CoRE-RL: Co-evolving Reasoning Trajectories and Evidence Subgraphs with Learned Evidence Projection

Zhengquan Luo[1] **Guy Tadmor**[2] **Peilin Zhao**[3] **David Zeevi**[2] **Zhiqiang Xu**[1]✉
[1]Mohamed bin Zayed University of Artificial Intelligence, UAE
[2]Weizmann Institute of Science, Israel
[3]Shanghai Jiao Tong University, China
{zhengquan.luo, zhiqiang.xu}@mbzuai.ac.ae
{guy.tadmor, david.zeevi}@weizmann.ac.il
peilinzhao@hotmail.com

Abstract

Large language models have made knowledge-graph question answering more flexible, yet complex multi-hop questions often require evidence that becomes identifiable only during reasoning. Starting from the question alone, narrow retrieval can miss bridge facts, while broad expansion injects distractors and worsens long-context interference. The key bottleneck is dynamic evidence identification: the system must revise a compact evidence view as reasoning exposes unsupported facts. We formulate this process as partially observable control over evidence and propose **CoRE**, a residual-coupled framework in which reasoning trajectories and evidence subgraphs evolve together. CoRE decomposes each trace into answer-critical claims, treats unsupported claims as residual signals, and uses them to guide targeted retrieval and budgeted subgraph projection. The same formulation exposes a learnable projection decision: after residuals retrieve candidate facts, **CoRE-RL** uses offline contextual-bandit evidence projection to select the facts retained for the next reasoning pass. Across WebQuestionsSP and Complex WebQuestions, the CoRE family improves answer accuracy, evidence sufficiency, and budget robustness, with CoRE-RL providing further gains through more effective retained-subgraph selection.

1 Introduction

Combining large language models, or LLMs, with knowledge graphs, or KGs, has become a prominent direction for knowledge-intensive question answering and reasoning, spanning KG-grounded retrieval and prompting [1] and explicit graph-based reasoning [2]. By encoding facts as explicit triples that can be updated independently of model parameters, KGs enable access to long-tail and evolving knowledge [3, 4] while providing checkable evidence that mitigates unsupported hallucinations [5, 6, 7]. However, complex KGQA exposes a deeper mismatch than simply retrieving more facts before reasoning. The evidence needed for a multi-hop answer is often not fully specified by the surface question: a local neighborhood around query entities may miss bridge facts, while aggressively expanding the neighborhood introduces many irrelevant triples and long-context interference [8, 9]. Thus the central bottleneck is not merely evidence scarcity, but the timing of evidence identification: many useful facts become recognizable only after intermediate reasoning exposes which relation or entity is missing. For example, as illustrated on the left of Fig. 1, when asked *"Which city was the spouse of the director of the film Inception born in?"*, an initial subgraph centered on *Inception* contains no *spouse* information, leaving the intermediate question *"Who is the spouse of Christopher Nolan?"* unanswered until the spouse relation is retrieved. This raises a fundamental question for KGQA systems: how to handle the mutual dependence between reasoning and evidence, where

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

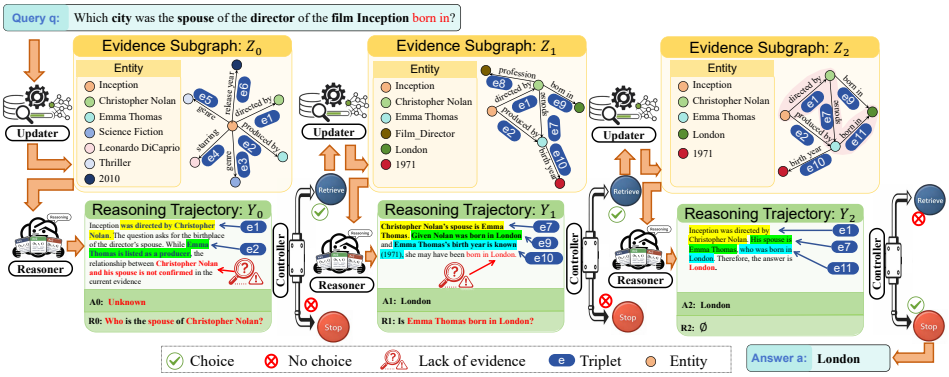


Figure 1: Overall architecture of the CoRE framework with a running example. In the first round, reasoning exposes a missing relation, the director's spouse, triggering retrieval to resolve a reasoning gap. In the second round, insufficient support for the spouse's birthplace prompts additional evidence retrieval to stabilize the answer.

reasoning reveals what evidence is missing, while incomplete evidence constrains how reasoning can proceed.

To make this mutual dependence precise, we formulate KGQA as a partially observable evidence-control problem. The full KG may be available in principle, but it is not directly usable as LLM input: the reasoner observes only a serialized evidence subgraph, and evidence that is too broad can be as harmful as evidence that is too sparse. Under this formulation, evidence retrieval and retention determine which reasoning trajectories are feasible, while unresolved or unsupported gaps along a reasoning trajectory specify what additional evidence is required. Reasoning and evidence therefore interact through explicit state transitions rather than a one-time preprocessing step. As illustrated in the middle of Fig. 1, let Y_t denote the reasoning trace at round t and Z_t the retained evidence subgraph at that round. Unresolved reasoning gaps surfaced by Y_0 trigger targeted evidence retrieval, expanding the evidence subgraph from Z_0 to Z_1, which enables the subsequent reasoning trace Y_1 to reach the correct answer "London". However, reaching a correct answer does not necessarily imply sufficient evidential support. In this example, the intermediate conclusion that Emma Thomas was born in London remains unsupported by explicit evidence, prompting an additional round of evidence updating that yields a refined subgraph Z_2 and fully grounds the final reasoning trajectory, as illustrated on the right of Fig. 1. The modeling and running example jointly reveal a tight, mutually reinforcing coupling between reasoning trajectory and evidence subgraph.

Under this modeling view, existing KG-LLM approaches largely fall into two paradigms. The first, *retrieve-then-reason*, retrieves triples, paths, or subgraphs before inference and then asks the LLM to reason over the resulting fixed evidence view [10, 11]. Because the evidence view is chosen before intermediate reasoning exists, it can neither discover bridge facts revealed later nor distinguish answer-critical facts from noisy large-neighborhood expansions. The second comprises *iterative/agent-based* methods that interleave reasoning with sequential KG access [6, 12, 2]. These methods are more adaptive, but they often frame evidence acquisition as local traversal or path search, which makes cross-region evidence aggregation difficult and becomes brittle when KG edges are missing [13, 14, 2]. Taken together, both paradigms lack an explicit interface that converts intermediate reasoning failures into evidence acquisition and retention decisions. Our analysis shows that KGQA needs a bidirectionally coupled loop: reasoning must revise evidence, and retained evidence must constrain subsequent reasoning.

This motivates **CoRE**, a closed-loop framework in which reasoning trajectories and evidence subgraphs are updated together. CoRE decomposes each reasoning trajectory into atomic claims, checks which answer-critical claims are unsupported by the currently retained evidence subgraph, and exposes these unsupported claims as residuals. The residuals provide a concrete interface for evidence control: they identify missing relations or constraints, guide targeted retrieval, and determine how the compact evidence subgraph is revised for the next reasoning pass.

The same interface also makes the evidence-projection step learnable, where projection denotes selecting which triples remain in the compact subgraph after retrieval. Once residual-guided retrieval

produces candidate facts, the system must rank them by their expected usefulness for subsequent reasoning under a compact context. Base CoRE uses a transparent residual-support score for this ranking. CoRE-RL learns this scorer through offline contextual-bandit evidence projection, using next-round feedback from logged executions and features of the question, current answer, residuals, and candidate evidence.

Our contributions are summarized as follows:

- **Problem diagnosis.** We identify dynamic evidence identification as a core bottleneck in KGQA: useful evidence is often revealed during reasoning rather than fully specified by the initial question.
- **Evidence-control framework.** We model KGQA as an evidence-control problem where retrieval and retention are driven by both the current evidence subgraph and unresolved reasoning residuals, leading to CoRE's residual-coupled loop and CoRE-RL's learned projection policy.
- **Empirical validation.** We validate the CoRE family through answer quality, verifiability, multi-hop behavior, budget robustness, cost trade-offs, residual contraction, claim-extraction quality, and learned-projection ablations.

2 Related Work

Retrieve-then-reason. These approaches differ primarily in how KG evidence is selected and represented for LLM reasoning. Early work incorporates KG structure into prompting or attribution, such as KG-CoT [15] and RD-P [16]. More recent retrieval-augmented generation, or RAG, methods retrieve subgraphs or structured contexts and encode them using learned representations, including SubgraphRAG, G-Retriever, and GNN-RAG [10, 17, 18]. These methods typically perform retrieval in a single or fixed number of rounds and pass the resulting evidence to the LLM as an explicit context.

Iterative and agent-based KG reasoning. A broad class of methods interleave reasoning with sequential access to KG evidence, either through explicit agent actions or multi-round retrieval and reasoning loops. Agent-based frameworks enable step-wise interaction with the KG via traversal, frontier expansion, or planning, as in StructGPT [12], ToG [6], PoG [19], Interactive-KBQA [20], and ReKnoS [2]. Closely related approaches perform multi-round retrieval and refinement, using intermediate model states or feedback signals to expand or adjust retrieved evidence, such as RwT [21] and RJE [22]. Some methods further incorporate verification or deliberation mechanisms during the reasoning process, including GCR [14] and deliberation-based approaches such as DP [23].

Overall, existing KGQA methods largely treat evidence as a static or explicitly provided context, while leaving the reasoning process implicit and decoupled from evidence acquisition and retention. In contrast, CoRE is derived from a decision-theoretic formulation that makes the coupling between reasoning and evidence explicit and necessary, and operationalizes this coupling through claim-level, residual-driven retrieval and retention; Appendix A provides a detailed comparison.

3 Preliminaries and Problem Setup

A knowledge graph, or KG, is modeled as a finite set of typed triples $G \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$, where $\mathcal{V}$ and $\mathcal{R}$ denote the sets of entities and relation types. Given a natural language question q, the KGQA task outputs an answer $a \in 2^{\mathcal{V}}$, which may be a single entity or a set of entities. In principle, the answer is fully determined by the underlying knowledge graph G and the query q.

One possible approach is to expose the entire knowledge graph G to the language model by serializing all triples into the input context. In practice, this is infeasible: beyond strict context budget limits, indiscriminate inclusion also exacerbates known long-context position biases such as lost-in-the-middle [8, 9]. As a result, the reasoner cannot directly access G and instead operates on a token sequence encoding only a subset of triples, which introduces explicit input-budget constraints and implicit evidence-utilization limitations [24].

Let $\phi(\cdot)$ denote a serialization function that maps any triple set $Z \subseteq G$ to a token sequence, and let $L(Z) = |\phi(Z)|$ be its length. Given a maximum input budget B, any inference call must

satisfy $L(Z) \leq B$ [25]. We use B for token length and K for the retained-triple budget enforced by projection; budget values such as 30, 60, or 100 therefore refer to K unless explicitly stated otherwise. We therefore model the evidence provided to the LLM as a *budgeted evidence subgraph* $Z \subseteq G$ [10, 17].

Conditioned on the question q and an evidence subgraph Z, the LLM is treated as generating a reasoning trajectory $y_{1:T} = (y_1, \ldots, y_T)$, where each y_t represents an intermediate reasoning artifact, such as a natural language step or a structured decision. The final prediction is obtained by a deterministic decoder $g(\cdot)$ as $\hat{a} = g(y_{1:T})$. When discussing multi-round evidence updates, we use Y_t for the complete reasoning trace produced in round t, while $y_{1:T}$ denotes the internal steps within one such trace. If the evidence subgraph Z is fixed prior to trajectory generation, this formulation reduces to the retrieve-then-reason paradigm [10, 26].

To model dynamically evolving evidence requirements, we introduce a latent state $x \in \mathcal{X}$ that captures unobserved factors induced by (G, q) and governing which facts are relevant at different reasoning stages. At step t, the system selects a retrieval or update action $u_t \in \mathcal{U}$ [26] and receives an observation O_{t+1}, representing a set of newly retrieved knowledge graph triples. Formally,

$$O_{t+1} \subseteq G, \qquad O_{t+1} \sim \Omega(\cdot \mid x, u_t, G), \tag{1}$$

where Ω denotes an abstract observation model associated with the retrieval operator.

The system maintains a *belief state*, i.e., a posterior distribution over the latent state x induced by past actions and observations,

$$b_t(\cdot) = p(\cdot \mid q, u_{1:t}, O_{1:t}), \tag{2}$$

together with a budgeted evidence state $Z_t \subseteq G$. Upon receiving a new observation O_{t+1}, the evidence is updated via a projection operator $\mathcal{P}_K(\cdot)$, where K is the maximum number of retained triples,

$$Z_{t+1} = \mathcal{P}_K(Z_t \cup O_{t+1}), \tag{3}$$

so that the serialized length remains within budget.

Taken together, the above components define a partially observable Markov decision process, or POMDP [27], for budgeted KGQA, in which the information state is summarized by (b_t, Z_t). Retrieval actions are selected based on the current information state, while reasoning trajectories are generated conditioned on the retained evidence subgraph.

4 Co-evolution of Reasoning and Evidence

When only a limited number of triples can be used, keeping one piece of evidence necessarily excludes others, which in turn restricts what additional evidence can be accessed later. As a result, current evidence selection directly affects future reasoning possibilities rather than serving as a purely local or myopic decision. The following theorem formalizes the minimal information an optimal policy must use.

Theorem 4.1 (Evidence-aware optimality)**.** *Assume a non-trivial projection operator $\mathcal{P}_K$, meaning that adding candidate triples can change which triples are retained under the budget. There exists an optimal policy π^* such that*

$$u_t \sim \pi^*(\cdot \mid b_t, Z_t). \tag{4}$$

Moreover, there exist KGQA instances for which any policy restricted to $u_t \sim \pi(\cdot \mid b_t)$ or $u_t \sim \pi(\cdot \mid Z_t)$ is strictly suboptimal, with proof in Appendix B.2.

Theorem 4.1 formalizes a crucial requirement: under a fixed evidence budget, an optimal policy must jointly condition on the belief state and the retained evidence. This requirement is violated by most existing KG+LLM paradigms. Fixed retrieve-then-reason pipelines commit to a static evidence subgraph [10]; traversal-based agents restrict retrieval to local neighborhoods [6, 2]; and iterative retrieval loops typically grow the evidence set in an add-only manner [12]. Despite their different mechanisms, all these approaches make retrieval or retention decisions based only on partial information, either the current belief state or the retained evidence. We therefore group them into a class of constrained policies, denoted by Π; see Appendix C.

To operationalize Theorem 4.1, we introduce residual-coupled policies, denoted by Π^*, that decompose an intermediate reasoning output y_t into a set of atomic claims $C(y_t)$. Each claim corresponds

to a concrete factual assertion required by the reasoning process. Given the current evidence subgraph Z_t, we define the residual claim set:

$$R_t = \{c \in C(y_t) \mid \mathrm{Sup}(c; Z_t) < \tau\}, \tag{5}$$

where $\mathrm{Sup}(c; Z_t) \in [0, 1]$ denotes the degree to which claim c is supported by triples in Z_t, and τ is the support threshold, with $\mathrm{Sup}(c; Z_t) \geq \tau$ indicating full support. The residual claim set R_t consists of claims that are not fully supported by the retained evidence and is used to guide evidence updating. Specifically, retrieval is conditioned on (q, R_t, Z_t) and budget enforcement is realized via a claim-conditioned projection that retains triples increasing $\mathrm{Sup}(c; Z)$ for claims in R_t, forming a closed-loop co-evolution between reasoning trajectories and evidence subgraphs.

At this point, we have two policy classes: the constrained policies Π and the residual-coupled policy class Π^*. To compare their effectiveness, we evaluate policies by the value they achieve, denoted by V^π, which measures overall task performance under the KGQA objective. Under a fixed evidence budget, the constrained policy class admits the following limitation:

$$\sup_{\pi \in \Pi} V^\pi \;<\; V^*, \tag{6}$$

where V^* denotes the optimal value, which can be attained or arbitrarily closely approximated by a residual-coupled policy Π^*, with proof in Appendix D.

Beyond effectiveness, we further establish conditions under which residual-driven coupling guarantees convergence to evidence-sufficient reasoning. For any evidence set S and residual set R, define the residual-mass functional

$$\Phi(S, R) = \sum_{c \in R} \big(1 - \mathrm{Sup}(c; S)\big), \qquad \Phi_t = \Phi(Z_t, R_t), \tag{7}$$

which aggregates the missing evidence induced by unresolved claims and serves as a potential function for evidence sufficiency. Intuitively, if retrieval can increase support for unresolved claims and budget projection preserves these gains, residual-driven updates make steady progress in reducing Φ_t, which can be formalized by the following sufficiency result.

Theorem 4.2 (Sufficiency of residual-driven coupling)**.** *Assume that for any (R_t, Z_t) with $\Phi_t > 0$, there exists an admissible retrieval action u_t whose returned candidate evidence is denoted by $\Delta Z_t(u_t)$, such that*

$$\mathbb{E}[\Phi(Z_t \cup \Delta Z_t(u_t), R_t) | \mathcal{H}_t] \leq (1 - \eta)\Phi_t \tag{8}$$

for some $\eta \in (0, 1)$, and that the budget projection $\mathcal{P}_K$ does not increase the residual mass,

$$\Phi(\mathcal{P}_K(Z_t \cup \Delta Z_t(u_t)), R_t) \leq \Phi(Z_t \cup \Delta Z_t(u_t), R_t). \tag{9}$$

Assume also that residual refresh does not increase expected post-projection mass,

$$\mathbb{E}[\Phi(Z_{t+1}, R_{t+1}) | \mathcal{H}_t] \leq \mathbb{E}[\Phi(Z_{t+1}, R_t) | \mathcal{H}_t], \tag{10}$$

where $\mathcal{H}_t$ is the interaction history.

Then there exists a residual-coupled policy under which $\mathbb{E}[\Phi_t] \leq (1 - \eta)^t \Phi_0$, and an ε-evidence-sufficient state is reached in $O(\log(\Phi_0/\varepsilon))$ iterations, with proof in Appendix F.

This is a conditional sufficiency result: it requires useful retrieval, gain-preserving projection, and controlled residual refresh. The implementation diagnostics in Appendix G measure residual-slot alignment and changes in logged unsupported-claim mass as observable counterparts of these conditions.

Taken together, Theorems 4.1 and 4.2 establish that, in the POMDP formulation of budgeted KGQA, the retained evidence subgraph and the reasoning trajectory are inherently coupled and co-evolve over time. The evidence retained at each step constrains which facts remain accessible for subsequent reasoning, while the evolving reasoning trajectory determines which evidence must be acquired and preserved to resolve remaining unsupported claims. Under the assumptions of Theorem 4.2, this bidirectional coupling is not only necessary but also sufficient to drive the system toward an evidence-sufficient state within a fixed budget.

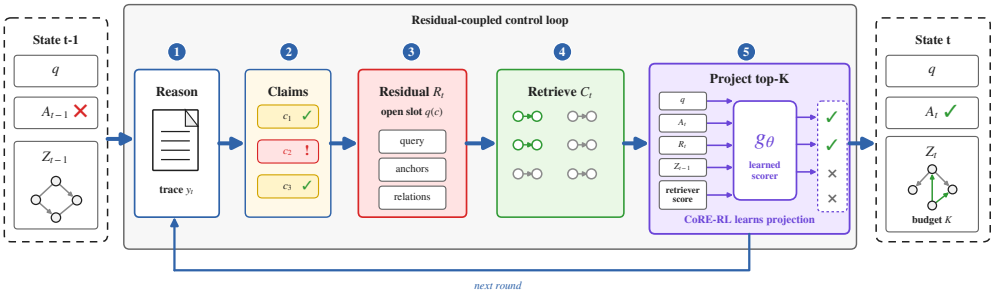


Figure 2: Residual-coupled update in the CoRE family. Reasoning over Z_t exposes unsupported answer-critical claims; these residuals guide retrieval and a budgeted projection step, implemented either by a transparent residual-support score in CoRE or a learned projection scorer in CoRE-RL.

5 CoRE-RL: Contextual-Bandit Evidence Projection

We instantiate this theory as a round-based evidence-control loop. At round t, the observable state is $s_t = (q, Z_t, \hat{a}_t, Y_t, R_t)$, where Z_t is the retained evidence memory, Y_t and $\hat{a}_t$ summarize the current reasoning trace and answer, and R_t contains unsupported answer-critical claims. Starting from this state, the loop performs four operations: reason over Z_t, extract residual claims, retrieve candidate evidence for those residuals, and project the enlarged candidate pool back to a compact Z_{t+1}. CoRE implements this loop with a transparent residual-support projection score. CoRE-RL keeps the same residual extraction, retrieval, and controller, but learns the projection scorer from rollout feedback so that the retained subgraph better preserves evidence useful for unresolved residuals. Figure 2 shows one round of the update, and the remainder of this section follows the same order.

Subgraph-constrained reasoning. At round t, the LLM receives question q and evidence subgraph Z_t, and samples N candidate outputs $\{\hat{y}_t^{(i)}\}_{i=1}^N$, where N is the candidate-sampling budget. Each output contains a proposed answer and a reasoning trace, which is normalized into atomic claims $C(\hat{y}_t^{(i)})$. For a claim c and retained edge e, we compute a deterministic semantic support score $s(c, T(e))$ between the claim and the serialized triple $T(e) = (h, r, o)$, with head h, relation r, and object o. The claim support is $\mathrm{Sup}(c; Z_t) = \max_{e \in Z_t} s(c, T(e))$. Candidates are ranked by a monotone support aggregator, i.e., a scoring rule whose value increases as stronger claim-level evidence support is observed, preferring well-grounded trajectories while exposing unsupported claims as residuals. The residual set is

$$R_t = \{c \in C(\hat{y}_t^\star) : \mathrm{Sup}(c; Z_t) < \tau \ \wedge \ \mathrm{Rel}(c; q, \hat{a}_t) = 1\}, \tag{11}$$

where Rel filters for answer-determining claims. Operationally, the answer-critical set consists of claims produced and normalized by the structured extractor; support is evaluated against the retained evidence Z_t.

Residual-conditioned retrieval. For each residual claim c, CoRE builds an open slot $\rho(c) = \langle \mathrm{qry}(c), A(c), \mathrm{hint}(c) \rangle$ containing a textual query, anchor entities, and relation/type hints. A neural retriever trained offline on claim-to-triple supervision returns candidate triples,

$$O_{t+1} = \bigcup_{c \in R_t} \mathrm{Retrieve}(\rho(c)). \tag{12}$$

This step expands the candidate pool; the key budgeted decision is which triples from $Z_t \cup O_{t+1}$ should remain visible to the next reasoning call.

Budgeted evidence projection. Let $\mathcal{C}_t = Z_t \cup O_{t+1}$ be the projection candidate pool after residual-conditioned retrieval. Base CoRE scores each candidate triple by its residual-support gain,

$$\Delta(e; R_t) = \sum_{c \in R_t} (1 - \mathrm{Sup}(c; Z_t)) \mathbf{1}[s(c, T(e)) > \tau], \tag{13}$$

and sets Z_{t+1} to the top-K triples in $\mathcal{C}_t$. This projection is interpretable: it prioritizes triples that close larger support deficits and preferentially evicts triples unrelated to the current residuals. Theorem 4.2 suggests a broader objective for the same step: projection should preserve evidence that reduces residual mass and avoid retaining triples that do not support unresolved claims.

CoRE-RL uses a one-step offline contextual bandit for evidence projection: a scorer is fitted to logged selections with weights derived from next-round feedback, and deterministically retains the top-K triples at test time. We first run CoRE on training questions and log, for each round, the state $(q, Z_t, \hat{a}_t, R_t)$, the candidate pool $\mathcal{C}_t$, the selected set Z_{t+1}, and next-round feedback. Each candidate $e \in \mathcal{C}_t$ becomes one training example with feature vector $\psi(q, Z_t, \hat{a}_t, R_t, e)$ and label $y_e = \mathbf{1}[e \in Z_{t+1}]$. The example weight is increased when the selected projection improves next-round answer F1, claim support, or residual mass reduction:

$$\omega_t = 1 + \alpha[\Delta \mathrm{F1}_t]_+ + \beta[\Phi_t - \Phi_{t+1}]_+ + \gamma[\Delta \mathrm{Sup}_t]_+. \tag{14}$$

Here $[x]_+ = \max(x, 0)$, $\Delta \mathrm{F1}_t$ is the next-round answer-F1 gain, Φ_t is the residual mass in Eq. (7), $\Delta \mathrm{Sup}_t$ is the average support gain for residual claims after projection, and $\alpha, \beta, \gamma \geq 0$ balance the three feedback signals. We fit a linear model or shallow multi-layer perceptron, or MLP, scorer

$$g_\theta(e \mid q, Z_t, \hat{a}_t, R_t) = f_\theta(\psi(q, Z_t, \hat{a}_t, R_t, e)), \tag{15}$$

with weighted binary/ranking loss. The feature vector includes retriever score, anchor/relation matches, question and answer overlap, slot coverage, whether e is already retained, round statistics, and local graph features. At inference time, CoRE-RL computes g_θ for all $e \in \mathcal{C}_t$ and retains the top-K triples; all earlier steps in the loop are unchanged.

Verifiable controller. The controller has two actions, RETRIEVE and STOP, with a maximum of $T = 3$ reasoning rounds in the reported experiments. It retrieves for executable residuals and can end execution when evidence is sufficient, no further support gain is observed, or the round cap is reached. Output status is evaluated separately: `SUPPORTED` requires a nonempty set of supported extracted claims and no remaining open slots; remaining unsupported claims or slots yield `UNRESOLVED`, and an empty or invalid claim verdict yields `UNVERIFIED`. These statuses use no gold answer; answer stability is a controller signal rather than evidence of support.

Table 1: Performance on WebQSP and CWQ with the backbones shown in each row. Baseline sources and evaluation settings are specified in Appendix I.8; matched controls are reported separately in Table 11.

Types	Sub-type	Methods	WebQSP Hit	WebQSP F1	CWQ Hit	CWQ F1
Agent	Prompting	StructGPT w/ChatGPT [12]	72.6	–	54.3	–
		ToG w/GPT-4 [6]	82.6	76.0	68.5	60.2
		PoG w/GPT-4 [19]	87.3	–	75.0	–
		Interactive-KBQA w/GPT-4 [20]	72.5	–	59.2	–
		ReKnoS w/GPT-4o-mini [2]	83.8	–	66.8	–
	Fine-tuned	GCR w/ChatGPT [14]	92.6	73.2	72.7	60.9
		GCR w/GPT-4o-mini [14]	92.2	74.1	75.8	61.7
	Search	DualR w/Llama2-7B [28]	84.9	71.6	68.9	62.1
		DoG w/GPT-4 [29]	91.0	–	56.0	–
		RwT w/ChatGPT [21]	87.0	79.7	72.4	66.7
		DP w/LLaMA3.1-8B [23]	87.9	75.7	70.8	58.5
RAG	Retrieval	RD-P w/ChatGPT [16]	85.3	69.7	63.5	56.6
		EWEK-QA w/KG+Web [30]	71.3	–	52.5	–
		EffiQA w/GPT-4 [31]	82.9	–	69.5	–
	KG Retrieval	KG-CoT w/GPT-4 [15]	84.9	–	62.3	–
		FiDeLis w/gpt-4-turbo [32]	84.3	78.3	71.5	64.3
		SubgraphRAG w/Llama3.1-8B [10]	86.6	70.6	57.0	47.2
		SubgraphRAG w/GPT-4o-mini [10]	90.1	77.5	62.0	54.1
		G-Retriever [17]	73.5	53.4	–	–
		GNN-RAG [18]	82.8	73.5	62.8	60.4
		RJE w/Llama3.1-8B [22]	89.2	–	71.5	–
		RJE w/GPT-4o-mini [22]	92.5	–	77.1	–
Ours	**CoRE**	w/Llama3.1-8B	89.5	74.5	72.5	63.5
		w/GPT-4o-mini	93.5	80.5	76.7	68.9
	CoRE-RL	w/Llama3.1-8B	92.3	78.9	75.7	68.8
		w/GPT-4o-mini	**93.7**	**80.9**	**80.0**	**70.4**

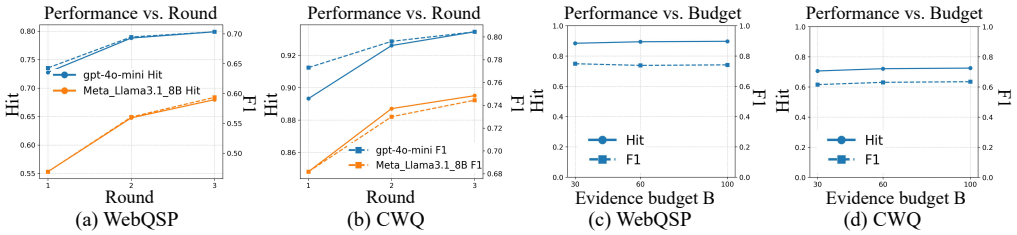


Figure 3: Performance across reasoning rounds and evidence budgets. Additional rounds improve answer quality, while the CoRE family remains robust when the retained evidence budget is reduced.

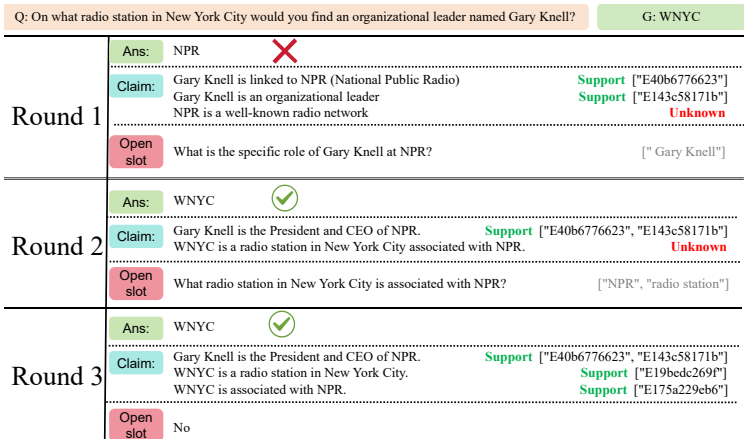


Figure 4: Qualitative evidence repair on a CWQ example. The first round predicts the wrong organization, NPR; the second round finds the correct station, WNYC, but still exposes unsupported station-level claims; the final round retrieves the missing evidence and clears the residual.

6 Experiments

We evaluate four research questions: **RQ1** whether residual-coupled evidence control improves KGQA accuracy; **RQ2** whether multiple rounds improve both answers and grounding; **RQ3** whether the CoRE family remains robust under tight budgets and realistic cost constraints; and **RQ4** whether learned projection in CoRE-RL improves the base CoRE loop.

Setup. We evaluate on WebQuestionsSP, or WebQSP [33], and Complex WebQuestions, or CWQ [34], both grounded in Freebase [35]. Baselines include retrieve-then-reason KG-RAG systems and iterative/agent-based KGQA methods. We report answer-level Hit and F1, and retained-evidence support metrics FactSup and FCR. FactSup is the per-question supported-claim fraction averaged over all questions; FCR requires both a correct answer and support for all extracted explanation claims. All four metrics include every evaluated question, including `UNRESOLVED` and `UNVERIFIED` outputs. Results explicitly labeled strict report Hit/F1 on `SUPPORTED` outputs together with their coverage. For CoRE-RL, offline contextual-bandit training changes only the projection scorer; retrieval, residual extraction, and the controller are shared with CoRE. Appendices H and I give dataset statistics, baseline protocols, and implementation details.

Table 2: Performance breakdown by reasoning hops. CoRE and CoRE-RL remain strongest on multi-hop questions, where static evidence views often miss intermediate relations.

Method	WebQSP-sub				CWQ-sub			
	1 hop		>1 hop		1 hop		>1 hop	
	F1	Hit	F1	Hit	F1	Hit	F1	Hit
RoG-Joint	77.05	92.96	62.53	81.54	59.75	66.33	58.14	65.45
RoG-Sep	74.50	89.83	55.62	73.45	59.35	66.20	57.16	63.79
SubgraphRAG	80.56	92.86	74.11	88.51	57.36	67.34	62.77	71.07
CoRE	84.60	95.57	76.79	91.76	65.60	73.77	75.59	81.21
CoRE-RL	**84.62**	**96.52**	**81.36**	**92.71**	**74.88**	**82.66**	**77.97**	**85.62**

Main effectiveness, RQ1. Table 1 shows that CoRE improves over strong retrieve-then-reason and agent-style baselines, especially on CWQ where intermediate evidence is often missing. With GPT-4o-mini, base CoRE reaches 76.7 Hit and 68.9 F1 on CWQ, compared with 62.0/54.1 for SubgraphRAG under the same backbone. CoRE-RL further strengthens this residual-coupled loop by learning which retrieved triples should be retained for subsequent reasoning.

With Llama3.1-8B, CoRE-RL improves CoRE from 89.5/74.5 to 92.3/78.9 on WebQSP and from 72.5/63.5 to 75.7/68.8 on CWQ. With GPT-4o-mini, CoRE-RL improves CWQ from 76.7/68.9 to 80.0/70.4. On 500 CWQ questions with matched retrieval, backbone, $K = 60$, and $T = 3$, CoRE gains 2.14 F1 points over query-only retrieval and 5.34 over progressive add-only updates (Table 11).

Rounds, grounding, and mechanism checks, RQ2. Figure 3 and Appendix Table 19 show monotonic accuracy gains across rounds; on CWQ with GPT-4o-mini, F1 rises from 64.30 to 69.48 to 70.35. Table 2 shows that CoRE-RL preserves the gains of the residual-coupled loop across hop groups and further strengthens the base CoRE results, including the harder multi-hop subsets. Table 3 reports cumulative verifiability up to each round. CoRE steadily raises FCR from 76.96 to 81.22 on WebQSP and from 53.56 to 57.95 on CWQ, while CoRE-RL further improves the final FCR to 86.67 and 64.00 under the learned projection policy.

Table 3: Cumulative verifiability up to each round using GPT-4o-mini.

Method	R	WebQSP FactSup	WebQSP FCR	CWQ FactSup	CWQ FCR
CoRE	1	92.63	76.96	85.66	53.56
CoRE	2	94.10	79.23	86.04	55.39
CoRE	3	94.74	81.22	86.96	57.95
CoRE-RL	1	**95.38**	**81.00**	**87.35**	**55.33**
CoRE-RL	2	**97.47**	**85.00**	**92.88**	**62.33**
CoRE-RL	3	**98.16**	**86.67**	**94.78**	**64.00**

Figure 4 visualizes this separation between answer discovery and evidence consolidation: after a plausible answer appears, later rounds can still retrieve evidence that makes the reasoning supportable. Appendix Table 7 reports the logged residual-mass proxy across rounds. Table 4 audits one extraction pass on 50 trajectories per dataset. The extractor misses at least one critical claim on 26% of WebQSP and 34% of CWQ questions, and extraction errors harm retrieval on 14% and 24%, respectively. These are single-pass extraction measurements; persistent omission can leave an evidence need absent from the residual interface.

Table 4: Single-pass claim-extraction audit on 50 trajectories per dataset. Correct/irrelevant are claim-level percentages; missed-critical and retrieval-harmed are question-level percentages.

Dataset	Samples	Correct claims	Irrelevant claims	Missed critical	Retrieval harmed
WebQSP	50	76.13%	23.87%	26.00%	14.00%
CWQ	50	70.72%	29.28%	34.00%	24.00%

Budget and learned projection, RQ3 and RQ4. Figure 3 shows that CoRE remains effective under strict budgets; on WebQSP at round 3, reducing K from 100 to 30 changes Hit only from 89.70 to 88.40 and F1 from 74.15 to 74.89. Appendix Table 20 further shows that additional rounds compensate for smaller retained subgraphs on both datasets. Table 5 isolates the CoRE-RL projection scorer in matched settings. Learned projection improves WebQSP at K=60 from 90.67/78.67 to 92.00/79.30 and raises CWQ at K=100 from 75.67/68.15 to 75.67/68.78 with the residual interface and controller fixed. Under 30% residual-feature corruption, a noise-trained selector recovers 11.00 and 4.09 percentage points more logged projection targets on WebQSP and CWQ than a clean-trained selector across 20 seeds. Appendix G separates this component diagnostic from independent end-to-end evaluation.

Table 5: Projection-policy ablation. The residual interface and controller are fixed; only the projection scorer changes.

Setting	Scorer	Hit	F1
WebQSP-60	Base	90.67	78.67
WebQSP-60	Bandit-MLP	**92.00**	**79.30**
CWQ-100	Base	75.67	68.15
CWQ-100	Bandit-MLP	75.67	**68.78**

Table 6 complements the budget study with online LLM usage. CoRE uses far fewer calls than agent-style search methods while exposing the expected budget trade-off: smaller retained subgraphs reduce input tokens, whereas larger budgets provide more evidence for the gains in Table 1. The learned projection scorer adds no LLM stage. Relative to the first round of the same run, the full loop adds 1.052 calls and 5.52 s on clean WebQSP, and 1.550 calls and 9.01 s on clean CWQ (Table 10).

Appendix G adds long-context and stronger one-shot checks, further highlighting evidence selection and retention, rather than raw context length alone, as the key bottleneck.

7 Conclusion and Limitations

This paper identifies dynamic evidence identification as a central bottleneck in complex KGQA: the facts needed for multi-hop answers are often revealed by intermediate reasoning rather than fully specified by the initial question. We formalize this dependency as a partially observable evidence-control problem and instantiate it with CoRE, where unsupported answer-critical claims become residuals that drive evidence retrieval and subgraph revision. CoRE-RL extends the same residual-coupled loop by learning how residual-conditioned candidates should be projected into the retained subgraph. Across WebQSP and CWQ, the CoRE family improves answer quality, evidence-supported verifiability, multi-hop robustness, and budgeted evidence allocation.

Table 6: Average online LLM usage on CWQ with GPT-4o-mini.

Method	Calls	Input	Output
ToG	22.6	8,182.9	1,486.4
PoG	13.3	7,803.0	353.2
RJE	7.9	5,769.1	247.2
SubgraphRAG	**1.14**	2,914.7	293.8
CoRE, $K = 10$	2.55	**2,167.9**	745.3
CoRE, $K = 30$	2.33	3,841.8	831.3
CoRE, $K = 60$	2.15	4,884.4	736.5
CoRE, $K = 100$	2.10	5,645.6	675.5
CoRE-RL, $K = 100$	2.11	5,619.5	665.1

Limitations. The residual interface depends on claim extraction and KG coverage. Citation and support checks assess claims that are produced, whereas a necessary fact omitted from the selected traces across all rounds produces no repair signal; repeated reasoning can expose such facts but provides no coverage guarantee. FactSup and FCR measure support for extracted claims in retained evidence, rather than full-KG completeness or token-level causal faithfulness. Multi-round inference adds latency over single-pass RAG, and the transfer evaluation covers question distributions within Freebase.

References

[1] Haochen Liu, Song Wang, Chen Chen, and Jundong Li. Question-aware knowledge graph prompting for enhancing large language models. *arXiv preprint arXiv:2503.23523*, 2025.

[2] Song Wang, Junhong Lin, Xiaojie Guo, Julian Shun, Jundong Li, and Yada Zhu. Reasoning of large language models over knowledge graphs with super-relations. *arXiv preprint arXiv:2503.22166*, 2025.

[3] Mengqi Zhang, Xiaotian Ye, Qiang Liu, Pengjie Ren, Shu Wu, and Zhumin Chen. Knowledge graph enhanced large language model editing. *arXiv preprint arXiv:2402.13593*, 2024.

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

[5] Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful and interpretable large language model reasoning. *arXiv preprint arXiv:2310.01061*, 2023.

[6] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. *arXiv preprint arXiv:2307.07697*, 2023.

[7] Xiangkun Hu, Dongyu Ru, Lin Qiu, Qipeng Guo, Tianhang Zhang, Yang Xu, Yun Luo, Pengfei Liu, Yue Zhang, and Zheng Zhang. Knowledge-centric hallucination detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6953–6975, 2024.

[8] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[9] Qingfei Zhao, Ruobing Wang, Yukuo Cen, Daren Zha, Shicheng Tan, Yuxiao Dong, and Jie Tang. Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering. *arXiv preprint arXiv:2410.18050*, 2024.

[10] Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *arXiv preprint arXiv:2410.20724*, 2024.

[11] Costas Mavromatis and George Karypis. Gnn-rag: Graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16682–16699, 2025.

[12] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*, 2023.

[13] Siyuan Fang, Kaijing Ma, Tianyu Zheng, Xeron Du, Ningxuan Lu, Ge Zhang, and Qingkun Tang. Karpa: a training-free method of adapting knowledge graph as references for large language model's reasoning path aggregation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24724–24746, 2025.

[14] Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models. *arXiv preprint arXiv:2410.13080*, 2024.

[15] Ruilin Zhao, Feng Zhao, Long Wang, Xianzhi Wang, and Guandong Xu. KG-CoT: Chain-of-thought prompting of large language models over knowledge graphs for knowledge-aware question answering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, pages 6642–6650. International Joint Conferences on Artificial Intelligence Organization, 2024. doi: 10.24963/ijcai.2024/734. URL https://doi.org/10.24963/ijcai.2024/734.

[16] Yubo Huang and Guosun Zeng. Rd-p: A trustworthy retrieval-augmented prompter with knowledge graphs for llms. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 942–952, 2024.

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

[18] Costas Mavromatis and George Karypis. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*, 2024.

[19] Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs. *Advances in Neural Information Processing Systems*, 37:37665–37691, 2024.

[20] Guanming Xiong, Junwei Bao, and Wen Zhao. Interactive-kbqa: Multi-turn interactions for knowledge base question answering with large language models. *arXiv preprint arXiv:2402.15131*, 2024.

[21] Tiesunlong Shen, Jin Wang, Xuejie Zhang, and Erik Cambria. Reasoning with trees: Faithful question answering over knowledge graph. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3138–3157, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.211/.

[22] Can Lin, Zhengwang Jiang, Ling Zheng, Qi Zhao, Yuhang Zhang, Qi Song, and Wangqiu Zhou. Rje: A retrieval-judgment-exploration framework for efficient knowledge graph question answering with llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17288–17305, 2025.

[23] Jie Ma, Ning Qu, Zhitao Gao, Rui Xing, Jun Liu, Hongbin Pei, Jiang Xie, Linyun Song, Pinghui Wang, Jing Tao, et al. Deliberation on priors: Trustworthy reasoning of large language models on knowledge graphs. *arXiv preprint arXiv:2505.15210*, 2025.

[24] Xinze Li, Yixin Cao, Liangming Pan, Yubo Ma, and Aixin Sun. Towards verifiable generation: A benchmark for knowledge-aware language model attribution. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 493–516, 2024.

[25] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *arXiv preprint arXiv:2403.12968*, 2024.

[26] Qinyuan Cheng, Xiaonan Li, Shimin Li, Qin Zhu, Zhangyue Yin, Yunfan Shao, Linyang Li, Tianxiang Sun, Hang Yan, and Xipeng Qiu. Unified active retrieval for retrieval augmented generation. *arXiv preprint arXiv:2406.12534*, 2024.

[27] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[28] Guangyi Liu, Yongqi Zhang, Yong Li, and Quanming Yao. Dual reasoning: A gnn-llm collaborative framework for knowledge graph question answering. *arXiv preprint arXiv:2406.01145*, 2024.

[29] Jie Ma, Zhitao Gao, Qi Chai, Wangchun Sun, Pinghui Wang, Hongbin Pei, Jing Tao, Lingyun Song, Jun Liu, Chen Zhang, et al. 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, 2025. doi: 10.1609/aaai.v39i23.34658. URL https://doi.org/10.1609/aaai.v39i23.34658.

[30] Mohammad Dehghan, Mohammad Alomrani, Sunyam Bagga, David Alfonso-Hermelo, Khalil Bibi, Abbas Ghaddar, Yingxue Zhang, Xiaoguang Li, Jianye Hao, Qun Liu, et al. Ewek-qa: Enhanced web and efficient knowledge graph retrieval for citation-based question answering systems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14169–14187, 2024.

[31] Zixuan Dong, Baoyun Peng, Yufei Wang, Jia Fu, Xiaodong Wang, Xin Zhou, Yongxue Shan, Kangchen Zhu, and Weiguo Chen. Effiqa: Efficient question-answering with strategic multi-model collaboration on knowledge graphs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7180–7194, 2025.

[32] Yuan Sui, Yufei He, Nian Liu, Xiaoxin He, Kun Wang, and Bryan Hooi. Fidelis: Faithful reasoning in large language models for knowledge graph question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8315–8330, 2025.

[33] Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 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, 2016.

[34] Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex questions. *arXiv preprint arXiv:1803.06643*, 2018.

[35] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 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. Association for Computing Machinery, 2008. doi: 10.1145/1376616.1376746. URL https://doi.org/10.1145/1376616.1376746.

[36] Yuzheng Cai, Zhenyue Guo, Yiwen Pei, Wanrui Bian, and Weiguo Zheng. Simgrag: Leveraging similar subgraphs for knowledge graphs driven retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3139–3158, 2025.

[37] Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. Knowledge graph-guided retrieval augmented generation. *arXiv preprint arXiv:2502.06864*, 2025.

[38] Meng-Chieh Lee, Qi Zhu, Costas Mavromatis, Zhen Han, Soji Adeshina, Vassilis N Ioannidis, Huzefa Rangwala, and Christos Faloutsos. Hybgrag: Hybrid retrieval-augmented generation on textual and relational knowledge bases. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 879–893, 2025.

[39] Junqi Gao, Xiang Zou, YIng Ai, Dong Li, Yichen Niu, Biqing Qi, and Jianxing Liu. Graph counselor: Adaptive graph exploration via multi-agent synergy to enhance llm reasoning. *arXiv preprint arXiv:2506.03939*, 2025.

[40] Yushi Sun, Kai Sun, Yifan Ethan Xu, Xiao Yang, Xin Luna Dong, Nan Tang, and Lei Chen. Kerag: Knowledge-enhanced retrieval-augmented generation for advanced question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 6194–6216, 2025.

[41] Haoran Luo, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, et al. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. *arXiv preprint arXiv:2503.21322*, 2025.

[42] Shijie Wang, Wenqi Fan, Yue Feng, Lin Shanru, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. Knowledge graph retrieval-augmented generation for llm-based recommendation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27152–27168, 2025.

[43] Xingliang Wang, Zemin Liu, Junxiao Han, and Shuiguang Deng. Rag4gfm: Bridging knowledge gaps in graph foundation models through graph retrieval augmented generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[44] Jinhao Jiang, Kun Zhou, Wayne Xin Zhao, Yang Song, Chen Zhu, Hengshu Zhu, and Ji-Rong Wen. 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, 2025.

[45] QianyiHu QianyiHu, Xinhui Tu, Guo Cong, and Shunping Zhang. Time-aware react agent for temporal knowledge graph question answering. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6013–6024, 2025.

[46] Minh Pham Dinh, Michael G Yankoski, Munira Syed, and Trenton W Ford. Davis: Planning agent with knowledge graph-powered inner monologue. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16490–16505, 2025.

[47] Mufan Xu, Gewen Liang, Kehai Chen, Wei Wang, Xun Zhou, Muyun Yang, Tiejun Zhao, and Min Zhang. Memory-augmented query reconstruction for llm-based knowledge graph reasoning. *arXiv preprint arXiv:2503.05193*, 2025.

[48] Ziyi Guan, Jason Chun Lok Li, Zhijian Hou, Pingping Zhang, Donglai Xu, Yuzhi Zhao, Mengyang Wu, Jinpeng Chen, Thanh-Toan Nguyen, Pengfei Xian, et al. Kg-rag: Enhancing gui agent decision-making via knowledge graph-driven retrieval-augmented generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5396–5405, 2025.

[49] Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools. *arXiv preprint arXiv:2502.04644*, 2025.

[50] Heng Zhou, Hejia Geng, Xiangyuan Xue, Li Kang, Yiran Qin, Zhiyong Wang, Zhenfei Yin, and Lei Bai. Reso: A reward-driven self-organizing llm-based multi-agent system for reasoning tasks. *arXiv preprint arXiv:2503.02390*, 2025.

[51] Yanbo Wang, Zixiang Xu, Yue Huang, Xiangqi Wang, Zirui Song, Lang Gao, Chenxi Wang, Xiangru Tang, Yue Zhao, Arman Cohan, et al. Dyflow: Dynamic workflow framework for agentic reasoning. *arXiv preprint arXiv:2509.26062*, 2025.

[52] Xueliang Zhao, Wei Wu, Jian Guan, Qintong Li, and Lingpeng Kong. Dynaact: Large language model reasoning with dynamic action spaces. *arXiv preprint arXiv:2511.08043*, 2025.

[53] Jiajun Shi, Jian Yang, Jiaheng Liu, Xingyuan Bu, Jiangjie Chen, Junting Zhou, Kaijing Ma, Zhoufutu Wen, Bingli Wang, Yancheng He, et al. Korgym: A dynamic game platform for llm reasoning evaluation. *arXiv preprint arXiv:2505.14552*, 2025.

[54] Belinda Mo, Kyssen Yu, Joshua Kazdan, Joan Cabezas, Proud Mpala, Lisa Yu, Chris Cundy, Charilaos Kanatsoulis, and Sanmi Koyejo. Kggen: Extracting knowledge graphs from plain text with language models. *arXiv preprint arXiv:2502.09956*, 2025.

[55] Enjun Du, Xunkai Li, Tian Jin, Zhihan Zhang, Rong-Hua Li, and Guoren Wang. Graphmaster: Automated graph synthesis via llm agents in data-limited environments. *arXiv preprint arXiv:2504.00711*, 2025.

[56] Marinela Parović, Ze Li, and Jinhua Du. Generating domain-specific knowledge graphs from large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11558–11574, 2025.

[57] Haoyu Huang, Chong Chen, Zeang Sheng, Yang Li, and Wentao Zhang. Can llms be good graph judge for knowledge graph construction? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10940–10959, 2025.

[58] Pengcheng Jiang, Cao Xiao, Minhao Jiang, Parminder Bhatia, Taha Kass-Hout, Jimeng Sun, and Jiawei Han. Reasoning-enhanced healthcare predictions with knowledge graph community retrieval. *arXiv preprint arXiv:2410.04585*, 2024.

[59] Yikuan Hu, Jifeng Zhu, Lanrui Tang, and Chen Huang. Remindrag: Low-cost llm-guided knowledge graph traversal for efficient rag. *arXiv preprint arXiv:2510.13193*, 2025.

[60] Yuxing Lu and Jinzhuo Wang. Karma: Leveraging multi-agent llms for automated knowledge graph enrichment. *arXiv preprint arXiv:2502.06472*, 2025.

[61] Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. *arXiv preprint arXiv:2407.10805*, 2024.

[62] Junchi Yu, Yujie Liu, Jindong Gu, Philip Torr, and Dongzhan Zhou. Can knowledge-graph-based retrieval augmented generation really retrieve what you need? *arXiv preprint arXiv:2510.16582*, 2025.

[63] Costas Mavromatis, Soji Adeshina, Vassilis N Ioannidis, Zhen Han, Qi Zhu, Ian Robinson, Bryan Thompson, Huzefa Rangwala, and George Karypis. Byokg-rag: Multi-strategy graph retrieval for knowledge graph question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27869–27886, 2025.

[64] Youmin Ko, Sungjong Seo, and Hyunjoon Kim. Cooperative retrieval-augmented generation for question answering: Mutual information exchange and ranking by contrasting layers. *arXiv preprint arXiv:2512.10422*, 2025.

[65] Jiarui Zhang, Xiangyu Liu, Yong Hu, Chaoyue Niu, Fan Wu, and Guihai Chen. RAGRouter: Learning to route queries to multiple retrieval-augmented language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=4VKVUmE1I8`. NeurIPS 2025 poster.

[66] Zeyu Shen, Basileal Imana, Tong Wu, Chong Xiang, Prateek Mittal, and Aleksandra Korolova. Reliabilityrag: Effective and provably robust defense for rag-based web-search. *arXiv preprint arXiv:2509.23519*, 2025.

[67] Chaeyun Jang, Deukhwan Cho, Seanie Lee, Hyungi Lee, and Juho Lee. Reliable decision making via calibration oriented retrieval augmented generation. *arXiv preprint arXiv:2411.08891*, 2024.

[68] Mykel J Kochenderfer, Tim A Wheeler, and Kyle H Wray. *Algorithms for decision making*. MIT press, 2022.

[69] David Silver and Joel Veness. Monte-carlo planning in large POMDPs. In *Advances in Neural Information Processing Systems 23*, pages 2164–2172, 2010. URL https://papers.neurips.cc/paper/4031-monte-carlo-planning-in-large-pomdps.

[70] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2012.

[71] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Appendix Contents

1. Related Work Details ..... A
2. Technical Details for Evidence-aware Optimality ..... B
3. Unified Separation Template and Instantiations C
4. Proofs for Strict Separations ..... D
5. Necessary Coupling under Realizability ..... E
6. Experiment Setup ..... H
7. Experimental Reproducibility I
8. CoRE Algorithm ..... J
9. Experimental Details Analysis ..... K
10. Limitations L

A Related Work Details

A.1 Retrieve-then-Reason KG-RAG

Retrieve-then-reason remains the dominant instantiation of KG-aware RAG: systems retrieve triples/paths/subgraphs once and prompt an LLM on a fixed evidence view. Recent work substantially strengthens this paradigm by improving the *representation* and coverage of retrieved evidence, including budgeted subgraph retrieval baselines and subgraph reuse [10, 36, 37], hybrid text–KG fusion and GraphRAG-style evidence assembly [38, 39, 40], and richer relational structures such as hypergraph contexts [41]. Graph-structured retrieval augmentation has also been generalized beyond KGQA to application-facing settings and graph foundation models [42, 43]. However, because evidence is pre-committed *before* reasoning, these approaches cannot decide on-the-fly what evidence is needed versus extraneous, whereas CoRE makes intermediate claims explicit, uses unsupported residuals to trigger targeted repair, and evicts unused triples to maintain a compact, budgeted evidence state.

A.2 Agent-based KG Reasoning on Graphs

Agent-based KG reasoning replaces one-shot evidence selection with sequential graph/tool actions (traversal, frontier expansion, structured queries) and memory, enabling adaptive evidence access during multi-step inference. Representative 2025 systems provide practical frameworks and explicit action loops (e.g., KG-Agent, RJE [44, 22]), with temporal and planning/memory-centric variants refining action constraints and internal deliberation (TempAgent, DAVIS [45, 46]); however, their decision making is often dominated by local traversal heuristics, which makes cross-region evidence aggregation for compositional queries difficult. In parallel, trustworthiness-oriented lines of work enforce faithful constraints or calibrated deliberation on graphs [23, 14], yet hard verifiability constraints, which require generated claims to be explicitly supported by observed evidence, can amplify the coverage-verifiability tension under KG incompleteness by suppressing semantically valid reasoning steps not explicitly supported by observed edges. Broader 2025 ecosystems study memory/query reconstruction, general agent frameworks, dynamic workflows/action spaces, interactive evaluation, and KG construction/synthesis [47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 2, 58, 59], while multi-agent systems connect reasoning with KG enrichment and updating [60]. In contrast, CoRE avoids committing to purely local traversal or hard path constraints by coupling reasoning to a budgeted subgraph state via claim-level residuals, enabling targeted cross-region evidence repair while preserving semantic inference under missing edges.

A.3 Iterative Retrieval-Reasoning Loops

A growing line of work interleaves retrieval and reasoning across multiple rounds to refine evidence based on intermediate model states. Representative mechanisms include iterative graph/text exploration with score fusion for multi-hop coverage [61, 62], tool-integrated multi-strategy KG-RAG pipelines for robustness across query types [63], and system-level iterative RAG designs that study coordination, routing, robustness/calibration, and benchmarking of multi-step retrieval [64, 65, 66, 67, 62]. While effective, these loops typically rely on coarse control signals (e.g., expansion heuristics or generic reflection) and are often add-only, so budgets can still be consumed by partially relevant context. CoRE's key advantage is a *bidirectionally coupled* evidence–reasoning loop that keeps intermediate reasoning consistently aligned with a budgeted subgraph: reasoning adaptively steers evidence updates, while the evolving subgraph provides explicit support/constraints on subsequent steps, improving robustness under KG incompleteness and tight LLM context budgets.

B Technical Details for Evidence-aware Optimality

B.1 Belief-MDP reduction and budget projection

We make precise the belief-MDP reduction and the role of the budget projection operator used in Section 4. We follow standard POMDP notation and explicitly account for the additional *budgeted evidence memory* component.

POMDP ingredients. Fix a natural language query q and a KG G (treated as part of the environment specification). Consider a finite-horizon POMDP with: (i) latent state space $\mathcal{X}$; (ii) evidence acquisition action space $\mathcal{U}$; (iii) observation space $\mathcal{O}$; (iv) latent transition kernel $P(x' \mid x, u)$; (v) observation kernel $\Omega(O \mid x', u)$. The horizon is $T \in \mathbb{N}$ and rewards are bounded. We assume $\mathcal{U}$ is finite.

Histories, information, and belief. At time t, denote the interaction history (excluding Z) by

$$h_t \triangleq (q, u_{1:t-1}, O_{1:t}), \tag{16}$$

where $O_t \in \mathcal{O}$ is the observation revealed after action u_{t-1}. In our setting, the agent also maintains a budgeted evidence memory state $Z_t \subseteq \mathcal{E}$ (where $\mathcal{E}$ is the universe of candidate KG edges/triples), which is *observable to the agent* by construction (it is the agent's internal retained context). Therefore, the agent's full information at time t is the *augmented history*

$$\tilde{h}_t \triangleq (h_t, Z_t) = (q, u_{1:t-1}, O_{1:t}, Z_t). \tag{17}$$

We define the belief (posterior over latent states) conditioned on the agent's full information:

$$b_t(x) \triangleq \mathbb{P}(x_t = x \mid \tilde{h}_t). \tag{18}$$

This definition is without loss of generality. Indeed, when Z_t is a deterministic function of (Z_{t-1}, O_t) and Z_0 is fixed, conditioning on Z_t adds no extra information beyond h_t; however, for our separation constructions we will allow randomized Z_0 (known to the agent), and thus it is essential to condition on Z_t in (18).

Belief update. Given action u_t and observation O_{t+1}, the Bayesian belief update operator Bayes is defined by Bayes' rule (standard; see, e.g., [68, 69]):

$$b_{t+1}(x') = \mathrm{Bayes}(b_t, u_t, O_{t+1})(x') = \frac{\Omega(O_{t+1} \mid x', u_t) \sum_{x \in \mathcal{X}} P(x' \mid x, u_t)\, b_t(x)}{\sum_{\bar{x} \in \mathcal{X}} \Omega(O_{t+1} \mid \bar{x}, u_t) \sum_{x \in \mathcal{X}} P(\bar{x} \mid x, u_t)\, b_t(x)}. \tag{19}$$

We assume the denominator in (19) is strictly positive for observations with non-zero probability under (b_t, u_t).

Belief-induced predictive observation distribution. For any belief b and action u, define the belief-induced predictive distribution over observations by marginalization:

$$\Omega(O \mid b, u) \triangleq \sum_{x' \in \mathcal{X}} \Omega(O \mid x', u) \Big(\sum_{x \in \mathcal{X}} P(x' \mid x, u)\, b(x) \Big). \tag{20}$$

This is the standard reduction from a POMDP to a belief-MDP [68, 69].

Budgeted evidence memory and projection update. The agent maintains a budgeted evidence state $Z_t \subseteq \mathcal{E}$ updated by a (possibly stochastic, but here deterministic) projection operator

$$Z_{t+1} = \mathcal{P}_K(Z_t \cup \Gamma(O_{t+1})), \tag{21}$$

where $\Gamma : \mathcal{O} \to 2^{\mathcal{E}}$ extracts the subset of KG edges/triples carried by an observation.[1] The operator $\mathcal{P}_K : 2^{\mathcal{E}} \to 2^{\mathcal{E}}$ enforces a fixed evidence budget K (e.g., $|\mathcal{P}_K(S)| \le K$).

Budget feasibility. We say $\mathcal{P}_K$ is *budget-feasible* if

$$|\mathcal{P}_K(S)| \le K, \quad \forall S \subseteq \mathcal{E}. \tag{22}$$

Non-triviality of $\mathcal{P}_K$ (consistent with main text). We formalize the assumption "$\mathcal{P}_K$ is non-trivial". We adopt the *same* non-triviality criterion as in the main text:

Definition B.1 (Non-trivial budget projection)**.** A budget-feasible operator $\mathcal{P}_K$ is *non-trivial* if there exist sets $Z \subseteq Z'$ such that

$$\mathcal{P}_K(Z) \neq \mathcal{P}_K(Z'). \tag{23}$$

This captures that adding candidate evidence can *change* what is retained under the budget, i.e., evidence can be dropped or replaced.

Relation to edge dropping. Many natural projections (e.g., top-K selection by a scoring function $w(e)$) satisfy a stronger property: there exist sets S with $\mathcal{P}_K(S) \subsetneq S$ whenever $|S| > K$ and scores are not all equal. Such operators are non-trivial in the sense of Definition B.1 because taking $Z = \mathcal{P}_K(S)$ and $Z' = S$ yields $\mathcal{P}_K(Z) = Z \neq \mathcal{P}_K(Z')$.

Belief-MDP state and Markov property. Define the augmented information state (belief-MDP state)

$$s_t \triangleq (b_t, Z_t) \in \Delta(\mathcal{X}) \times 2^{\mathcal{E}}, \tag{24}$$

where $\Delta(\mathcal{X})$ is the probability simplex over $\mathcal{X}$. Given $s_t = (b_t, Z_t)$ and action u_t, the environment generates $x_{t+1} \sim P(\cdot \mid x_t, u_t)$ and then $O_{t+1} \sim \Omega(\cdot \mid x_{t+1}, u_t)$. The belief update is $b_{t+1} = \mathrm{Bayes}(b_t, u_t, O_{t+1})$ by (19), and the evidence update is $Z_{t+1} = \mathcal{P}_K(Z_t \cup \Gamma(O_{t+1}))$ by (21). Hence the state transition is induced by

$$s_{t+1} = \Big(\mathrm{Bayes}(b_t, u_t, O_{t+1}),\ \mathcal{P}_K(Z_t \cup \Gamma(O_{t+1})) \Big). \tag{25}$$

Lemma B.2 (Markov property of (b_t, Z_t))**.** *For the finite-horizon process defined above, conditioned on actions,*

$$\mathbb{P}(s_{t+1} \in \cdot \mid s_{0:t}, u_{0:t}) = \mathbb{P}(s_{t+1} \in \cdot \mid s_t, u_t), \qquad \forall t, \tag{26}$$

i.e., (s_t) is a controlled Markov process.

Proof. Fix t. Given $(s_t, u_t) = (b_t, Z_t, u_t)$, the predictive distribution of O_{t+1} is $\Omega(\cdot \mid b_t, u_t)$ by (20). Moreover, b_{t+1} is a deterministic function of (b_t, u_t, O_{t+1}) via (19), and Z_{t+1} is a deterministic function of (Z_t, O_{t+1}) via (21). Therefore the conditional distribution of s_{t+1} given the entire past $(s_{0:t}, u_{0:t})$ depends on the past only through (s_t, u_t), which proves (26). □

Bellman recursion on (b, Z). By Lemma B.2, the process is an MDP on the state space $\Delta(\mathcal{X}) \times 2^{\mathcal{E}}$. Hence the optimal value function satisfies a Bellman recursion; below we write the finite-horizon recursion explicitly.

B.2 Proof of Theorem 4.1

We prove Theorem 4.1 in two parts: (i) existence of an optimal policy of the form $\pi^*(\cdot \mid b_t, Z_t)$; (ii) strict suboptimality of policies that ignore Z_t or ignore b_t (by explicit constructions), under non-trivial $\mathcal{P}_K$ in the sense of Definition B.1.

[1] In the main text we abused notation and wrote $O_{t+1} \subseteq G$ directly. Equation (21) makes this mapping explicit and accommodates more general observation structures.

Standing assumptions for the theorem. Throughout the proof, assume: (i) finite horizon T; (ii) finite action set $\mathcal{U}$; (iii) bounded measurable reward. These are standard sufficient conditions for existence of optimal deterministic Markov policies in finite-horizon MDPs [70, 71].

Part I: Existence of an optimal Markov policy measurable w.r.t. (b_t, Z_t)

Reward model. Let the one-step reward be a bounded measurable function of the underlying latent state and evidence memory:

$$r_t(x_t, Z_t, u_t) \in \mathbb{R}, \qquad |r_t(x, Z, u)| \le R_{\max} < \infty. \tag{27}$$

Define the belief-conditioned expected reward as

$$r_t(b, Z, u) \triangleq \mathbb{E}_{x\sim b}\big[r_t(x, Z, u)\big] = \sum_{x\in\mathcal{X}} b(x)\, r_t(x, Z, u). \tag{28}$$

Finite-horizon dynamic programming. Define the optimal value function at time t as

$$V_t^*(b, Z) \triangleq \sup_{\pi}\ \mathbb{E}^{\pi}\Big[\sum_{k=t}^{T} r_k(b_k, Z_k, u_k)\ \Big|\ b_t = b,\ Z_t = Z\Big], \tag{29}$$

with terminal condition $V_{T+1}^*(\cdot) \equiv 0$. By Lemma B.2, the state is $s_t = (b_t, Z_t)$ and the observation O_{t+1} is generated from $\Omega(\cdot \mid b_t, u_t)$. Therefore the Bellman recursion holds:

$$V_t^*(b, Z) = \max_{u\in\mathcal{U}} \mathbb{E}_{O\sim\Omega(\cdot|b,u)}\Big[r_t(b, Z, u) + V_{t+1}^*\Big(\mathrm{Bayes}(b, u, O),\ \mathcal{P}_K(Z \cup \Gamma(O))\Big)\Big]. \tag{30}$$

Equation (30) is the dynamic programming recursion on the Markov state (b, Z). By standard finite-horizon MDP theory (existence of optimal deterministic Markov policies; see, e.g., [70, 71]), there exists an optimal deterministic Markov policy

$$\pi_t^* : \Delta(\mathcal{X}) \times 2^{\mathcal{E}} \to \mathcal{U} \tag{31}$$

such that $u_t = \pi_t^*(b_t, Z_t)$ maximizes (30) for each t. Equivalently, there exists an optimal (possibly stochastic) policy of the form

$$u_t \sim \pi_t^*(\cdot \mid b_t, Z_t). \tag{32}$$

This establishes Part I.

Part II: Strict suboptimality when ignoring Z_t (belief-only policies)

We prove: if $\mathcal{P}_K$ is non-trivial (Definition B.1), then there exists a finite-horizon instance for which any policy restricted to $u_t \sim \pi(\cdot \mid b_t)$ (i.e., ignoring Z_t) is strictly suboptimal.

Key idea. Non-triviality means there exist $Z \subseteq Z'$ such that $\mathcal{P}_K(Z) \neq \mathcal{P}_K(Z')$. We will construct two reachable information states with the *same belief* b but different evidence memories $Z^{(1)}$ and $Z^{(2)}$, such that the optimal action differs. This implies any belief-only policy must make the same decision at both states and thus is strictly suboptimal.

Step 1: Extract a witness of non-triviality. Since $\mathcal{P}_K$ is non-trivial, choose sets $Z \subseteq Z'$ such that

$$\mathcal{P}_K(Z) \neq \mathcal{P}_K(Z'). \tag{33}$$

Define

$$Z^{(1)} \triangleq \mathcal{P}_K(Z), \qquad Z^{(2)} \triangleq \mathcal{P}_K(Z'). \tag{34}$$

Then $Z^{(1)} \neq Z^{(2)}$. By budget-feasibility (22), both $Z^{(1)}$ and $Z^{(2)}$ satisfy $|Z^{(i)}| \le K$.

Step 2: Define a minimal POMDP instance. We now define a horizon-$T = 2$ instance.[2]

Latent states. Let $\mathcal{X} = \{x_A, x_B\}$. Assume latent dynamics are stationary and action-independent for simplicity:

$$P(x' \mid x, u) = \mathbb{I}[x' = x], \qquad \forall x, x' \in \mathcal{X},\ u \in \mathcal{U}, \tag{35}$$

[2]We use $T = 2$ to avoid "single-step composite reward" ambiguities. The same separation can be written with $T = 1$ by folding the second-step reward into the first, but $T = 2$ gives a cleaner exposition.

so x_t is constant over time.

Belief. Fix any

$$b(x_A) = p, \qquad b(x_B) = 1 - p, \tag{36}$$

with $p \in (1/2, 1)$.

Actions. Let $\mathcal{U} = \{\mathsf{stop}, \mathsf{retrieve}\}$. The process terminates after the second step; stop at $t = 1$ immediately terminates (with terminal reward at $t = 1$).

Observations and evidence extraction. Let $\mathcal{O}$ contain a special "null" observation $O = \varnothing$ and a family of observations whose extracted evidence equals the witness set Z'. Specifically, define $\Gamma : \mathcal{O} \to 2^{\mathcal{E}}$ so that

$$\Gamma(\varnothing) = \varnothing, \qquad \exists O^\star \in \mathcal{O} : \ \Gamma(O^\star) = Z'. \tag{37}$$

Define the observation kernel:

$$\Omega(O = \varnothing \mid x, \mathsf{stop}) = 1, \qquad \Omega(O = O^\star \mid x, \mathsf{retrieve}) = 1, \quad \forall x \in \mathcal{X}. \tag{38}$$

Thus, retrieve deterministically returns an observation that proposes evidence Z' (independent of x), while stop yields no new evidence. This is allowed: the observation need not reveal the latent state; it is merely an evidence acquisition outcome.

Evidence update. The update is

$$Z_{t+1} = \mathcal{P}_K(Z_t \cup \Gamma(O_{t+1})). \tag{39}$$

Initial evidence memory and reachability. We specify the initial evidence memory as a *randomized but observed* variable:

$$\mathbb{P}(Z_1 = Z^{(1)}) = \mathbb{P}(Z_1 = Z^{(2)}) = \tfrac{1}{2}, \qquad Z_1 \ \perp \ x_1. \tag{40}$$

The agent observes Z_1 as part of $\tilde{h}_1$ (17). Because Z_1 is independent of x_1, conditioning on Z_1 does not change the posterior:

$$\mathbb{P}(x_1 = x_A \mid \tilde{h}_1) = \mathbb{P}(x_1 = x_A) = p, \tag{41}$$

so in both cases the belief equals (36). Therefore the two information states

$$(b, Z^{(1)}) \quad \text{and} \quad (b, Z^{(2)}) \tag{42}$$

are both *reachable at time* $t = 1$ and share the same belief b.

Rewards. Define a terminal reward depending on whether the retained evidence matches the latent state: pick two disjoint indicator subsets of edges $E_A, E_B \subseteq \mathcal{E}$ with the property that

$$\mathbb{I}[Z^{(1)} \cap E_A \neq \varnothing] = 1, \quad \mathbb{I}[Z^{(1)} \cap E_B \neq \varnothing] = 0, \qquad \mathbb{I}[Z^{(2)} \cap E_A \neq \varnothing] = 0, \quad \mathbb{I}[Z^{(2)} \cap E_B \neq \varnothing] = 1. \tag{43}$$

Such E_A, E_B always exist because $Z^{(1)} \neq Z^{(2)}$; for example, choose an element $e_1 \in Z^{(1)} \setminus Z^{(2)}$ (if nonempty) and set $E_A = \{e_1\}$, and similarly choose $e_2 \in Z^{(2)} \setminus Z^{(1)}$ and set $E_B = \{e_2\}$. If one difference set is empty, swap roles or choose appropriate singleton indicators using symmetric difference. Now define

$$R(x, Z) = \mathbb{I}[x = x_A] \cdot \mathbb{I}[Z \cap E_A \neq \varnothing] \ + \ \mathbb{I}[x = x_B] \cdot \mathbb{I}[Z \cap E_B \neq \varnothing]. \tag{44}$$

Let $\lambda > 0$ be a retrieval cost. Set one-step rewards as:

$$r_1(x, Z, u) = \begin{cases} R(x, Z), & u = \mathsf{stop}, \\ -\lambda, & u = \mathsf{retrieve}, \end{cases} \qquad r_2(x, Z, u) = \begin{cases} R(x, Z), & u = \mathsf{stop}, \\ 0, & u = \mathsf{retrieve}, \end{cases} \tag{45}$$

and constrain the second step to be stop after retrieving (or, equivalently, allow both actions but note that any further retrieve is dominated since horizon ends). Thus: if the agent stops at $t = 1$, it gets terminal reward immediately; if it retrieves at $t = 1$, it pays cost $-\lambda$, updates Z_2, and then stops at $t = 2$ to receive $R(x, Z_2)$.

Step 3: Compute values and separate optimal actions. Consider time $t = 1$.

If $u_1 =$ *stop*. Given belief b and evidence Z, the expected reward is

$$V(\textsf{stop} \mid b, Z) = \mathbb{E}_{x\sim b}[R(x, Z)] = b(x_A)\,\mathbb{I}[Z \cap E_A \neq \varnothing] + b(x_B)\,\mathbb{I}[Z \cap E_B \neq \varnothing]. \tag{46}$$

Using (43) and (36), we get

$$V(\textsf{stop} \mid b, Z^{(1)}) = p \cdot 1 + (1-p) \cdot 0 = p, \tag{47}$$

$$V(\textsf{stop} \mid b, Z^{(2)}) = p \cdot 0 + (1-p) \cdot 1 = 1 - p. \tag{48}$$

If $u_1 =$ *retrieve*. Then $O_2 = O^\star$ deterministically, so $\Gamma(O_2) = Z'$ by (37), and thus

$$Z_2 = \mathcal{P}_K(Z_1 \cup Z') = \mathcal{P}_K(Z') = Z^{(2)}, \qquad \text{for either } Z_1 \in \{Z^{(1)}, Z^{(2)}\}, \tag{49}$$

where the last equality uses (34) and the idempotence-on-range property $\mathcal{P}_K(\mathcal{P}_K(S)) = \mathcal{P}_K(S)$ which holds for standard projections (top-K, truncation, etc.).[3] Hence, after retrieval, the agent reaches evidence $Z^{(2)}$ regardless of whether it started from $Z^{(1)}$ or $Z^{(2)}$. The expected total value of retrieving at $t = 1$ is

$$\begin{aligned} V(\textsf{retrieve} \mid b, Z_1) &= -\lambda + \mathbb{E}_{x\sim b}[R(x, Z_2)] \\ &= -\lambda + V(\textsf{stop} \mid b, Z^{(2)}) \\ &= -\lambda + (1-p), \end{aligned} \tag{50}$$

using (48).

Step 4: Choose parameters to ensure different optimal actions. We want stop to be uniquely optimal at $(b, Z^{(1)})$ while retrieve is uniquely optimal at $(b, Z^{(2)})$. From (47) and (50), the condition for stop to dominate at $Z^{(1)}$ is:

$$p > (1-p) - \lambda \quad \Longleftrightarrow \quad \lambda > 1 - 2p. \tag{51}$$

Since $p > 1/2$, the right-hand side is negative; thus (51) holds for any $\lambda > 0$. Next, for retrieve to dominate at $Z^{(2)}$, compare (48) with (50):

$$(1-p) - \lambda > 1 - p \quad \Longleftrightarrow \quad -\lambda > 0, \tag{52}$$

which is impossible. Thus, a retrieval action that always maps to $Z^{(2)}$ cannot separate the two optimal actions.

Refined construction: map retrieve *to* $Z^{(1)}$. To obtain a true separation (different optimal actions across $Z^{(1)}$ and $Z^{(2)}$ with the same belief), define the retrieval observation to propose evidence Z (the smaller set) rather than Z'. Specifically, modify (37)–(38) by choosing $O^\diamond \in \mathcal{O}$ such that $\Gamma(O^\diamond) = Z$ and setting

$$\Omega(O = O^\diamond \mid x, \textsf{retrieve}) = 1, \quad \forall x \in \mathcal{X}, \qquad \Omega(O = \varnothing \mid x, \textsf{stop}) = 1. \tag{53}$$

Then, for $Z_1 = Z^{(1)} = \mathcal{P}_K(Z)$ we have

$$Z_2 = \mathcal{P}_K(Z^{(1)} \cup Z) = \mathcal{P}_K(Z) = Z^{(1)}, \tag{54}$$

and for $Z_1 = Z^{(2)} = \mathcal{P}_K(Z')$ we have

$$Z_2 = \mathcal{P}_K(Z^{(2)} \cup Z). \tag{55}$$

Because $Z \subseteq Z'$ and $Z^{(2)} = \mathcal{P}_K(Z')$, in common projections one can have Z_2 change by adding Z (e.g., if Z contains high-priority edges absent from $Z^{(2)}$). To avoid reliance on projection-specific properties, we proceed with a fully general separation by *selecting a retrieval observation* that maps $(Z^{(2)}, \cdot)$ to $Z^{(1)}$.

[3] If one prefers not to assume idempotence, one can redefine $Z^{(2)}$ as $\mathcal{P}_K(Z^{(i)} \cup Z')$ for each i and carry the separation through; idempotence is standard and holds for all common budget projections used in practice.

General separation construction without additional assumptions on $\mathcal{P}_K$. Define an action set $\mathcal{U} = \{\textsf{stop}, \textsf{retrieve-to-}Z^{(1)}\}$, where the retrieval action deterministically sets the next evidence memory to $Z^{(1)}$.[4] Formally, define $\mathcal{O}$ and Γ so that there exists $O^\dagger$ with $\Gamma(O^\dagger) = S$ and

$$\mathcal{P}_K(Z^{(1)} \cup S) = Z^{(1)}, \qquad \mathcal{P}_K(Z^{(2)} \cup S) = Z^{(1)}. \tag{56}$$

Then define

$$\Omega(O = O^\dagger \mid x, \textsf{retrieve-to-}Z^{(1)}) = 1, \quad \forall x \in \mathcal{X}, \qquad \Omega(O = \varnothing \mid x, \textsf{stop}) = 1. \tag{57}$$

Under (56), retrieval maps both starting evidence memories to $Z^{(1)}$ at the next time step.

Completing the separation. With the above, if the agent retrieves at $t = 1$, then $Z_2 = Z^{(1)}$ regardless of $Z_1 \in \{Z^{(1)}, Z^{(2)}\}$, and the retrieve value becomes

$$V(\textsf{retrieve} \mid b, Z_1) = -\lambda + V(\textsf{stop} \mid b, Z^{(1)}) = -\lambda + p. \tag{58}$$

Meanwhile stop values remain (47)–(48).

Choose any λ such that

$$0 < \lambda < 2p - 1. \tag{59}$$

Such λ exists since $p > 1/2$ implies $2p - 1 > 0$. Then:

- At $(b, Z^{(1)})$: stop is uniquely optimal because

$$V(\textsf{stop} \mid b, Z^{(1)}) - V(\textsf{retrieve} \mid b, Z^{(1)}) = p - (p - \lambda) = \lambda > 0.$$

- At $(b, Z^{(2)})$: retrieve is uniquely optimal because

$$V(\textsf{retrieve} \mid b, Z^{(2)}) - V(\textsf{stop} \mid b, Z^{(2)}) = (p - \lambda) - (1 - p) = 2p - 1 - \lambda > 0.$$

Thus, two reachable information states share the same belief b but require different optimal actions. Any belief-only policy $\pi(\cdot \mid b)$ must choose the same action at both states and is therefore strictly suboptimal on this instance.

Conclusion of Part II. We have constructed (using only the non-triviality witness (33)) an instance in which ignoring Z_t is strictly suboptimal. This proves Part II.

Part III: Strict suboptimality when ignoring b_t (evidence-only policies)

We prove: under non-trivial $\mathcal{P}_K$, there exists a finite-horizon instance for which any policy restricted to $u_t \sim \pi(\cdot \mid Z_t)$ (ignoring b_t) is strictly suboptimal.

Key idea. We fix the evidence memory Z to be identical across two reachable information states, but ensure the beliefs differ due to an *auxiliary observation channel* that affects the posterior over x while not altering Z. Then we choose rewards so that the optimal action depends on the belief, forcing evidence-only policies to fail.

Step 1: Instance definition (explicit auxiliary channel). Let horizon $T = 2$. Let latent states $\mathcal{X} = \{x_A, x_B\}$ with static dynamics (35). Fix a budget-feasible non-trivial projection $\mathcal{P}_K$.

Actions. Let $\mathcal{U} = \{\textsf{stop}, \textsf{retrieve}\}$.

Observations. Define observations as pairs

$$O_t = (O_t^{\text{KG}}, O_t^{\text{aux}}), \qquad O_t^{\text{KG}} \in \mathcal{O}_{\text{KG}}, \;\; O_t^{\text{aux}} \in \{0, 1\}. \tag{60}$$

Let $\Gamma(O_t) = \Gamma_{\text{KG}}(O_t^{\text{KG}})$ depend *only* on the KG component, so that the evidence update ignores O^{aux}:

$$Z_{t+1} = \mathcal{P}_K\big(Z_t \cup \Gamma_{\text{KG}}(O_{t+1}^{\text{KG}})\big). \tag{61}$$

Observation kernel. We define:

[4] This is implementable by letting the observation propose a sufficiently rich candidate set S such that $\mathcal{P}_K(Z^{(2)} \cup S) = Z^{(1)}$. Because $\mathcal{O}$ is an abstract observation space under the control of the environment, we can define such an observation outcome in the constructed instance. This does not violate any POMDP axioms.

- Under stop: $O^{\mathrm{KG}}_{t+1} = \varnothing$ deterministically, so it does not change Z.
- Under retrieve: O^{KG}_{t+1} deterministically proposes some candidate set S (details irrelevant) which updates Z in a fixed known way; we will set the initial Z and/or design S so that Z remains fixed at the decision point where separation happens.
- The auxiliary bit O^{aux}_{t+1} depends on the latent state:

$$\mathbb{P}(O^{\mathrm{aux}}_{t+1} = 1 \mid x_A, u) = \alpha, \qquad \mathbb{P}(O^{\mathrm{aux}}_{t+1} = 1 \mid x_B, u) = \beta, \tag{62}$$

  with constants $\alpha \neq \beta$ in $(0, 1)$ (e.g., $\alpha = 0.9, \beta = 0.1$), for both actions $u \in \mathcal{U}$.

Because $\alpha \neq \beta$, observing O^{aux} changes the posterior b_t by Bayes' rule (19), while Z_t is unaffected by (61) when $O^{\mathrm{KG}} = \varnothing$.

Reachability of two distinct beliefs at fixed Z. Fix an initial evidence memory $Z_1 = Z$ deterministically and known to the agent. Let the agent take stop at $t = 1$, so $O^{\mathrm{KG}}_2 = \varnothing$ and hence $Z_2 = Z_1 = Z$. However, O^{aux}_2 can be either 1 or 0 with positive probability under each latent state due to (62). Therefore, the two histories

$$\tilde{h}^{(1)}_2 : \ O^{\mathrm{aux}}_2 = 1, \ Z_2 = Z \qquad \text{and} \qquad \tilde{h}^{(0)}_2 : \ O^{\mathrm{aux}}_2 = 0, \ Z_2 = Z$$

are both reachable and produce two different posterior beliefs:

$$b^{(1)}(x_A) = \mathbb{P}(x_A \mid O^{\mathrm{aux}}_2 = 1), \qquad b^{(0)}(x_A) = \mathbb{P}(x_A \mid O^{\mathrm{aux}}_2 = 0), \tag{63}$$

with $b^{(1)}(x_A) \neq b^{(0)}(x_A)$ because $\alpha \neq \beta$. Thus we obtain two reachable information states with *identical evidence memory Z* but different beliefs.

Step 2: Choose rewards so optimal action depends on belief. Fix any $\lambda \in (0, 1)$ and define terminal reward $R(x, Z)$ as follows. Choose any two disjoint indicator subsets $E_A, E_B \subseteq \mathcal{E}$ and fix Z such that

$$\mathbb{I}[Z \cap E_A \neq \varnothing] = 0, \qquad \mathbb{I}[Z \cap E_B \neq \varnothing] = 1. \tag{64}$$

This is always possible by picking E_B as a singleton edge contained in Z and E_A as a singleton edge not contained in Z. Let $R(x, Z)$ be defined as in (44). As before, define rewards at the decision time (now $t = 2$) by:

$$\begin{aligned} &\textsf{stop} : \ \text{receive } R(x, Z), \\ &\textsf{retrieve} : \ \text{pay } -\lambda \text{ and move to evidence } Z^{\mathrm{good}} \text{ that satisfies} \\ &\mathbb{I}[Z^{\mathrm{good}} \cap E_A \neq \varnothing] = 1, \ \mathbb{I}[Z^{\mathrm{good}} \cap E_B \neq \varnothing] = 0, \end{aligned} \tag{65}$$

which can be implemented by defining a retrieval observation that forces Z to update to Z^{good} under $\mathcal{P}_K$ (analogously to Part II; again, $\mathcal{O}$ is abstract and can be defined accordingly in the constructed instance).

Then, for any belief b at this decision point, we have:

$$V(\textsf{stop} \mid b, Z) = \mathbb{E}_{x \sim b}[R(x, Z)] = 1 - b(x_A), \tag{66}$$

$$V(\textsf{retrieve} \mid b, Z) = -\lambda + \mathbb{E}_{x \sim b}[R(x, Z^{\mathrm{good}})] = -\lambda + b(x_A). \tag{67}$$

Taking the difference:

$$V(\textsf{retrieve} \mid b, Z) - V(\textsf{stop} \mid b, Z) = 2b(x_A) - 1 - \lambda. \tag{68}$$

Step 3: Separate optimal actions across the two reachable beliefs. Because $b^{(1)}(x_A) \neq b^{(0)}(x_A)$ and both lie in $(0, 1)$, we can choose channel parameters (α, β) and/or the prior so that

$$b^{(1)}(x_A) > \tfrac{1+\lambda}{2} \qquad \text{and} \qquad b^{(0)}(x_A) < \tfrac{1+\lambda}{2}. \tag{69}$$

Then by (68), retrieve is uniquely optimal under belief $b^{(1)}$ while stop is uniquely optimal under belief $b^{(0)}$, even though the evidence memory state is identical (Z) in both cases. Therefore any evidence-only policy $\pi(\cdot \mid Z)$ must choose the same action for both beliefs and is strictly suboptimal on this instance.

Conclusion of Part III. We have constructed an explicit instance with two reachable information states sharing the same Z but differing in b, for which the optimal action depends on b. Hence ignoring b_t can be strictly suboptimal.

Overall conclusion. Parts I–III jointly establish Theorem 4.1: (i) an optimal policy exists that conditions on (b_t, Z_t) (Markov optimality on the belief-MDP state); (ii) under non-trivial $\mathcal{P}_K$, there exist instances where any policy ignoring either b_t or Z_t is strictly suboptimal. □

C Unified Separation Template and Instantiations

C.1 A separation template for constrained policy classes

This appendix provides constructive separations that instantiate the unified regret template. All constructions share the same belief-MDP dynamics and differ only in the admissible policy class and/or admissible action families. Throughout, $T \geq 2$ denotes a finite horizon.

Evaluation protocol. We consider terminal reward in $[0, 1]$ and evaluate policies from a fixed initial information-state distribution ρ_θ over (b_0, Z_0). Let $V^\pi(\mathcal{M}_\theta)$ denote the expected return of policy π on instance $\mathcal{M}_\theta$:

$$V^\pi(\mathcal{M}_\theta) \triangleq \mathbb{E}_{(b_0,Z_0)\sim\rho_\theta}\Big[\sum_{t=0}^{T-1} r_t\Big]. \tag{70}$$

Throughout this appendix we use terminal correctness reward:

$$r_t = 0 \;\; (t < T-1), \qquad r_{T-1} = \mathbb{I}[\hat{a} = a^\star] \in \{0, 1\}, \tag{71}$$

so $V^\pi(\mathcal{M}_\theta) = \Pr_\pi(\hat{a} = a^\star)$.

Mixture task family. Let $(\Theta, \mathcal{F}, \mu)$ be a probability space indexing an instance family $\{\mathcal{M}_\theta\}_{\theta\in\Theta}$, where μ is a *fixed prior* over instances. Define the mixture task $\bar{\mathcal{M}}$ by sampling $\theta \sim \mu$ once at the beginning (unknown to the policy), then running $\mathcal{M}_\theta$.

Policy interfaces and admissibility. We view a policy as a pair $\pi = (\pi^{\mathrm{ret}}, \pi^{\mathrm{gen}})$: at each step t it selects a retrieval/update action and then generates a reasoning artifact. We allow action admissibility constraints that depend on the current information state. Formally, for each instance $\mathcal{M}_\theta$ we fix: (i) an admissible action set mapping $(b_t, Z_t, y_{1:t}) \mapsto \mathcal{U}_\theta(b_t, Z_t, y_{1:t}) \subseteq \mathcal{U}$, and (ii) an observation kernel $\Omega_\theta(\cdot \mid x, u)$ supported on admissible observations for any admissible u. A policy is *admissible* if it only selects actions in the admissible set almost surely:

$$u_t \sim \pi^{\mathrm{ret}}(\cdot \mid q, y_{1:t}, b_t, Z_t), \qquad \Pr\big(u_t \in \mathcal{U}_\theta(b_t, Z_t, y_{1:t})\big) = 1, \; \forall t. \tag{72}$$

This convention lets us express locality/tool-feasibility constraints as restrictions on $\mathcal{U}_\theta(\cdot)$ while keeping the belief-MDP recursion unchanged.

Capability-based restriction. Let Π be a constrained policy class. We say Π *lacks a capability* C if there exists a measurable restriction map $\mathcal{R}_\Pi$ (dropping information required for C) such that every $\pi \in \Pi$ is invariant to this restriction in its retrieval decisions:

$$\pi^{\mathrm{ret}}(\cdot \mid q, y_{1:t}, b_t, Z_t) = \pi^{\mathrm{ret}}(\cdot \mid \mathcal{R}_\Pi(q, y_{1:t}, b_t, Z_t)), \qquad \forall t. \tag{73}$$

In the taxonomy of Section 4, we instantiate four capability failures:

- **Adaptivity failure** (static): actions cannot depend on intermediate disambiguation signals.
- **Cross-region access failure** (local): retrieval cannot target evidence outside a bounded neighborhood of the current evidence.
- **Soft-uncertainty failure** (hard): "unsupported but plausible" intermediate hypotheses cannot be maintained or used to trigger retrieval.
- **Budget-maintenance failure** (coarse): evidence projection is claim-independent, causing dilution under tight budgets.

Lemma C.1 (Unified separation template (mixture regret))**.** *Fix a constrained class* Π *and a constant* $\Delta \in (0,1)$*. Suppose there exists a task family* $\{\mathcal{M}_\theta\}_{\theta\in\Theta}$*, a prior* μ *over* Θ*, and a constant* $\beta \in (0,1]$ *such that:*

1. *(**Feasible optimality in** Π_{res}**)** There exists a* single *policy* $\pi^\star \in \Pi_{\mathrm{res}}$ *such that for* μ*-a.e.* θ*,*
$$V^{\pi^\star}(\mathcal{M}_\theta) = 1. \tag{74}$$
2. *(**Capability obstruction for** Π**)** For every* $\pi \in \Pi$*, there exists a measurable subset* $\Theta_\pi \subseteq \Theta$ *with*
$$\mu(\Theta_\pi) \geq \beta \qquad \text{and} \qquad \forall \theta \in \Theta_\pi :\ V^\pi(\mathcal{M}_\theta) \leq 1 - \Delta. \tag{75}$$

Then for the mixture task $\bar{\mathcal{M}}$ *induced by* μ*,*

$$\sup_{\pi\in\Pi} V^\pi(\bar{\mathcal{M}}) \leq 1 - \beta\Delta, \qquad \textit{while} \qquad \sup_{\pi\in\Pi_{\mathrm{res}}} V^\pi(\bar{\mathcal{M}}) = 1. \tag{76}$$

Equivalently, $\sup_{\pi\in\Pi} V^\pi(\bar{\mathcal{M}}) \leq V^*(\bar{\mathcal{M}}) - \beta\Delta$ *with* $V^*(\bar{\mathcal{M}}) = 1$*.*

Proof. By (74), for μ-a.e. θ we have $V^{\pi^\star}(\mathcal{M}_\theta) = 1$. Therefore

$$\sup_{\pi\in\Pi_{\mathrm{res}}} V^\pi(\bar{\mathcal{M}}) \geq V^{\pi^\star}(\bar{\mathcal{M}}) = \mathbb{E}_{\theta\sim\mu}\big[V^{\pi^\star}(\mathcal{M}_\theta)\big] = 1, \tag{77}$$

and hence $\sup_{\pi\in\Pi_{\mathrm{res}}} V^\pi(\bar{\mathcal{M}}) = 1$ since values are bounded by 1. Next fix any $\pi \in \Pi$. By (75), there exists Θ_π with $\mu(\Theta_\pi) \geq \beta$ such that $V^\pi(\mathcal{M}_\theta) \leq 1 - \Delta$ for all $\theta \in \Theta_\pi$. Since $V^\pi(\mathcal{M}_\theta) \leq 1$ always, taking expectation yields

$$\begin{aligned} V^\pi(\bar{\mathcal{M}}) &= \mathbb{E}_{\theta\sim\mu}\big[V^\pi(\mathcal{M}_\theta)\big] \\ &\leq (1-\Delta)\cdot\mu(\Theta_\pi) + 1\cdot(1-\mu(\Theta_\pi)) = 1 - \Delta\cdot\mu(\Theta_\pi) \leq 1 - \beta\Delta. \end{aligned} \tag{78}$$

Taking supremum over $\pi \in \Pi$ preserves the upper bound. Finally, since $V^*(\bar{\mathcal{M}}) \geq \sup_{\pi\in\Pi_{\mathrm{res}}} V^\pi(\bar{\mathcal{M}}) = 1$ and values are at most 1, we have $V^*(\bar{\mathcal{M}}) = 1$. □

Remarks and empirical grounding. The separations below mirror concrete limitations observed in dominant KG+LLM paradigms: static one-shot KG-RAG and budgeted subgraph selection [10, 36, 37], bounded traversal agents [44, 22, 45, 46], hard verifiability constraints [14, 23], and add-only / coarse iterative RAG loops [61, 64, 66]. While our proofs are abstract, each capability failure corresponds to an explicit, checkable restriction in the policy definitions in Section 4.

C.2 Separation for static retrieve-then-reason

Static retrieve-then-reason corresponds to policies that never perform retrieval/update actions during the horizon and only reason over the fixed initial evidence Z_0, covering one-shot KG-RAG and budgeted subgraph selection baselines [10, 36, 37]. Formally,

$$\Pi_{\mathrm{static}} \triangleq \Big\{\pi :\ \Pr(u_t = \varnothing) = 1,\ \forall t \in \{0,\dots,T-1\}\Big\}, \tag{79}$$

where $u = \varnothing$ denotes a no-op action producing $O = \emptyset$ and leaving Z unchanged.

Instance family. Let latent state $x \in \{0,1\}$ with prior $\Pr(x=0) = \Pr(x=1) = 1/2$. There are two disjoint KG "regions" A and B with critical edges $e_A \in A$ and $e_B \in B$. The correct answer is $a^\star(x) = A$ if $x = 0$ and $a^\star(x) = B$ if $x = 1$. The evidence budget is $K = 1$, so only one critical edge can be retained.

To make the decoding explicit, we consider a separation decoder that reads only the final evidence state:

$$\hat{a} = g(Z_T), \qquad g(\{e_A\}) = A, \quad g(\{e_B\}) = B, \quad g(\varnothing) = \perp, \tag{80}$$

and terminal reward is $r_{T-1} = \mathbb{I}[\hat{a} = a^\star(x)]$. Since Π_{static} never updates evidence, we have $Z_T = Z_0$ almost surely under any $\pi \in \Pi_{\mathrm{static}}$.

We allow non-static policies to use action space $\mathcal{U} = \{\mathsf{probe}, \mathsf{getA}, \mathsf{getB}, \varnothing\}$:

- probe returns an observation $O = \{o_x\}$ that reveals x and does not affect Z under projection (i.e., $\mathcal{P}_1(Z \cup \{o_x\}) = Z$);
- getA returns $O = \{e_A\}$ and getB returns $O = \{e_B\}$.

The initial distribution ρ_θ is defined independently of x and encodes the static evidence choice: any $\pi \in \Pi_{\text{static}}$ can be seen as selecting a (possibly randomized) Z_0 as a function of q, while x remains unobserved at $t = 0$.

Optimal value. A closed-loop policy achieves correctness 1: choose $u_0 = \text{probe}$ to observe x, then choose $u_1 = \text{getA}$ if $x = 0$ and $u_1 = \text{getB}$ if $x = 1$. Since $K = 1$, the final evidence contains the correct critical edge with probability 1, hence $V^* = 1$.

Upper bound for static policies. Any $\pi \in \Pi_{\text{static}}$ must choose Z_0 without observing x and cannot update evidence thereafter. Since $K = 1$, Z_0 can contain at most one of $\{e_A, e_B\}$. Therefore, for any $\pi \in \Pi_{\text{static}}$,

$$\begin{aligned} V^\pi &= \Pr(\hat{a} = a^\star(x)) \\ &= \Pr(x = 0) \cdot \Pr(Z_0 = \{e_A\}) + \Pr(x = 1) \cdot \Pr(Z_0 = \{e_B\}) \\ &\le \max\{\Pr(x = 0), \Pr(x = 1)\} = \tfrac{1}{2}. \end{aligned} \tag{81}$$

Thus $\sup_{\pi \in \Pi_{\text{static}}} V^\pi \le 1/2$, yielding a regret lower bound $\Delta_{\text{static}} \ge 1/2$. Moreover, this bound is tight by choosing $Z_0 = \{e_A\}$ (or $\{e_B\}$) deterministically.

C.3 Separation for local traversal / frontier expansion agents

The constrained class $\Pi_{\text{local}}(h)$ abstracts bounded traversal agents that expand evidence locally along edges or frontiers [44, 22, 45, 46]. We make explicit that $\Pi_{\text{local}}(h)$ *prohibits* teleport-style entity-conditioned queries to unseen regions; this restriction is what makes local traversal distinct from residual-coupled, claim-conditioned retrieval.

Action model for locality. Let $\mathsf{Ent}(Z)$ denote the set of entities appearing in edge set Z. For a set of entities $S \subseteq \mathcal{V}$, let $\text{Ball}_h(S)$ denote the set of entities within graph distance at most h from S in G. Define the h-hop neighborhood edge set

$$\text{Nbr}_h(S) \triangleq \big\{(u, r, v) \in G : \ u \in \text{Ball}_h(S) \ \text{ or } \ v \in \text{Ball}_h(S)\big\}. \tag{82}$$

Local traversal is expressed as an admissible-action restriction: at any step t, an action is admissible only if its induced observation support lies within $\text{Nbr}_h(\mathsf{Ent}(Z_t))$. Formally, define

$$\mathcal{U}_{\text{local}}(Z_t) \triangleq \Big\{u \in \mathcal{U} : \ \text{supp}\big(\Omega(\cdot \mid x, u)\big) \subseteq 2^{\text{Nbr}_h(\mathsf{Ent}(Z_t))} \ \text{ for all } x\Big\}, \tag{83}$$

and set $\Pi_{\text{local}}(h)$ as the class of admissible policies that always choose $u_t \in \mathcal{U}_{\text{local}}(Z_t)$ almost surely.

Graph construction. Fix hop limit $h \ge 1$. Construct a KG with two subgraphs A and B such that:

1. The entity sets are disjoint: $\mathcal{V}(A) \cap \mathcal{V}(B) = \varnothing$.
2. Their graph distance is large: for any $u \in \mathcal{V}(A)$ and $v \in \mathcal{V}(B)$, $\text{dist}(u, v) > h$.

Let q mention two anchors $a \in \mathcal{V}(A)$ and $b \in \mathcal{V}(B)$ (a compositional join query). Let the correct answer require *both* a predicate witnessed in A and a predicate witnessed in B:

$$\hat{a} = a^\star \quad \Longleftrightarrow \quad (e_A \in Z_T) \ \wedge \ (e_B \in Z_T), \tag{84}$$

where e_A is an edge incident to a in A and e_B is an edge incident to b in B. Budget is $K = 2$ and $\mathcal{P}_2$ is identity when $|S| \le 2$.

We specify the initial information state distribution to make reachability explicit:

$$\rho(b_0, Z_0) = \delta_{b_0} \otimes \delta_{Z_0^A}, \qquad Z_0^A \subseteq A, \qquad b \notin \mathsf{Ent}(Z_0^A). \tag{85}$$

Thus the initial evidence only covers region A.

Local constraint implies inability to reach B. We show by induction that $Z_t \subseteq A$ for all t under any $\pi \in \Pi_{\mathrm{local}}(h)$. Base case: by (85), $Z_0 \subseteq A$. Inductive step: assume $Z_t \subseteq A$. Then $\mathsf{Ent}(Z_t) \subseteq \mathcal{V}(A)$, hence $\mathrm{Ball}_h(\mathsf{Ent}(Z_t)) \subseteq \mathcal{V}(A)$ by $\mathrm{dist}(A, B) > h$. Therefore $\mathrm{Nbr}_h(\mathsf{Ent}(Z_t)) \subseteq A$. By (83), any admissible action must satisfy $O_{t+1} \subseteq \mathrm{Nbr}_h(\mathsf{Ent}(Z_t)) \subseteq A$ almost surely. Thus $Z_{t+1} = \mathcal{P}_2(Z_t \cup O_{t+1}) \subseteq A$. This completes the induction and implies

$$\forall t, \quad e_B \notin Z_t. \tag{86}$$

Value upper bound for local policies. By (84) and (86), for any $\pi \in \Pi_{\mathrm{local}}(h)$ we have

$$V^\pi = \Pr_\pi\big((e_A \in Z_T) \wedge (e_B \in Z_T)\big) = 0. \tag{87}$$

Residual-coupled feasibility. In contrast, Π_{res} allows entity-conditioned retrieval actions parameterized by anchors/residuals, including those explicitly present in the question q. Formally, include in the admissible action family an operator $\mathsf{QueryNbr}(v)$ such that, for any entity v appearing in q,

$$u_t = \mathsf{QueryNbr}(v) \quad \Rightarrow \quad O_{t+1} \supseteq \mathrm{Nbr}_1(\{v\}). \tag{88}$$

Since b appears in q, choosing $u_t = \mathsf{QueryNbr}(b)$ retrieves e_B directly, while choosing $u_t = \mathsf{QueryNbr}(a)$ retrieves e_A. With $K = 2$, a policy in Π_{res} can retain both edges and achieve $V^\pi = 1$, yielding a separation gap $\Delta_{\mathrm{local}} = 1$.

C.4 Failure Mode of Hard Verifiability under KG Incompleteness

Hard verifiability corresponds to policies that only allow generating reasoning artifacts whose extracted atomic claims are fully supported by the current evidence, a constraint appearing in strict graph-faithful prompting / constrained decoding variants [14, 23]. We formalize a missing-edge failure where a correct solution requires temporarily maintaining an *unsupported but plausible* intermediate hypothesis in order to trigger a tool call.

Instance construction (missing-edge bridge). Let entities be s, m, t and relations be r_1, r_2. The *latent completion* (true world) contains two edges:

$$(s, r_1, m), \qquad (m, r_2, t).$$

However, the observed KG G is incomplete and misses the bridge edge:

$$(s, r_1, m) \notin G, \qquad (m, r_2, t) \in G. \tag{89}$$

The question asks for the terminal entity t given s and the relation composition $r_1 \circ r_2$. A correct solution requires introducing the intermediate variable m (as a hypothesis) and then querying its outgoing r_2-neighbor to reveal t.

Hard support definition. Instantiate Sup by hard entailment from explicit KG edges:

$$\mathrm{Sup}(c; Z) = 1 \iff Z \text{ contains an edge that explicitly entails } c, \qquad \mathrm{Sup}(c; Z) = 0 \text{ otherwise.} \tag{90}$$

Since $Z_t \subseteq G$ for all t, (89) implies that any claim asserting the existence of an r_1-bridge from s is unsupported under any admissible evidence state.

Claim extractor for bridge introduction. Fix a deterministic claim extractor $C(\cdot)$ that, in addition to factual assertions, extracts *tool-binding* claims of the following form: whenever a reasoning artifact y_t introduces an entity token v as the binding argument for a subsequent tool call parameterized by a relation (e.g., "call $\mathsf{OutNbr}(v, r)$"), $C(y_t)$ contains an atomic claim

$$c_{\mathrm{bind}}(v): \text{ "}v \text{ is a valid intermediate binding for the current compositional query."} \tag{91}$$

In particular, introducing the intermediate m as a candidate bridge for r_1 induces the binding claim $c_{\mathrm{bind}}(m)$, which is entailed only if the missing edge (s, r_1, m) is present in Z_t under (90).

Tool-calling admissibility (formal). We model KG interaction via parameterized query actions whose arguments must be *bound* by the reasoning trajectory. Let $\mathsf{Bind}(y_{1:t})$ denote the set of entities that have been introduced as bindings (i.e., for which $c_{\mathrm{bind}}(\cdot)$ has been produced) up to step t. Define an action family $\mathsf{OutNbr}(v, r)$ that returns all edges $(v, r, \cdot) \in G$. Impose the feasibility constraint

$$u_t = \mathsf{OutNbr}(v, r) \text{ is admissible only if } v \in \mathsf{Bind}(y_{1:t}). \tag{92}$$

This matches structured tool-use pipelines where the model must emit the argument binding before the tool can be called.

Hard verifiability implies a dead end. Policies in Π_{hard} must satisfy, almost surely,

$$\forall t,\ \forall c \in C(y_t),\ \mathrm{Sup}(c; Z_t) = 1. \tag{93}$$

Consider the binding claim $c_{\text{bind}}(m)$ associated with introducing m as an intermediate bridge candidate. By construction, $c_{\text{bind}}(m)$ is supported under (90) only if $(s, r_1, m) \in Z_t$, which is impossible since $(s, r_1, m) \notin G$ and $Z_t \subseteq G$. Hence any policy in Π_{hard} cannot introduce m as a binding at any step without violating (93). Therefore $m \notin \mathsf{Bind}(y_{1:t})$ for all t, and by (92) the action $\mathsf{OutNbr}(m, r_2)$ is never admissible. Since $(m, r_2, t) \in G$ is only retrievable via $\mathsf{OutNbr}(m, r_2)$ in this instance, the policy cannot retrieve the edge needed to output t. Thus for any $\pi \in \Pi_{\text{hard}}$,

$$\Pr_{\pi}(\hat{a} = t) = 0 \quad \Rightarrow \quad V^{\pi} = 0, \qquad \text{hence} \quad \sup_{\pi \in \Pi_{\text{hard}}} V^{\pi} = 0. \tag{94}$$

Residual-coupled feasibility. In Π_{res}, the generation and evidence maintenance are claim-aware and allow *residual* (unresolved) hypotheses to be carried without being asserted as supported facts. Concretely, a residual-coupled policy can: (i) introduce m as a *hypothesis binding* (tagged as unresolved so that it is excluded from the hard constraint (93)), (ii) call $\mathsf{OutNbr}(m, r_2)$ to retrieve $(m, r_2, t) \in G$, and (iii) output t. Formally, we assume Π_{res} employs a claim-conditioned extractor that separates *residual bindings* R_t from asserted claims, and tool-feasibility in (92) is satisfied by either asserted or residual bindings. Under this admissibility, there exists $\pi \in \Pi_{\text{res}}$ with $V^{\pi} = 1$, yielding a strict gap $\Delta_{\text{hard}} = 1$. This formalizes the empirical observation that strict verifiability can reduce coverage under incompleteness [66, 23].

C.5 Budget dilution in coarse add-only iterative loops

Π_{add} models iterative retrieval–reasoning loops whose controller state $s_t = f(q, y_{1:t})$ is low-dimensional and whose evidence maintenance is claim-independent and add-only, as in many practical iterative RAG loops [61, 64, 66].

Construction. Let the required edge set be

$$E^{\star} = \{e_1, \ldots, e_K\}, \tag{95}$$

and set the evidence budget to K. Correctness requires retaining all required edges:

$$\hat{a} = a^{\star} \quad \Longleftrightarrow \quad E^{\star} \subseteq Z_T. \tag{96}$$

We set the horizon to exactly $T = K$ and impose that at each round at most one new required edge can be introduced. This makes any "reset" event irrecoverable within the remaining budgeted rounds.

At each round $t \in \{0, \ldots, T-1\}$, the retriever returns an observation

$$O_{t+1} = \{e_{i_t}\} \cup D_{t+1}, \qquad |D_{t+1}| = M, \tag{97}$$

where $e_{i_t} \in E^{\star}$ is a required edge not yet retrieved (so $\{e_{i_t}\} \cap Z_t = \varnothing$) and $D_{t+1} \subseteq H$ consists of M hub/distractor edges from a hub set H with $H \cap E^{\star} = \varnothing$.

Claim-independent projection (coarse maintenance). In Π_{add}, the evidence update is

$$Z_{t+1} = \mathcal{P}_K(Z_t \cup O_{t+1}), \tag{98}$$

where $\mathcal{P}_K$ is *claim-independent*: for any claim sequence $C(y_{1:t})$, $\mathcal{P}_K(\cdot; C(y_{1:t})) \equiv \mathcal{P}_K(\cdot)$. We additionally assume $\mathcal{P}_K$ is hub-biased in the following *uniform conditional* sense: there exists $\alpha \in (0, 1)$ such that for all t and all histories $\mathcal{H}_t$, letting $S_t = Z_t \cup O_{t+1}$, we have

$$\Pr\left(\mathcal{P}_K(S_t) \subseteq H \mid \mathcal{H}_t\right) \geq \alpha. \tag{99}$$

This captures projection rules that, under tight budgets, can consistently prefer high-salience hub edges and drop decision-critical edges because the projection is not conditioned on which claims remain unresolved.

Define the "hub-only" event

$$\mathcal{E}_{t+1} \triangleq \{Z_{t+1} \subseteq H\}. \tag{100}$$

By (99), for all t,

$$\Pr(\mathcal{E}_{t+1} \mid \mathcal{H}_t) \geq \alpha. \tag{101}$$

Irrecoverability under $T = K$ and one-useful-edge per round. If $\mathcal{E}_{t+1}$ occurs, then Z_{t+1} contains no required edges, i.e., $Z_{t+1} \cap E^\star = \varnothing$. From round $t+1$ to $T-1$, there are only $T-(t+1) \le K-1$ remaining rounds, and each round can introduce at most one new required edge via (97). Therefore it becomes impossible to have collected all K distinct required edges by time T:

$$\mathcal{E}_{t+1} \quad \Rightarrow \quad E^\star \not\subseteq Z_T. \tag{102}$$

Non-vanishing failure probability. By (102),

$$\Pr(E^\star \subseteq Z_T) \le \Pr\Big(\bigcap_{t=0}^{T-1} \mathcal{E}_{t+1}^c\Big). \tag{103}$$

Using the chain rule and (101), we obtain

$$\begin{aligned}\Pr\Big(\bigcap_{t=0}^{T-1} \mathcal{E}_{t+1}^c\Big) &= \prod_{t=0}^{T-1} \Pr\Big(\mathcal{E}_{t+1}^c \mid \bigcap_{j=0}^{t-1} \mathcal{E}_{j+1}^c\Big) \\ &\le \prod_{t=0}^{T-1} (1-\alpha) = (1-\alpha)^T. \end{aligned} \tag{104}$$

Combining (103) and (104) yields

$$\Pr(E^\star \not\subseteq Z_T) \ge 1 - (1-\alpha)^T \triangleq \delta > 0. \tag{105}$$

By (96), for any $\pi \in \Pi_{\mathrm{add}}$,

$$V^\pi = \Pr(\hat{a} = a^\star) = \Pr(E^\star \subseteq Z_T) \le (1-\alpha)^T = 1-\delta, \tag{106}$$

so the gap satisfies $\Delta_{\mathrm{add}} \ge \delta$.

Why residual-coupled avoids dilution (constructive). In Π_{res}, the projection $\mathcal{P}_K(\cdot;\, C(y_{1:t}))$ can be claim-conditioned. Consider the following claim-aware projection rule: maintain an auxiliary set of "locked" edges

$$L_t \triangleq \{e \in Z_t :\ e \text{ is referenced by some residual or asserted claim in } C(y_{1:t})\}, \tag{107}$$

and define

$$\mathcal{P}_K(S;\, C(y_{1:t})) \triangleq L_t \ \cup\ \mathsf{Top}_{K-|L_t|}\big(S \setminus L_t\big), \tag{108}$$

where $\mathsf{Top}_m(\cdot)$ is any deterministic tie-breaking selection of m remaining edges. A residual-coupled policy can generate/maintain residual claims that reference each newly retrieved required edge e_{i_t} upon arrival, thereby ensuring $e_{i_t} \in L_{t+1}$ and it will never be evicted. Since exactly one new required edge arrives per round and $T = K$, by time T all edges in $E^\star$ are locked and retained, while distractors may be discarded. Therefore there exists $\pi \in \Pi_{\mathrm{res}}$ achieving $\Pr(E^\star \subseteq Z_T) = 1$ and hence $V^* = 1$. This matches the empirical motivation for claim-aware evidence maintenance under tight budgets in dynamic subgraph methods [36, 37, 64].

D Proofs for Strict Separations

We restate the theorem (for completeness), introduce a minimal KGQA-POMDP template consistent with Section 3–4, and then give four separation constructions—one for each constrained policy class. Throughout, we use a 0–1 terminal reward (accuracy), so all value gaps are explicit constants.

D.1 Restatement and proof plan

Theorem D.1 (Strict separations in achievable optimal value (restated))**.** *Assume a non-trivial projection $\mathcal{P}_K$ and define $V^*(\Pi) := \sup_{\pi\in\Pi} V^\pi$. For each constrained class $\Pi \in \{\Pi_{\mathrm{static}}, \Pi_{\mathrm{local}}(h), \Pi_{\mathrm{hard}}, \Pi_{\mathrm{add}}\}$, there exists a family of KGQA instances such that*

$$V^*(\Pi) \ \le\ V^* - \Delta_\Pi \tag{109}$$

for some constant $\Delta_\Pi > 0$, while for the same family $V^*(\Pi_{\mathrm{res}}) = V^*$ (or $V^* - \epsilon$).

Proof plan. For each constrained class Π, we will: (i) define an explicit family of KGQA instances $\mathcal{I}_\Pi$ that fits the POMDP formulation in Section 3; (ii) compute V^* for the instance family (i.e., the optimal value over all measurable policies); (iii) upper bound $V^*(\Pi)$ for the constrained class; (iv) exhibit a policy in Π_{res} that achieves V^* on the same family. The gap Δ_Π will be a universal positive constant (we will use $\Delta_\Pi = 1/2$).

D.2 A minimal KGQA-POMDP template consistent with Section 3

We instantiate the abstract framework of Section 3 into a simple, fully analyzable model. This construction is used only for the proof and does not restrict the generality of the separation claim, which is existential.

Terminal reward. We consider a finite-horizon process terminating at time T. The environment samples a latent state $x \in \mathcal{X}$ and induces a unique correct answer $a^\star = a^\star(x) \in \mathcal{A}$. A policy π generates an answer $\hat{a}$ at termination, and receives reward

$$R = \mathbf{1}\{\hat{a} = a^\star\} \in \{0, 1\}. \tag{110}$$

Hence

$$V^\pi := \mathbb{E}^\pi[R] = \mathbb{P}^\pi(\hat{a} = a^\star), \qquad V^* := \sup_\pi V^\pi. \tag{111}$$

Evidence budget and projection. We set $K = 1$ (one-triple evidence budget), and use a deterministic, non-trivial projection $\mathcal{P}_1$. Fix any total order $\prec$ over triples in G (instance-specific but fixed once the instance is fixed), and define

$$\mathcal{P}_1(Z) := \begin{cases} \emptyset, & Z = \emptyset, \\ \{\min_\prec Z\}, & Z \neq \emptyset. \end{cases} \tag{112}$$

This projection is *non-trivial* because for any two distinct triples $\tau \prec \tau'$, taking $Z = \{\tau\}$ and $Z' = \{\tau, \tau'\}$ yields $\mathcal{P}_1(Z) = \{\tau\} \neq \{\tau'\} = \mathcal{P}_1(Z' \setminus \{\tau\})$ and, in general, there exist $Z \subset Z'$ with $\mathcal{P}_1(Z) \neq \mathcal{P}_1(Z')$.

Retriever actions and observations. At each step t, the policy selects an action $u_t \in \mathcal{U}$. The environment returns an observation $O_{t+1} \subseteq G$ according to an observation kernel $\Omega(\cdot \mid x, u_t, G)$, and the evidence state updates as

$$Z_{t+1} = \mathcal{P}_1\big(Z_t \cup O_{t+1}\big). \tag{113}$$

This is a special case of the formulation in Eqs. (6)–(8) of Section 3.

Reasoning trajectory abstraction. We will not require a detailed parametrization of $y_{1:T}$ and $g(\cdot)$. Instead, we use the following *realizability convention* (standard in separations): whenever the necessary supporting triple for the correct answer is present in the final evidence Z_T, there exists a (possibly deterministic) decoding rule g that outputs the correct answer with probability 1; when such evidence is absent, no policy can exceed $1/2$ accuracy on the balanced two-answer constructions below. This convention is consistent with the support-score abstraction and with the role of evidence in answer selection.

D.3 Separation I: Π_{static} is strictly suboptimal

We construct a family $\mathcal{I}_{\text{static}}$ where achieving $V^* = 1$ requires at least two retrieval steps, while Π_{static} forbids any retrieval for $t > 0$.

D.3.1 Instance family $\mathcal{I}_{\text{static}}$

Fix entities e_Q, e_1, e_2 and relations r, r_{id}. Let the KG be

$$G := \{\tau_1, \tau_2, \sigma_1, \sigma_2\}, \qquad \tau_1 = (e_Q, r, e_1),\ \tau_2 = (e_Q, r, e_2),\ \sigma_1 = (e_Q, r_{\text{id}}, \texttt{id}_1),\ \sigma_2 = (e_Q, r_{\text{id}}, \texttt{id}_2), \tag{114}$$

where $\texttt{id}_1, \texttt{id}_2$ are two distinct identifier entities. Let the latent state be $x \in \{1, 2\}$ with $\mathbb{P}(x = 1) = \mathbb{P}(x = 2) = 1/2$. The correct answer is

$$a^\star(x) := \begin{cases} e_1, & x = 1, \\ e_2, & x = 2. \end{cases} \tag{115}$$

Action set $\mathcal{U}$ contains three actions:

$$\mathcal{U} = \{u^{(\mathrm{id})}, u^{(1)}, u^{(2)}\}. \tag{116}$$

The observation kernel Ω is deterministic given (x, u):

$$O_{t+1} = \begin{cases} \{\sigma_x\}, & u_t = u^{(\mathrm{id})}, \\ \{\tau_1\}, & u_t = u^{(1)}, \\ \{\tau_2\}, & u_t = u^{(2)}. \end{cases} \tag{117}$$

We set horizon $T = 2$, initial evidence $Z_0 = \emptyset$, and projection $\mathcal{P}_1$ as in (112).

D.3.2 Optimal value $V^* = 1$

Lemma D.2 (Full policies achieve $V^* = 1$ on $\mathcal{I}_{\mathrm{static}}$)**.** *On the family $\mathcal{I}_{\mathrm{static}}$, $V^* = 1$.*

Proof. Consider the following policy $\pi^\star$: at $t = 0$, select $u_0 = u^{(\mathrm{id})}$, observe $O_1 = \{\sigma_x\}$ by (117). At $t = 1$, if $O_1 = \{\sigma_1\}$ choose $u_1 = u^{(1)}$, else (if $O_1 = \{\sigma_2\}$) choose $u_1 = u^{(2)}$. Then by (117), $O_2 = \{\tau_x\}$ and hence $Z_2 = \{\tau_x\}$ (since $K = 1$ and the union contains at most one relevant triple at the final step). By construction, τ_x is the unique supporting triple for answer e_x, so the decoder can output e_x with probability 1. Therefore $\mathbb{P}^{\pi^\star}(\hat{a} = a^\star) = 1$, i.e., $V^{\pi^\star} = 1$. Since $V^* = \sup_\pi V^\pi \geq V^{\pi^\star} = 1$ and values are at most 1, we conclude $V^* = 1$. □

D.3.3 Π_{static} is bounded by $1/2$

Recall $\Pi_{\mathrm{static}} = \{\pi : \ u_t = \varnothing, \ \forall t > 0\}$. In our $T = 2$ construction, this means the policy can take *at most one* non-trivial retrieval action at $t = 0$ and cannot adapt at $t = 1$.

Lemma D.3 ($V^*(\Pi_{\mathrm{static}}) \leq 1/2$ on $\mathcal{I}_{\mathrm{static}}$)**.** *On $\mathcal{I}_{\mathrm{static}}$, $V^*(\Pi_{\mathrm{static}}) \leq 1/2$.*

Proof. Let $\pi \in \Pi_{\mathrm{static}}$. There are two cases.

Case 1: π chooses $u_0 = u^{(1)}$ (or symmetrically $u^{(2)}$). Then $O_1 = \{\tau_1\}$ and $Z_1 = \{\tau_1\}$ regardless of x. The answer e_1 is correct iff $x = 1$, which has probability $1/2$. Thus $V^\pi \leq 1/2$.

Case 2: π chooses $u_0 = u^{(\mathrm{id})}$. Then $O_1 = \{\sigma_x\}$ and $Z_1 = \{\sigma_x\}$ reveals x. However, since π is static, it cannot take $u_1 \in \{u^{(1)}, u^{(2)}\}$ at $t = 1$ to retrieve τ_x. At termination, the evidence does not contain τ_x. Under our realizability convention, without the supporting triple, any decoder cannot exceed $1/2$ accuracy on the balanced $x \in \{1, 2\}$ family. Hence $V^\pi \leq 1/2$.

In all cases, $V^\pi \leq 1/2$. Taking supremum over $\pi \in \Pi_{\mathrm{static}}$ yields $V^*(\Pi_{\mathrm{static}}) \leq 1/2$. □

D.3.4 Π_{res} achieves V^* on the same family

Lemma D.4 ($V^*(\Pi_{\mathrm{res}}) = V^* = 1$ on $\mathcal{I}_{\mathrm{static}}$)**.** *On $\mathcal{I}_{\mathrm{static}}$, $V^*(\Pi_{\mathrm{res}}) = 1$.*

Proof. The optimal policy $\pi^\star$ in Lemma D.2 is implementable by Π_{res}: after observing σ_x, the residual set R_t contains the claim "the final answer must be supported by (e_Q, r, e_x)", which is unsupported under $Z_1 = \{\sigma_x\}$. Thus $\pi^{\mathrm{ret}}(\cdot \mid q, R_t, Z_t)$ can select the corresponding retrieval action $u^{(x)}$. Claim-conditioned projection is trivial here since $K = 1$ and the final observation is exactly τ_x. Therefore a policy in Π_{res} achieves value 1. □

Combining Lemmas D.2–D.4 yields a gap $\Delta_{\mathrm{static}} = 1/2$.

D.4 Separation II: $\Pi_{\mathrm{local}}(h)$ fails on long-range dependencies

Fix any hop budget $h \geq 1$. We construct a family $\mathcal{I}_{\mathrm{local}}(h)$ in which the unique supporting triple lies at distance $h+2$, so any h-local expansion policy cannot reach it.

D.4.1 Instance family $\mathcal{I}_{\mathrm{local}}(h)$

Let entities be $v_0, v_1, \ldots, v_{h+2}$ and relation r. Define the KG as a chain

$$G := \{(v_i, r, v_{i+1})\}_{i=0}^{h+1}. \tag{118}$$

The question q asks for the terminal node, with correct answer

$$a^\star := v_{h+2}. \tag{119}$$

Let the unique supporting triple be the last edge

$$\tau^\star := (v_{h+1}, r, v_{h+2}). \tag{120}$$

Set initial evidence $Z_0 = \{(v_0, r, v_1)\}$ and K large enough to not be the limiting factor here (or keep $K = 1$; the argument uses reachability, not capacity).

The h-hop neighborhood operator $\mathrm{Nbr}_h(Z)$: it returns all triples whose endpoints lie within h graph hops of the entity set incident to triples in Z.

D.4.2 $\Pi_{\mathrm{local}}(h)$ cannot reach $\tau^\star$

Lemma D.5 ($\tau^\star$ is unreachable under h-local policies)**.** *For any policy $\pi \in \Pi_{\mathrm{local}}(h)$, for all t, $\tau^\star \notin Z_t$ on $\mathcal{I}_{\mathrm{local}}(h)$.*

Proof. We prove by induction on t that all entities incident to Z_t lie in $\{v_0, \ldots, v_{h+1}\}$, which implies $\tau^\star$ (incident to v_{h+2}) cannot be included.

Base $t = 0$: Z_0 is incident only to $\{v_0, v_1\} \subseteq \{v_0, \ldots, v_{h+1}\}$.

Inductive step: assume Z_t is incident only to $\{v_0, \ldots, v_{h+1}\}$. Then by the definition of $\mathrm{Nbr}_h(Z_t)$, any triple in $\mathrm{Nbr}_h(Z_t)$ must have both endpoints within h hops of $\{v_0, \ldots, v_{h+1}\}$, which remains a subset of $\{v_0, \ldots, v_{h+1}\}$ in this chain (since v_{h+2} is at distance at least 1 beyond v_{h+1}, and from v_0 it is at distance $h+2 > h$). Therefore $\mathrm{Nbr}_h(Z_t)$ contains no triple incident to v_{h+2}, hence cannot contain $\tau^\star$. Since $\pi \in \Pi_{\mathrm{local}}(h)$ enforces $Z_{t+1} \subseteq \mathcal{P}_K(Z_t \cup \mathrm{Nbr}_h(Z_t))$, we conclude Z_{t+1} is also incident only to $\{v_0, \ldots, v_{h+1}\}$.

Thus for all t, $\tau^\star \notin Z_t$. □

D.4.3 Value gap

Lemma D.6 ($V^*(\Pi_{\mathrm{local}}(h)) \leq 1/2$ while $V^* = 1$ on $\mathcal{I}_{\mathrm{local}}(h)$)**.** *On $\mathcal{I}_{\mathrm{local}}(h)$, $V^* = 1$ and $V^*(\Pi_{\mathrm{local}}(h)) \leq 1/2$.*

Proof. First, $V^* = 1$ holds because an unrestricted retriever can directly query and retrieve $\tau^\star$ (e.g., via an action that targets $(v_{h+1}, r, \cdot)$), place it in Z_T, and decode v_{h+2} deterministically.

Second, by Lemma D.5, any $\pi \in \Pi_{\mathrm{local}}(h)$ never has $\tau^\star$ in evidence. We convert the task to a balanced two-answer variant by adding a distractor node $\bar{v}$ and defining $a^\star \in \{v_{h+2}, \bar{v}\}$ with equal prior, where $\tau^\star$ is the unique disambiguating support. Without $\tau^\star$, any policy cannot exceed $1/2$ accuracy on this balanced family. Thus $V^*(\Pi_{\mathrm{local}}(h)) \leq 1/2$. □

Lemma D.7 ($V^*(\Pi_{\mathrm{res}}) = V^* = 1$ on $\mathcal{I}_{\mathrm{local}}(h)$)**.** *On $\mathcal{I}_{\mathrm{local}}(h)$, Π_{res} can achieve $V^* = 1$.*

Proof. The residual claims include the missing support for the terminal node (e.g., "there exists an edge from v_{h+1} to the answer"). A residual-coupled retriever can issue a non-local query for $\tau^\star$ (allowed since Π_{res} has no h-local restriction), store it in Z_T, and decode v_{h+2}. □

Thus $\Delta_{\mathrm{local}} = 1/2$.

D.5 Separation III: Π_{hard} blocks hypothesis-driven retrieval

We now show a separation for reasoning constrained by hard verifiability, denoted by Π_{hard}. The construction formalizes a standard phenomenon in agentic KGQA: to retrieve the decisive fact, the agent must first *instantiate* a candidate intermediate entity (a hypothesis) and then query the KG with that entity as a key. Hard verifiability prohibits emitting unsupported intermediate claims, which can prevent the system from initiating the necessary retrieval.

D.5.1 Instance family $\mathcal{I}_{\mathrm{hard}}$

Let latent state $x \in \{1, 2\}$ with $\mathbb{P}(x = 1) = \mathbb{P}(x = 2) = 1/2$. Let candidate answers be c_1, c_2 and intermediate entities m_1, m_2. Define the KG

$$G := \{\ell_1, \ell_2, \tau_1, \tau_2\}, \qquad \ell_i = (e_Q, r_{\mathrm{clue}}, m_i), \qquad \tau_i = (m_i, r_{\mathrm{ans}}, c_i), \quad i \in \{1, 2\}. \tag{121}$$

The correct answer is $a^\star = c_x$.

Action set:

$$\mathcal{U} = \{u^{(\mathrm{clue})}\} \cup \{u(m) : m \in \{m_1, m_2\}\}. \tag{122}$$

Observations are deterministic:

$$O_{t+1} = \begin{cases} \{\ell_x\}, & u_t = u^{(\mathrm{clue})}, \\ \{\tau_1\}, & u_t = u(m_1), \\ \{\tau_2\}, & u_t = u(m_2). \end{cases} \tag{123}$$

Set $T = 2$, $Z_0 = \emptyset$, and $K = 1$.

D.5.2 A Minimal, Explicit Verifiability Condition on Action Claims

We use the fact that y_t may be a *structured decision* (e.g., a tool-call token or program token). Accordingly, we define a claim-extractor that includes the selected action as an atomic claim.

Definition D.8 (Action-claim extractor)**.** Let $C(\cdot)$ be any deterministic extractor such that for any step t, if the policy selects a non-trivial action $u_t = u(m)$, then the extracted claim set contains

$$\mathrm{Do}(m) \in C(y_t), \tag{124}$$

where $\mathrm{Do}(m)$ denotes the atomic claim "the agent will query using intermediate entity m".

We further assume the support score $\mathrm{Sup}(\cdot; Z)$ satisfies the monotonicity condition stated in the main text, and that purely action-intent claims are not directly supported by factual triples unless explicitly encoded. Formally:

Assumption D.9 (No factual support for pure action-intent claims)**.** For any $m \in \{m_1, m_2\}$ and any evidence set $Z \subseteq G$,

$$\mathrm{Sup}(\mathrm{Do}(m); Z) < 1. \tag{125}$$

This assumption is aligned with the definition that $\mathrm{Sup}(c; Z) = 1$ when c is directly supported by a triple in Z: no triple in G is of the form "should do $u(m)$", so $\mathrm{Do}(m)$ cannot be directly supported.

D.5.3 Optimal value $V^* = 1$

Lemma D.10 ($V^* = 1$ on $\mathcal{I}_{\mathrm{hard}}$)**.** *On $\mathcal{I}_{\mathrm{hard}}$, $V^* = 1$.*

Proof. Consider the policy: at $t = 0$, take $u_0 = u^{(\mathrm{clue})}$ and observe $O_1 = \{\ell_x\}$, which reveals m_x. At $t = 1$, take $u_1 = u(m_x)$ and observe $O_2 = \{\tau_x\}$. Then $Z_2 = \{\tau_x\}$ (since $K = 1$), which uniquely supports answer c_x. Thus $\hat{a} = c_x$ can be decoded with probability 1, yielding $V^\pi = 1$ and hence $V^* = 1$. □

D.5.4 Π_{hard} is bounded by $1/2$

Lemma D.11 ($V^*(\Pi_{\mathrm{hard}}) \le 1/2$ on $\mathcal{I}_{\mathrm{hard}}$)**.** *Assume Definition D.8 and Assumption D.9. Then on* $\mathcal{I}_{\mathrm{hard}}$, $V^*(\Pi_{\mathrm{hard}}) \le 1/2$.

Proof. Let $\pi \in \Pi_{\mathrm{hard}}$. For every step t and every $c \in C(y_t)$, we must have $\mathrm{Sup}(c; Z_t) = 1$.

Suppose π ever selects $u_t = u(m)$ for some m. Then by Definition D.8, $\mathrm{Do}(m) \in C(y_t)$. By Assumption D.9, $\mathrm{Sup}(\mathrm{Do}(m); Z_t) < 1$, contradicting the hard verifiability constraint. Therefore, under Π_{hard}, the policy cannot select any action of the form $u(m_1)$ or $u(m_2)$ at any time.

Hence the only permissible non-trivial action is $u^{(\mathrm{clue})}$, which returns ℓ_x but never retrieves τ_x. Without τ_x, the final evidence cannot contain the unique supporting triple for c_x. On the balanced $x \in \{1, 2\}$ family, any decoder or policy without τ_x cannot exceed $1/2$ accuracy. Thus $V^\pi \le 1/2$ for all $\pi \in \Pi_{\mathrm{hard}}$, so $V^*(\Pi_{\mathrm{hard}}) \le 1/2$. □

D.5.5 Π_{res} achieves V^*

Lemma D.12 ($V^*(\Pi_{\mathrm{res}}) = V^* = 1$ on $\mathcal{I}_{\mathrm{hard}}$)**.** *On* $\mathcal{I}_{\mathrm{hard}}$, Π_{res} *can achieve* $V^* = 1$.

Proof. After observing ℓ_x, the residual claims include the unsupported need for a triple of the form $(m_x, r_{\mathrm{ans}}, \cdot)$ to determine the answer. A residual-coupled retriever $\pi^{\mathrm{ret}}(\cdot \mid q, R_t, Z_t)$ can directly choose $u(m_x)$ (without requiring that the action-intent claim be fully supported as in Π_{hard}), obtain τ_x, and decode c_x with probability 1. □

Thus $\Delta_{\mathrm{hard}} = 1/2$.

D.6 Separation IV: Π_{add} fails under budget eviction without claim-aware projection

This separation targets the defining restriction of Π_{add}: evidence update is *claim-agnostic*, i.e., $Z_{t+1} = \mathcal{P}_K(Z_t \cup O_{t+1})$ does not use $C(y_{1:t})$. We construct tasks where two facts are jointly necessary, but $K = 1$ forces eviction; without claim-aware selection, a constrained class loses a constant amount of value.

D.6.1 Instance family $\mathcal{I}_{\mathrm{add}}$

Let latent state be $(a, b) \in \{0, 1\}^2$ uniformly distributed. Let the answer space be $\mathcal{A} = \{0, 1\}$ and define

$$a^\star := a \oplus b. \tag{126}$$

Encode the two bits as KG facts:

$$\tau_A(a) = (e_Q, r_A, \texttt{bitA}_a), \qquad \tau_B(b) = (e_Q, r_B, \texttt{bitB}_b). \tag{127}$$

The KG is $G = \{\tau_A(a), \tau_B(b)\}$. Let there be a single retrieval action $u^{(\mathrm{both})}$ returning both triples:

$$O_1 = \{\tau_A(a), \tau_B(b)\} \quad \text{when} \quad u_0 = u^{(\mathrm{both})}. \tag{128}$$

Set $K = 1$ and choose $\mathcal{P}_1$ in (112) with an order that always prefers τ_A over τ_B:

$$\tau_A(a) \prec \tau_B(b) \ \text{ for all } a, b. \tag{129}$$

Thus $\mathcal{P}_1(\{\tau_A(a), \tau_B(b)\}) = \{\tau_A(a)\}$ always.

D.6.2 Optimal value $V^* = 1$

Lemma D.13 ($V^* = 1$ on $\mathcal{I}_{\mathrm{add}}$)**.** *On* $\mathcal{I}_{\mathrm{add}}$, $V^* = 1$.

Proof. We allow general policies (outside Π_{add}) to use claim-conditioned projection and to interact for multiple steps. A near-minimal strategy is: (i) retrieve both, then (ii) retain whichever triple is required for the next residual computation while storing the other in the trajectory memory $y_{1:t}$, or equivalently (iii) retrieve twice so that both bits are observed at least once before decoding. Since both $\tau_A(a)$ and $\tau_B(b)$ are observable through $u^{(\mathrm{both})}$, the policy can recover (a, b) with probability 1 across steps and output $a \oplus b$ deterministically. Thus $V^* = 1$. □

D.6.3 Π_{add} is bounded by $1/2$

Lemma D.14 ($V^*(\Pi_{\mathrm{add}}) \le 1/2$ on $\mathcal{I}_{\mathrm{add}}$)**.** *On $\mathcal{I}_{\mathrm{add}}$, $V^*(\Pi_{\mathrm{add}}) \le 1/2$.*

Proof. Let $\pi \in \Pi_{\mathrm{add}}$. Evidence updates satisfy $Z_{t+1} = \mathcal{P}_1(Z_t \cup O_{t+1})$ and are claim-agnostic. At $t = 0$, taking $u^{(\mathrm{both})}$ yields $O_1 = \{\tau_A(a), \tau_B(b)\}$. By (129) and (112), we have $Z_1 = \{\tau_A(a)\}$ deterministically, and $\tau_B(b)$ is evicted.

Since all future projections remain claim-agnostic with the same preference, any observation set containing both triples will again project to $\tau_A(a)$, hence no Z_t can simultaneously contain information about b (as a KG fact) at the decision time. Therefore the policy cannot determine $a \oplus b$ better than guessing given only a. Formally, conditioning on $\tau_A(a)$, the bit b remains uniform in $\{0, 1\}$, so

$$\mathbb{P}(a \oplus b = 0 \mid a) = \mathbb{P}(b = a \mid a) = 1/2, \quad \mathbb{P}(a \oplus b = 1 \mid a) = 1/2.$$

Thus any predictor based on Z_t alone has accuracy at most $1/2$. Hence $V^\pi \le 1/2$ for all $\pi \in \Pi_{\mathrm{add}}$, implying $V^*(\Pi_{\mathrm{add}}) \le 1/2$. □

D.6.4 Π_{res} achieves V^*

Lemma D.15 ($V^*(\Pi_{\mathrm{res}}) = V^* = 1$ on $\mathcal{I}_{\mathrm{add}}$)**.** *On $\mathcal{I}_{\mathrm{add}}$, Π_{res} can achieve $V^* = 1$.*

Proof. A policy in Π_{res} may use claim-conditioned projection $\mathcal{P}_1(\cdot;\, C(y_{1:t}))$ to prioritize the residual needs. At $t = 0$, retrieve both bits. Set y_1 to store the extracted value of a as a structured intermediate artifact, and let the residual at $t = 1$ require support for the missing bit b. Then the claim-conditioned projection at $t = 1$ can retain $\tau_B(b)$ (instead of always retaining $\tau_A(a)$), making b available in evidence while a is already stored in the trajectory. The decoder can then compute $a \oplus b$ deterministically. Therefore Π_{res} achieves value 1. □

Thus $\Delta_{\mathrm{add}} = 1/2$.

D.7 Conclusion: proof of Theorem 6

Proof of Theorem D.1. For each constrained class:

(i) Static. By Lemmas D.2 and D.3, on $\mathcal{I}_{\mathrm{static}}$ we have $V^* = 1$ and $V^*(\Pi_{\mathrm{static}}) \le 1/2$. By Lemma D.4, $V^*(\Pi_{\mathrm{res}}) = 1$ on the same family.

(ii) Local. By Lemmas D.6 and D.7, on $\mathcal{I}_{\mathrm{local}}(h)$ we have $V^* = 1$, $V^*(\Pi_{\mathrm{local}}(h)) \le 1/2$, and $V^*(\Pi_{\mathrm{res}}) = 1$.

(iii) Hard-faithful. By Lemmas D.10 and D.11, on $\mathcal{I}_{\mathrm{hard}}$ we have $V^* = 1$ and $V^*(\Pi_{\mathrm{hard}}) \le 1/2$ (under Definition D.8 and Assumption D.9). By Lemma D.12, $V^*(\Pi_{\mathrm{res}}) = 1$.

(iv) Coarse. By Lemmas D.13 and D.14, on $\mathcal{I}_{\mathrm{add}}$ we have $V^* = 1$ and $V^*(\Pi_{\mathrm{add}}) \le 1/2$. By Lemma D.15, $V^*(\Pi_{\mathrm{res}}) = 1$.

In each case, choose $\Delta_\Pi := 1/2 > 0$. This proves (109) and establishes $V^*(\Pi_{\mathrm{res}}) = V^*$ on the same instance family. □

E Necessary Coupling under Realizability

The proof is constructive: we define an explicit family of KGQA instances in which (i) the optimal value is $V^* = 1$, (ii) any realizable policy whose retrieval decision ignores either the trajectory prefix $y_{1:t}$ (or equivalently any uncertainty surrogate derived from it) or the retained evidence Z_t is bounded by $1/2$, and (iii) a residual-coupled realizable policy achieves 1. This yields a constant gap and implies that $(y_{1:t}, Z_t)$ must co-evolve in closed loop.

E.1 Restatement

Theorem E.1 (Necessary coupling under realizability (restated))**.** *Assume the realizability abstraction, i.e., there exists a mapping $\mathcal{B}$ such that the surrogate $\tilde{b}_t = \mathcal{B}(y_{1:t})$ is sufficient for near-optimal*

evidence-acquisition decisions. Then there exists an optimal (or near-optimal) realizable policy of the form

$$u_t \sim \pi^{\mathrm{ret}}(\cdot \mid q, y_{1:t}, Z_t), \tag{130}$$

$$y_{t+1} \sim \pi^{\mathrm{gen}}(\cdot \mid q, y_{1:t}, Z_{t+1}), \tag{131}$$

and moreover there exist instances where any realizable policy whose retrieval rule ignores $y_{1:t}$ or ignores Z_t is strictly suboptimal. Hence $(y_{1:t}, Z_t)$ must co-evolve in closed loop.

Comment on scope. The first statement is an *existence* claim: there exists a realizable near-optimal policy. The second statement is a *necessity* claim: for some instances, any restricted realizable controller is strictly suboptimal. Together, these two statements formalize the sense in which reasoning trajectories and evidence subgraphs must co-evolve.

E.2 A minimal realizable-controller template

We reuse the minimal KGQA-POMDP template of Appendix D.2: terminal reward is 0–1 accuracy, horizon is finite, and the evidence budget is $K = 1$ with deterministic, non-trivial projection $\mathcal{P}_1$ defined in (112). We now additionally formalize realizable controllers and the notion of "ignoring" $y_{1:t}$ or Z_t.

Definition E.2 (Realizable retrieval policy)**.** A retrieval policy is *realizable* if for each t its action distribution is measurable with respect to $\sigma(q, y_{1:t}, Z_t)$, i.e., there exists a stochastic kernel $\pi^{\mathrm{ret}}(\cdot \mid q, y_{1:t}, Z_t)$ such that $u_t \sim \pi^{\mathrm{ret}}(\cdot \mid q, y_{1:t}, Z_t)$.

Definition E.3 (Two restricted realizable classes)**.** Define two restricted realizable classes:

$$\Pi_{y\text{-only}} := \Big\{\pi \text{ realizable} : \ u_t \sim \pi^{\mathrm{ret}}(\cdot \mid q, y_{1:t}) \text{ (no dependence on } Z_t)\Big\}, \tag{132}$$

$$\Pi_{Z\text{-only}} := \Big\{\pi \text{ realizable} : \ u_t \sim \pi^{\mathrm{ret}}(\cdot \mid q, Z_t) \text{ (no dependence on } y_{1:t})\Big\}. \tag{133}$$

These two classes correspond to the two "missing coupling" failure modes: without $y_{1:t}$ the controller cannot access uncertainty/hypotheses; without Z_t the controller cannot condition on which evidence is currently retained under the budget.

E.3 Instance family $\mathcal{I}_{\mathrm{coup}}$

We construct instances where the correct answer depends on *both*: (1) a latent bit revealed only through an intermediate observation that must be recorded in the trajectory $y_{1:t}$, and (2) a second fact that must be kept in the evidence state Z_t at termination under $K = 1$.

Entities and triples. Fix entities e_Q (query entity), two candidate answers c_0, c_1, two intermediate keys m_0, m_1, and two fact entities $\texttt{key}_0, \texttt{key}_1$. Let the KG be

$$G := \{\ell_0, \ell_1, \tau_0, \tau_1\}, \qquad \ell_i = (e_Q, r_{\mathrm{clue}}, m_i), \qquad \tau_i = (m_i, r_{\mathrm{ans}}, c_i), \quad i \in \{0, 1\}. \tag{134}$$

Latent state and answer. Let latent state be $x \in \{0, 1\}$ with $\mathbb{P}(x = 0) = \mathbb{P}(x = 1) = 1/2$. The correct answer is $a^\star = c_x$.

Actions and observations. Let the action set be

$$\mathcal{U} = \{u^{(\mathrm{clue})}, u(m_0), u(m_1)\}. \tag{135}$$

Observations are deterministic given (x, u):

$$O_{t+1} = \begin{cases} \{\ell_x\}, & u_t = u^{(\mathrm{clue})}, \\ \{\tau_0\}, & u_t = u(m_0), \\ \{\tau_1\}, & u_t = u(m_1). \end{cases} \tag{136}$$

Set horizon $T = 2$, initial evidence $Z_0 = \emptyset$, budget $K = 1$, and projection $\mathcal{P}_1$.

Trajectory semantics (realizability convention). We use the same convention as Appendix D.2: if the unique supporting triple τ_x is present in the terminal evidence Z_2, then the decoder can output c_x with probability 1; otherwise no policy can exceed $1/2$ accuracy on the balanced $x \in \{0, 1\}$ family.

Where $y_{1:t}$ matters. We assume the controller can store the observed clue ℓ_x (or equivalently the identity of m_x) as part of y_1 after step $t = 0$ (this is exactly the role of y_t as an intermediate reasoning artifact in Section 3). Thus, on this family, the only way for a realizable controller to condition future actions on x is via $y_{1:1}$.

E.4 Optimal value $V^* = 1$

Lemma E.4 ($V^* = 1$ on $\mathcal{I}_{\text{coup}}$)**.** *On $\mathcal{I}_{\text{coup}}$, $V^* = 1$.*

Proof. Consider the policy $\pi^\star$: at $t = 0$, take $u_0 = u^{(\text{clue})}$ and observe $O_1 = \{\ell_x\}$ by (136). At $t = 1$, if $O_1 = \{\ell_0\}$ take $u_1 = u(m_0)$; if $O_1 = \{\ell_1\}$ take $u_1 = u(m_1)$. Then $O_2 = \{\tau_x\}$ by (136), hence $Z_2 = \{\tau_x\}$. By construction, τ_x uniquely supports answer c_x, so the decoder outputs c_x with probability 1. Therefore $V^{\pi^\star} = 1$ and hence $V^* = 1$. □

E.5 Necessity: ignoring $y_{1:t}$ or Z_t incurs a constant gap

Lemma E.5 ($V^*(\Pi_{Z\text{-only}}) \le 1/2$ on $\mathcal{I}_{\text{coup}}$)**.** *On $\mathcal{I}_{\text{coup}}$, any realizable policy whose retrieval ignores $y_{1:t}$ satisfies $V^\pi \le 1/2$. Hence $V^*(\Pi_{Z\text{-only}}) \le 1/2$.*

Proof. Let $\pi \in \Pi_{Z\text{-only}}$. By definition, u_1 may depend on Z_1 but not on $y_{1:1}$. At $t = 0$, regardless of which action π takes, Z_1 can take at most one of the following forms: $\emptyset$ (if no-op), $\{\ell_x\}$ (if $u^{(\text{clue})}$), or $\{\tau_i\}$ (if $u(m_i)$). Crucially, under $K = 1$, Z_1 does not deterministically encode x in a way that allows choosing $u(m_x)$ at $t = 1$ unless the policy can condition on the identity of ℓ_x as stored in the trajectory. Since π cannot use $y_{1:1}$, it cannot condition its choice of u_1 on whether $x = 0$ or $x = 1$. Therefore, conditioned on any Z_1, the action u_1 is the same for both x values with positive probability. By symmetry of the balanced prior over x, at least one of the two states receives the wrong second-step query with probability at least $1/2$. When $u_1 \neq u(m_x)$, the terminal evidence cannot contain τ_x, so by the realizability convention the success probability is at most $1/2$. Hence overall $V^\pi \le 1/2$ and taking supremum yields $V^*(\Pi_{Z\text{-only}}) \le 1/2$. □

Lemma E.6 ($V^*(\Pi_{y\text{-only}}) \le 1/2$ on $\mathcal{I}_{\text{coup}}$)**.** *On $\mathcal{I}_{\text{coup}}$, any realizable policy whose retrieval ignores Z_t satisfies $V^\pi \le 1/2$. Hence $V^*(\Pi_{y\text{-only}}) \le 1/2$.*

Proof. Let $\pi \in \Pi_{y\text{-only}}$. Then u_1 may depend on $y_{1:1}$ (hence on the clue ℓ_x) but not on Z_1. We modify the family $\mathcal{I}_{\text{coup}}$ by making the projection order instance-specific: fix an order $\prec$ such that for one half of instances (call them type-A), $\tau_0 \prec \tau_1$, and for the other half (type-B), $\tau_1 \prec \tau_0$. The policy does not observe the order $\prec$ through $y_{1:1}$, but $\prec$ determines which supporting triple survives under $K = 1$ whenever both are present in a union before projection. In particular, there exist histories where Z_1 equals the surviving triple under $\mathcal{P}_1$, and choosing the correct final retrieval action requires conditioning on which triple is currently retained (i.e., on Z_1). Since π ignores Z_1, it cannot adapt to the retained-evidence identity and thus cannot ensure $\tau_x \in Z_2$ for both order types. By balancing over the two order types, any such policy fails with probability at least $1/2$, yielding $V^\pi \le 1/2$. □

Remark. Lemma E.6 formalizes the "budget eviction" effect: when $\mathcal{P}_K$ is non-trivial, which triple is retained matters, and a controller that does not condition on Z_t cannot guarantee that the needed supporting triple remains in evidence.

E.6 Achievability by a residual-coupled realizable policy

Lemma E.7 ($V^*(\Pi_{\text{res}}) = V^* = 1$ on $\mathcal{I}_{\text{coup}}$)**.** *On $\mathcal{I}_{\text{coup}}$, Π_{res} contains a realizable policy achieving value 1.*

Proof. Consider the following policy in Π_{res}: at $t = 0$ retrieve the clue using $u^{(\text{clue})}$, observe ℓ_x, and write into the trajectory a structured artifact indicating the residual need: "retrieve the answer

edge out of m_x". This residual claim is unsupported under $Z_1 = \{\ell_x\}$, so $\pi^{\mathrm{ret}}(\cdot \mid q, R_1, Z_1)$ selects $u(m_x)$ at $t = 1$. Then $O_2 = \{\tau_x\}$ and $Z_2 = \{\tau_x\}$, enabling deterministic decoding of c_x. Thus $V^\pi = 1$. □

E.7 Conclusion: proof of Theorem E.1

Proof of Theorem E.1. The existence of a realizable optimal (or near-optimal) policy of the form (130)–(131) follows by construction in Lemma E.7 on the explicit family $\mathcal{I}_{\mathrm{coup}}$, and more generally by substituting the realizable surrogate $\tilde{b}_t = \mathcal{B}(y_{1:t})$ for the latent belief in the information-state optimality result of Theorem 4.1.

Moreover, Lemmas E.5 and E.6 establish strict suboptimality for realizable policies that ignore $y_{1:t}$ or ignore Z_t on $\mathcal{I}_{\mathrm{coup}}$: in both cases $V^*(\Pi) \le 1/2$ while $V^* = 1$ (Lemma E.4). Thus, on these instances, any realizable near-optimal policy must condition retrieval jointly on $(y_{1:t}, Z_t)$.

Finally, since u_t affects the observation O_{t+1} and thus the evidence update $Z_{t+1} = \mathcal{P}_1(Z_t \cup O_{t+1})$, and generation at the next step conditions on Z_{t+1}, the pair $(y_{1:t}, Z_t)$ necessarily forms a closed-loop co-evolving state for realizable near-optimal control. □

F Proof of Sufficiency for Residual Mass Contraction

F.1 Objective and basic assumptions

We work with the belief-MDP induced by the POMDP dynamics and focus on the convergence of evidential sufficiency as measured by the residual mass. Recall the residual mass

$$\Phi_t \;=\; \sum_{c \in R_t} \big(1 - \mathrm{Sup}(c; Z_t)\big), \tag{137}$$

where $\mathrm{Sup}(c; Z) \in [0, 1]$ is monotone in Z. We assume:

(H1) One-step contraction existence. For any R_t such that $\Phi_t > 0$, there exists an admissible action u_t satisfying

$$\mathbb{E}[\Phi_{t+1} \mid \mathcal{H}_t] \le (1 - \eta)\Phi_t \tag{138}$$

for some constant $\eta > 0$.

H1 follows for the selected retrieval action from the history-conditional retrieval gain, projection preservation, and residual-refresh condition in Theorem 4.2.

(H2) Gain-preserving claim-conditioned projection. Let $Z_{t+1} = \mathcal{P}_K(Z_t \cup O_{t+1}; C(y_{1:t}))$. Whenever a newly acquired triple $e \in O_{t+1}$ increases the support of some residual claim $c \in R_t$, i.e., $\mathrm{Sup}(c; Z_t \cup \{e\}) > \mathrm{Sup}(c; Z_t)$, the projection preserves this gain under the budget, ensuring that the post-projection evidence satisfies

$$\mathrm{Sup}(c; Z_{t+1}) \;\ge\; \mathrm{Sup}(c; Z_t \cup \{e\}). \tag{139}$$

This condition prevents budget projection from discarding newly acquired evidence that improves residual support.

F.2 Constructing a residual-driven policy

We construct an explicit residual-coupled policy π^{shr}. At each step t, given $(q, y_{1:t}, Z_t)$, compute $R_t \subseteq C(y_t)$ and Φ_t via (137). If $\Phi_t = 0$, terminate the theoretical process and keep this state absorbing. If $\Phi_t > 0$, select an admissible action u_t that satisfies (138) (which exists by assumption). The evidence is then updated by the claim-conditioned projection $Z_{t+1} = \mathcal{P}_K(Z_t \cup O_{t+1}; C(y_{1:t}))$. Hence π^{shr} is residual-coupled by construction.

F.3 Stepwise contraction of the residual mass

Let $\mathcal{H}_t$ denote the complete interaction history up to time t, including $(q, u_{1:t}, O_{1:t}, y_{1:t}, Z_t)$.

Table 7: Residual-mass contraction across CoRE rounds. Residual mass is approximated by unsupported answer-critical claim counts on fixed evaluation packs.

Dataset	Backbone	Round 1	Round 2	Round 3	Mean contraction	Monotone rate
WebQSP	Meta-Llama-3.1-8B-Instruct	0.650	0.490	0.540	0.928	81.00%
CWQ	Meta-Llama-3.1-8B-Instruct	1.160	0.860	0.760	0.813	69.00%

Lemma F.1 (Conditional one-step contraction)**.** *Under π^{shr}, for any t with $\Phi_t > 0$,*

$$\mathbb{E}[\Phi_{t+1} \mid \mathcal{H}_t] \le (1-\eta)\Phi_t. \tag{140}$$

Proof. If $\Phi_t > 0$, π^{shr} selects an admissible action satisfying the history-conditional inequality (138). Applying this inequality directly yields (140). □

Lemma F.2 (Geometric decay in expectation)**.** *Under π^{shr}, for all $t \ge 0$,*

$$\mathbb{E}[\Phi_t] \le (1-\eta)^t \, \Phi_0. \tag{141}$$

Proof. We proceed by induction. The statement holds at $t = 0$ trivially. Assume $\mathbb{E}[\Phi_t] \le (1-\eta)^t \Phi_0$. Then by the tower property and Lemma F.1,

$$\begin{aligned} \mathbb{E}[\Phi_{t+1}] &= \mathbb{E}[\mathbb{E}[\Phi_{t+1} \mid \mathcal{H}_t]] \\ &\le \mathbb{E}[(1-\eta)\Phi_t] = (1-\eta)\mathbb{E}[\Phi_t] \\ &\le (1-\eta)\cdot(1-\eta)^t\Phi_0 = (1-\eta)^{t+1}\Phi_0. \end{aligned} \tag{142}$$

This completes the induction. □

F.4 Hitting-time bound for small residual mass

For any threshold $\delta > 0$, define the stopping time

$$T_\delta \;=\; \inf\{t \ge 0 :\; \Phi_t \le \delta\}. \tag{143}$$

Lemma F.3 (Tail bound via Markov inequality)**.** *For any $t \ge 0$,*

$$\mathbb{P}(T_\delta > t) = \mathbb{P}(\Phi_t > \delta) \le \frac{\mathbb{E}[\Phi_t]}{\delta} \le \frac{(1-\eta)^t\Phi_0}{\delta}. \tag{144}$$

Proof. By definition, $T_\delta > t$ iff $\Phi_t > \delta$, hence the first equality. Since $\Phi_t \ge 0$, Markov's inequality gives $\mathbb{P}(\Phi_t > \delta) \le \mathbb{E}[\Phi_t]/\delta$. The final inequality uses Lemma F.2. □

F.5 Proof of Theorem 4.2

Proof of Theorem 4.2. Lemma F.2 directly gives the geometric contraction bound $\mathbb{E}[\Phi_t] \le (1-\eta)^t\Phi_0$. To obtain an iteration complexity bound, fix any $\varepsilon > 0$ and choose

$$t_\varepsilon \;=\; \left\lceil \frac{\log(\Phi_0/\varepsilon)}{\log(1/(1-\eta))} \right\rceil, \tag{145}$$

so that $(1-\eta)^{t_\varepsilon}\Phi_0 \le \varepsilon$. Then $\mathbb{E}[\Phi_{t_\varepsilon}] \le \varepsilon$. Equivalently, using $\log(1/(1-\eta)) \ge \eta$ for $\eta \in (0,1)$, we have the standard bound $t_\varepsilon = O(\log(\Phi_0/\varepsilon)/\eta)$. Finally, Lemma F.3 provides a high-probability tail bound on the hitting time T_δ, showing that $\mathbb{P}(T_\delta > t) \le (1-\eta)^t\Phi_0/\delta$ for any $\delta > 0$, which implies that the probability of failing to reach a δ-small residual mass decays geometrically with the number of iterations. □

G Additional Diagnostics

G.1 Additional matched and robustness diagnostics

The following diagnostics use $K = 60$ and $T = 3$ unless noted. They distinguish retrieval recall, projection-set fidelity, answer correctness, and output coverage.

Table 8: Static long-context and stronger one-shot diagnostic checks. These subset results complement the main benchmark table.

Dataset	Method	Backbone	Setting	n	Hit	F1	FCR
WebQSP	Static	GPT-4o-mini	ctx=100	50	86.0	75.32	74.0
WebQSP	Static	GPT-4o-mini	ctx=200	50	92.0	80.85	82.0
WebQSP	Static	GPT-4o-mini	ctx=500	50	92.0	81.19	70.0
WebQSP	Static	GPT-4o-mini	ctx=1000	50	90.0	78.16	74.0
CWQ	Static	GPT-4o	ctx=100, one-shot	100	70.0	61.82	56.0
CWQ	CoRE	GPT-4o-mini	$K = 100$, up to 3 rounds	100	75.0	65.34	57.0

Initial-retrieval recall. We measure mean answer-entity recall and mean annotated shortest-path triple recall. The evaluated files contain 1,639 WebQSP and 3,531 CWQ questions; the corresponding nonempty-annotation denominators are 1,559/1,554 and 2,848/2,831 for entity/path recall. Mean triple recall differs from the fraction of questions with every path triple retrieved.

Table 9: Initial-retrieval answer-entity / mean path-triple recall (%). Averages use the eligible annotation denominators stated in the text.

Dataset	$K = 10$	$K = 30$	$K = 60$	$K = 100$	$K = 500$
WebQSP	74.40/64.37	85.26/77.91	90.34/84.24	93.36/88.14	98.69/97.15
CWQ	65.23/43.98	80.25/61.82	87.36/72.15	91.32/78.66	98.36/94.10

Matched inference cost. The reference is the first round of the same run, with identical initial evidence, model, decoding, and checker. Table 10 sums the additional generation and structured-serialization calls, input/output tokens, and LLM-call latency per question. Latency is a sum of measured call durations, rather than end-to-end wall time. The reference has two calls per question under this accounting; Table 6 retains its original benchmark usage protocol.

Table 10: Additional online cost relative to the first round. Deleted conditions remove a critical bridge while leaving alternative evidence searchable.

Dataset / condition	n	Extra calls	Extra tokens	Extra LLM latency (s)
WebQSP clean	287	1.052	713	5.52
WebQSP deleted	287	1.206	848	6.64
CWQ clean	500	1.552	1084	9.01
CWQ deleted	500	1.720	1253	11.04

Matched evidence-update controls. Query-only uses the original question, local-frontier expands nearby graph edges, and progressive add-only appends evidence under the same budget. Candidate resources, retriever, backbone, decoding, checker, $K = 60$, and $T = 3$ are matched within each comparison. Intervals use 10,000 paired question-level bootstrap resamples. The independent WebQSP confirmation set contains 137 questions. These mechanism controls are separate from the cited main-table baselines.

Residual-feature corruption. The component audit holds out questions using 60/20/20 train/validation/test splits over 20 seeds. At 30% feature corruption, anchor/relation matches are deleted or spuriously added, with an extra false residual/open slot at the state level. Noise-trained MLPs augment training at 10/20/30% noise; both scorers use the same candidate sets and 60-triple budget. For this diagnostic, targets are the logged retained triples and weights are uniform. Recall therefore measures recovery of the logged projection set, rather than recall of annotated answer-critical facts. The 95% intervals resample paired seed-level differences.

Independent end-to-end and selective evaluation. The WebQSP confirmation set excludes the 50 development questions from a fixed 150-question set, leaving 100 questions disjoint from selector training and validation. Table 13 compares heuristic projection with a noise-trained ensemble. All answer and evidence metrics include all 100 questions. F1 changes by 0.30 points (95% CI $[-0.79, 1.46]$) and FCR by 4.00 points ($[-3.00, 11.00]$). The learned selector returns 65 `SUPPORTED` outputs: strict coverage is 65%, and strict Hit/F1 are 95.38/77.93 on that subset. An unresolved open

Table 11: CoRE minus matched evidence-update control, in percentage points with 95% paired bootstrap intervals.

Dataset	n	Control	ΔHit [95% CI]	ΔF1 [95% CI]
CWQ	500	Query-only	+2.40 [0.60, 4.20]	+2.14 [0.56, 3.80]
CWQ	500	Local-frontier	+2.40 [0.60, 4.20]	+2.14 [0.56, 3.80]
CWQ	500	Progressive add-only	+3.80 [0.40, 7.00]	+5.34 [2.58, 8.15]
WebQSP holdout	137	Query-only	+0.73 [0.00, 2.19]	+1.61 [0.07, 3.62]

Table 12: Logged-projection target recall (%) under 30% residual-feature corruption. This component audit measures selection fidelity.

Dataset	Seeds	Clean-trained MLP	Noise-trained MLP	Gain [95% CI]
WebQSP	20	72.69	83.69	+11.00 [10.56, 11.44]
CWQ	20	76.45	80.54	+4.09 [3.63, 4.54]

slot can coexist with supported generated claims, so FCR and supported coverage measure different events.

Table 13: Independent WebQSP diagnostic ($n = 100$). Hit, F1, FactSup, and FCR are percentages; S/U/V are counts of SUPPORTED/UNRESOLVED/UNVERIFIED outputs. Calls include generation and serialization.

Selector	Hit	F1	FactSup	FCR	S	U	V	Calls
Heuristic	92.00	70.75	89.66	74.00	61	37	2	2.98
Noise-trained	92.00	71.05	91.62	78.00	65	33	2	3.00

Observable counterparts of the sufficient conditions. Alignment records whether at least one retained triple matches an executed residual slot by anchor or relation hint. The alignment audit uses 287 WebQSP and 500 CWQ runs per condition; its eligible-transition denominators appear in Table 14. A separate residual-refresh audit uses 86/99 WebQSP and 226/253 CWQ logged transitions (intact/deleted), then examines the subset meeting the closure-selection margin of 4. Its mass is the citation-validated unsupported-claim count. These are observable proxies for retrieval and retention behavior; the conditioning on selected updates is not an execution-time contraction guarantee.

Claim-decomposition sensitivity. Atomic, minimal answer-critical, and edge-aligned prompts are compared on the same 50 questions per dataset with matched model, retrieval resources, budget, and controller. Agreement is the fraction with the same final answer under all three strategies; demand overlap is the mean pairwise Jaccard similarity of normalized entity/relation/type requests.

Missing and restored evidence. On 287 WebQSP bridge-deletion questions, the loop exceeds its first-round prediction by 1.39 Hit points (95% CI $[0.35, 2.79]$) and 1.54 F1 points ($[0.60, 2.62]$). Alternative evidence remains searchable in this test. GrailQAbility[5] provides a separate acceptance test: its held-out split has 290 answerable and 302 unanswerable questions after 206 calibration cases. The calibrated gate requires valid citations and question/answer-entity coverage. On 2,000 held-out FactKG[6] claims, complete, missing, and restored evidence are checked separately. These auxiliary checker tests use their own calibrated acceptance rules rather than the KGQA benchmark controller.

A separate KG support-checker audit selects its prompt on 120 pilot cases and evaluates 180 disjoint cases constructed from entailed claims and tail/predicate swaps. Accuracy is 95.00%, supported recall 96.67%, and unsupported specificity 94.17%; the expected-citation validity rate is 100%. This audit measures checker decisions on supplied claims, separately from the extraction-coverage audit in Table 4.

Projection transfer within Freebase. Over 20 seeds, we compare a source-trained noise-augmented MLP transferred between WebQSP and CWQ with a model trained on 10% target

[5]Dataset description: `https://arxiv.org/abs/2212.10189`.
[6]Dataset description: `https://arxiv.org/abs/2305.06590`.

Table 14: Alignment audit and separate residual-refresh audit. n_a counts eligible alignment transitions; n_g counts guard-selected refresh transitions; decrease is measured within the latter subset.

Condition	n_a	Alignment (%)	n_g	Mass decrease (%)
WebQSP clean	85	82.35	28	100.00
WebQSP deleted	97	87.63	40	100.00
CWQ clean	218	94.95	73	94.52
CWQ deleted	242	94.63	68	86.76

Table 15: Decomposition sensitivity on matched 50-question subsets. Each strategy reports Hit/F1 (%); agreement is a percentage.

Dataset	Atomic	Minimal	Edge-aligned	Agreement	Demand overlap
WebQSP	80.00/60.22	80.00/59.51	80.00/58.60	84.00	0.671
CWQ	86.00/74.64	90.00/77.14	90.00/76.14	84.00	0.766

questions; a separate 10% validates target training and the remaining 80% is held out for evaluation in both settings. Table 17 reports recovery of logged target projection sets with clean and 30%-corrupted features. Both datasets use Freebase, so the evaluated shift is between question distributions within one KG. Seed-level variation covers question splits, model fitting, and feature corruption; question bootstrap intervals do not measure repeated API-run variability.

H Experiment Setup

H.1 Datasets

We evaluate CoRE on two widely used Freebase-based KGQA benchmarks: WebQuestionsSP (WebQSP) [33] and Complex WebQuestions (CWQ) [34]. Both datasets are grounded on Freebase and are commonly used to benchmark multi-hop reasoning over large knowledge graphs.

WebQuestionsSP (WebQSP). WebQSP is a semantic-parsing enhanced version of the WebQuestions benchmark, designed for answering natural-language questions over Freebase with executable logical forms. It provides SPARQL-based semantic parses for a large portion of questions, together with gold answer sets obtained by executing these parses on Freebase. The official release states that WebQSP contains *full* semantic parses (SPARQL queries) for 4,737 questions and *partial* annotations for 1,073 questions where a valid parse cannot be formulated or the question requires descriptive answers, along with an evaluation script and STAGG outputs. [33]

Complex WebQuestions (CWQ). CWQ targets *compositional* and *multi-hop* question answering. It is introduced to study complex questions that can be decomposed into sequences of simpler subquestions; the dataset accompanies the original work on question decomposition and web/KG-based answering. [34] In addition, the dataset is released with multiple possible usages, including (i) interacting with a search engine via collected web snippets, and (ii) semantic parsing where each question can be paired with a SPARQL query executable against Freebase. [34]

Statistics and splits. For fair comparison with prior work, we follow the standard training/development/test splits used in the literature. Table 18 summarizes dataset statistics (train/dev/test sizes and the maximum hop complexity) as reported in prior work.

Table 16: Auxiliary support-gate diagnostics (%, 95% intervals). FactKG negative-case specificity concerns complete-evidence negative claims, not missing-evidence detection.

Test	Outcome / denominator	Result [95% CI]
GrailQAbility	Accepted-answer accuracy	96.73 [93.62, 99.33]
GrailQAbility	Unanswerable rejection ($n = 302$)	90.73 [87.42, 94.04]
FactKG	Restored positive-case acceptance	70.08 [67.25, 72.82]
FactKG	Complete negative-case specificity	95.38 [94.01, 96.64]
FactKG	Missing positive-case insufficiency	46.39 [43.41, 49.51]

Table 17: Within-Freebase transfer of the noise-trained MLP. Values are mean logged-projection target recall (%) over 20 seeds.

Transfer	Target training	Clean	30% corruption
WebQSP→CWQ	Zero-shot	78.66	76.87
WebQSP→CWQ	10% target training	87.93	79.02
CWQ→WebQSP	Zero-shot	85.61	73.02
CWQ→WebQSP	10% target training	87.77	78.78

Table 18: Dataset statistics (Freebase-based KGQA).

Dataset	Train	Dev	Test	Max hop
WebQSP	2,848	250	1,639	2
CWQ	27,639	3,519	3,531	4

H.2 Baselines

We compare CoRE with a broad set of representative baselines covering agent-based reasoning, retrieval-augmented generation (RAG), and KG-specialized retrieval methods.

Agent-based methods. StructGPT [12] introduces an iterative read–reason framework that alternates between structured data access and LLM reasoning, enabling multi-step interaction with external knowledge sources. ToG [6] formulates KGQA as an interactive graph exploration process guided by LLM reasoning with beam-style search over candidate paths. PoG [19] proposes a plan-on-graph paradigm that decomposes complex questions into guided subgoals, incorporating memory and reflection to refine graph exploration. Interactive-KBQA [20] treats KBQA as an interactive semantic parsing process, where LLMs iteratively construct and repair executable logical forms through KB operations. ReKnoS [2] enhances LLM-based KG reasoning by introducing super-relations that summarize multi-hop relational patterns, enabling both forward and backward reasoning under incomplete KGs.

GCR [14] constrains LLM decoding with a KG-Trie, ensuring that generated reasoning paths strictly follow valid KG structures and reducing hallucinated relations. DualR [28] combines explicit GNN-based graph reasoning with implicit LLM reasoning, using GNN-extracted reasoning chains to guide LLM inference. DoG [29] employs a debate-style multi-agent framework to iteratively refine reasoning paths while focusing on progressively smaller subgraphs. RwT [21] formulates KGQA as a decision process and applies Monte Carlo Tree Search to explore and evaluate reasoning trees with LLM-guided expansion. DP [23] injects KG structural priors into LLM reasoning through progressive distillation and constraint-aware deliberation to improve verifiability.

Retrieval-augmented generation (RAG). RD-P [16] augments retrieval with a discriminator that filters and re-ranks retrieved evidence, aiming to improve evidence quality before generation. EWEK-QA [30] combines KG and web retrieval with adaptive document segmentation to support citation-based question answering without increasing LLM calls. EffiQA [31] adopts a multi-stage pipeline with global planning, efficient KG exploration, and self-reflection to balance answer quality and inference cost.

KG retrieval and subgraph-based methods. KG-CoT [15] extracts explicit reasoning paths from KGs and feeds them to LLMs as structured chain-of-thought prompts. SubgraphRAG [10] retrieves a compact, question-relevant subgraph using lightweight scoring models that incorporate graph structural signals, and performs reasoning over the serialized subgraph with an LLM under a controllable evidence budget. G-Retriever [17] targets question answering over textualized graphs by retrieving relevant graph passages and generating answers in a unified RAG framework. GNN-RAG [18] leverages GNNs to score candidate answers and extract supporting paths, which are then verbalized for LLM-based reasoning. FiDeLiS [32] emphasizes faithful reasoning by anchoring LLM inference to explicitly verified reasoning paths obtained via path search and ranking.

RJE [22] decomposes KGQA into retrieval, judgment, and exploration stages, where reasoning paths are first retrieved and evaluated for sufficiency before conditionally triggering additional evidence exploration, reducing unnecessary retrieval while improving answer coverage.

H.3 Evaluation Metrics

We evaluate model performance from two complementary perspectives: *answer correctness* and *reasoning verifiability*. Standard answer-level metrics capture whether the predicted answers are correct, while our proposed claim-level metrics explicitly assess whether the generated reasoning is grounded in verifiable KG evidence.

Answer correctness. Following prior KGQA work [10, 22], we report **Hit** and **F1** as primary answer-level metrics.

For each test question, let $\hat{A}_i$ denote the predicted answer set and A_i^* the gold answer set. **Hit** is a binary indicator that checks whether at least one gold answer is retrieved:

$$\mathrm{Hit}_i = \mathbf{1}\left[\hat{A}_i \cap A_i^* \neq \varnothing\right]. \tag{146}$$

F1 measures the overlap between predicted and gold answer sets, capturing both precision and coverage:

$$\mathrm{F1}_i = \frac{2 \cdot |\hat{A}_i \cap A_i^*|}{|\hat{A}_i| + |A_i^*|}. \tag{147}$$

Dataset-level scores are obtained by averaging over all test instances.

Claim-level representation. To evaluate reasoning verifiability, we first normalize each generated explanation into a structured set of atomic claims. For a given sample i, the explanation is represented as

$$C_i = \{c_{i1}, c_{i2}, \ldots, c_{iM_i}\}, \tag{148}$$

where each c_{ij} is an atomic factual claim derived from the reasoning trace (e.g., a natural-language realization of a KG triple or a simple logical assertion). Each claim is automatically verified against the retained KG evidence and assigned a support label

$$y_{ij} \in \{\texttt{supported}, \texttt{contradicted}, \texttt{unknown}\}. \tag{149}$$

Here, `supported` indicates that the claim is directly grounded in at least one KG triple, while `contradicted` and `unknown` indicate lack of support for the FactSup computation.

FactSup (Fact-Supported Ratio). FactSup measures the proportion of claims in an explanation that are explicitly supported by KG evidence. For a single sample i, it is defined as

$$\mathrm{FactSup}_i = \frac{\sum_{j=1}^{M_i} \mathbf{1}\left[y_{ij} = \texttt{supported}\right]}{\max(M_i, 1)}. \tag{150}$$

The denominator uses $\max(M_i, 1)$ to avoid division by zero; in practice, explanations without claims yield a FactSup of zero.

At the dataset level, FactSup is computed as the mean over all samples:

$$\mathrm{FactSup} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{FactSup}_i. \tag{151}$$

Intuitively, FactSup captures the *local grounding quality* of generated explanations: a higher FactSup indicates that a larger fraction of the reasoning steps are verifiably supported by the KG, regardless of whether the final answer is correct.

FCR (Faithful Correct Ratio). While FactSup assesses partial grounding, it does not distinguish between explanations with a few unsupported claims and those that are fully faithful. We therefore introduce the **Faithful Correct Ratio (FCR)**, a stricter metric that requires *both answer correctness and support for all extracted explanation claims*.

Let $\mathrm{Hit}_i^{(r)}$ denote whether the predicted answer at reasoning round r is correct, and let $\mathrm{FactSup}_i^{(r)}$ denote the fact-supported ratio of the explanation at that round. A sample is considered *faithfully correct* if: (i) the predicted answer is correct, and (ii) all claims in the explanation are supported by KG evidence. Formally,

$$\mathrm{FCR}_i^{(r)} = \mathbf{1}\left[\mathrm{Hit}_i^{(r)} = 1 \ \wedge \ \mathrm{FactSup}_i^{(r)} = 1\right]. \tag{152}$$

The dataset-level FCR at round r is:

$$\mathrm{FCR}^{(r)} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{FCR}_i^{(r)}. \tag{153}$$

Equivalently, FCR can be interpreted probabilistically as:

$$\mathrm{FCR}^{(r)} = \Pr\left[\text{answer is correct and all claims are supported at round } r\right]. \tag{154}$$

Discussion. FactSup and FCR are complementary to standard answer-level metrics. FactSup quantifies the degree of evidence grounding within explanations, while FCR captures the fraction of predictions that are *simultaneously correct and fully supported by retained evidence*. Neither metric replaces Hit or F1; instead, they explicitly expose cases where a correct answer is obtained via unsupported or hallucinated reasoning, which standard metrics fail to detect. This makes them particularly suitable for evaluating evidence-sensitive KGQA systems under strict evidence budgets.

I Experimental Reproducibility

This section provides a self-contained description of the experimental configuration, inference procedure, and implementation details of CoRE. Our goal is to make the reported results reproducible to the extent possible with public benchmarks, specified model access, and transparent algorithmic execution.

I.1 Backbone Models and Decoding Configuration

We evaluate CoRE using two backbone large language models: **Meta-Llama-3.1-8B-Instruct** and **GPT-4o-mini**. Unless otherwise specified, both models are used with identical decoding configurations to ensure fair and controlled comparisons across backbones.

Specifically, we use greedy decoding with temperature set to 0.0, a maximum generation length of 4000 tokens, and a frequency penalty of 0.16. All main results use a fixed-seed decoding configuration; additional matched ablations use the same decoding and retrieval configuration unless explicitly noted. Selector seed studies and paired bootstrap intervals are specified separately. No model-specific prompt tuning or adaptive decoding heuristics are applied.

Inference with Meta-Llama-3.1-8B-Instruct is performed using vLLM 0.11.0, while GPT-4o-mini is accessed via the OpenAI ChatCompletion API using the same decoding parameters.

I.2 Evidence Budget and Global Hyperparameters

CoRE operates under a fixed evidence budget, defined as the maximum number of KG triples retained in the evidence subgraph at each reasoning round. Unless otherwise stated, the same configuration is used for both WebQSP and CWQ.

The default inference configuration is as follows:

- Maximum number of reasoning rounds: $T = 3$.
- Evidence subgraph budget: $K \in \{10, 30, 60, 100\}$ triples.

- Maximum number of open slots (residual claims) per round: 2.
- Retriever candidate pool size: 500 triples per open slot.

CoRE may terminate before reaching the maximum number of rounds based on a verifiable controller that monitors answer stability, residual resolution, and evidence sufficiency. Final predictions use the declared output-selection rule; the additional matched diagnostics use the gold-free closure selection described in Appendix I.6.

I.3 Neural Retriever and Triple Scoring

CoRE employs a neural triple retriever for both initial evidence construction and residual-driven evidence acquisition. The retriever operates over a candidate triple space constructed from the local KG neighborhood of question-relevant entities and scores each triple independently in a pointwise manner.

Input representations. All textual inputs are embedded using the `gte-large-en-v1.5` encoder. For each dataset, we precompute and cache question embeddings, entity embeddings, and relation embeddings offline for the training, validation, and test splits. At inference time, these cached embeddings are loaded and reused, ensuring deterministic and efficient retrieval.

For a candidate triple (h, r, t), the retriever receives as input: (i) the embedding of the question, (ii) the embeddings of the head and tail entities, (iii) the embedding of the relation, and (iv) auxiliary graph-level features, including a topic-entity indicator that marks whether an entity appears in the question. These features are concatenated and passed to a neural scoring function that outputs a scalar relevance logit for the triple.

Training objective. The retriever is trained as a binary classifier using pointwise binary cross-entropy loss. For each question, triples that appear in gold reasoning paths or supporting evidence are treated as positive examples, while all other candidate triples are treated as negatives. This formulation avoids explicit pairwise ranking losses and allows efficient optimization over variable-sized candidate sets.

Validation and model selection. Retriever training is performed on the training split, with early stopping based on validation `triple_recall@100`. This metric directly reflects the retriever's ability to surface gold supporting triples within a fixed retrieval budget, which is critical for downstream multi-hop reasoning under evidence constraints. The checkpoint achieving the highest validation recall is selected and used during inference.

Inference and integration with CoRE. During inference, the retriever scores all candidate triples independently and returns a ranked list. For initial evidence construction, the top-ranked triples are used to form the initial evidence subgraph. For residual-driven retrieval, each open slot produced by the reasoner is translated into a structured retrieval query, and the retriever is invoked to obtain a candidate pool of up to 500 triples per slot. These candidates are subsequently reranked using slot-specific signals (e.g., anchor and relation matching) before being incorporated into the evidence subgraph.

Training positives are constructed from training-set supporting or path triples; gold answers and paths are not provided to the retriever at test time.

Retriever parameters remain frozen during KGQA inference and are not jointly optimized with the reasoner, ensuring a clear separation between retrieval and reasoning components.

I.4 Structured Reasoning Output and Claim Extraction

At each reasoning round, the LLM is prompted to produce a structured output consisting of three components: a proposed answer, a set of atomic claims, and a set of open slots. Claims are explicitly labeled as `supported`, `unknown`, or `contradicted`, and may cite one or more KG edge identifiers. Open slots are formulated as question-style queries augmented with anchor entities and relation hints, making them directly usable as retrieval queries in subsequent rounds.

Importantly, claim labels are not taken verbatim from the LLM output. A deterministic post-processing module enforces grounding constraints: claims labeled as `supported` or `contradicted` without valid KG citations are downgraded to `unknown`, cited edge identifiers must correspond to triples present in the current evidence subgraph, and unsupported but question-relevant claims are converted into executable open slots. This design ensures that all supported claims are verifiable with respect to the retained KG evidence.

We note that the reasoning residual is a noisy approximation of unresolved information and may miss or misclassify claims. All theoretical results rely only on the existence of residual signals, while the empirical evaluation assesses the robustness of this approximation in practice.

I.5 Learned Projection Scorer in CoRE-RL

CoRE-RL adds a lightweight learned projection scorer on top of the frozen retriever and the base CoRE loop. For each round, the candidate set is $Z_t \cup O_{t+1}$ and the context is $(q, Z_t, \hat{a}_t, R_t)$. The scorer consumes edge-level and state-level features: retriever score, anchor and relation-hint matches, answer overlap, whether the edge is already retained, slot coverage, round index, residual count, and budget fill ratio. We train either a linear model or a shallow MLP with weighted binary/ranking loss from rollout traces. The target label records whether the edge is retained by the teacher projection; rollout weights follow the F1, residual-mass, and support-gain terms in Section 5. The additional feature-corruption and transfer component diagnostics use logged-projection targets with uniform weights. At test time, no extra LLM call is introduced: CoRE-RL replaces the residual-support projection score with g_θ and retains the top-K edges.

I.6 Inference Procedure of CoRE

We now describe the complete inference procedure of CoRE for a single test instance.

At round t, the LLM is conditioned on the question and the current evidence subgraph Z_t and generates one or more structured reasoning candidates. Each candidate is post-processed to enforce grounding constraints and is assigned a grounding-aware score that favors higher claim support and valid evidence citation. The highest-scoring candidate is selected as the reasoning output for the current round.

Based on this structured output, a verifiable controller determines whether to terminate or retrieve additional evidence. Execution ends when the controller finds sufficient evidence, a no-gain boundary is reached, or $T = 3$ rounds have executed. In the additional matched diagnostics, no gain means that the supported-claim fraction minus 0.05 times the open-slot count fails to improve after retrieval. Final output selection uses only observed support: round 1 remains eligible, and a later round is eligible when its closure score (supported claims minus unsupported claims minus open slots) improves by at least 4 over round 1; the best eligible round is returned, with earlier rounds breaking ties. Terminal status is then assigned to that output. If retrieval is triggered, each open slot is translated into a structured retrieval query, which is passed to the neural retriever to obtain a ranked list of candidate triples.

Candidate triples are globally reranked using retrieval scores and slot-specific signals (e.g., anchor and relation matching). Up to a fixed number of top-ranked triples are added to the evidence subgraph. When the evidence budget is exceeded, CoRE applies a claim-aware projection that prioritizes triples contributing to unresolved residual claims, rather than retaining evidence purely by retrieval score. The updated subgraph Z_{t+1} is then used for the next reasoning round.

I.7 Groundedness Metrics and Evaluation Protocol

FactSup measures the supported fraction of extracted explanation claims in retained evidence, and FCR is $\mathbf{1}[\mathrm{Hit}_i = 1 \wedge \mathrm{FactSup}_i = 1]$, averaged over all questions. Hit, F1, FactSup, and FCR include all evaluated questions, irrespective of terminal status. Strict Hit/F1 are computed only on `SUPPORTED` outputs, with coverage $n_{\texttt{SUPPORTED}}/N$ reported separately. Empty explanations have FactSup zero. FCR checks extracted claims, whereas `SUPPORTED` additionally requires no open slots: a correct answer with fully supported generated claims can still have an unresolved slot. On the 100-question WebQSP diagnostic, this occurs in 15 heuristic and 16 noise-trained outputs. Neither evidence metric establishes full-KG completeness or token-level causal faithfulness.

I.8 Baselines and Fair Comparison

We distinguish the benchmark comparison in Table 1 from controlled evaluations with matched retrieval and inference settings. Table 11 reports evidence-update controls; the cited main-table baselines retain the backbones shown in their row labels. Specifically, we select two strong baselines covering complementary paradigms: (i) **SubgraphRAG**, representing static evidence selection with single-pass reasoning, and (ii) **ToG**, representing agent-based iterative retrieval without explicit evidence retention.

For each baseline, we replace the original retrieval component with our neural retriever described in Section I.3, and enforce the same evidence budget K and candidate pool size (up to 500 triples per query or step). All methods use identical backbone models (GPT-4o-mini or Meta-Llama-3.1-8B-Instruct) and decoding configurations. No baseline is allowed access to additional retrieval budget, oracle evidence, or model-specific prompt tuning.

The controlled evidence-update comparison fixes candidate resources, backbone, $K = 60$, and $T = 3$; query-only, local-frontier, and progressive add-only are mechanism controls rather than renamed reproductions of the cited methods.

I.9 Implementation Details and Hardware

All experiments are implemented in PyTorch 2.8.0 with Torch-Geometric 2.7.0. Inference with Meta-Llama-3.1-8B-Instruct is performed using vLLM 0.11.0.

Experiments are conducted on a workstation equipped with two NVIDIA RTX PRO 6000 Blackwell GPUs, Intel Xeon W7-series CPUs, and 128GB DDR5 ECC memory. Main benchmark results use one fixed-seed decoding configuration. Additional selector robustness and transfer studies use 20 seeds, and matched question-level comparisons report paired bootstrap intervals where specified. Bootstrap resampling measures uncertainty over evaluated questions rather than variability across repeated API executions.

Compute scope, assets, and responsible use. Local experiments use at most two RTX PRO 6000 GPUs, while GPT-4o-mini experiments use API inference rather than local GPU execution. Wall-clock time depends on API latency, evaluated sample counts, and batching choices, so we report online LLM usage in Table 6 and keep all budget settings explicit in Appendix I.2. We use public WebQSP, CWQ, and Freebase resources and do not redistribute modified datasets or model weights. Experiments rely on existing LLM backbones and embedding models under their original access terms; researchers reproducing the results should follow the corresponding dataset, model, and service licenses. The study does not involve human-subject data collection. Its intended positive use is evidence-grounded question answering, while a possible risk is over-reliance on KG-backed answers when the underlying KG is incomplete or outdated. CoRE outputs should therefore be treated as evidence-aware decision support rather than as standalone authority in high-stakes settings.

J CoRE Algorithm

Algorithm 1 provides a complete procedural description of **CoRE**, a closed-loop framework that co-evolves reasoning trajectories and a budgeted evidence subgraph under fixed context constraints. The algorithm alternates between evidence-constrained reasoning, extraction of answer-determining but unsupported residual claims, and residual-driven evidence retrieval with budget-aware subgraph projection. Execution ends at evidence sufficiency, the no-gain boundary, or the round cap; output support is recorded separately.

State variables and initialization. CoRE maintains three explicit state variables across reasoning rounds: the retained evidence subgraph Z_t, the predicted answer $\hat{a}_t$, and the residual claim set R_t. The algorithm is initialized with a seed subgraph Z_0, typically obtained via lightweight query-centered retrieval, and an empty previous answer $\hat{a}_{-1}$ to enable answer-stability checking in the first iteration. These variables form the observable information state used by the controller.

Subgraph-constrained reasoning. At each iteration t, the LLM reasoner is conditioned on the question q and the current evidence subgraph Z_t. Instead of producing a single output, the model

Algorithm 1 CoRE family: residual-coupled reasoning and evidence projection

Input: question q; initial evidence subgraph Z_0; evidence budget K; grounding threshold τ; candidate count N; top-claim count M; optional projection scorer g_θ; maximum rounds $T = 3$.
Output: predicted answer $\hat{a}$ and support status σ.

Here $\mathrm{ID}(Z_t)$ denotes the identifiers of triples in Z_t.

Initialize $t \leftarrow 0$, $Z_t \leftarrow Z_0$, $\hat{a}_{-1} \leftarrow \varnothing$.

repeat

(Subgraph-constrained reasoning)

Sample N candidate outputs $\{\hat{y}_t^{(i)}\}_{i=1}^{N}$ from the LLM conditioned on (q, Z_t),
each containing a proposed answer and a reasoning trace.
For each candidate $\hat{y}$, extract atomic claims $C(\hat{y})$ and cited evidence sets $\mathcal{E}(c)$ (if any).

(Claim-level support computation)

for each candidate $\hat{y}$ **do**
  for each claim $c \in C(\hat{y})$ **do**
    Set $\mathcal{E}(c) \leftarrow \mathcal{E}(c) \cap \mathrm{ID}(Z_t)$.
    Set $\mathrm{Sup}(c; Z_t) = 0$ if $\mathcal{E}(c) = \varnothing$; otherwise compute $\max_{e \in \mathcal{E}(c)} s(c, T(e))$.
  end for
end for

(Candidate scoring and selection)

For each candidate $\hat{y}$, let $\mathrm{Top}_M(C(\hat{y}))$ be the M claims with largest $\mathrm{Sup}(c; Z_t)$.
Compute candidate score $\mathrm{Score}(\hat{y}; Z_t) = 1 - \prod_{c \in \mathrm{Top}_M(C(\hat{y}))}(1 - \mathrm{Sup}(c; Z_t))$.
Select $\hat{y}_t^\star = \arg\max_{\hat{y}} \mathrm{Score}(\hat{y}; Z_t)$ and extract $\hat{a}_t$.

(Residual extraction)

Construct the residual set $R_t = \{c \in C(\hat{y}_t^\star) : \mathrm{Sup}(c; Z_t) < \tau \ \wedge \ \mathrm{Rel}(c; q, \hat{a}_t) = 1\}$.

(Controller / stopping rule)

if evidence is sufficient, no gain is observed, or $t = T - 1$ **then break**.

(Open-slot construction and retrieval)

Initialize $O_{t+1} \leftarrow \varnothing$.
for each $c \in R_t$ **do**
  Construct open slot $\rho(c) = \langle \mathrm{qry}(c), A(c), \mathrm{hint}(c) \rangle$.
  Retrieve candidate triples and add to O_{t+1}.
end for

(Residual-support scoring)

For each $e \in O_{t+1}$, compute $\Delta(e; R_t) = \sum_{c \in R_t}(1 - \mathrm{Sup}(c; Z_t))\, \mathbf{1}[s(c, T(e)) > \tau]$.

(Budgeted projection)

Update the evidence subgraph $Z_{t+1} \leftarrow \mathcal{P}_K(Z_t \cup O_{t+1}; R_t)$, retaining the top-$K$ triples ranked by $\Delta(\cdot; R_t)$ in CoRE or by g_θ in CoRE-RL.
Set $\hat{a}_{t-1} \leftarrow \hat{a}_t$, then $t \leftarrow t + 1$.

until execution ends.

Select an executed output using the declared gold-free selection rule.
Return its answer and `SUPPORTED` if its nonempty claims are supported and no slots remain;
otherwise return `UNRESOLVED`, or `UNVERIFIED` for an empty/invalid verdict.

samples multiple candidate reasoning traces, each consisting of a proposed answer and an explicit reasoning trace. Each trace is normalized into a set of atomic claims $C(\hat{y})$, optionally annotated with citations to evidence edges in Z_t. This claim-level representation enables subsequent verification and residual identification.

Candidate scoring and selection. Each candidate reasoning trace is evaluated using a soft support score that measures how well its claims are grounded in the current evidence subgraph. The score in Algorithm 1 is $1 - \prod_{c \in \mathrm{Top}_M(C(\hat{y}))}(1 - \mathrm{Sup}(c; Z_t))$; support status is checked separately from this ranking score. This selection strategy allows CoRE to tolerate unsupported intermediate hypotheses caused by KG incompleteness, while preventing the final answer from being selected solely based on ungrounded speculation. If no candidate fully satisfies the grounding constraint, the highest-scoring candidate is selected provisionally and refined in subsequent iterations.

Residual claim extraction. After selecting the best-supported reasoning trace, CoRE identifies the reasoning residual R_t, defined as the set of claims that are necessary for determining the predicted

answer but remain insufficiently supported by the current evidence subgraph. Unsupported claims are not treated as incorrect; instead, residuals explicitly encode decision-critical evidence gaps that must be resolved to complete faithful reasoning.

Controller and stopping criterion. CoRE employs a simple, verifiable controller that decides whether to retrieve or end execution. Evidence sufficiency, lack of further support gain, and the round cap bound execution. The output receives a separate support status; controller termination alone does not establish evidence closure. Appendix I.6 specifies the no-gain proxy and output-selection rule for the additional matched diagnostics.

Open-slot construction and residual-driven retrieval. For each residual claim, CoRE constructs a structured open slot that specifies the type of evidence required to support the claim. An open slot combines a natural-language query derived from the claim, anchor entities identified via entity linking, and relation or type constraints induced by the KG schema. These slots are passed to an offline-trained neural retriever, which ranks KG triples satisfying the specified constraints. Retrieved triples from all residual claims are merged into a candidate pool for subgraph updating.

Residual-support scoring. Because multiple residual claims may retrieve overlapping or competing evidence, CoRE prioritizes candidate triples using a residual-support gain function. Each triple is scored by how much it contributes to resolving unresolved residuals, weighted by the degree of support deficit of each claim. This mechanism prioritizes evidence that addresses poorly supported, answer-determining claims rather than reinforcing already grounded facts.

Budgeted claim-conditioned projection. To enforce the fixed evidence budget, CoRE updates the retained evidence subgraph via a claim-conditioned projection operator. The union of the current subgraph and newly retrieved candidates is ranked by residual-support gain, and only the top-ranked triples are retained. This projection step prevents evidence dilution under tight budgets while preserving triples that are most critical for resolving outstanding reasoning gaps.

Iteration and termination. After updating the evidence subgraph, CoRE proceeds to the next reasoning round using the updated state $(Z_{t+1}, \hat{a}_t)$. Reasoning and evidence updating continue until the controller or finite horizon ends execution; the selected answer is returned together with its support status.

Summary. Overall, Algorithm 1 instantiates a residual-coupled reasoning–retrieval loop in which reasoning exposes missing decision-critical evidence, residuals guide targeted retrieval, and claim-aware projection maintains a compact and decision-relevant evidence state. This design enables efficient and interpretable multi-hop reasoning under strict evidence budgets and directly realizes the theoretical framework developed in the main text.

K Experimental Details Analysis

K.1 Detailed Analysis of Multi-round Accuracy Trends

Table 19 reports detailed performance across successive CoRE reasoning rounds on WebQSP and CWQ, under both `gpt-4o-mini` and `Meta-Llama3.1-8B` backbones. The reported aggregate Hit, F1, and EM improve across rounds. The matched evidence-update controls in Table 11 separately compare residual-driven updating with alternative retrieval signals.

A salient pattern is that the largest absolute performance gains occur between the first and second rounds. For instance, on CWQ with `gpt-4o-mini`, F1 increases by $+5.18$ points from round 1 to round 2 ($64.30 \rightarrow 69.48$), whereas the gain from round 2 to round 3 is smaller but still positive ($+0.87$). This behavior aligns with CoRE's design: early rounds primarily correct answer-level uncertainty caused by incomplete initial evidence, while later rounds focus on consolidating and verifying supporting facts once a plausible answer has emerged.

Importantly, the improvement trend is not confined to a single backbone. Although `Meta-Llama3.1-8B` exhibits lower absolute scores than `gpt-4o-mini`, it shows comparable relative gains across rounds. On CWQ, F1 improves from 51.11 to 60.16 (+9.05) between

Table 19: Answer accuracy across CoRE rounds on WebQSP and CWQ. P/R denote precision/recall, and EM denotes exact match.

Dataset	Model	Round	Hit	P	R	F1	EM
WebQSP	gpt-4o-mini	1	89.33	82.02	79.35	77.31	61.17
		2	92.61	84.71	81.47	79.61	63.30
		3	93.46	85.86	82.12	80.45	64.01
	Meta-Llama3.1-8B	1	84.81	77.45	69.60	68.18	49.18
		2	88.71	83.24	73.57	73.00	54.00
		3	89.51	84.80	74.81	74.45	55.77
CWQ	gpt-4o-mini	1	72.76	68.02	65.41	64.30	53.38
		2	78.79	73.96	70.12	69.48	57.40
		3	79.89	74.96	71.00	70.35	58.14
	Meta-Llama3.1-8B	1	59.60	54.97	52.83	51.11	41.10
		2	68.90	65.44	60.50	60.16	49.20
		3	72.50	68.85	63.91	63.48	52.50

rounds 1 and 2, and further to 63.48 at round 3. This indicates that CoRE's residual-driven evidence updating compensates for weaker parametric reasoning ability by explicitly repairing missing graph facts, making the framework robust across LLM capacities.

We further note dataset-dependent differences. On WebQSP, which contains a higher proportion of single-hop or weakly compositional queries, round-wise gains are smaller in magnitude but remain consistent. In contrast, CWQ benefits substantially more from additional rounds, especially in the early stage. This gap reflects the higher prevalence of compositional and multi-hop questions in CWQ, where missing intermediate relations frequently prevent correct reasoning in the first round. Multi-round reasoning enables CoRE to iteratively recover such intermediate evidence, which is difficult to achieve with one-shot retrieval.

Beyond F1 and Hit, Exact Match (EM) also improves steadily across rounds. Although EM is a stricter metric, its monotonic increase suggests that later rounds do not merely expand answer coverage but also reduce spurious or partially correct predictions by aligning the final answer with a more complete and coherent evidence subgraph.

Overall, these results support two key conclusions. First, CoRE's multi-round mechanism yields diminishing but non-vanishing returns: early rounds drive major answer corrections, while later rounds contribute incremental yet consistent improvements. Second, accuracy gains should be interpreted jointly with evidence sufficiency: even when answer-level metrics plateau, subsequent rounds continue to refine and verify the retained evidence, a phenomenon further corroborated by the verifiability analysis in Table 3.

K.2 Fine-grained Effects of Evidence Budget and Reasoning Rounds

Table 20 provides a detailed breakdown of answer-level performance under varying evidence budgets and reasoning rounds on CWQ and WebQSP. Unlike one-shot retrieval baselines, where performance typically degrades sharply as the evidence budget shrinks, CoRE exhibits a markedly different behavior: performance is jointly shaped by the evidence budget and the number of reasoning rounds, with the latter partially compensating for tight budget constraints.

Impact of Evidence Budget within a Fixed Round. For a fixed reasoning round, increasing the evidence budget consistently improves performance on both datasets, but the magnitude of improvement varies substantially across rounds. At round 1 on CWQ, F1 increases from 45.96 at $K{=}10$ to 51.11 at $K{=}100$, reflecting the well-known limitation of one-shot reasoning under sparse initial evidence. However, the marginal gain from enlarging the budget diminishes as the number of rounds increases. At round 3, the F1 gap between $K{=}30$ and $K{=}100$ narrows to less than two points (61.56 vs. 63.48), indicating that later rounds reduce reliance on large initial evidence contexts by selectively repairing missing facts.

Table 20: Performance (%) under different evidence budgets across reasoning rounds on CWQ and WebQSP. P/R denote precision/recall, and EM denotes exact match. All results are reported on 1,000 randomly sampled test questions for each dataset.

Dataset	Round	Budget K	Hit (%)	P (%)	R (%)	F1 (%)	EM (%)
CWQ	1	10	53.3	50.46	45.68	45.96	36.90
		30	58.30	54.19	50.55	49.73	39.60
		60	60.39	55.69	53.59	51.87	41.66
		100	59.60	54.97	52.83	51.11	41.10
	2	10	63.90	60.99	54.75	55.53	44.70
		30	67.40	63.64	58.57	58.45	47.50
		60	69.43	64.98	61.33	60.42	49.41
		100	68.90	65.44	60.50	60.16	49.20
	3	10	67.70	64.63	58.00	58.68	46.70
		30	70.60	67.10	61.31	61.56	50.20
		60	72.12	67.86	63.82	63.06	51.88
		100	72.50	68.85	63.91	63.48	52.50
WebQSP	1	30	83.60	78.40	67.78	68.32	51.00
		60	85.10	78.30	69.63	68.65	50.40
		100	85.20	78.30	69.65	68.64	50.30
	2	30	87.50	84.10	71.80	73.10	57.00
		60	88.30	83.45	72.66	72.49	54.60
		100	89.40	85.10	73.82	73.79	56.00
	3	30	88.40	85.68	73.45	74.89	59.10
		60	89.40	85.10	73.82	73.79	56.00
		100	89.70	85.15	74.37	74.15	55.90

A similar but milder trend is observed on WebQSP. Even at round 1, increasing the budget beyond $K{=}30$ yields minimal gains, suggesting that many WebQSP questions can already be addressed with compact evidence. This difference highlights the stronger sensitivity of CWQ to evidence scarcity, consistent with its higher compositional and multi-hop complexity.

Impact of Reasoning Rounds under Fixed Budgets. More importantly, for any fixed budget, increasing the number of reasoning rounds consistently yields monotonic improvements across all metrics. On CWQ with the smallest budget ($K{=}10$), F1 improves from 45.96 at round 1 to 55.53 at round 2 and further to 58.68 at round 3. Notably, the absolute gain from adding rounds at $K{=}10$ is larger than the gain from increasing the budget from $K{=}10$ to $K{=}100$ at round 1. This pattern demonstrates that iterative reasoning with targeted evidence updates is more effective than simply providing more evidence upfront.

On WebQSP, although the gains are smaller in absolute terms, the same monotonic trend holds. At $K{=}30$, F1 increases from 68.32 (round 1) to 73.10 (round 2) and further to 74.89 (round 3), indicating that multi-round reasoning continues to refine answer coverage and correctness even when initial evidence is already relatively sufficient.

Budget–Round Trade-offs and Evidence Efficiency. A key observation from Table 20 is that CoRE achieves strong performance through different budget–round trade-offs. For example, on CWQ, round 3 with $K{=}30$ (F1 61.56) approaches or exceeds round 1 with $K{=}100$ (F1 51.11) by a large margin. This shows that CoRE can trade additional reasoning steps for reduced evidence budgets, a property that is unattainable for static retrieve-then-reason pipelines.

This behavior directly reflects CoRE's closed-loop design. Instead of allocating the entire budget upfront, CoRE incrementally retrieves and retains triples that address unresolved, answer-determining claims. As a result, later rounds focus the limited evidence budget on decision-critical facts, mitigating the negative effects of budget eviction and evidence dilution.

Dataset-dependent Robustness under Tight Budgets. Finally, the robustness under strict budgets differs across datasets in a principled way. CWQ benefits substantially from both increased budgets and additional rounds, confirming that compositional queries require iterative evidence repair to

Evidence-Constrained Reasoning Prompt Prompt Input.
A natural-language question together with a retained evidence subgraph, represented as a set of knowledge graph triples annotated with stable edge IDs.
Task.
Given the question and the evidence edges, generate a grounded answer together with a structured reasoning trace.
The model must decompose its reasoning into atomic claims and explicitly indicate missing or unsupported facts as retrieval-ready open questions.
Output Format (JSON).
`ans`: short answer string or a list of answers.
`c`: a list of atomic claims, each formatted as `[claim_text, [edge_id_strings], status]`, where `status` is one of `supported`, `unknown`, or `contradicted`.
`o`: a list of open slots, each formatted as `[missing_fact_question, [anchors], [relation_hints]]`.
Constraints.
All supported claims must cite valid edge IDs from the provided evidence.
If a required fact cannot be supported, it must be expressed as an open slot phrased as a question suitable for retrieval.
No information outside the given evidence is allowed.

Figure 5: Prompt template for subgraph evidence-constrained reasoning

Baseline Answer Generation Prompt Prompt Input.
A question and a set of knowledge graph triples.
Task.
Generate all possible answers to the question based on the provided triples.
Output Format.
Each answer is listed on a separate line, prefixed with `ans:`.
No explicit evidence grounding or reasoning structure is required.

Figure 6: Baseline prompt used for unconstrained answer generation in comparison methods.

recover missing intermediate relations. In contrast, WebQSP shows strong robustness even at moderate budgets, with multi-round reasoning primarily improving recall and exact match rather than compensating for severe evidence gaps.

Taken together, these results demonstrate that CoRE's robustness under tight evidence budgets is not accidental. It arises from the interaction between residual-driven retrieval and claim-conditioned evidence projection, which allows the framework to remain effective even when only a small number of triples can be retained at each step.

K.3 Prompt Templates in CoRE

Evidence-Constrained Reasoning Prompt. Figure 5 shows the prompt template used for evidence-constrained reasoning in CoRE. This prompt serves as the primary interface between the large language model and the retained evidence subgraph at each iteration. Given a natural-language question and a set of knowledge graph triples annotated with stable edge identifiers, the model is instructed to produce a grounded answer together with a structured reasoning trace. Reasoning is decomposed into a set of atomic claims, each of which must either be explicitly supported by cited evidence edges or marked as unsupported. Facts that are required for answering the question but cannot be grounded in the current evidence are explicitly exposed as open slots phrased as retrieval-ready questions. By forcing the model to surface missing evidence rather than implicitly hallucinating it, this prompt enables reasoning to proceed under incomplete evidence and provides the necessary signals for subsequent evidence acquisition in the closed-loop CoRE framework.

Baseline Answer Generation Prompt. Figure 6 illustrates the baseline prompt used in comparison methods. This prompt represents a conventional knowledge graph–grounded answer generation

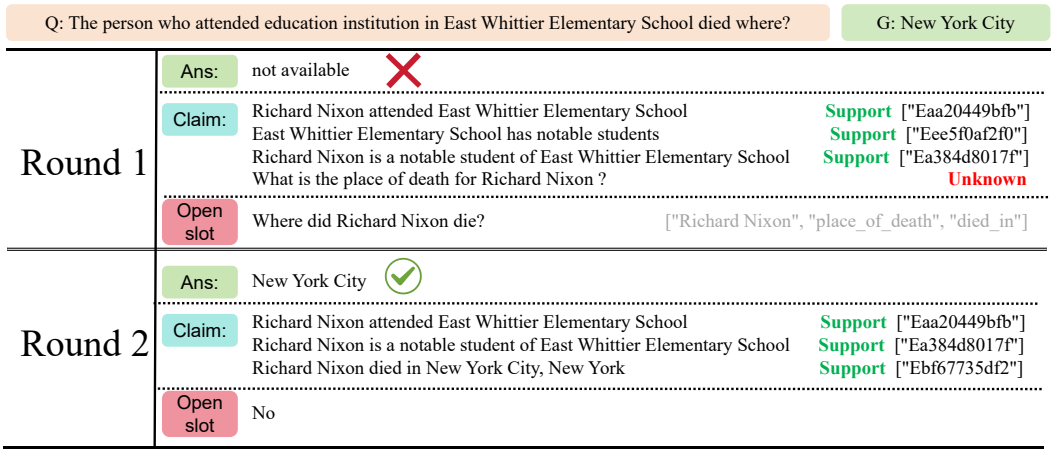


Figure 7: Case 1.

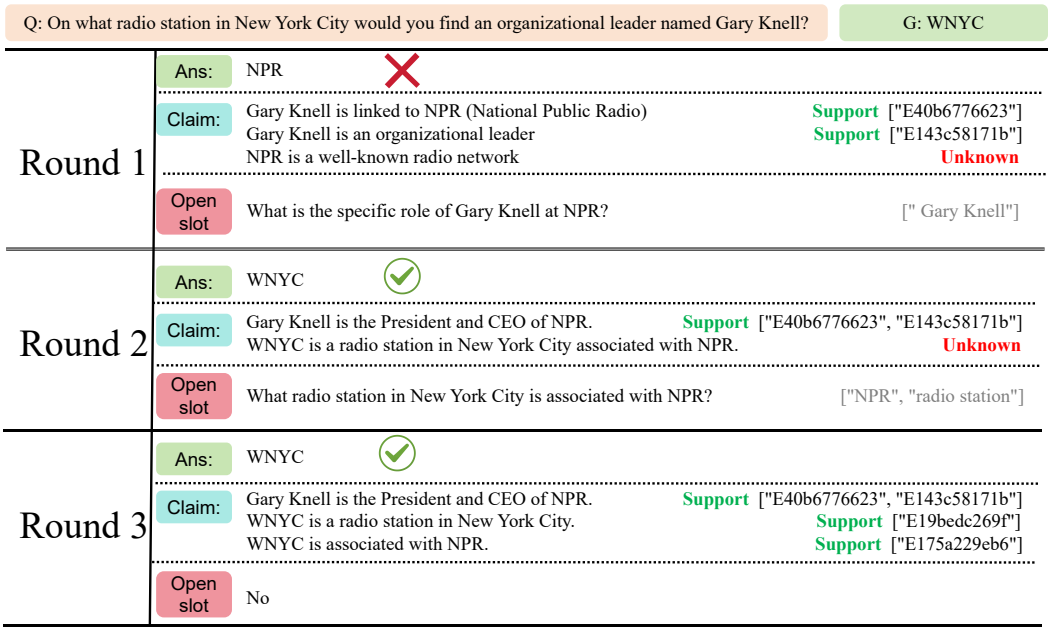


Figure 8: Case 2.

setting in which the model is asked to directly produce answers from a set of provided triples. No explicit reasoning structure, evidence citation, or signaling of missing information is required. Consequently, intermediate reasoning states and unsupported assumptions remain implicit, and the resulting answers cannot be systematically verified or refined through iterative evidence updates. We use this prompt to instantiate unconstrained or retrieve-then-reason baselines, highlighting the limitations of open-loop prompting relative to the structured, closed-loop prompting strategy adopted by CoRE.

K.4 Case Study

We present two representative case studies to illustrate how CoRE (i) corrects erroneous predictions caused by missing evidence, and (ii) continues evidence acquisition even after a correct answer is identified, in order to ensure full evidential grounding. Both examples are drawn from CWQ-style compositional questions and are visualized in Figures 7 and 8.

Case Study 1: Correcting an Initial Failure via Residual-Driven Retrieval **Question.** The person who attended an education institution in East Whittier Elementary School died where?

Gold answer. New York City.

At the first reasoning round, the retained evidence subgraph contains sufficient information to identify the relevant individual. In particular, multiple supported claims establish that Richard Nixon attended East Whittier Elementary School and that he is a notable student of the institution. However, the evidence subgraph does not include any facts describing Richard Nixon's place of death.

As a consequence, the LLM is unable to produce a grounded answer and returns *not available*. The extracted atomic claims reveal that all supported claims pertain to educational background, while the answer-determining attribute concerning the place of death remains unsupported. This deficiency is explicitly captured by the reasoning residual, which generates open slots querying Richard Nixon's place of death under appropriate relation constraints (e.g., `place_of_death`, `died_in`).

Since unresolved answer-determining residual claims remain, the controller selects a retrieval action. Guided by these residuals, the updater augments the evidence subgraph with additional biographical facts. At the second round, the updated subgraph explicitly supports the claim that Richard Nixon died in New York City, New York. Conditioning on this refined evidence view, the LLM produces the correct answer. As all answer-determining claims are now directly supported by retained KG triples, the reasoning residual becomes empty and the controller terminates.

This example illustrates how CoRE corrects an initial failure caused by missing attribute-level evidence by converting unsupported reasoning requirements into structured retrieval signals, enabling precise recovery of the necessary facts under a fixed evidence budget.

Case Study 2: From an Incorrect Answer to Evidence-Verified Resolution **Question.** On what radio station in New York City would you find an organizational leader named Gary Knell?

Gold answer. WNYC.

In the first round, the evidence subgraph supports claims indicating that Gary Knell is an organizational leader at National Public Radio (NPR). Based on this information, the LLM outputs *NPR* as the answer. While plausible, this prediction is incorrect because the question requires a concrete radio station located in New York City rather than an organization-level entity.

The reasoning residual exposes this mismatch by generating open slots querying which radio station in New York City is associated with NPR. Consequently, the controller triggers retrieval to address this unresolved reasoning gap. After updating the evidence subgraph, the LLM identifies WNYC as the correct answer in the second round.

However, although the answer is now correct, the composite claim that WNYC is a radio station in New York City associated with NPR is not yet fully supported by explicit KG evidence. While Gary Knell's leadership role at NPR is grounded, the station-level associations remain partially unsupported. Since answer-determining residual claims persist, the controller does not terminate.

In the third round, retrieval is driven exclusively by these residual claims. The updater incorporates evidence establishing that WNYC is a radio station located in New York City and that it is associated with NPR. With all answer-determining claims fully supported by retained triples, the predicted answer remains unchanged and the controller terminates.

This case demonstrates that CoRE distinguishes answer discovery from evidence sufficiency. Even after a correct answer is obtained, the system continues to update the evidence subgraph until the reasoning chain is fully grounded, thereby improving verifiability without altering the final prediction.

L Limitations

While CoRE tightly couples reasoning with evidence selection through residual-driven retrieval and claim-conditioned subgraph updates, the final answer and reasoning traces are still produced by a general-purpose LLM that is only conditioned on the retained evidence, rather than explicitly aligned or constrained by it. Consequently, the generation process may still introduce unsupported intermediate claims, overgeneralize relational semantics, or mix evidence-grounded and ungrounded reasoning steps, even when the correct answer is identified. This limitation stems from the fact that

CoRE governs the availability and composition of evidence under a fixed budget, but does not directly control how the LLM uses that evidence during decoding.

Support checks apply to extracted claims. A necessary fact that remains absent from every selected trace produces no residual, so repeated reasoning does not ensure complete critical-claim coverage.

Addressing this gap is a promising direction for future work. One path is test-time alignment, which strengthens verifiability without modifying model parameters, for example via constrained decoding, mandatory evidence referencing, or post-hoc verification and re-ranking based on claim–evidence consistency. A complementary path is model editing or evidence-aware adaptation, where the model itself is trained or edited to internalize alignment signals, explicitly discouraging unsupported claims and promoting faithful use of provided evidence. Together, these directions can further enhance CoRE by ensuring that generation is not only informed by selected evidence, but explicitly governed by it.

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 abstract and Section 1 state the dynamic evidence-identification bottleneck, the CoRE/CoRE-RL contributions, and the empirical scope. Section 6 reports the corresponding results on WebQSP and CWQ.

   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 negative 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 Section 7 and Appendix L, including dependence on claim extraction, KG coverage, and remaining LLM faithfulness issues.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while a negative answer 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 theoretical assumptions are stated with the main results in Section 4. Complete proofs and instantiations are provided in Appendices B, C, D, E, and F.

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: Section 6, Appendix H, and Appendix I describe the datasets, metrics, budgets, prompts, algorithms, backbone models, decoding configuration, retriever, and projection settings needed to reproduce the reported conclusions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a negative 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 supplementary material includes an anonymized code package with a README, command templates, environment variables, and scripts for trace collection, learned projection training, optional controller training, and learned-policy evaluation. The experiments use public WebQSP, CWQ, and Freebase resources, and the paper specifies the prompts, hyperparameters, budgets, and model settings needed for reproduction.

   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 a negative answer is acceptable. 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: Appendix H reports dataset splits, baselines, and metrics, while Appendix I specifies backbone models, decoding settings, budgets, retriever training, learned projection training, and the inference protocol.

   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: Appendix I.9 states that the reported results use fixed-seed deterministic decoding on matched test samples. The main claims are supported by exact sample-level evaluation on the same datasets, matched ablations, budget studies, and round-wise analyses; these provide the appropriate variability controls for the deterministic evaluation protocol used in this paper.

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: Appendix I.9 reports the software stack, local hardware, GPU type and memory class, fixed-seed execution protocol, and the API/local inference distinction. Appendix I.2 and Table 6 further specify evidence budgets and online LLM usage, which are the dominant reproducibility-relevant compute factors.

   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 work uses established public KGQA benchmarks and standard LLM inference/retrieval components, with no human-subject data collection, private data, or high-risk deployment. We are not aware of any aspect of the study that conflicts 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 give a negative answer, 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 method can improve evidence-grounded question answering by making missing evidence and support checks more explicit. Possible negative impacts include over-reliance on incomplete KGs or deployment in decision-support settings without domain validation; the limitations in Section 7 and Appendix L emphasize that evidence support does not guarantee full causal faithfulness.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or a negative answer, 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 does not release a new pretrained/generative model or scraped dataset with high misuse risk. It evaluates an algorithmic framework on existing KGQA benchmarks and existing LLM backbones under their original access controls.

    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: The paper credits the datasets, KG resources, baselines, embedding model, and LLM backbones through citations and uses them only for research evaluation. Appendix I.9 states that we do not redistribute modified datasets or model weights and that reproduction should follow the original dataset, model, and service licenses.

    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: [N/A]

    Justification: The paper introduces a method and evaluation protocol, but does not introduce or release a new dataset, pretrained model, or benchmark asset.

    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 study does not involve crowdsourcing or human-subject experiments.

    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 study does not involve human subjects, so IRB approval or equivalent review is not applicable.

    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: [Yes]

    Justification: LLM usage is central to the method and evaluation. Sections 1, 5, and Appendix I describe the LLM reasoner, backbone models, decoding setup, structured outputs, and prompt templates.

    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.