# In-Situ Eval: A Modular Framework for Custom and Real-Time RAG Benchmarking

**Ritvik Garimella**[1], **Kaushik Roy**[2], **Chathurangi Shyalika**[1], **Amit Sheth**[1]
[1]Artificial Intelligence Institute, University of South Carolina, Columbia, SC, USA
[2]University of Alabama, Tuscaloosa, AL, USA
ritvikg@sc.edu, kroy2@ua.edu, jayakodc@email.sc.edu, amit@sc.edu

## Abstract

Retrieval-Augmented Generation (RAG) has become the standard approach for integrating domain knowledge into Large Language Models (LLMs). However, fair comparison of RAG pipelines remains difficult: data preparation is often ad hoc, sub-sampling methods are opaque, parameters vary across implementations, and evaluation is fragmented. We present `In-Situ Eval`, a unified and reproducible framework that operationalizes the full RAG pipeline with configurable sub-sampling strategies and both RAG-specific and generic evaluation metrics. The platform supports two execution modes: an offline Dataset mode for evaluating precomputed outputs, and a live Retrieval mode for benchmarking RAG variants with state-of-the-art LLMs. Users can flexibly select datasets, retrieval techniques, models, and metrics, enabling side-by-side comparisons, ablations, and targeted analyses. This holistic approach reduces computational costs, clarifies the impact of sub-sampling techniques, and provides actionable insights for real-world deployments. By facilitating transparent, customizable, and interative benchmarking, `In-Situ Eval` empowers both researchers and practitioners to make informed decisions in adapting RAG pipelines to domain-specific needs.

**Code** — https://github.com/Ritvik-G/insitu-eval/
**Demo Video** — https://youtu.be/DoKKwQbclIg

## Introduction

Large Language Models (LLMs) have become foundational in both academic research and industrial applications, powering tasks such as question answering, summarization, and code generation. As their capabilities are extended into specialized domains through retrieval-based techniques such as Retrieval-Augmented Generation (RAG) (Lewis et al. 2020), evaluating domain-specific performance becomes essential. Practitioners increasingly need flexible model configurations, baseline comparisons, and real-time evaluations. Existing benchmarks offer broad performance metrics (Wang, Hertzmann, and Russakovsky 2024) but rarely capture domain-specific and deployment-specific constraints. Evaluating state-of-the-art LLMs across full datasets is computationally and financially prohibitive, often requiring extensive GPU resources and costly API calls (Yu, Li, and Fan 2023; Maier, Menold, and McComb 2022). These barriers limit comprehensive benchmarking, particularly for researchers and smaller organizations. In addition, models that top public leaderboards frequently underperform in domain-specific contexts. Even holistic frameworks like HELM (Liang et al. 2023) often fail to align with the practical constraints and goals of applied use cases (Alzahrani et al. 2024), resulting in benchmarking incongruence.

To address these limitations, we present an interactive, modular, and extensible framework for real-time evaluation of LLMs and retrieval pipelines. The system enables users to benchmark domain-specific datasets with state-of-the-art LLMs and retrieval techniques under configurable parameters, while also supporting comparisons against baselines. By incorporating user-defined sub-sampling strategies, evaluations can be run in near real time, significantly reducing computational and financial costs. The framework integrates both generic and retrieval-oriented evaluation metrics, providing flexibility for diverse experimental needs. Overall, it lowers barriers to practical LLM adaptation and offers a unified platform for reproducible, domain-specific benchmarking.

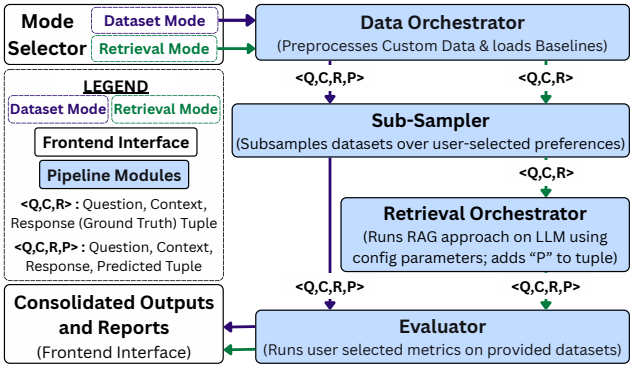


Figure 1: System Architecture

## System Overview

`In-Situ Eval` employs multiple internal modules (Figure 1) designed to provide near real-time scores and feedback. Each module exposes user-configurable options and

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

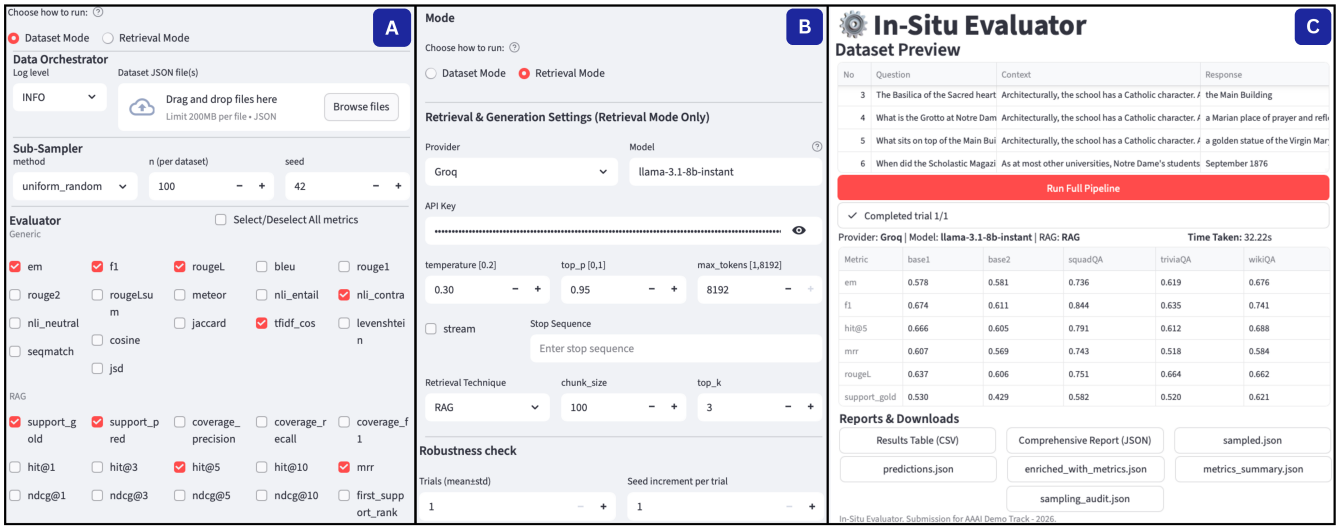


Figure 2: Interface design and customization components for In-Situ Eval. (a) Top portion of side panel interface when in Dataset Mode, (b) Side panel interface when in Retrieval Mode, (c) Outputs obtained from the input(s) provided through side panel (results are for retrieval mode). We show an example of input and output (b & c) where evaluations are done over Squad, WikiQA, HotpotQA and two custom datasets.

supports multi-trial execution to ensure robustness. The key components are:

**Data Orchestrator:** This module lets the user upload multiple datasets or prepare three selected baselines of varying difficulty: SQuAD (Rajpurkar et al. 2016) [Easy], with primarily single-span questions; TriviaQA (Joshi et al. 2017) [Medium], with some requiring multi-hop reasoning; and WikiQA (Yang, Yih, and Meek 2015) [Difficult], where questions demand multi-hop reasoning. The module accepts `<Question, Context, Response>` tuples in Retrieval Mode, and `<Question, Context, Response, Predicted>` tuples in Dataset Mode.

**Sub-Sampler:** This module applies user-specified subsampling strategies to reduce computational cost and support faster experimentation. It currently supports eight techniques, including random, stratified, and active learning.

**Retrieval Orchestrator:** This module, available only in Retrieval Mode, executes user-selected retrieval techniques (e.g., Vanilla RAG, RAPTOR (Sarthi et al. 2024), GraphRAG (Edge et al. 2024)) on sub-sampled datasets. It integrates with user-provided LLMs via Groq (Groq 2025) and OpenAI (OpenAI 2025), with configurable hyperparameters, and generates predicted answers for the input questions as output.

**Evaluator:** This module supports both generic (17 methods) and RAG-specific (15 methods) quantitative evaluations, allowing users to select task-appropriate metrics for their experiments.

## Online Demonstration Platform

**User Interface and Interactivity** The pipeline is accessible through a browser-based interface implemented in Python. Upon accessing the platform, users are presented with a side panel that offers two execution modes: **Dataset Mode**, designed for evaluating existing predicted outputs (Figure 2:a), and **Retrieval Mode**, designed for applying RAG techniques over API-accessible LLMs (Figure 2:b). Each module of the pipeline exposes configurable options, allowing users to select models, adjust parameters, and tailor the workflow to their specific needs. The modular backend supports seamless integration of additional datasets, models, and evaluation methods with minimal overhead.

**Modular Workflow:** At the core of the platform is a multi-stage modular setup that structures the process of RAG-based Q&A evaluation from data loading to final assessments. The modules support custom dataset uploads with logging, sub-sampling methods with configurable parameters, LLM provider and model selections, RAG techniques, evaluation suites for both generic Q&A and RAG-specific tasks, and robustness checks through multi-trial execution with incremental seeds.

**Scores Generation and Execution:** Users can download results at every stage of the pipeline (Figure 2:c). They can examine intermediate scores and outputs, and export comprehensive reports, results, and audits for further analysis and integration into local pipelines.

## Discussion and Conclusion

Our unified framework streamlines RAG benchmarking by integrating retrieval methods, sampling strategies, modes, and evaluation suites into one modular pipeline that lets users test workflows, compare baselines, export configurations, and visualize results in real-time. Together, these features make `In-Situ Eval` a practical, flexible tool for reproducible, customizable real-time RAG benchmarking.

## References


Alzahrani, N.; Alyahya, H.; Alnumay, Y.; AlRashed, S.; Alsubaie, S.; Almushayqih, Y.; Mirza, F.; Alotaibi, N.; Al-Twairesh, N.; Alowisheq, A.; Bari, M. S.; and Khan, H. 2024. When Benchmarks are Targets: Revealing the Sensitivity of Large Language Model Leaderboards. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 13787–13805. Bangkok, Thailand: Association for Computational Linguistics.

Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R. O.; and Larson, J. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Groq. 2025. Groq Platform. Last Accessed: 2025-09-15.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1601–1611. Vancouver, Canada: Association for Computational Linguistics.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474.

Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; Newman, B.; Yuan, B.; Yan, B.; Zhang, C.; Cosgrove, C.; Manning, C. D.; Ré, C.; Acosta-Navas, D.; Hudson, D. A.; Zelikman, E.; Durmus, E.; Ladhak, F.; Rong, F.; Ren, H.; Yao, H.; Wang, J.; Santhanam, K.; Orr, L.; Zheng, L.; Yuksekgonul, M.; Suzgun, M.; Kim, N.; Guha, N.; Chatterji, N.; Khattab, O.; Henderson, P.; Huang, Q.; Chi, R.; Xie, S. M.; Santurkar, S.; Ganguli, S.; Hashimoto, T.; Icard, T.; Zhang, T.; Chaudhary, V.; Wang, W.; Li, X.; Mai, Y.; Zhang, Y.; and Koreeda, Y. 2023. Holistic Evaluation of Language Models. *arXiv preprint arXiv:2211.09110*.

Maier, T.; Menold, J.; and McComb, C. 2022. The relationship between performance and trust in AI in E-Finance. *Frontiers in Artificial Intelligence*, 5: 891529.

OpenAI. 2025. OpenAI API. Last Accessed: 2025-09-15.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Su, J.; Duh, K.; and Carreras, X., eds., *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392. Austin, Texas: Association for Computational Linguistics.

Sarthi, P.; Abdullah, S.; Tuli, A.; Khanna, S.; Goldie, A.; and Manning, C. D. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Wang, A.; Hertzmann, A.; and Russakovsky, O. 2024. Benchmark suites instead of leaderboards for evaluating AI fairness. *Patterns*, 5(11).

Yang, Y.; Yih, W.-t.; and Meek, C. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In Màrquez, L.; Callison-Burch, C.; and Su, J., eds., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2013–2018. Lisbon, Portugal: Association for Computational Linguistics.

Yu, L.; Li, Y.; and Fan, F. 2023. Employees' appraisals and trust of artificial intelligences' transparency and opacity. *Behavioral Sciences*, 13(4): 344.