now you need to directly output relations highly related to the following question and its subgoals in list format without other information or notes.
Q: {}
Subgoals: {}
Topic Entity: {}
Relations: {}

A.2.2 Entity Filtering

Which entities in the following list ([] in Triples) can be used to answer question? Please provide the minimum possible number of entities, strictly following the constraints in the question.

In-Context Few-Shot

Now you need to directly output the entities from [] in Triplets for the following question in list format without other information or notes.
Q: {}
Triplets: {}

A.2.3 Memory Update

Based on the provided information (which may have missing parts and require further retrieval) and your own knowledge, output the currently known information required to achieve the subgoals.

In-Context Few-Shot

Now you need to directly output the results of the following question in JSON format without other information or notes.
Q: {}
Subgoals: {}
Memory: {}

A.3 Answer Generation

Please answer the question based on the memory, related knowledge triplets and your knowledge.

In-Context Few-Shot

Now you need to directly output the results of the following question in JSON format (must include "A" and "R") without other information or notes. If the triplets explicitly contains the answer to the question, prioritize the fact of the triplet over memory.
Q: {}
Memory: {}
Knowledge Triplets: {}

A.4 Failure-Aware Refinement

A.4.1 Retrieval Necessity Diagnosis

Based on the current frontier $\mathcal{E}_t$ and the historical evidence in memory $\mathcal{M}$, determine whether expanding the search space by adding previously pruned entities is necessary to resolve the reasoning impasse.

In-Context Few-Shot

Now you need to directly output the results of the following question in the JSON format (must include "Add" and "Reason") without other information or notes.
Q: {}
Entities set to be retrieved: {}
Memory: {}
Knowledge Triplets: {}

A.4.2 Targeted Backtracking and Re-routing

Under the guidance of memory $\mathcal{M}$ and current subgoals, select the minimum set of high-risk pruned entities to resume expansion and restore a verifiable evidence chain.

In-Context Few-Shot

Now you need to directly output the results for the following Q in the list format without other information or notes.
Q: {}
Reason: {}
Candidate Entities: {}
Memory: {}

A.4.3 Grounded Inference

Based on the verified path segments and unmet constraints, synthesize the most plausible answer. Prioritize entities in the [Graph Evidence], but leverage internal knowledge to bridge missing links and resolve entity identifiers.

In-Context Few-Shot

Now you need to directly output the final answer string based on the following question and provided evidence. Do not output IDs or refusal phrases.
Question: {}
[Graph Evidence]: {}
Answer:

A.5 Blueprint Adaptation

Task: Generate or adapt a relational blueprint for the new question under the structural constraints of provided exemplars.

In-Context Few-Shot

Now, identify the core semantic intent of the [New Question] and directly output its relational blueprint in list format. Use the exemplars only for structural reference.
Q: {}
Output:

B Search SPARQL

To automatically process the KG data, we predefine the SPARQL queries for Freebase, which

can be executed by filling in the entity's *mid* and relation.

B.1 Relation Search

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT DISTINCT ?relation
WHERE {
   ns:mid ?relation ?x .
}
```

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT DISTINCT ?relation
WHERE {
   ?x ?relation ns:mid .
}
```

B.2 Entity Search

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?tailEntity
WHERE {
   ns:mid ns:relation ?tailEntity .
}
```

```
PREFIX ns: <http://rdf.freebase.com/ns/>
SELECT ?tailEntity
WHERE {
   ?tailEntity ns:relation ns:mid .
}
```

B.3 Entity ID Resolution

```
PREFIX ns: <http://rdf.freebase.com/
    ns/>
SELECT DISTINCT ?tailEntity
WHERE {
  BIND(ns:id AS ?entity)
  {
    ?entity ns:type.object.name ?
        tailEntity .
    FILTER(LANG(?tailEntity) = "" ||
        LANGMATCHES(LANG(?tailEntity),
         "en"))
    BIND(1 AS ?priority)
  }
  UNION
  {
    ?entity ns:common.topic.alias ?
        tailEntity .
    FILTER(LANG(?tailEntity) = "" ||
        LANGMATCHES(LANG(?tailEntity),
         "en"))
    BIND(2 AS ?priority)
  }
  UNION
  {
    ?entity <http://www.w3.org
        /2002/07/owl#sameAs> ?
        tailEntity .
    BIND(3 AS ?priority)
  }
}
ORDER BY ASC(?priority) LIMIT 1
```

C Datasets

We evaluate our method on three complex KGQA benchmarks: **WebQSP** (Yih et al., 2016) , **ComplexWebQuestions (CWQ)** (Talmor and Berant, 2018), and **GrailQA** (Gu et al., 2021). All datasets are constructed on the Freebase knowledge graph (Bollacker et al., 2008). The detailed statistics are summarized in Table 5.

WebQSP WebQSP consists of 4,737 natural language questions that require 1-hop or 2-hop reasoning over Freebase. It is widely used to evaluate the robustness of Entity Linking and multi-hop reasoning capabilities.

ComplexWebQuestions (CWQ) CWQ extends WebQSP by introducing four types of complex constraints: *conjunction*, *composition*, *comparative*, and *superlative*. It contains 34,689 questions requiring multi-hop reasoning (up to 4 hops) and strict constraint handling.

GrailQA GrailQA is a large-scale dataset with 64,331 questions designed to test three levels of generalization: *I.I.D.*, *Compositional*, and *Zero-Shot*. It poses significant challenges for models to handle novel schemas and diverse logical structures not seen during training.

Dataset	Answer Format	Train	Test	Licence
ComplexWebQuestions	Entity	27,734	3,531	-
WebQSP	Entity/Number	3,098	1,639	CC Licence
GrailQA	Entity/Number	44,337	1,000	-

Table 5: Statistics of KGQA datasets.

D Baseline Descriptions

To comprehensively evaluate our method, we compare it against two categories of state-of-the-art approaches: (1) LLM-only methods without external knowledge access; and (2) KG-augmented LLM methods, which include both fine-tuned models and training-free prompting frameworks.

D.1 LLM-Only Methods

- **Standard Prompting (IO Prompt)** (Brown et al., 2020b): A direct prompting baseline where the LLM is instructed to generate answers immediately based on the input question, serving as a reference for the model's intrinsic knowledge capacity in a few-shot setting.
- **Chain-of-Thought (CoT)** (Wei et al., 2022): Enhances the reasoning capability of LLMs by

prompting them to generate a sequence of intermediate reasoning steps before deriving the final answer, rather than outputting the result directly.
- **Self-Consistency (SC)** (Wang et al., 2023b): An advanced prompting strategy that samples multiple diverse reasoning paths via CoT and aggregates the results through majority voting to improve answer stability and accuracy.

D.2 Fine-tuned KG-Augmented LLM Methods

- **RE-KBQA** (Cao et al., 2023): Improves reasoning path selection by emphasizing relation exploration and incorporating additional supervision signals to enhance the representation of KG entities.
- **UniKGQA** (Jiang et al., 2023b): A unified framework that integrates the retrieval and reasoning modules into a single shared model architecture to facilitate semantic alignment between questions and KG entities.
- **RoG** (Luo et al., 2024): Synergizes LLMs with KGs by generating faithful reasoning plans grounded in the graph structure, thereby enhancing both interpretability and reasoning reliability.
- **DeCAF** (Yu et al., 2022): A hybrid approach that combines the precision of semantic parsing with the generalization of LLMs to jointly decode valid logical forms and natural language answers.
- **KG-Agent** (Jiang et al., 2025): An autonomous agent framework that formalizes multi-hop reasoning as an executable program, incorporating KG querying tools and memory mechanisms to facilitate step-by-step logical derivation.
- **RnG-KBQA** (Ye et al., 2022): Employ a generate-and-rank paradigm where candidate logical programs are first enumerated and ranked by BERT, then refined into more complex structures using T5.
- **TIARA** (Shu et al., 2022): Adopts a multi-grained retrieval strategy using BERT for schema linking, followed by a T5-based generation module that outputs constrained logical plans.
- **FC-KBQA** (Zhang et al., 2023): Utilizes a fine-to-coarse composition strategy to decouple knowledge acquisition from reasoning, ensuring better generalization across diverse KG schemas.
- **Pangu** (Gu et al., 2023): An end-to-end KBQA model that emphasizes compositional generalization, allowing it to handle more complex query structures.
- **FlexKBQA** (Li et al., 2024): A flexible few-shot framework designed to adapt to new KGs and query languages using only a small number of annotated samples.
- **GAIN** (Shu and Yu, 2024): Focuses on improving model robustness against distribution shifts by introducing a specialized data augmentation mechanism for KGQA tasks.

D.3 Prompting KG-Augmented LLM Methods

- **KB-BINDER** (Li et al., 2023): A few-shot in-context learning approach capable of generating logical forms for heterogeneous KGs by binding question semantics to schema items.
- **KD-CoT** (Wang et al., 2023a): A knowledge-driven chain-of-thought method that dynamically retrieves KG context to guide the LLM in generating faithful and grounded reasoning steps.
- **StructGPT** (Jiang et al., 2023a): Enables LLMs to reason over structured data by defining specific interfaces for iterative information access, filtering, and reasoning.
- **ToG** (Sun et al., 2024): An iterative framework where the LLM performs beam search over the KG, dynamically assessing whether the retrieved triplets are sufficient to answer the question at each hop.
- **PoG** (Chen et al., 2024): PoG decomposes complex queries into structured subgoals and adaptively plans reasoning paths on the KG, enabling better compositional reasoning.

E Implementation Details

E.1 Experiment Settings

In our experiments, we evaluate **CoG** using three different language models: GPT-3.5 and GPT-4 accessed via the OpenAI API [1], and Qwen2.5 [2]. We set the temperature parameter to 0.3, frequency penalty to 0, and presence penalty to 0. The maximum token length for generation is 1024. In all experiments, the depth of exploration is set to 4 to avoid endless exploration. The experiments are conducted on a Linux server equipped with two Intel(R) Xeon(R) Gold 6148 CPUs @ 2.40GHz and 256 GB RAM.

[1] https://platform.openai.com/docs
[2] https://huggingface.co/Qwen

E.2 Blueprint Construction and Copy–Adapt Analysis

This section details the offline relational blueprint template library construction and provides an empirical analysis of how the copy threshold τ_{copy} regulates the **copy–adapt mechanism**—the trade-off between utilizing structural priors and maintaining generative flexibility. For semantic retrieval, the anchors of these templates are encoded using `msmarco-distilbert-base-tas-b`[3]. Our sensitivity experiments are conducted on the WebQSP dataset using GPT-4o-mini.

E.2.1 Blueprint Library Statistics

We construct dataset-specific relational blueprint template libraries by distilling structural patterns from training splits. Table 6 summarizes the statistics, where a defining characteristic is the extreme **structural compression**. For instance, on the large-scale GrailQA dataset, over 44,000 training questions condense into merely 3,703 unique blueprint templates—a ratio of only 8.3%.

This significant reduction empirically validates that the logical topologies of KG reasoning are far more finite and recurrent than their natural language surface forms. By abstracting away concrete entities, the library captures generic reasoning prototypes (e.g., multi-hop filtration, comparative logic) that are highly transferable to unseen queries.

Dataset	Train Size (Queries)	Lib Size (Templates)	Ratio (%)
WebQSP	3,098	569	18.4%
CWQ	27,734	6,747	24.3%
GrailQA	44,337	3,703	8.3%

Table 6: Statistics of offline relational blueprint libraries. The compression ratio highlights the high reusability of structural patterns.

E.2.2 Sensitivity of Copy Threshold

The threshold τ_{copy} acts as the gatekeeper of the **copy–adapt strategy**, determining when to trust structural priors (copy) versus when to rely on LLM generalization (adapt) . Table 7 presents the impact of τ_{copy} on performance, revealing an "inverted-U" trajectory where the optimal accuracy (83.7%) is achieved at $\tau_{copy} = 0.92$. Crucially, we observe a robust performance plateau within the $[0.9, 1.0]$ interval, where Hits@1 consistently exceeds 82%, indicating that CoG is not hypersensitive to specific threshold tuning.

Balancing copy and adapt. A potential concern regarding methods utilizing historical data is whether relying on training-derived blueprints limits generalization compared to pure zero-shot approaches like ToG or GoG. The strategy distribution at the optimal threshold ($\tau_{copy} = 0.92$) provides compelling counter-evidence:

- **Minimal Dependency, Maximum Gain:** At peak performance, the model invokes the copy mode for only **8.7%** of queries (high-confidence matches), while adapting via LLM generation for the remaining **91.3%**. This confirms that CoG does not rigidly overfit to training distributions.
- **Dynamic Fallback:** The blueprint library serves as a "compass" for frequent patterns. For novel or unseen structures (similarity $< \tau_{copy}$), CoG seamlessly transitions to its adapt phase, retaining the full generative flexibility of zero-shot baselines while benefiting from the stability of structural priors when applicable .

Threshold	Copy–Adapt Strategy (Queries)		Hits@1
(τ_{copy})	Copy (Memory)	Adapt (Gen.)	(%)
0.70	1639 (100%)	0	78.9
0.80	1355 (82.7%)	284	80.6
0.90	193 (11.8%)	1446	82.8
0.92	**143 (8.7%)**	**1496**	**83.7**
0.95	74 (4.5%)	1565	82.2
1.00	0 (0.0%)	1639	82.7

Table 7: Impact of the copy threshold τ_{copy} on model behavior. Optimal performance balances selective memory usage with generative fallback.

E.3 Sensitivity Analysis of Reranking Weights

In our main experiments, we employ `gpt-4o-mini` as the backbone agent for reasoning and scoring. The final reranking score is a weighted sum of three components: local semantic relevance (λ_{loc}), step-wise structural alignment (λ_{step}), and global compatibility (λ_{glob}), subject to $\sum \lambda = 1$. In this section, we provide a detailed sensitivity analysis to justify our hyperparameter selection ($\lambda_{loc} = 0.6, \lambda_{step} = 0.25, \lambda_{glob} = 0.15$) and demonstrate the robustness of our approach.

[3] https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b

E.3.1 Experimental Design and Rationale

To effectively decouple the impact of semantic signals versus structural constraints, we designed a two-stage sensitivity test:

(1) Main Trend: Balancing Semantics vs. Structure.

- **Setup:** We vary the primary weight λ_{loc} from 0.4 to 0.8. Crucially, while adjusting λ_{loc}, we maintain a fixed ratio between the two structural weights ($\lambda_{step} : \lambda_{glob} \approx 5 : 3$).
- **Rationale:** This setup treats "structural constraint" as a unified force. By adjusting λ_{loc}, we control the trade-off between *trusting the LLM's local semantic matching* (high λ_{loc}) versus *trusting the Blueprint's structural guidance* (low λ_{loc}).

(2) Internal Ratio Variants: The Composition of Structure.

- **Setup:** We fix λ_{loc} at its optimal value (0.6) and disturb the internal distribution of the remaining 0.4 weight mass. We test variants where the structural weight is dominated by global compatibility (e.g., $\lambda_{glob} = 0.2, 0.3$) or equally distributed.
- **Rationale:** We hypothesize that *step-wise alignment* (checking the validity of each reasoning hop) provides a more precise signal than *global compatibility* (which only checks the final path shape). Therefore, λ_{step} should logically be assigned a higher weight than λ_{glob}.

(3) Uniform Baseline. We also compare against a naive baseline where $\lambda_{loc} = \lambda_{step} = \lambda_{glob} \approx 1/3$, to verify that the performance gains stem from our specific weighting strategy rather than simple score ensembling.

E.3.2 Results and Analysis

The results on WebQSP are visualized in Figure 3.

Optimal Balance Found at $\lambda_{loc} = 0.6$. The blue curve exhibits a clear inverted-U trajectory.

- **Under-weighting Semantics ($\lambda_{loc} < 0.6$):** When λ_{loc} is low (e.g., 0.4), the strict structural constraints may filter out semantically correct entities that have weaker structural signals (e.g., due to sparse KG connections), leading to a performance drop (82.73%).
- **Over-weighting Semantics ($\lambda_{loc} > 0.6$):** As λ_{loc} approaches 0.8, the agent behaves like a greedy semantic searcher, ignoring global blueprint constraints, which drops accuracy to 82.79%.

Superiority of Step-wise Verification. The comparison at $\lambda_{loc} = 0.6$ (orange markers) strongly supports our hypothesis regarding internal structural ratios. Even with the optimal primary weight, allocating more weight to global compatibility ($\lambda_{glob} > \lambda_{step}$) leads to significant degradation (82.43% and 82.36%). This confirms that `gpt-4o-mini` benefits more from fine-grained, step-by-step structural verification than from coarse-grained global checks.

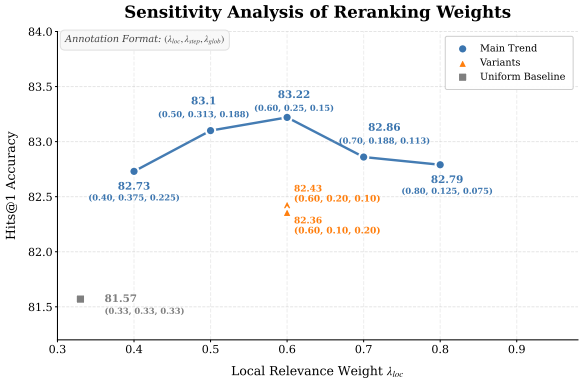

Figure 3: Sensitivity analysis of reranking weights. The blue line tracks performance as λ_{loc} varies. The **orange triangles** represent variants with suboptimal internal structural ratios at the peak $\lambda_{loc} = 0.6$. The **grey square** denotes the naive uniform baseline. The explicit weight configuration $(\lambda_{loc}, \lambda_{step}, \lambda_{glob})$ is annotated for each point.

F Detailed Performance Breakdown

Figure 4 details the performance comparison across four query types on CWQ. Our method (CoG) consistently outperforms the baseline across all categories, with the performance gap widening as structural complexity increases.

Most notably, we observe the substantial accuracy gains in **Conjunction (+4.7%)** and **Superlative (+4.6%)** queries. These categories demand rigorous logic: conjunctions require satisfying multi-branch constraints simultaneously, while superlatives necessitate global comparison over candidate sets. The significant boost here substantiates that our blueprint-guided constraints effectively prevent reasoning drift in complex scenarios. Meanwhile, performance on standard **Composition (+2.6%)**

and **Comparative (+3.3%)** chains remains robust, confirming that our structural constraints enhance reliability without compromising efficiency on fundamental tasks.

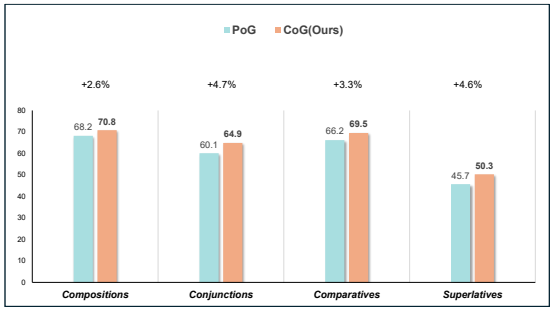

Figure 4: Performance breakdown by query type. Our method achieves significant accuracy gains, particularly in structurally complex categories like **Conjunction (+4.7%)** and **Superlative (+4.6%)**.

G Analysis of Failure-Aware Refinement

We scrutinize the activation frequency and error-correction capability of the *Failure-Aware Refinement* module across datasets. Table 8 details the triggering statistics, while Figure 5 breaks down the outcomes of these triggered cases.

Activation and Complexity Correlation. Table 8 reveals a clear correlation between query complexity and refinement activation. The module is triggered most frequently on **CWQ (45.3%)**, significantly higher than on WebQSP (32.3%) and GrailQA (29.9%). This high activation rate on CWQ reflects the inherent difficulty of its multi-hop questions, where intermediate reasoning steps are prone to "dead ends," thereby necessitating frequent intervention by our safety-net mechanism.

Recovery Effectiveness. Figure 5 illustrates the success rate of the refinement module in rectifying potential failures.

- **I.I.D. vs. Zero-Shot:** The module proves most effective on the I.I.D. WebQSP dataset, recovering **65.4%** of triggered queries. In contrast, performance dips on the zero-shot GrailQA dataset (38.8%). This variance suggests that while self-correction is powerful for standard reasoning, it faces challenges when grounding novel entities without prior schema exposure.
- **Contribution to Robustness:** Despite the lower recovery rate on complex datasets, rectifying nearly **40%** of potential failures on CWQ and GrailQA constitutes a critical portion of the overall performance gains reported in the main results.

Dataset	Total Queries	Overall Hits@1	Refinement Trigger Count	Refinement Trigger Rate
CWQ	3531	66.9	1599	45.3%
WebQSP	1639	86.8	529	32.3%
GrailQA	1000	79.2	299	29.9%

Table 8: Statistics of refinement activation. We report the frequency (**Count**) and proportion (**Rate**) of queries necessitating self-correction intervention.

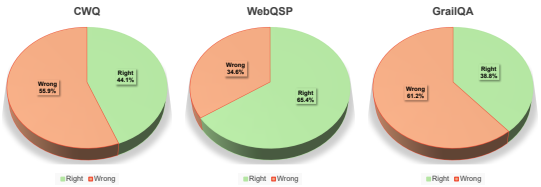

Figure 5: Outcome distribution of the refinement process. The "Right" segment represents the proportion of queries successfully rectified by the module after being triggered.

H Quantitative Error Analysis

To explicitly quantify how CoG mitigates the two core dimensions of cognitive rigidity—*Error Cascading* and *Reasoning Stagnation*—we conducted an ablation-based attribution analysis. We define the performance gap between the full CoG framework and its component-ablated variants as a proxy for the volume of specific errors corrected by our mechanisms. Figure 6 visualizes these gains across three benchmarks.

Mitigating Error Cascading (Blue Bars). The blue bars in Figure 6 quantify the contribution of *Blueprint-Guided Exploration* (System 1). By enforcing structural constraints, CoG consistently filters out neighborhood noise that typically misleads baselines into irreversible deviations. On the noise-intensive CWQ dataset, this mechanism rescues **3.4%** of queries from cascading errors. Even on simpler datasets like WebQSP and GrailQA, the consistent gains (+2.8% and +2.4%) confirm that structural blueprints serve as a robust, universal filter against the "butterfly effect" of early selection errors.

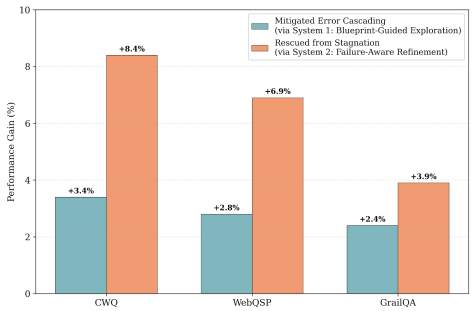

Figure 6: Quantitative attribution of performance gains to CoG's core mechanisms. The **blue bars** quantify the mitigation of *Error Cascading* via **System 1** (Blueprint-Guided Exploration), acting as a structural filter against noise. The **orange bars** represent the recovery from *Reasoning Stagnation* via **System 2** (Failure-Aware Refinement), resolving structural misalignment through diagnostic backtracking. The pronounced impact on CWQ highlights the necessity of the refinement mechanism for complex multi-hop reasoning.

Resolving Stagnation (Orange Bars). The orange bars highlight the critical role of *Failure-Aware Refinement* (System 2) in overcoming reasoning dead-ends. This effect is particularly dominant on the complex CWQ benchmark, where the diagnostic re-routing mechanism recovers a substantial **8.4%** of queries that initially stagnated. The sharp contrast between the high gain on CWQ and the moderate gains on other datasets suggests a clear correlation: as reasoning depth and complexity increase, the risk of structural misalignment grows, making CoG's "safety net" mechanism increasingly indispensable for navigating treacherous search spaces.

I Generalization Analysis on Heterogeneous Knowledge Graphs

To assess the robustness of CoG beyond the specific schema of Freebase, we extended our evaluation to **Wikidata** (Vrandečić and Krötzsch, 2014), a knowledge graph characterized by significantly larger scale and greater heterogeneity. This setting serves as a rigorous stress test for the model's ability to handle schema variations and navigate noisy environments without specific fine-tuning.

Experimental Setup. We mapped the entity annotations in WebQSP and CWQ from Freebase IDs (MIDs) to Wikidata IDs (QIDs). It is important to acknowledge that due to ontological misalignment between the two KGs, a subset of entities could not be perfectly mapped, introducing an inherent layer of noise and information loss. We benchmarked CoG against the strongest baselines, ToG and PoG, under this challenging setting.

Method	WebQSP	CWQ
Source Domain: Freebase		
ToG	76.2	57.1
PoG	82.0	63.2
CoG (Ours)	**86.8**	**66.9**
Target Domain: Wikidata		
ToG	68.6	54.9
PoG	73.8	60.7
CoG (Ours)	**76.5**	**62.8**

Table 9: Performance comparison using different source KGs (Freebase vs. Wikidata). Despite the domain shift, CoG demonstrates superior robustness against schema heterogeneity compared to ToG and PoG.

Results and Discussion. The comparative results are summarized in Table 9. We first observe a universal performance decline across all methods when transitioning from Freebase to Wikidata. This drop is anticipated and stems primarily from two factors: (1) the *annotation bias*, as the datasets were originally curated for Freebase, rendering the mapping process lossy; and (2) the *structural complexity*, where Wikidata's dense topology substantially expands the search space and complicates relation filtering.

Despite these environmental hurdles, CoG maintains a distinct advantage over the baselines. Specifically, CoG outperforms the strongest competitor, PoG, by margins of **2.7%** on WebQSP and **2.1%** on CWQ. These results offer two critical insights into the mechanism of CoG:

- **Schema Agnosticism:** The Relational Blueprint mechanism appears to capture abstract reasoning patterns (e.g., *Subject* $\rightarrow$ *Attribute* $\rightarrow$ *Value*) rather than overfitting to specific naming conventions (e.g., Freebase's `people.person.place_of_birth`). This abstraction allows the generated blueprints to adapt effectively to the distinct property predicates found in Wikidata.

- **Resilience to Noise:** The performance gap suggests that the *Failure-Aware Refinement* module is particularly effective in heterogeneous settings. When initial search paths are misled by Wikidata's noise, CoG's diagnostic backtracking prevents the reasoning chain

from premature collapse—a failure mode frequently observed in baselines lacking such corrective mechanisms.

J Case Studies

We conduct a qualitative analysis of two representative cases to scrutinize the interplay between structural priors and reasoning stability.

Case 1 (Figure 7) exemplifies CoG's capacity to navigate the "neighborhood noise" inherent in super-nodes, where local semantic cues often diverge from the underlying reasoning logic. In this query concerning Angelina Jolie—an entity with an overwhelming density of acting-related metadata—the baseline PoG, lacking global structural constraints, succumbs to the high-frequency semantic traps of her acting career. Consequently, PoG exhausts 5,544 tokens drifting through unrelated films (e.g., *The English Patient*) before eventually hallucinating *By the Sea* as a partially correct but logically invalid answer. In contrast, CoG strictly adheres to the adapted relational blueprint ⟨film.director.film, film.film.music⟩. Even though the local semantic score for the "actor" relation is dominant, the step-wise alignment signal ϕ_{step} effectively penalizes these noisy distractors. By anchoring the search within the directorial slot, CoG identifies the correct answer, *In the Land of Blood and Honey*, with minimal overhead, demonstrating that relational blueprints function as a robust "structural compass" in complex semantic landscapes.

Case 2 (Figure 8) illustrates how CoG resolves structural misalignment through diagnostic refinement, distinguishing its systematic backtracking from the stochastic retries of existing agents. When searching for the specific college attended by Kevin Hart under postgraduate constraints, PoG correctly identifies the *Community College of Philadelphia* but falls into a state of cognitive rigidity. Because its self-correction mechanism is purely heuristic and lacks structural foresight, PoG enters a loop of 26 redundant calls (14,187 tokens) on the same invalid node, unable to escape the local search impasse. Conversely, CoG leverages Failure-Aware Refinement to execute a structural diagnosis upon reaching the *Castlemont High School* node. Recognizing that this entity type is ontologically incompatible with the university-level slots required by the blueprint, CoG avoids blind retries. Instead, this diagnostic failure triggers a targeted re-routing back to the education hub, leading directly to *Temple University*, where it extracts the specific value (5,478) and satisfies the condition. This success underscores CoG's ability to maintain reasoning momentum where baseline agents stagnate.

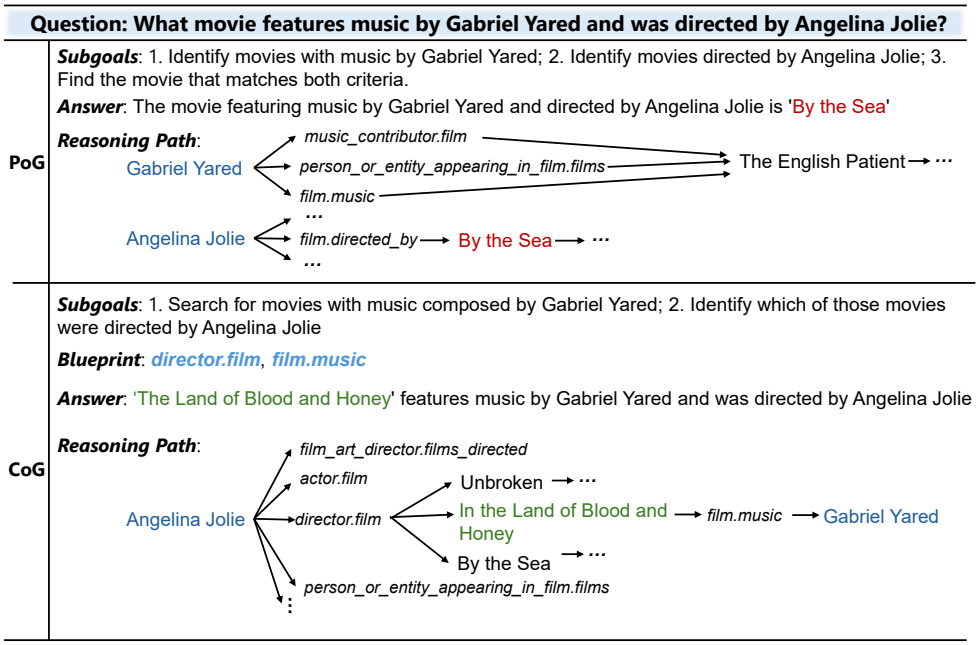

Figure 7: Case 1: Comparison of reasoning trajectories on a super-node. PoG is misled by the local density of acting roles, leading to semantic drift, while CoG utilizes the relational blueprint to filter distractors and maintain structural consistency.

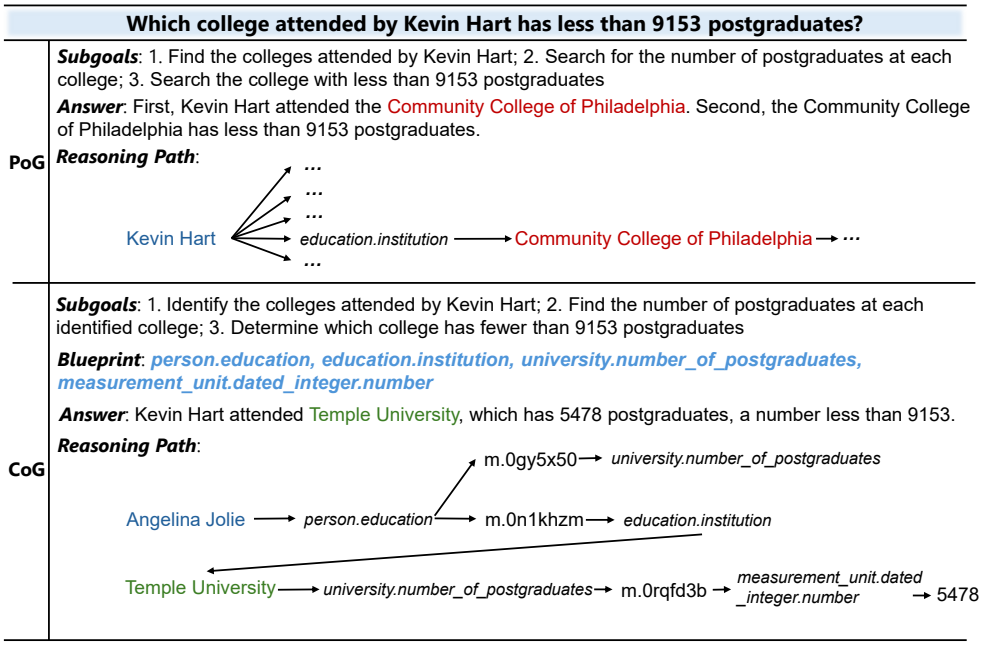

Figure 8: Case 2: Visualization of CoG's diagnostic re-routing. Unlike the stochastic stagnation observed in PoG (26 failed repetitive calls), CoG identifies structural misalignment at the High School node and utilizes System 2 reflection to re-route toward a valid reasoning branch.