

Feedback Adaptation for Retrieval-Augmented Generation

Jihwan Bang¹ Seunghan Yang¹ Kyuhong Shim^{2*} Simyung Chang^{3*}
Juntae Lee¹ Sungha Choi^{4*†}

¹Qualcomm AI Research[‡], Qualcomm Korea YH, Seoul, Republic of Korea

²Sungkyunkwan University ³H1R.ai ⁴Kyung Hee University

{jihwbang, seunghan, juntlee}@qti.qualcomm.com khshim@skku.edu

simyung@h1r.ai sunghac@khu.ac.kr

Abstract

Retrieval-Augmented Generation (RAG) systems are typically evaluated under static assumptions, despite being frequently corrected through user or expert feedback in deployment. Existing evaluation protocols focus on overall accuracy and fail to capture how systems adapt after feedback is introduced. We introduce *feedback adaptation* as a problem setting for RAG systems, which asks how effectively and how quickly corrective feedback propagates to future queries. To make this behavior measurable, we propose two evaluation axes: *correction lag*, which captures the delay between feedback provision and behavioral change, and *post-feedback performance*, which measures reliability on semantically related queries after feedback. Using these metrics, we show that training-based approaches exhibit a trade-off between delayed correction and reliable adaptation. We further propose **PatchRAG**, a minimal inference-time instantiation that incorporates feedback without retraining, demonstrating immediate correction and strong post-feedback generalization under the proposed evaluation. Our results highlight feedback adaptation as a previously overlooked dimension of RAG system behavior in interactive settings.

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as a dominant paradigm for grounding large language models (Dubey et al., 2024; Achiam et al., 2023; Team et al., 2024) in external knowledge. By coupling a retriever with a generator, RAG systems (Karpukhin et al., 2020; Yang et al., 2025; Wang et al., 2024; Lin et al., 2024; Sun et al., 2023) have demonstrated strong performance across question answering and knowledge-intensive tasks. However, existing RAG research

largely assumes that the underlying knowledge or system behavior remains static after deployment. In practice, deployed RAG systems are frequently corrected through user or expert feedback when they produce outdated, incorrect, or undesirable outputs. Despite its practical importance, how a RAG system should adapt after receiving feedback remains underexplored and poorly formalized.

Most prior approaches (Siriwardhana et al., 2023; Zhang et al., 2024; Mao et al., 2024) address feedback by retraining or fine-tuning components of the system, such as the retriever or the generator. While effective in improving accuracy in aggregate, these training-based updates introduce an inherent delay between the moment feedback is received and the moment the system’s behavior actually changes. Moreover, existing evaluation protocols (Petroni et al., 2021; Trivedi et al., 2022; Thakur et al., 2021) focus almost exclusively on overall task accuracy, making it difficult to reason about how quickly and reliably feedback influences future predictions. As a result, current RAG benchmarks and methods conflate correctness with adaptability, obscuring a critical dimension of system behavior in real-world deployments.

In this work, we argue that adapting to feedback is a distinct research problem that cannot be adequately studied through standard RAG evaluation. We introduce *feedback adaptation* as a new problem setting for RAG systems, and here is the research question: *once a system is corrected, how effectively and how quickly does this correction propagate to future queries?* Crucially, feedback adaptation is not about improving average accuracy, but about characterizing the dynamics of knowledge updates under interaction.

To make this problem explicit and measurable, we formalize two orthogonal evaluation axes. First, we define *correction lag*, which captures the delay between the time feedback is provided and the time at which the system’s outputs consistently

* Work completed while employed at Qualcomm AI Research.

† Corresponding author.

‡ Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

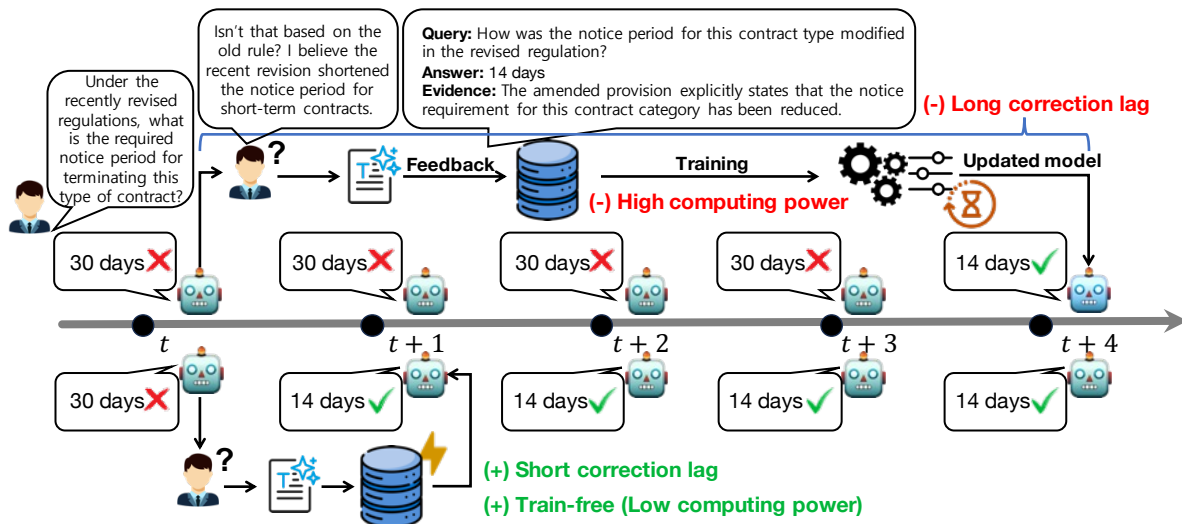


Figure 1: **Illustration of feedback adaptation in deployed RAG systems.** (Upper) Under training-based update paradigms, feedback is incorporated only after offline or periodic retraining, resulting in substantial correction lag. Even after expert feedback identifies an incorrect answer, the system may continue to reproduce the same error until the update is completed. (Lower) Inference-time feedback incorporation enables immediate behavioral correction by conditioning generation on stored feedback, achieving low correction lag and reliable generalization to semantically related queries.

reflect that feedback. As illustrated in Fig. 1, different update paradigms induce fundamentally different temporal behaviors after feedback is provided. Any approach that relies on retraining or redeployment necessarily incurs a significant correction lag, regardless of final accuracy. Second, we define *post-feedback performance*, which measures a system’s reliability on queries that are semantically consistent with the provided feedback. Unlike standard test accuracy, this metric isolates the effect of feedback and evaluates whether the system has meaningfully internalized the correction rather than merely memorizing labels.

Together, these two axes expose a trade-off that is largely invisible under existing evaluation protocols. Training-based methods can achieve strong performance, but only after incurring substantial correction lag. Conversely, systems that react immediately to feedback risk shallow or brittle generalization. We show empirically that this trade-off is not an artifact of specific models or datasets, but a structural property of how feedback is incorporated in current RAG paradigms.

To ground the feedback adaptation problem, we present **PatchRAG**, a minimal reference instantiation that incorporates feedback at inference time by storing and retrieving feedback patches. **PatchRAG** is intentionally simple, serving as a proof-of-concept that immediate adaptation with strong post-feedback performance is possible with-

out retraining. By evaluating **PatchRAG** alongside training-based baselines under the proposed metrics, we illustrate behaviors that are obscured by standard accuracy-driven evaluation.

Our contributions are threefold: (1) we introduce *feedback adaptation* as a new problem setting for RAG systems. (2) we formalize two evaluation metrics—*correction lag* and *post-feedback performance*—that capture complementary aspects of adaptation; and (3) we provide a minimal instantiation and empirical study, including stress tests under practically relevant imperfect-feedback conditions, that demonstrates the practical and conceptual implications of this problem formulation.

2 Feedback Adaptation

2.1 Motivation: Why Existing RAG Evaluation Is Insufficient

Retrieval-Augmented Generation (RAG) systems are typically evaluated under the assumption that model parameters and retrieval corpora remain fixed during evaluation. Under this setting, performance is measured by aggregate accuracy metrics such as EM or F1 over a held-out test set. However, this evaluation paradigm fails to capture a critical aspect of deployed RAG systems: they are routinely corrected after making mistakes.

When a system produces an incorrect or undesirable output, users or domain experts often provide

feedback in the form of corrected answers, supporting evidence, or clarifications. Once such feedback is given, the relevant question is no longer whether the system performs well on average, but whether it adapts its future behavior in response to that correction. Existing RAG benchmarks and metrics are not designed to answer this question, as they do not condition evaluation on the occurrence of feedback.

Crucially, this limitation is not merely an evaluation artifact. It reflects a deeper gap in how RAG systems are conceptualized: adaptation after feedback is implicitly treated as a training or maintenance concern, rather than as a first-class research problem. As a result, different system designs that behave identically under static evaluation can exhibit dramatically different behaviors once feedback is introduced.

2.2 Feedback Adaptation: Problem Definition

We formalize *feedback adaptation* as a problem setting for RAG systems that explicitly models how system behavior evolves after feedback. Consider a RAG system that receives a sequence of user queries, $q_u = \{q_1, q_2, \dots, q_t\}$. At time step t , the system produces an output a_t for query q_t . If the output is incorrect or unsatisfactory, an external agent provides feedback f_t , which may include a corrected answer or additional context. A system exhibits feedback adaptation if, after receiving feedback f_t , its subsequent outputs reflect that feedback when answering future queries that are semantically consistent with q_t .

Importantly, feedback adaptation does not require memorizing the exact feedback instance. Rather, it requires generalizing the correction to a set of related queries, potentially expressed with different surface forms or contexts. This distinguishes feedback adaptation from supervised learning, continual learning, and model editing, which focus on aggregate performance, retention of past knowledge, or targeted parameter updates, respectively. In contrast, feedback adaptation concerns the **temporal dynamics of system behavior under interaction**, conditioned on when and how feedback is received.

2.3 Evaluation Axis I: Correction Lag

The first axis of feedback adaptation is *correction lag*, which measures how quickly feedback influences system behavior. Given feedback f_t at time t , correction lag is defined as the elapsed time until the system consistently produces corrected outputs

that reflect the feedback for semantically consistent queries.

This notion captures a fundamental limitation of training-based approaches. Any method that relies on retraining, fine-tuning, or redeployment after feedback necessarily incurs a significant correction lag, regardless of the final accuracy achieved after the update. Correction lag is therefore orthogonal to accuracy: two systems may achieve identical post-update performance, yet differ substantially in how long they continue to produce incorrect outputs after feedback is given.

2.4 Evaluation Axis II: Post-Feedback Performance

While correction lag captures when a system adapts, it does not capture how well the system adapts once feedback has taken effect. We therefore introduce *post-feedback performance*. Post-feedback performance measures a system’s accuracy on queries that are semantically consistent with previously provided feedback, after the system has incorporated that feedback. Unlike standard test accuracy, this metric explicitly conditions evaluation on the presence of feedback and focuses on intent-consistent generalization. A system that simply memorizes the feedback instance without generalizing to related queries may exhibit low post-feedback performance.

Together with correction lag, post-feedback performance captures whether a system has meaningfully internalized feedback rather than treating it as an isolated exception.

3 PatchRAG

This section describes **PatchRAG**, a minimal instantiation of feedback adaptation introduced in [Sec. 2](#). **PatchRAG** incorporates feedback at inference time by storing and retrieving feedback patches, enabling immediate behavioral updates without retraining. The design avoids architectural modifications to isolate the effect of feedback incorporation under the proposed evaluation axes. As such, **PatchRAG** satisfies two requirements implied by feedback adaptation: (i) immediate incorporation of feedback, resulting in minimal correction lag; and (ii) generalization beyond individual feedback instances, enabling strong post-feedback performance on semantically related queries.

3.1 Intent-Context Retrieval for Feedback Adaptation

PatchRAG augments standard content-based retrieval with *intent-aware feedback retrieval*. Each stored feedback item is represented as a tuple $f_i = (q_i, a_i, c_i)$, consisting of the original query, the corrected answer, and supporting evidence.

Given a new query q , **PatchRAG** assigns each feedback item a relevance score

$$S_i(q) = \lambda \cdot \text{sim}(q, q_i) + (1 - \lambda) \cdot \text{sim}(q, c_i), \quad (1)$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity between embeddings, and $\lambda \in [0, 1]$ controls the balance between intent matching and context grounding.

Intent matching. The term $\text{sim}(q, q_i)$ captures similarity at the level of user intent, enabling retrieval of relevant feedback even when the query and corrected evidence share little surface or semantic overlap.

Context grounding. The term $\text{sim}(q, c_i)$ anchors retrieval to content-level relevance, preserving stability for queries whose intent has not been previously observed and preventing spurious matches driven solely by intent similarity.

The top- k feedback items selected by $S_i(q)$ are used as corrective exemplars for generation.

3.2 Inference with In-Context Learning

PatchRAG incorporates retrieved feedback directly into generation through in-context learning. Given the selected feedback items $\{p_{i_1}, \dots, p_{i_k}\}$, we construct an augmented prompt (see [Appendix A](#) for prompt template) that presents the feedback, followed by the current user query q .

This approach allows the generator to immediately condition its predictions on the newly provided feedback without any gradient-based updates or parameter modification. As a result, feedback can influence subsequent predictions as soon as it is added to memory, yielding low correction lag while maintaining strong post-feedback performance.

Scope. **PatchRAG** is intentionally minimal. Its purpose is not to serve as a final solution to feedback adaptation, but to provide a concrete reference point within the design space exposed by the proposed evaluation axes. By avoiding retraining, architectural changes, or specialized optimization objectives, **PatchRAG** isolates the effect of feedback incorporation itself.

4 Experiments

4.1 Feedback Adaptation Setup

As described in [Sec. 2](#), feedback adaptation evaluates how a RAG system responds to newly introduced feedback at a specific point in time within an otherwise continuous interaction stream. We operationalize this setting by examining system behavior immediately before and after feedback injection at time t .

At time t , the system is assumed to have accumulated feedback from prior interactions. To instantiate this pre-existing state in a controlled and reproducible manner, we populate the feedback memory with synthetic feedback generated as described in [Appendix B](#). This pre- t feedback serves only to stabilize retrieval and memory behavior and does not provide task-specific supervision relevant to the evaluation queries at time t .

To assess adaptation, we introduce expert feedback at time t . The injected feedback consists of paraphrased query–answer–context instances that share the underlying intent with the evaluation queries while exhibiting minimal lexical overlap. This design deliberately emphasizes semantic alignment over surface-level matching, thereby testing the system’s ability to bridge the semantic gap inherent in retrieval. By construction, this protocol avoids data leakage and prevents trivial keyword-based retrieval.

Snapshot Evaluation. Although feedback adaptation conceptually describes an online process, we employ a snapshot-based evaluation protocol to isolate the marginal effect of feedback injection. By comparing system behavior immediately before and after time t , this approach controls for confounding factors such as cumulative memory drift or changes in the query distribution. Snapshot evaluation enables precise measurement of *correction lag* and *post-feedback performance* on a well-defined set of intent-consistent but lexically distinct queries, providing a focused test of feedback adaptation under the proposed metrics.

4.2 Experimental Setup

Datasets. We evaluate feedback adaptation performance on three standard open-domain QA benchmarks: Natural Questions (NQ) ([Kwiatkowski et al., 2019](#)), TriviaQA ([Joshi et al., 2017](#)), and HotpotQA ([Yang et al., 2018](#)). Following common practice, we report Exact Match (EM) for NQ and

Method	Step	NQ	TriviaQA	HotpotQA	Average
Standard RAG	-	28.7	67.1	28.5	42.1
Self-RAG (Asai et al., 2024)	-	36.4	38.2	29.6	34.7
Auto-RAG (Yu et al., 2024a)	-	37.9	60.9	44.9	47.9
ChatQA-1.5 (Liu et al., 2024)	-	42.4	61.0	44.6	49.3
Golden-only	< t	35.3	75.5	43.7	51.5
RAFT (Zhang et al., 2024)	< t	36.4	76.5	44.6	52.5
PatchRAG	< t	36.2	76.8	44.8	52.6
Golden-only	t	40.2 (+4.9)	80.0 (+4.5)	48.1 (+4.4)	56.1 (+4.6)
RAFT (Zhang et al., 2024)	t	41.9 (+5.5)	80.5 (+4.0)	49.4 (+4.8)	57.3 (+4.8)
PatchRAG	t	49.8 (+13.6)	83.9 (+7.1)	53.2 (+8.4)	62.3 (+9.7)

Table 1: **Performance comparison under feedback adaptation across three QA benchmarks.** We report performance before feedback injection (< t) and after incorporating expert feedback at time t . **Green numbers** indicate post-feedback performance gains, measuring how effectively corrective feedback generalizes to semantically related queries. Results are reported on Natural Questions (NQ), TriviaQA, and HotpotQA using Llama-3 8B as the generator and bge-m3 as the retriever.

TriviaQA, and F1 for HotpotQA. Unless otherwise noted, all methods retrieve from the full Wikipedia corpus associated with each dataset. For more details, see Appendix C.

Feedback adaptation metrics. As described in Sec. 2.3 and Sec. 2.4, we evaluate feedback adaptation using two distinct metrics. *Correction lag* is measured as the wall-clock time from when feedback is provided until the system is ready to process subsequent evaluation queries using the updated state. *Post-feedback performance* is measured as the system’s performance (*i.e.*, EM or F1 score) at step t after the update, evaluated on queries that require the same correction implied by the feedback.

Implementation details. We compare **PatchRAG** against (i) non-feedback RAG enhancement methods such as Self-RAG (Asai et al., 2024), Auto-RAG (Yu et al., 2024a), and ChatQA-1.5 (Liu et al., 2024), and (ii) a feedback-driven adaptation baseline, RAFT (Zhang et al., 2024) (See Appendix D for training details). For fairness, all systems use the same Llama-3 8B generator (Dubey et al., 2024). For retrieval, we employ bge-m3 (Chen et al., 2024) as a retriever to generate embeddings for both user intents and evidence contexts within a shared vector space. Additionally, we populate the feedback memory with synthetic instances scaling to the full size of the retrieval corpus (*e.g.*, $\sim 150\text{K}$ for TriviaQA) to simulate a realistic deployed state. The weighting parameter λ in Eq. 1 is fixed to 0.5 across all experiments. All experiments are conducted on two NVIDIA A5000 GPUs and an Intel Xeon Gold 6342 CPU.

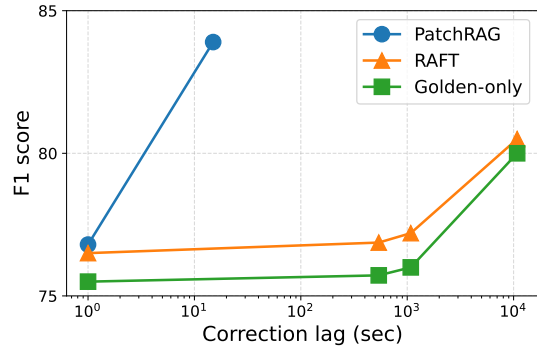


Figure 2: **Correction lag vs. Post-feedback Performance (F1) on TriviaQA.** This figure illustrates a structural trade-off inherent to training-based feedback incorporation, which inference-time methods mitigate. To investigate the trade-off between latency and performance, we evaluated training-based baselines (RAFT, Golden-only) under varying compute budgets by reducing update steps from full fine-tuning down to $\sim 5\%$ of steps (See Appendix E).

4.3 Experimental Results

We evaluate **PatchRAG** under a feedback adaptation setting to assess how efficiently it incorporates corrective feedback. Following the metrics introduced in Sec. 2.3 and Sec. 2.4, we analyze system behavior along two dimensions: (i) correction lag and (ii) post-feedback performance.

PatchRAG improves performance before expert feedback is introduced. As shown in Tab. 1, incorporating pre- t feedback stabilizes retrieval behavior, providing a stronger initial state for subsequent feedback adaptation. Even before expert feedback is injected, this stabilization leads to consistent performance gains over models that do not leverage

feedback. In particular, the dual intent-context retrieval mechanism reduces early-stage retrieval mismatch by better aligning retrieved evidence with the underlying query intent.

PatchRAG achieves the best post-feedback performance. We next evaluate post-feedback performance, indicated by the green numbers in Tab. 1, which measures how reliably a system produces corrected behavior after expert feedback is incorporated at step t . In this setting, **PatchRAG** achieves the highest absolute accuracy across all three QA benchmarks, with an average score of 62.3, surpassing training-based baselines such as Golden-only and RAFT. Furthermore, **PatchRAG** exhibits the largest post-feedback performance gain, with an average improvement of 9.7. This result validates post-feedback performance as a distinct evaluation axis, revealing differences that are obscured by standard accuracy metrics.

PatchRAG has the lowest correction lag while achieving the highest post-feedback performance. We finally analyze correction lag, defined as the time required for a system to reflect newly provided feedback in its predictions. As illustrated in Fig. 2, training-based methods (*e.g.*, RAFT) exhibit a fundamental trade-off: deliberately reducing the number of update steps to shorten correction lag results in a sharp degradation of post-feedback performance. Even under these strictly constrained training budgets, training-based approaches fail to match the immediate responsiveness required for feedback adaptation. Conversely, **PatchRAG** avoids this trade-off by incorporating feedback at inference time, achieving low correction lag without sacrificing post-feedback performance.

4.4 Robustness to Imperfect Feedback

To evaluate whether **PatchRAG** remains effective under more realistic feedback conditions, we further stress-test the quality of corrective feedback. In addition to the clean setting used in the main experiments, we consider five imperfect-feedback variants: replacing gold evidence with retrieved top-1 evidence, corrupting the answer field in a fraction of feedback instances (**Noise-X%**), omitting the answer entirely (**Blank**), providing the answer only implicitly in a verbose explanation (**Vague**), and storing both correct and incorrect feedback for the same intent (**Conflict**).

As shown in Tab. 2, replacing gold evidence with automatically retrieved top-1 evidence causes only

Answer	Evidence	NQ	TriviaQA	HotpotQA
G.T.	G.T.	49.8	83.9	53.2
G.T.	Top-1	49.2	81.6	52.5
Noise (25%)	Top-1	48.7	81.0	49.9
Noise (50%)	Top-1	44.1	78.0	42.6
Noise (75%)	Top-1	37.2	74.8	32.5
Blank	Top-1	33.2	77.2	37.1
Vague (Long)	Top-1	34.7	77.5	39.8
Conflict	Top-1	43.9	78.2	42.4

Table 2: **Stress-test results under imperfect feedback conditions.** We report EM on Natural Questions (NQ) and TriviaQA, and F1 on HotpotQA.

Method	NQ	TriviaQA	HotpotQA
Standard RAG	28.7	67.1	28.5
+ Retr. Eq. 1	45.6	77.8	47.6
+ Add feedback (ours)	49.8	83.9	53.2

Table 3: **Ablation study under feedback adaptation.** We incrementally introduce feedback-related components into a standard RAG system and evaluate post-feedback performance. Replacing standard retrieval scoring with the dual-similarity metric improves reliability by enabling intent-aware feedback retrieval. Adding feedback exemplars further strengthens post-feedback behavior, highlighting the necessity of exemplar-based conditioning for effective feedback adaptation.

a small drop (49.8→49.2 on NQ, 83.9→81.6 on TriviaQA, and 53.2→52.5 on HotpotQA), indicating that **PatchRAG** does not rely heavily on explicitly curated evidence and remains effective when feedback context must be retrieved automatically.

PatchRAG also degrades gradually as the corrective signal becomes noisier. Under answer poisoning, performance decreases monotonically from Noise 25% to Noise 75%, rather than collapsing abruptly, suggesting that the method tracks the strength of the available corrective signal and retains useful adaptation under moderate corruption.

The most challenging cases arise when the corrective signal is weak or inconsistent. Both **Blank** and **Vague** feedback substantially reduce performance, showing that correction requires an explicit and retrievable answer signal rather than additional text alone. The **Conflict** setting also degrades performance, although it remains above the most severely corrupted noise condition, suggesting that retrieval-based adaptation can partially tolerate contradictory feedback.

Overall, our stress tests broaden feedback adaptation evaluation by explicitly covering practically relevant imperfect-feedback conditions encountered in real-world deployment.

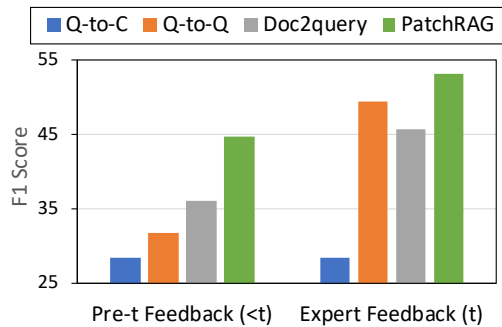


Figure 3: **Ablation of query-based retrieval strategies under feedback adaptation on HotpotQA.** We compare query-to-context (Q-to-C; standard RAG), query-to-query (Q-to-Q), and Doc2query (Nogueira et al., 2019) with an intent-context retrieval strategy before and after expert feedback injection. The results illustrate how intent-level retrieval becomes critical for propagating corrective feedback under semantic variation.

4.5 Further Analyses

We conduct further analyses to better understand the structural requirements of feedback adaptation. Rather than comparing methods in isolation, these analyses examine (i) why standard retrieval paradigms fail to propagate feedback, (ii) which components are necessary for reliable adaptation, (iii) how grounding and generalization trade off during adaptation, and (iv) whether these behaviors persist across different retrieval backbones.

Why standard retrieval fails under feedback adaptation. We first analyze why conventional retrieval strategies struggle to propagate corrective feedback. Fig. 3 compares four query-based retrieval approaches on HotpotQA: Q-to-C (standard RAG retrieval), Q-to-Q matching, Doc2query (Nogueira et al., 2019), and the proposed dual intent–context retriever. Standard Q-to-C retrieval exhibits the weakest performance and shows almost no improvement after expert feedback is introduced. This behavior highlights a fundamental limitation of content-only retrieval. Q-to-Q matching substantially improves performance after feedback injection with 21.0 points, indicating that intent-level similarity becomes critical once corrections are available. Doc2query improves pre- t performance by expanding lexical coverage, but yields limited gains (+7.6 points) after expert feedback, reflecting the constraints of operating within a single embedding space.

In contrast, combining both intent and context signals enables both grounding and semantic generalization, allowing feedback to propagate reliably

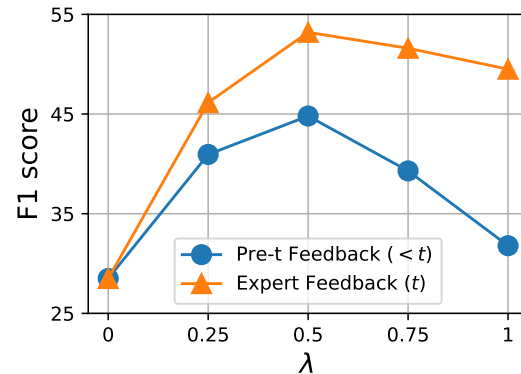


Figure 4: **Impact of the balancing parameter λ on feedback adaptation behavior on HotpotQA.** We report performance before and after expert feedback injection. Varying λ highlights the balance between content grounding and semantic generalization, which influences the reliability of post-feedback behavior.

to related queries. These results illustrate why standard retrieval paradigms are insufficient for feedback adaptation under semantic variation.

Component analysis. We examine which components are required to support reliable post-feedback behavior. Tab. 3 presents an ablation study of key system components for feedback adaptation. Replacing standard retrieval scoring with the dual-similarity formulation substantially improves post-feedback performance, demonstrating that intent-aware retrieval is necessary to surface relevant corrective feedback. In addition, explicitly incorporating retrieved feedback into the generation prompt further improves reliability after feedback injection. This indicates that feedback adaptation requires not only retrieving corrections, but also conditioning generation on explicit corrective exemplars.

Together, these results suggest that reliable feedback adaptation depends on both intent-level retrieval and feedback-conditioned generation, rather than on either component in isolation.

Intent-context trade-off in feedback adaptation. We analyze how feedback adaptation behavior varies along the intent-context axis by examining the effect of the balancing parameter λ in the intent-context retrieval mechanism. Fig. 4 reports performance on HotpotQA before and after expert feedback injection.

When only pre- t feedback is available, performance peaks at intermediate λ values, indicating that excessive reliance on either content grounding or intent generalization can degrade retrieval reliability. After expert feedback is incorporated at

\mathcal{R}	Step	NQ	TriviaQA	HotpotQA	Average
BM25	$< t$	26.0	75.6	30.1	43.9
	t	27.0 (+1.0)	77.8 (+2.2)	30.8 (+0.7)	45.2 (+1.3)
DPR	$< t$	36.9	74.7	43.8	51.8
	t	50.0 (+13.1)	81.6 (+6.9)	52.7 (+8.9)	61.4 (+9.6)
bge-m3	$< t$	36.2	76.8	44.8	52.6
	t	49.8 (+13.6)	83.9 (+7.1)	53.2 (+8.4)	62.3 (+9.7)

Table 4: **Performance across different retrieval backbones under feedback adaptation.** We report results before ($< t$) and after incorporating expert feedback at time t . **Green numbers** indicate post-feedback performance gains. Across sparse (BM25) and dense (DPR, bge-m3) retrievers, incorporating feedback consistently improves post-feedback performance, with stronger embedding spaces enabling more reliable propagation to semantically related queries.

step t , performance becomes less sensitive to larger λ values, reflecting improved alignment once explicit corrections are available.

These trends highlight a trade-off in feedback adaptation: effective correction requires balancing semantic generalization with content grounding. We adopt $\lambda = 0.5$ as a stable operating point that consistently balances these requirements across feedback settings.

Robustness across retrieval backbones. Finally, we evaluate whether feedback adaptation behavior depends on a specific retrieval backbone. Tab. 4 reports results for BM25, DPR, and bge-m3 under different feedback availability settings.

Across all retrievers, incorporating expert feedback at step t consistently improves post-feedback performance, indicating that feedback adaptation generalizes across heterogeneous retrieval settings. Sparse retrieval with BM25 exhibits smaller gains due to limited semantic matching, which constrains intent-level generalization. Dense retrievers such as DPR and bge-m3 yield larger improvements, reflecting higher-quality embedding spaces that better support feedback retrieval.

5 Related Works

Evaluation of RAG and knowledge adaptation. RAG systems are commonly evaluated using static open-domain QA benchmarks such as TriviaQA (Joshi et al., 2017), Natural Questions (Kwiatkowski et al., 2019), and HotpotQA (Yang et al., 2018). While these standard benchmarks rigorously assess retrieval and reasoning on fixed knowledge snapshots, they fail to measure how systems adapt to errors over time. To address model updates, recent research has expanded into knowledge editing (Meng et al., 2022, 2023) and memory-centric RAG (Jimenez Gutierrez et al.,

2024; Packer et al., 2023), focusing on mechanisms to construct or edit memory structures. However, a critical gap remains in evaluation: neither static QA benchmarks nor editing-specific metrics quantify the operational dynamics of feedback adaptation—the correction lag and post-feedback performance in a deployed stream. In this work, we address this limitation by formalizing *feedback adaptation* to explicitly measure these temporal trade-offs.

RAG adaptation for specific domains. Recent studies view domain-specific RAG as a training-time adaptation problem that jointly optimizes retrieval and generation components. RAG-end2end (Siriwardhana et al., 2023) demonstrates that end-to-end training improves robustness to domain shifts and noisy retrieval compared to modular pipelines. To further strengthen grounding, RAFT (Zhang et al., 2024) trains language models with both relevant and distracting documents, promoting selective evidence usage and citation-aware reasoning in open-book settings. Complementary to model-centric approaches, data-centric methods (Mao et al., 2024; Xu et al., 2025) exploit unlabeled in-domain corpora to synthesize contrastive QA pairs, enabling self-supervised RAG adaptation without manual annotations. Beyond unstructured text retrieval, recent works (Barron et al., 2024; Sharma et al., 2025) integrates structured knowledge sources to retrieve compact and verifiable contexts, effectively reducing hallucinations in technical domains. However, they are not designed for direct feedback-driven adaptation and rely on training-time updates, leading to substantial correction lag.

RAG Enhancement. Efforts to enhance RAG span both training-time and inference-time methodologies, each aiming to improve evidence utilization and robustness. Train-based approaches optimize

the retriever, generator, or both: RA-DIT (Lin et al., 2024) and InstructRetro (Wang et al., 2024) fine-tune LLMs with retrieval-augmented instructions; RankRAG (Yu et al., 2024c), Chain-of-Note (Yu et al., 2024b), ChatQA (Liu et al., 2024), and Self-RAG (Asai et al., 2024) strengthen generation robustness to noisy or incomplete retrieval; and preference-based methods such as RPO (Yan et al., 2025) promote faithful grounding. However, these techniques rely on parameter updates, making adaptation to new user feedback dependent on additional fine-tuning. In contrast, train-free strategies improve RAG at inference time by optimizing queries and retrieved contexts, via hypothetical document generation (Gao et al., 2023) and multi-query fusion (Rackauckas, 2024), or by employing LLM-based listwise reranking (Sun et al., 2023; Sachan et al., 2022). More recent directions leverage inference-time scaling and structural signals, with graph-based methods like GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2024), and agentic or reasoning-driven designs such as MemoRAG (Qian et al., 2025) and TSSS (Bang et al., 2025). Overall, while these methods substantially improve RAG effectiveness, they leave the direct incorporation of user feedback understudied.

FAQ-based retrieval. To mitigate the semantic gap between user queries and declarative documents, several approaches propose transforming the retrieval task into a query-to-query matching problem, effectively constructing a synthetic FAQ index. Doc2Query (Nogueira et al., 2019) pioneered this direction by expanding documents with predicted queries. With the advent of LLMs, methods like InPars (Bonifacio et al., 2022) have advanced this paradigm by generating high-quality synthetic QA pairs via few-shot prompting.

6 Conclusion

We introduced *feedback adaptation* as a problem setting for evaluating RAG systems under interactive feedback. We proposed two complementary evaluation axes, *correction lag* and *post-feedback performance*, to capture adaptation behavior that is overlooked by standard accuracy-based evaluation. Through empirical analysis, we showed that training-based approaches exhibit a trade-off between delayed correction and reliable adaptation. We further presented **PatchRAG** as a minimal inference-time instantiation that illustrates the practical implications of feedback adaptation.

7 Limitations

This work focuses on feedback adaptation in retrieval-augmented generation, specifically on how corrective feedback propagates through retrieval and generation once it is introduced. Accordingly, we evaluate adaptation under a controlled setting that enables precise measurement of correction lag and post-feedback behavior. While real-world feedback may be more heterogeneous, noisy, or interactive, these complexities primarily affect feedback acquisition rather than the propagation mechanisms studied in this work. Extending the framework to richer human-in-the-loop feedback settings is an important direction for future research, but does not undermine the validity of the feedback adaptation behaviors evaluated here.

Our approach performs adaptation through non-parametric memory updates and inference-time conditioning, enabling immediate behavioral adjustment without retraining. This design choice intentionally targets deployment scenarios where rapid response to feedback is critical. However, as feedback accumulates over time, unresolved or conflicting feedback may require higher-level consolidation or reconciliation mechanisms. Such long-horizon consistency management is complementary to the feedback adaptation framework introduced in this work and remains an open challenge.

8 Potential Risk

Feedback adaptation in retrieval-augmented generation relies on storing and retrieving external feedback and documents, which may include sensitive or private information. While **PatchRAG** focuses on the mechanisms of feedback propagation rather than data collection, improper handling of feedback memory could raise privacy concerns in real-world deployments. As with other RAG-based systems, practical applications should ensure that external databases and feedback stores comply with data protection and privacy regulations, and that sensitive information is handled appropriately.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *ArXiv Preprint*.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to re-

- trieve, generate, and critique through self-reflection. In *Proceedings of International Conference on Learning Representations*.
- Jihwan Bang, Juntae Lee, Seunghan Yang, and Sungha Choi. 2025. Think straight, stop smart: Structured reasoning for efficient multi-hop rag. In *Proceedings of Advances in Neural Information Processing Systems Workshop on Efficient Reasoning*.
- Ryan C Barron, Vesselin Grantcharov, Selma Wanna, Maksim E Eren, Manish Bhattarai, Nicholas Solovyev, George Tompkins, Charles Nicholas, Kim Ø Rasmussen, Cynthia Matuszek, et al. 2024. Domain-specific retrieval-augmented generation using vector stores, knowledge graphs, and tensor factorization. In *Proceedings of International Conference on Machine Learning and Applications*, pages 1669–1676. IEEE.
- Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *ArXiv Preprint*.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *ArXiv Preprint*.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitanansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *ArXiv Preprint*.
- Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.
- Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *ArXiv Preprint*.
- Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of International Conference on Learning Representations*.
- Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Proceedings of Advances in Neural Information Processing Systems*, 37:59532–59569.
- 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 Annual Meeting of the Association for Computational Linguistics*, pages 1601–1611.
- 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 Empirical Methods in Natural Language Processing*, pages 6769–6781.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2024. Ra-dit: Retrieval-augmented dual instruction tuning. In *Proceedings of International Conference on Learning Representations*.
- Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Building gpt-4 level conversational qa models. *ArXiv Preprint*.
- Kelong Mao, Zheng Liu, Hongjin Qian, Fengran Mo, Chenlong Deng, and Zhicheng Dou. 2024. Rag-studio: Towards in-domain adaptation of retrieval augmented generation through self-alignment. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 725–735.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Proceedings of Advances in Neural Information Processing Systems*, 35:17359–17372.
- Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *Proceedings of International Conference on Learning Representations*.
- Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *ArXiv Preprint*.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. MemGPT: Towards llms as operating systems. *ArXiv Preprint*.
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard,

- et al. 2021. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of North American Chapter of the Association for Computational Linguistics*, pages 2523–2544.
- Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference*, pages 2366–2377.
- Zackary Rackauckas. 2024. Rag-fusion: a new take on retrieval-augmented generation. *International Journal on Natural Language Computing*, 13.
- Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 3781–3797.
- Kartik Sharma, Peeyush Kumar, and Yunqing Li. 2025. Og-rag: Ontology-grounded retrieval-augmented generation for large language models. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 32950–32969.
- Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.
- Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 14918–14937.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv Preprint*.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of Advances in Neural Information Processing Systems*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Boxin Wang, Wei Ping, Lawrence McAfee, Peng Xu, Bo Li, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Instructretro: Instruction tuning post retrieval-augmented pretraining. In *Proceedings of International Conference on Machine Learning*.
- Ran Xu, Hui Liu, Sreyashi Nag, Zhenwei Dai, Yaochen Xie, Xianfeng Tang, Chen Luo, Yang Li, Joyce C Ho, Carl Yang, et al. 2025. Simrag: Self-improving retrieval-augmented generation for adapting large language models to specialized domains. In *Proceedings of North American Chapter of the Association for Computational Linguistics*, pages 11534–11550.
- Shi-Qi Yan, Quan Liu, and Zhen-Hua Ling. 2025. Rpo: Retrieval preference optimization for robust retrieval-augmented generation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 5228–5240.
- Seunghan Yang, Juntae Lee, Jihwan Bang, Kyuhong Shim, Minsoo Kim, and Simyung Chang. 2025. Learning contextual retrieval for robust conversational search. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 12002–12014.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 2369–2380.
- Tian Yu, Shaolei Zhang, and Yang Feng. 2024a. Auto-rag: Autonomous retrieval-augmented generation for large language models. *ArXiv Preprint*.
- Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2024b. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of Empirical Methods in Natural Language Processing*.
- Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiakuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024c. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *ArXiv Preprint*.
- Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *ArXiv Preprint*.

Feedback Adaptation for Retrieval-Augmented Generation

A Prompt Template for RAG

Prompt engineering with retrieved chunks or Q&As is another important factor to improve performance. For fair comparison between **PatchRAG** and standard RAG, we unified the prompt template.

Justification for Q&A inclusion. The inclusion of explicit Q&A pairs in the prompt is a deliberate design choice to simulate a feedback adaptation scenario where the system must prioritize and adhere to specific expert corrections. Crucially, the answer components provided in-context are synthetic responses generated by an auxiliary model (GPT-4) to mimic human feedback (See [Appendix B](#)), and are independent of the ground-truth labels used for final evaluation. Consequently, this setup does not constitute data leakage; rather, it rigorously evaluates the system’s ability to generalize correction instructions to lexically distinct queries through semantic retrieval, measuring behavioral alignment rather than trivial answer copying.

Prompt template for PatchRAG

```
Question: {QUESTION #1}
Answer: {ANSWER #1}
Question: {QUESTION #2}
Answer: {ANSWER #2}
...
Question: {QUESTION #5}
Answer: {ANSWER #5}
Context1: {CONTEXT #1}
Context2: {CONTEXT #2}
...
ContextN: {CONTEXT #N}
Please answer the below question based on given
above question and answer pairs and contexts.
Note that you should generate the response only
for answering the question within a few words.
Do not contain extra comments.
Question: {TARGET QUESTION}
```

Prompt template for standard RAG

```
Context1: {CONTEXT #1}
Context2: {CONTEXT #2}
...
Context5: {CONTEXT #5}
Please answer the below question based on given
above contexts. Note that you should generate
the response only for answering the question
within a few words. Do not contain extra comments.
Question: {TARGET QUESTION}
```

B Feedback Preparation

B.1 Pre- t Feedback Generation

As described in [Sec. 4.2](#), the feedback available before step t (*i.e.*, pre- t feedback) is synthetically

generated using Llama-3 8B. For each document chunk in the corpus, we prompt Llama-3 8B to generate corresponding question–answer pairs using the template shown below. The resulting outputs are parsed into structured Q&A pairs and stored as feedback instances in the memory.

Prompt template for Q&A Generation from LLM

```
please generate question and answer pairs to check
the understandability of following context.
Format should be Q: question A: answer.
{CONTEXT}
```

B.2 Expert Feedback at Step t

Expert feedback at step t is constructed by first paraphrasing each evaluation query using GPT-4. The paraphrased query is then provided to GPT-4 together with the corresponding evidence context, and the resulting response is used as the corrected answer in the feedback instance. Each feedback entry thus consists of a paraphrased query, a generated answer, and its supporting context, simulating an expert correction with the same underlying intent as the evaluation query.

To ensure a clean evaluation protocol, the original test queries are not directly stored in the feedback memory. The paraphrased queries differ in surface form from the evaluation queries and are used solely to provide intent-aligned but lexically distinct feedback. This design allows us to assess whether corrections generalize to semantically related inputs rather than relying on direct exposure to the test instances.

C Implementation Details

C.1 Datasets

TriviaQA ([Joshi et al., 2017](#)). It offers a complex question answering dataset featuring 950,000 pairs of questions and answers sourced from over 662,000 documents on Wikipedia and the internet. Unlike simpler QA benchmarks like SQuAD, TriviaQA presents a tougher challenge because answers often require more than just predicting a text span from lengthy contexts. The dataset includes both subsets that are verified by humans and those created by machines, and we only utilized verified datasets in this experiment.

Parameters	Value
Rank	8
Alpha	16
Train on Inputs	True
LoRA Modules	Q, K, V, O
Dropout	0.05
Batch Size	128
Loss Function	Cross Entropy Loss
Learning Rate	3e-4
Scheduler	Cosine Annealing
Optimizer	AdamW
Epochs	1

Table 5: Training hyperparameters.

Natural Questions (Kwiatkowski et al., 2019). It is designed for training question answering systems, featuring over 300,000 training, nearly 8,000 development, and approximately 8,000 test examples, each pairing a Google search query with a relevant Wikipedia article. These articles include marked sections that provide a detailed response to the query, as well as shorter snippets that directly answer the question, although some annotations may be absent, indicating no answer is available. Additionally, a small fraction of the dataset, about 1%, contains binary "yes" or "no" answers instead of detailed excerpts.

HotpotQA (Yang et al., 2018). It is a dataset featuring roughly 113,000 questions based on English Wikipedia, designed to need the lead sections from two articles for answers. Accompanying each question are the key paragraphs and a selection of sentences marked by crowdworkers as essential facts for responding. However, there are several examples that are missing some key paragraphs, and it causes the performance drop in RAG.

C.2 Evaluation Metrics

EM. Exact Match is a strict metric that measures the percentage of predictions that match the ground truth exactly. It is often used in the context of machine comprehension and question answering tasks where the goal is to produce an exact answer. For a given dataset, the Exact Match score is calculated by dividing the number of predictions that are exactly the same as the true answers by the total number of predictions made.

F1. The F1 score is a more nuanced metric that considers both precision and recall. It is the harmonic mean of precision and recall, providing a balance between the two.

D Training Configuration

To validate the superiority of our method compared to train-based RAG, RAFT and golden-only

Method	Retrieval	Generation	Total
RAFT	23.7	459.4	483.2
PatchRAG	24.6	450.5	475.2

Table 6: Inference time (milliseconds) comparison between RAFT and PatchRAG.

finetuning methods followed the parameters as described in Tab. 5. We adopted LoRA tuning (Hu et al., 2022) that is one of the parameter efficient finetuning methods. Despite of imbalance problem of three datasets, both RAFT and golden-only methods get higher EM as using more relevant dataset as a training set, but they have lower performance than PatchRAG that is only updating the dataset into the database. Lastly, all the experiments for training are conducted on two Nvidia A5000s.

E Further Analyses

E.1 Inference Computation Cost

To assess the inference computation cost, we measured the inference time of PatchRAG and train-based RAG (*i.e.*, RAFT) using dense retrieval with DPR. The inference time was averaged over 100 examples after a warm-up run with another 100 examples, conducted on a server equipped with two NVIDIA A5000 GPUs and an Intel Xeon Gold 6342 CPU @ 2.80GHz. As shown in Tab. 6, while train-based RAG achieves faster retrieval (23.7 ms vs. 24.6 ms), it incurs higher generation latency (459.4 ms vs. 450.5 ms), leading to a slightly higher total inference time (483.2 ms vs. 475.2 ms). This increased generation time stems from the integration of LoRA into the base LLM in our setup. Although train-based RAG benefits from marginally faster retrieval, the overall time difference is negligible. This suggests that inference-time feedback incorporation does not impose prohibitive overhead in practice.

E.2 Failure Scenario

We analyze a challenging failure case that is representative of realistic deployed RAG systems, where user feedback is available but retrieval remains ambiguous due to semantically adjacent yet intent-misaligned evidence. In this scenario, even with the dual intent-context retrieval strategy described in Eq. 1, some retrieved question-answer pairs are not directly aligned with the target query intent.

As illustrated in Tab. 7, this setting reflects a common practical condition in open-domain QA, where relevant information is scattered across re-

Question: What do you practice in a dojo? Answer: martial art		
Method	Type	Contents
Standard RAG	context	Sport Gichin Funakoshi said, "There are no contests in karate." In pre-World War II Okinawa, kumite was not part of karate training. Shigeru Egami relates that, in 1940, some karateka were ousted from their dojo because they adopted sparring after having learned it in Tokyo. Karate is divided into style organizations. These organizations sometimes cooperate in non-style specific sport karate organizations or federations. Examples of sport organizations include AAKF/ITKF, AOK, TKL, AKA, WKF, NWUKO, WUKF and WKC. Organizations hold competitions (tournaments) from local to international level. Tournaments are designed to match
	context	and in some rare cases even time-limited grappling on the ground are also allowed. Free sparring is performed in a marked or closed area. The bout runs for a fixed time (2 to 3 minutes.) The time can run continuously (iri kume) or be stopped for referee judgment. In light contact or semi contact kumite, points are awarded based on the criteria: good form, sporting attitude, vigorous application, awareness/zanshin, good timing and correct distance. In full contact karate kumite, points are based on the results of the impact, rather than the formal appearance of the scoring technique.
	context	Karate appeared in the Soviet Union in the mid-1960s, during Nikita Khrushchev's policy of improved international relations. The first Shotokan clubs were opened in Moscow's universities. In 1973, however, the government banned karate—together with all other foreign martial arts —endorsing only the Soviet martial art of sambo. Failing to suppress these uncontrolled groups, the USSR's Sport Committee formed the Karate Federation of USSR in December 1978. On 17 May 1984, the Soviet Karate Federation was disbanded and all karate became illegal again. In 1989, karate practice became legal again, but under strict government regulations, only after the dissolution of the Soviet Union in 1991 did independent karate schools resume functioning, and so federations were formed and national tournaments in authentic styles began.
PatchRAG	context	Shigeru Egami, Chief Instructor of Shotokan Dojo, opined that "the majority of followers of karate in overseas countries pursue karate only for its fighting techniques...Movies and television... depict karate as a mysterious way of fighting capable of causing death or injury with a single blow... the mass media present a pseudo art far from the real thing." Shoshin Nagamine said, "Karate may be considered as the conflict within oneself or as a life-long marathon which can be won only through self-discipline, hard training and one's own creative efforts.
	Q&A	Question: When did karate practice become legal again in the Soviet Union? Answer: In 1989
	Q&A	Question: What do Shigeru Egami think is the main reason why people in overseas countries practice karate? Answer: They practice karate only for its fighting techniques.
	Q&A	Question: What did the JKA do for Bell's grades? Answer: They ratified his black belt on February 5, 1964, but he had to relinquish his Yoseikan grade
	Q&A	Question: What do organizations hold? Answer: Organizations hold competitions (tournaments) from local to international level.
	Q&A	Question: What is a useful tool to understand a kata? Answer: Bunkai

Table 7: **Qualitative comparison under a realistic retrieval failure scenario.** We present a representative open-domain QA example where retrieved evidence is semantically related but weakly aligned with the true query intent, a common failure mode in deployed RAG systems. For clarity, we show the top-2 retrieved contexts for both standard RAG and **PatchRAG**, along with the feedback question-answer patches retrieved from Eq. 1. Note that **yellow highlight** indicates the answer for the given question, and it indicates that contexts or question-answer pairs are relevant to the user query q .

lated topics and retrieved contexts may partially overlap with, but not fully resolve, the user's underlying intent. Despite this ambiguity, we observe that the feedback patches retrieved by **PatchRAG** consistently contain stronger intent-aligned signals than those surfaced by standard RAG, resulting in evidence that is more actionable for correcting the model's behavior. This qualitative analysis highlights that **PatchRAG** does not rely on idealized retrieval conditions. Instead, it remains effective under realistic retrieval noise, demonstrating its robustness in scenarios where feedback is imperfect and evidence relevance is only weakly aligned with the query.

F Acknowledgements

We utilized Large Language Models (*e.g.*, ChatGPT) to assist in refining the textual content and to

generate feedback on this work. These tools were used solely for writing support, and were not employed for code generation or implementation.