AdaCache: Efficient and Robust Noise-Resistant Semantic Caching for RAG

Shulin Chen
Department of Artificial Intelligence
College of Informatics
Korea University
Seoul, South Korea
chenshulin@korea.ac.kr

Lintong Zhang
Department of Artificial Intelligence
College of Informatics
Korea University
Seoul, South Korea
zhanglintong@korea.ac.kr

Seong-Whan Lee*
Department of Artificial Intelligence
College of Informatics
Korea University
Seoul, South Korea
sw.lee@korea.ac.kr

***Abstract*—Despite the success of Retrieval-Augmented Generation (RAG), existing frameworks inherently suffer from redundant reasoning for semantically similar queries and excessive sensitivity to input perturbations. To address these universal limitations, this paper presents AdaCache, a dual-mode semantic caching framework designed to transform stateless RAG pipelines into robust, stateful systems. Demonstrated within the state-of-the-art Probing-RAG architecture, our approach employs a Dual-Mode Selective Persistence policy. Designed to bypass expensive model re-computation for recurring intents, AdaCache selectively persists internal knowledge when retrieval is deemed unnecessary (Condition 1) and verified outcomes from successful retrieval (Condition 2). Furthermore, to overcome the fragility of static similarity thresholds in noisy environments, we introduce a Local-Density Adaptive strategy via local background analysis. By assessing the Signal-to-Noise Ratio (SNR) of the top candidate against local background noise, our framework ensures dynamic and robust hit detection. Experimental results on three open-domain QA benchmarks demonstrate that AdaCache reduces end-to-end latency by over 30% and storage overhead by up to 85% while maintaining superior accuracy under input perturbations compared to state-of-the-art baselines.**

***Index Terms*—Retrieval-Augmented Generation, Semantic Caching, Self-Probing, Noise-Resistant RAG, Robustness**

I. Introduction

Large Language Models (LLMs) have demonstrated exceptional capabilities across a wide range of tasks [1], [2], yet they remain prone to severe hallucinations that can lead to erroneous outputs [3]. To mitigate this issue, prior work has proposed Retrieval-Augmented Generation (RAG), which incorporates external knowledge retrieval to guide the generation process and thereby reduce hallucinations [4].

Typical RAG pipelines involve retrieval, augmentation, and generation [5], but they face efficiency challenges because repeated reasoning for similar queries incurs significant computational overhead [6]. State-of-the-art RAG optimizations focus on adaptive retrieval, such as Probing-RAG [7], which uses internal LLM states to decide retrieval necessity and addresses the limits of Adaptive-RAG [8]. However, such methods still exhibit structural limitations, most notably the fragility of probing-based strategies to input perturbations. As shown in Fig. 1 (Left), even slight surface-level noise (e.g., misspelling "True Colors" as "True Coiros") can cause substantial drift in the LLM's internal representations/activations, misleading the prober to predict "No Retrieval" and ultimately producing a "confident but wrong" generation. This suggests that, under inevitable real-world input noise, relying solely on internal model states for retrieval decisions lacks robustness and transferable reliability.

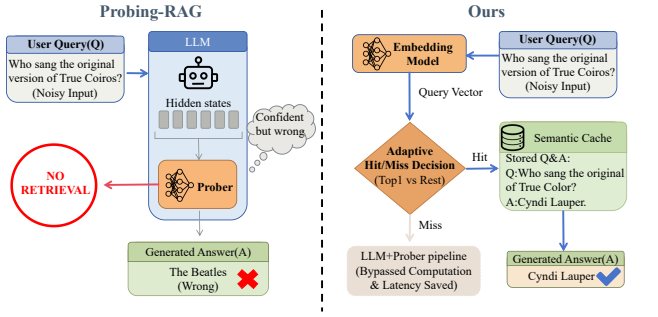


Fig. 1. Comparison between the baseline Probing-RAG (Left) under noisy conditions and our Adaptive Cache System (Right). As shown, the baseline prober may erroneously predict "No Retrieval" due to input noise (e.g., the misspelled "True Coiros"), leading to a "confident but wrong" answer. In contrast, our system robustly identifies the semantic match in the cache and returns the verified correct answer.

Semantic caching offers a potential solution by reusing prior answers [9]. However, most methods like GPTCache [10] use static similarity thresholds, leading to excessive storage costs and noise sensitivity. Newer approaches like vCache [11] selectively cache verified responses but lack deep RAG integration [12], limiting adaptability in knowledge-intensive tasks [13].

To bridge this gap, we propose **AdaCache**, a dual-mode semantic caching framework that jointly optimizes caching and retrieval decisions by exploiting the stability of semantic vector spaces to counteract the instability of internal model states. The framework consists of three tightly coupled components. **(i)** An Adaptive Cache Hit System robustly handles noisy inputs by assessing local SNR, distinguishing high-confidence cache hits from noise, recovering underlying semantic intent, and aligning perturbed queries with verified cache entries, thereby bypassing the costly and noise-sensitive probing stage when reuse is reliable. **(ii)** A Probing-RAG Core [7] preserves

knowledge adaptivity by triggering retrieval only when internal knowledge is insufficient. **(iii)** A Dual-Mode Selective Persistence strategy selectively updates the vector database using verified-only reasoning paths, transforming transient probing signals into persistent knowledge assets while ensuring a compact storage footprint (0.3–1.5 MB; up to ∼85% smaller than GPTCache [10]) without compromising reliability. Collectively, this closed-loop design transforms stateless probing into a robust, memory-augmented RAG system. Experiments on HotpotQA [14], NQ [15], and MuSiQue [16] demonstrate that AdaCache reduces end-to-end runtime by 30–40% while delivering superior robustness and consistently strong performance in practical deployments.

- **Framework Architecture:** We introduce AdaCache, a dual-mode semantic caching framework with Dual-Mode Selective Persistence, transforming stateless RAG into a robust, stateful system.
- **Noise-Resistant Mechanism:** We propose a Local-Density Adaptive Strategy using Signal-to-Noise Ratio (SNR) analysis, replacing fragile static thresholds for robust hit detection.
- **Efficiency and Scalability:** Our system achieves superior resource efficiency, reducing end-to-end latency by over 30% and storage overhead by up to 85% on standard QA benchmarks.

II. Related Work

A. Adaptive Retrieval and Internal State Probing

RAG couples parametric memory with external knowledge to mitigate hallucinations and static training limitations in LLMs [2]–[6]. To reduce retrieval overhead, adaptive methods such as FLARE [17], DRAGIN [18], Self-RAG [19], and Adaptive-RAG [8] trigger retrieval using confidence, critique, or external classifiers. However, these methods remain memory-less and repeatedly execute reasoning for similar queries. Internal state probing instead decodes cognitive signals from intermediate layers. Azaria et al. [20] showed that hidden states reflect factual familiarity, and Probing-RAG [7] uses a lightweight prober for on-demand retrieval. Yet probing remains stateless, recalculating similar internal states without persisting high-value outcomes. Our work converts these transient signals into persistent caching decisions.

B. Semantic Caching Mechanisms

Semantic caching reduces LLM serving cost and improves throughput [12]. GPTCache [10] established a vector-similarity framework, but static thresholds are noise-sensitive and misrepresent semantic proximity in non-uniform embedding spaces. Subsequent methods such as vCache [11], PromptCache [21], RAGCache [9], and CRAG [22] improve reliability or retrieval efficiency, yet most treat caching as an isolated add-on and do not integrate retrieval logic [23] or internal-state signals for storage prioritization. In contrast, our framework combines Dual-Mode Selective Persistence with Local-Density-aware significance assessment to replace static thresholds and reduce storage overhead under noise.

III. Methodology

We propose AdaCache, a dual-mode framework transforming stateless RAG into a stateful, efficient, and noise-robust system. Unlike architectures re-executing redundant reasoning, AdaCache jointly optimizes inference computation and knowledge persistence [9], [12]. As shown in Fig. 2, the system comprises: (A) an Adaptive Cache Hit System using significance assessment against perturbations; (B) the Probing-RAG Core for retrieval assessment; and (C) Dual-Mode Selective Persistence strategy for a compact, high-fidelity repository. This framework effectively mitigates computational redundancy while defending against input noise.

A. Probing-RAG Architecture for Adaptive Retrieval

We integrate Probing-RAG [7] as our core component for retrieval assessment (Module B, Fig. 2). Following the protocol in [7], the system determines whether internal parametric memory suffices by decoding intermediate hidden signals.

Specifically, for a query Q, the LLM generates a sequence of intermediate hidden states $\mathbf{H}^{(l)} = [\mathbf{h}_1^{(l)}, \ldots, \mathbf{h}_u^{(l)}]$ at layer l, where u denotes the combined length of the rationale and answer. A summary representation $\mathbf{z}^{(l)}$ is derived by mean-pooling followed by L_2 normalization to ensure consistency across layers:

$$\mathbf{z}^{(l)} = \frac{\sum_{i=1}^{u} \mathbf{h}_i^{(l)}}{\left\| \sum_{i=1}^{u} \mathbf{h}_i^{(l)} \right\|_2}. \tag{1}$$

The vector $\mathbf{z}^{(l)}$ is fed into layer-specific probers to assess knowledge boundaries. Based on multi-layer aggregation strategy in [7], we derive the final internal confidence score by aggregating signals across a predefined set of intermediate layers l:

$$\mathcal{S}_{\text{conf.}} = \sum_{l \in \mathcal{L}} \left(\mathbf{W}_2^{(l)} \cdot \sigma(\mathbf{W}_1^{(l)} \mathbf{z}^{(l)} + \mathbf{b}_1^{(l)}) + b_2^{(l)} \right). \tag{2}$$

If $\mathcal{S}_{\text{conf.}} < \theta$, external retrieval of documents is triggered. In AdaCache, while $\mathcal{S}_{\text{conf.}} \geq \theta$, the query is identified as meeting Condition 1, and its response is persisted to the semantic cache to bypass future re-computation.

B. Dual-Mode Selective Persistence Strategy

To reduce inference redundancy and storage overhead, we propose a Dual-Mode Selective Persistence (DMSP) strategy. Unlike indiscriminate "cache-all" methods [10], AdaCache adopts a selective policy triggered by two conditions: internal probing signals and posterior answer verification, as shown in Module C of Fig. 2. This converts transient internal decisions into a compact, high-fidelity state representation.

For Condition 1, we leverage the LLM's internal confidence. If the Probing-RAG prober predicts that no retrieval is needed ($\mathcal{S}_{\text{conf.}} \geq \theta$), the query can be answered using the model's internal knowledge alone. The resulting pair (Q, A) is then stored in $\mathcal{V}$ as an "internal-use" entry. Such pre-verified persistence allows future identical queries to bypass the full Prober-LLM forward pass [12], improving throughput.

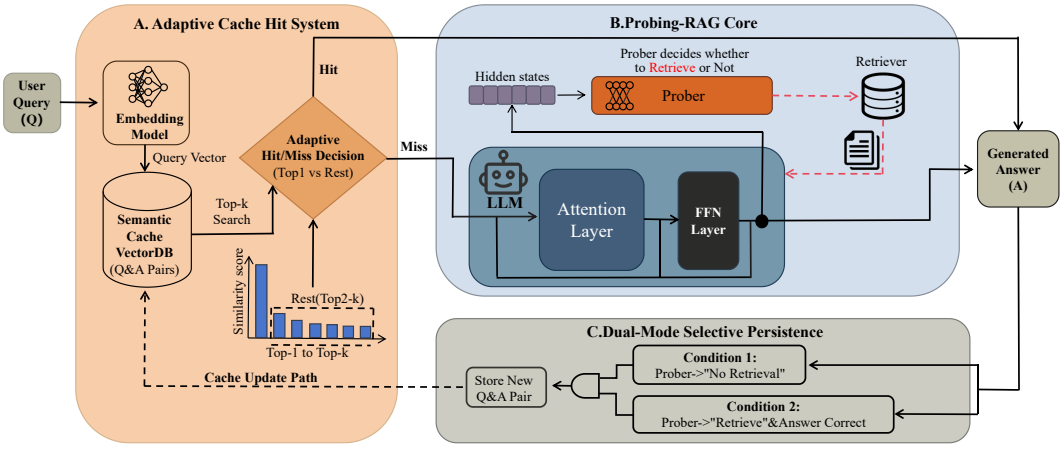


Fig. 2. The proposed AdaCache framework operates as a synergistic pipeline comprising three distinct, integrated stages: (A) the Adaptive Cache Hit System assesses local SNR to adaptively distinguish high-confidence hits from noise; (B) the Probing-RAG Core decodes LLM hidden states to adaptively trigger external retrieval only when internal knowledge is insufficient; and (C) the Dual-Mode Selective Persistence strategy ensures a compact storage footprint by selectively updating the vector database with verified-only reasoning paths. This closed-loop design transforms the stateless probing mechanism into a robust, memory-augmented system.

For Condition 2, we apply quality control to complex reasoning. When a query requires the full RAG pipeline [5], the pair (Q, A) is stored in $\mathcal{V}$ only if the answer A is verified through feedback (proxied by references). This selective mechanism filters out low-quality reasoning and failed retrievals. By storing only validated outcomes, the system achieves an approximate 85% reduction in storage overhead compared with conventional caches [9], [10], while maintaining the purity and reliability of the knowledge repository [11].

C. Adaptive Cache Hit System via Local SNR

A limitation of existing semantic caching is the sensitivity of static similarity thresholds to input noise [10], [21]. When a query contains typos, its similarity to the correct cache entry may fall below a fixed threshold, causing a false miss. To address this, we propose a local SNR-based scoring strategy for adaptive cache-hit determination.

Given a new query Q_{new}, the system retrieves the top-k most similar candidates from $\mathcal{V}$ with cosine similarity scores $\mathcal{S} = \{s_1, s_2, \ldots, s_k\}$. An absolute similarity gate ϵ is first applied to avoid matches in sparse or irrelevant embedding regions. If $s_1 < \epsilon$, the system directly declares a cache miss.

If s_1 passes the gate, we test whether it is a distinctive outlier relative to the local background. Let $\mathcal{S}_{bg} = \{s_2, \ldots, s_k\}$ and $n_{bg} = k - 1$. The mean μ_{bg} and standard deviation σ_{bg} are computed as

$$\mu_{bg} = \frac{1}{n_{bg}} \sum_{i=2}^{k} s_i, \quad \sigma_{bg} = \sqrt{\frac{\sum_{i=2}^{k} (s_i - \mu_{bg})^2}{n_{bg} - 1}}. \tag{3}$$

We then define the Significance Score (S) as

$$S = \frac{s_1 - \mu_{bg}}{\sigma_{bg} / \sqrt{n_{bg}}}, \tag{4}$$

which measures the local Signal-to-Noise Ratio (SNR) of the top candidate.

The system compares S with a heuristic threshold based on the local neighbor distribution. If $S > \gamma$, where γ is the significance limit, s_1 is treated as a high-confidence hit and the cached answer A_1 is returned directly, bypassing the downstream Probing-RAG Core. This adaptive criterion preserves the detectability of the correct entry under noise, while a stability check falls back to direct comparison when $\sigma_{bg} \approx 0$.

IV. Experiments

A. Datasets and Metrics

We utilized three representative open-domain QA datasets. Following the protocol in [7], we categorized them into in-domain (for prober training) and out-of-domain (unseen) to test generalization. **In-domain:** Natural Questions (NQ) [15] and HotpotQA [14]. **Out-of-domain:** MuSiQue [16], a complex multi-hop dataset requiring intricate reasoning chains. We evaluated the methods using the following metrics: **Performance:** Exact Match (EM) and Accuracy (Acc) to measure answer quality. **Efficiency:** Total inference time (s) for 500 queries to measure throughput. **Storage Overhead:** We measure space efficiency by comparing the Cache Storage Volume (disk usage) of AdaCache with the GPTCache baseline.

B. Baselines

We compare our approach against the following baseline methods. **LLM-based**: Decides retrieval [24] based on the model's response consistency, **FLARE** [17] & **DRAGIN** [18] Active methods using token generation confidence. **Adaptive-RAG** [8]: Routes queries via an external complexity classifier. **Probing-RAG** [7]: Our core backbone and primary performance baseline. **Probing-RAG+GPTCache** [10]: Employs a static thresholding strategy ($\tau = 0.7$).

TABLE I

COMPARATIVE EVALUATION OF VARIOUS METHODS UNDER IDENTICAL $\eta = 0.4$ NOISE INTERFERENCE ACROSS TWO LLM BACKBONES. ADACACHE EXHIBITS SUPERIOR ROBUSTNESS AND INFERENCE EFFICIENCY, ACHIEVING THE HIGHEST PERFORMANCE. (TOTAL TIME REPORTED IN SECONDS FOR THE FULL TEST SET).

Model	Methods	In-Domain						Out-of-Domain			Overall Performance		
		HotpotQA			NQ			MuSiQue			Average		
		EM	ACC	Total Time (s)	EM	ACC	Total Time (s)	EM	ACC	Total Time (s)	EM	ACC	Total Time (s)
Gemma-2b	LLM-based	16.5	23.4	5309	15.9	18.1	4945	3.5	4.2	6731	12.0	15.2	5662
	FLARE	12.1	19.1	12038	8.2	19.4	11212	1.0	4.4	15261	7.1	14.3	12837
	DRAGIN	17.6	20.5	30724	17.1	19.8	28614	4.0	4.1	38949	12.9	14.8	32762
	Adaptive-RAG	12.0	21.4	6946	10.1	23.4	6469	1.3	2.6	8806	7.8	15.8	7407
	Probing-RAG	21.0	37.6	4501	18.8	33.3	4192	3.6	7.6	5706	13.0	23.3	4800
	Probing-RAG + GPTCache	22.4	40.5	3049	18.5	33.7	2416	4.8	11.6	5543	15.3	28.6	3669
	Probing-RAG + AdaCache(Ours)	**23.6**	**41.5**	**2830**	**19.2**	**35.5**	**2193**	**6.4**	**14.2**	**4286**	**16.4**	**30.4**	**3103**
Mistral-7b	LLM-based	18.1	28.5	16458	14.4	31.8	14340	5.1	8.3	20866	12.5	22.9	17221
	FLARE	18.6	29.1	36114	13.6	30.7	33636	3.9	5.8	45783	12.0	21.9	38511
	DRAGIN	19.4	25.4	92172	15.1	33.2	85842	4.5	6.4	116847	13.0	21.7	98287
	Adaptive-RAG	17.2	23.1	20838	15.4	34.0	19407	3.7	5.1	26418	12.1	20.7	22221
	Probing-RAG	20.3	34.4	13953	18.4	35.8	12157	5.8	11.2	17118	14.8	27.1	14409
	Probing-RAG + GPTCache	23.9	41.3	9265	21.0	40.6	6588	8.2	19.8	16290	17.7	33.9	10714
	Probing-RAG + AdaCache(Ours)	**25.4**	**43.1**	**8340**	**21.5**	**42.2**	**6425**	**11.6**	**24.1**	**13140**	**19.5**	**36.5**	**9301**

C. Implementation Details

To ensure a fair comparison, all frameworks used BM25 [25] as the sparse retriever, with the same document corpus and preprocessing pipeline as in [7]. We implemented the LLM backbones—Gemma-2B [26] and Mistral-7B [27]—via HuggingFace Transformers [28], while the semantic cache employed the all-MiniLM-L6-v2 encoder [29] together with FAISS-accelerated vector search [30]. Regarding hyperparameters, the prober was trained for 2 epochs; for the AdaCache module, the candidate set size for significance appraisal was $k = 5$, the absolute similarity gate was $\epsilon = 0.5$, and the confidence threshold was $\gamma = 3.0$.

Noise Injection. To rigorously evaluate system robustness to realistic user typos, we additionally adopted a NoiseInjector to perturb input queries. Specifically, we swept noise levels from $\eta = 0.1$ to 1.0, where η indicates that approximately $\eta \times 100\%$ of query tokens are randomly corrupted via character-level typos, word deletions, and local shuffling. This protocol enables us to trace the performance degradation curve as input quality progressively deteriorates. All experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 5070 Ti GPU.

V. RESULTS AND ANALYSIS

In this section, we analyze the experimental findings from multiple perspectives, focusing on the synergy between internal probing and semantic memory. Specifically, we examine: (A) accuracy and inference latency under high-noise conditions ($\eta = 0.4$); (B) storage overhead across tasks of varying complexity; and (C) robustness as input noise increases from 0.1 to 1.0.

A. Overall Performance and Inference Efficiency

Table I compares Probing-RAG + AdaCache with competitive baselines under $\eta = 0.4$ noise. In terms of accuracy and generative quality, active retrieval frameworks such as FLARE [17] and DRAGIN [18] are highly fragile under noise: on Mistral-7B, they achieve only about 22% average ACC, because character-level perturbations distort token distributions and invalidate confidence calibration. Their reliance on dynamic token-level retrieval signals makes them vulnerable to surface noise, causing frequent erroneous retrieval and degraded answer quality.

TABLE II

COMPARATIVE ANALYSIS OF STORAGE FOOTPRINT ACROSS THREE DATASETS. ADACACHE ACHIEVES SIGNIFICANTLY LOWER STORAGE OVERHEAD COMPARED TO FULL-CACHING METHODS WHILE MAINTAINING PERFORMANCE ROBUSTNESS.

Dataset	GPTCache (Full)	AdaCache (Ours)	Reduction
HotpotQA	3154 KB	**1069 KB**	∼66.1%
NQ	4679 KB	**1461 KB**	∼68.8%
MuSiQue	2503 KB	**338 KB**	∼84.9%

By contrast, Probing-RAG + AdaCache maintains a clear advantage. On Mistral-7B, it reaches 36.5% average Accuracy, improving over stateless Probing-RAG [7] by 9.4% and outperforming static-threshold GPTCache [10]. The gain is most pronounced on the out-of-domain MuSiQue dataset, where AdaCache doubles performance. This suggests that Dual-Mode Selective Persistence serves as a correction layer: retrieving validated reasoning paths gives the LLM a cleaner second chance, while directly returning the final verified answer avoids the cumulative error risk of multi-step chain-of-thought reasoning under noisy inputs.

In efficiency, Table I shows that Probing-RAG + AdaCache is the fastest framework. On Mistral-7B, it reduces total inference time from 14,409 s to 9,301 s (35.4%). This comes from the caching layer's "short-circuit" effect, which bypasses expensive FFN and Attention forward passes through contrastive cache hits, in sharp contrast to DRAGIN, which exceeds 90,000 s due to iterative reformulations.

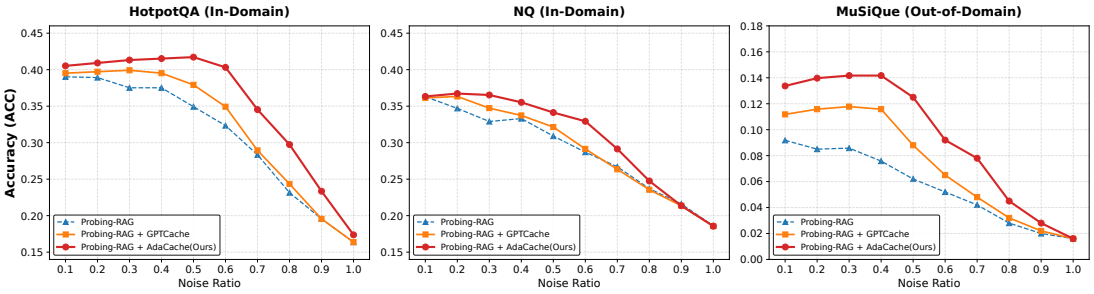


Fig. 3. Robustness analysis of caching strategies across varying noise ratios (0.1 to 1.0), primarily testing surface-level input typos. Left: HotpotQA (In-Domain); Center: NQ (In-Domain); Right: MuSiQue (Out-of-Domain). AdaCache (Red Line) consistently outperforms GPTCache (Orange Line) and Probing-RAG (Blue Dashed Line), showing superior resilience particularly after the 0.4 noise threshold.

B. Multi-Task Storage Overhead Optimization

Table II compares the disk storage footprints. Conventional strategies indiscriminately store low-quality reasoning debris, leading to rapid database inflation. Our dual-mode selective policy demonstrates superior efficiency across all task types:

In-Domain Stability (NQ & HotpotQA): We achieved storage reductions of 68.8% and 66.1%, respectively. This proves that even for familiar tasks, over two-thirds of the reasoning attempts are redundant or insufficient to warrant persistent storage. **Out-of-Domain Excellence (MuSiQue):** The reduction reaches an extreme 84.9%.

This cross-dataset trend confirms that more complex tasks produce more failed reasoning attempts in the RAG pipeline. By only persisting high-confidence intrinsic answers and verified results, AdaCache filters out this noise, maintaining a high-fidelity repository with only 1/4 to 1/6 of the hardware overhead of traditional methods.

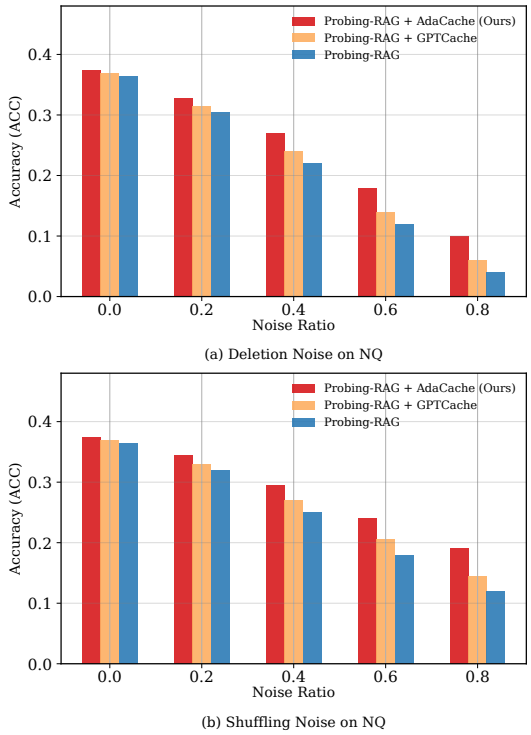
C. Robustness Evolution across Noise Spectrum

While Table I reports a single-point result, Fig. 3 tracks accuracy as the noise ratio η increases from 0.1 to 1.0.

As shown in Fig. 3 (Left and Center), GPTCache (orange) collapses once $\eta > 0.3$, because typos reduce absolute embedding similarity below the fixed $\tau = 0.7$ threshold, causing "False Misses." By contrast, our framework (red) maintains a high-performance plateau over 0.4–0.6. This verifies the effectiveness of the adaptive SNR policy: instead of relying on absolute scores, it measures the Signal-to-Noise Ratio (SNR) of the top candidate against background noise, allowing the decision boundary to adapt to noise-induced vector shifts.

The MuSiQue plot (Fig. 3 Right) further highlights this robustness in multi-hop reasoning. Because a noise-induced error in the first step typically derails the entire chain, robustness is especially critical. Even at high noise levels (0.5–0.8), AdaCache successfully matches noisy inputs to cached, verified multi-hop reasoning paths. This provides a "cognitive safety net" that preserves absolute leadership over baselines across the full noise spectrum, demonstrating viability for knowledge-intensive tasks.

To further validate the general robustness of AdaCache, we analyze Natural Questions (NQ), a representative open-domain QA task, under two corruption types: character deletion and local shuffling (Fig. 4). The results show that AdaCache's robustness advantage is largely independent of the specific noise mechanism. As η increases from 0.0, the performance gap between AdaCache and the baselines widens markedly. Probing-RAG + GPTCache (orange) drops sharply under both noise types (Fig. 4a and Fig. 4b), because its static similarity threshold ($\tau = 0.7$) cannot handle noise-induced vector drift. In contrast, Probing-RAG + AdaCache (red) maintains the highest accuracy across most of the η spectrum, especially outperforming both baselines up to $\eta = 0.6$ for all noise types,



(a) Deletion Noise on NQ

(b) Shuffling Noise on NQ

Fig. 4. Robustness Analysis on NQ under Different Noise Types. The figure compares performance on NQ under (a) Deletion Noise and (b) Shuffling Noise. AdaCache (Red) consistently maintains the highest accuracy across all noise ratios for both types of perturbations, demonstrating superior robustness compared to Probing-RAG and Probing-RAG + GPTCache.

confirming the effectiveness of the Local-Density Adaptive Strategy. By computing the Signal-to-Noise Ratio (SNR), AdaCache dynamically adjusts hit sensitivity, recovers semantic intent from corrupted inputs, and provides a reliable defense against diverse real-world typos.

VI. Discussion and conclusion

Experimental results show that AdaCache addresses the efficiency–robustness trade-off in RAG by coupling internal-state probing with a locally adaptive semantic cache. Dual-Mode Selective Persistence replaces "cache-all" with signal-aware filtering, storing only intrinsic knowledge shortcuts identified by the prober and verified reasoning paths confirmed by correct retrieval, thereby preventing error propagation and reducing storage overhead by up to 85% on MuSiQue while preserving a high-fidelity cache. Adaptive significance verification further improves noise robustness by replacing fixed absolute-similarity hit decisions with a contrastive, density-aware criterion: by computing the SNR between the top candidate and the background, AdaCache adapts the decision boundary to the noise floor and reduces false misses under degraded embeddings. Overall, AdaCache achieves over 30% end-to-end latency reduction and up to 85% storage savings without sacrificing accuracy; future work will extend it to multimodal RAG and accelerate iterative reasoning.

Acknowledgment

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant, funded by the Korea government (MSIT) (No. RS-2019-II190079, Artificial Intelligence Graduate School Program (Korea University), No. RS-2024-00457882, AI Research Hub Project.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[2] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.

[3] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–55, 2025.

[4] S. Wu, Y. Xiong, Y. Cui, H. Wu, C. Chen, Y. Yuan, L. Huang, X. Liu, T.-W. Kuo, N. Guan *et al.*, "Retrieval-augmented generation for natural language processing: A survey," *arXiv preprint arXiv:2407.13193*, 2024.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 9459–9474, 2020.

[6] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "A survey on rag meeting llms: Towards retrieval-augmented large language models," in *Proc. ACM SIGKDD*, 2024, pp. 6491–6501.

[7] I. Baek, H. Chang, B. Kim, J. Lee, and H. Lee, "Probing-rag: Self-probing to guide language models in selective document retrieval," in *Findings Assoc. Comput. Linguist.: NAACL 2025*, 2025, pp. 3287–3304.

[8] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, "Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity," *arXiv preprint arXiv:2403.14403*, 2024.

[9] C. Jin, Z. Zhang, X. Jiang, F. Liu, S. Liu, X. Liu, and X. Jin, "Ragcache: Efficient knowledge caching for retrieval-augmented generation," *ACM Trans. Comput. Syst.*, vol. 44, no. 1, pp. 1–27, 2025.

[10] F. Bang, "Gptcache: An open-source semantic cache for llm applications enabling faster answers and cost savings," in *Proc. Workshop Natural Language Processing Open Source Software (NLP-OSS 2023)*, 2023, pp. 212–218.

[11] L. G. Schroeder, A. Desai, A. Cuadron, K. Chu, S. Liu, M. Zhao, S. Krusche, A. Kemper, I. Stoica, M. Zaharia *et al.*, "vcache: Verified semantic prompt caching," *arXiv preprint arXiv:2502.03771*, 2025.

[12] X. Liu, B. Atalar, X. Dai, J. Zuo, S. Wang, J. Lui, W. Chen, and C. Joe-Wong, "Semantic caching for low-cost llm serving: From offline learning to online adaptation," *arXiv preprint arXiv:2508.07675*, 2025.

[13] J. Yan, W. Ni, L. Chen, X. Lin, P. Cheng, Z. Qin, and K. Ren, "Contextcache: Context-aware semantic cache for multi-turn queries in large language models," *arXiv preprint arXiv:2506.22791*, 2025.

[14] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in *Proc. EMNLP*, 2018, pp. 2369–2380.

[15] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee *et al.*, "Natural questions: a benchmark for question answering research," *Trans. Assoc. Comput. Linguist.*, vol. 7, pp. 453–466, 2019.

[16] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, " musique: Multihop questions via single-hop question composition," *Trans. Assoc. Comput. Linguist.*, vol. 10, pp. 539–554, 2022.

[17] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active retrieval augmented generation," in *Proc. EMNLP*, 2023, pp. 7969–7992.

[18] W. Su, Y. Tang, Q. Ai, Z. Wu, and Y. Liu, "Dragin: dynamic retrieval augmented generation based on the information needs of large language models," *arXiv preprint arXiv:2403.10081*, 2024.

[19] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-rag: Learning to retrieve, generate, and critique through self-reflection," 2024.

[20] A. Azaria and T. Mitchell, "The internal state of an llm knows when it's lying," *arXiv preprint arXiv:2304.13734*, 2023.

[21] X. Hou *et al.*, "Prompt cache: Modular attention reuse for low-latency inference," in *Proc. NeurIPS*, 2024.

[22] X. Yang, K. Sun, H. Xin, Y. Sun, N. Bhalla, X. Chen, S. Choudhary, R. D. Gui, Z. W. Jiang, Z. Jiang *et al.*, "Crag-comprehensive rag benchmark," *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 10 470–10 490, 2024.

[23] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, no. 1, 2023.

[24] Z. Zhang, M. Fang, and L. Chen, "Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering," *arXiv preprint arXiv:2402.16457*, 2024.

[25] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: Bm25 and beyond," *Found. Trends Inf. Retr.*, 2009.

[26] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love *et al.*, "Gemma: Open models based on gemini research and technology," *arXiv preprint arXiv:2403.08295*, 2024.

[27] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023. [Online]. Available: https://arxiv.org/abs/2310.06825

[28] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proc. EMNLP: Syst. Demos*, 2020, pp. 38–45.

[29] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 5776–5788, 2020.

[30] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, 2019.