BalanceRAG: Joint Risk Calibration for Cascaded Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

Large language models (LLMs) can enhance factuality via retrieval-augmented generation (RAG), but applying RAG to every query is unnecessary when the model-only answer is reliable. This motivates cascaded RAG: each query is first handled by an LLM-only branch, escalated to a RAG fallback only if the primary branch is uncertain, and abstained from when neither branch is sufficiently trustworthy. However, calibrating such cascades stage by stage may be conservative, since the final utility depends on joint uncertainty thresholding of LLM-only and RAG. In this work, we develop **BalanceRAG** to certify threshold pairs at a target risk level. Given uncertainty scores from the two branches, BalanceRAG frames each threshold pair as an operating point on a two-dimensional lattice and identifies safe operating points using sequential graphical testing. This enables risk-adaptive threshold calibration, controlling the system-level error rate among accepted points, while retaining more examples. Furthermore, BalanceRAG extends to multi-risk calibration, allowing retrieval usage to be bounded together with the selection-conditioned risk. Experiments on three open-domain question answering (QA) benchmarks across multiple LLM backbones demonstrate that BalanceRAG meets prescribed risk levels, preserves higher coverage and more accepted correct examples, and reduces unnecessary retrieval calls compared with always-on RAG.

1 Introduction

Large language models (LLMs) have been widely deployed into real-world open-ended question answering (QA) scenarios (Pandit et al., 2025; Duan et al., 2025), but they still often produce fluent yet factually incorrect responses (Wang et al., 2025d; Huang et al., 2024). Retrieval-augmented generation (RAG) can mitigate this issue by grounding model output in external evidence, improving factuality on knowledge-intensive tasks (Lewis et al., 2020; Karpukhin et al., 2020; Asai et al., 2024; Jiang et al., 2023). However, always-on retrieval can be unnecessary and costly, since retrieval and long-context generation increase latency, memory use, and token consumption relative to direct LLM inference (Wang et al., 2024a; Li et al., 2024).

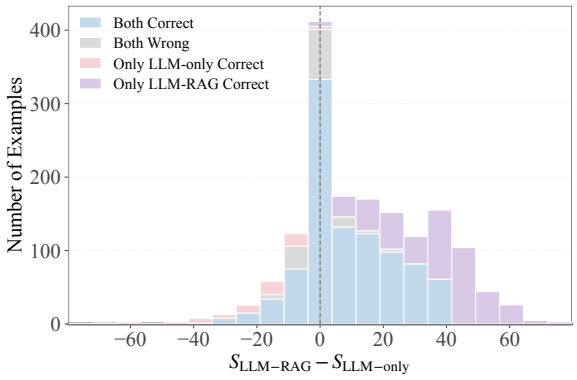


Figure 1: Distribution of the per-example score differences between RAG and LLM-only. $S_{\text{LLM-RAG}}$ and $S_{\text{LLM-only}}$ are the similarity scores between each path's prediction and the ground-truth answer. The x-axis reports $S_{\text{LLM-RAG}} - S_{\text{LLM-only}}$, with positive values favoring RAG and negative values favoring LLM-only, while the y-axis reports the number of examples. Colors distinguish whether both branches are correct, both are wrong, or only one branch is correct.

Figure 1 shows that LLM-only and RAG have similar correctness scores for many queries, suggesting that retrieval mainly adds overhead when the model-only answer is already reliable. At the same time, a non-negligible fraction of examples are answered correctly only with retrieval. Thus, retrieval is neither uniformly necessary nor safely removable (Mallen et al., 2023). A desirable system should keep easy examples on the LLM-only path, escalate uncertain queries to RAG, and abstain when neither path is reliable enough. Prior adaptive RAG frameworks follow this principle by routing queries with uncertainty or complexity signals (Wang et al., 2025a; Jeong et al., 2024), but they are mostly heuristic policies optimized for

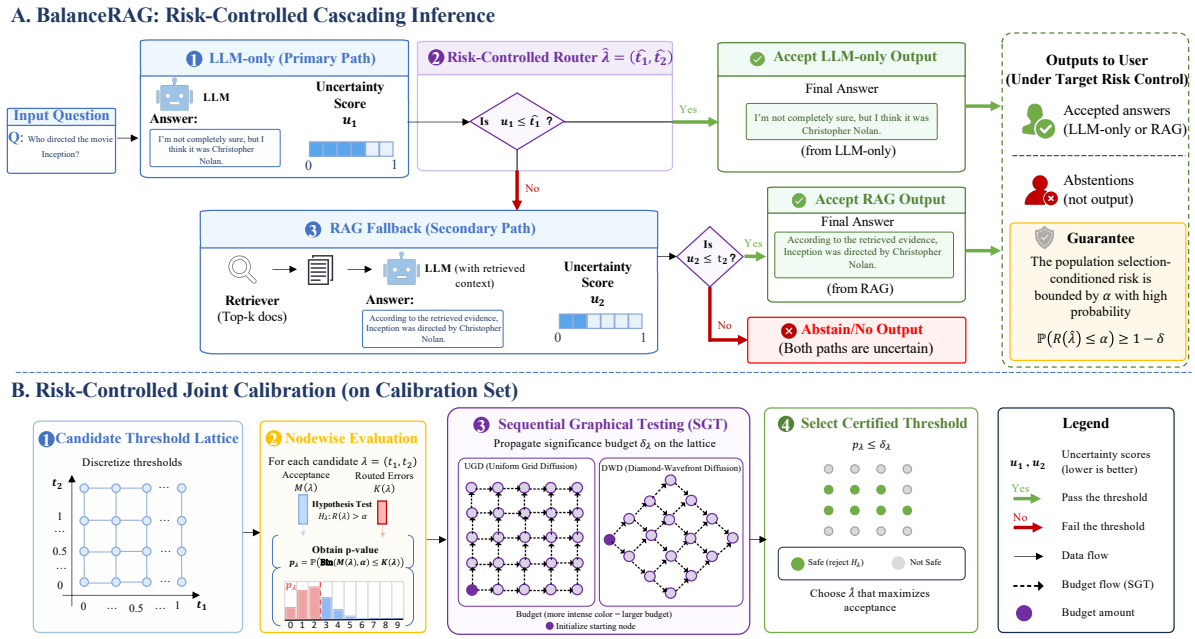


Figure 2: Overview of BalanceRAG: risk-controlled cascading inference (A) and joint threshold calibration via Sequential Graphical Testing (B). α is the user-specified risk level (upper bound).

average accuracy without finite-sample guarantees at a user-specified level. This leaves a deployment-critical question: *after adaptively combining LLM-only and RAG outputs, how can we control the error rate of the final accepted answers?*

To address this gap, we propose **BalanceRAG**, a training-free framework for risk-controlled cascaded RAG. As shown in Figure 2, given uncertainty scores from the LLM-only and RAG paths, BalanceRAG first attempts the LLM-only answer, invokes the RAG fallback only when the LLM-only is uncertain, and abstains if the fallback is also unreliable. The router is governed by two uncertainty thresholds, and BalanceRAG calibrates their joint operating point rather than tuning them stage by stage. Stage-wise calibration can be valid in cascaded systems (Jung et al., 2024), but it may be overly conservative as the reliability and utility of the final routing cascade depend on the system-level threshold choices for both branches.

Specifically, BalanceRAG formulates joint calibration as a multiple testing problem over a two-dimensional threshold lattice, where each threshold pair defines a candidate operating point, for which we test the null hypothesis that the system-level selection-conditioned risk exceeds α. Rejecting this null certifies the corresponding pair as safe. Although a Bonferroni correction provides valid error control (Bauer, 1991), it is overly conservative for this structured lattice: neighboring threshold pairs typically induce highly correlated routing decisions, and safe operating points often form contiguous regions. To exploit this structure, BalanceRAG performs Sequential Graphical Testing (SGT) (Bretz et al., 2009), which sequentially propagates significance budget across the lattice and reallocates unused budget from certified nodes to their successors. The final router then selects, among all certified threshold pairs, the one with the largest acceptance rate, while preserving high-probability control of the system-level selection-conditioned error rate (Wang et al., 2026, 2025c). Furthermore, we extend BalanceRAG to multi-risk calibration, enabling simultaneous control of answer error and retrieval usage.

Our contributions are threefold. First, to the best of our knowledge, BalanceRAG is the first framework to study finite-sample risk control for cascaded LLM/RAG routing, shifting adaptive retrieval from heuristic confidence estimation to statistically certified reliability. Second, we formulate joint threshold calibration as multiple testing over a two-dimensional lattice and adopt SGT to certify high-utility operating points. Third, we establish high-probability control of the system-level selection-conditioned error rate and extend the framework to multi-risk calibration.

2 Related Work

Adaptive Retrieval. Adaptive RAG aims to avoid unnecessary retrieval by routing queries based on

empirical signals. Previous studies explore when RAG is needed by comparing parametric and non-parametric memory (Mallen et al., 2023), or route queries using complexity estimation and learned policies (Jeong et al., 2024; Tang et al., 2025). Other methods trigger retrieval during generation based on token-level confidence, self-reflection, or information needs (Jiang et al., 2023; Asai et al., 2024; Su et al., 2024), while recent approaches further exploit retrieval-score statistics and agentic search (Wang et al., 2025a; Li et al., 2025). These methods reduce retrieval usage while preserving average answer quality, but they are mainly heuristic routing policies without statistical guarantees. BalanceRAG is orthogonal to these signals: they can serve as uncertainty estimates, while our calibration layer certifies the final LLM-RAG cascade at a prescribed selection-conditioned risk level.

Risk Control in Selective Prediction. Conformal prediction (Wang et al., 2024b, 2025e,b; Tan et al., 2025; Jia et al., 2025) and risk control (Angelopoulos et al., 2024; Li et al., 2026) provide finite-sample guarantees for task-specific risk targets. Recent work extends risk control to selected foundation-model outputs (Gui et al., 2024), AI-assigned labeling (Huang et al., 2025), cascaded LLM judges (Jung et al., 2024), and selective QA or routing systems (Wang et al., 2026, 2025c). These works provide the statistical foundation for reliable selection, but they do not investigate joint risk calibration for cascaded LLM-only/RAG routing. BalanceRAG targets this setting by certifying threshold pairs on a two-dimensional routing lattice, reducing conservativeness compared with stage-wise or Bonferroni-style calibration while preserving finite-sample risk control.

3 Methodology

3.1 Problem Formulation

Given a prompt x, BalanceRAG considers two inference branches: an LLM-only branch and an RAG fallback branch. Branch $b \in \{1, 2\}$ returns an answer $\hat{y}_b(x)$ together with an uncertainty score $u_b(x)$, where smaller values indicate higher confidence. Given a threshold pair $\lambda = (t_1, t_2)$, the router first accepts the LLM-only answer if $u_1(x) \leq t_1$. Otherwise, it invokes the RAG fallback and accepts $\hat{y}_2(x)$ if $u_2(x) \leq t_2$. If both checks fail, the system abstains by returning $\varnothing$. Thus, the final output $\hat{y}_\lambda(x) \in \{\hat{y}_1(x), \hat{y}_2(x), \varnothing\}$ is determined by the two thresholds jointly.

Let $A(y^\star, \hat{y}) \in \{0, 1\}$ indicate whether a generated answer $\hat{y}$ is admissible with respect to the ground truth $y^\star$. We define the acceptance indicator as

$$S_\lambda(x) = \mathbb{1}[\hat{y}_\lambda(x) \neq \emptyset]. \tag{1}$$

The accepted-error indicator is

$$Z_\lambda(x, y^\star) = S_\lambda(x)\,(1 - A(y^\star, \hat{y}_\lambda(x)))\,. \tag{2}$$

Here, $Z_\lambda(x, y^\star) = 1$ means that the router outputs an incorrect answer rather than abstaining. The population risk of interest is the selection-conditioned error rate:

$$R(\lambda) = \mathbb{P}\left(Z_\lambda(X, Y^\star) = 1 \mid S_\lambda(X) = 1\right). \tag{3}$$

Given a target risk level α and error level δ, our goal is to select a threshold pair $\hat{\lambda}$ with high acceptance while satisfying

$$\mathbb{P}\Big(R(\hat{\lambda}) \leq \alpha\Big) \geq 1 - \delta. \tag{4}$$

3.2 Joint Calibration with SGT

BalanceRAG calibrates the composed LLM-only/RAG router directly. Let $\Lambda = \{\lambda_j\}_{j=1}^N$ denote the candidate threshold pairs on a two-dimensional lattice. For each candidate λ_j, we test

$$H_j : \; R(\lambda_j) > \alpha. \tag{5}$$

Rejecting H_j certifies λ_j as safe with respect to the target risk level. Since the final threshold pair will be selected adaptively from all certified candidates, we require family-wise error rate (FWER) (Holm, 1979) control at level δ. Letting $\widehat{\Lambda}$ denote the certified set, FWER control ensures

$$\mathbb{P}\left(\sup_{\lambda \in \widehat{\Lambda}} R(\lambda) \leq \alpha\right) \geq 1 - \delta. \tag{6}$$

Therefore, any later choice from $\widehat{\Lambda}$, including the acceptance-maximizing one, remains valid. A Bonferroni correction also provides this guarantee, but it is often overly conservative on dense threshold grids. BalanceRAG instead uses SGT (Bretz et al., 2009), which preserves FWER control while exploiting the lattice structure to recycle significance budget to more useful operating points.

Step 1: p-value computation. On the calibration set $\mathcal{D}_{\text{cal}} = \{(x_\ell, y_\ell^\star)\}_{\ell=1}^n$, for each threshold pair $\lambda_{i,j} = (t_1^{(i)}, t_2^{(j)})$, we compute the accepted count and accepted-error count:

$$M_{i,j} = \sum_{\ell=1}^n S_{\lambda_{i,j}}(x_\ell), \; K_{i,j} = \sum_{\ell=1}^n Z_{\lambda_{i,j}}(x_\ell, y_\ell^\star). \tag{7}$$

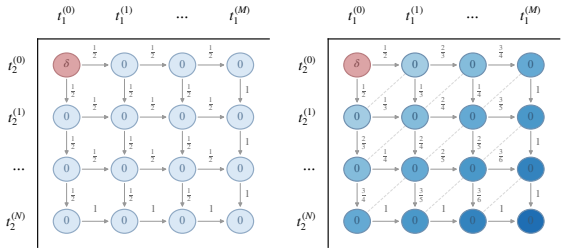


Figure 3: Budget diffusion on the threshold lattice. UGD splits budget evenly across successors, while DWD gives equal accumulated budget to nodes on the same diagonal.

For a fixed candidate $\lambda_{i,j}$, conditioned on $M_{i,j}$,

$$K_{i,j} \mid M_{i,j} \sim \mathrm{Bin}(M_{i,j}, R(\lambda_{i,j}))\,. \tag{8}$$

We therefore use the one-sided exact binomial p-value

$$p_{i,j} = \mathbb{P}(\mathrm{Bin}(M_{i,j}, \alpha) \le K_{i,j})\,, \tag{9}$$

which is super-uniform under $H_{\lambda_{i,j}}$.

Step 2: graph prior and budget diffusion. The candidate thresholds form a natural partial order: increasing either threshold makes the router more permissive and usually increases acceptance. We therefore direct edges from each node to its immediate relaxed neighbors:

$$\mathcal{N}(i,j) \subseteq \{(i+1,j),(i,j+1)\}. \tag{10}$$

As the safe region varies across datasets and models, we split off a small portion of the calibration set as an initialization split. On this split, we choose the highest-acceptance node among those satisfying $p \le \delta$; if no such node exists, we choose the node with the smallest p-value. The selected node receives the initial budget δ, while all formal SGT p-values are computed on the remaining calibration data.

As shown in Figure 3, we consider two diffusion rules over successors $u \in \mathcal{N}(i,j)$. Uniform Grid Diffusion (UGD) assigns

$$g^{\mathrm{UGD}}_{(i,j)\to u} = \frac{1}{|\mathcal{N}(i,j)|}. \tag{11}$$

Diamond-Wavefront Diffusion (DWD) assigns

$$\begin{aligned} g^{\mathrm{DWD}}_{(i,j)\to u} = &\frac{(i+1)\mathbb{1}[u=(i+1,j)]}{i+j+2} \\ &+ \frac{(j+1)\mathbb{1}[u=(i,j+1)]}{i+j+2}. \end{aligned} \tag{12}$$

Under DWD, each node on diagonal $i+j=d$ receives accumulated budget

$$\delta^{\mathrm{DWD}}_{i,j} = \frac{\delta}{d+1}, \tag{13}$$

so threshold pairs with the same total relaxation receive the same testing opportunity. Appendix B.2 proves this property.

Step 3: SGT budget propagation. SGT maintains a local testing budget $\delta_{i,j}$ at each active node and certifies the node whenever

$$p_{i,j} \le \delta_{i,j}. \tag{14}$$

Once node (i,j) is certified, its budget is transferred to uncertified successors:

$$\delta_u \leftarrow \delta_u + \delta_{i,j} g_{(i,j)\to u}. \tag{15}$$

Because the graph is acyclic, SGT can apply the corresponding rewiring update

$$g_{k,u} \leftarrow g_{k,u} + g_{k,(i,j)} g_{(i,j),u}, \tag{16}$$

where k is a predecessor and u is a successor. This budget recycling preserves validity while giving later, more permissive candidates greater power to be certified.

Step 4: final threshold selection. After SGT terminates, BalanceRAG selects the certified threshold pair with the largest accepted count:

$$\hat{\lambda} = \arg\max_{\lambda \in \widehat{\Lambda}} M(\lambda). \tag{17}$$

Since this selection is restricted to the certified set, it inherits the simultaneous risk guarantee in Eq. (6).

Theorem 1 (Risk control of calibrated routing)**.** *Under i.i.d. calibration data, for any target risk α and error level δ, any threshold pair $\hat{\lambda}$ selected from the SGT-certified set satisfies*

$$\mathbb{P}\left(R(\hat{\lambda}) \le \alpha\right) \ge 1-\delta. \tag{18}$$

The proof is given in Appendix B.3.

3.3 Multi-Risk Control

The same certification procedure can control multiple system-level risks. Besides the selection-conditioned answer error,

$$R^{(1)}(\lambda) = \mathbb{P}\left(Z_\lambda = 1 \mid S_\lambda = 1\right), \tag{19}$$

we also control the fallback invocation rate,

$$R^{(2)}(\lambda) = \mathbb{P}(B_\lambda = 1), \tag{20}$$

where B_λ indicates whether the input is routed to the RAG fallback. Given targets α_1 and α_2, each

node receives one p-value per risk, denoted by $p_{i,j}^{(1)}$ and $p_{i,j}^{(2)}$. The intersection-union rule (Berger, 1982) combines them as

$$p_{i,j} = \max\left(p_{i,j}^{(1)}, p_{i,j}^{(2)}\right). \tag{21}$$

Feeding this node-level p-value into the same SGT procedure yields

$$\mathbb{P}\left(R^{(1)}(\hat{\lambda}) \le \alpha_1,\ R^{(2)}(\hat{\lambda}) \le \alpha_2\right) \ge 1-\delta. \tag{22}$$

Thus, multi-risk BalanceRAG jointly certifies answer reliability and fallback usage, allowing the final router to trade off coverage and retrieval frequency under explicit risk constraints.

4 Experiment

4.1 Experimental Settings

Datasets and Base LLMs. We evaluate BalanceRAG on three open-ended QA datasets: TriviaQA (Joshi et al., 2017), SQuAD v2 (Rajpurkar et al., 2018), and Natural Questions (NQ) (Kwiatkowski et al., 2019). To assess robustness across model families and scales, we employ eight backbone LLMs from Qwen (Yang et al., 2025, 2024), LLaMA (Grattafiori et al., 2024), and Vicuna (Zheng et al., 2023).

Baselines. We compare BalanceRAG with three groups of baselines. First, we include fixed-path systems, LLM-only and LLM-RAG, both calibrated by fixed sequence testing (Bauer, 1991) as single-branch risk-controlled baselines. Second, we also evaluate two BalanceRAG variants: BalanceRAG-UGD, which uniformly diffuses the significance budget to neighboring nodes, and BalanceRAG-DWD, which uses diamond-wavefront diffusion to balance testing opportunities among threshold pairs with the same total relaxation. We employ DWD as the default variant. Third, we compare with UCB-based methods for selection-conditioned risk control. UCB-Cascaded-CP adapts the cascaded judge framework of Jung et al. (2024) to our LLM-only/RAG routing setting with the Clopper-Pearson bound, while UCB-Cascaded-HFD replaces it with a distribution-free Hoeffding bound. For completeness, Appendix 6 reports additional baselines, including Adaptive-RAG (Jeong et al., 2024), Self-Route (Li et al., 2024), empirical risk selection, and Bonferroni correction, covering uncalibrated adaptive routing, simple empirical selection, and conservative multiple-testing correction under the same target risk levels.

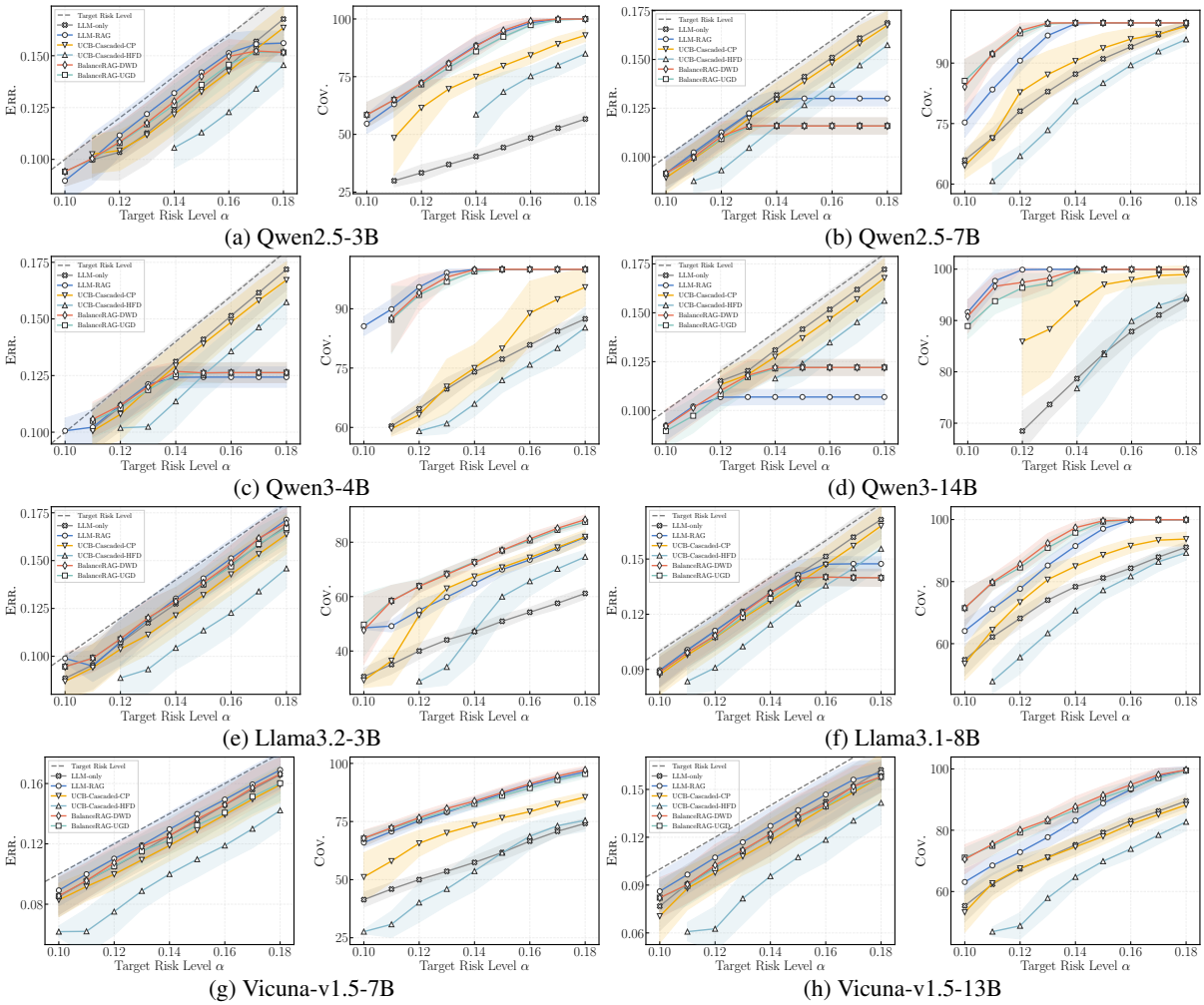
Evaluation Metrics. We consider four dimensions. Reliability is measured by the empirical selection-conditioned error rate (Err.) under different target risk levels α. Coverage is measured by the acceptance rate (Cov.). Utility is measured by the number of accepted correct samples (Corr.), which captures how many useful answers the system produces. Cost is measured by the average number of tokens and the average memory increase per sample after uncertainty scores are obtained, so it reflects the execution cost induced by the calibrated routing policy rather than the overhead of any specific uncertainty estimator.

Correctness Criterion and Uncertainty Estimator. By default, we use semantic similarity (Reimers and Gurevych, 2019) with a threshold of 0.7 to determine whether a model answer aligns with the ground truth in the admission function. In the robustness analysis, we further consider bidirectional entailment (Kuhn et al., 2023) and LLM-as-a-Judge (Zhang et al., 2024) as alternative correctness criteria. For uncertainty estimation, we consider five estimators: semantic entropy (SE) (Farquhar et al., 2024), the sum of graph Laplacian eigenvalues (EIGV), eccentricity (ECC), the degree matrix statistic (DEG) (Lin et al., 2023), and length-normalized predictive entropy from the model outputs (SELF) (Malinin and Gales, 2020). Unless otherwise stated, we use SE as the default uncertainty estimator.

Hyperparameters. Following standard split-calibration protocols (Angelopoulos et al., 2024), we use a calibration-test split ratio of 0.5 and repeat each experiment over 100 random calibration-test splits. In each calibration set, we reserve 40% of the examples for initial-node selection and use the remaining data for formal SGT calibration.

4.2 Utility and Coverage under Risk Control

Reliability under Target Risk Control. We first examine whether BalanceRAG achieves system-level risk control at user-specified target levels. Figure 4 reports the empirical test-set selection-conditioned error rate across backbones and target risk levels, where solid curves denote the mean over repeated calibration/test splits and shaded regions show one standard deviation. BalanceRAG provides a finite-sample, high-probability guarantee that the population selection-conditioned risk of the calibrated router is at most the target level α. The empirical test-set Err. closely tracks the target line, which is consistent with this guarantee.



(a) Qwen2.5-3B (b) Qwen2.5-7B (c) Qwen3-4B (d) Qwen3-14B (e) Llama3.2-3B (f) Llama3.1-8B (g) Vicuna-v1.5-7B (h) Vicuna-v1.5-13B

Figure 4: Err. (left in each panel pair) and Cov. (right) under different target risk levels α across eight LLMs spanning four model families on TriviaQA.

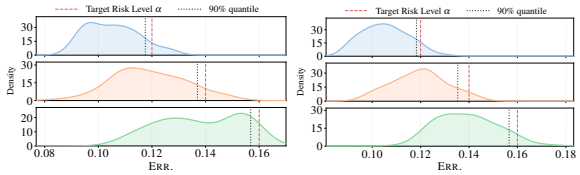


Figure 5: Err. distributions for Qwen2.5-3B (Left) and Llama-3.2-3B (Right) on TriviaQA.

Figure 5 further shows the distribution of test-set Err. over repeated random splits. When $\delta = 0.1$, the fraction of splits with test-set Err. exceeding α remains below 10% at every target level. Moreover, joint calibration keeps BalanceRAG feasible under strict risk targets, whereas branch-wise baselines such as LLM-only and UCB-Cascaded often fail to find a feasible operating point due to branch-level constraints and stage-wise δ splitting.

Coverage and Utility under Risk Control. We next evaluate how much useful output is retained at the same risk level α. Figure 4 and Table 1 show that BalanceRAG consistently achieves the best or near-best coverage and utility among the baselines. It outperforms non-RAG baselines and matches or exceeds RAG in many settings. Notably, BalanceRAG accepts more test samples and retains more correctly accepted predictions (Corr.), while maintaining empirical Err. near or below the risk level. This advantage reflects the heterogeneous value of retrieval: retrieval can help knowledge-intensive cases, but may also introduce noisy or distracting context for easier queries. By jointly calibrating the LLM-only and fallback RAG branches, BalanceRAG keeps cases where the LLM-only answer is sufficient on the primary path and routes harder cases to LLM-RAG when the fallback is sufficiently reliable. Appendix 6 further compares BalanceRAG with additional baselines under the same target risk levels. BalanceRAG is more reliable than Adaptive-RAG, Self-Route, and empirical risk selection, while achieving higher coverage than Bonferroni. Overall, joint calibration better balances reliability, coverage, and utility.

Table 1: Results on TriviaQA, SQuAD v2, and NQ under different target risk levels. Bold numbers indicate the best Corr. under the same dataset, model, and risk level. "-" denotes infeasibility, i.e., no non-empty certified operating point is found under the target risk level.

	Method	Metric	*Llama3.2-3B*					*Llama3.1-8B*				
	α		0.11	0.12	0.13	0.14	0.15	0.11	0.12	0.13	0.14	0.15
TriviaQA	LLM-only	Corr.	988.0	1116.0	1214.0	1283.0	1371.0	1780.0	1928.0	2070.0	2165.0	2218.0
		Err.	0.0955	0.1070	0.1175	0.1274	0.1373	0.1004	0.1108	0.1215	0.1318	0.1415
	LLM-RAG	Corr.	1390.0	1532.0	1646.0	1760.0	1876.0	2036.0	2198.0	2382.0	2529.0	2652.0
		Err.	0.0949	0.1075	0.1192	0.1302	0.1407	0.1006	0.1110	0.1214	0.1314	0.1413
	UCB-Cascaded	Corr.	1028.0	1488.0	1749.0	1850.0	1917.0	1849.0	2084.0	2264.0	2360.0	2434.0
		Err.	0.0941	0.1037	0.1113	0.1214	0.1320	0.0978	0.1072	0.1178	0.1272	0.1367
	BalanceRAG	Corr.	**1643.0**	**1774.0**	**1873.0**	**1973.0**	**2073.0**	**2288.0**	**2430.0**	**2584.0**	**2691.0**	**2727.0**
		Err.	0.0990	0.1091	0.1201	0.1287	0.1384	0.0991	0.1084	0.1206	0.1316	0.1396
	α		0.11	0.12	0.13	0.14	0.15	0.11	0.12	0.13	0.14	0.15
SQuAD v2	LLM-only	Corr.	-	-	-	-	137.0	-	-	-	194.0	211.0
		Err.	-	-	-	-	0.1345	-	-	-	0.1319	0.1324
	LLM-RAG	Corr.	**1546.0**	**1617.0**	**1666.0**	**1714.0**	**1755.0**	**1803.0**	**1831.0**	**1850.0**	**1853.0**	**1853.0**
		Err.	0.1008	0.1112	0.1216	0.1316	0.1416	0.1000	0.1095	0.1175	0.1193	0.1193
	UCB-Cascaded	Corr.	-	-	-	-	1744.0	-	-	-	1835.0	1837.0
		Err.	-	-	-	-	0.1354	-	-	-	0.1263	0.1269
	BalanceRAG	Corr.	1484.0	1591.0	1654.0	1699.0	1745.0	1784.0	1819.0	1836.0	1842.0	1843.0
		Err.	0.0975	0.1066	0.1164	0.1271	0.1375	0.1010	0.1118	0.1210	0.1241	0.1240
	α		0.15	0.16	0.17	0.18	0.19	0.15	0.16	0.17	0.18	0.19
NQ	LLM-only	Corr.	-	-	119.0	125.0	134.0	158.0	191.0	240.0	283.0	320.0
		Err.	-	-	0.1568	0.1552	0.1572	0.1055	0.1184	0.1362	0.1493	0.1630
	LLM-RAG	Corr.	**489.0**	**532.0**	**562.0**	**590.0**	**615.0**	**734.0**	**752.0**	**760.0**	**763.0**	**763.0**
		Err.	0.1283	0.1390	0.1497	0.1601	0.1697	0.1331	0.1422	0.1466	0.1478	0.1478
	UCB-Cascaded	Corr.	-	-	-	563.0	590.0	698.0	718.0	736.0	743.0	747.0
		Err.	-	-	-	0.1520	0.1582	0.1214	0.1307	0.1408	0.1492	0.1581
	BalanceRAG	Corr.	447.0	492.0	531.0	575.0	615.0	709.0	741.0	753.0	757.0	759.0
		Err.	0.1220	0.1312	0.1404	0.1506	0.1662	0.1315	0.1420	0.1511	0.1523	0.1517

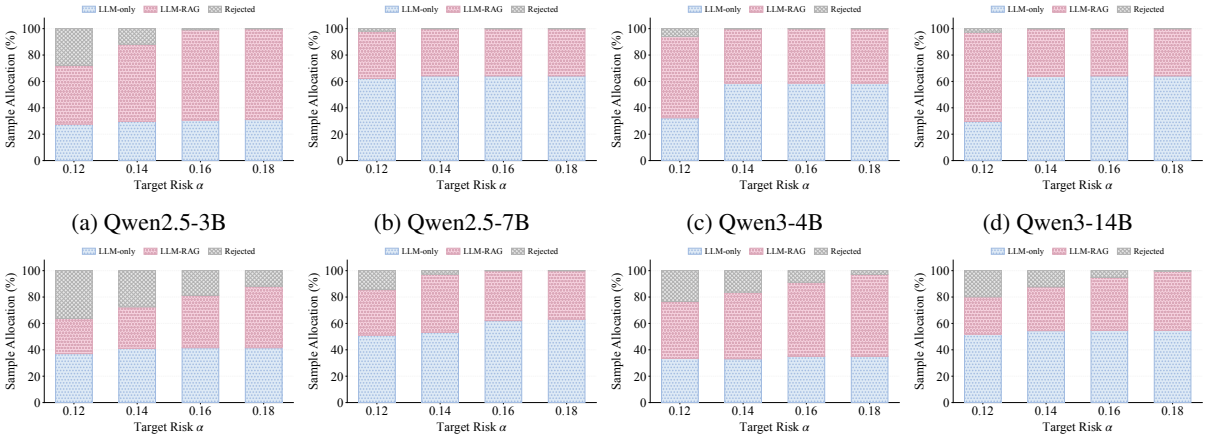
4.3 Routing Allocation and Cost Efficiency

Routing Allocation across Models. Figure 6 shows the routing allocation of BalanceRAG across eight backbones. BalanceRAG consistently keeps a substantial fraction of queries on the LLM-only branch, thereby reducing unnecessary fallback usage while maintaining the prescribed risk control. The LLM-only share also reflects the strength of the backbone: stronger models can safely answer more queries without retrieval, whereas weaker models require more frequent fallback to satisfy the same risk target. In our results, stronger backbones keep more than 60% of accepted queries on the LLM-only branch, medium-strength backbones stay around 40-50%, and weaker backbones remain around 25-30%.

Routed Execution Cost. We further evaluate the routed execution cost of BalanceRAG at calibrated operating points. Since BalanceRAG does not depend on a specific uncertainty estimator, we measure cost after uncertainty scores are obtained; this isolates the execution cost induced by the calibrated routing policy rather than the overhead of a particular uncertainty-estimation method. Under the same backbone and decoding setup, LLM-only and RAG perform fixed single-path inference, whereas BalanceRAG first attempts the LLM-only branch and invokes the RAG fallback only when the primary answer does not pass its calibrated threshold. We report the average memory increase and average token usage per sample. As shown in Figure 7, BalanceRAG lies between the two fixed-path endpoints: it incurs more cost than LLM-only, but substantially lower cost than always-on RAG, because many queries can be accepted directly by the primary branch without invoking retrieval.

4.4 Multi-Risk Control

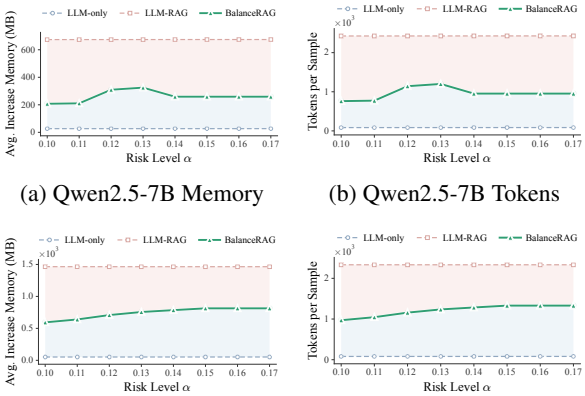
In deployment, practitioners may want to control not only the selection-conditioned answer risk but also the frequency of fallback retrieval. We therefore extend BalanceRAG to the multi-risk setting by adding a constraint α_2 on the fallback invocation rate in addition to the answer-risk target α_1. Table 2 shows that stricter α_2 values lead BalanceRAG to certify operating points with lower fallback rates, shifting more queries to the cheaper



(a) Qwen2.5-3B (b) Qwen2.5-7B (c) Qwen3-4B (d) Qwen3-14B

(e) Llama3.2-3B (f) Llama3.1-8B (g) Vicuna-v1.5-7B (h) Vicuna-v1.5-13B

Figure 6: Routing allocation of test samples in BalanceRAG on TriviaQA (mean), including LLM-only acceptance, LLM-RAG fallback acceptance, and rejection.



(a) Qwen2.5-7B Memory (b) Qwen2.5-7B Tokens

(c) Llama-3.1-8B Memory (d) Llama-3.1-8B Tokens

Figure 7: Memory and token cost on TriviaQA across risk levels for Qwen2.5-7B and Llama-3.1-8B

LLM-only branch. This reduces retrieval usage, with only a moderate decrease in coverage and correctly accepted samples. At the same time, both risks remain controlled in all reported configurations: the empirical selection-conditioned risk stays below $\alpha_1 = 0.15$, and the fallback invocation rate stays below α_2. These results show that multi-risk BalanceRAG provides an explicit mechanism for navigating the reliability-utility-retrieval trade-off.

5 Conclusion

In this paper, we develop BalanceRAG, a training-free framework for joint threshold calibration in risk-controlled cascaded LLM-RAG systems. By certifying candidate threshold pairs on a two-dimensional lattice with SGT, BalanceRAG selects high-utility operating points while providing finite-sample, high-probability control of the selection-conditioned error rate. The framework also supports multi-risk calibration to bound fallback retrieval usage together with answer risk. Across open-domain QA benchmarks and diverse backbones, BalanceRAG meets prescribed risk levels, preserves high coverage and accepted correct answers, and reduces unnecessary retrieval calls. These results suggest a principled path from heuristic routing to reliable cascaded LLM-RAG.

Table 2: Single-risk vs. multi-risk routing at selection-conditioned risk target $\alpha_1 = 0.15$. α_2 caps the fallback rate to LLM-RAG; "Route" shows the LLM-only/LLM-RAG split.

Model	Method	α_2	Route (Only/RAG)	Cov.	Corr.	Err.
Qwen2.5-3B	LLM-only	-	1.0000 / 0.0000	0.4432	1137	0.1341
	Single-risk	-	0.2995 / 0.6505	0.9499	2421	0.1400
	Multi-risk	0.6	0.3464 / **0.5620**	0.9084	2332	0.1338
	Multi-risk	0.5	0.3859 / **0.4709**	0.8568	2206	0.1313
Qwen2.5-7B	LLM-only	-	1.0000 / 0.0000	0.9105	2422	0.1412
	Single-risk	-	0.6414 / 0.3582	0.9996	2738	0.1159
	Multi-risk	0.3	0.7559 / **0.2438**	0.9996	2719	0.1221
	Multi-risk	0.2	0.8257 / **0.1729**	0.9986	2692	0.1298
Llama3.1-8B	LLM-only	-	1.0000 / 0.0000	0.8118	2218	0.1415
	Single-risk	-	0.5935 / 0.4024	0.9959	2727	0.1396
	Multi-risk	0.3	0.7200 / **0.2593**	0.9793	2689	0.1372
	Multi-risk	0.2	0.7748 / **0.1702**	0.9450	2599	0.1357
Llama3.2-3B	LLM-only	-	1.0000 / 0.0000	0.5092	1371	0.1373
	Single-risk	-	0.4162 / 0.3539	0.7701	2073	0.1384
	Multi-risk	0.3	0.4872 / **0.2504**	0.7376	1989	0.1366
	Multi-risk	0.2	0.5320 / **0.1279**	0.6599	1781	0.1355

Limitations

BalanceRAG relies on the standard split-calibration assumption that the calibration and deployment examples are drawn from the same distribution. If the query distribution, retrieval corpus, or backbone model changes substantially, the calibrated thresholds may need to be updated. This is a common limitation of finite-sample

calibration methods and can be addressed in practice through periodic recalibration.

The guarantee is also defined with respect to the chosen correctness criterion. Although we evaluate semantic similarity, bidirectional entailment, and LLM-as-a-Judge variants, open-domain QA correctness can still be ambiguous, especially for partially correct or underspecified answers. This reflects a broader challenge in evaluating free-form generation rather than a limitation specific to BalanceRAG.

Finally, our experiments focus on two-branch LLM-RAG cascades. Extending the calibration principle to multi-stage retrieval, tool-augmented systems, or agentic workflows is a promising direction, but may require more scalable search and calibration over larger routing spaces.

References

Anastasios Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. 2024. Conformal risk control. In *International Conference on Learning Representations*, volume 2024, pages 55198–55218.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, volume 2024, pages 9112–9141.

Peter Bauer. 1991. Multiple testing in clinical trials. *Statistics in Medicine*, 10(6):871–890.

Roger L. Berger. 1982. Multiparameter hypothesis testing and acceptance sampling. *Technometrics*, 24(4):295–300.

Frank Bretz, Willi Maurer, Werner Brannath, and Martin Posch. 2009. A graphical approach to sequentially rejective multiple test procedures. *Statistics in Medicine*, 28(4):586–604.

Jinhao Duan, Xinyu Zhao, Zhuoxuan Zhang, Eunhye Grace Ko, Lily Boddy, Chenan Wang, Tianhao Li, Alexander Rasgon, Junyuan Hong, Min Kyung Lee, Chenxi Yuan, Qi Long, Ying Ding, Tianlong Chen, and Kaidi Xu. 2025. GuideLLM: Exploring LLM-guided conversation with applications in autobiography interviewing. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5558–5588.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

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

Yu Gui, Ying Jin, and Zhimei Ren. 2024. Conformal alignment: Knowing when to trust foundation models with guarantees. *Advances in Neural Information Processing Systems*, 37:73884–73919.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70.

Huipeng Huang, Wenbo Liao, Huajun Xi, Hao Zeng, Mengchen Zhao, and Hongxin Wei. 2025. Selective labeling with false discovery rate control. *arXiv preprint arXiv:2510.14581*.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, and 52 others. 2024. Position: TrustLLM: Trustworthiness in large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 20166–20270.

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: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050.

Zijun Jia, Diyin Tang, Hongyu Long, and Jinsong Yu. 2025. Coverage-guaranteed speech emotion recognition via calibrated uncertainty-adaptive prediction sets. *Engineering Applications of Artificial Intelligence*, 159:111721.

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 *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 7969–7992.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Jaehun Jung, Faeze Brahman, and Yejin Choi. 2024. Trust or escalate: Llm judges with provable guarantees for human agreement. *arXiv preprint arXiv:2407.18370*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and

Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5420–5438.

Ye Li, Anqi Hu, Yuanchang Ye, Shiyan Tong, Zhiyuan Wang, and Bo Fu. 2026. Set-valued prediction for large language models with feasibility-aware coverage guarantees. *arXiv preprint arXiv:2603.22966*.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

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.

Shrey Pandit, Jiawei Xu, Junyuan Hong, Zhangyang Wang, Tianlong Chen, Kaidi Xu, and Ying Ding. 2025. Medhallu: A comprehensive benchmark for detecting medical hallucinations in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2858–2873.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

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 and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013.

Binyu Tan, Zhiyuan Wang, Jinhao Duan, Kaidi Xu, Heng Tao Shen, Xiaoshuang Shi, and Fumin Shen. 2025. Conformal lesion segmentation for 3d medical images. *arXiv preprint arXiv:2510.17897*.

Xiaqiang Tang, Qiang Gao, Jian Li, Nan Du, Qi Li, and Sihong Xie. 2025. Mba-rag: a bandit approach for adaptive retrieval-augmented generation through question complexity. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3248–3254.

Hairu Wang, Yuan Feng, Yukun Cao, Xike Xie, and S Kevin Zhou. 2025a. Skewroute: Training-free llm routing for knowledge graph retrieval-augmented generation via score skewness of retrieved context. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11324–11340.

Qingni Wang, Tiantian Geng, Zhiyuan Wang, Teng Wang, Bo Fu, and Feng Zheng. 2025b. Sample then identify: A general framework for risk control and assessment in multimodal large language models. In *International Conference on Learning Representations*, volume 2025, pages 64280–64297.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, and 1 others. 2024a. Searching for best practices in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17716–17736.

Zhiyuan Wang, Tianlong Chen, Yue Zhang, Heng Tao Shen, Xiaoshuang Shi, Kaidi Xu, and 1 others. 2025c. Lec: Linear expectation constraints for false-discovery control in selective prediction and routing systems. *arXiv preprint arXiv:2512.01556*.

Zhiyuan Wang, Jinhao Duan, Lu Cheng, Yue Zhang, Qingni Wang, Xiaoshuang Shi, Kaidi Xu, Heng Tao Shen, and Xiaofeng Zhu. 2024b. ConU: Conformal uncertainty in large language models with correctness coverage guarantees. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6886–6898.

Zhiyuan Wang, Jinhao Duan, Qingni Wang, Xiaofeng Zhu, Tianlong Chen, Xiaoshuang Shi, and Kaidi Xu. 2026. Coin: Uncertainty-guarding selective question answering for foundation models with provable risk guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33764–33772.

Zhiyuan Wang, Jinhao Duan, Chenxi Yuan, Qingyu Chen, Tianlong Chen, Yue Zhang, Ren Wang, Xiaoshuang Shi, and Kaidi Xu. 2025d. Word-sequence entropy: Towards uncertainty estimation in free-form medical question answering applications and beyond. *Engineering Applications of Artificial Intelligence*, 139:109553.

Zhiyuan Wang, Qingni Wang, Yue Zhang, Tianlong Chen, Xiaofeng Zhu, Xiaoshuang Shi, and Kaidi Xu. 2025e. SConU: Selective conformal uncertainty in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19052–19075.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Ruiyang Zhang, Hu Zhang, and Zhedong Zheng. 2024. Vl-uncertainty: Detecting hallucination in large vision-language model via uncertainty estimation. *arXiv preprint arXiv:2411.11919*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

A Use of AI Assistants

We used AI assistants (ChatGPT) to assist with grammar and proofreading in our paper writing.

B Proofs

This appendix provides the proofs deferred from Section 3. Throughout, $\{(x_\ell, y_\ell^\star\}_{\ell=1}^n$ is the i.i.d. calibration set, and for any threshold pair $\lambda = (t_1, t_2)$ we use the notation $M(\lambda)$, $K(\lambda)$, $R(\lambda)$ from Section 3.1. We also write $F_{m,q}$ for the CDF of the binomial distribution $\mathrm{Bin}(m, q)$.

B.1 Super-uniformity of the binomial p-value

Lemma 1 (Super-uniformity)**.** *For each lattice node $\lambda_{i,j}$, let*

$$p_{i,j} \;=\; F_{M(\lambda_{i,j}),\alpha}\big(K(\lambda_{i,j})\big). \tag{23}$$

Under the null hypothesis $H_{\lambda_{i,j}} : R(\lambda_{i,j}) \geq \alpha$, $p_{i,j}$ is super-uniform, i.e. for every $t \in [0, 1]$,

$$\mathbb{P}_{H_{\lambda_{i,j}}}\big(p_{i,j} \leq t\big) \;\leq\; t. \tag{24}$$

Proof. Fix $\lambda = \lambda_{i,j}$ and abbreviate $M = M(\lambda)$, $K = K(\lambda)$, $R = R(\lambda)$. Conditionally on M, the samples accepted by λ are i.i.d. Bernoulli with error probability R, hence

$$K \mid M \;\sim\; \mathrm{Bin}(M, R). \tag{25}$$

Under H_λ we have $R \geq \alpha$, which implies the stochastic ordering $\mathrm{Bin}(M, R) \succeq_{\mathrm{st}} \mathrm{Bin}(M, \alpha)$; equivalently, for every integer k,

$$F_{M,R}(k) \;\leq\; F_{M,\alpha}(k). \tag{26}$$

By the generalized probability integral transform applied to the (possibly discrete) CDF $F_{M,R}$, for every $t \in [0, 1]$,

$$\mathbb{P}\big(F_{M,R}(K) \leq t \mid M\big) \;\leq\; t. \tag{27}$$

Combining Eq. 26 and Eq. 27 gives

$$\begin{aligned}\mathbb{P}(p_{i,j} \leq t \mid M) &= \mathbb{P}\big(F_{M,\alpha}(K) \leq t \mid M\big)\\ &\leq \mathbb{P}\big(F_{M,R}(K) \leq t \mid M\big)\\ &\leq t.\end{aligned} \tag{28}$$

Marginalizing over M yields $\mathbb{P}_{H_{\lambda_{i,j}}}(p_{i,j} \leq t) \leq t$. □

B.2 Shell-equal budget property of DWD

We record here a structural property of DWD that motivates its design (cf. Eq. 13 in the main text). The result is self-contained and does not depend on any hypothesis-testing content; while it is not used in the proof of Theorem 1, it provides useful intuition for the behavior of DWD on the lattice.

Lemma 2 (Shell-equal budget)**.** *Let the lattice Λ have root $(0, 0)$ with initial budget $\delta_{0,0} = \delta$ and all other budgets set to zero. If the edge weights follow the DWD rule in Eq. 12 and each node forwards its budget to neighbors once reached, then every node (i, j) with $i + j = d$ accumulates*

$$\delta_{i,j}^{\mathrm{DWD}} \;=\; \frac{\delta}{d+1}. \tag{29}$$

Proof. We prove Eq. 29 by induction on d.

Base case ($d = 0$). The only node is $(0, 0)$ itself, which trivially receives $\delta_{0,0}^{\mathrm{DWD}} = \delta = \delta/(0 + 1)$.

Inductive step. Assume the claim holds on diagonal d. Let (a, b) be any node on diagonal $d+1$, so $a+b = d+1$. Budget reaches (a, b) only from its in-lattice parents $(a - 1, b)$ and $(a, b - 1)$. Using Eq. 12,

$$g_{(a-1,b)\to(a,b)}^{\mathrm{DWD}} = \frac{a}{(a-1)+b+2} = \frac{a}{d+2}, \tag{30}$$

$$g_{(a,b-1)\to(a,b)}^{\mathrm{DWD}} = \frac{b}{a+(b-1)+2} = \frac{b}{d+2}. \tag{31}$$

By the inductive hypothesis, both parents carry budget $\delta/(d + 1)$. Summing the incoming contributions,

$$\begin{aligned}\delta_{a,b}^{\mathrm{DWD}} &= \frac{\delta}{d+1}\cdot\frac{a}{d+2} + \frac{\delta}{d+1}\cdot\frac{b}{d+2}\\ &= \frac{\delta}{d+1}\cdot\frac{a+b}{d+2}\\ &= \frac{\delta}{d+1}\cdot\frac{d+1}{d+2} = \frac{\delta}{d+2},\end{aligned} \tag{32}$$

where we used $a + b = d + 1$ in the penultimate step.

For boundary nodes with only one in-lattice parent ($a = 0$ or $b = 0$), the missing term in Eq. 32 has numerator zero, and the "sole neighbor receives weight 1" convention makes the lone parent contribute $\delta/(d + 1) \cdot 1 = \delta/(d + 2)$ after the same algebra. Hence every node on diagonal $d + 1$ accumulates $\delta/(d + 2)$, completing the induction. □

Remark 1. *Lemma 2 implies that the total budget on each diagonal is conserved: summing $\delta/(d+1)$ over the $d+1$ nodes of diagonal d yields δ.*

B.3 Proof of Theorem 1

Proof of Theorem 1. The proof combines two ingredients: the validity of the binomial p-value (Lemma 1) and the FWER control of SGT.

(i) Valid p-values. By Lemma 1, the p-value $p_{i,j}$ defined in Eq. 9 is super-uniform under the null $H_{\lambda_{i,j}}$ for every lattice node.

(ii) FWER control by SGT. Both the UGD weights (Eq. 11) and the DWD weights (Eq. 12) are non-negative, satisfy $g_{(i,j)\to(i,j)} = 0$, and obey $\sum_{u\in\mathcal{N}(i,j)} g_{(i,j)\to u} \leq 1$ at every node. Hence, the proposed graph satisfies the regularity conditions of Bretz et al. (2009). Combined with the initial budget allocation $\sum_{(i,j)} \delta_{i,j} \leq \delta$ and the super-uniform p-values from (i), Algorithm 1 of Bretz et al. (2009) yields a sequentially rejective multiple testing procedure that strongly controls the family-wise error rate at level δ. Therefore, with the null hypotheses defined as $H_\lambda : R(\lambda) > \alpha$, the certified set $\widehat{\Lambda}$ returned by the procedure satisfies

$$\mathbb{P}\Big(\sup_{\lambda\in\widehat{\Lambda}} R(\lambda) \leq \alpha\Big) \;\geq\; 1-\delta, \tag{33}$$

which recovers Eq. 6.

(iii) From $\widehat{\Lambda}$ to $\hat{\lambda}$. Let $\mathcal{E} := \{\sup_{\lambda\in\widehat{\Lambda}} R(\lambda) \leq \alpha\}$; by (ii), $\mathbb{P}(\mathcal{E}) \geq 1-\delta$. On $\mathcal{E}$, the selection rule in Eq. 17 picks $\hat{\lambda} \in \widehat{\Lambda}$, so

$$R(\hat{\lambda}) \;\leq\; \sup_{\lambda\in\widehat{\Lambda}} R(\lambda) \;\leq\; \alpha. \tag{34}$$

Hence $\mathcal{E} \subseteq \{R(\hat{\lambda}) \leq \alpha\}$, and

$$\mathbb{P}\big(R(\hat{\lambda}) \leq \alpha\big) \;\geq\; \mathbb{P}(\mathcal{E}) \;\geq\; 1-\delta. \tag{35}$$

□

Remark 2. *The proof does* not *rely on the specific form of the selection rule in Eq. 17; any measurable rule that picks $\hat{\lambda} \in \widehat{\Lambda}$ preserves the guarantee.*

C Details of Experimental Setup

Details of Datasets and Models We evaluate BalanceRAG on three open-domain QA datasets, TriviaQA, SQuAD v2, and Natural Questions (NQ). All datasets are processed into a unified question-answer evaluation format, where each example contains a question, an answer, an id, and the corresponding prompt. The preprocessing stage filters out examples with empty questions, empty answers, non-ASCII questions or answers, or inputs that exceed the maximum context length of the model. The default maximum numbers of retained examples are 7000 for TriviaQA, 5000 for SQuAD v2, and 2000 for NQ. Since the cleaning results after generation may differ across models, the actual number of examples used in the experiments can vary slightly. We therefore report the final number of evaluated examples as N in Table 3. We use eight backbone LLMs, including Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Qwen3-4B, Qwen3-14B, Llama3.2-3B-Instruct, Llama3.1-8B-Instruct, Vicuna-v1.5-7B, and Vicuna-v1.5-13B. Table 3 reports the base error rates of the two branches before calibration, where Only denotes the error rate of the LLM-only branch, RAG denotes the error rate of the LLM-RAG branch, and $\Delta = Only - RAG$ denotes the change in error rate brought by retrieval.

Table 3: Branch-level base error rates. N denotes the number of examples. Only and RAG are the pre-calibration error rates of the LLM-only and LLM-RAG branches, and $\Delta = Only - RAG$ indicates the retrieval gain. Blue cells highlight the largest retrieval gain within each dataset.

Model	N	Only ↓	RAG ↓	Δ ↑
TriviaQA				
Qwen2.5-3B	5929	0.2739	0.1558	**0.1181**
Qwen2.5-7B	6196	0.1699	0.1298	0.0401
Qwen3-4B	6510	0.2167	0.1246	0.0921
Qwen3-14B	6063	0.1921	0.1070	0.0851
Llama3.2-3B	6248	0.2769	0.2130	0.0639
Llama3.1-8B	6365	0.2016	0.1475	0.0541
Vicuna-7B	4183	0.2417	0.1831	0.0586
Vicuna-13B	3036	0.1989	0.1607	0.0382
SQuAD				
Qwen2.5-3B	4475	0.2291	0.1321	0.0970
Qwen2.5-7B	4375	0.2027	0.1191	0.0836
Qwen3-4B	4557	0.2203	0.0898	0.1305
Qwen3-14B	4360	0.2122	0.0954	0.1168
Llama3.2-3B	4482	0.3135	0.1772	0.1363
Llama3.1-8B	4210	0.3192	0.1202	**0.1990**
Vicuna-7B	3461	0.3389	0.1410	0.1979
Vicuna-13B	3031	0.2748	0.1488	0.1260
NQ				
Qwen2.5-3B	1732	0.3816	0.1819	0.1997
Qwen2.5-7B	1755	0.3419	0.1744	0.1675
Qwen3-4B	1858	0.3617	0.1335	0.2282
Qwen3-14B	1807	0.3298	0.1384	0.1914
Llama3.2-3B	1808	0.3711	0.2163	0.1548
Llama3.1-8B	1792	0.3203	0.1479	0.1724
Vicuna-7B	1599	0.4953	0.1932	**0.3021**
Vicuna-13B	1196	0.3972	0.1898	0.2074

Details of Corpus Construction For each dataset, we build a dataset-level retrieval corpus from the raw evidence fields. Each document consists of a title and a passage. TriviaQA uses evidence from Wikipedia entity pages and web search results, SQuAD v2 uses the article title and con-

text passage, and Natural Questions uses the preprocessed long-answer passage or the majority-voted long-answer candidate. We remove documents with missing titles or passages, strip extra whitespace, and deduplicate exact title–passage pairs across the corpus. The resulting corpus is cached and reused for dense retrieval.

Details of Answer Generation and Retrieval All datasets use a one-shot prompt. The LLM-only branch provides only the question itself and one example question-answer pair. The LLM-RAG branch prepends retrieved evidence to the same question-answer format and asks the model to answer based on the evidence. The most likely answer is generated with beam search using a beam size of 5, without sampling. The sampled generations for uncertainty estimation are generated with sampling, where each input has 20 sampled answers, the temperature is 1.0, and top-p is 0.9. The default maximum generation length is 30 tokens. The RAG branch uses dense retrieval with all-MiniLM-L6-v2 as the retriever encoder. During retrieval, the document title and passage are concatenated and encoded, the embeddings are L2-normalized, and the documents are ranked by dot product. For each question, we use the top-4 retrieved documents by default, and each passage is truncated to at most 1200 characters.

Details of Correctness Criteria By default, we use semantic similarity as the correctness criterion. Specifically, we compute the similarity between the ground-truth answer and the most likely generation using a sentence-transformers cross-encoder, and set 0.7 as the default threshold. An example is treated as correct when its similarity score is no lower than this threshold. This correctness label is used to construct the error indicator in the calibration risk. In addition to the default setting, we also use entailment and LLM-as-a-Judge in the robustness experiments. The entailment setting uses DeBERTa-v3-large-mnli-fever-anli-ling-wanli for NLI-based judgment. The LLM-as-a-Judge setting uses Qwen3-14B as the judge and formulates the judgment as a three-way classification problem, where both correct and partial are treated as acceptable correct answers.

Details of Uncertainty Estimators The default uncertainty estimator is semantic entropy based on sampled answer clusters, denoted as SE. For each input, we sample 20 answers to compute the uncertainty score. In addition to SE, we also evaluate EigV, Deg, Ecc, and SELF. SE computes the entropy of the semantic-cluster distribution over sampled generations. EigV, Deg, and Ecc construct a graph from the pairwise similarity matrix of sampled generations and derive uncertainty scores from graph-structure statistics. SELF uses the length-normalized sentence entropy of the most likely generation.

Details of Baselines We provide three algorithms to clarify the calibration procedures of different baselines. Algorithm 1 describes single-branch calibration, which is used for LLM-only and LLM-RAG. These methods use the uncertainty score from a single branch and search over a one-dimensional threshold space for a feasible point that satisfies the target risk. Algorithm 2 describes UCB-Cascaded. This method first calibrates the LLM-only threshold on the full calibration set, then calibrates the RAG threshold on the residual subset not accepted by the LLM-only branch. It splits δ equally between the two stages, with each stage using $\delta/2$. UCB-Cascaded includes two variants based on the Clopper–Pearson upper bound and the Hoeffding upper bound. Algorithm 3 describes BalanceRAG, which jointly calibrates the LLM-only and RAG branches on a two-dimensional threshold lattice and uses sequential graphical testing for high-probability risk control. We report two graph diffusion priors, UGD and DWD, where DWD is the default main method. Additional baselines in the appendix include Adaptive RAG, Self-Route, Empirical Risk $< \alpha$, and Bonferroni correction. Adaptive RAG routes queries using question-complexity labels. Self-Route routes queries using route-judge labels. Empirical Risk $< \alpha$ selects the threshold pair with the largest number of accepted examples within the empirically feasible region. Bonferroni correction applies a conservative adjustment by dividing δ by the number of tested nodes.

Algorithm 1 Single-branch calibration for LLM-only and LLM-RAG

Require: $\mathcal{D}_{\text{cal}} = \{(x_\ell, u_b(x_\ell), c_b(x_\ell))\}_{\ell=1}^n$, where u_b is uncertainty and $c_b = 1$ denotes correctness; branch $b \in \{\text{LLM}, \text{RAG}\}$; target risk α; error level δ
Ensure: threshold $\hat{t}$

$\mathcal{T} \leftarrow$ sorted unique values of $\{u_b(x_\ell)\}_{\ell=1}^n$
$\hat{t} \leftarrow \varnothing$
for $t \in \mathcal{T}$ in ascending order **do**
  $m \leftarrow |\{\ell : u_b(x_\ell) \le t\}|;\ w \leftarrow |\{\ell : u_b(x_\ell) \le t,\ c_b(x_\ell) = 0\}|$
  if $m = 0$ **then continue**
  end if
  $p(t) \leftarrow \mathbb{P}\big(\text{Bin}(m, \alpha) \le w\big)$
  if $p(t) \le \delta$ **then**
    $\hat{t} \leftarrow t$ ▷ keep scanning for a looser threshold
  else
    break ▷ monotone stopping
  end if
end for
return $\hat{t}$

Algorithm 2 UCB-Cascaded calibration

Require: $\mathcal{D}_{\text{cal}} = \{(x_\ell, u_1(x_\ell), u_2(x_\ell), c_1(x_\ell), c_2(x_\ell))\}_{\ell=1}^n$, where u_1, u_2 are branch uncertainties and $c_1, c_2 = 1$ denote correctness; target risk α; error level δ; bound type $B \in \{\text{CP}, \text{HFD}\}$
Ensure: thresholds $(\hat{t}_1, \hat{t}_2)$

Split the confidence budget: $\delta_{\text{stage}} \leftarrow \delta/2$
 # Risk upper-bound subroutine
function $\textsc{UpperBound}_B(m, w; \eta)$
  if $B = \text{CP}$ **then**
    return $\text{BetaInv}(1-\eta;\ w+1,\ m-w)$
  else
    return $\dfrac{w}{m} + \sqrt{\dfrac{\log(1/\eta)}{2m}}$
  end if
end function
 # Stage 1: calibrate t_1 on the full calibration set
$\mathcal{T}_1 \leftarrow$ sorted unique values of $\{u_1(x_\ell)\}_{\ell=1}^n$
$\hat{t}_1 \leftarrow -\infty$
for $t \in \mathcal{T}_1$ in ascending order **do**
  $m \leftarrow |\{\ell : u_1(x_\ell) \le t\}|;\ w \leftarrow |\{\ell : u_1(x_\ell) \le t,\ c_1(x_\ell) = 0\}|$
  $\bar{R}(t) \leftarrow \textsc{UpperBound}_B(m, w;\ \delta_{\text{stage}})$
  if $\bar{R}(t) \le \alpha$ **then**
    $\hat{t}_1 \leftarrow t$ ▷ largest feasible so far
  else
    break
  end if
end for
 # Stage 2: calibrate t_2 on the residual set
$\mathcal{R} \leftarrow \{\ell : u_1(x_\ell) > \hat{t}_1\}$ ▷ deferred samples
$\mathcal{T}_2 \leftarrow$ sorted unique values of $\{u_2(x_\ell) : \ell \in \mathcal{R}\}$
$\hat{t}_2 \leftarrow -\infty$
for $t \in \mathcal{T}_2$ in ascending order **do**
  $m \leftarrow |\{\ell \in \mathcal{R} : u_2(x_\ell) \le t\}|;\ w \leftarrow |\{\ell \in \mathcal{R} : u_2(x_\ell) \le t,\ c_2(x_\ell) = 0\}|$
  $\bar{R}(t) \leftarrow \textsc{UpperBound}_B(m, w;\ \delta_{\text{stage}})$
  if $\bar{R}(t) \le \alpha$ **then**
    $\hat{t}_2 \leftarrow t$
  else
    break
  end if
end for
return $(\hat{t}_1, \hat{t}_2)$

Algorithm 3 BalanceRAG calibration via SGT

Require: $\mathcal{D}_{\text{cal}}$; target risk α; error level δ; graph prior $\mathcal{P} \in \{\text{UGD}, \text{DWD}\}$
Ensure: thresholds $(\hat{t}_1, \hat{t}_2)$

 # Step 1: choose the initial node on a seed split
Discretize u_1, u_2 into grids of size $M \times N$, yielding lattice $\Lambda = \{\lambda_{i,j}\}$
Split $\mathcal{D}_{\text{cal}}$ into a seed split $\mathcal{D}_{\text{seed}}$ and an SGT split $\mathcal{D}_{\text{sgt}}$
for each $\lambda_{i,j} \in \Lambda$ **do**
  $M_{ij}^{\text{seed}} \leftarrow M_{\mathcal{D}_{\text{seed}}}(\lambda_{i,j});\ K_{ij}^{\text{seed}} \leftarrow K_{\mathcal{D}_{\text{seed}}}(\lambda_{i,j})$
  $p_{ij}^{\text{seed}} \leftarrow \mathbb{P}\big(\text{Bin}(M_{ij}^{\text{seed}}, \alpha) \le K_{ij}^{\text{seed}}\big)$
end for
$\mathcal{S}_{\text{seed}} \leftarrow \{\lambda_{i,j} : M_{ij}^{\text{seed}} > 0,\ p_{ij}^{\text{seed}} \le \delta\}$
if $\mathcal{S}_{\text{seed}} \ne \varnothing$ **then**
  $s \leftarrow \arg\max_{\lambda_{i,j} \in \mathcal{S}_{\text{seed}}} M_{ij}^{\text{seed}}$
else
  $s \leftarrow \arg\min_{\lambda_{i,j}} p_{ij}^{\text{seed}}$
end if

 # Step 2: compute formal node-wise p-values
for each $\lambda_{i,j} \in \Lambda$ **do**
  $M_{ij} \leftarrow M_{\mathcal{D}_{\text{sgt}}}(\lambda_{i,j});\ K_{ij} \leftarrow K_{\mathcal{D}_{\text{sgt}}}(\lambda_{i,j})$
  $p_{ij} \leftarrow \mathbb{P}\big(\text{Bin}(M_{ij}, \alpha) \le K_{ij}\big)$ ▷ Eq. 9
end for

 # Step 3: initialize budget and graph prior
$\delta_s \leftarrow \delta;\quad \delta_{i,j} \leftarrow 0$ for all $\lambda_{i,j} \ne s$
$\widehat{\Lambda} \leftarrow \varnothing$ ▷ set of certified thresholds
Set edge weights $g_{(i,j)\to u}$ from $\mathcal{P}$ with root node s (Eq. 11 or 12)

 # Step 4: sequential graphical testing
while $\exists\, \lambda_{i,j} \in \Lambda \setminus \widehat{\Lambda}$ such that $p_{ij} \le \delta_{i,j}$ **do**
  Choose any such $\lambda_{i,j}$ maximizing M_{ij}
  $\widehat{\Lambda} \leftarrow \widehat{\Lambda} \cup \{\lambda_{i,j}\}$ ▷ certify $\lambda_{i,j}$
  for each successor $u \in \mathcal{N}(i,j)$ with $\lambda_u \notin \widehat{\Lambda}$ **do**
    $\delta_u \leftarrow \delta_u + \delta_{i,j} \cdot g_{(i,j)\to u}$ ▷ propagate budget
  end for
  $\delta_{i,j} \leftarrow 0$
end while

 # Step 5: select the best certified threshold pair
$\hat{\lambda} \leftarrow \arg\max_{\lambda_{i,j} \in \widehat{\Lambda}} M_{ij}$ ▷ Eq. 17
return $\hat{\lambda} = (\hat{t}_1, \hat{t}_2)$

Prompt for LLM-Only QA

```
### System:
This is a bot that correctly answers questions.
### User:
{few-shot question}
### Assistant:
{few-shot answer}
### User:
{question}
### Assistant:
```

Prompt for RAG QA

```
### System:
This is a bot that correctly answers questions.
Answer the question based on the provided evidence.
### Evidence:
[Doc 1] Title: {title_1}
{passage_1}
[Doc 2] Title: {title_2}
{passage_2}
...
### User:
{few-shot question}
### Assistant:
{few-shot answer}
### Evidence:
{retrieved evidence}
### User:
{question}
### Assistant:
```

Prompt for LLM-as-a-Judge Correctness Evaluation

You are an expert evaluator for open-ended QA correctness.
Given a question, a ground-truth answer, and a model's answer, decide which option best describes the model's answer:

A. **correct** – semantically equivalent to the ground-truth answer.

B. **partial** – related and contains some correct information but is incomplete or partially wrong.

C. **incorrect** – not compatible with the ground-truth answer.

Respond by selecting exactly one of **A**, **B**, or **C**.

```
### Question:
{question}
### Ground truth answer:
{ground-truth answer}
### Model answer:
{model answer}
### Answer:
```

Figure 8: Prompt templates used for LLM-only QA, RAG QA, and LLM-as-a-Judge correctness evaluation.

D Additional Experimental Results

Risk Control Holds Across Datasets Figures 13 and 14 show the results of BalanceRAG on SQuAD v2 and NQ. Consistent with the TriviaQA results in the main text, BalanceRAG achieves stable risk control across different models and target risk levels, while its coverage is usually close to or even higher than that of LLM-RAG. More importantly, under stricter values of α, the UCB-Cascaded baselines often fail to find a feasible point, which suggests that calibrating the two branches separately can be overly restrictive under a tight risk budget. In contrast, BalanceRAG still returns feasible solutions, showing that system-level calibration preserves reliability across datasets with different retrieval difficulty.

Table 4: Effect of uncertainty quantification methods on BalanceRAG on TriviaQA. AUROC evaluates uncertainty quality on the LLM-only and RAG branches; coverage is reported after joint risk calibration.

Uncertainty	AUROC Only/RAG ↑	Cov. $\alpha = 0.10$ ↑	Cov. $\alpha = 0.14$ ↑	Cov. $\alpha = 0.18$ ↑
Deg	0.7476 / 0.6978	81.97	99.95	99.96
SELF	0.5839 / 0.6800	70.25	99.43	99.93
SE	0.7248 / 0.6544	84.11	99.96	99.96
EigV	0.7488 / 0.6987	82.87	99.95	99.95
Ecc	0.6900 / 0.6706	72.89	99.49	99.95

Correct Acceptances Remain Competitive Under Risk Control Tables 7, 8, and 9 compare the number of correctly accepted examples under the same target risk levels. The results show that the risk control of BalanceRAG is not obtained by simply rejecting a large number of examples. Instead, BalanceRAG preserves high utility while satisfying the risk constraint. On TriviaQA, BalanceRAG is often close to or better than RAG, and clearly outperforms UCB-Cascaded. On SQuAD v2 and NQ, RAG usually retains more correctly accepted examples, but BalanceRAG still maintains comparable utility and is more likely to remain feasible than the UCB-Cascaded baselines under strict values of α. These results show that BalanceRAG maintains utility close to RAG while providing feasible risk control.

Risk Control Is Stable Across Uncertainty Estimators Figure 9 and Table 4 examine whether BalanceRAG depends on a particular uncertainty estimator. Deg, SELF, SE, EigV, and Ecc provide different forms of uncertainty signals, and therefore affect sample ranking, calibrated thresholds, and final coverage. The results show that, although different estimators lead to different utility levels, BalanceRAG still maintains risk control near the target levels across these settings. This finding suggests that the reliability of the method mainly comes from the calibration procedure itself, rather than from any specific uncertainty score. In other words, the uncertainty estimator affects utility and acceptance size, but does not change the basic ability of BalanceRAG to achieve risk control.

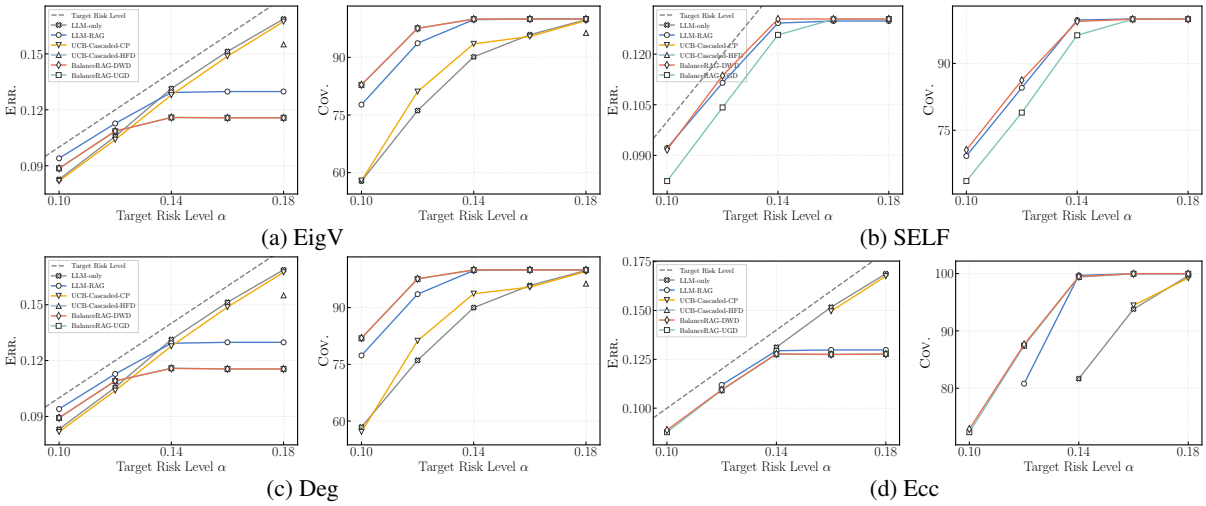
Table 5: Branch-level error rates under different correctness evaluation methods on TriviaQA.

Model	N	LLM-as-a-Judge		Entailment	
		Only ↓	RAG ↓	Only ↓	RAG ↓
Llama3.2-3B	6248	0.2623	0.2308	0.3782	0.2591
Llama3.1-8B	6365	0.1277	0.2687	0.2353	0.1983
Qwen2.5-3B	5929	0.2889	0.2663	0.3719	0.1881
Qwen2.5-7B	6196	0.1921	0.2515	0.2352	0.1451

Table 6: Reliability–coverage comparison on Llama3.1-8B with $\delta = 0.1$ over 500 data splits. Err., Cov., and Succ. denote mean accepted-answer risk, coverage, and guarantee success rate.

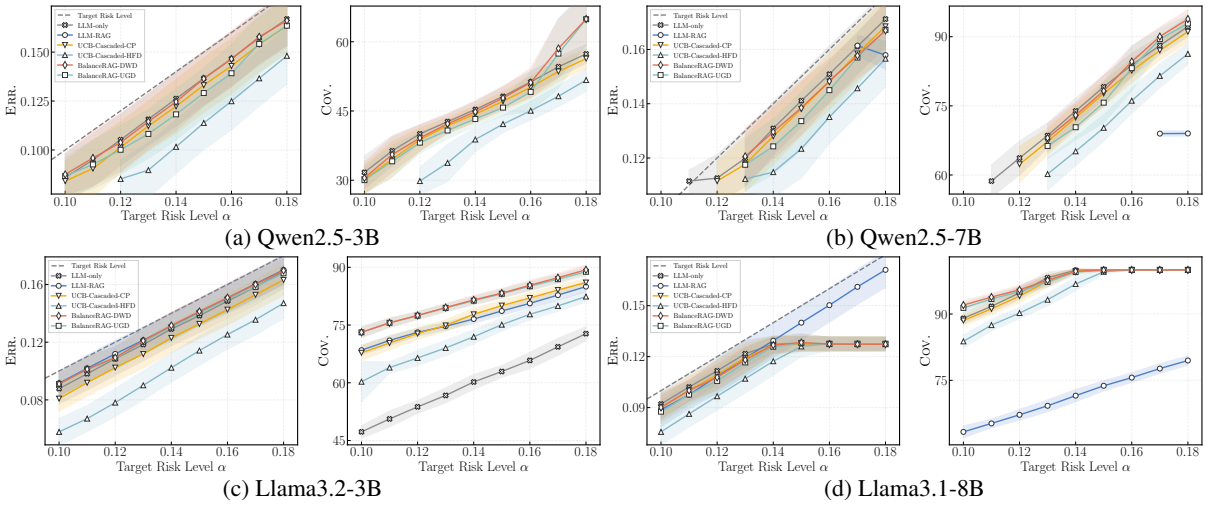
Method	α	Err. ↓	Cov. ↑	Succ. ↑
Adaptive-RAG (uncal.)	0.10	0.1516	100.0	0.0
	0.11	0.1516	100.0	0.0
	0.12	0.1516	100.0	0.0
Self-Route (uncal.)	0.10	0.1389	100.0	0.0
	0.11	0.1389	100.0	0.0
	0.12	0.1389	100.0	0.0
Empirical Risk $< \alpha$	0.10	0.1017	81.2	40.2
	0.11	0.1118	87.9	41.6
	0.12	0.1227	93.5	34.8
Bonferroni	0.10	0.0783	51.9	100.0
	0.11	0.0780	57.6	100.0
	0.12	0.0884	72.0	100.0
BalanceRAG	0.10	**0.0894**	**71.8**	**94.2**
	0.11	**0.0997**	**79.7**	**95.4**
	0.12	**0.1093**	**86.0**	**95.6**

Risk Control Is Stable Across Correctness Criteria Figures 10 and 11 evaluate BalanceRAG under LLM-as-a-Judge and entailment criteria. These two correctness criteria change which examples are treated as correct, and therefore affect the error signals observed during calibration and the final accepted set. The results show that BalanceRAG still maintains risk control under different correctness criteria, indicating that it does not rely on a single correctness definition. More importantly, BalanceRAG does not assume that one branch is always more reliable. Instead, it adjusts the accepted set and routing decisions according to the calibration results under the current evaluation criterion. Therefore, when the correctness criterion changes, the system can adapt through recalibration rather than being fixed toward either LLM-only or RAG.



(a) EigV (b) SELF (c) Deg (d) Ecc

Figure 9: Err. and Cov. on TriviaQA with Qwen2.5-7B using different UQ methods.



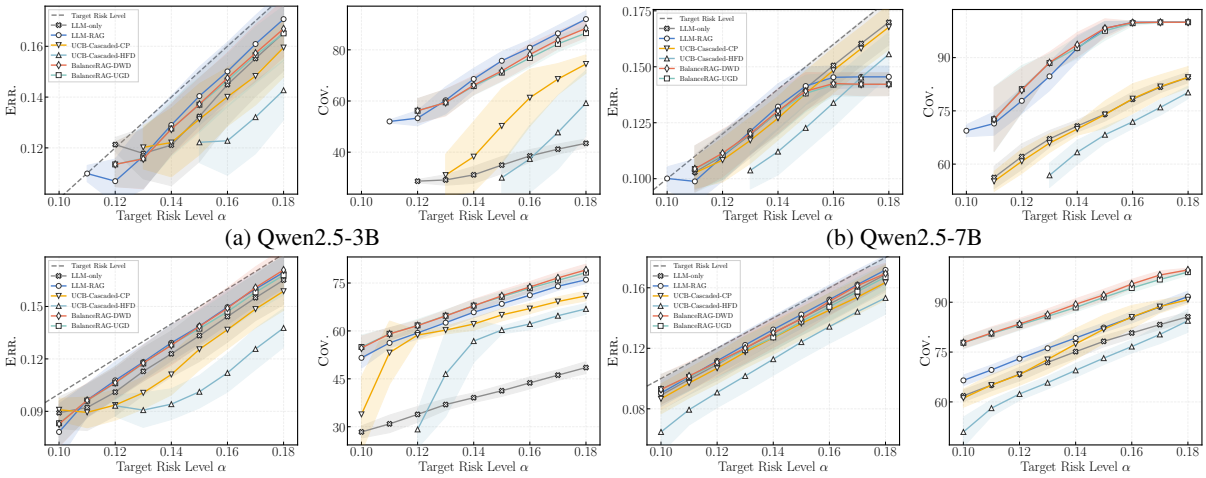
(a) Qwen2.5-3B (b) Qwen2.5-7B (c) Llama3.2-3B (d) Llama3.1-8B

Figure 10: Err. and Cov. on TriviaQA with LLM-as-a-Judge for correctness evaluation.

Calibration Size Mainly Affects Stability Figures 12 analyze the effect of the calibration-test split ratio. With fewer calibration examples, threshold estimation is more sensitive to random splits, and the results may fluctuate more. With more calibration examples, the estimates are generally more stable. In the experiments, the empirical risk of BalanceRAG consistently satisfies risk control as the split ratio changes. Moreover, under some settings of α, the variance of Err. decreases as the calibration size increases. These results show that the calibration size mainly affects stability, rather than changing the basic mechanism by which the method achieves risk control.

Routing Allocation Reflects Branch Complementarity Table 10 shows how accepted examples are allocated between the LLM-only and RAG branches. The allocation ratios vary across models and datasets, indicating that BalanceRAG neither always selects LLM-RAG nor always preserves LLM-only. In some settings, stronger models allow more examples to remain in the LLM-only branch, thereby reducing retrieval calls. In other settings, more examples are routed to RAG to use the additional information provided by retrieval. This variation shows that the two branches are complementary, and that BalanceRAG uses calibrated risk signals to decide when to trust each branch.

Multi-Risk Calibration Provides a Cost Control Knob Table 11 evaluates the multi-risk variant, where the second risk constraint limits the fallback invocation rate of RAG. Compared with single-risk routing, the multi-risk setting can explicitly control the frequency of RAG calls by adjusting α_2. When this constraint becomes stricter, the system reduces retrieval usage and may also lower overall coverage. This result should be understood as a controllable cost-coverage trade-off, rather than a cost-free performance gain. In practical deployment, this setting allows users to adjust system



(a) Qwen2.5-3B (b) Qwen2.5-7B

(c) Llama3.2-3B (d) Llama3.1-8B

Figure 11: Err. and Cov. on TriviaQA with entailment for correctness evaluation.

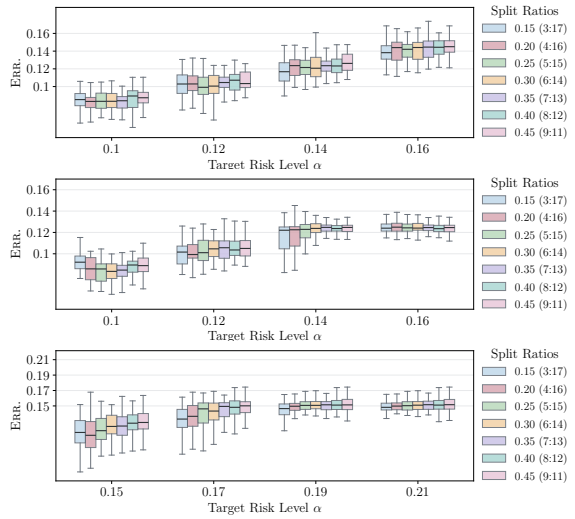
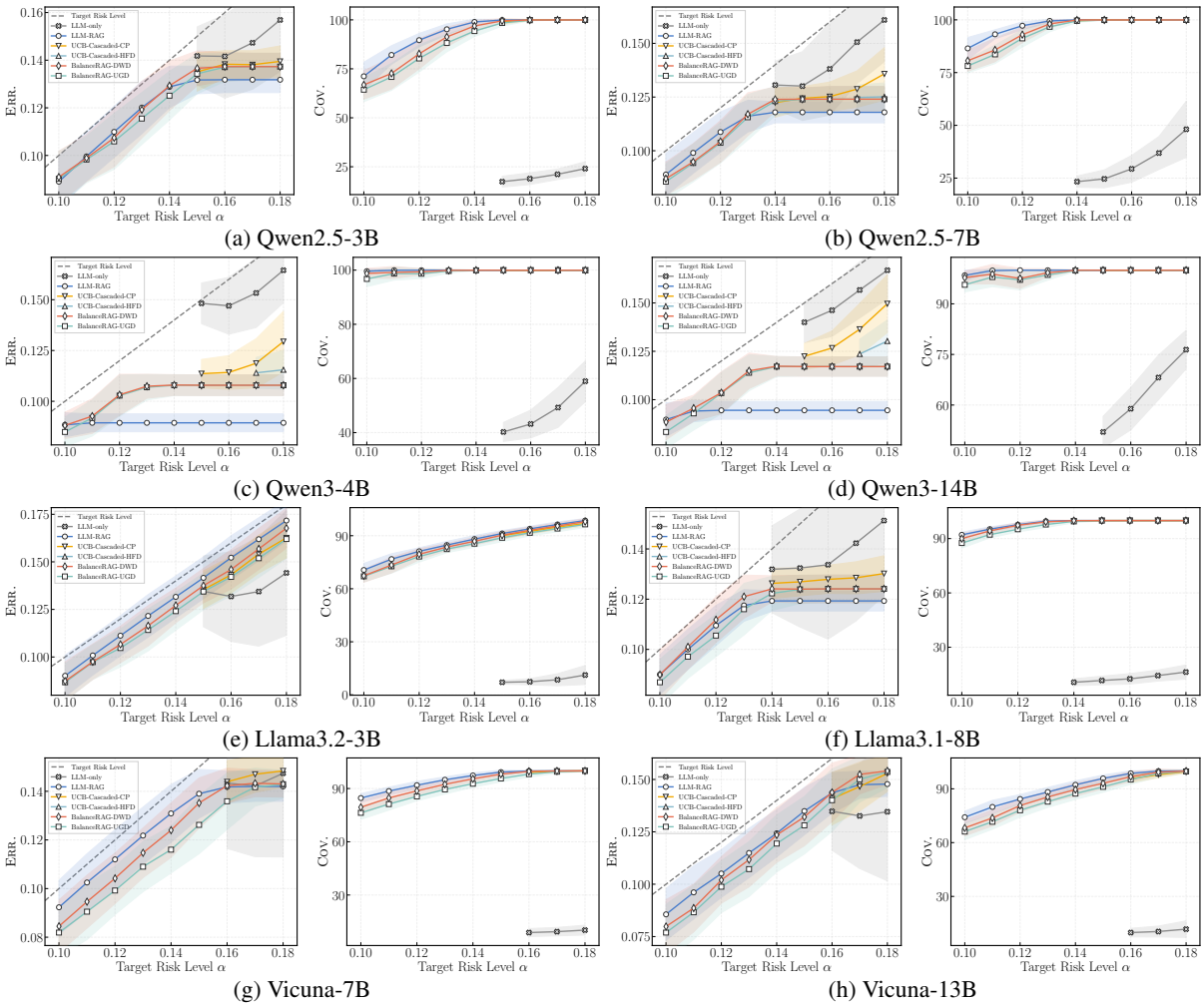


Figure 12: Risk control across various calibration-test split ratios. From top to bottom, the results are reported on TriviaQA with Vicuna-v1.5-7B, SQuAD v2 with Qwen2.5-7B, and NQ with LLaMA-3.1-8B.

behavior according to the retrieval budget while maintaining calibrated answer reliability.

Additional Baselines Table 6 compares BalanceRAG with Adaptive RAG, the Empirical Risk $< \alpha$ baseline, and Bonferroni correction. Adaptive RAG and the Empirical Risk $< \alpha$ baseline have mean risks above the target α and lower guarantee success rates, showing that simple adaptive retrieval or empirical-risk selection does not provide the desired risk control. Bonferroni correction is more reliable, but its coverage is much lower because the correction is more conservative. In contrast, BalanceRAG achieves mean risk below the target level, obtains a high guarantee success rate, and preserves higher coverage than Bonferroni correction. These results show that BalanceRAG strikes a better balance between high-probability reliability and coverage.

Qualitative Cases Explain Adaptive Routing Qualitative examples illustrate the routing behavior behind the aggregate results. In some cases, the LLM-only branch already produces the correct answer, while the retrieved content introduces misleading information. In such cases, BalanceRAG preserves the cheaper direct answer. In other cases, the LLM-only branch makes an error, while LLM-RAG corrects the answer by using external evidence, making routing to retrieval more appropriate. There are also cases where both branches are correct, suggesting that always invoking retrieval is not necessary. These cases do not serve as standalone statistical evidence, but they provide an intuitive explanation of the routing policy. The goal of BalanceRAG is not to favor a fixed branch, but to decide which output can be safely accepted under calibrated risk.



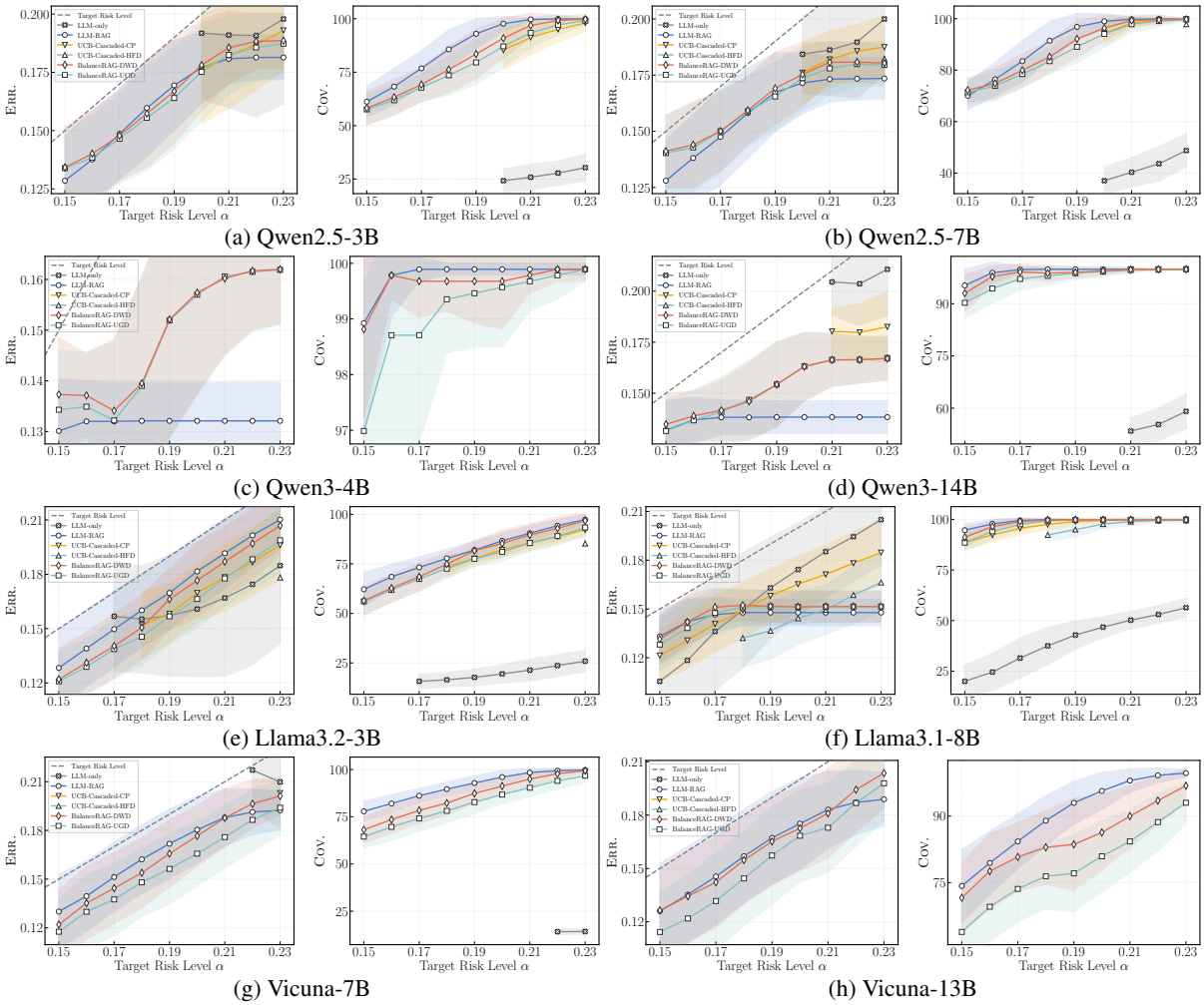
(a) Qwen2.5-3B (b) Qwen2.5-7B

(c) Qwen3-4B (d) Qwen3-14B

(e) Llama3.2-3B (f) Llama3.1-8B

(g) Vicuna-7B (h) Vicuna-13B

Figure 13: Err. (left in each panel pair) and Cov. (right) under different target risk levels α across eight LLMs spanning four model families on SQuAD v2.



(a) Qwen2.5-3B (b) Qwen2.5-7B

(c) Qwen3-4B (d) Qwen3-14B

(e) Llama3.2-3B (f) Llama3.1-8B

(g) Vicuna-7B (h) Vicuna-13B

Figure 14: Err. (left in each panel pair) and Cov. (right) under different target risk levels α across eight LLMs spanning four model families on NQ.

Table 7: **Correctly accepted samples comparison on TriviaQA.** We report the number of correctly accepted samples (Corr.) under different target risk levels α. Bold numbers indicate the best Corr. under the same LLM and risk level.

LLMs	**Methods / α**	**0.1**	**0.11**	**0.12**	**0.13**	**0.14**	**0.15**	**0.16**	**0.17**	**0.18**
Qwen2.5-3B	LLM-only	–	798	885	971	1049	1137	1227	1316	1397
	LLM-RAG	1473	1680	1907	2105	**2280**	2400	2475	2500	2501
	UCB-Cascaded-CP	–	1291	1633	1834	1952	2049	2143	2237	2304
	UCB-Cascaded-HFD	–	–	–	–	1555	1799	1955	2051	2152
	BalanceRAG	**1569**	**1733**	**1907**	**2110**	2279	**2421**	**2498**	**2512**	**2514**
Qwen2.5-7B	LLM-only	1856	1988	2148	2255	2347	2422	2473	2522	2564
	LLM-RAG	2118	2320	2490	2633	2691	2694	2694	2694	2694
	UCB-Cascaded-CP	1820	1992	2284	2377	2442	2499	2531	2534	2557
	UCB-Cascaded-HFD	–	1716	1880	2035	2205	2301	2392	2455	2502
	BalanceRAG	**2366**	**2573**	**2702**	**2737**	**2738**	**2738**	**2738**	**2738**	**2738**
Llama3.2-3B	LLM-only	870	988	1116	1214	1283	1371	1443	1513	1589
	LLM-RAG	**1367**	1390	1532	1646	1760	1876	1949	2036	2117
	UCB-Cascaded-CP	832	1028	1488	1749	1850	1917	1991	2065	2142
	UCB-Cascaded-HFD	–	–	820	964	1328	1661	1799	1899	1990
	BalanceRAG	1347	**1643**	**1774**	**1873**	**1973**	**2073**	**2159**	**2229**	**2290**
Llama3.1-8B	LLM-only	1587	1780	1928	2070	2165	2218	2277	2345	2402
	LLM-RAG	1856	2036	2198	2382	2529	2652	2711	2713	2713
	UCB-Cascaded-CP	1553	1849	2084	2264	2360	2434	2486	2505	2482
	UCB-Cascaded-HFD	–	1395	1607	1810	1992	2148	2248	2351	2400
	BalanceRAG	**2073**	**2288**	**2430**	**2584**	**2691**	**2727**	**2734**	**2736**	**2737**
Vicuna-7B-v1.5	LLM-only	792	868	935	992	1049	1115	1189	1250	1294
	LLM-RAG	1256	1330	1401	1455	1512	1564	1611	1648	1675
	UCB-Cascaded-CP	981	1102	1236	1306	1354	1395	1426	1469	1504
	UCB-Cascaded-HFD	542	603	776	876	1011	1147	1266	1333	1355
	BalanceRAG	**1297**	**1367**	**1430**	**1487**	**1533**	**1580**	**1629**	**1667**	**1694**
Vicuna-13B-v1.5	LLM-only	775	861	919	961	1002	1043	1081	1110	1138
	LLM-RAG	875	939	987	1041	1101	1163	1207	1250	1270
	UCB-Cascaded-CP	750	868	925	962	998	1031	1070	1100	1127
	UCB-Cascaded-HFD	–	669	694	806	888	946	989	1035	1078
	BalanceRAG	**983**	**1041**	**1091**	**1124**	**1168**	**1203**	**1237**	**1262**	**1274**
Qwen3-4B	LLM-only	–	1764	1871	1995	2093	2159	2233	2301	2357
	LLM-RAG	**2505**	**2627**	**2759**	**2835**	**2849**	**2849**	**2849**	**2849**	**2849**
	UCB-Cascaded-CP	–	1746	1834	2016	2126	2240	2463	2533	2587
	UCB-Cascaded-HFD	–	–	1726	1781	1902	2048	2133	2224	2338
	BalanceRAG	–	2549	2718	2807	2839	2843	2843	2843	2843
Qwen3-14B	LLM-only	–	–	1836	1964	2073	2176	2259	2314	2363
	LLM-RAG	**2521**	**2660**	**2706**	**2707**	**2707**	**2707**	**2707**	**2707**	**2707**
	UCB-Cascaded-CP	–	–	2310	2362	2467	2538	2533	2525	2497
	UCB-Cascaded-HFD	–	–	–	–	2058	2215	2359	2410	2420
	BalanceRAG	2499	2634	2629	2629	2660	2661	2661	2661	2661

Table 8: **Correctly accepted samples comparison on SQuAD v2.** We report the number of correctly accepted samples (Corr.) under different target risk levels α. Bold numbers indicate the best Corr. under the same LLM and risk level; underlined numbers indicate the second best.

LLMs	**Methods / α**	**0.1**	**0.11**	**0.12**	**0.13**	**0.14**	**0.15**	**0.16**	**0.17**	**0.18**
Qwen2.5-3B	LLM-only	–	–	–	–	–	333	362	402	452
	LLM-RAG	**1450**	**1653**	**1785**	**1873**	**1929**	**1942**	**1942**	**1942**	**1942**
	UCB-Cascaded-CP	–	–	–	–	–	1916	1927	1928	1925
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	1357	1464	1649	1798	1887	1922	1929	1930	1930
Qwen2.5-7B	LLM-only	–	–	–	–	443	468	551	682	878
	LLM-RAG	**1723**	**1836**	**1896**	**1924**	**1929**	**1929**	**1929**	**1929**	**1929**
	UCB-Cascaded-CP	–	–	–	–	1913	1915	1913	1906	1890
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	1914	1914
	BalanceRAG	1611	1698	1821	1895	1915	1916	1916	1916	1916
Llama3.2-3B	LLM-only	–	–	–	–	–	137	144	164	211
	LLM-RAG	**1438**	**1546**	**1617**	**1666**	**1714**	**1755**	**1784**	**1809**	**1829**
	UCB-Cascaded-CP	–	–	–	–	–	1744	1775	1798	1818
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	1377	1484	1591	1654	1699	1745	1781	1805	1828
Llama3.1-8B	LLM-only	–	–	–	–	194	211	227	257	288
	LLM-RAG	**1763**	**1803**	**1831**	**1850**	**1853**	**1853**	**1853**	**1853**	**1853**
	UCB-Cascaded-CP	–	–	–	–	1835	1837	1835	1834	1830
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	1723	1784	1819	1836	1842	1843	1843	1843	1843
Vicuna-7B-v1.5	LLM-only	–	–	–	–	–	–	130	138	150
	LLM-RAG	**1330**	**1376**	**1411**	**1443**	**1464**	**1479**	**1484**	**1484**	**1484**
	UCB-Cascaded-CP	–	–	–	–	–	–	1467	1473	1473
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	1258	1327	1374	1415	1447	1468	1479	1482	1482
Vicuna-13B-v1.5	LLM-only	–	–	–	–	–	–	128	136	152
	LLM-RAG	**1027**	**1094**	**1143**	**1183**	**1224**	**1256**	**1280**	**1291**	**1292**
	UCB-Cascaded-CP	–	–	–	–	–	–	1245	1266	1279
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	950	1018	1097	1149	1189	1224	1256	1276	1281
Qwen3-4B	LLM-only	–	–	–	–	–	781	838	948	1120
	LLM-RAG	**2069**	**2075**	**2075**	**2075**	**2075**	**2075**	**2075**	**2075**	**2075**
	UCB-Cascaded-CP	–	–	–	–	–	2019	2018	2008	1983
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	2019	2015
	BalanceRAG	2053	2050	2031	2032	2032	2032	2032	2032	2032
Qwen3-14B	LLM-only	–	–	–	–	–	974	1094	1251	1386
	LLM-RAG	**1953**	**1972**	**1973**	**1973**	**1973**	**1973**	**1973**	**1973**	**1973**
	UCB-Cascaded-CP	–	–	–	–	–	1912	1903	1882	1853
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	1910	1895
	BalanceRAG	1942	1948	1905	1914	1923	1924	1924	1924	1924

Table 9: **Correctly accepted samples comparison on NQ.** We report the number of correctly accepted samples (Corr.) under different target risk levels α. Bold numbers indicate the best Corr. under the same LLM and risk level; underlined numbers indicate the second best.

LLMs	**Methods / α**	**0.15**	**0.16**	**0.17**	**0.18**	**0.19**	**0.20**	**0.21**	**0.22**	**0.23**
Qwen2.5-3B	LLM-only	–	–	–	–	–	168	180	193	209
	LLM-RAG	**461**	**508**	**565**	**622**	**667**	**696**	**707**	**708**	**708**
	UCB-Cascaded-CP	–	–	–	–	–	607	645	669	685
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	435	470	510	554	600	646	683	697	701
Qwen2.5-7B	LLM-only	–	–	–	–	–	264	287	309	341
	LLM-RAG	535	**577**	**623**	**674**	**707**	**720**	**724**	**725**	**725**
	UCB-Cascaded-CP	–	–	–	–	–	682	706	712	712
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	703
	BalanceRAG	**544**	562	595	628	672	697	715	716	718
Llama3.2-3B	LLM-only	–	–	119	125	134	146	159	174	189
	LLM-RAG	**489**	**532**	**562**	**590**	**615**	**640**	**661**	**680**	**694**
	UCB-Cascaded-CP	–	–	–	563	590	615	635	653	669
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	633
	BalanceRAG	447	492	531	575	615	634	657	674	694
Llama3.1-8B	LLM-only	158	191	240	283	320	345	365	382	401
	LLM-RAG	**734**	**752**	**760**	**763**	**763**	**763**	**763**	**763**	**763**
	UCB-Cascaded-CP	698	718	736	743	747	744	740	735	730
	UCB-Cascaded-HFD	–	–	–	717	735	749	753	750	744
	BalanceRAG	709	741	753	757	759	759	759	759	759
Vicuna-7B-v1.5	LLM-only	–	–	–	–	–	–	–	88	90
	LLM-RAG	**541**	**564**	**584**	**600**	**615**	**629**	**639**	**643**	**644**
	UCB-Cascaded-CP	–	–	–	–	–	–	–	–	629
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	478	508	536	556	582	601	617	628	635
Vicuna-13B-v1.5	LLM-only	–	–	–	–	–	–	–	–	–
	LLM-RAG	**387**	**410**	**430**	**448**	**462**	**471**	**478**	**482**	**483**
	UCB-Cascaded-CP	–	–	–	–	–	–	–	–	–
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	373	400	413	418	416	427	439	449	460
Qwen3-4B	LLM-only	–	–	–	–	–	–	–	–	–
	LLM-RAG	**799**	**804**	**806**	**806**	**806**	**806**	**806**	**806**	**806**
	UCB-Cascaded-CP	–	–	–	–	–	–	–	–	–
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	792	799	802	797	785	780	779	778	777
Qwen3-14B	LLM-only	–	–	–	–	–	–	383	397	421
	LLM-RAG	**747**	**772**	**777**	**778**	**778**	**778**	**778**	**778**	**778**
	UCB-Cascaded-CP	–	–	–	–	–	–	740	741	738
	UCB-Cascaded-HFD	–	–	–	–	–	–	–	–	–
	BalanceRAG	727	761	769	763	757	752	752	753	752

Table 10: **Route allocation (%) of BalanceRAG.** We report the allocation ratio of accepted samples routed to the LLM-only branch (Only) and the RAG branch. Cov. denotes the total accepted ratio, i.e., Only+RAG .

Dataset	LLMs	$\alpha = 0.12$			$\alpha = 0.14$			$\alpha = 0.16$			$\alpha = 0.18$		
		Only	**RAG**	**Cov.**	**Only**	**RAG**	**Cov.**	**Only**	**RAG**	**Cov.**	**Only**	**RAG**	**Cov.**
TriviaQA	Qwen2.5-3B	27.7	44.5	72.2	29.9	58.3	88.2	30.7	68.4	99.1	31.1	68.9	100.0
	Qwen2.5-7B	62.2	35.8	98.0	64.1	35.8	99.9	64.1	35.8	99.9	64.1	35.8	99.9
	LLaMA-3.2-3B	37.1	26.7	63.8	41.1	31.4	72.5	41.6	39.7	81.3	41.5	46.7	88.2
	Vicuna-7B-v1.5	33.5	43.2	76.7	33.2	50.6	83.8	35.0	56.2	91.2	35.0	62.2	97.2
	Vicuna-13B-v1.5	51.9	28.2	80.1	54.5	33.2	87.7	54.7	40.1	94.8	54.7	45.0	99.7
	LLaMA-3.1-8B	51.2	34.5	85.7	53.3	44.1	97.4	62.1	37.8	99.9	63.1	36.8	99.9
	Qwen3-4B	32.4	61.7	94.1	58.7	41.2	99.9	58.8	41.2	100.0	58.8	41.2	100.0
	Qwen3-14B	29.8	67.7	97.5	64.0	35.9	99.9	64.1	35.8	99.9	64.1	35.8	99.9

Dataset	LLMs	$\alpha = 0.12$			$\alpha = 0.14$			$\alpha = 0.16$			$\alpha = 0.18$		
		Only	**RAG**	**Cov.**	**Only**	**RAG**	**Cov.**	**Only**	**RAG**	**Cov.**	**Only**	**RAG**	**Cov.**
SQuAD v2	Qwen2.5-3B	15.8	66.9	82.7	17.3	79.5	96.8	18.3	81.6	99.9	18.8	81.1	99.9
	Qwen2.5-7B	21.7	71.3	93.0	24.0	75.9	99.9	25.1	74.8	99.9	25.1	74.8	99.9
	LLaMA-3.2-3B	7.4	72.1	79.5	8.4	78.5	86.9	7.8	85.3	93.1	8.0	90.1	98.1
	Vicuna-7B-v1.5	6.8	81.9	88.7	6.4	89.0	95.4	6.4	93.2	99.6	6.5	93.4	99.9
	Vicuna-13B-v1.5	9.7	71.0	80.7	9.7	79.9	89.6	10.2	86.6	96.8	10.0	89.9	99.9
	LLaMA-3.1-8B	7.7	89.6	97.3	8.0	91.9	99.9	8.4	91.6	100.0	8.4	91.6	100.0
	Qwen3-4B	29.2	70.2	99.4	38.6	61.4	100.0	38.6	61.4	100.0	38.6	61.4	100.0
	Qwen3-14B	28.3	69.2	97.5	45.6	54.3	99.9	45.7	54.3	100.0	45.7	54.3	100.0

Dataset	LLMs	$\alpha = 0.17$			$\alpha = 0.19$			$\alpha = 0.21$			$\alpha = 0.23$		
		Only	**RAG**	**Cov.**	**Only**	**RAG**	**Cov.**	**Only**	**RAG**	**Cov.**	**Only**	**RAG**	**Cov.**
NQ	Qwen2.5-3B	13.8	55.4	69.2	20.0	63.5	83.5	20.9	76.0	96.9	20.7	79.1	99.8
	Qwen2.5-7B	17.8	62.2	80.0	25.0	67.2	92.2	34.9	64.4	99.3	36.6	63.2	99.8
	LLaMA-3.2-3B	15.1	53.4	68.5	14.3	67.4	81.7	17.5	72.0	89.5	18.0	78.8	96.8
	Vicuna-7B-v1.5	12.4	66.0	78.4	12.2	75.2	87.4	12.4	82.6	95.0	11.9	87.5	99.4
	Vicuna-13B-v1.5	1.9	78.8	80.7	13.2	70.3	83.5	19.6	70.3	89.9	23.5	73.3	96.8
	LLaMA-3.1-8B	22.4	76.6	99.0	24.5	75.3	99.8	25.1	74.7	99.8	25.1	74.7	99.8
	Qwen3-4B	1.9	97.7	99.6	26.4	73.3	99.7	39.8	60.0	99.8	43.2	56.7	99.9
	Qwen3-14B	5.1	93.9	99.0	25.0	74.1	99.1	46.1	53.7	99.8	48.7	51.1	99.8

Table 11: **Route allocation (%) of multi-risk BalanceRAG.** We report the allocation ratio of accepted samples routed to the LLM-only branch (Only) and the RAG branch. α_2 denotes the fallback invocation rate cap, chosen as the nearest lower multiple of 10% below the corresponding risk-only RAG allocation. Cov. denotes the total accepted ratio.

Dataset	**LLMs**	$\alpha = 0.12$				$\alpha = 0.14$				$\alpha = 0.16$				$\alpha = 0.18$			
		α_2	**Only**	**RAG**	**Cov.**	α_2	**Only**	**RAG**	**Cov.**	α_2	**Only**	**RAG**	**Cov.**	α_2	**Only**	**RAG**	**Cov.**
TriviaQA	Qwen2.5-3B	40	34.2	**34.0**	68.2	50	35.3	**46.8**	82.2	60	37.1	**57.7**	94.8	60	42.7	**57.1**	99.7
	Qwen2.5-7B	30	70.1	**25.0**	95.0	30	74.8	**25.1**	99.9	30	75.7	**24.2**	100.0	30	75.8	**24.1**	100.0
	LLaMA-3.2-3B	20	38.0	**0.4**	38.4	30	45.3	**25.5**	70.8	30	51.9	**25.7**	77.6	40	51.1	**35.1**	86.2
	Vicuna-7B-v1.5	40	40.6	**34.2**	74.8	50	37.9	**45.2**	83.1	50	42.9	**45.9**	88.8	60	40.5	**55.8**	96.3
	Vicuna-13B-v1.5	20	62.9	**13.1**	76.0	30	63.8	**21.3**	85.0	40	63.3	**30.5**	93.8	40	65.4	**33.8**	99.2
	LLaMA-3.1-8B	30	59.4	**24.1**	83.5	40	61.7	**33.4**	95.1	30	74.0	**25.8**	99.8	30	74.6	**25.4**	100.0
	Qwen3-4B	60	59.1	**31.2**	90.3	40	62.4	**36.7**	99.2	40	63.7	**36.3**	100.0	40	63.8	**36.2**	100.0
	Qwen3-14B	60	63.9	**30.0**	93.9	30	71.8	**27.2**	99.0	30	73.0	**27.0**	100.0	30	73.2	**26.8**	100.0

Dataset	**LLMs**	$\alpha = 0.12$				$\alpha = 0.14$				$\alpha = 0.16$				$\alpha = 0.18$			
		α_2	**Only**	**RAG**	**Cov.**	α_2	**Only**	**RAG**	**Cov.**	α_2	**Only**	**RAG**	**Cov.**	α_2	**Only**	**RAG**	**Cov.**
SQuAD v2	Qwen2.5-3B	60	19.4	**55.5**	74.9	70	25.1	**66.4**	91.5	80	24.9	**74.9**	99.8	80	25.4	**74.5**	99.9
	Qwen2.5-7B	70	25.6	**65.3**	90.9	70	33.1	**65.9**	99.0	70	35.1	**64.8**	99.9	70	35.6	**64.4**	100.0
	LLaMA-3.2-3B	70	10.4	**65.9**	76.3	70	18.8	**64.6**	83.5	80	17.4	**72.6**	90.0	90	10.3	**86.1**	96.4
	Vicuna-7B-v1.5	80	9.3	**77.1**	86.4	80	12.9	**77.2**	90.1	90	9.9	**88.2**	98.1	90	12.7	**87.2**	99.9
	Vicuna-13B-v1.5	70	11.1	**64.9**	76.0	70	18.0	**65.4**	83.4	80	17.2	**75.9**	93.0	80	21.6	**76.0**	97.5
	LLaMA-3.1-8B	80	16.4	**76.7**	93.0	90	12.3	**87.4**	99.7	90	13.9	**86.0**	99.9	90	14.1	**85.8**	99.9
	Qwen3-4B	70	38.3	**60.2**	98.5	60	44.1	**55.8**	99.9	60	44.5	**55.4**	100.0	60	44.7	**55.2**	100.0
	Qwen3-14B	60	44.4	**50.5**	94.9	50	52.8	**46.6**	99.4	50	54.5	**45.4**	99.9	50	54.7	**45.2**	100.0

Dataset	**LLMs**	$\alpha = 0.17$				$\alpha = 0.19$				$\alpha = 0.21$				$\alpha = 0.23$			
		α_2	**Only**	**RAG**	**Cov.**	α_2	**Only**	**RAG**	**Cov.**	α_2	**Only**	**RAG**	**Cov.**	α_2	**Only**	**RAG**	**Cov.**
NQ	Qwen2.5-3B	50	21.3	**39.7**	61.0	60	24.2	**50.8**	75.0	70	26.0	**63.8**	89.8	70	32.3	**64.2**	96.5
	Qwen2.5-7B	60	35.0	**42.9**	77.9	60	37.3	**49.6**	87.0	60	43.1	**52.4**	95.5	60	48.7	**50.5**	99.2
	LLaMA-3.2-3B	50	20.1	**41.6**	61.6	60	20.5	**53.7**	74.2	70	21.9	**63.0**	84.9	70	27.1	**63.3**	90.4
	Vicuna-7B-v1.5	60	15.7	**53.9**	69.5	70	16.4	**64.4**	80.8	80	15.3	**75.7**	91.0	80	18.5	**76.3**	94.8
	Vicuna-13B-v1.5	70	3.3	**63.2**	66.6	70	15.9	**57.7**	73.6	70	23.4	**60.9**	84.4	70	29.1	**62.3**	91.4
	LLaMA-3.1-8B	70	33.0	**61.6**	94.5	70	37.3	**61.9**	99.2	70	38.8	**61.0**	99.8	70	38.9	**60.9**	99.8
	Qwen3-4B	90	9.2	**80.1**	89.3	70	43.3	**55.4**	98.7	50	52.5	**44.5**	97.0	50	55.2	**44.1**	99.3
	Qwen3-14B	90	11.2	**78.0**	89.2	70	50.0	**47.4**	97.5	50	54.9	**44.0**	98.9	50	56.1	**43.6**	99.7

Case 1: LLM-only is correct while RAG is misled

ID: sfq_8482_5657 $\alpha = 0.10$ $t_1 = 0.286$ $t_2 = 0.748$

$u_{Only} = 0.286$ $u_{RAG} = 2.466$ **Route:** Only ✓

Question

Charleston is the capital of which US state?

Gold Answer

West Virginia

LLM-only Answer ✓

Charleston is the capital of West Virginia.

RAG Answer

Charleston is not the capital of any US state.

Case 2: LLM-only gives the exact answer while RAG rejects it

ID: qz_453_1207 $\alpha = 0.10$ $t_1 = 0.748$ $t_2 = 0.469$

$u_{Only} = 0.469$ $u_{RAG} = 2.020$ **Route:**Only ✓

Question

For what novel did J. K. Rowling win the 1999 Whitbread Children's book of the year award?

Gold Answer

Harry Potter and the Prisoner of Azkaban

LLM-only Answer ✓

Harry Potter and the Prisoner of Azkaban.

RAG Answer

She did not win the 1999 Whitbread Children's book of the year award.

Figure 15: Qualitative examples where BalanceRAG preserves the LLM-only branch. In both examples, the LLM-only answer is correct and passes the first-stage uncertainty threshold, while retrieval introduces misleading evidence and produces an incorrect RAG answer.

Case 3: RAG corrects a wrong entity generated by LLM-only

ID: bt_4365_238 $\alpha = 0.10$ $t_1 = 0.286$ $t_2 = 0.748$

$u_{Only} = 4.222$ $u_{RAG} = 0.286$ **Route:** RAG ✓

Question

Who in verse sailed to the land where the bong tree grows?

Gold Answer

The owl and the pussycat

LLM-only Answer

Alfred Noyes in his poem "The Highwayman" sailed to the land where the bong tree grows.

RAG Answer ✓

The Owl and the Pussy Cat sailed to the land where the bong tree grows.

Case 4: RAG fixes an incorrect surname answer

ID: dpql_4593_2651 $\alpha = 0.10$ $t_1 = 0.286$ $t_2 = 0.748$

$u_{Only} = 3.304$ $u_{RAG} = 0.286$ **Route:** RAG ✓

Question

What is the surname of the title character of Anne of Green Gables?

Gold Answer

Shirley

LLM-only Answer

Cuthbert is not the surname of the title character of Anne of Green Gables.

RAG Answer ✓

The surname of the title character of Anne of Green Gables is Shirley.

Figure 16: Qualitative examples where BalanceRAG routes to the RAG branch. The LLM-only branch has high uncertainty and gives an incorrect answer, while the RAG branch has lower uncertainty and produces the correct answer.

Case 5: Both branches are correct, but LLM-only is selected earlier

ID: dpql_6316_2767 $\alpha = 0.10$ $t_1 = 0.286$ $t_2 = 0.748$

$u_{Only} = 0.286$ $u_{RAG} = 2.139$ **Route:** Only ✓

Question

Who co-starred with George Clooney in the 2013 movie Gravity?

Gold Answer

Sandra Bullock

LLM-only Answer ✓

Sandra Bullock co-starred with George Clooney in the 2013 movie Gravity.

RAG Answer

Sandra Bullock co-starred with George Clooney in the 2013 movie Gravity.

Case 6: Both branches answer correctly, so the cheaper LLM-only route is sufficient

ID: qf_3204_522 $\alpha = 0.10$ $t_1 = 0.286$ $t_2 = 0.748$

$u_{Only} = 0.286$ $u_{RAG} = 2.020$ **Route:** Only ✓

Question

If you were eating a Knickerbocker Glory, what would you be eating?

Gold Answer

An ice-cream sundae

LLM-only Answer ✓

A Knickerbocker Glory is a type of ice cream sundae.

RAG Answer

A Knickerbocker Glory is a type of ice cream sundae.

Figure 17: Qualitative examples where both branches produce correct answers, but BalanceRAG selects the cheaper LLM-only route. These cases illustrate that always invoking RAG is unnecessary when the LLM-only branch is already sufficiently confident.