STRIDE: Structural Reasoning for In-context Demonstration Extraction in Multi-hop Question Answering

Shiwei Chen[1,2], **Bin Liang**[1,4,5], **Yue Yu**[2], **Kam-Fai Wong**[4,5], **Hui Wang**[2,*], **Ruifeng Xu**[1,2,3,*]
[1]Harbin Institute of Technology (Shenzhen) [2]Pengcheng Laboratory
[3]Shenzhen Loop Area Institute [4]The Chinese University of Hong Kong
[5]Ministry of Education Key Laboratory of High Confidence Software Technologies, CUHK
[*]**Correspondence:** xuruifeng@hit.edu.cn,wangh06@pcl.ac.cn

Abstract

In-context learning (ICL) for multi-hop question answering (MHQA) depends critically on the choice of demonstrations. Yet most existing methods prioritize semantic similarity, even though semantically similar examples may follow different reasoning paths and therefore be poor demonstrations for the target query. We present **STRIDE** (**ST**ructural **R**easoning for **I**n-context **D**emonstration **E**xtraction), a structure-centric framework for multi-hop QA. STRIDE first infers the latent reasoning structure of a query. It then performs positive–contrastive structural routing, selecting structurally compatible demonstrations as positives and semantically plausible but structurally mismatched examples as contrastives. Finally, it constructs structural prompts that expose compact, role-consistent reasoning traces over supporting evidence. Together, it turns demonstration retrieval from semantic matching into structure-aware problem-solving guidance. Empirical results show that STRIDE consistently improves downstream QA performance across settings.

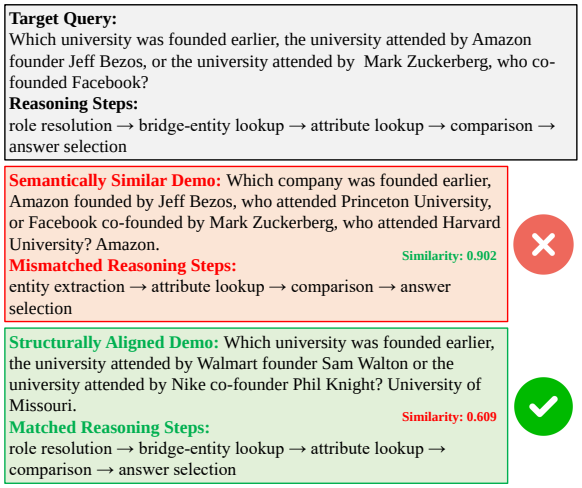


Figure 1: Motivating example showing the gap between semantic similarity and reasoning compatibility. The grey panel presents the target query and its required reasoning structure. The red panel shows a semantically similar demonstration that induces a mismatched reasoning structure, whereas the green panel shows a less semantically similar but structurally aligned demonstration. Semantic similarity scores are computed on Huggingface (Wolf et al., 2020) using `all-MiniLM-L6-v2` (Reimers and Gurevych, 2019).

1 Introduction

Large language models (LLMs) have shown remarkable capacity for complex tasks through in-context learning (ICL), where a handful of demonstrations can steer the model toward task-appropriate reasoning (Dong et al., 2022; Chen et al., 2023; Luo et al., 2024b; Mao et al., 2024; Schoch and Ji, 2025). This paradigm has proven particularly effective for question answering, yet its reliability degrades sharply as tasks require multi-hop reasoning across multiple documents (Zhu et al., 2024; Yu et al., 2025; Wu et al., 2025; Xiang et al., 2026). In such settings, solving a query demands not only identifying relevant information, but also composing intermediate steps into a coherent reasoning structure (Press et al., 2023; Biran et al., 2024). This makes the choice of in-context demonstrations one of the critical variables: seemingly plausible examples can lead to sharply different outcomes when they fail to match the reasoning demands of the query (Bhope et al., 2025; Schoch and Ji, 2025). Despite substantial progress, it remains unclear what makes in-context demonstrations effective for tasks that require structured, multi-step reasoning.

This has motivated a growing body of work on demonstration retrieval, which seeks to improve in-context learning by selecting the most useful examples from a candidate pool for a target query (Liu et al., 2022; Luo et al., 2024b). Existing retrieval-based approaches have explored a range of selection signals, including semantic similarity, diversity, coverage, and model preference (Wang et al.,

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 4035–4056
November 6-10, 2026 ©2026 Association for Computational Linguistics

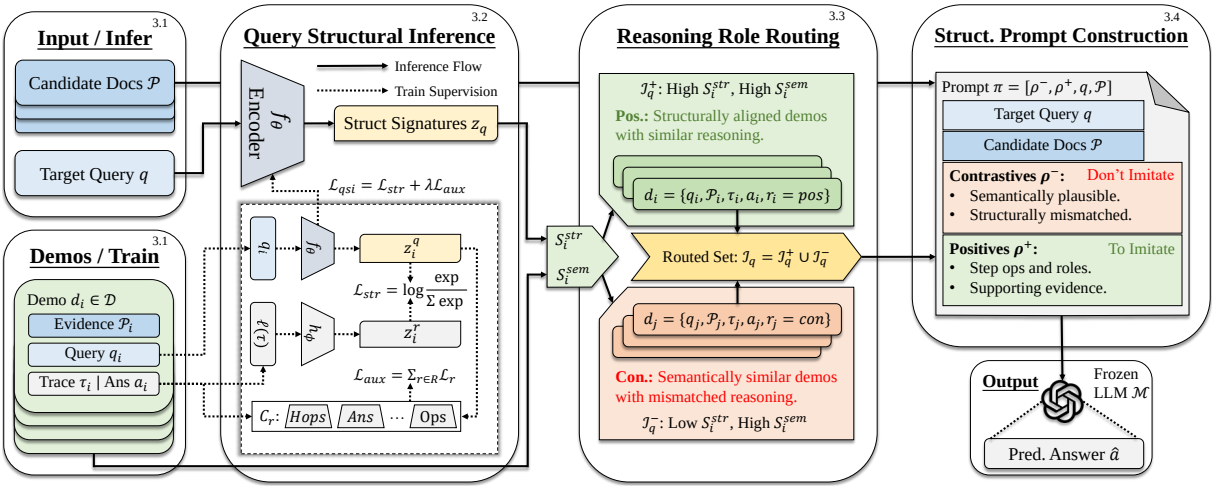


Figure 2: Overview of STRIDE. Query Structure Inference learns a query-side structural signature from decomposition traces during training and infers z_q from the query alone at test time. Reasoning Role Routing assigns candidate demonstrations into positive and contrastive roles according to structural compatibility and semantic plausibility. Structural Prompt Construction converts the routed demonstrations into role-aware prompts for a frozen language model.

2024b,a; Kweon et al., 2025; Zhang et al., 2025). Among these, semantic matching remains a common and influential paradigm, often implemented by encoding the query and candidate demonstrations with sentence or dense retrievers and ranking them by embedding similarity (Purohit et al., 2025; Wang et al., 2026). This line of work rests on the intuitive premise that examples closer to the target query should provide the most useful guidance. Yet for tasks that require structured, multi-step reasoning, semantic relatedness is often an incomplete signal (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2021). As illustrated in Figure 1, questions that appear similar on the surface may still differ in reasoning depth, intermediate inference steps, or patterns of evidence composition. As a result, seemingly relevant demonstrations can provide the wrong reasoning signal and may even impair downstream performance. The challenge, therefore, is not simply to retrieve relevant examples, but to retrieve demonstrations whose reasoning structure is compatible with the demands of the query.

Building on this observation, we argue that effective demonstrations for multi-hop question answering should match the latent reasoning structure of the target query, not merely its surface semantics. To pursue this idea, we propose **STRIDE**, a structure-aware framework for in-context demonstration extraction. As illustrated in Figure 2, it first performs *query structure inference*, where a trainable query encoder learns to infer latent structural signatures from questions alone by aligning them with normalized decomposition traces available during training. These traces provide a data-driven view of the reasoning process while abstracting away surface entities. Auxiliary probes such as hop count, answer type, and coarse operation patterns further stabilize training and make the learned structural space interpretable.

STRIDE then performs *reasoning role routing*. Instead of returning an undifferentiated top-k list, it assigns candidate demonstrations into instructional roles. Structurally compatible examples are selected as positive demonstrations, while semantically plausible but structurally mismatched examples are selected as contrastive demonstrations. Finally, *structural prompt construction* converts the routed demonstrations into compact role traces: positive examples expose reasoning structures to imitate, whereas contrastive examples reveal near-miss patterns that should not be copied as solution templates. In this way, STRIDE turns demonstration retrieval into structure-aware problem-solving guidance. Our contributions are three-fold:

- We introduce a structural compatibility criterion for selecting demonstrations for MHQA.
- We propose STRIDE, which learns query-side structural signatures from decomposition traces and selects role-routed demonstrations.
- We show that structural alignment, contrastive routing, and structural prompting improve MHQA performance over baselines.

2 Related Works

2.1 Demonstration Selection for ICL

ICL is highly sensitive to demonstration selections, motivating retrieval-based methods that choose examples semantically close to the input (Liu et al., 2022). Later work learns retrievers from LM feedback, such as EPR (Rubin et al., 2022), UDR (Li et al., 2023), and recent feedback-adaptive retrievers (Wang et al., 2024b,a). Another line formulates selection as subset construction, balancing relevance, diversity, and interaction among demonstrations through DPPs, reasoning-augmented reranking, topic coverage, correlation structure, or LLM preference (Ye et al., 2023; Xiong et al., 2024; Kweon et al., 2025; Du et al., 2025; Zhang et al., 2025). STRIDE moves beyond surface similarity by selecting demonstrations whose latent reasoning structure matches the target query.

2.2 Structure-Aware Retrieval and Reasoning

Explicit structure can improve complex reasoning by modeling intermediate steps, dependencies, or paths rather than flat text sequences. Tree-of-Thoughts and Graph-of-Thoughts organize reasoning as search over structured thought states (Yao et al., 2023; Besta et al., 2024), while IRCoT interleaves partial reasoning with retrieval (Trivedi et al., 2023). Other methods decompose questions into reasoning trees, ground reasoning in knowledge-graph paths, or rerank demonstrations using reasoning graphs (Zhang et al., 2024a; Luo et al., 2024a; Lin et al., 2025). Although the value of structure for reasoning has been confirmed, STRIDE differs by learning latent structural signatures for demonstration selection.

2.3 Contrastive and Negative Demonstrations

Negative or contrastive examples can be used to improve ICL by exposing models to errors or failures. Contrastive CoT provides valid and invalid rationales to guide step-by-step reasoning (Chia et al., 2023), while learning-from-mistakes methods use incorrect solutions or feedback to improve reasoning (An et al., 2024). This is also prevalent in task-specific ICL such as event causality, information extraction, and few-shot selection (Chao et al., 2024; Mo et al., 2024; Liang et al., 2025; Gu et al., 2025). Instead of wrong or failed instances, STRIDE instead derives contrastive demonstrations from semantic near-misses that are structurally mismatched with the target query.

Symbol	Definition
$q, \mathcal{P}$	Target question, candidate paragraph set.
$a, \hat{a}$	Gold, predicted answers.
$\tau, \ell(\tau)$	trace, normalized trace.
$\mathcal{D}$	Demonstration pool.
$\mathcal{I}_q; \mathcal{I}_q^+, \mathcal{I}_q^-$	Routed demo set; positive, contrastive demos.
f_θ, h_ϕ	Query encoder, trace encoder.
$z_q; z_i^q, z_i^\tau$	Structural signature; query, trace embeddings.
$s_i^{\text{str}}, s_i^{\text{sem}}$	Structural compatibility, semantic similarity.
r_i	Routing label, pos or con.
ρ_i^+, ρ_i^-	Positive, contrastive role traces.
π	Final structure-aware prompt.
$\mathcal{M}$	Frozen language model.

Table 1: Summary of the main notation defined.

3 Method

3.1 Task and Problem Formulation

We consider in-context learning for multi-hop question answering. Each instance is denoted as $x = (q, \mathcal{P}, a)$, where q is the question, $\mathcal{P} = \{p_1, \ldots, p_m\}$ is the set of candidate paragraphs, and a is the gold answer. For training instances, we additionally assume access to a decomposition trace $\tau = \{s_1, \ldots, s_T\}$, where each step s_t represents an intermediate sub-question or operation required to derive the final answer.

Let $\mathcal{D} = \{d_i\}_{i=1}^N$ be a demonstration pool, where each element is $d_i = (q_i, \mathcal{P}_i, a_i, \tau_i)$. Given a target instance $(q, \mathcal{P})$, the goal is to select a small subset of demonstrations $\mathcal{I}_q \subset \mathcal{D}$ that can guide the model toward the correct reasoning process, and then construct a prompt π for a frozen language model $\mathcal{M}$ to generate an answer $\hat{a}$:

$$\hat{a} = \mathcal{M}(\pi), \quad \pi = \Pi(q, \mathcal{P}, \mathcal{I}_q), \quad |\mathcal{I}_q| = k. \quad (1)$$

The main objective is to choose $\mathcal{I}_q$ such that the selected demonstrations are useful for solving q, or more formally, have high utility. In STRIDE, demonstration utility is modeled not only by semantic similarity, but also by structural compatibility. It therefore defines a latent structural signature z_q for each query, derived from its decomposition trace τ during training and inferred from the query alone at test time. This signature represents the problem-solving structure required by q. The remainder of this section describes how STRIDE learns f_θ, uses z_q to route demonstrations, and constructs a structural prompt π. Figure 2 illustrates the overall architecture of STRIDE and Table 1 summarizes the main notation used throughout this section.

3.2 Query Structure Inference

STRIDE first infers the latent problem-solving structure of a query rather than treating it as a semantic retrieval key. It obtains structural supervision from gold decomposition traces when available, or from pseudo-traces constructed from dataset-provided evidence annotations otherwise.

Each trace $\tau_i = \{s_1, \ldots, s_T\}$ is converted into a normalized structural trace $\ell(\tau_i)$, where each step $s_t = (u_t, y_t)$ contains a sub-question or operation u_t and an intermediate answer y_t. Surface entities are abstracted into placeholders, while step order, cross-step references, and operation-indicative phrases are preserved:

$$\ell(\tau_i) = [\mathrm{Norm}(s_1); \ldots; \mathrm{Norm}(s_T); \mathcal{E}_i], \quad (2)$$

where $\mathcal{E}_i$ denotes dependencies between steps. Specifically, entity mentions are replaced with indexed placeholders such as $\langle \mathrm{ENT}_j \rangle$, references to previous steps with $\langle \mathrm{REF}_t \rangle$, and intermediate answers with coarse type markers such as $\langle \mathrm{DATE} \rangle$, $\langle \mathrm{NUM} \rangle$, and $\langle \mathrm{LOC} \rangle$.

The normalized trace and the query are mapped into a shared structural space:

$$z_i^q = \frac{W_q f_\theta(q_i)}{\|W_q f_\theta(q_i)\|_2}, \quad z_i^\tau = \frac{W_\tau h_\phi(\ell(\tau_i))}{\|W_\tau h_\phi(\ell(\tau_i))\|_2}, \quad (3)$$

where f_θ is the query encoder, h_ϕ is the trace encoder, and W_q, W_τ are projection layers. The two encoders are trained to align each query q_i with its corresponding normalized decomposition trace $\ell(\tau_i)$ using an in-batch contrastive loss (Chernyavskiy et al., 2022):

$$\mathcal{L}_{\mathrm{str}} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log \frac{\exp(\mathrm{sim}(z_i^q, z_i^\tau)/\tau_c)}{\sum_{j \in \mathcal{B}} \exp(\mathrm{sim}(z_i^q, z_j^\tau)/\tau_c)}. \quad (4)$$

Lightweight auxiliary probes are attached to the query-side representation retained at inference. Let $\mathcal{R}$ denote decomposition-derived properties. For each probe $r \in \mathcal{R}$, a head C_r predicts $\hat{y}_i^r = C_r(\bar{z}_i^q)$ from the pre-normalized query representation $\bar{z}_i^q = W_q f_\theta(q_i)$. These probes stabilize training and improve interpretability without defining the structural space:

$$\mathcal{L}_{\mathrm{QSI}} = \mathcal{L}_{\mathrm{str}} + \frac{\lambda_{\mathrm{aux}}}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{r \in \mathcal{R}} \mathcal{L}_r\big(C_r(\bar{z}_i^q), y_i^r\big), \quad (5)$$

where y_i^r is the trace-derived label. At inference time, only the query encoder is retained to produce $z_q = W_q f_\theta(q)/\|W_q f_\theta(q)\|_2$.

3.3 Reasoning Role Routing

The structural signature z_q enables STRIDE to route candidates into instructional roles rather than returning an undifferentiated top-k list.

Each candidate demonstration d_i is structurally represented by applying the query encoder to its question q_i, yielding z_i. A frozen semantic encoder m_ω is used only to estimate surface relevance:

$$s_i^{\mathrm{str}} = \mathrm{sim}(z_q, z_i), \quad s_i^{\mathrm{sem}} = \mathrm{sim}(m_\omega(q), m_\omega(q_i)), \quad (6)$$

where s_i^{str} measures structural compatibility and s_i^{sem} measures semantic similarity. Given these scores, STRIDE first selects positive demonstrations that instantiate problem-solving structures compatible with the target query:

$$\mathcal{I}_q^+ = \mathrm{TopK}_{k_+}\left[\alpha s_i^{\mathrm{str}} + (1-\alpha) s_i^{\mathrm{sem}}\right]_{d_i \in \mathcal{D}}, \quad (7)$$

where α balances structural and semantic matching. This prevents routing from relying exclusively on either latent structure or topical relevance.

To select contrastive demonstrations, STRIDE forms a semantic neighborhood

$$\mathcal{C}_q = \mathrm{TopK}_M[s_i^{\mathrm{sem}}]_{d_i \in \mathcal{D}} \setminus \mathcal{I}_q^+. \quad (8)$$

It then chooses semantically plausible but structurally mismatched near-misses:

$$\mathcal{I}_q^- = \mathrm{TopK}_{k_-}\left[s_i^{\mathrm{sem}} - \gamma s_i^{\mathrm{str}}\right]_{d_i \in \mathcal{C}_q}, \quad (9)$$

where γ penalizes structural compatibility. In this way, contrastive demonstrations remain relevant to the target query while being unsuitable as reasoning templates.

The routed in-context demonstration set is $\mathcal{I}_q = \mathcal{I}_q^+ \cup \mathcal{I}_q^-$, where each selected demonstration carries a routing label $r_i \in \{\mathrm{pos}, \mathrm{con}\}$. Positive demonstrations provide structures to imitate, whereas contrastive demonstrations identify semantically plausible but structurally incompatible examples that should not be copied as solution templates.

3.4 Structural Prompt Construction

The final component converts the routed demonstrations into a prompt that makes their instructional roles explicit, moving beyond plain question–answer exemplars toward structure-aware reasoning guidance.

Inspired by recent work on decomposition-based reasoning, evidence-grounded reasoning chains, and contrastive demonstrations (Huang et al., 2024;

Categories	Methods	MuSiQue		HotpotQA		2WikiQA		Rank
		EM	F1	EM	F1	EM	F1	Δ
Control	Vanilla-QA	9.62	14.71	45.00	55.80	31.50	37.80	16
	Random-Demo	17.47	24.12	48.20	58.60	33.80	39.60	14
	Facets-Match	22.16	30.19	53.10	63.10	41.20	47.80	6
Semantic	BM25 (Robertson and Zaragoza, 2009)	20.47	28.81	52.36	63.27	47.83	51.46	4
	Contriever (Izacard et al., 2021)	24.03	31.35	51.18	62.54	38.27	42.13	6
	ColBERT-V2 (Santhanam et al., 2021)	15.60	26.42	43.72	57.43	33.18	43.56	14
	DeBERTa-V3-Large (He et al., 2023)	15.69	26.48	46.00	58.20	34.80	42.20	12
RAG	Vanilla RAG (Lewis et al., 2020)	15.02	20.66	42.31	56.17	45.26	51.38	10
	RAPTOR (Sarthi et al., 2024)	14.05	23.90	28.42	42.65	29.31	39.48	16
	HippoRAG (Gutiérrez et al., 2024)	<u>29.04</u>	<u>38.73</u>	<u>57.24</u>	<u>72.08</u>	<u>63.35</u>	<u>72.41</u>	2
ICL	Vanilla CoT (Wei et al., 2022)	18.61	24.35	50.65	57.59	34.56	40.73	11
	CoT-SC (Wang et al., 2022)	21.48	28.02	56.46	64.05	36.30	41.07	7
	Complex-CoT (Fu et al., 2023)	19.86	26.75	53.42	61.36	35.74	42.18	9
	Auto-CoT (Zhang et al., 2024b)	20.69	27.31	54.76	62.48	36.92	43.05	8
	RGER (Lin et al., 2025)	24.23	32.58	58.14	66.72	39.86	46.31	3
	STRIDE$_{qwen3\text{-}8b}$ (ours)	**33.85**	**41.63**	**61.50**	**73.20**	**65.20**	**73.60**	**1**
	w/o Query Structure Inference	28.68	36.44	56.87	67.59	48.97	56.35	−9.02
	w/o Reasoning Role Routing	31.26	39.23	59.49	70.66	62.23	70.41	−2.62
	w/o Structural Prompt Construction	33.29	40.57	60.86	72.14	64.06	72.03	−1.01

Table 2: Main and ablation results. Rank is computed by average performance over the six dataset metrics, excluding ablation rows; Δ denotes the average drop from full STRIDE. Best and second-best scores are bolded and underlined, respectively, excluding ablation rows.

Wu et al., 2024; Chia et al., 2023), STRIDE represents each selected demonstration as a compact role trace rather than a plain question–answer pair. For a positive demonstration $d_i \in \mathcal{I}_q^+$, the trace exposes the ordered reasoning steps, the supporting evidence used by each step, and the final answer:

$$\rho_i^+ = \mathrm{Trace}(q_i, \mathcal{P}_i, \tau_i, a_i, r_i = \mathrm{pos}). \quad (10)$$

This format encourages the model to imitate demonstrations whose problem-solving structure is compatible with the target query.

For a contrastive demonstration $d_i \in \mathcal{I}_q^-$, the trace is rendered differently:

$$\rho_i^- = \mathrm{Trace}(q_i, \tau_i, r_i = \mathrm{con}, \Delta_i), \quad (11)$$

where Δ_i summarizes the structural mismatch between d_i and the target query. In practice, Δ_i is instantiated from routing signals and auxiliary structural probes, such as differences in hop count, answer type, or operation pattern. Thus, contrastive demonstrations are not treated as noise; instead, they serve as near-miss examples that should be recognized as structurally incompatible rather than copied as solution templates.

Given the positive and contrastive traces, the final prompt is constructed as

$$\begin{aligned} \pi &= \Pi(q, \mathcal{P}, \mathcal{I}_q) \\ &= [\rho_1^-; \ldots; \rho_{k_-}^-; \rho_1^+; \ldots; \rho_{k_+}^+; q; \mathcal{P}] \end{aligned} \quad (12)$$

where the target question and its candidate paragraphs are placed after the routed demonstrations. Because LLMs can be sensitive to where information appears in the prompt (Liu et al., 2023), STRIDE places contrastive traces before positive traces, leaving the final demonstration block to contain structures intended for imitation. A simple theoretical view of why structural compatibility can improve demonstration utility is presented in Appendix D.

Dataset	Trace Source	Labels
MuSiQue	Gold traces available	Hop, answer type, operation, support
HotpotQA	Supporting facts, question type	reasoning type, answer type, evidence role
2WikiQA	Evidence chains, reasoning path	Hop, reasoning type, answer type, support

Table 3: Trace supervision used for Query Structure Inference. MuSiQue provides gold traces; HotpotQA and 2WikiQA are converted into pseudo-traces from dataset-provided evidence annotations.

Method	Hops	Ans.	Op. F1	Avg.	Struct.
Random	45.69	58.63	49.28	51.16	0.249
Semantic	53.35	66.60	62.88	60.93	0.458
STRIDE	**79.27**	**80.62**	**78.51**	**79.45**	**0.915**

Table 4: Structural alignment of retrieved demonstrations on MuSiQue. Hops, Ans., and Op. F1 are external trace-label metrics computed from hop count, answer type, and operation-pattern labels, and are not based on the learned structural embedding. Avg. averages these three non-embedding metrics. Struct. is reported only as a learned-space diagnostic.

4 Experiments

4.1 Experimental Settings

Experiments are conducted on MuSiQue (Trivedi et al., 2021), HotpotQA (Yang et al., 2018), and 2WikiQA (Ho et al., 2020). For each dataset, demonstrations are drawn from the training split while the development split containing the target queries is used for inference. More information are available in Appendix A. Because STRIDE learns structural signatures from trace supervision, datasets are converted into trace-like forms according to their available annotations. Table 3 summarizes the supervision source used for each benchmark.

Baselines are grouped by input protocol. Controls test direct answering, random demonstrations, and discrete facet matching. Semantic retrievers and ICL baselines use the same target context and demonstration budget as STRIDE, but differ in how demonstrations are selected or prompted. RAG baselines retrieve external evidence passages rather than in-context demonstrations, and are included to compare STRIDE with evidence-oriented retrieval systems. STRIDE variants ablate QSI, RRR, and SPC by replacing structural signatures with semantic signals, removing positive-contrastive routing, or using plain demonstrations. More details are in Appendix B.

Evaluation uses EM and token-level F1. All methods use the same number of demonstrations and the same frozen generator. STRIDE initializes its query and trace encoders from DeBERTa-v3-Large (He et al., 2023) then train according to Section 3.2 for 10 epochs. Backbone uses Qwen3-8B with temperature zero, with Llama-3.1-8B (Dubey et al., 2024) and Ministral-8B (Liu et al., 2026) used for robustness analyses. Implementation Details are in Appendix C.

4.2 Main and Ablation Analysis

Table 2 reports the main comparison and module ablations. STRIDE achieves the best performance on all six metrics and ranks first overall. Compared with the strongest non-STRIDE baseline, HippoRAG, STRIDE improves average performance by 2.69 points, indicating that structure-aware demonstration extraction provides benefits beyond conventional retrieval and prompting.

Semantic retrievers remain consistently below STRIDE, showing that surface relevance alone is insufficient for selecting useful demonstrations in multi-hop QA. The ablations further confirm the contribution of each module. Removing Query Structure Inference causes the largest drop (-9.02), highlighting the importance of learned structural signatures. Removing Reasoning Role Routing reduces performance by -2.62, showing the value of positive–contrastive demonstration routing. Removing Structural Prompt Construction leads to a smaller but consistent drop (-1.01), suggesting that role-aware prompt realization helps the model better use selected demonstrations.

The relative behavior of the baselines also clarifies where STRIDE gains come from. RAG methods such as HippoRAG remain strong because they improve evidence access, while ICL methods improve reasoning by changing the prompting strategy. STRIDE is complementary to these directions: it focuses on selecting demonstrations that better match the target query's reasoning structure. This explains why the largest gains appear when structural inference and role routing are both retained, and why semantic-only or prompt-only variants cannot fully recover the performance of the full framework.

Demo	Sem. Sim.	Struct. Sim.	Hops	Op. F1
$\mathcal{I}_q^+$	0.461	0.878	77.30	79.00
$\mathcal{I}_q^-$	0.470	-0.140	28.80	32.45

Table 5: Role separation in the RRR module. Sem. Sim. and Struct. Sim. are average cosine similarities to the target query; Hops and Op. F1 measure hop-count agreement and operation-pattern overlap.

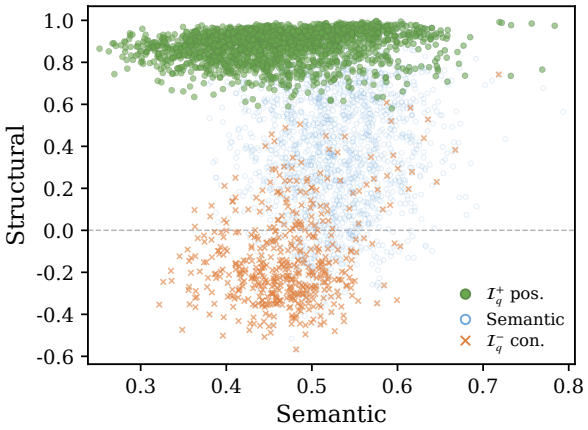


Figure 3: Semantic vs. structural similarity distribution of retrieved demonstrations on MuSiQue.

4.3 Further Analysis

4.3.1 Retrieval Quality and Structural Alignment

To test whether STRIDE improves demonstration selection before generation, we evaluate retrieval quality on MuSiQue using structure-oriented metrics: hop-count match, answer-type match, operation-pattern F1, and cosine similarity in the learned structural space. Table 4 shows that STRIDE improves not only learned-space similarity, but also external trace-label metrics that are independent of the structural encoder used for retrieval. In particular, STRIDE improves the external alignment average from 60.93 to 79.45 over semantic retrieval.

Table 5 and Figure 3 further validate Reasoning Role Routing. Positive and contrastive demonstrations have comparable semantic similarity, but sharply different structural compatibility. Thus, $\mathcal{I}_q^+$ provides structures to imitate, while $\mathcal{I}_q^-$ serves as semantic near-misses that expose the gap between surface relevance and reasoning-structure compatibility. This separation is important because contrastive examples should remain plausible enough to be informative, but structurally different enough to discourage shallow imitation. The cleanly separated results confirm the distinction.

# Demos			alpha			gamma		
k	EM	F1	α	EM	F1	γ	EM	F1
0	9.62	14.71	0.00	28.63	36.44	0.00	31.95	39.71
2	31.11	38.62	**0.50**	**33.85**	**41.63**	**1.00**	**33.85**	**41.63**
4	**33.85**	**41.63**	0.75	33.31	41.04	1.50	33.26	40.99
8	32.43	40.25	1.00	32.79	40.52	2.00	32.65	40.28

Table 6: Sensitivity analysis of STRIDE on MuSiQue. The best setting is $k = 4$, $\alpha = 0.5$, and $\gamma = 1.0$.

4.3.2 Sensitivity to Key Hyperparameters

We further analyze the sensitivity of STRIDE to three key hyperparameters: the number of in-context demonstrations k, the structural–semantic balance coefficient α, and the contrastive penalty coefficient γ. All experiments are conducted on MuSiQue with Qwen3-8B.

Table 6 shows that STRIDE is robust to key hyperparameters. Performance peaks at $k = 4$, $\alpha = 0.5$, and $\gamma = 1.0$, indicating that the best setting balances prompt length, structural compatibility, semantic relevance, and contrastive strength. The slight drop at larger k or extreme α, γ values suggests that excessive demonstrations may reduce the clarity and utility of in-context guidance.

Backbone	Semantic		STRIDE		Gain
	EM	F1	EM	F1	F1
Qwen3-8B	28.60	36.44	33.85	41.63	**+5.19**
Llama-3.1-8B	27.24	35.16	32.19	39.73	**+4.57**
Ministral-8B	25.85	33.64	30.56	38.17	**+4.53**

Table 7: Backbone robustness on MuSiQue.

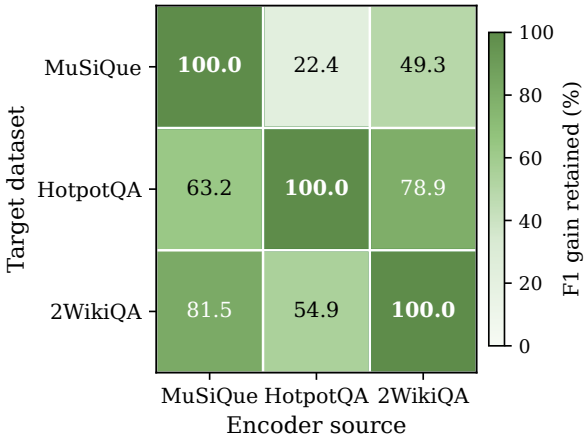


Figure 4: Cross-dataset transfer of the QSI encoder.

4.3.3 Backbone and Transfer Robustness

We further examine whether STRIDE is robust to the choice of frozen generator and whether the learned structural encoder transfers across datasets. For backbone robustness, the retrieval pipeline, prompt format, and demonstration budget are fixed, while only the generator is changed. Table 7 shows that STRIDE consistently outperforms semantic demonstration retrieval across three local 8B-scale backbones, suggesting that the gains do not depend on a specific generator.

To assess transferability, we train the QSI structural encoder on one dataset and apply it to another while keeping the target demonstration pool, generator, prompt format, and routing hyperparameters fixed. We report transfer retention:

$$\text{Retention} = \frac{\text{F1}_{\text{transfer}} - \text{F1}_{\text{semantic}}}{\text{F1}_{\text{target}} - \text{F1}_{\text{semantic}}} \times 100. \tag{13}$$

As shown in Figure 4, off-diagonal retention remains above semantic retrieval in most settings, indicating that the learned structural space captures reusable multi-hop reasoning patterns rather than purely dataset-specific shortcuts. Transfer is strongest between datasets with similar evidence-chain structures, while target-trained encoders still provide the best alignment.

4.3.4 Case Study and Error Analysis

Figure 5 shows a representative case. Semantic retrieval selects a demonstration with strong topical overlap but a mismatched reasoning structure, yielding the wrong answer. STRIDE instead retrieves a positive demonstration with a compatible four-step dependency pattern and a contrastive demonstration that is semantically plausible but structurally mismatched. This illustrates how role routing separates surface relevance from reasoning-structure compatibility and converts retrieved demonstrations into more useful guidance for the frozen generator.

We analyze 100 sampled false STRIDE predictions on MuSiQue. Errors mainly involve intermediate-entity resolution (39.0%), mixed or unclear causes (25.0%), answer normalization (14.0%), composition (13.0%), evidence grounding (8.0%), and prompt/format issues (1.0%). The dominance of intermediate-entity errors suggests that remaining failures often stem from incorrect bridge-step grounding by the frozen generator, highlighting evidence grounding, step verification, and answer calibration as future directions.

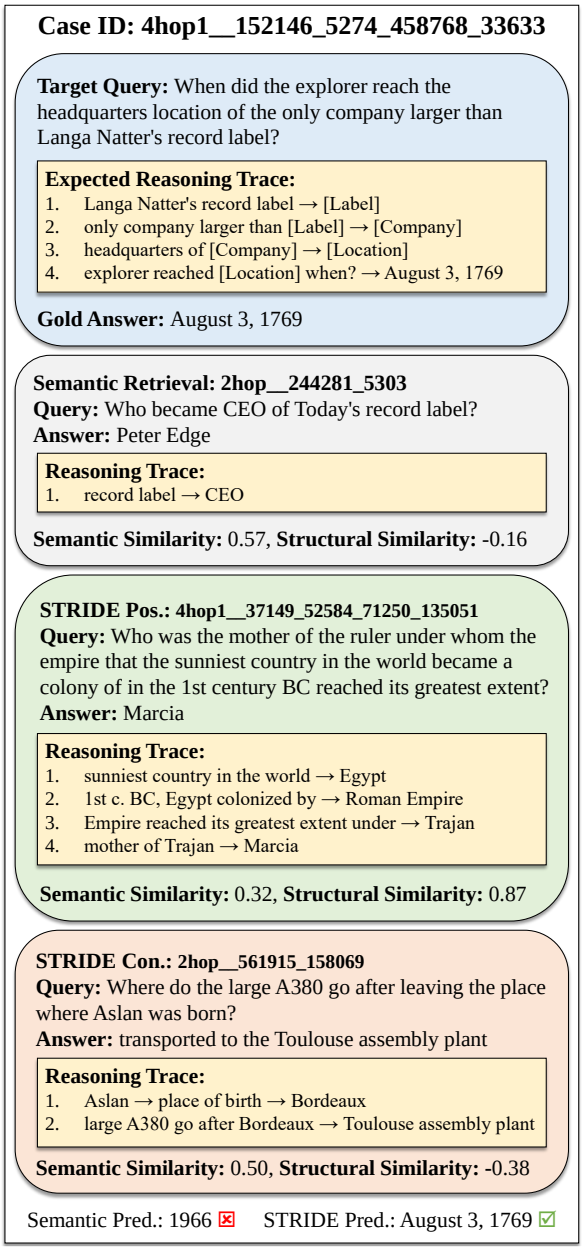


Figure 5: Case study illustrating semantic retrieval versus STRIDE routing.

5 Conclusion

We presented STRIDE, a structure-aware framework for in-context demonstration extraction in multi-hop question answering. STRIDE learns latent query structures from decomposition traces, routes demonstrations into positive and contrastive roles, and constructs role-aware prompts for frozen language models. Experiments across three multi-hop QA benchmarks show consistent gains over baselines. These results suggest that effective demonstrations should be selected not only by surface relevance, but also by compatibility with the target query's underlying reasoning structure.

Limitations

STRIDE has several limitations. First, its structural encoder relies on decomposition traces or trace-like supervision during training, which may not be available for all QA datasets. Although the learned encoder is used from the query alone at inference time, extending STRIDE to domains without explicit decomposition annotations may require automatic trace induction or weak supervision. Second, contrastive demonstrations improve role-aware prompting but also increase prompt length, making the method sensitive to context budget and prompt ordering. Third, STRIDE improves demonstration selection rather than the reasoning ability of the frozen generator itself; remaining errors still arise from incorrect evidence grounding, intermediate-entity resolution, and answer normalization. Finally, the current experiments focus on English multi-hop QA benchmarks, and broader evaluation on other languages, domains, and reasoning-intensive tasks is needed to test generality. Potential risks include propagating biases or annotation artifacts from the demonstration pool and producing overconfident answers when retrieved demonstrations are structurally misleading or when the frozen generator grounds intermediate steps incorrectly.

Acknowledgments

We would like to express our sincere gratitude to the anonymous reviewers for their thoughtful comments, insightful feedback, and thorough evaluation of our work. Their constructive suggestions were instrumental in refining the presentation and technical depth of this paper. This work was supported in part by the Major Key Project of PCL (Grant No. 2025A09) and the National Natural Science Foundation of China (Grant No. 62576120).

References

Shengnan An, Zexiong Ma, Siqi Cai, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2024. Can LLMs learn from mistakes? an empirical study on reasoning tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 833–854, Miami, Florida, USA. Association for Computational Linguistics.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michał Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: solving elaborate problems with large language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Rahul Atul Bhope, Praveen Venkateswaran, K. R. Jayaram, Vatche Isahagian, Vinod Muthusamy, and Nalini Venkatasubramanian. 2025. OptiSeq: Ordering examples on-the-fly for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24864–24887, Suzhou, China. Association for Computational Linguistics.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. 2024. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *Conference on Empirical Methods in Natural Language Processing*.

Liang Chao, Wei Xiang, and Bang Wang. 2024. In-context contrastive learning for event causality identification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 868–881, Miami, Florida, USA. Association for Computational Linguistics.

Jiuhai Chen, Lichang Chen, Chen Zhu, and Tianyi Zhou. 2023. How many demonstrations do you need for in-context learning? In *Conference on Empirical Methods in Natural Language Processing*.

Anton Chernyavskiy, Dmitry Ilvovsky, Pavel Kalinin, and Preslav Nakov. 2022. Batch-softmax contrastive loss for pairwise sentence scoring tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 116–126, Seattle, United States. Association for Computational Linguistics.

Yew Ken Chia, Guizhen Chen, Anh Tuan Luu, Soujanya Poria, and Lidong Bing. 2023. Contrastive chain-of-thought prompting. *ArXiv*, abs/2311.09277.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2022. A survey on in-context learning. In *Conference on Empirical Methods in Natural Language Processing*.

Qiunan Du, Zhiliang Tian, Zhen Huang, Kailun Bian, Tianlun Liu, Zhaoning Zhang, Xinwang Liu, Feng Liu, and Dongsheng Li. 2025. Correlation-aware example selection for in-context learning with nonsymmetric determinantal point processes. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27066–27082, Suzhou, China. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela

Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Yao Fu, Hao-Chun Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. *ArXiv*, abs/2210.00720.

Jiawei Gu, Ziting Xian, Yuanzhen Xie, Ye Liu, Enjie Liu, Ruichao Zhong, Mochi Gao, Yunzhi Tan, Bo Hu, and Zang Li. 2025. Toward structured knowledge reasoning: Contrastive retrieval-augmented generation on experience. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23891–23910, Vienna, Austria. Association for Computational Linguistics.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: neurobiologically inspired long-term memory for large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Guangming Huang, Yunfei Long, Cunjin Luo, Jiaxing Shen, and Xia Sun. 2024. Prompting explicit and implicit knowledge for multi-hop question answering based on human reading process. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13179–13189, Torino, Italia. ELRA and ICCL.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Wonbin Kweon, SeongKu Kang, Runchu Tian, Pengcheng Jiang, Jiawei Han, and Hwanjo Yu. 2025. Topic coverage-based demonstration retrieval for in-context learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19900–19912, Suzhou, China. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668, Toronto, Canada. Association for Computational Linguistics.

Yunhao Liang, Ruixuan Ying, Takuya Taniguchi, and Zhe Cui. 2025. Failures are the stepping stones to success: Enhancing few-shot in-context learning by leveraging negative samples. *ArXiv*, abs/2507.23211.

Yukang Lin, Bingchen Zhong, Shuoran Jiang, Joanna Siebert, and Qingcai Chen. 2025. Reasoning graph enhanced exemplars retrieval for in-context learning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9737–9759, Abu Dhabi, UAE. Association for Computational Linguistics.

Alexander Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sad'e, Alan Jeffares, Albert Q. Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Am'elie H'eliou, Amos You, Andy Ehrenberg, Andy Desman Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 100 others. 2026. Ministral 3. *ArXiv*, abs/2601.08584.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

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

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. 2024b. In-context learning with retrieved demonstrations for language models: A

survey. *Transactions on Machine Learning Research*. Survey Certification.

Haitao Mao, Guang-Da Liu, Yao Ma, Rongrong Wang, Kristen Marie Johnson, and Jiliang Tang. 2024. A survey to recent progress towards understanding in-context learning. In *North American Chapter of the Association for Computational Linguistics*.

Ying Mo, Jiahao Liu, Jian Yang, Qifan Wang, Shun Zhang, Jingang Wang, and Zhoujun Li. 2024. C-ICL: Contrastive in-context learning for information extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10099–10114, Miami, Florida, USA. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Kiran Purohit, Venktesh V, Sourangshu Bhattacharya, and Avishek Anand. 2025. Sample efficient demonstration selection for in-context learning. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Keshav Santhanam, O. Khattab, Jon Saad-Falcon, Christopher Potts, and Matei A. Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *North American Chapter of the Association for Computational Linguistics*.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

S. Schoch and Yangfeng Ji. 2025. The good, the bad, and the debatable: A survey on the impacts of data for in-context learning. In *Conference on Empirical Methods in Natural Language Processing*.

H. Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2021. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Huazheng Wang, Jinming Wu, Haifeng Sun, Zixuan Xia, Daixuan Cheng, Jingyu Wang, Qi Qi, and Jianxin Liao. 2024a. MDR: Model-specific demonstration retrieval at inference time for in-context learning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4189–4204, Mexico City, Mexico. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2024b. Learning to retrieve in-context examples for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1752–1767, St. Julian's, Malta. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Tenth International Conference on Learning Representations*.

Zihua Wang, Jiarui Wang, Haiyang Xu, Mingshi Yan, Fei Huang, Xu Yang, Xiu-Shen Wei, Siya Mi, and Yu Zhang. 2026. Efficient and effective in-context demonstration selection with coreset. In *AAAI Conference on Artificial Intelligence*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS'22, Red Hook, NY, USA. Curran Associates Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jian Wu, Linyi Yang, Yuliang Ji, Wenhao Huang, Börje F. Karlsson, and Manabu Okumura. 2024. Gendec: A robust generative question-decomposition method for multi-hop reasoning. *ArXiv*, abs/2402.11166.

Jian Wu, Linyi Yang, Dongyuan Li, Yuliang Ji, Manabu Okumura, and Yue Zhang. 2025. Mmqa: Evaluating llms with multi-table multi-hop complex questions. In *International Conference on Learning Representations*.

Biao Xiang, Soyeon Caren Han, and Yihao Ding. 2026. Bridge: Benchmark for multi-hop reasoning in long multimodal documents with grounded evidence. *ArXiv*.

Jing Xiong, Zixuan Li, Chuanyang Zheng, Zhijiang Guo, Yichun Yin, Enze Xie, Zhicheng Yang, Qingxing Cao, Haiming Wang, Xiongwei Han, Jing Tang, Chengming Li, and Xiaodan Liang. 2024. Dq-lore: Dual queries with low rank approximation re-ranking for in-context learning. In *The Twelfth International Conference on Learning Representations*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Sangwon Yu, Ik-hwan Kim, Jongyoon Song, Saehyung Lee, Junsung Park, and Sungroh Yoon. 2025. Unleashing multi-hop reasoning potential in large language models through repetition of misordered context. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6450–6470, Albuquerque, New Mexico. Association for Computational Linguistics.

Kun Zhang, Jiali Zeng, Fandong Meng, Yuanzhuo Wang, Shiqi Sun, Long Bai, Huawei Shen, and Jie Zhou. 2024a. Tree-of-reasoning question decomposition for complex question answering with large language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Zheng Zhang, Shaocheng Lan, Lei Song, Jiang Bian, Yexin Li, and Kan Ren. 2025. Learning to select in-context demonstration preferred by large language model. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11345–11360, Vienna, Austria. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alexander J. Smola. 2024b. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. Fanoutqa: A multi-hop, multi-document question answering benchmark for large language models. In *Annual Meeting of the Association for Computational Linguistics*.

A Further Information on Datasets

Table 8 summarizes the metadata of the three multi-hop QA benchmarks used in the experiments. These datasets differ in how questions are constructed and how evidence is annotated, which makes them suitable for evaluating whether STRIDE can exploit structural supervision beyond surface semantic matching.

MuSiQue. MuSiQue is used as the main benchmark because it directly exposes the kind of decomposition structure targeted by STRIDE. Its answerable subset contains 24,814 questions split into 19,938 training, 2,417 development, and 2,459 test examples, with questions distributed across 2-, 3-, and 4-hop settings. Each instance contains 20 paragraphs, including gold supporting paragraphs and hard distractors, and provides question decompositions that can be normalized into structural traces for query–trace contrastive learning.

HotpotQA. HotpotQA contains approximately 113K Wikipedia-based multi-hop questions. It is designed to require reasoning over multiple supporting documents and provides sentence-level supporting facts, making it useful for evaluating whether a model can ground its answer in relevant evidence. It also includes comparison-style factoid questions, which test whether models can extract and compare entity properties.

2WikiMultiHopQA. 2WikiMultiHopQA combines structured Wikidata information with unstructured Wikipedia text. It introduces evidence information that contains reasoning paths, enabling both answer evaluation and reasoning-step evaluation. The commonly used split contains 167,454 training, 12,576 development, and 12,576 test examples. Its construction pipeline is designed to guarantee multi-hop reasoning steps and includes diverse reasoning patterns such as inference, compositional, comparison, and bridge-style questions.

B Further Information on Baselines

This appendix describes the baselines and distinguishes demonstration selection from evidence retrieval. The controls, semantic retrievers, and ICL methods operate on the supplied target context; RAG methods provide evidence-oriented comparisons rather than controlled tests of demonstration selection.

Vanilla-QA. Vanilla-QA directly prompts the frozen language model to answer the target question using the provided context, without any in-context demonstrations. It measures the base ability of the backbone model under the same input context.

Random-Demo. Random-Demo samples demonstrations uniformly from the candidate pool. It tests whether gains come from meaningful demonstration selection rather than simply adding examples to the prompt.

Facets-Match. Facets-Match retrieves demonstrations using discrete structural facets, such as hop count, answer type, or coarse operation pattern. It serves as a hard matching baseline for manually exposed structural features.

BM25. BM25 (Robertson and Zaragoza, 2009) is a sparse lexical retrieval baseline based on term matching. It evaluates whether word-level overlap is sufficient for selecting useful demonstrations or evidence.

Contriever. Contriever (Izacard et al., 2021) is a dense retriever trained without supervised relevance labels. It represents a semantic retrieval baseline based on learned dense representations.

ColBERT-v2. ColBERT-v2 (Santhanam et al., 2021) is a late-interaction retriever that matches query and document token representations. It provides a stronger semantic retrieval baseline than single-vector dense matching.

DeBERTa-v3-Large. DeBERTa-v3-Large (He et al., 2023) is used as an encoder-based semantic retrieval baseline. Candidate demonstrations are ranked by embedding similarity without STRIDE's structural training objective.

Vanilla RAG. Vanilla RAG (Lewis et al., 2020) retrieves external evidence and conditions the language model on the retrieved passages. It differs from STRIDE because it focuses on evidence retrieval rather than in-context demonstration extraction.

RAPTOR. RAPTOR (Sarthi et al., 2024) builds hierarchical summaries over documents and retrieves from this tree-structured representation. It is included as a representative hierarchical RAG baseline.

Dataset	Size	Split	Structure	Context	Evidence Annotation
MuSiQue	24.8K	19,938 / 2,417 / 2,459	2–4 hop compositional questions; six reasoning graph shapes.	20 paragraphs per instance, including supporting and retrieved distractor paragraphs.	Gold question decompositions, intermediate answers, and supporting paragraphs.
HotpotQA	113K	90,447 / 7,405 / 7,405	Natural multi-hop questions, mainly bridge and comparison reasoning.	Wikipedia paragraphs; commonly evaluated in distractor and full-wiki settings.	Sentence-level supporting facts for explainable reasoning.
2WikiQA	192.6K	167,454 / 12,576 / 12,576	2–4 hop questions generated from structured and unstructured sources; includes comparison, inference, compositional, and bridge-style reasoning.	Wikipedia paragraphs linked with Wikidata-derived reasoning paths.	Evidence chains, supporting facts, and reasoning-path information.

Table 8: Detailed metadata of the multi-hop QA datasets. Splits are reported as train/dev/test.

HippoRAG. HippoRAG (Gutiérrez et al., 2024) augments retrieval with memory-inspired graph structures for multi-hop reasoning. It represents a strong graph-based RAG baseline.

Vanilla CoT. Vanilla CoT (Wei et al., 2022) prompts the language model to generate intermediate reasoning steps before answering. It evaluates whether explicit reasoning prompts alone can improve multi-hop QA.

CoT-SC. CoT-SC (Wang et al., 2022) samples multiple reasoning chains and selects the final answer by consistency. It tests whether decoding-time reasoning diversity improves answer accuracy.

Complex-CoT. Complex-CoT (Fu et al., 2023) selects more complex demonstrations for chain-of-thought prompting. It is relevant because it also studies demonstration choice, but uses reasoning complexity rather than structural compatibility.

Auto-CoT. Auto-CoT (Zhang et al., 2024b) automatically constructs chain-of-thought demonstrations through clustering and rationale generation. It provides an automatic demonstration-construction baseline for ICL.

RGER. RGER (Lin et al., 2025) is a reasoning-enhanced retrieval baseline for selecting demonstrations. It incorporates reasoning signals into example retrieval, making it a close ICL baseline to STRIDE.

B.1 Learned, Sequential, and Policy-Based Selection

Learned retrieval. EPR (Rubin et al., 2022) trains a retriever using language-model feedback on candidate demonstrations. UDR (Li et al., 2023) learns demonstration rankings from language-model feedback across tasks. Both are relevant trained-selector comparisons, whereas STRIDE learns query–trace alignment from structural supervision. EPR and UDR are not evaluated in the reported tables; their absence limits claims about superiority over learned demonstration retrieval in general. A matched comparison would require the same training pool, target paragraphs, generator, and demonstration budget, together with a stated feedback-labeling and retriever-training procedure.

Sequential and policy-based selection. Selection need not rank demonstrations independently. Active Example Selection formulates it as a sequential decision problem and learns a selection policy with reinforcement learning.[1] Se^2 uses feedback on evolving contexts to account for relationships and order among examples.[2] These approaches address the utility of an example sequence. STRIDE instead computes structural and semantic scores, assigns positive and contrastive roles, and renders

[1] Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. *Active Example Selection for In-Context Learning*. EMNLP, pp. 9134–9148. DOI: 10.18653/v1/2022.emnlp-main.622.

[2] Haoyu Liu et al. 2024. *Se^2: Sequential Example Selection for In-Context Learning*. Findings of ACL, pp. 5262–5284. DOI: 10.18653/v1/2024.findings-acl.312.

a fixed prompt; it does not learn a sequential selection policy or revise the selected examples using target-time generator feedback.

Subset construction and contrastive guidance. Determinantal point processes (DPPs) and correlation-aware selection model relationships within a selected subset (Ye et al., 2023; Du et al., 2025). Contrastive CoT and learning-from-mistakes approaches expose invalid rationales or failed solutions (Chia et al., 2023; An et al., 2024). STRIDE's contrastives differ in role: a demonstration can have a correct answer to its own question yet be an unsuitable reasoning template for the target. Its contribution is the learned structural criterion used for both selection and role assignment, not the introduction of demonstration retrieval, structured reasoning, or negative examples individually.

Interpretation of RAG comparisons. RAPTOR and HippoRAG address evidence access, while STRIDE selects instructional examples. Their QA scores are informative system-level comparisons, but they do not isolate the effect of structural demonstration selection. The matched ICL controls and interventions in Appendix H are more directly relevant to that question.

C Implementation Details

Table 9 summarizes the main implementation details used in STRIDE. Unless explicitly ablated, ICL and semantic demonstration-retrieval methods use the same demonstration budget and frozen generator. RAG baselines use the same frozen generator and a comparable evidence budget, but differ in the information they retrieve.

D Formal Motivation

This section gives an assumption-based motivation for structural routing, not a guarantee of downstream QA improvement. The analysis concerns pairwise rankings of individual demonstrations. It does not model interactions among demonstrations, prompt order, or the internal reasoning process of the frozen generator.

Let q be a target query and d_i a candidate demonstration with question q_i. Assume that each query has a latent problem-solving structure $z(q) \in \mathbb{R}^d$, with $\|z(q)\|_2 = 1$. Define

$$\kappa_i = \mathrm{sim}(z(q), z(q_i)), \quad \sigma_i = \mathrm{sim}_{\mathrm{sem}}(q, q_i), \tag{14}$$

where κ_i denotes structural compatibility and σ_i denotes semantic similarity. The expected utility of using d_i as a demonstration is assumed to be

$$U_i = \beta\kappa_i + (1-\beta)\sigma_i + \xi_i, \quad \beta \in [0,1], \tag{15}$$

where β measures the contribution of structural compatibility and ξ_i denotes unmodeled prompt-specific residuals. Semantic retrieval selects

$$d_{\mathrm{sem}} = \arg\max_{d_i \in \mathcal{D}} \sigma_i. \tag{16}$$

For two candidates d_i, d_j with $\sigma_i > \sigma_j$, semantic retrieval prefers d_i. However, d_j has higher expected utility whenever

$$\begin{aligned} U_j > U_i \iff \beta(\kappa_j - \kappa_i) > \\ (1-\beta)(\sigma_i - \sigma_j) + (\xi_i - \xi_j). \end{aligned} \tag{17}$$

Thus, a semantically closer candidate can be suboptimal when its structural compatibility is sufficiently lower. Motivated by this, STRIDE estimates structural compatibility using the learned signature $\hat{z}(q)$:

$$\hat{\kappa}_i = \mathrm{sim}(\hat{z}(q), \hat{z}(q_i)). \tag{18}$$

The positive routing score is

$$R_i = \alpha\hat{\kappa}_i + (1-\alpha)\sigma_i. \tag{19}$$

If $\alpha = \beta$ and $\hat{\kappa}_i = \kappa_i$, then

$$R_i = U_i - \xi_i. \tag{20}$$

Hence, R_i recovers the expected utility up to the residual term. Assume that the structural estimator has bounded error:

$$\|\hat{z}(q) - z(q)\|_2 \le \eta, \quad \|\hat{z}(q_i) - z(q_i)\|_2 \le \eta. \tag{21}$$

Since all vectors are normalized,

$$\begin{aligned} |\hat{\kappa}_i - \kappa_i| &= \left|\hat{z}(q)^\top \hat{z}(q_i) - z(q)^\top z(q_i)\right| \\ &\le \|\hat{z}(q) - z(q)\|_2 + \|\hat{z}(q_i) - z(q_i)\|_2 \\ &\le 2\eta. \end{aligned} \tag{22}$$

It follows that

$$|R_i - (U_i - \xi_i)| \le 2\alpha\eta + |\alpha - \beta|\,|\kappa_i - \sigma_i|. \tag{23}$$

Define the individual approximation error

$$E_i = R_i - (U_i - \xi_i), \tag{24}$$

Item	Setting
Evaluation subsets	Fixed development-set subsets are used for all methods on each dataset.
Demonstration pool	Candidate demonstrations are drawn from the corresponding training split.
Sampling seed	A fixed random seed is used for subset sampling, random demonstrations, and analysis sampling.
Number of demonstrations	Default setting uses $k = 4$ total demonstrations, with $k_+ = 2$ positive demonstrations and $k_- = 2$ contrastive demonstrations when contrastive routing is enabled.
Semantic neighborhood	Contrastive candidates are selected from the top-M semantic neighborhood; default $M = 64$.
Routing coefficients	Default $\alpha = 0.5$ for structural–semantic positive routing and $\gamma = 1.0$ for contrastive structural penalization.
Query / trace encoder	DeBERTa-v3-Large initialized encoder with query–trace contrastive learning.
Encoder training	Learning rate 2×10^{-5}, batch size 8, gradient accumulation 4, maximum length 96, 10 epochs, fp16 training.
Auxiliary probes	Hop count, answer type, and coarse operation-pattern probes are trained jointly with the structural objective.
Generator	Qwen3-8B is used as the default frozen generator; Llama-3.1-8B and Ministral-8B are used for backbone robustness.
Decoding	Default STRIDE setting: temperature 0, deterministic decoding, and a fixed maximum generation length. Self-consistency sampling is a separate baseline protocol.
Prompt format	STRIDE uses contrastive role traces followed by positive role traces and the target question with candidate paragraphs (illustrated in Figure 2).

Table 9: Implementation and reproducibility details for STRIDE.

with

$$|E_i| \le \Omega_i = 2\alpha\eta + |\alpha - \beta|\,|\kappa_i - \sigma_i|. \tag{25}$$

For two candidates d_i, d_j,

$$\begin{aligned} R_i - R_j &= (U_i - U_j) - (\xi_i - \xi_j) + (E_i - E_j) \\ &\ge (U_i - U_j) - |\xi_i - \xi_j| - \Omega_i - \Omega_j. \end{aligned} \tag{26}$$

Let

$$\begin{aligned} \Gamma_{ij} &= \Omega_i + \Omega_j \\ &= 4\alpha\eta + |\alpha - \beta|\,(|\kappa_i - \sigma_i| + |\kappa_j - \sigma_j|)\,. \end{aligned} \tag{27}$$

If

$$U_i - U_j > \Gamma_{ij} + |\xi_i - \xi_j|, \tag{28}$$

then

$$R_i > R_j. \tag{29}$$

Thus, structure-aware routing preserves the utility preference when the utility margin exceeds structural estimation error, score-weight mismatch, and residual noise. For contrastive routing with $\gamma \ge 0$, STRIDE scores candidates in the semantic neighborhood $\mathcal{C}_q$ by

$$C_i = \sigma_i - \gamma\hat{\kappa}_i, \quad d_i \in \mathcal{C}_q. \tag{30}$$

Let the ideal contrastive score be

$$C_i^* = \sigma_i - \gamma\kappa_i. \tag{31}$$

Then

$$|C_i - C_i^*| \le 2\gamma\eta. \tag{32}$$

For two candidates d_i, d_j, if

$$C_i^* - C_j^* > 4\gamma\eta, \tag{33}$$

then

$$C_i > C_j. \tag{34}$$

Therefore, contrastive routing remains reliable when the structural estimation error is smaller than the contrastive-score margin. Equivalently, selected contrastive demonstrations approximate

$$d_{\text{con}} = \arg\max_{d_i \in \mathcal{C}_q} \left(\sigma_i - \gamma\hat{\kappa}_i\right), \tag{35}$$

favoring candidates with high semantic plausibility and low structural compatibility. These bounds explain a possible benefit of structural routing when structural compatibility contributes to utility and semantic similarity is an imperfect proxy. Neither $\beta > 0$ nor imperfect correlation between semantic and structural scores, however, is sufficient by itself to ensure an improvement: the relevant utility margins must also exceed estimation error and residual differences. Likewise, stability of a contrastive-score ranking does not establish that adding the selected contrastives improves generation. The assumptions on β, η, and the residuals are not empirical estimates; the experiments assess downstream utility separately.

E Operational Setting and Reasoning Structure

E.1 What Is Retrieved and What Is Frozen?

STRIDE is a non-agentic, single-generation-call ICL framework. At inference, the target question q and its candidate paragraphs $\mathcal{P}$ are fixed. The retrieval pool consists of training demonstrations, not an external evidence corpus. STRIDE encodes the target question, selects and routes demonstrations, builds one prompt, and invokes a frozen generator. There is no iterative evidence retrieval, tool use, or feedback-driven replanning. A single generation call can still decode multiple output tokens; it does not mean a single token-level forward evaluation.

The distinction between training and inference is important. During training, the query encoder and trace encoder are optimized using paired questions and normalized traces, with auxiliary structural labels. At inference, the target structural signature is computed from q alone. The target gold answer, decomposition, and supporting-fact labels are not inputs to query structure inference. Demonstration traces and answers come from the training pool and are used to render the selected examples. The generator is never updated by STRIDE, but the selector is trained; STRIDE is therefore not a wholly training-free method.

E.2 An Operational, Task-Bounded Definition

Reasoning structure denotes the abstract multi-hop problem-solving pattern remaining after surface entities are normalized. It includes the number of steps, dependencies among intermediate results, operation patterns, expected answer type, and evidence roles. Table 10 makes these components explicit. The learned embedding is not a hand-coded concatenation of these five fields: normalized traces provide the alignment target, while hop, answer-type, and operation probes supply auxiliary supervision.

For example, two questions about different people may both require finding an intermediate person and then retrieving that person's birthplace. They can share a dependency pattern despite different topics. Conversely, questions about the same people may ask for either an attribute lookup or a comparison; topical overlap does not make the two operation patterns interchangeable. This is an illustrative distinction, not an additional benchmark example or measured result.

The present normalization is designed for entity- and relation-centric MHQA with identifiable intermediate steps. It does not establish coverage of open-ended research questions, unconstrained synthesis, or all forms of reasoning. Transfer across the three evaluated datasets tests reuse within this scope rather than domain-independent reasoning competence.

Component	Meaning and example
Hop count	Number of intermediate reasoning steps; e.g., a two-step rather than four-step problem.
Dependency pattern	How a step uses a previous result; e.g., step 3 consumes the entity found in step 2.
Operation pattern	Abstract operations and their order; e.g., bridge lookup, attribute lookup, and comparison.
Answer type	Coarse expected output type, such as date, number, location, or person.
Evidence role/path	Which evidence supports a step, such as identifying a bridge entity versus obtaining the final attribute.

Table 10: Operational components of reasoning structure. These describe the supervision and diagnostics, not a claim of a universal reasoning ontology.

F Trace Normalization and Pseudo-Trace Supervision

F.1 From Annotations to Structural Supervision

The normalized trace $\ell(\tau)$ is a serialized abstraction of an annotated decomposition or a pseudo-trace. Its representation preserves the supplied step order, references to intermediate results, and operation-indicative phrases. Entity mentions are replaced by indexed placeholders such as $\langle \mathrm{ENT}_j \rangle$, cross-step references by $\langle \mathrm{REF}_t \rangle$, and intermediate answers by coarse type markers, including $\langle \mathrm{DATE} \rangle$, $\langle \mathrm{NUM} \rangle$, and $\langle \mathrm{LOC} \rangle$. The dependency structure $\mathcal{E}$ is retained in the serialized trace.

This produces two related forms of supervision: the normalized text for query–trace contrastive learning, and trace-derived labels for the auxiliary probes. Dependency and evidence information can contribute through the trace without each requiring a separate prediction head. A normalized trace used to train the encoder is also distinct from a role trace shown to the generator: the latter includes the selected demonstration's question, instructional role, and the evidence or answer fields described in Appendix G.

MuSiQue: annotated decompositions. The dataset supplies decomposition steps, intermediate answers, and supporting paragraphs. These support direct normalization of each step, preservation of references to earlier intermediate answers, and derivation of hop, answer-type, operation, and support labels. This is stronger supervision than constructing an ordering from supporting facts alone, although coarse operation labels can still be ambiguous.

HotpotQA: supporting facts and question type. Pseudo-traces use sentence-level supporting facts together with the annotated question type. Supporting facts identify the evidence to associate with intermediate operations; bridge and comparison types constrain the kind of reasoning pattern represented. The question and evidence provide cues to answer type and evidence role. However, a set of supporting sentences is not an ordered decomposition. In particular, supporting-fact order does not by itself establish the order in which reasoning steps must be executed. The resulting trace is weak structural supervision, not a gold step-by-step program.

2WikiQA: evidence chains and reasoning paths. The evidence-chain and reasoning-path annotations supply linked entities and relations for constructing intermediate operations and cross-step dependencies. These annotations provide hop, reasoning-type, answer-type, and support information. Mapping evidence entities to mentions and linking successive steps can remain ambiguous, so a path-derived pseudo-trace is not assumed to be noise-free.

Table 11 summarizes what each annotation source supports and the associated ambiguity. This specifies the representation and supervision sources; it is not an executable specification of unreported entity-linking, tie-breaking, or template-induction heuristics.

F.2 Sensitivity to Perturbed Trace Supervision

The noise intervention perturbs trace-derived supervision during QSI training rather than corrupting evaluation questions or answers. At noise rate ϵ, a randomly selected fraction ϵ of training instances has one structural field corrupted before the trace is rendered. The field-specific perturbations are: dropping or duplicating one step for hop count; replacing the answer type; substituting an operation label; rewiring a dependency to another valid previous step; or swapping support/evidence roles within the candidate paragraph set. Both the normalized trace and its corresponding auxiliary labels are updated consistently with the corruption.

The target questions, candidate paragraphs, gold answers, demonstration pool, and test-time inference protocol remain unchanged. Thus, the rate is a fraction of perturbed training instances, not a token error rate. The zero-perturbation condition retains the original pseudo-traces and should not be interpreted as gold or noise-free supervision.

At 10% perturbation, F1 decreases by 0.77 points on HotpotQA and 0.79 on 2WikiQA; at 20%, the decreases are 2.02 and 2.14 points. At 40%, the decreases reach 4.63 and 8.36 points. These results show sensitivity to supervision quality, with larger degradation at the highest tested rate. They do not identify the real-world noise rate, separate the effects of individual corrupted fields, or establish robustness to every kind of annotation error.

G Prompt Templates and Role Rendering

QSI and RRR are encoder-and-score computations, not separately prompted language-model agents. The prompt template belongs to SPC, after demonstration selection. Figure 6 provides a complete schematic rendering of the fixed prompt and the two demonstration formats. Braced names denote fields populated from the target instance or the selected training demonstrations; they are not literal input tokens.

The ordering shown follows Section 3.4: contrastives precede positives, followed by the target question and its candidate paragraphs. This is a schema-level specification rather than a byte-for-byte record of a tokenizer's chat template. It specifies the block order used to interpret the reversed-order ablation without assuming unreported chat-wrapper tokens.

Positive demonstrations. The question, ordered abstract steps, supporting evidence, and answer belong to the demonstration. Evidence identifiers refer to that demonstration's paragraphs or sentences, not to identically numbered target paragraphs. The model is instructed to use the compatible organization of reasoning, not to transfer demonstration-specific facts into the target answer.

Contrastive demonstrations. A contrastive is a structural near-miss, not necessarily an incorrect training example. Its answer can be correct for

Dataset	Annotation source	Structural information	Principal ambiguity
MuSiQue	Gold decompositions, intermediate answers, and supporting paragraphs	Step count, cross-step references, answer type, operations, and support	Assigning a unique coarse operation label to a step
HotpotQA	Supporting facts and question type	Reasoning type, answer type, and evidence roles used to build pseudo-steps	Supporting facts do not uniquely determine step order
2WikiQA	Evidence chains and reasoning paths	Hop count, linked-step dependencies, reasoning type, answer type, and support	Linking evidence entities to mentions and intermediate steps

Table 11: Supervision sources and limitations of trace construction. Annotation availability is not itself a quantitative measure of trace correctness.

Perturbed instances (%)	HotpotQA F1	2WikiQA F1
0	73.20	73.60
10	72.43	72.81
20	71.18	71.46
40	68.57	65.24

Table 12: Reported point estimates under additional perturbation of training trace supervision. This is a controlled corruption analysis, not an estimate of naturally occurring annotation noise.

its own question. The mismatch summary Δ_i is derived from routing signals and auxiliary predictions, such as a difference in hop count, answer type, or operation pattern. It must not depend on a gold target decomposition. The role instruction warns against copying the mismatched pattern; it does not label every statement in the demonstration as false.

H Supplementary Analyses

The following analyses separate questions about selection, role assignment, trace rendering, transfer, and generator choice. EM and F1 are reported on a 0–100 scale. The supplementary tables contain point estimates; uncertainty across training seeds and significance tests are not reported. Their contrasts are therefore descriptive rather than inferential. The original configuration names are retained so that differently controlled interventions are not conflated.

H.1 Routing and Prompt Construction

Table 13 compares structural positives, trace rendering, negative selection, and block order on MuSiQue. Structural positives with plain QA demonstrations reach 38.52 F1, compared with 36.44 for the semantic control; exposing structured traces raises the positive-only score to 39.96. Full STRIDE reaches 41.63 F1, exceeding the random-negative and reversed-order conditions by 1.82 and 0.99 points, respectively.

These contrasts support contributions from both selecting compatible examples and rendering their instructional roles. They do not yield an additive decomposition of the total gain. Removing a role can alter prompt composition, removing traces changes token length, and block-order changes affect position. In particular, the results should not be interpreted as a controlled estimate of a universal recency effect or as showing that prompt length is irrelevant.

H.2 Counterfactual Demonstration Interventions

The interventions in Table 14 keep the role template, demonstration count, and output format fixed while changing selection or the information exposed by demonstrations. Random selection removes query-specific selection; semantic selection retains surface relevance; hiding traces retains the selected examples but removes explicit structural steps; and shuffling positives across queries disrupts query–demonstration compatibility. The random-negative condition weakens the specifically selected contrastive role.

Relative to full STRIDE, shuffling positives reduces F1 by 3.42 points and hiding traces reduces it by 1.06 points. This is consistent with query-specific selection contributing beyond a shared role format, and with selected demonstrations retaining useful information even when explicit traces are hidden. A trace-hidden prompt and a positive-only plain-QA prompt need not have the same role composition; their separately reported scores should not be treated as duplicate measurements of one condition.

The interventions test output sensitivity to demonstrations. They do not directly observe which intermediate steps the generator executes,

Complete prompt schema (default: two contrastives and two positives)

```
You are given a multi-hop question and a set of candidate paragraphs.
Answer the question using only the provided candidate paragraphs.

The examples below are not facts for the target question.
They only show reasoning structures.

[Contrastive Demonstrations: structurally mismatched examples]
The following examples may look semantically related, but their
reasoning structures are not suitable for the target question.
Do not copy their reasoning pattern.
{CONTRASTIVE_DEMO_1}
{CONTRASTIVE_DEMO_2}

[Positive Demonstrations: structurally aligned examples]
The following examples have reasoning structures that are suitable
for the target question. Use them as guidance for how to organize
the reasoning.
{POSITIVE_DEMO_1}
{POSITIVE_DEMO_2}

[Target Question]
{TARGET_QUESTION}

[Candidate Paragraphs]
{TARGET_PARAGRAPHS}

Return the final answer only.
```

Positive demonstration schema

```
Positive Example:
Question: {DEMO_QUESTION}

Reasoning structure to imitate:
1. {ABSTRACT_STEP_1}
   Supporting evidence:
   {EVIDENCE_TEXT_OR_ID_1}
2. {ABSTRACT_STEP_2}
   Supporting evidence:
   {EVIDENCE_TEXT_OR_ID_2}
...
Answer: {DEMO_ANSWER}
```

Contrastive demonstration schema

```
Contrastive Example:
Question: {DEMO_QUESTION}

Reasoning structure not to imitate:
1. {ABSTRACT_STEP_1}
2. {ABSTRACT_STEP_2}
...

Why it is mismatched:
{MISMATCH_SUMMARY}

Answer: {DEMO_ANSWER}
```

Figure 6: Schematic SPC prompt and role-specific demonstration formats. The target is placed after the positive block, matching the method equation. The ellipses in the demonstration schemas indicate additional steps when present, not a requirement to truncate a trace.

Variant	EM	F1
Semantic retrieval + plain demonstrations	28.63	36.44
Structural positives only + plain QA demonstrations	31.08	38.52
Structural positives only + structured traces	32.17	39.96
Structural positives + random negatives	32.04	39.81
Structural positives + contrastives, reversed order	32.72	40.64
Full STRIDE	**33.85**	**41.63**

Table 13: Fine-grained routing and prompt analysis on MuSiQue. Positive-only conditions remove the contrastive role; the reversed-order condition reverses the positive/contrastive block order. These configurations are not a complete factorial ablation; no token-length-matched comparison is reported.

Intervention	Controlled or changed information	EM	F1
Random demos with the same role template	Prompt format and demonstration count fixed; selection broken	19.84	27.63
Semantic demos with the same role template	Semantic relevance and format preserved; structural matching weakened	28.63	36.44
STRIDE positives, traces removed	Selected demonstrations preserved; explicit structural traces hidden	33.29	40.57
STRIDE positives shuffled across queries	Prompt format preserved; query-specific compatibility disrupted	30.74	38.21
STRIDE positives + random negatives	Positive selection retained; contrastive selection weakened	32.04	39.81
Full STRIDE	Structural positives with semantic near-miss contrastives	**33.85**	**41.63**

Table 14: Counterfactual demonstration interventions on MuSiQue. These are separately reported controls, not replacements for the main-table Random-Demo or module-ablation rows. Fixed example counts and templates do not imply identical token lengths.

whether it faithfully follows the displayed trace, or whether evidence selection rather than another prompt effect explains a particular gain. Establishing those stronger process-level claims would require additional controlled measurements.

H.3 Which Structural Factors Transfer?

The transfer protocol in Section 4.3.3 trains QSI on a source dataset and applies it to a target while retaining the target demonstration pool, generator, prompt format, and routing parameters. Table 15 supplements answer-level transfer retention with the reported trace-label diagnostics. Hop and answer-type agreement concern coarse structural labels; operation-pattern F1 concerns overlap in operation labels; evidence-path agreement concerns the annotated evidence structure. None is an answer-accuracy metric or a measurement of the generator's internal reasoning.

Across the six directions, the reported agreement is higher for hop count and answer type than for evidence paths. This pattern is consistent with coarse structural properties being easier to reuse than dataset-specific evidence conventions. However, the metrics differ in label granularity and difficulty, so their numerical levels alone do not prove an intrinsic ordering of transferability. Positive transfer within these MHQA datasets also does not exclude dataset-specific shortcuts or establish generalization to free-form reasoning.

H.4 A Reasoning-Oriented Generator

Table 16 reports the additional MuSiQue comparison with the Qwen3-8B-Thinking condition. The gain column is STRIDE F1 minus semantic-ICL F1 within the same generator condition, rather than a comparison across different generators.

The thinking condition has stronger direct QA performance and a smaller, but positive, incremental gain from STRIDE over semantic ICL. This suggests that structural selection can remain useful in the reported reasoning-oriented setting without implying that demonstrations create new reasoning ability. The comparison does not establish robustness across reasoning-model families or a compute-matched advantage; the exact thinking-token budget is not specified in the reported configuration.

Source → target	Hop agreement	Answer-type agreement	Operation-pattern F1	Evidence-path agreement
MuSiQue → HotpotQA	71.8	76.4	61.2	55.6
MuSiQue → 2WikiQA	78.6	79.1	67.4	62.8
HotpotQA → MuSiQue	68.3	73.5	58.9	51.7
HotpotQA → 2WikiQA	80.2	82.5	70.6	65.3
2WikiQA → MuSiQue	74.7	77.8	63.5	57.9
2WikiQA → HotpotQA	70.1	75.2	60.4	54.1

Table 15: Reported cross-dataset structural diagnostics (0–100 scale). The four columns evaluate different properties and are not interchangeable measures of transfer accuracy.

Generator condition	Direct QA F1	Semantic ICL F1	STRIDE F1	Gain
Qwen3-8B	14.71	36.44	41.63	+5.19
Qwen3-8B-Thinking	33.87	39.28	42.16	+2.88

Table 16: Reported generator comparison on MuSiQue. The thinking condition is one additional setting, not a survey of reasoning-oriented models.

I Interpretation, Training Cost, and Scope

I.1 Demonstration Utility and Process Faithfulness

Min et al. (2022) show that demonstrations can help through label-space exposure, input distribution, and formatting, even with perturbed labels, in classification and multiple-choice settings.[3] GuideNER emphasizes explicit guidelines for conveying entity-label definitions.[4] These findings motivate caution about interpreting ICL gains as faithful execution of demonstrated reasoning, but do not determine how a particular MHQA prompt is used.

STRIDE's evidence concerns useful demonstration selection and output sensitivity to the reported interventions. External trace-label metrics characterize selected examples, not hidden generator computations. Improved selection and prompting therefore should not be equated with increased intrinsic reasoning ability or process faithfulness.

I.2 Training Cost and Unevaluated Comparisons

STRIDE incurs the cost of trace preparation and query–trace encoder training. Inference retains the query-side structural encoder and the frozen semantic encoder for routing; the generator remains frozen. Backbone robustness tests reuse across generators, not the absence of selector-training cost.

LoRA and instruction tuning instead adapt the generator. No such baselines, training-time measurements, peak-memory measurements, or end-to-end latency comparisons are reported here. The results therefore do not establish a cost–performance advantage over generator adaptation. A matched comparison should account for supervision preparation, training, prompt length, and inference cost, as well as QA accuracy. EPR/UDR evaluation and per-field trace ablations likewise remain open comparisons rather than additional reported results.

I.3 Scope

The current evidence is limited to English, entity- and relation-centric MHQA. Open-ended settings may require richer trace induction or abstract operation schemas. Annotation artifacts and biases can propagate from the demonstration pool, and structural compatibility cannot guarantee correct target-evidence grounding, intermediate-entity resolution, or answer normalization.

[3] Sewon Min et al. 2022. *Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?* EMNLP, pp. 11048–11064. DOI: 10.18653/v1/2022.emnlp-main.759.

[4] Shizhou Huang, Bo Xu, Yang Yu, Changqun Li, and Xin Alex Lin. 2025. *GuideNER: Annotation Guidelines Are Better than Examples for In-Context Named Entity Recognition*. AAAI, 39(23):24159–24166. DOI: 10.1609/aaai.v39i23.34591.