Interactive Combinatorial Reinforcement Learning for Knowledge Graph Reasoning

Jun Nie[1,2] **Yonggang Zhang**[3†] **Tongliang Liu**[4] **Chengqi Zhang**[5]
Xinmei Tian[1] **Bo Han**[2†]

[1]MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China
[2]TMLR Group, Hong Kong Baptist University [4]The University of Sydney
[3]The Hong Kong University of Science and Technology, HKGAI
[5]The Hong Kong Polytechnic University

Abstract

Knowledge Graph Question Answering (KGQA) increasingly requires multi-turn interaction with knowledge graphs (KGs) to derive final answers. Such interactive capabilities often rely on proprietary large-scale LLMs, which hinders cost-efficient and local deployment. Reinforcement learning offers a scalable alternative for endowing smaller open-source LLMs with multi-turn interaction ability, as it enables models to improve from self-explored interaction trajectories rather than relying on expensive expert demonstrations. However, KGQA presents a distinctive challenge for RL: unlike tasks where exploration can proceed in a relatively unconstrained textual space, KG reasoning is governed by a sparse graph topology. Free-form policy rollouts frequently generate relation sequences that cannot be executed on the graph, making reward signals sparse and unstable. We propose ICOR, an Interactive Combinatorial Reinforcement Learning framework that grounds policy exploration in the combinatorial structure of KGs. ICOR first retrieves a query-specific set of candidate relations and then restricts policy rollouts to relation-path compositions within this graph-derived space. This design increases the likelihood that sampled trajectories are executable and thus informative for policy optimization. Building on this constrained rollout space, ICOR applies GRPO with a staged feedback mechanism that guides learning from structural validity to answer-consistent reasoning. Experiments on multiple KGQA datasets demonstrate the effectiveness of ICOR.

1 Introduction

The integration of Large Language Models (LLMs) with Knowledge Graphs has evolved from simple retrieval-augmented generation to complex, agent-based reasoning. In tackling intricate Knowledge Graph Question Answering (KGQA) tasks, recent studies (Sun et al., 2024) have demonstrated that static, one-shot retrieval is often insufficient. Instead, dynamic, multi-turn interactions—where an agent actively formulates reasoning plans, executes them against the knowledge graph, and iteratively refines its strategy based on execution feedback—are crucial for handling multi-hop queries. Pioneering frameworks such as ToG (Sun et al., 2024) and PoG (Chen et al., 2024a) have validated this interactive paradigm, enabling LLMs to treat the KG as an external environment to verify logical chains and correct errors, achieving impressive performance on complex benchmarks.

[†]Correspondence to Yonggang Zhang (zhangyg@ust.hk) and Bo Han (bhanml@comp.hkbu.edu.hk).
Code: https://github.com/junnie00/ICOR

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

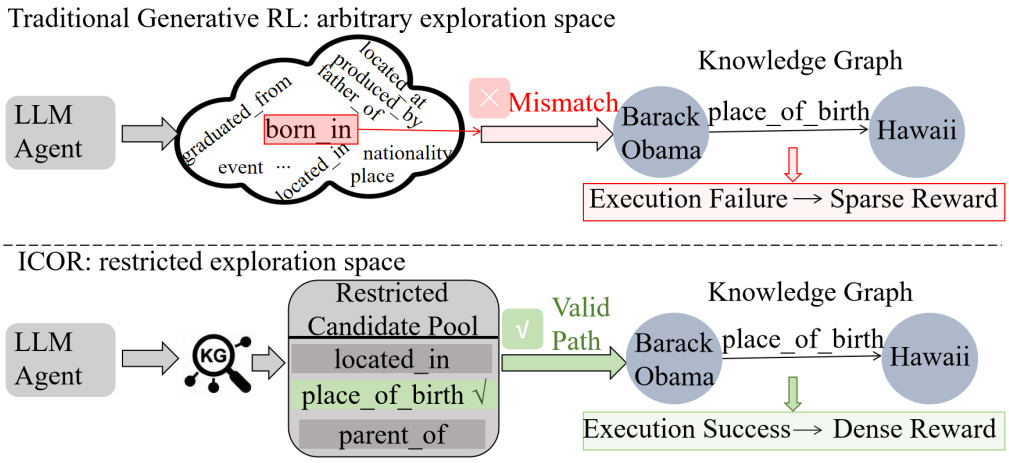

Figure 1: Constrained rollout optimization in ICOR. Unlike unconstrained on-policy rollouts that often fail to produce executable graph actions, ICOR constructs a query-conditioned relation subspace so that sampled trajectories are more likely to yield valid execution feedback for GRPO updates.

However, these state-of-the-art approaches often rely on proprietary, commercial foundation models such as GPT-4 (Achiam et al., 2023). While powerful, these closed-source models incur high inference costs, suffer from substantial latency, and raise data privacy concerns. Consequently, there is a pressing need to transfer such interactive reasoning capabilities to smaller, open-source models (e.g., 7B parameters), enabling efficient and localized deployment.

To endow smaller models with this proficiency, a natural approach is Supervised Fine-Tuning (SFT). However, SFT relies on high-quality interaction trajectories, whose annotation is particularly expensive for KGQA because they must capture both valid graph execution and error-correcting reasoning behavior. Furthermore, relying solely on static datasets limits the model's ability to adapt to its own exploration errors during graph interaction. To circumvent this data bottleneck, Reinforcement Learning (RL) has emerged as a scalable paradigm (Schulman et al., 2017; Ouyang et al., 2022). By enabling models to learn from self-exploration and environmental feedback, RL offers a pathway for agents to improve their reasoning policies without relying on extensive expert supervision.

However, applying general-purpose RL to KGQA presents a distinctive optimization challenge. Unlike tasks where exploration can proceed in a relatively unconstrained textual space, KG reasoning is governed by a sparse and discrete graph topology. A free-form policy rollout may generate relation sequences that are linguistically plausible but cannot be executed on the KG, yielding sparse or uninformative rewards for policy optimization. Although prior KGQA methods have explored structurally valid path generation or constrained inference, how to incorporate graph constraints into on-policy RL for learning from self-explored interaction trajectories remains underexplored. In this setting, graph constraints should not merely serve as post-hoc validity checks; they should define the rollout space in which executable and reward-bearing trajectories can be sampled.

To address this challenge, we propose Interactive Combinatorial Reinforcement Learning (ICOR), a framework that reformulates KG interaction as constrained combinatorial rollout optimization. Different from open-ended relation-path generation, ICOR first leverages a dense retriever to identify a query-specific set of candidate relations from the KG. It then restricts policy rollouts to relation-path compositions within this graph-derived space, increasing the likelihood that sampled trajectories are executable and informative for GRPO-based policy optimization. Building on this constrained rollout space, ICOR further organizes graph execution feedback into a staged learning process: the policy is initialized toward valid path generation and then progressively optimized from structural validity to answer-consistent reasoning. This staged design reflects the dependency structure of KGQA, where answer-level feedback becomes informative only after the model has identified a correct reasoning path.

By integrating graph-derived combinatorial action spaces with GRPO, ICOR enables a 7B parameter model to learn multi-hop KG interaction from execution feedback. Our approach reduces the dependence on manually annotated multi-turn interaction trajectories and provides an effective route toward local, open-source KG reasoning agents.

Our main contributions are summarized as follows:

- We formulate KGQA with smaller open-source LLMs as a constrained on-policy reinforcement learning problem, where graph constraints are used to construct executable and reward-bearing rollouts rather than only to validate generated paths.
- We propose ICOR, a GRPO-based training framework that combines query-specific relation-space construction, SFT-based cold-start initialization, and staged feedback optimization to stabilize learning from graph execution signals.
- Experiments on multiple KGQA datasets demonstrate the effectiveness of ICOR, showing that constrained rollout optimization improves the graph-grounded reasoning ability of open-source LLMs.

2 Related Work

2.1 Knowledge Graph Question Answering

Knowledge Graph Question Answering (KGQA) has been widely studied through semantic parsing, retrieval-augmented reasoning, and more recently LLM-based graph interaction. Semantic parsing methods translate natural language questions into executable logical forms, such as SPARQL queries (Yu et al., 2023; Shu et al., 2022). Although such methods can achieve precise schema grounding, they typically rely on annotated logical forms and may struggle with complex or open-domain schema variations. Retrieval-augmented methods instead retrieve relevant subgraphs or relation paths before answer prediction. Representative approaches include PullNet (Sun et al., 2019), GNN-RAG (Mavromatis and Karypis, 2025), and RoG (Luo et al., 2024), where LLMs are fine-tuned to generate relation paths as retrieval plans. Beyond static retrieval or path generation, recent interactive KGQA systems treat the KG as an external environment and iteratively refine reasoning based on graph feedback. Methods such as ToG (Sun et al., 2024) and PoG (Chen et al., 2024a) demonstrate the effectiveness of multi-turn graph exploration for complex questions. However, these systems are often driven by proprietary LLMs and rely on prompting-based interaction rather than directly training a local policy model. In contrast, ICOR focuses on enabling smaller open-source LLMs to acquire interactive KG reasoning ability through constrained on-policy reinforcement learning.

Recent learning-based systems provide complementary approaches. KBQA-o1 (Luo et al., 2025) explores logical-form trajectories with policy- and reward-model-guided MCTS, followed by incremental SFT. Graph-R1 (Luo et al., 2026) optimizes natural-language retrieval queries over a document-derived knowledge hypergraph with end-to-end RL. KBQA-R1 (Sun et al., 2026) learns multi-turn S-Expression actions with GRPO, dynamic relation grounding, and a referenced-rejection-sampling warm start. ICOR instead retrieves exact KG relations once per question and directly composes and revises relation paths within this candidate set, using dependency-aware staged GRPO without inference-time tree search.

2.2 RL for Tool-Integrated Reasoning

Tool-Integrated Reasoning (TIR) enables LLMs to interact with external tools or environments, such as code interpreters, search engines, and structured databases (Mai et al., 2025; Jin et al., 2025; Zhang et al., 2025). Early approaches often rely on supervised fine-tuning with expert trajectories (Chen et al., 2024b), but collecting high-quality trajectories for long-horizon interaction is costly, and static imitation learning can suffer from distribution shift when the model encounters its own mistakes during inference (Fu et al., 2025). Reinforcement learning provides a natural alternative by optimizing policies from execution feedback (Schulman et al., 2017; Ouyang et al., 2022), allowing models to improve through self-exploration.

Applying RL to KGQA, however, requires special consideration of the graph structure. Unlike general tool-use tasks where actions may be relatively flexible, KG interaction is constrained by a discrete relation schema and sparse graph topology. Free-form rollouts can easily produce non-executable relation sequences, leading to sparse and noisy feedback. ICOR addresses this issue by constructing a query-specific relation space from the KG and optimizing policy rollouts within this constrained combinatorial space. Moreover, instead of aggregating path and answer rewards into a

single objective, ICOR adopts dependency-aware staged optimization, reflecting the fact that answer prediction is meaningful only after a correct reasoning path has been identified.

3 Methodology

We introduce Interactive Combinatorial Reinforcement Learning (ICOR), a framework for learning multi-turn KG interaction policies through constrained on-policy reinforcement learning. ICOR uses graph-derived relation constraints to construct executable rollout spaces, and optimizes the policy with staged GRPO feedback from graph execution. This formulation turns KG constraints from inference-time validity rules into training-time structures for policy improvement.

3.1 Problem Reformulation: Graph Reasoning as MDP

We formulate multi-hop reasoning on a KG $\mathcal{G} = (\mathcal{E}, \mathcal{R})$ as a finite-horizon Markov Decision Process (MDP). Given a question q, let $\mathcal{E}_0$ denote the topic entity set identified from the question. At each interaction step, the agent maintains a current entity frontier $\mathcal{E}_t$ and an interaction history H_{t-1}.

Given a query-specific candidate relation set $\mathcal{C}_q \subseteq \mathcal{R}$ constructed by the retriever in Section 3.2, we define the path-level action space as

$$\mathcal{A}_q = \bigcup_{\ell=1}^{L_{\max}} \mathcal{C}_q^{\ell}, \tag{1}$$

where each action is a relation path $p_t = (r_1, \ldots, r_\ell)$ with $r_i \in \mathcal{C}_q$ and $\ell \leq L_{\max}$. In other words, the policy does not generate arbitrary text actions from the full vocabulary, but selects a relation-path action from a query-conditioned combinatorial space.

- **State Space** $\mathcal{S}$: A state is defined as $s_t = (q, \mathcal{E}_t, H_{t-1})$, where q is the natural language question, $\mathcal{E}_t$ is the current entity frontier, and H_{t-1} records previous reasoning paths and graph execution observations.
- **Action Space** $\mathcal{A}$: For each query q, the available action space is the path-level space $\mathcal{A}_q$ defined in Eq. (1). A policy action $p_t \in \mathcal{A}_q$ is a relation sequence composed only of candidate relations retrieved from the KG.
- **Transition** $\mathcal{P}$: The environment executes p_t on the KG from the current entity frontier:

  $$\mathcal{E}_{t+1} = \mathrm{Exec}_{\mathcal{G}}(\mathcal{E}_t, p_t), \tag{2}$$

  where $\mathrm{Exec}_{\mathcal{G}}(\mathcal{E}_t, p_t)$ returns the set of entities reachable by following the relation path p_t from entities in $\mathcal{E}_t$. If no entity is reachable, the execution result is empty, indicating an invalid or non-executable path. The observation o_t is then derived from $\mathcal{E}_{t+1}$, such as the returned entity set, intermediate subgraph, or an empty observation.
- **Reward** $\mathcal{R}_{\mathrm{rew}}$: The reward is a multi-tier signal that evaluates syntactic validity, path-level correctness, and answer-level consistency.

The objective is to learn a policy $\pi_\theta(p_t \mid s_t, \mathcal{C}_q)$ that maximizes the expected cumulative reward over the interaction horizon:

$$\max_{\theta} \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=1}^{T} \mathcal{R}_{\mathrm{rew}}(s_t, p_t, o_t) \right], \tag{3}$$

where $\tau = \{(s_t, p_t, o_t)\}_{t=1}^{T}$ denotes an interaction trajectory.

3.2 Semantic Action Space Pruning

A critical challenge in applying LLMs to KGs is the "vocabulary mismatch" between the model's generative space and the graph's schema. To mitigate hallucinations, we employ Semantic Action Space Pruning. We train a dense dual-encoder E_ϕ to map the query q and relations r into a shared latent manifold, optimized via a Contrastive InfoNCE objective:

$$\mathcal{L}_{\text{align}} = -\mathbb{E}\left[\log \frac{\exp(s(q, r^+))}{\exp(s(q, r^+)) + \sum_{r^- \in \mathcal{N}} \exp(s(q, r^-))} \right], \tag{4}$$

where r^+ and r^- represent the positive and negative relations for query q, respectively, and $\mathcal{N}$ denotes the set of negative samples.

This alignment projects the vast relation set $\mathcal{R}$ onto a query-specific candidate set $\mathcal{C}_q$ consisting of the top-K retrieved relations:

$$\mathcal{C}_q = \mathrm{TopK}_{r \in \mathcal{R}}\, s(q, r). \tag{5}$$

The resulting set $\mathcal{C}_q$ serves as the building block of the path-level action space $\mathcal{A}_q$ in Eq. (1), thereby transforming open-ended text generation into constrained relation-path composition.

3.3 Policy Initialization via Instruction Tuning

To jump-start the model's ability to map questions to valid relation paths, we employ a SFT Cold Start phase. We utilize high-quality relation path annotations from RoG blended with the general-purpose Alpaca dataset to mitigate catastrophic forgetting. The policy is initialized by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{sft}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{mix}}} \left[\sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t}) \right]. \tag{6}$$

3.4 Self-Evolving Policy Optimization via GRPO

Although SFT provides a useful initialization, it is trained on static demonstrations and cannot fully cover the states induced by the model's own graph-interaction rollouts. We therefore further optimize the policy with Group Relative Policy Optimization (GRPO), allowing the model to improve from self-explored trajectories under graph execution feedback.

Constrained Group Rollouts. For each query q, the retriever constructs a query-specific candidate relation set $\mathcal{C}_q$. The policy samples a group of trajectories

$$\mathcal{B}_q = \{T_1, \ldots, T_G\}, \qquad T_i \sim \pi_\theta(\cdot \mid q, \mathcal{C}_q), \tag{7}$$

where each trajectory T_i contains a generated relation path p_i, graph execution observations, and a final predicted answer set $\hat{\mathcal{Y}}_i$. If no relation path can be parsed from T_i, we set $p_i = \varnothing$. Since each valid path is composed within the path-level action space induced by $\mathcal{C}_q$, policy exploration is grounded in the combinatorial structure of the KG rather than the full open-ended textual space.

Dependency-Aware Reward Design. A straightforward GRPO implementation may aggregate multiple reward terms into a single scalar objective. However, KGQA has an inherent dependency structure: answer correctness is meaningful only after the model identifies a correct reasoning path. If the generated path is wrong, the answer reward is usually uninformative for improving graph navigation. Therefore, ICOR separates path discovery from answer prediction while preserving their conditional dependency.

Let $\mathcal{P}_q^\star$ denote the set of correct relation paths for query q, and let $\mathcal{Y}_q^\star$ denote the ground-truth answer set. We define three reward components:

$$R_{\text{syn}}(T_i) = \mathbb{I}[T_i \in \mathcal{T}_{\text{syn}}], \quad R_{\text{path}}(T_i) = \mathbb{I}[T_i \in \mathcal{T}_{\text{syn}}]\, \mathbb{I}\big[p_i \in \mathcal{P}_q^\star\big], \quad R_{\text{ans}}(T_i) = \mathrm{F1}\Big(\hat{\mathcal{Y}}_i, \mathcal{Y}_q^\star\Big). \tag{8}$$

Here, R_{syn} checks whether the trajectory follows the required interaction format, R_{path} checks whether a syntactically valid trajectory generates a relation path belonging to the correct path set, and R_{ans} measures answer-level consistency. The concrete reward coefficients are specified in the experimental setup.

Tiered Optimization. Instead of optimizing all rewards over all rollouts, ICOR applies rewards over nested validity subsets:

$$\mathcal{B}_q \supseteq \mathcal{T}_{\text{syn}} \supseteq \mathcal{T}_{\text{path}}, \qquad \mathcal{T}_{\text{path}} = \{T_i \in \mathcal{T}_{\text{syn}} \mid p_i \in \mathcal{P}_q^\star\}. \tag{9}$$

The optimization proceeds in three stages:

$$(\mathcal{V}_q^{(1)}, R^{(1)}) = (\mathcal{B}_q, R_{\text{syn}}{+}R_{\text{path}}), \quad (\mathcal{V}_q^{(2)}, R^{(2)}) = (\mathcal{T}_{\text{syn}}, R_{\text{path}}), \quad (\mathcal{V}_q^{(3)}, R^{(3)}) = (\mathcal{T}_{\text{path}}, R_{\text{ans}}). \tag{10}$$

The first stage optimizes both syntax and path rewards over the full sampled group, encouraging parseable trajectories while already providing path-level guidance when possible. The second stage restricts optimization to syntactically valid trajectories and focuses on correct path discovery. The third stage further restricts optimization to trajectories with correct paths and applies answer-level feedback, so that answer prediction is learned conditioned on successful path discovery. This avoids conflating path-search failures with answer-selection failures in a single reward space.

Optimization Objective. For stage m, rewards are normalized within the current valid subset $\mathcal{V}_q^{(m)}$:

$$A_i = \frac{R^{(m)}(T_i) - \mu_q^{(m)}}{\sigma_q^{(m)} + \epsilon}, \tag{11}$$

where $\mu_q^{(m)}$ and $\sigma_q^{(m)}$ are computed over trajectories in $\mathcal{V}_q^{(m)}$. If $\mathcal{V}_q^{(m)}$ is empty, the corresponding group is skipped to avoid undefined normalization.

The stage-wise GRPO objective is

$$\mathcal{J}^{(m)}(\theta) = \mathbb{E}_q \left[\frac{1}{|\mathcal{V}_q^{(m)}|} \sum_{T_i \in \mathcal{V}_q^{(m)}} \left(\Psi(r_i(\theta), A_i) - \beta \mathbb{D}_{\mathrm{KL}} \left(\pi_\theta \| \pi_{\mathrm{ref}} \right) \right) \right], \tag{12}$$

where

$$r_i(\theta) = \frac{\pi_\theta(T_i \mid q, \mathcal{C}_q)}{\pi_{\theta_{\mathrm{old}}}(T_i \mid q, \mathcal{C}_q)}, \tag{13}$$

and $\Psi(\cdot)$ denotes the clipped GRPO surrogate. By computing relative advantages within stage-specific validity subsets, ICOR preserves the dependency between path discovery and answer prediction, instead of collapsing all learning signals into a single reward aggregation objective.

3.5 Probabilistic Advantage of Constrained Action Spaces

We model the LLM's reasoning process as a trajectory $\tau = (a_1, a_2, \ldots, a_T)$ within a Markov Decision Process (MDP). Let $\mathcal{V}$ be the full vocabulary of the LLM, and $\mathcal{C}_q \subset \mathcal{R} \subset \mathcal{V}$ be the pruned combinatorial action space defined in Section 3.2.

Definition 3.1 (Constrained Policy)**.** Let $\pi_{\text{gen}}(a|s)$ be the original policy distribution over $\mathcal{V}$. The ICOR policy $\pi_\theta(a|s)$ is defined as the renormalization of π_{gen} over the query-specific subspace $\mathcal{C}_q$:

$$\pi_\theta(a_t|s_t) = \frac{\pi_{\text{gen}}(a_t|s_t) \cdot \mathbb{I}(a_t \in \mathcal{C}_q)}{\sum_{a' \in \mathcal{C}_q} \pi_{\text{gen}}(a'|s_t)}, \tag{14}$$

where $\mathbb{I}(\cdot)$ is the indicator function and the denominator acts as a local partition function $Z(s_t, \mathcal{C}_q)$.

Theorem 3.2 (Conditional Probability Amplification under Retrieved Action Spaces)**.** *For any target reasoning trajectory τ^* composed of relations in $\mathcal{C}_q$, the ICOR policy π_θ guarantees $P_{ICOR}(\tau^*) > P_{gen}(\tau^*)$, provided the base policy π_{gen} assigns non-zero probability to invalid transitions $\mathcal{V} \setminus \mathcal{C}_q$.*

Theorem 3.2 demonstrates that by explicitly grounding the LLM within a topologically valid action space, ICOR reallocates the probability mass that would otherwise be wasted on hallucinations or syntax-breaking tokens. This results in an exponential amplification of the likelihood of correct paths as the reasoning horizon T increases. This result provides an intuitive explanation for why constraining rollouts to retrieved relations can increase the probability of sampling executable paths during policy optimization. The rigorous derivation of this density amplification factor is provided in Appendix A.3.1.

3.6 Variance Reduction Analysis of Hierarchical Reward Filtering

Standard Reinforcement Learning in KGQA often suffers from reward sparsity and objective conflict, where a single failure in the reasoning chain (e.g., a minor tool-format error) results in a total reward of zero, providing no informative gradient for semantic learning.

Table 1: Performance comparison with different baselines on the two KGQA datasets. † denotes our reproduction with the Qwen2.5-7B-Instruct policy backbone.

Type	Methods	WebQSP		CWQ	
		F1	Hits@1	F1	Hits@1
Non-LLMs methods	Rigel	-	73.3	-	48.7
	TIARA	78.9	75.2	-	-
	UniK-QA	<u>79.1</u>	-	-	-
	UniKGQA	72.2	77.2	49.4	51.2
Prompting-LLMs Only methods	Zero-shot (gpt-4)	59.7	62.3	37.9	42.7
	Few-shot (gpt-4)	62.7	68.7	43.7	51.5
	CoT (gpt-4)	65.3	72.1	44.7	53.5
Prompting-LLMs + KG	ToG (gpt-3.5)	72.3	75.1	56.9	57.5
	ToG (gpt-4)	75.9	81.8	<u>60.2</u>	68.5
	InteractiveKBQA(gpt-4)	66.3	72.9	56.5	60.3
	SubgraphRAG (GPT4o)	78.2	**90.9**	59.4	67.5
	PoG (GPT4)	-	87.3	-	**75.0**
Finetuning-LLMs + KG	NSM	62.8	68.7	42.4	47.6
	KD-CoT	52.5	68.6	49.7	53.3
	DECAF	78.8	82.1	-	-
	RoG	69.8	83.2	56.2	61.4
	GNN-RAG	71.3	85.7	59.4	66.8
	KBQA-R1†	**80.2**	86.9	-	-
	ICOR	75.4	<u>88.8</u>	**61.3**	<u>69.5</u>

Lemma 3.3 (Signal Interference)**.** *In a multi-objective optimization where $R_{total} = \sum \omega_i R_i$, if the gradients of the syntax constraint ∇R_{syn} and semantic reasoning ∇R_{sem} are non-aligned, the high variance of R_{sem} in the early training stages acts as noise, leading to sub-optimal local minima.*

Lemma 3.3 identifies the root cause of optimization failure in naive KGQA-RL: the "vanishing Signal-to-Noise Ratio (SNR)." When a model cannot even satisfy basic syntactic constraints, any sporadic success in semantic reasoning introduces high-variance gradients that confuse the policy update, effectively stalling the model's progress on format mastery.

Proposition 3.4 (Auto-Curriculum and Variance Reduction)**.** *The hierarchical advantage $A_i = \frac{R(T_i) - \mu_{tier}}{\sigma_{tier}}$ induces a dynamic auto-curriculum. By centering rewards relative to a "competency-matched" peer group, ICOR ensures that the gradient $\nabla_\theta \mathcal{J}$ is always dominated by the current learning frontier.*

Proposition 3.4 establishes that hierarchical filtering functions as a powerful variance reduction technique. It ensures that the model only begins optimizing for fine-grained semantic reasoning after it has reached a stable competency plateau in syntax and topological navigation. This tiered decomposition minimizes the KL-divergence $\mathbb{D}_{\text{KL}}(\pi_\theta || \pi_{\text{ref}})$ required to reach the optimal policy region, providing a more stable optimization signal in high-dimensional combinatorial spaces. We provide an in-depth analysis of the variance decomposition and the application of the Law of Total Variance in Appendix A.3.2.

4 Experiments

4.1 Setup

Datasets. Following previous work (Luo et al., 2024), we evaluate ICOR on two standard Freebase-based KGQA benchmarks, WebQuestionSP (WebQSP) (Yih et al., 2016) and Complex WebQuestions (CWQ) (Talmor and Berant, 2018), with Freebase (Bollacker et al., 2008) as the background knowledge graph. To further examine cross-KG generalization, we additionally conduct experiments on MetaQA-3hop (Zhang et al., 2018), which is built on the WikiMovies knowledge graph.

Implementation details. ICOR consists of a relation retriever, an SFT-initialized policy model, and a GRPO-based self-evolving training stage. For semantic action-space pruning, we instantiate the

Question: what religions are popular in france? **Answer:** Catholicism

Generation

Prompt. Generate a valid relation path for answering the question.

Generated retrieval plan.

`location.statistical_region.religion`
→ `location.religion_percentage.religion`

Prediction.

`None`

The free-form path uses a schema-mismatched relation and fails to retrieve the answer.

Combination under ICOR

Prompt. Select relations from the candidate set and combine them into a valid path.

Candidate relation set.

`location.statistical_region.religions`
`location.location.contains` ···

Composed retrieval plan.

`location.statistical_region.religions`
→ `location.religion_percentage.religion`

Prediction.

`Catholicism`

Figure 2: Faithful Retrieval under ICOR.

retriever E_ϕ with Qwen3-Embedding-0.6B and train it with the contrastive objective in Eq. 4. At inference time, the top-K retrieved relations are used to construct the query-specific candidate relation set $\mathcal{C}_q$, with $K = 30$ by default. For the policy model, we use Qwen2.5-7B-Instruct as the backbone and initialize it through SFT on a mixture of RoG relation-path annotations and general instruction data. We then optimize the policy with GRPO under the proposed staged feedback mechanism, using a maximum interaction horizon of $T = 4$ and group size $G = 8$ for rollout sampling. More implementation details and hyperparameter configurations are provided in Appendix A.4.

Evaluation metrics. Following previous work (Luo et al., 2024), we take Hits@1 and F1 score as the metrics. Hits@1 measures whether the top-ranked predicted answer matches any ground-truth answer. F1 evaluates the overlap between the predicted set and the ground-truth answer set.

Baselines. To ensure a thorough comparison, we compare our method with the following four distinct types of methods: non-LLMs methods, which directly leverages knowledge from KGs to answer user questions; prompting-LLMs only methods, which directly utilizes LLMs to answer user questions; prompting-LLMs + KG methods, which retrieves knowledge from KGs by prompting LLMs to answer user questions; and finetuning-LLMs + KG methods, which combines fine-tuned LLMs with KGs to generate responses. For non-LLMs methods, we include Rigel (Sen et al., 2021), TIARA (Shu et al., 2022), UniK-QA (Oguz et al., 2020) and UniKGQA (Jiang et al., 2023). For prompting-LLMs only methods, We employ three prompting techniques on GPT-4: zero-shot, few-shot (Brown et al., 2020), and CoT prompting (Wei et al., 2022). For prompting-LLMs + KG methods, we include ToG (Sun et al., 2024), InteractiveKBQA (Xiong et al., 2024), SubgraphRAG (Li et al., 2024) and PoG (Chen et al., 2024a). For finetuning-LLMs + KG methods, we include NSM (He et al., 2021), KD-CoT (Wang et al., 2023), DECAF (Yu et al., 2023), RoG (Luo et al., 2024) and GNN-RAG (Mavromatis and Karypis, 2025). We additionally reproduce KBQA-R1 (Sun et al., 2026) with the same Qwen2.5-7B-Instruct policy backbone as ICOR.

Table 2: Efficiency comparison.

Method	LLM Calls	time (s)	Hits@1
ToG	16.3	63.9	81.8
RoG	2.0	1.5	83.2
ICOR	3.6	4.1	88.8

Table 3: ICOR with different retriever.

Retriever	Recall	Hits@1	F1
BERT	89.3	87.2	73.4
T5	88.4	86.7	72.1
Qwen3 Embedding	93.5	88.8	75.4
Perfect	100.0	89.6	76.9

4.2 Main Results

Comparison with other baselines. As shown in Table 1, we compare our method with other baselines on WebQSP and CWQ. Our approach achieves strong Hits@1 on WebQSP. On CWQ, our method demonstrates comparable performance to that achieved using GPT-4. Notably, our approach achieves performance comparable to powerful commercial models using only a fine-tuned Qwen2.5 7B model. This demonstrates the effectiveness of small models on KGQA. Furthermore, our method doesn't require frequent invocations of LLMs to interact with KGs, resulting in higher efficiency. As shown in Table 2, we compare the average number of LLM calls and the time required to answer

one question on WebQSP. The results reveal that the learned interaction policy of ICOR requires substantially fewer LLM calls than step-by-step prompting methods such as ToG.

We compare ICOR and KBQA-R1 using the same Qwen2.5-7B-Instruct policy backbone, retaining their respective warm-start data, action representations, and grounding mechanisms. ICOR achieves higher WebQSP Hits@1 (88.8 vs. 86.9), while our KBQA-R1 reproduction achieves higher F1 (80.2 vs. 75.4).

Faithful Retrieval under ICOR. As shown in Figure 2, we present an empirical case study from the CWQ dataset to illustrate the performance of different retrieval plan generation approaches. Both the directly generated retrieval plan and the plan derived from the model's exploration of relational pathways demonstrate a robust understanding of the user query, yielding conceptually sound retrieval strategies. However, the directly generated plan exhibits minor structural inconsistencies with the knowledge graph, rendering it impractical and impeding effective knowledge retrieval. In contrast, the structured reasoning approach employed by ICOR ensures that the generated retrieval plans align precisely with the knowledge graph's structure, thereby facilitating successful and efficient knowledge retrieval.

4.3 Ablation study

Analysis of the retriever. In ICOR, the retriever constructs the query-specific candidate relation set before policy rollout. We evaluate different retriever backbones, including BERT, T5, and Qwen3-Embedding, and also report an oracle setting where all relevant relations are included. As shown in Table 3, stronger retrieval recall generally leads to better downstream KGQA performance, and the oracle setting indicates that further improvements are possible with more accurate relation retrieval.

The effect of beam size. As shown in Figure 3, we analyze the effect of beam size on ICOR. Beam size is a critical parameter in the beam search decoding strategy employed by large language models to generate responses. It determines the number of candidate sequences retained at each decoding step, balancing exploration of diverse outputs with computational efficiency. To investigate its impact on the performance of ICOR, we vary the beam size from 1 to 7 and measure the Hits@1 and F1 score. The results indicate that the F1 stabilizes when the beam size reaches 2, while the Hits@1 increases slightly with increasing beam size. To balance computational cost and performance, we set the beam size to 5 in our experiments.

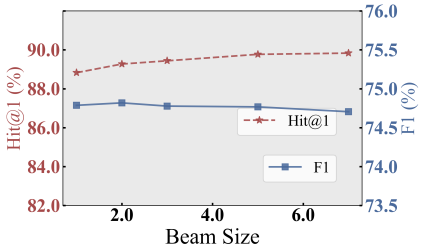

Figure 3: The effect of beam size.

Backbone generality. To examine whether ICOR generalizes across different policy backbones, we evaluate it with Qwen2.5-7B-Instruct, Llama-3-8B-Instruct, and Qwen2.5-3B-Instruct. As shown in Table 4, ICOR achieves comparable performance with Qwen2.5-7B-Instruct and Llama-3-8B-Instruct, suggesting that the proposed constrained rollout optimization is not tied to a specific model family. The substantial drop on Qwen2.5-3B-Instruct indicates that ICOR still requires a sufficiently capable base policy to reliably compose multi-hop relation paths.

Table 4: Backbone generality of ICOR on CWQ.

Backbone	Hits@1	F1
Qwen2.5-7B	69.5	61.3
Llama-3-8B	68.6	60.4
Qwen2.5-3B	33.4	24.3

Component ablations. Table 5 compares the main ICOR components and reward variants on CWQ. Semantic action-space pruning improves the SFT model from 61.4/56.2 to 65.9/57.8 in Hits@1/F1. Flat GRPO further improves performance to 66.7/59.1, and staged GRPO reaches 69.5/61.3, adding 2.8/2.2 points over flat GRPO. Removing any of the three rewards from full ICOR lowers performance, with larger drops from removing R_{path} or R_{ans} than R_{syn}. Experimental settings are provided in Appendix A.10.

Table 5: Component ablations on CWQ. The upper block adds semantic pruning and GRPO to SFT; the lower block removes one reward from full ICOR.

Configuration	Hits@1	F1
SFT	61.4	56.2
SFT + Semantic pruning	65.9	57.8
SFT + Semantic pruning + Flat GRPO	66.7	59.1
Full ICOR (staged GRPO)	**69.5**	**61.3**
Full ICOR w/o R_{syn}	67.8	59.5
Full ICOR w/o R_{path}	65.2	57.4
Full ICOR w/o R_{ans}	63.1	56.3

5 Conclusion

In this paper, we present ICOR, a constrained on-policy reinforcement learning framework for multi-turn KG reasoning with local LLMs. Rather than treating KG constraints only as inference-time validity rules, ICOR uses them to construct executable rollout spaces for GRPO-based policy optimization. By combining query-conditioned relation pruning, SFT cold start, and hierarchical reward filtering, ICOR improves the learnability of graph execution feedback and achieves a favorable accuracy-efficiency trade-off on KGQA benchmarks.

Acknowledgments and Disclosure of Funding

This work was supported in part by the Natural Science Foundation of China under Grant 62571507. JN and BH were supported by NSFC Major Research Plan No. 92570109, Guangdong Basic and Applied Basic Research Foundation No. 2024A1515012399, RGC Research Fellow Scheme No. RFS2627-2S02, RGC Young Collaborative Research Grant No. C2005-24Y, RGC General Research Fund No. 12202026, RGC General Research Fund No. 12200725, NSFC General Program No. 62376235, Guangdong and Hong Kong Universities "1+1+1" Joint Research Collaboration Scheme, HKBU Faculty Niche Research Areas No. RC-FNRA-IG/25-26/SCI/05, and HKBU CSD Departmental Incentive Scheme. TLL was partially supported by the following Australian Research Council projects: DP260102466, FT220100318, DP220102121, LP220100527, LP220200949.

Competing interests: The authors declare no competing interests.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bollacker, K. D., Evans, C., Paritosh, P. K., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems,NeurIPS*.

Chen, L., Tong, P., Jin, Z., Sun, Y., Ye, J., and Xiong, H. (2024a). Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. *Advances in Neural Information Processing Systems,NeurIPS*.

Chen, Z., Liu, K., Wang, Q., Zhang, W., Liu, J., Lin, D., Chen, K., and Zhao, F. (2024b). Agent-flan: Designing data and methods of effective agent tuning for large language models. *arXiv preprint arXiv:2403.12881*.

Fu, D., He, K., Wang, Y., Hong, W., Gongque, Z., Zeng, W., Wang, W., Wang, J., Cai, X., and Xu, W. (2025). Agentrefine: Enhancing agent generalization through refinement tuning. *arXiv preprint arXiv:2501.01702*.

He, G., Lan, Y., Jiang, J., Zhao, W. X., and Wen, J. (2021). Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel*.

Jiang, J., Zhou, K., Zhao, X., and Wen, J. (2023). Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph. In *The Eleventh International Conference on Learning Representations, ICLR*.

Jin, B., Zeng, H., Yue, Z., Yoon, J., Arik, S., Wang, D., Zamani, H., and Han, J. (2025). Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Li, J., Hui, B., Qu, G., Yang, J., Li, B., Li, B., Wang, B., Qin, B., Geng, R., Huo, N., Zhou, X., Ma, C., Li, G., Chang, K., Huang, F., Cheng, R., and Li, Y. (2023). Can LLM already serve as a database interface? a BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems*, volume 36.

Li, M., Miao, S., and Li, P. (2024). Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *arXiv preprint arXiv:2410.20724*.

Luo, H., E, H., Chen, G., Lin, Q., Guo, Y., Xu, F., Kuang, Z., Song, M., Wu, X., Zhu, Y., and Luu, A. T. (2026). Graph-R1: Towards agentic GraphRAG framework via end-to-end reinforcement learning. In *International Conference on Machine Learning*.

Luo, H., E, H., Guo, Y., Lin, Q., Wu, X., Mu, X., Liu, W., Song, M., Zhu, Y., and Luu, A. T. (2025). KBQA-o1: Agentic knowledge base question answering with monte carlo tree search. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 41177–41199. PMLR.

Luo, L., Li, Y., Haffari, G., and Pan, S. (2024). Reasoning on graphs: Faithful and interpretable large language model reasoning. In *International Conference on Learning Representations, ICLR*.

Mai, X., Xu, H., Li, Z.-Z., Wang, W., Hu, J., Zhang, Y., Zhang, W., et al. (2025). Agent rl scaling law: Agent rl with spontaneous code execution for mathematical problem solving. *arXiv preprint arXiv:2505.07773*.

Mavromatis, C. and Karypis, G. (2025). GNN-RAG: graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics, ACL*.

Oguz, B., Chen, X., Karpukhin, V., Peshterliev, S., Okhonko, D., Schlichtkrull, M., Gupta, S., Mehdad, Y., and Yih, S. (2020). Unik-qa: Unified representations of structured and unstructured knowledge for open-domain question answering. *arXiv preprint arXiv:2012.14610*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems,NeurIPS*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sen, P., Oliya, A., and Saffari, A. (2021). Expanding end-to-end question answering on differentiable knowledge graphs with intersection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Shu, Y., Yu, Z., Li, Y., Karlsson, B. F., Ma, T., Qu, Y., and Lin, C.-Y. (2022). Tiara: Multi-grained retrieval for robust question answering over large knowledge bases. *arXiv preprint arXiv:2210.12925*.

Sun, H., Bedrax-Weiss, T., and Cohen, W. (2019). Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*.

Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., and Cohen, W. W. (2018). Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Sun, J., Xu, C., Tang, L., Wang, S., Lin, C., Gong, Y., Ni, L. M., Shum, H., and Guo, J. (2024). Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations, ICLR*.

Sun, X., Chen, Z., Zheng, X., Liu, Q., Wu, S., Song, B., Wang, Z., Wang, W., and Wang, L. (2026). KBQA-R1: Reinforcing large language models for knowledge base question answering. In *International Conference on Machine Learning*.

Talmor, A. and Berant, J. (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, NAACL-HLT 2018,*.

Wang, K., Duan, F., Wang, S., Li, P., Xian, Y., Yin, C., Rong, W., and Xiong, Z. (2023). Knowledge-driven cot: Exploring faithful reasoning in llms for knowledge-intensive question answering. *arXiv preprint arXiv:2308.13259*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems, NeurIPS*.

Xiong, G., Bao, J., and Zhao, W. (2024). Interactive-kbqa: Multi-turn interactions for knowledge base question answering with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*.

Yih, W., Richardson, M., Meek, C., Chang, M., and Suh, J. (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, ACL*.

Yu, D., Zhang, S., Ng, P., Zhu, H., Li, A. H., Wang, J., Hu, Y., Wang, W. Y., Wang, Z., and Xiang, B. (2023). Decaf: Joint decoding of answers and logical forms for question answering over knowledge bases. In *International Conference on Learning Representations, ICLR*.

Zhang, G., Geng, H., Yu, X., Yin, Z., Zhang, Z., Tan, Z., Zhou, H., Li, Z., Xue, X., Li, Y., et al. (2025). The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*.

Zhang, Y., Dai, H., Kozareva, Z., Smola, A., and Song, L. (2018). Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence, AAAI*.

A Appendix

A.1 Limitations

- **Graph Incompleteness:** ICOR relies on the connectivity of the background knowledge graph. In scenarios where the KG contains significant missing triples or erroneous schemas, the policy's ability to navigate valid paths may be compromised.
- **Training Overhead:** Although GRPO is more efficient than PPO by eliminating the critic network, the multi-phase training (SFT + 3-phase RL) still requires substantial computational resources compared to simple prompting methods.

A.2 Social Impact

ICOR seeks to improve knowledge graph reasoning with open-source LLMs by grounding policy exploration in graph-derived relation spaces. A potential positive impact is that it reduces the dependence on proprietary LLMs for interactive KGQA, enabling more cost-efficient and locally deployable reasoning systems. By constraining rollouts to relation paths derived from the KG, ICOR may also improve the structural faithfulness and interpretability of model reasoning, since generated answers can be associated with explicit graph paths. However, the reliability of ICOR is still bounded by the quality of the underlying knowledge graph and the accuracy of relation retrieval. Incomplete, outdated, or biased graph content may lead to incorrect or biased answers. Therefore, ICOR should not be used as a standalone decision-making system in high-stakes domains without human oversight. Practical deployments should include mechanisms for provenance tracking, uncertainty awareness, and inspection of the retrieved paths and supporting graph evidence. In addition, more efficient KGQA agents may be misused for large-scale querying or information extraction over sensitive knowledge bases. Appropriate access control, logging, and privacy-preserving data governance are therefore important for responsible deployment. Overall, ICOR has the potential to support more transparent and efficient graph-grounded reasoning, while requiring careful consideration of reliability, fairness, and privacy risks.

A.3 Theoretical Derivations

In this appendix, we provide formal proofs for the search efficiency of the constrained action space and the variance reduction properties of the hierarchical filtering mechanism.

A.3.1 Proof of Theorem 3.2: Probability Mass Reallocation

Theorem 3.2. *For any target reasoning trajectory $\tau^* = (a_1, \ldots, a_T)$ where $a_t \in \mathcal{C}_q$, the ICOR policy π_θ satisfies $P_{ICOR}(\tau^*) > P_{gen}(\tau^*)$, provided the LLM assigns non-zero probability to the complement set $\mathcal{V} \setminus \mathcal{C}_q$.*

Proof. Consider a state s_t at step t. Let the raw distribution of the LLM over the full vocabulary $\mathcal{V}$ be $\pi_{\text{gen}}(a|s_t)$. The ICOR policy π_θ is defined by the restricted distribution over the semantic subspace $\mathcal{C}_q \subset \mathcal{V}$:

$$\pi_\theta(a_t|s_t) = \frac{\pi_{\text{gen}}(a_t|s_t)}{Z(s_t, \mathcal{C}_q)}, \quad \text{where } Z(s_t, \mathcal{C}_q) = \sum_{a' \in \mathcal{C}_q} \pi_{\text{gen}}(a'|s_t). \tag{15}$$

Here, $Z(s_t, \mathcal{C}_q)$ is the local partition function (or the "valid probability mass"). Since LLMs are prone to generating hallucinated relations or invalid syntax, there exists $a \in \mathcal{V} \setminus \mathcal{C}_q$ such that $\pi_{\text{gen}}(a|s_t) > 0$. Given that $\sum_{a \in \mathcal{V}} \pi_{\text{gen}}(a|s_t) = 1$, it follows that:

$$0 < Z(s_t, \mathcal{C}_q) = 1 - \sum_{a \in \mathcal{V} \setminus \mathcal{C}_q} \pi_{\text{gen}}(a|s_t) < 1. \tag{16}$$

The probability of a complete trajectory τ^* under the unconstrained generative model is $P_{\text{gen}}(\tau^*) = \prod_{t=1}^{T} \pi_{\text{gen}}(a_t|s_t)$. Under ICOR, the trajectory probability is:

$$P_{\text{ICOR}}(\tau^*) = \prod_{t=1}^{T} \frac{\pi_{\text{gen}}(a_t|s_t)}{Z(s_t, \mathcal{C}_q)} = P_{\text{gen}}(\tau^*) \cdot \prod_{t=1}^{T} \frac{1}{Z(s_t, \mathcal{C}_q)}. \tag{17}$$

Let $\eta = \prod_{t=1}^{T} Z(s_t, \mathcal{C}_q)^{-1}$. Since $Z(s_t, \mathcal{C}_q) \in (0, 1)$, then $\eta > 1$. Thus, we obtain:

$$P_{\text{ICOR}}(\tau^*) = \eta \cdot P_{\text{gen}}(\tau^*) > P_{\text{gen}}(\tau^*). \tag{18}$$

This confirms that the pruning strategy effectively concentrates the probability mass on valid graph-grounded paths, exponentially reducing the search entropy as the horizon T increases. □

A.3.2 Variance Reduction via Hierarchical Filtering

We now analyze why the hierarchical advantage A_i is superior to a flat reward sum $R_{\text{total}} = \sum R_k$.

The Signal Interference Problem. In standard GRPO, the gradient is $\nabla \mathcal{J} \approx \mathbb{E}[A_i \nabla \log \pi(T_i)]$. If a flat reward is used, the variance of the gradient is proportional to the variance of the total reward: $\text{Var}(\nabla \mathcal{J}) \propto \text{Var}(R_{\text{syn}} + R_{\text{topo}} + R_{\text{sem}})$. In early training, R_{sem} is essentially noise because the model has not yet mastered R_{syn}. This results in a low Signal-to-Noise Ratio (SNR), where the gradient for semantic reasoning is "washed out" by the high variance of basic formatting failures.

Derivation of Tiered Advantage. Let $\mathcal{T}_k$ be the set of trajectories satisfying the k-th tier of the hierarchy. ICOR computes:

$$A_i^{(k)} = \frac{R(T_i) - \mu_{\mathcal{T}_k}}{\sigma_{\mathcal{T}_k} + \epsilon}, \quad T_i \in \mathcal{T}_k. \tag{19}$$

By conditioning the advantage on the highest reached tier, we effectively perform a **Variance Decomposition**. Using the Law of Total Variance:

$$\text{Var}(R) = \mathbb{E}[\text{Var}(R|\mathcal{T})] + \text{Var}(\mathbb{E}[R|\mathcal{T}]). \tag{20}$$

The term $\text{Var}(\mathbb{E}[R|\mathcal{T}])$ represents the variance across tiers (e.g., the gap between syntax-fail and semantic-success). In flat RL, this term dominates the gradient, causing the model to optimize only for the coarsest signal.

ICOR's filtering eliminates this term by centering rewards within each tier ($\mu_{\mathcal{T}_k}$). Consequently, the policy π_θ only receives gradients that distinguish the *relative quality* of trajectories within the same competency level.

Proposition A.1 (Conditional Variance Reduction)**.** *The hierarchical mechanism ensures that the gradient $\nabla_\theta \mathcal{J}$ is more focused on the current learning frontier under the tier-conditioned reward normalization. When π_θ saturates $\mathcal{T}_{syn}$, $\sigma_{\mathcal{T}_{syn}} \to 0$, and the optimization automatically focuses on the next available signal R_{topo} without manual curriculum tuning.*

A.4 Implementation Details

Datasets and knowledge graphs. For WebQSP and CWQ, we follow prior KGQA settings and use Freebase as the background knowledge graph. For MetaQA, we use the WikiMovies knowledge graph to evaluate the transferability of ICOR across different graph schemas. In all datasets, the model interacts with the KG by generating relation paths, executing them over the graph, and producing final answers based on the retrieved entities.

Construction of training paths. For each training question, we construct $\mathcal{P}_q^\star$ by searching all executable relation paths connecting the linked topic entities to any ground-truth answer entity within the dataset-specific hop limit: two hops for WebQSP, four for CWQ, and three for MetaQA-3hop. We validate the paths through KG execution and remove duplicates, retaining multiple valid paths when they exist. These paths are used during training and are not required at inference time. When path supervision is unavailable during RL, ICOR can use syntax and answer rewards, as evaluated in the ablation without R_{path} in Table 5.

Semantic action-space pruning. We instantiate the retriever E_ϕ with Qwen3-Embedding-0.6B. The retriever is trained as a dual encoder that maps questions and KG relations into a shared embedding space. Positive relations are obtained from annotated logical forms or gold relation paths, while negative relations are sampled from the remaining relation schema. The retriever is optimized with the contrastive objective in Eq. 4 for 10 epochs, using a learning rate of 2×10^{-5} and a batch size of 128. At inference time, we retrieve the top-K relations according to the similarity score $s(q, r)$ and use them to construct the query-specific candidate relation set $\mathcal{C}_q$. We set $K = 30$ by default.

Policy model and SFT cold start. We use Qwen2.5-7B-Instruct as the backbone policy model π_θ. Before reinforcement learning, the policy is initialized with SFT to acquire basic relation-path generation and instruction-following abilities. The SFT data consist of relation-path annotations from RoG and general instruction data from Alpaca, mixed at an 8:2 ratio. The SFT stage is trained for 3 epochs with a learning rate of 2×10^{-5} and a global batch size of 128.

GRPO training. Starting from the SFT-initialized policy, we further optimize the model using GRPO under the proposed staged feedback mechanism. The maximum interaction horizon is set to $T = 4$. For each query, the policy samples a group of $G = 8$ trajectories to compute the group-normalized relative advantage. The rollout sampling temperature is set to 0.7, and the maximum sequence length is set to 8,192 tokens. The initial learning rate for RL training is 1×10^{-5}, and the KL coefficient is set to $\beta = 0.05$ to prevent the policy from drifting excessively from the reference policy.

The training process consists of three stages. The first stage runs for 500 steps and optimizes both syntax and path rewards over the full sampled group. The second stage runs for 200 steps and optimizes the path reward only over syntactically valid trajectories. The third stage runs for 200 steps and optimizes answer-level feedback only over trajectories whose generated paths match the correct path set.

Reward instantiation. In the implementation, the syntax reward assigns a positive score to trajectories that follow the required interaction format and a small penalty to malformed trajectories. Specifically, we use $+0.2$ for syntactically valid outputs and -0.03 otherwise. The path reward assigns $+0.7$ when the generated relation path belongs to the correct path set, and 0 otherwise. The answer reward is computed as the F1 score between the predicted answer set and the ground-truth answer set, with a maximum value of 1.0. The stage-wise reward used in each phase follows Eq. 10. All experiments were conducted on four NVIDIA H20 GPUs, each with 96 GB of memory.

Inference. During inference, the retriever first constructs the candidate relation set $\mathcal{C}_q$ with $K = 30$. The policy then generates relation-path actions conditioned on the question and the retrieved relation set. Each generated path is executed on the KG, and the resulting entities are used to produce the final answer. The interaction terminates when the model outputs a final answer or reaches the maximum horizon $T = 4$.

A.5 Details of dataset

In this paper, we conduct experiments on two widely used KGQA datasets: WebQuestionSP (WebQSP) and Complex WebQuestions (CWQ). To ensure a fair comparison, the training and testing splits are the same as in previous works (Luo et al., 2024; Sun et al., 2018), as shown in Table 6. More detailed statistical results for WebQSP and CWQ are provided in Tables 7 and 8.

Table 6: Split of training and test sets.

Datasets	#Train	#Test	Max #hop
WebQSP	2,826	1,628	2
CWQ	27,639	3,531	4

Table 7: Statistics of the number of answers to questions in WebQSP and CWQ.

Dataset	#Ans = 1	$2 \leq$ #Ans ≤ 4	$5 \leq$ #Ans ≤ 9	#Ans ≥ 10
WebQSP	51.2%	27.4%	8.3%	12.1%
CWQ	70.6%	19.4%	6%	4%

A.6 Sensitivity to candidate relation size.

The candidate relation size K controls the trade-off between relation recall and combinatorial search complexity.

Table 8: Statistics of the answer hops in WebQSP and CWQ.

Dataset	1 hop	2 hop	≥ 3 hop
WebQSP	65.49%	34.51%	0.00%
CWQ	40.91%	38.34%	20.75%

Table 9: Sensitivity to K on CWQ.

K	Hits@1	F1
20	68.3	59.4
30	69.5	61.3
50	64.2	58.7

As shown in Table 9, $K = 30$ achieves the best performance. A smaller K may exclude necessary relations from the candidate set, while a larger K introduces more distracting relations and enlarges the combinatorial rollout space. This result supports the use of a moderately sized relation subspace for constrained policy exploration.

A.7 Effect of SFT Data Mixture

In the SFT cold-start stage, we mix relation-path annotations from RoG with general instruction data from Alpaca. The default mixture ratio is set to 8:2, which prioritizes relation-path learning while preserving the instruction-following ability of the backbone model. To examine whether ICOR is sensitive to this data mixture, we additionally evaluate a 5:5 mixture ratio on CWQ.

Table 10: Effect of the SFT data mixture on CWQ.

SFT mixture	**Hits@1**	**F1**
RoG:Alpaca = 8:2	69.5	61.3
RoG:Alpaca = 5:5	68.3	60.6

As shown in Table 10, the 8:2 mixture achieves slightly better performance than the 5:5 mixture. This suggests that assigning a larger proportion to relation-path annotations provides a stronger initialization for KG reasoning. Meanwhile, the relatively small performance gap indicates that ICOR is not highly sensitive to this mixture ratio after GRPO-based policy optimization.

A.8 Cross-KG Generalization on MetaQA

To further evaluate the generalization ability of ICOR beyond Freebase-based benchmarks, we conduct additional experiments on MetaQA, which is built on the WikiMovies knowledge graph. This setting differs from WebQSP and CWQ in both graph schema and question distribution, and therefore provides a complementary testbed for cross-KG reasoning.

We first evaluate the zero-shot transfer ability of ICOR by applying the Freebase-trained model directly to MetaQA-3hop without additional fine-tuning on the target KG. The results are shown in Table 11.

For this zero-shot setting, neither the retriever nor the policy is retrained. The top-3 retrieved MetaQA relations retain at least one complete executable gold path for 98.28% of questions. This is question-level complete-path coverage, whereas the recall reported in Table 3 measures retrieval of the relations required by executable gold paths.

Table 11: Zero-shot transfer of ICOR on MetaQA-3hop.

Method	**Training KG**	**Test KG**	**Hits@1 / F1**
ICOR	Freebase	WikiMovies	52.7 / 47.8

The zero-shot result indicates that ICOR can transfer part of its graph-interaction ability to an unseen KG schema. This is because the policy composes reasoning paths from relations retrieved from the target graph, rather than generating relation paths entirely from memorized Freebase schemas.

For a fair in-domain comparison, we further fine-tune different methods on MetaQA-3hop and report the results in Table 12. Under this setting, ICOR achieves stronger performance than the compared fine-tuning-based KGQA methods.

Table 12: Fine-tuned performance on MetaQA-3hop.

Method	Hits@1	F1
RoG	89.0	50.7
GNN-RAG	93.3	62.9
ICOR	95.1	69.3

These results show that ICOR remains effective on a KG schema different from Freebase. Nevertheless, MetaQA is still a relatively structured benchmark, and evaluating ICOR on larger and more heterogeneous KGs such as Wikidata remains an important direction for future work.

A.9 Details of Prompts

In the ICOR framework, we employ a **Unified State-Machine Prompting** strategy. Unlike traditional multi-stage pipelines that use different prompts for varied tasks, ICOR provides the full tool definitions, behavioral logic, and execution constraints to the model within the initial system context. This enables the agent to autonomously manage the reasoning process through a structured "State-Machine" logic.

1. Integrated System Prompt Structure. The initial prompt delivered to the LLM is composed of three functional blocks: (1) **Tool Specifications**, which define the API for graph traversal; (2) **Role and Logic Definitions**, which establish the expert persona and the thinking-before-action discipline; and (3) **State Constraints**, which strictly define the transition between Exploration and Answer modes.

The core of this unified prompt is the tool signature and the state-machine instruction:

Unified ICOR System Prompt (Abridged)

 # Tools: You may call `search_rule_path` with `rule_path` and `question_entities`...
Role: You are a Knowledge Graph Question Answering Expert...
State-Machine Logic: You are allowed to output ONLY one of the following two formats in any single response:

- `<think>...</think> <tool_call>...</tool_call>` (Exploration Mode)
- `<think>...</think> <answer>...</answer>` (Answer Mode)

Rules: Select relations ONLY from the provided list. Copy entities EXACTLY from search results.

2. Structural Reasoning via XML Tags. To facilitate automated interaction and reinforcement learning, ICOR utilizes XML-style tags to wrap specific output types. The `<think>` block serves as an internal scratchpad where the model analyzes the query and justifies its next action. This explicit reasoning trace allows the policy to resolve multi-hop dependencies before committing to a graph query.

3. Dynamic Context Injection. While the behavioral logic is fixed in the system prompt, the action space is dynamically grounded for each query. In the first interaction, the user message injects the specific **Question**, the **Available Relations** (pruned by E_ϕ), and a placeholder for **Graph Data**.

- **Exploration Mode Trigger**: If the search results are empty or further verification is needed, the model is instructed to iterate by proposing a new `rule_path` from the available relations.
- **Finality Constraint**: The prompt instructs the model to generate an answer after receiving valid path instances and to copy answer entities from the tool output.

By providing this comprehensive instruction set at the onset, ICOR ensures that the LLM maintains a coherent reasoning state across multi-turn interactions. This structured environment is critical for the GRPO algorithm to accurately calculate rewards based on the model's adherence to both syntactic formats and logical constraints.

4. Multi-turn Interaction Example. The policy input retains the question, linked topic entities, candidate relations, and preceding actions and KG observations. The initial prompt does not expose the underlying KG triples; the executor returns graph evidence after a tool call. The following abridged example preserves the action–feedback–revision sequence, omitting the full relation list and intermediate reasoning text for readability.

ICOR Interaction Example (Abridged)

Question. Where is the Howard University located, and that has a govermental body named D.C. Emancipation Commemoration Commission?
Topic entities. `["m.0217m9", "m.04khhd4"]`.
Relation identifiers used below.
r_a: `government.government_position_held.jurisdiction_of_office`
r_b: `government.government_position_held.governmental_body`
First action. Call `search_rule_path` with `rule_path` $= [r_a, r_b]$.
Execution feedback. `success: false`; `error_type: NoPathFound`. The executor reports that no reasoning path was found and suggests trying a different relation sequence.
Revised action. Call the same tool with `rule_path` $= [r_b, r_a]$, reversing the relation order.
Execution feedback. `success: true`; the returned path reaches Washington DC.
Final answer. The model copies the returned entity string: `["washington dc\uff0e"]` (JSON-escaped form).

This example illustrates that a relation sequence can be syntactically valid yet non-executable. ICOR uses the structured failure observation to revise its graph action before producing the final answer.

A.10 Component Ablation Settings

The flat-GRPO control in Table 5 uses the same SFT initialization, candidate-relation space, graph environment, rollout budget, and reward components as full ICOR. It aggregates the three rewards and normalizes advantages over the full rollout group without stage-specific filtering. The base model evaluated with direct answer generation and no KG interaction obtains 13.8 Hits@1 and 10.8 F1 on CWQ.

A.11 Additional Evaluations

Larger policy backbone. We additionally evaluate ICOR with Qwen2.5-14B-Instruct under the same experimental setting. Table 13 reports the results, extending the backbone comparison in Table 4. The larger policy improves performance on both WebQSP and CWQ relative to the 7B results in Table 1.

Table 13: Additional ICOR results with Qwen2.5-14B-Instruct.

Policy backbone	WebQSP		CWQ	
	Hits@1	F1	Hits@1	F1
Qwen2.5-14B-Instruct	90.5	76.3	71.4	62.7

Transfer of staged optimization to NL2SQL. We evaluate the prerequisite-aware staged optimization principle on BIRD (Li et al., 2023) using Qwen3-4B. Stage 1 optimizes SQL syntax and schema validity over all trajectories; stage 2 optimizes execution success over syntax- and schema-valid trajectories; stage 3 optimizes result correctness over executable trajectories. Under the same backbone and experimental protocol, staged GRPO increases execution accuracy from 59.9 to 62.4 (Table 14).

Table 14: Controlled comparison on BIRD with Qwen3-4B.

Optimization	Execution accuracy
Standard GRPO	59.9
Staged GRPO	62.4

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper provides a full set of assumptions and a complete proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully discloses all the information needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The source code and implementation instructions are available at `https://github.com/junnie00/ICOR`. The benchmark datasets are available from their original sources, which are cited in the paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides details of implementation in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the average results under three different random seeds and report the variance in Figure 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provide sufficient information on the computer resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper has discussed societal impacts in Appendix

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original paper that produced the code package and dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We communicate the details of the code as part of our submission via structured templates.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.