X-Router: Decoupling Knowledge and Reasoning for Cost-Effective LLM Inference

Zixuan Wang[1,2], Yinze Ding[1], Zihan Wang[1], Jinyu Guo[3], Zhenhong Zhou[4], Junhao Dong[4], Chaomeng Chen[2,5]*

[1]Harbin Institute of Technology [2]Great Bay University
[3]University of Electronic Science and Technology of China
[4]Nanyang Technological University [5]Tsinghua University
zixuanwang.cs@stu.hit.edu.cn, dingyinze@gmail.com,
2023112983@stu.hit.edu.cn, guojinyu@uestc.edu.cn,
zhenhong001@e.ntu.edu.sg, junhao003@ntu.edu.sg,
cmchen@gbu.edu.cn

Abstract

Large Language Models (LLMs) are often augmented with Retrieval-Augmented Generation (RAG) and Chain-of-Thought (CoT) prompting, yet static "always-on" use is computationally wasteful. Existing adaptive methods typically optimize a single axis, overlooking that evidence need and reasoning depth are only partially correlated. We present X-ROUTER, a dual-axis routing framework that separates retrieval necessity from reasoning necessity under a user-defined cost–quality trade-off. Offline, X-ROUTER profiles four pipelines (DIRECT, RAG, COT, RAG+COT) and derives supervision by selecting the utility-maximizing strategy that trades answer quality against token usage and latency. Online, a compact dual-head router, conditioned on cost weights, uses lightweight probes—retrieval-score dispersion (NQC) and single-pass draft negative log-likelihood (NLL)—to decide whether to invoke RAG and/or CoT without sampling or model internals. Across six QA benchmarks, X-ROUTER reduces token usage by up to 86% and latency by up to 84% while improving answer quality over strong baselines.

1 Introduction

Large language models (LLMs) have evolved from standalone predictors to the core of complex systems. To handle knowledge-intensive and reasoning-heavy tasks, two complementary mechanisms have become standard: Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) for bridging evidence gaps, and Chain-of-Thought (CoT) prompting (Wei et al., 2022) for eliciting multi-step logic. While effective, both mechanisms incur substantial overhead (tokens, retrieval calls, and latency). Consequently, a static "always-retrieve-and-always-reason" strategy is often wasteful. The desideratum is a system that invokes these expensive capabilities *only when they improve a user-defined cost–quality trade-off.*

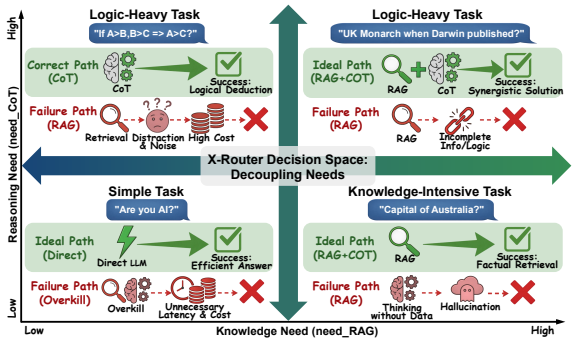

Figure 1: Decision space for routing queries along **Knowledge Need** (*need_RAG*) and **Reasoning Need** (*need_CoT*), inducing four pipelines: DIRECT, RAG, COT, and RAG+COT. Misrouting yields predictable failures, e.g., retrieval distraction/noise on logic-heavy queries, ungrounded answers when evidence is missing, and wasted tokens/latency from unnecessary CoT.

A central difficulty is that queries vary along (at least) two distinct dimensions. Some questions primarily lack *external evidence* and benefit from retrieval(Mallen et al., 2023); others require *reasoning depth* even when the evidence is present; many require both; and some require neither. Existing adaptive approaches typically emphasize a single axis. Adaptive RAG methods (Asai et al., 2024; Jiang et al., 2023) focus on whether to retrieve, implicitly treating reasoning strategy as fixed. Conversely, CoT routers (Zhang et al., 2025; Fang et al., 2025) decide when to reason deeply, often under

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 19856–19874
July 2-7, 2026 ©2026 Association for Computational Linguistics

a fixed knowledge context. In practice, however, knowledge difficulty and reasoning difficulty are only partially correlated, and conflating them leads to systematic inefficiencies: retrieval can inject irrelevant context for logic-dominant queries(Yoran et al., 2024; Liu et al., 2024), while CoT can amplify hallucinations and token usage when key evidence is missing. Figure 1 illustrates this two-axis view and the characteristic failure modes of misrouting. Applying RAG when reasoning is the bottleneck can distract generation with irrelevant evidence, while forcing CoT without sufficient evidence yields long but ungrounded rationales(Lanham et al., 2024). These observations suggest that difficulty should not be modeled as a single scalar; instead, an efficient system should explicitly decide *(i) whether retrieval is worth it* and *(ii) whether additional reasoning budget is worth it*, under a cost-aware objective.

We propose X-ROUTER, a dual-axis routing framework that operationalizes this idea. The key design choice is that X-ROUTER learns routing *from the cross-product of strategies*: for each query, we profile four executable pipelines (DIRECT, RAG, COT, RAG+COT) offline and select the one that maximizes a normalized, cost-aware utility trading answer quality against token and latency costs. This yields supervision that is inherently *two-dimensional*—the joint optimum over the two binary decisions—rather than independently assigned "retrieve?" and "think?" labels. At deployment time, X-ROUTER makes routing decisions using only lightweight, observable probes, avoiding sampling-based uncertainty estimators and access to model internals. Concretely, we use Normalized Query Commitment (NQC) (Shtok et al., 2012) from retriever score dispersion as a proxy for evidence adequacy, and a single-pass negative log-likelihood (NLL) (Bakman et al., 2024; Malinin and Gales, 2021) from a short direct draft as a proxy for generation-side uncertainty. Importantly, retrieval confidence is not identical to retrieval necessity; therefore, X-ROUTER combines retrieval-side signals with draft uncertainty to avoid retrieving when the model can already answer confidently.

To ensure practical tunability, X-ROUTER explicitly parameterizes the routing objective with user-specified cost weights, enabling different operating points under varying budget preferences (we further analyze sensitivity to these weights in experiments). Overall, our contributions are:

- **Dual-axis formulation:** We formalize efficient LLM inference as a two-dimensional routing problem that separates evidence acquisition from reasoning depth.
- **Utility-supervised routing:** We derive cost-aware supervision by profiling four pipelines and selecting the utility-maximizing strategy, yielding two axis labels (*need_RAG*, *need_CoT*) grounded in the joint optimum.
- **Lightweight, observable probing:** We design an efficient probe set based on retriever score dispersion (NQC) and generation-side uncertainty (NLL), enabling low-overhead routing without sampling or model internals.
- **Strong empirical gains:** On HotpotQA and five additional QA benchmarks, X-ROUTER reduces token usage by up to 86% and latency by up to 84% while improving answer quality.

2 Related Work

2.1 Adaptive Retrieval-Augmented Generation

A large body of work studies when and how to invoke retrieval for large language models instead of always conditioning on retrieved documents. Self-triggered frameworks such as Self-RAG (Asai et al., 2024) and DRAGIN (Su et al., 2024) let the model estimate its information need during generation, using reflection tokens or token-level uncertainty to decide whether and where to retrieve. Active retrieval approaches, including FLARE-style prompting (Jiang et al., 2023) and corrective retrieval (Yan et al., 2024), trigger retrieval only when draft continuations appear uncertain or current evidence is judged low quality(Yu et al., 2024). Complementary work develops lightweight controllers that gate retrieval using external features such as score dispersion and query properties (Zhang et al., 2024; Jeong et al., 2024), often casting "retrieve or not" as a supervised or online decision problem. These methods adapt retrieval usage along a single dimension, whereas we consider the choice of reasoning mode.

2.2 Efficient Reasoning and Chain-of-Thought Routing

Another line of research investigates when large language models should invoke costly chain-of-thought (CoT) reasoning. Early work on CoT prompting (Wei et al., 2022; Kojima et al., 2022) and automated CoT construction (Zhang et al.,

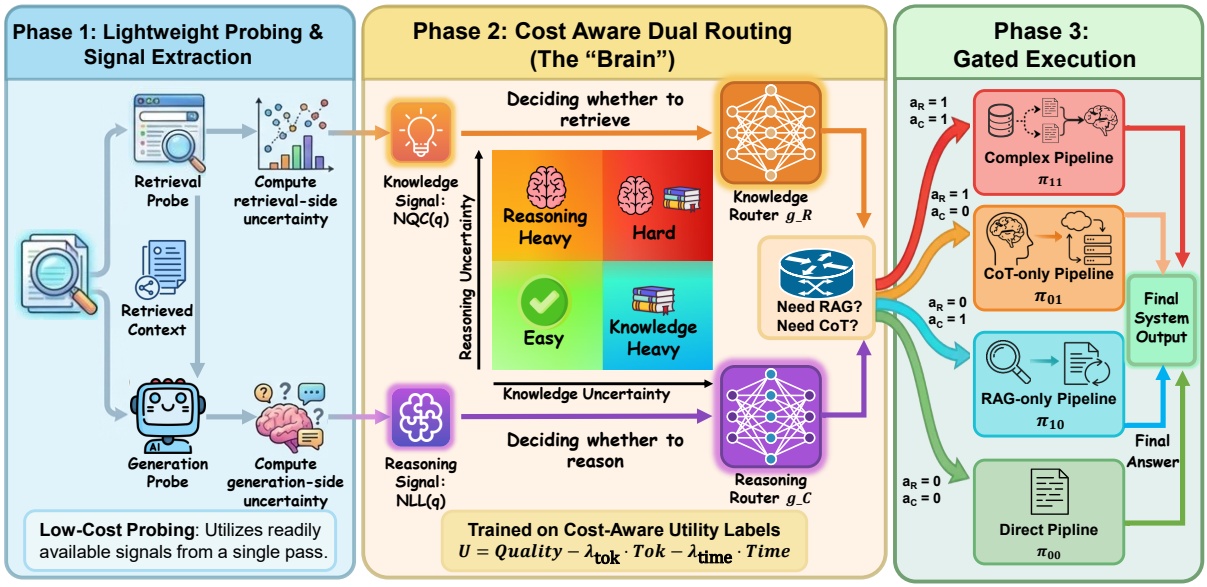

Figure 2: Overview of X-ROUTER. The inference workflow proceeds in three phases: **(1) Lightweight Probing**: Extracts uncertainty signals (NQC and draft NLL) via low-cost probes. **(2) Cost-Aware Dual Routing**: Dual routers, trained on offline utility profiles, predict knowledge and reasoning needs (a_R, a_C). **(3) Gated Execution**: Dynamically activates RAG and/or CoT pipelines to maximize the user-defined cost–quality trade-off.

2023) showed that narrated reasoning can substantially improve accuracy, while search-based methods such as Tree-of-Thoughts explore deeper reasoning traces when initial attempts are uncertain. More recent CoT routers learn to switch between short and long reasoning (Fang et al., 2025; Liang et al., 2025) or to continue thinking based on uncertainty and budget constraints (Zhang et al., 2025), and dynamic CoT methods in multimodal or tool-augmented settings use self-verification signals to decide whether to deepen trajectories. In parallel, input-adaptive computation allocation methods route inputs to cheaper or more expensive decoding procedures based on predicted utility (Schuster et al., 2022; Damani et al., 2024; Chen et al., 2023; Shnitzer et al., 2024). These approaches primarily view difficulty through the lens of reasoning effort, whereas we also model difficulty arising from missing external knowledge.

2.3 Joint Retrieval–Reasoning Frameworks

A third line of work tightly couples retrieval and reasoning in a single trajectory. IRCoT (Trivedi et al., 2023) alternates CoT steps with retrieval, using intermediate reasoning states to guide document selection on multi-hop QA. Related frameworks interleave retrieval with tool use (Yao et al., 2023) or train reinforcement learning policies that jointly control when to retrieve and how to reason over evidence, yielding strong performance on knowledge-intensive benchmarks. These systems design integrated retrieval–reasoning pipelines, whereas we focus on the higher-level problem of deciding when such advanced pipelines are warranted versus when simpler ones suffice.

3 Methodology

In this section, we present X-ROUTER, a framework that formalizes efficient inference as a two-dimensional routing problem along the axes of external information need (retrieval) and reasoning depth (CoT). We first define a family of four execution pipelines and a cost-aware utility function (Section 3.1). We then introduce a minimal probe that quantifies difficulty via Normalized Query Commitment (NQC) and generation-side negative log-likelihood (NLL) (Section 3.2). Next, we derive cost-aware supervision and train a dual-head MLP router (Sections 3.3–3.4), before describing the end-to-end inference procedure (Section 3.5).

3.1 Problem Formulation

Our goal is to adaptively disentangle the necessity of external knowledge acquisition from the depth of internal reasoning. Let q denote an input question. We assume access to a retrieval system $\mathcal{R}$ and a generator $\mathcal{G}$. We construct four distinct inference pipelines, $\Pi = \{\pi_{00}, \pi_{01}, \pi_{10}, \pi_{11}\}$, defined by the binary states of two decision variables:

- $a_R \in \{0, 1\}$: Controls knowledge source. State 0

relies solely on the model's parametric memory (NoRAG), while state 1 augments generation with retrieved context (RAG).

- $a_C \in \{0, 1\}$: Controls reasoning mode. State 0 produces a direct answer, while state 1 enables chain-of-thought reasoning (CoT).

For any pipeline π_{ab}, let $Q_{ab}(q) \in [0, 1]$ denote the answer quality (e.g., F1 score). We further measure the token cost $\mathrm{Tok}_{ab}(q)$ (total tokens) and the wall-clock latency $T_{ab}(q)$. We define a cost-aware utility:

$$U(\pi_{ab}; q) = Q_{ab}(q) - \lambda_{\text{tok}} \cdot \mathrm{Tok}_{ab}(q) - \lambda_{\text{time}} \cdot T_{ab}(q), \quad (1)$$

where $\lambda_{\text{tok}}, \lambda_{\text{time}} \geq 0$ are user-specified trade-off coefficients, allowing the system to penalize computationally expensive strategies (e.g., π_{11}) unless they yield significant quality gains.

In our experiments, we treat λ_{tok} and λ_{time} as two independent cost weights; Section 5.7 reports a sensitivity sweep over λ_{tok} while keeping λ_{time} fixed. Our objective is to learn a router mapping $g : q \to (a_R, a_C)$ that maximizes this utility using only lightweight signals derived from observable inputs and outputs of $\mathcal{R}$ and $\mathcal{G}$.

3.2 Difficulty Probing via NQC and Negative Log-Likelihood

We avoid accessing model internals (e.g., hidden states). We compute retrieval-side dispersion from retriever scores and estimate generation-side uncertainty from a lightweight likelihood signal of the model's own output. We then employ a "probe-then-route" strategy using these two observable uncertainty proxies.

Retrieval-Side Signal: NQC. To judge whether external knowledge is reliable, we execute a probing retrieval step to obtain the top-K documents D_q^K with retrieval scores $\{s(q, d) \mid d \in D_q^K\}$. Following the query performance prediction (QPP) literature on query commitment (Shtok et al., 2012; Meng et al., 2023), we use score dispersion of the top-ranked results as a lightweight proxy for retrieval confidence. We first compute the mean score

$$\mu(q) = \frac{1}{K} \sum_{d \in D_q^K} s(q, d), \quad (2)$$

and the standard deviation over the top-ranked scores

$$\sigma(q) = \sqrt{\frac{1}{K} \sum_{d \in D_q^K} \big(s(q, d) - \mu(q)\big)^2}. \quad (3)$$

We then define our retrieval-side probe as

$$\mathrm{NQC}(q) = \frac{\sigma(q)}{|\mu(q)|}. \quad (4)$$

Intuitively, $\mathrm{NQC}(q)$ measures the discriminative power of the retriever for a given query: when the top-K scores are flat relative to their scale (low $\mathrm{NQC}(q)$), the retriever exhibits low commitment and retrieved evidence is less reliable; when a few documents stand out (high $\mathrm{NQC}(q)$), retrieval is more confident and more likely to help. In our framework, this serves as a critical feature to decide between relying on parametric memory ($a_R = 0$) or enforcing retrieval augmentation ($a_R = 1$).

Generation-Side Signal: Negative Log-Likelihood. To estimate generation-side uncertainty, we use the (length-normalized) negative log-likelihood (NLL) of a single decoding pass. Let $\mathcal{G}$ generate an output sequence $y = (y_1, \ldots, y_T)$ under a direct-answer setting ($a_C = 0$), optionally conditioned on retrieved context. When token-level log-likelihoods are available, we compute

$$\mathrm{NLL}(q) = -\frac{1}{T} \sum_{t=1}^{T} \log p_{\mathcal{G}}(y_t \mid y_{<t}, q), \quad (5)$$

which provides a lightweight confidence signal without repeated sampling. Higher NLL indicates lower model confidence in its own generation, suggesting that additional reasoning budget (CoT, $a_C = 1$) may be beneficial.

Decoupled use. We use $\mathrm{NQC}(q)$ to make retrieval decisions and $\mathrm{NLL}(q)$ to make the CoT decision. To avoid conflating knowledge deficiency with reasoning difficulty, we compute $\mathrm{NLL}(q)$ under the no-CoT pipeline consistent with the retrieval decision; Section 4 details the construction.

3.3 Cost-Aware Supervision

Unlike traditional routing that predicts static difficulty labels (e.g., hard vs. easy), we directly optimize for execution utility. A query is routed to a more complex pipeline only if the performance gain justifies the marginal computational cost. We cast this as a supervised classification problem, where labels are derived automatically from offline executions of all four pipelines.

For each training query q, we first compute the utilities $U(\pi_{00}; q), U(\pi_{01}; q), U(\pi_{10}; q), U(\pi_{11}; q)$ for all four pipelines. We then generate binary labels $y_R, y_C \in \{0, 1\}$ by selecting the globally

best pipeline under the utility function. Formally, we identify

$$(a_R^*, a_C^*) = \arg \max_{(a_R, a_C) \in \{0,1\}^2} U\big(\pi_{a_R a_C}; q\big), \quad (6)$$

and define

$$y_R(q) = a_R^*, \qquad y_C(q) = a_C^*, \quad (7)$$

where $y_R(q)$ (resp. $y_C(q)$) indicates whether the utility-maximizing pipeline for q uses retrieval (resp. CoT). In our experiments, we also refer to these oracle labels as *need_RAG* and *need_CoT*. Therefore, a more complex strategy (e.g., enabling RAG or CoT) is labeled as "necessary" only when it belongs to the pipeline with the highest overall utility. This avoids the common pitfall of routing based solely on difficulty, which often leads to over-spending on intractable queries. In practice, we estimate $U(\pi_{ab}; q)$ by running all four pipelines offline under the same backbone and retrieval configuration, and then train a lightweight router to select among them at inference time.

3.4 Dual MLP Routers

We train two binary classifiers to predict the retrieval and reasoning labels from their corresponding probes:

$$\begin{aligned} \hat{y}_R(q) &= g_R\big(\mathrm{NQC}(q); \theta_R\big), \\ \hat{y}_C(q) &= g_C\big(\mathrm{NLL}(q); \theta_C\big), \end{aligned} \quad (8)$$

where g_R and g_C are lightweight Multilayer Perceptrons (MLPs) with two hidden layers, θ_R and θ_C are learnable parameters of MLPs. We employ lightweight MLPs to map the probe signals to routing probabilities. While simple thresholds are often used, the MLP approach allows for calibrated probability estimates $\hat{y} \in (0, 1)$ and can capture potentially non-monotonic relationships between uncertainty signals and optimal routing decisions (e.g., extremely low NLL might sometimes indicate hallucination in specific contexts).

Given the cost-aware labels $y_R(q)$ and $y_C(q)$ derived in Section 3.3, we optimize the router parameters with standard binary cross-entropy losses ℓ_{BCE} (Goodfellow et al., 2016; Bishop, 2006). $\hat{y}_R(q), \hat{y}_C(q) \in (0, 1)$ denote the predicted probabilities after a sigmoid output layer.

The training objectives for the retrieval and reasoning routers are then

$$\mathcal{L}_R = \mathbb{E}_q\Big[\ell_{\mathrm{BCE}}\big(y_R(q), \hat{y}_R(q)\big)\Big], \quad (9)$$

$$\mathcal{L}_C = \mathbb{E}_q\Big[\ell_{\mathrm{BCE}}\big(y_C(q), \hat{y}_C(q)\big)\Big]. \quad (10)$$

At inference time, the continuous outputs are discretized into routing decisions:

$$a_R(q) = \mathbb{I}\big[\hat{y}_R(q) > \tau_R\big], \quad (11)$$

$$a_C(q) = \mathbb{I}\big[\hat{y}_C(q) > \tau_C\big]. \quad (12)$$

These decisions jointly select one of the four executable pipelines $\pi_{a_R a_C}$. Maintaining two complementary routers makes it straightforward to attribute failures to either a knowledge decision error (misclassified a_R) or a reasoning decision error (misclassified a_C).

3.5 End-to-End Inference Workflow

The full lifecycle of a query within X-ROUTER proceeds as follows:

- **Probe retrieval:** Execute a probing retrieval to compute $\mathrm{NQC}(q)$.
- **Route retrieval:** Predict a_R from $\mathrm{NQC}(q)$.
- **Probe generation:** Run a no-CoT generation under $\pi_{a_R 0}$ to obtain $\mathrm{NLL}(q)$.
- **Route CoT and execute:** Predict a_C from $\mathrm{NLL}(q)$ and execute $\pi_{a_R a_C}$, reusing the no-CoT completion when $a_C = 0$.

This "probe-then-route" architecture ensures overhead is amortized: easy queries exit early, while the system reserves its computational budget for queries that benefit from RAG or CoT. All efficiency numbers reported in Section 5.2 and Appendix A are measured under this full end-to-end workflow, including both probing stages. When $a_C = 0$, the draft completion is reused as the final answer; when $a_C = 1$, the query incurs an additional CoT generation pass, so the average latency depends on how often a workload is routed to CoT.

4 Experimental Setup

4.1 Benchmarks

We evaluate X-ROUTER on six QA benchmarks that span knowledge-intensive multi-hop QA, commonsense reasoning, math reasoning, and domain-specific QA: **HotpotQA** (Yang et al., 2018), **2WikiMultiHopQA** (2WIKI) (Ho et al., 2020), **StrategyQA** (Geva et al., 2021), **GSM8K** (Cobbe et al., 2021), **FiQA** (Maia et al., 2018), and **MedQA** (Jin et al., 2020). Following Trivedi et al. (2023), we report on 500-example subsets sampled from the official development or test splits.

4.2 Four Pipelines

We use the four pipelines in Section 3.1, corresponding to the cross-product of retrieval (a_R)

Dataset	Metric	IRCoT	Self-RAG	LearnHard	Vote	To-CoT	Joint-Router	X-ROUTER	Router-SC	Oracle-U
HotpotQA	F1	29.3	31.3	**46.4**	31.7	38.9	43.8	**52.9** (↑14.0%)	57.8 (↑24.6%)	64.3 (↑38.6%)
	EM	18.1	19.0	**33.0**	19.4	24.8	32.0	**37.8** (↑14.5%)	43.8 (↑32.7%)	49.4 (↑49.7%)
	ROUGE-L	26.3	31.3	**46.3**	31.6	32.8	43.7	**52.8** (↑14.0%)	57.4 (↑24.0%)	64.2 (↑38.7%)
2WIKI	F1	24.2	24.6	36.0	28.7	**38.9**	38.1	**42.3** (↑8.7%)	49.0 (↑26.0%)	58.8 (↑51.2%)
	EM	13.4	14.8	27.0	17.8	**32.7**	32.4	**35.2** (↑7.6%)	41.8 (↑27.8%)	50.0 (↑52.9%)
	ROUGE-L	24.2	24.6	36.0	28.7	**38.9**	38.0	**42.5** (↑9.3%)	49.1 (↑26.2%)	58.8 (↑51.2%)
FiQA	F1	28.4	28.4	30.2	**31.4**	21.7	22.3	**31.5** (↑0.3%)	31.8 (↑1.3%)	33.0 (↑5.1%)
	ROUGE-L	14.6	14.7	16.4	**16.6**	13.7	14.0	**16.9** (↑1.8%)	17.6 (↑6.0%)	18.2 (↑9.6%)
StrategyQA	AC	**58.4**	58.2	58.2	57.4	55.4	58.2	**61.4** (↑5.1%)	67.0 (↑14.7%)	74.4 (↑27.4%)
GSM8K	AC	32.9	35.6	38.4	58.4	**60.9**	32.4	**62.8** (↑3.1%)	67.3 (↑10.5%)	76.2 (↑25.1%)
MedQA	AC	38.0	47.4	46.6	38.4	46.0	**50.2**	**58.8** (↑17.1%)	62.7 (↑24.9%)	76.2 (↑51.8%)

Table 1: Main results on the evaluation subsets in Section 4 (all numbers are in %). We highlight the best and second-best scores among non-oracle methods (excluding Router-SC and Oracle-U). Parenthetical (↑r%) reports the relative improvement over ·.

and reasoning (a_C): π_{00} (NoRAG+Direct), π_{01} (NoRAG+CoT), π_{10} (RAG+Direct), and π_{11} (RAG+CoT). All pipelines share the same backbone generator and differ only in whether they retrieve external context and/or enable CoT.

4.3 Retriever and Prompts

We use a hybrid (dense+sparse) retriever with Maximum Marginal Relevance (MMR) reranking. Detailed hyperparameters for document chunking, retrieval candidates, and reranking thresholds are provided in Appendix C.2. Prompt templates for direct answering and CoT prompting are listed in Appendix C.3; for NoRAG pipelines, the context field is empty.

4.4 Backbone Model

Unless otherwise specified, we use **Qwen2.5-7B** (Team, 2024) (denoted as Q2.5-7B) as the default generator for all datasets. For the model swap study (Section 5.6), we additionally report results with **Qwen3-8B** (Yang et al., 2025) (Q3-8B), **Qwen2.5-32B-Instruct** (Team, 2024) (Q2.5-32B), and **DeepSeek-R1-Distill-Qwen-14B** (DeepSeek-AI et al., 2025) (DS-R1-14B).

4.5 Baselines

We compare against baselines that adaptively gate retrieval and/or reasoning, spanning retrieval-centric RAG controllers (Self-RAG (Asai et al., 2024), LearnHard (Damani et al., 2024)), CoT-centric reasoning routers (To-CoT-or-Not (Sprague et al., 2025)), and hybrid retrieve–reason approaches that interleave the two but do not explicitly perform joint 2D routing over the cross-product space (IRCoT (Trivedi et al., 2023)); we also include an aggregation-style baseline (Vote (Wang et al., 2023)) that selects among multiple reasoning traces. We additionally include two routing-focused baselines that operate over the same four pipelines $\{\pi_{00}, \pi_{01}, \pi_{10}, \pi_{11}\}$. **Joint-Router** is a single 4-way router that directly predicts one of the four pipelines with a multi-class classifier; for a fair comparison, it uses the same lightweight probe features. **Threshold-Only** replaces each MLP head with a single fitted threshold on its scalar probe (NQC or NLL), trained on the same 500 HotpotQA routing examples; its results are reported in Section 5.4. **Router-SC** executes all four pipelines for every query and uses a fixed judge model (Q2.5-7B) to select the best answer by quality, ignoring any budget constraints. More details about baselines are provided in Appendix A.2.

4.6 Metrics

We follow each benchmark's evaluation protocol. For multi-hop QA (HotpotQA (Yang et al., 2018) and 2WIKI (Ho et al., 2020)), we report **exact match (EM)** and **token-level F1** between the normalized prediction and reference answers, following the SQuAD-style QA evaluation (Rajpurkar et al., 2016). We report **ROUGE-L** (Lin, 2004) to capture partial overlap for generative outputs. For financial QA (FiQA (Maia et al., 2018)), we report **F1** and **ROUGE-L**. For classification-style tasks (StrategyQA (Geva et al., 2021), GSM8K (Cobbe et al., 2021), and MedQA (Jin et al., 2020)), we report **accuracy (AC)** via exact match on the nor-

malized final answer. All quality metrics are reported as percentages. For efficiency, we report average **end-to-end latency** (ms) and **token usage** per query, which includes the cost of probing.

4.7 Router Training

We train the router on 500 examples sampled from the HotpotQA training split and evaluate it on the other five benchmarks without retraining router parameters, making the main evaluation a strict zero-shot cross-domain transfer test. We choose HotpotQA because it contains a broad mix of knowledge- and reasoning-heavy questions, yielding sufficiently rich oracle supervision for both routing heads (Table 7). We derive oracle labels by maximizing the cost-aware utility over the four pipelines and train the two heads on the corresponding probes (NQC and NLL).

5 Experimental Results

5.1 Main Results

Table 1 reports answer quality across six benchmarks. The router is trained only on 500 HotpotQA examples and transferred to the other benchmarks without retraining, so these results also evaluate zero-shot cross-domain robustness. Overall, X-ROUTER is competitive against strong adaptive baselines while remaining cost-aware (Section 3.1). We include the fixed pipelines π_{00}–π_{11} in Table 7 (Appendix A.4) to contextualize routing gains. Oracle-U is the *theoretical upper bound* under our cost-aware objective: it selects the utility-optimal pipeline per query. Our MLP router is trained to approximate these oracle decisions. Router-SC is an *engineering upper bound* that ignores budget constraints by running all pipelines and selecting the best answer via a judge model; therefore, its answer quality can exceed Oracle-U even though Oracle-U is optimal for utility.

Takeaway. The best pipeline varies across datasets (Table 7), confirming that "always retrieve" or "always think" strategies are not robust. X-ROUTER also tends to surpass Joint-Router on all datasets, showing that decoupling knowledge and reasoning decisions improves routing reliability.

5.2 Efficiency

We report end-to-end latency and token usage under the full workflow in Section 3.5. For routing-based methods, every reported cost already includes both probes: (i) a probing retrieval to compute $\mathrm{NQC}(q)$ and (ii) a single no-CoT decoding pass to estimate $\mathrm{NLL}(q)$. Figure 3 summarizes efficiency on three representative datasets (2WIKI, GSM8K, and FiQA), comparing seven baselines (including Router-SC), X-ROUTER, and Oracle-U. We additionally provide efficiency–quality trade-off curves for all datasets in Figures 6–7.

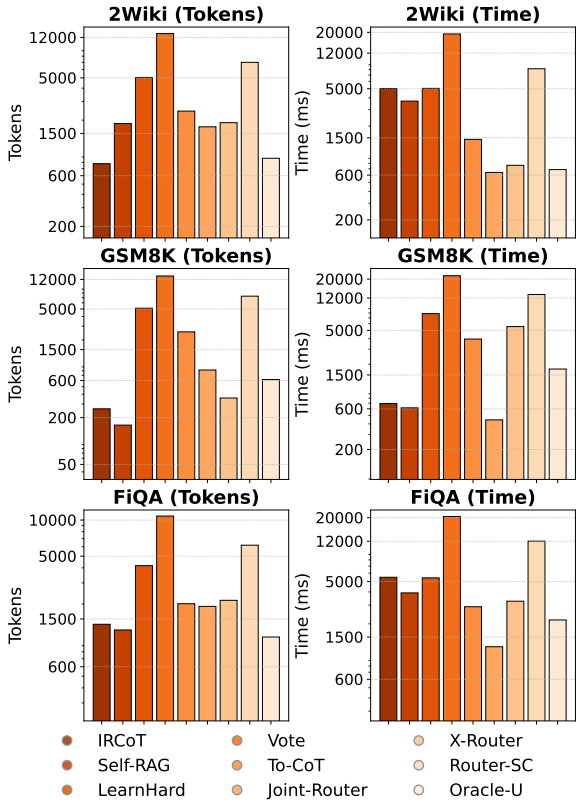

Figure 3: Efficiency comparison on three datasets. Bars show average token usage and end-to-end latency for seven baselines.

Discussion. We observe noticeable differences in efficiency profiles across methods. **Vote** and **Router-SC** are consistently the most expensive, since they execute multiple pipelines and/or sample multiple candidate paths per query. While these strategies can be effective, their high latency (e.g., $\sim$3$\times$ that of X-ROUTER on GSM8K) limits practical deployment. In contrast, X-ROUTER achieves a stronger efficiency–quality trade-off than adaptive baselines such as **Self-RAG**. Self-RAG incurs substantial overhead from self-reflection tokens and iterative generation, whereas X-ROUTER relies on lightweight probes to make a single, upfront routing decision. Importantly, the average E2E latency of X-ROUTER is workload-dependent rather than a universal constant: queries routed to $a_C{=}1$ pay for both the draft pass and the final CoT pass, while queries with $a_C{=}0$ reuse the draft directly. This explains why X-ROUTER can remain token-efficient yet still exhibit higher latency on CoT-heavy work-

Dataset	π_{00}	π_{01}	π_{10}	π_{11}	π_{11} rate
2WIKI	304	38	122	36	7.2%
GSM8K	286	130	25	59	11.8%
FiQA	78	206	88	128	25.6%

Table 2: Oracle workload mix on representative datasets. The most expensive quadrant (π_{11}) remains a minority, leaving room for routing gains in cheaper quadrants.

loads such as GSM8K. Overall, X-ROUTER reduces token usage by up to 86% and latency by up to 84% while improving answer quality over strong baselines. This enables X-ROUTER to match the low latency of the cheapest fixed pipeline (π_{00}), while selectively allocating expensive mechanisms (RAG and/or CoT) only when they are most beneficial. A detailed comparison against the four atomic fixed pipelines (π_{00}–π_{11}) is reported in Appendix A.4, Table 7.

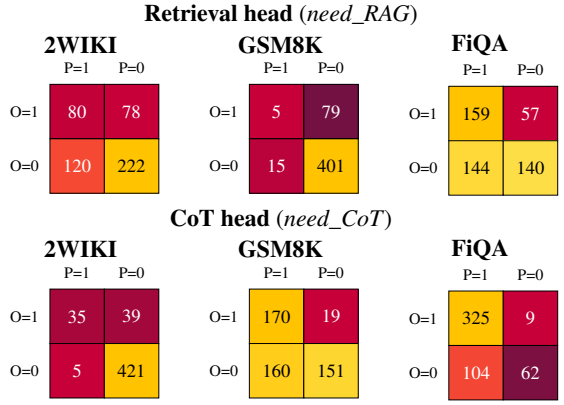

Figure 4: Router confusion matrices between oracle labels (O) and router predictions (P), where positives are label 1. Cell color (viridis) reflects the number of examples (darker = fewer; lighter = more).

5.3 Router Diagnostics

We compare router predictions to oracle-derived labels (*need_RAG*, *need_CoT*); Figure 4 reports TP/FN/FP/TN on three datasets.

Workload mix. Table 2 clarifies why routing still helps even when retrieval and reasoning are complementary. Across all three representative datasets, the expensive π_{11} quadrant remains a minority under oracle utility, so the main upside comes from avoiding unnecessary retrieval, unnecessary CoT, or both on the majority of queries.

Retrieval head. On GSM8K, the retrieval head is strongly conservative (TP/FN=5/79; recall ≈0.06), consistent with retrieval being rarely utility-optimal for self-contained math. On 2WIKI and FiQA, where evidence helps more often, it favors recall over precision (2WIKI: P≈0.40, R≈0.51; FiQA: P≈0.52, R≈0.74). Most errors lie near the utility boundary, so false positives mainly increase cost rather than sharply harming quality.

CoT head. The CoT head is precise but conservative on 2WIKI (P≈0.88, R≈0.47), suggesting many multi-hop questions become answerable once retrieval succeeds, without lengthy reasoning traces. On GSM8K and FiQA it prioritizes recall (R≈0.90/0.97), aligning with CoT being frequently utility-optimal; the dominant failure mode is false positives that inflate token usage.

Head-specific error types. The harder head depends on the workload. On retrieval-sensitive datasets such as 2WIKI and FiQA, the retrieval head makes more total mistakes than the CoT head (198 vs. 44 on 2WIKI; 201 vs. 113 on FiQA), indicating that deciding whether external evidence is worth the cost is the sharper boundary. On GSM8K, the pattern flips (94 vs. 179), because the main uncertainty is how often extra reasoning budget should be allocated. Error impacts are also asymmetric: retrieval false negatives directly harm answer quality by suppressing needed evidence, whereas retrieval false positives mostly add token and retrieval cost; conversely, CoT false negatives miss bridge or arithmetic reasoning, while CoT false positives primarily increase latency and sometimes induce verbose answers.

Dataset	X-ROUTER	Joint	Δ	p
2WIKI (F1)	42.3 [37.8,46.1]	38.1 [34.0,42.1]	+4.21 [+1.52,+6.34]	$<10^{-3}$
GSM8K (AC)	62.8 [58.6,67.2]	32.4 [28.2,36.6]	+30.40 [+25.20,+35.60]	$<10^{-3}$
FiQA (F1)	31.5 [30.7,32.4]	22.3 [21.3,23.3]	+9.24 [+8.24,+10.28]	$<10^{-3}$

Table 3: Paired bootstrap comparison against Joint-Router on representative datasets.

5.4 Robustness and Threshold Baselines

To quantify uncertainty under the same 500-example evaluation protocol, we run paired bootstrap resampling over query-level outputs (B=10,000; one-sided H_1: X-ROUTER > Joint-Router). Table 3 shows consistently positive deltas with $p < 10^{-3}$ on representative datasets, supporting the robustness of the decoupled design under cross-domain transfer. Appendix A.5 provides an expanded view with explicit metric and CI columns, together with a short interpretation.

We also compare our method against **Threshold-Only**, which replaces each MLP head with a single fitted threshold on NQC/NLL. Table 4 shows that

Dataset	Threshold-Only Quality	Tok.	Time	X-ROUTER Quality	Tok.	Time
2WIKI	39.1	1904	743	42.3	1893	763
GSM8K	38.4	2427	1106	62.8	357	5542
FiQA	21.8	1928	1210	31.5	2145	3266

Table 4: Threshold-only baseline on representative datasets. Quality is F1 for 2WIKI/FiQA and accuracy for GSM8K; all costs are end-to-end.

the simple thresholding is competitive on 2WIKI, but degrades more sharply on GSM8K and FiQA under domain shift. This pattern suggests that a single threshold is too rigid to capture the non-linear utility boundary, especially for CoT-heavy or mixed-regime workloads.

5.5 Ablation Study

Table 5 ablates the decoupled design by removing one head and forcing the other decision to be always on. Across datasets, the two decisions are complementary: routing only one axis degrades performance in a dataset-dependent manner.

Method	2Wiki (F1)	GSM8K (AC)	FiQA (F1)
Only CoT head	37.6	57.6	30.6
Only RAG head	32.7	60.1	31.3
X-ROUTER	**42.3**	**62.8**	**31.5**

Table 5: Ablation of the two-head router on three datasets. Numbers are quality (F1/AC) in %.

Interpretation. On 2WIKI, keeping retrieval always on but routing CoT (*Only CoT head*) is substantially stronger than routing retrieval while always using CoT (*Only RAG head*), consistent with evidence insufficiency being the dominant bottleneck. On GSM8K, the opposite trend holds: routing retrieval while always using CoT is closer to the full model; CoT is broadly useful while retrieval is mostly unnecessary. On FiQA, both matter, and the full model best balances the trade-off.

5.6 Model Swap

Table 6 reports X-ROUTER answer quality under model swaps while keeping the retrieval, cost weights, and evaluation fixed. We provide a full model-swap summary (including Oracle-U and cost) in Table 11 (Appendix A.7).

Discussion. Model swaps change the answer quality in a non-monotonic way under fixed cost weights. For example, DS-R1-14B yields the strongest accuracy on GSM8K, while Q2.5-32B provides the strongest 2WIKI scores. Table 11 further shows that stronger backbones can shift the oracle toward cheaper pipelines under the same utility, highlighting the need to calibrate cost weights jointly with the deployment stack.

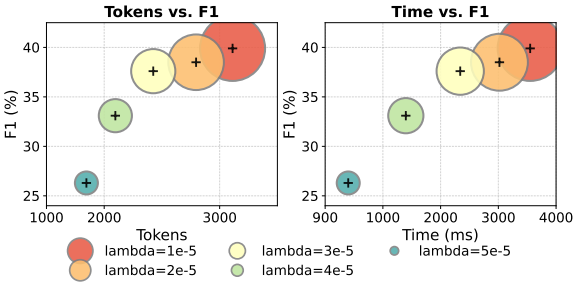

Figure 5: Hyperparameter sensitivity on 2WIKI. Five λ_{tok} settings (1×10^{-5}–5×10^{-5}) show a gradual quality–cost trade-off under the cost-aware utility.

Dataset	Metric	Q2.5-7B	Q3-8B	DS-R1-14B	Q2.5-32B
2WIKI	F1	42.3	51.5	54.2	**54.4**
	EM	35.2	43.0	40.8	**46.2**
	ROUGE-L	42.5	51.5	54.1	**54.4**
GSM8K	AC	62.8	60.0	**92.2**	66.4
FiQA	F1	**31.5**	27.7	22.5	29.7
	ROUGE-L	**16.9**	14.6	12.4	15.8

Table 6: Model swap results (X-ROUTER). All numbers are in % for answer quality.

5.7 Hyperparameter Sensitivity

We sweep the token cost weight λ_{tok} in the utility (Section 3.1) on 2WIKI while keeping λ_{time} fixed (Appendix C). Figure 5 shows the trade-off: increasing λ_{tok} steadily reduces token usage and latency, while degrading F1 under the cost-aware objective. For example, λ_{tok} from 1×10^{-5} to 5×10^{-5} reduces tokens from 1893 to 983 and latency from 762 ms to 430 ms, at the cost of a 16.0 F1 drop.

6 Conclusions and Future Works

We proposed X-ROUTER, a cost-aware dual-axis router that jointly routes retrieval and CoT by utility supervision over four pipelines, using lightweight probes (NQC and draft NLL) at inference time. Across six QA benchmarks, X-ROUTER cuts token/latency costs while matching or improving answer quality. Future work includes finer-grained budget control, robustness under distribution shift, and extensions to multimodal tasks.

Limitations

Our approach relies on two lightweight signals that may not be uniformly available across deployment settings: retriever scores (for NQC) and token-level

likelihoods (for NLL). When only black-box text outputs are accessible, alternative uncertainty proxies may be required. In addition, our oracle supervision requires offline execution of all four pipelines to compute cost-aware utilities; while we use this only for a limited profiling set, scaling the labeling procedure to larger corpora or richer pipeline families would increase compute cost. Furthermore, our training process uses oracle labels for both retrieval and reasoning, whereas inference relies on predicted retrieval states to feed the reasoning router. This discrepancy could introduce cascade errors, although our ablation studies suggest the decoupled routers remain robust. Finally, the utility trade-offs depend on the deployment environment (e.g., latency characteristics and pricing). Although we use fixed weights in our main experiments, transferring the system to a new setting may require calibrating these weights to match the desired budget–quality regime.

Additionally, our "probe-then-route" design introduces a sequential overhead for queries routed to the most complex pipeline (π_{11}). Since the decision to enable CoT depends on the NLL of a direct-answer pass, these queries incur the latency of both the draft pass and the subsequent CoT generation. As a result, the average end-to-end latency is workload-dependent: if many queries in a deployment are routed to CoT, the extra sequential stage becomes more visible; if most queries exit at a_C=0, the draft is reused and the overhead is much smaller. Table 7 shows that the direct pass is still substantially faster than CoT generation (e.g., $\sim$360ms vs. $\sim$2300ms), so this overhead is usually a minority term rather than the dominant cost, but it should not be interpreted as a universal constant across workloads.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7752–7767, Bangkok, Thailand. Association for Computational Linguistics.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. In *Advances in Neural Information Processing Systems*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. 2024. Learning how hard to think: Input-adaptive allocation of LM computation. *arXiv preprint arXiv:2410.04707*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z.F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 69 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Gongfan Fang, Yifan Zhang, and Yizhe Wang. 2025. Thinkless: LLM learns when to think. *arXiv preprint arXiv:2505.13379*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*. Accepted by COLING 2020.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Empirical Methods in Natural Language Processing*, pages 7969–7992.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Saurav Kadavath, and Jared Kaplan. 2024. Measuring faithfulness in chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8792–8808.

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 *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Guosheng Liang, Longguang Zhong, Ziyi Yang, and Xiaojun Quan. 2025. Thinkswitcher: When to think hard, when to think fast. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

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

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. WWW'18 open challenge: Financial opinion mining and question answering. In *Companion of the The Web Conference 2018*.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Chuan Meng, Negar Arabzadeh, Mohammad Aliannejadi, and Maarten de Rijke. 2023. Query performance prediction: From ad-hoc to conversational search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2583–2593.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, volume 35, pages 17456–17472.

Tal Shnitzer, Anthony Ou, Ankit Deoras, Prateek Bhatia, Xiang Liu, and Ravi Kumar. 2024. Large language model routing with benchmark datasets. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Anna Shtok, Oren Kurland, David Carmel, Fiana Raiber, and Gad Markovits. 2012. Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems*, 30(2):11:1–11:35.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *International Conference on Learning Representations*.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the information needs of large language models. In *Annual Meeting of the Association for Computational Linguistics*.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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 10014–10029.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,

Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 27 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Wenhao Yu, Hongming Zhang, Xiaoman Wang, Joseph Tan, Dur Chen, Zaitang Chu, Bo Li, and Dian Yu. 2024. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *The Twelfth International Conference on Learning Representations*.

Xiaoyun Zhang, Jingqing Ruan, Xing Ma, Yawen Zhu, Haodong Zhao, Hao Li, Jiansong Chen, Ke Zeng, and Xunliang Cai. 2025. When to continue thinking: Adaptive thinking mode switching for efficient reasoning. *arXiv preprint arXiv:2505.15400*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Zihan Zhang, Meng Fang, and Ling Chen. 2024. Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2024*.

A Additional Experiment Results

A.1 Datasets

We include dataset details referenced in Section 4.

2WikiMultiHopQA (Ho et al., 2020). A Wikipedia-based multi-hop QA dataset where questions typically require linking evidence across two different articles, enabling explicit cross-entity reasoning chains.

HotpotQA (Yang et al., 2018). A multi-hop reasoning benchmark built from Wikipedia; each question requires integrating information from multiple paragraphs to infer the correct answer, and includes annotated supporting facts.

StrategyQA (Geva et al., 2021). A commonsense yes/no QA dataset designed to require implicit, multi-step reasoning beyond surface-level retrieval, often relying on everyday knowledge and pragmatic inference.

GSM8K (Cobbe et al., 2021). Grade-school math word problems that demand multi-step arithmetic and symbolic reasoning, serving as a representative setting where CoT-style reasoning is often beneficial.

FiQA (Maia et al., 2018). A financial-domain QA dataset that requires understanding finance-related text (e.g., reports and discussions) and handling domain-specific terminology and context.

MedQA (Jin et al., 2020). A medical-domain multiple-choice QA dataset with highly specialized questions, aimed at testing a model's accuracy and knowledge in a specific expert domain.

A.2 Additional Baseline Details

This appendix provides concise implementation details for the baselines in Section 4. Unless otherwise noted, all baselines use the same retriever/corpus and the same backbone generator as X-ROUTER to isolate routing effects.

- **IRCoT** (Trivedi et al., 2023). An interleaved retrieve–reason framework for multi-hop QA. The model alternates between generating intermediate reasoning steps and issuing retrieval queries conditioned on the partial chain-of-thought; retrieved passages are appended to the context and the process repeats for a fixed hop budget (or until the model outputs a final answer).
- **Self-RAG** (Asai et al., 2024). A self-reflective RAG procedure that performs on-demand retrieval and critique during generation. At inference, the model emits lightweight reflection signals to (i) trigger retrieval when needed and (ii) assess whether retrieved evidence supports the current draft, then continues generation conditioned on the accepted evidence.
- **To-CoT-or-Not** (Sprague et al., 2025). A heuristic CoT gating baseline motivated by the observation that CoT gains concentrate on math/symbolic problems. We enable CoT only when the

query (or a short direct draft) exhibits symbolic-execution cues (e.g., explicit arithmetic patterns such as "="), and otherwise answer with direct decoding.

- **LearnHard** (Damani et al., 2024). An input-adaptive test-time compute allocation method: it predicts the marginal benefit of additional decoding computation for each query and allocates compute accordingly, instantiated as (i) adaptive best-of-k sampling and (ii) routing between a cheaper decoder and a more expensive but more accurate procedure.
- **Vote (Self-Consistency)** (Wang et al., 2023). A sampling-based test-time compute baseline inspired by self-consistency: we use the same backbone model to generate multiple stochastic candidates (CoT-enabled) and select the final answer by majority vote over normalized answers, improving robustness at the cost of proportional token/latency overhead.
- **Threshold-Only**. A lightweight variant of X-ROUTER that replaces each MLP head with a single fitted scalar threshold on its probe (NQC for retrieval, NLL for CoT). It uses the same 500 HotpotQA routing examples as X-ROUTER but removes the non-linear mapping capacity of the MLP heads.

A.3 Experimental Environment

All experiments are run on an Ubuntu 22.04 LTS server with 8× NVIDIA H100 80GB GPUs, an Intel Xeon Platinum 8558P CPU @ 2.70GHz, and 1 TB RAM.

A.4 Pipelines and Routers

Table 7 reports the quality and cost of the four fixed pipelines π_{00}–π_{11}, Joint-Router (Router2), X-ROUTER, and Oracle-U on the evaluation subsets (Section 4). The results highlight substantial heterogeneity across datasets: multi-hop QA generally benefits from retrieval (π_{10}), self-contained math problems benefit from CoT without retrieval (π_{01}), and some domain QA tasks remain strong under direct answering (π_{00}). Oracle-U further shows that the utility-optimal pipeline selection can be markedly better than any single fixed strategy.

Implications. The table makes the routing problem concrete: the four fixed pipelines can be strongly suboptimal depending on whether a dataset's bottleneck is missing evidence (π_{10}/π_{11}) or multi-step reasoning (π_{01}/π_{11}). X-ROUTER is often close to the best fixed pipeline while preserving the ability to mix strategies (e.g., it matches or slightly exceeds π_{10} on HotpotQA/2WIKI and exceeds π_{01} on GSM8K), indicating that the probes recover many near-oracle choices without executing all pipelines online. In contrast, Joint-Router can reduce cost on some datasets but is noticeably less reliable, especially on GSM8K where it collapses to low-utility decisions and fails to capture the CoT-without-retrieval regime that dominates quality gains. Finally, Oracle-U shows substantial headroom under the same cost-aware objective (e.g., +11.4 F1 on HotpotQA and +17.4 AC on MedQA over X-ROUTER), suggesting that improved calibration and slightly richer but still lightweight features could further close the gap without heavy multi-sampling overhead.

A.5 Robustness Details

Section 5.4 reports the compact robustness table used in the main paper. Table 9 expands the same comparison by separating the metric label from the confidence-interval columns for easier lookup.

Interpretation. The expanded appendix view makes two points easier to read off directly. First, all three confidence intervals for Δ stay strictly above zero, so the advantage of the decoupled router is not driven by a few favorable resamples. Second, the gain is modest but stable on 2WIKI, while it becomes much larger on GSM8K and FiQA, where the routing boundary is more sensitive to CoT allocation and mixed utility trade-offs.

A.6 Threshold-Only Baseline Details

Table 10 reports the threshold-only baseline introduced in Section 5.4. Compared with X-ROUTER, fixed thresholds are reasonably competitive on 2WIKI, but they transfer poorly to workloads where the utility boundary is dominated by CoT-heavy or mixed-regime decisions.

A.7 Model Swap Summary

Table 6 in the main paper reports representative X-ROUTER quality under model swaps. Table 11 provides a full summary across all benchmarks, reporting both quality and cost (tokens and time).

A.8 Full Efficiency Plots

We visualize the full cost–quality trade-offs (token usage vs. F1/AC, and latency vs. F1/AC) for all six datasets. Each subplot includes the seven baselines

Table 7: Quality and cost of fixed pipelines, routers, and Oracle-U (quality in %; cost is raw counts per query).

Dataset	Metric	π_{00}	π_{01}	π_{10}	π_{11}	Joint-Router	X-ROUTER	Oracle-U
HotpotQA	F1	24.3	18.0	52.7	32.9	43.8	52.9	64.3
	EM	16.4	9.8	37.6	20.4	32.0	37.8	49.4
	ROUGE-L	24.2	17.8	52.6	32.9	43.7	52.8	64.2
	Tokens	127.2	285.9	2683.7	2853.2	1583.9	2285.8	1433.4
	Time (ms)	360.1	2325.9	736.0	2689.9	584.8	765.3	796.2
2WIKI	F1	26.8	18.3	41.0	32.7	38.1	42.3	58.8
	EM	23.4	10.6	34.0	20.0	32.4	35.2	50.0
	ROUGE-L	26.7	18.3	41.0	32.7	38.0	42.5	58.8
	Tokens	124.6	275.5	2439.7	2612.4	1730.6	1893.0	875.8
	Time (ms)	338.1	2208.9	802.0	3023.9	640.4	762.9	686.7
FiQA	F1	13.3	26.0	22.4	31.0	22.3	31.5	33.0
	ROUGE-L	8.5	13.2	14.1	15.3	14.0	16.9	18.2
	Tokens	152.0	343.6	2002.4	2623.4	1910.0	2145.4	1061.4
	Time (ms)	942.1	4314.6	1216.6	3495.6	1216.3	3265.6	2174.5
StrategyQA	AC	61.0	58.6	56.2	54.6	58.2	61.4	74.4
	Tokens	123.4	319.5	2569.1	2700.2	1199.4	172.1	305.2
	Time (ms)	929.9	4798.4	1818.4	6205.8	1401.4	1967.1	937.4
GSM8K	AC	33.4	61.1	27.8	57.6	32.4	62.8	76.2
	Tokens	161.7	445.4	2460.2	2661.9	819.6	357.4	617.6
	Time (ms)	355.2	4941.4	711.7	4336.7	445.3	5541.7	1759.5
MedQA	AC	56.6	27.4	54.8	27.8	50.2	58.8	76.2
	Tokens	321.7	553.2	2984.4	3216.4	1790.5	322.7	606.7
	Time (ms)	383.5	4931.0	878.0	4983.5	641.0	398.6	784.0

Table 8: Oracle label priors (positives) under the default utility weights.

Dataset	*need_RAG*	*need_CoT*
HotpotQA	250/500	87/500
2WIKI	158/500	74/500
FiQA	216/500	334/500
StrategyQA	34/500	42/500
GSM8K	84/500	189/500
MedQA	49/500	54/500

(including Router-SC), X-ROUTER, and Oracle-U; the four π pipelines are omitted to keep the plots readable.

B Case Study and Error Analysis

B.1 Case Study

We present three illustrative HotpotQA examples that align with our motivation in Section 1: (i) sometimes NORAG+NOCOT is *better* than enabling retrieval or CoT, (ii) sometimes retrieval is *necessary* while CoT can still hurt (or is simply inefficient), and (iii) sometimes *both* retrieval and CoT are required—either one alone fails. Together, these cases highlight why the two decisions should be decoupled: retrieval addresses *missing knowledge*, while CoT addresses *reasoning difficulty* given evidence.

B.2 Error Analysis

We observe four recurring failure patterns in our experiments. (i) **Over-reasoning:** enabling CoT for evidence-sufficient questions can introduce unnecessary tokens and, in some cases, hurt answer quality (Figure 8). (ii) **Under-retrieval for knowledge-missing queries:** disabling retrieval when the relevant entity is absent from parametric memory leads to hallucinated answers, especially on multi-hop QA (Figure 9). (iii) **Missing bridge reasoning:** retrieval-only pipelines can surface correct evidence but still answer the wrong variable (e.g., returning an actor instead of the character), requiring explicit reasoning to complete the bridge (Figure 10). (iv) **Imbalanced oracle labels:** when oracle decisions are highly skewed on a dataset, the router may become conservative and collapse to a dominant class, which we diagnose via confusion matrices in Figure 4.

C Hyperparameters

This section summarizes hyperparameter settings and sweep details. Unless otherwise specified, our main experiments use $\lambda_{\text{tok}} = \lambda_{\text{time}} = 1\times10^{-5}$, where time is measured in milliseconds. For the utility term Q (Section 3.1), we use F1 for generative QA tasks (HotpotQA/2WIKI/FiQA)

Table 9: Expanded paired-bootstrap comparison corresponding to Table 3. We resample query-level outputs with replacement for B=10,000 trials and report one-sided p-values for H_1: X-ROUTER > Joint-Router.

Dataset	Metric	X-ROUTER	95% CI	Joint-Router	95% CI	Δ (95% CI)	p
2WIKI	F1	42.3	[37.8, 46.1]	38.1	[34.0, 42.1]	+4.21 [+1.52, +6.34]	$< 10^{-3}$
GSM8K	AC	62.8	[58.6, 67.2]	32.4	[28.2, 36.6]	+30.40 [+25.20, +35.60]	$< 10^{-3}$
FiQA	F1	31.5	[30.7, 32.4]	22.3	[21.3, 23.3]	+9.24 [+8.24, +10.28]	$< 10^{-3}$

Table 10: Threshold-only baseline compared with X-ROUTER on representative datasets. Quality numbers are in %; cost metrics are raw counts per query.

Dataset	Metric	Threshold-Only			X-ROUTER		
		Quality	Tokens	Time (ms)	Quality	Tokens	Time (ms)
2WIKI	F1	39.1	1904	743	42.3	1893	763
	EM	32.4	1904	743	35.2	1893	763
	ROUGE-L	39.1	1904	743	42.5	1893	763
GSM8K	AC	38.4	2427	1106	62.8	357	5542
FiQA	F1	21.8	1928	1210	31.5	2145	3266
	ROUGE-L	13.7	1928	1210	16.9	2145	3266

and accuracy for classification-style tasks (StrategyQA/GSM8K/MedQA).

We sweep the token cost weight λ_{tok} in the utility function (Section 3.1) on 2WikiMultiHopQA in the range $[1\times10^{-5}, 5\times10^{-5}]$, while keeping λ_{time} fixed. Sensitivity is reported in Section 5.7.

C.1 Implementation Details

The official code repository is available at `https://github.com/718shame/X-router`. We train two scikit-learn `MLPClassifier` routers (one per head) on oracle labels derived from offline profiling. Unless otherwise specified, we use `hidden_layer_sizes=(16, 8)`, `activation=relu`, `solver=adam`, `alpha=1e-4`, `max_iter=500`, and `random_state=42`, and apply `StandardScaler` to the probe features before training.

C.2 Retrieval Configuration

We implement a hybrid retrieval pipeline combining dense and sparse methods. For document processing, we segment the corpus into passages with a `chunk_size` of 256 and a `chunk_overlap` of 100 tokens. During the retrieval phase, we fetch an initial candidate pool consisting of the top-20 results from the dense retriever (`dense_top_k=20`) and the top-40 results from the sparse retriever (`sparse_top_k=40`).

From this combined pool, we select the `final_top_k=8` passages using Maximum Marginal Relevance (MMR) reranking with a diversity parameter `mmr_lambda=0.5`. The concatenated context provided to the model is capped at `max_context_tokens=1800`. For the sparse retriever (BM25), we use standard hyperparameters $k_1 = 1.5$ and $b = 0.75$.

C.3 Prompt Templates

Table 11: Model swap summary. Quality numbers are in %; cost metrics are raw counts per query.

Dataset	Metric	Q2.5-7B		Q3-8B		DS-R1-14B		Q2.5-32B	
		X-ROUTER	Oracle-U	X-ROUTER	Oracle-U	X-ROUTER	Oracle-U	X-ROUTER	Oracle-U
HotpotQA	F1	52.9	64.3	47.2	63.2	56.8	60.8	59.7	72.5
	EM	37.8	49.4	29.2	45.8	38.4	42.6	41.0	55.8
	ROUGE-L	52.8	64.2	47.1	63.1	56.7	60.6	59.6	72.3
	Tokens	2285.8	1433.4	2670.8	1596.5	3280.3	1537.2	2684.4	1508.4
	Time (ms)	765.3	796.2	1650.1	2497.2	23134.3	8835.5	1580.4	1398.9
2WIKI	F1	42.3	58.8	51.5	50.0	54.2	51.6	54.4	69.2
	EM	35.2	50.0	43.0	37.2	40.8	39.4	46.2	61.0
	ROUGE-L	42.5	58.8	51.5	49.9	54.1	51.6	54.4	69.2
	Tokens	1893.0	875.8	2584.7	676.6	3032.4	1323.2	2582.4	1007.4
	Time (ms)	762.9	686.7	13148.4	1538.9	18104.1	5222.0	4921.6	995.9
FiQA	F1	31.5	33.0	27.7	21.1	22.5	20.5	29.7	28.5
	ROUGE-L	16.9	18.2	14.6	12.5	12.4	12.0	15.8	15.5
	Tokens	2145.4	1061.4	2137.7	427.0	2687.1	1199.5	2141.7	679.7
	Time (ms)	3265.6	2174.5	14112.7	3262.9	16034.7	7323.9	5719.6	2942.8
StrategyQA	AC	61.4	74.4	64.8	74.6	75.2	80.0	73.8	86.6
	Tokens	172.1	305.2	122.3	358.4	3118.1	590.9	127.7	302.4
	Time (ms)	1967.1	937.4	1879.8	2028.5	20242.8	5998.8	550.7	861.8
GSM8K	AC	62.8	76.2	60.0	82.8	92.2	95.4	66.4	83.0
	Tokens	357.4	617.6	2631.2	865.0	2882.8	602.1	355.3	392.6
	Time (ms)	5541.7	1759.5	13747.2	4307.7	15356.5	5693.8	3635.9	1303.1
MedQA	AC	58.8	76.2	62.2	80.2	74.8	83.6	71.6	84.6
	Tokens	322.7	606.7	306.6	589.9	1599.1	1270.3	323.3	477.7
	Time (ms)	398.6	784.0	1026.0	2120.5	20108.4	13289.6	464.1	937.2

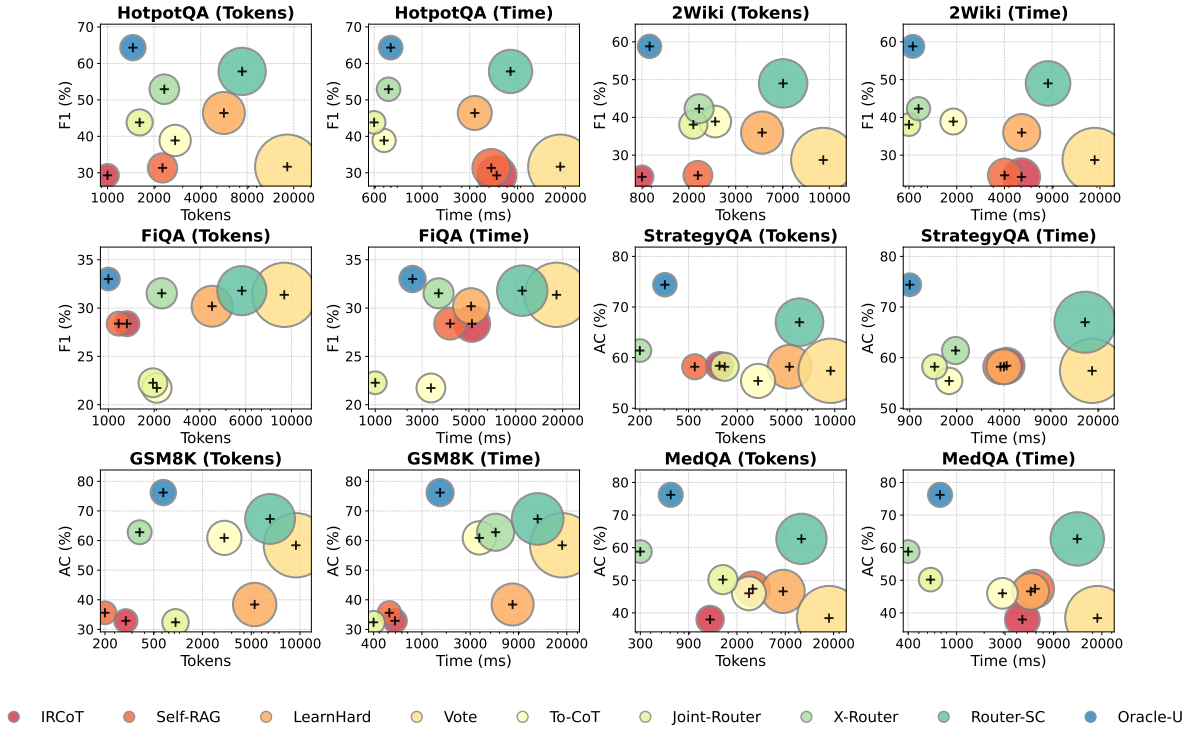

Figure 6: Efficiency–quality scatter plots on six datasets.

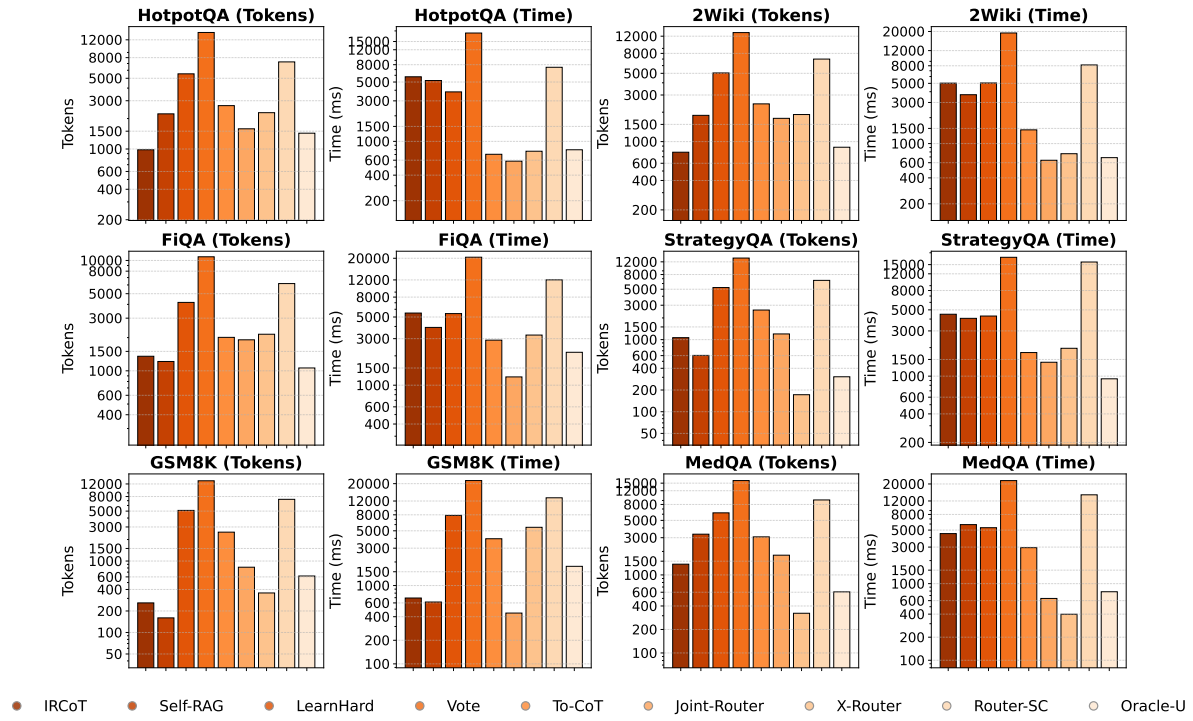

Figure 7: Efficiency bar plots (tokens and latency) on six datasets.

Case 1: π_{00} (NoRAG, NoCoT)

Question

The Twelfth United States Army Group commander was the first chairman of what?

Retrieved Context

(No retrieval performed)

Answer

π_{00} **(Direct):** Joint Chiefs of Staff (✓Correct)
π_{01} **(CoT):** [Insufficient information to determine] (✗ Fail)

Figure 8: **Knowledge-Rich, Reasoning-Easy.** The model possesses the parametric knowledge (General Omar Bradley) to answer directly. Forcing CoT without retrieval causes the model to doubt its internal knowledge and refuse to answer, illustrating that *reasoning without context* can be detrimental for factoid queries.

Case 2: π_{10} (RAG only)

Question

Barnstable County Hospital was the location of the autopsy of the son of President John F. Kennedy and Jacqueline Kennedy, and the younger brother of who?

Retrieved Context

[Doc 2] ...John Fitzgerald Kennedy Jr... was a son of President John F. Kennedy and First Lady Jacqueline Kennedy, and a younger brother of former Ambassador to Japan **Caroline Kennedy**...

Answer

π_{00} **(NoRAG):** John F. Kennedy Jr. **(✗ Incorrect Entity)**
π_{10} **(RAG+Direct):** Caroline Kennedy **(✓Correct)**
π_{11} **(RAG+CoT):** Barnstable County Hospital was the location of the autopsy of the son of President John F. Kennedy and Jacqueline Kennedy, and the younger brother of Caroline Kennedy. **(Contains correct entity but verbose)**

Figure 9: **Knowledge-Poor, Reasoning-Easy.** The model lacks specific details about the autopsy location and family relations, leading to a hallucination or entity swap in π_{00}. Retrieval (π_{10}) provides the exact relation. However, CoT (π_{11}) is unnecessary and often copies the context verbatim, which wastes tokens and can be penalized by overlap-based metrics even when the correct entity appears in the output.

Case 3: π_{11} (RAG + CoT)

Question

Domestic Disturbance stars an actor who voiced what character in "Monsters, Inc."?

Retrieved Context

[Doc 1] ...Domestic Disturbance... starring John Travolta, Vince Vaughn, Teri Polo, **Steve Buscemi**...
[Doc 1] ...Buscemi provides the voice of **Randall Boggs** in the "Monsters, Inc." franchise.

Answer

π_{10} **(RAG+Direct):** Steve Buscemi **(✗ Premature Answer)**
π_{11} **(RAG+CoT):** The actor who starred in Domestic Disturbance is Steve Buscemi. Steve Buscemi voiced the character Randall Boggs in Monsters, Inc. So the answer is Randall Boggs. **(✓Correct)**

Figure 10: **Knowledge-Poor, Reasoning-Hard.** This multi-hop query requires bridging two facts. Even with the correct documents, the Direct pipeline (π_{10}) latches onto the first entity found (the actor). CoT (π_{11}) is strictly necessary to perform the multi-step deduction (Movie $\rightarrow$ Actor $\rightarrow$ Character).

Default Prompt

```
You are a helpful assistant.
Answer the provided question given the context information. Be concise and precise! Provide only the essential information directly answering the question. Avoid unnecessary elaboration or background information.
Context information is below.
---------------------
{context_str}
---------------------
Question: {query_str}

Provide your answer in the format:
Answer: [your answer here]
```

CoT Prompt

```
You are a helpful assistant.
Answer the provided question given the context step-by-step.
Context information is below.
---------------------
{context_str}
---------------------
Question: {query_str}

Please think through this step-by-step and then provide your final answer in the format:
Answer: [your answer here]
```

Figure 11: Prompt templates used for the four pipelines (Section 3.1). `{context_str}` is empty for NoRAG pipelines.