Generate but Verify: Answering with Faithfulness in RAG-based Question Answering

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) enhances LLMs by grounding answers in retrieved passages, which is key in factual Question Answering. However, generated answers may still be unfaithful, either due to retrieval or generation errors. We introduce the problem of Answering with Faithfulness (AwF), which brings faithfulness prediction to the forefront, explicitly coupling it with answer generation. We define precision-recall metrics tailored to this problem and present a unified framework allowing for (1) tunable control over faithfulness precision and (2) direct evaluation and comparison of different AwF methods. We conduct a comprehensive empirical study across multiple models and benchmarks, evaluating diverse AwF methods, and identifying consistent performance trends. Additionally, we demonstrate the usage of AwF methods in applications that incorporate different strategies for handling unfaithful answers. Our findings establish AwF as a robust framework, providing a principled approach to balance between providing answers and applying corrective actions in RAG-based Question Answering.

1 Introduction

Retrieval Augmented Generation (RAG) enhances Large Language Models (LLMs) by grounding their responses in an external corpus, ensuring that answers are based on retrieved evidence rather than only the LLM parametric memory. An answer is said to be faithful when it is indeed grounded by the retrieved content. This property is key in factual questions where the user is asking for a well-defined piece of information (as opposed to opinion-based or creativity-seeking queries).

A key challenge in RAG-based factual Question Answering, is thus to ensure that generated answers are supported by retrieved passages. Errors in this process can stem from two sources: retrieval failures, where the retrieved passages are misleading or irrelevant; and generation failures, where the LLM produces an incorrect or unsupported answer, due to hallucinations, confusion, or misinterpretation of the passages. Approaches to mitigate such issues include adapting retrieval to generation (Zhang et al., 2023; Shi et al., 2024), chain-of-thought (CoT) reasoning to ignore irrelevant passages and extract useful information from relevant ones (Wei et al., 2024), and parallel generation, i.e., generating an answer for each retrieved passage and choosing the best one (Lewis et al., 2020). In these examples, the objective is answer quality, and the intermediate task of faithfulness prediction typically is solved only implicitly. In this paper, we bring this problem to the spotlight and evaluate this task explicitly. Beyond its clear contribution to the answering quality, we note that this problem is also fundamental for transparency: without a supporting passage, the system cannot provide users with a verifiable basis for the generated answer.

In our setting, given a user question and a set of retrieved passages, the goal is to generate both an answer and a faithfulness prediction, indicating whether the answer is supported by the passages. We refer to this problem as “Answering with Faithfulness" (AwF). This formulation couples answer generation and faithfulness assessment into one component that produces both an answer and the assessment, allowing for explicit control over when to trust a generated response.

While related problems have been studied, most approaches address faithfulness implicitly or indirectly. For example, Query Performance Prediction (QPP)(Asai et al., 2024) assess whether retrieval is likely to be useful before generation, while CoT reasoning promotes faithfulness by first reasoning about which passages contain relevant data. However, these methods lack explicit faithfulness prediction, offering no direct evaluation, control, or transparency over the response’s trustworthiness.

Our AwF framework generalizes these approaches by making faithfulness prediction an explicit output, along with the generated answer. Respectively, we define metrics of *AwF precision* and *AwF recall*, tailored to our setting. Using these definitions, our AwF framework provides three key advantages: (1) It allows for tuning the balance between providing answers and applying corrective actions (e.g., abstaining, invoking a stronger LLM, generating a different answer), as different applications have different tolerances for uncertainty. (2) It enables a systematic comparison between different methods within a unified evaluation framework. (3) It supports method composition, where different techniques can be combined to improve answering faithfulness.

We conduct a line of experiments on top of a diverse collection of benchmarks, to evaluate the performance of different AwF methods for answering with faithfulness. We consider (1) unified approaches that simultaneously provide an answer along with its faithfulness prediction, (2) compositions of answer generation methods with faithfulness prediction methods.

Beyond evaluating AwF methods, we explore their use in applications that incorporate different strategies for handling unfaithful answers, such as reverting to non-RAG answering or switching to a larger, more expensive LLM. Our analysis reveals consistent trends, showing that AwF methods can be chosen based on their performance within our framework, enabling informed selection and fine-tuning of the application's operating point.

Summarizing, our contributions are as follows: (1) We define the AwF problem, allowing for explicit tuning of faithfulness prediction. (2) We introduce tailored precision-recall metrics and propose a unified framework enabling comparison across AwF methods. (3) We conduct a comprehensive study across models and benchmarks, revealing consistent performance trends of AwF methods. (4) We exemplify the use of AwF methods and our framework in applications with different strategies for handling unfaithful answers.

2 Related Work

We divide the existing works related to our task by the input they use: just the query, the query and retrieved data, and the query, retrieved data, and generated response. This division is inspired by the pre-retrieval and post-retrieval categorization used in Query Performance Prediction (QPP) literature (Arabzadeh et al., 2024). This IR task of predicting the retrieval performance is highly related to ours: rather than predicting IR metrics, in the RAG setting, we would like to predict whether retrieved content will actually improve the quality of the generated response.

Decision based on the query. A few publications tackled this RAG-QPP problem, though it was part of a wider effort: (Asai et al., 2024) have a complete RAG system that among other things, before generating predicts whether retrieval would be helpful. Wang et al. (2024a) propose an adaptive RAG system for conversations that decides whether retrieval should be invoked via prompting the LLM or an external model. In our experiments, we did not explore these strategies since they are intuitively but also empirically (Wang et al., 2024a) less effective than post-retrieval QPP.

Decision based on the query and retrieved content. A direct approach towards solving this problem is given by Thakur et al. (2024). They provide a dataset (NoMiracl) of queries and retrieved passages along with labels for answerability of queries. The passages are related to the query but in the unanswerable case, do not contain the information needed to answer the query. Using this dataset, they show how LLMs perform poorly in identifying these unanswerable cases. Wang et al. (2024a) propose an adaptive RAG system that decides whether retrieved content should be used in the generation phase and show that fine-tuned LLMs perform better than a BERT-based model. Ye et al. (2024) and Wei et al. (2024) fine-tune an LLM to generate a response using CoT, where it first decides which passages are useful, then generates a response. Meng et al. (2024) propose using LLM-generated binary relevance labels that are subsequently used to compute continuous QPP scores tailored towards a desired retrieval metric, such as the precision-oriented reciprocal rank, or the recall-oriented NDCG. Finally, some papers (Yoran et al., 2024; Jin et al., 2024) have an implicit approach to the problem, where rather than letting the LLM or another model decide whether retrieved content is useful, they fine-tune the LLM to be robust to irrelevant data.

Decision based on the query, retrieved content, and response. Here, the challenge is to decide whether a given response has sufficiently high quality given the retrieved content. A natural way of do-

ing so is determining whether the answer is implied by the retrieved content, otherwise, retrieval was in retrospect unnecessary. This challenge is closely related to fact-checking (Wang et al., 2024b), where NLI is a popular approach for verifying a statement given evidence (see (Honovich et al., 2022) and references within).

A computationally expensive alternative to standard NLI models is represented by RAGAS faithfulness (Es et al., 2024), a metric evaluating whether the generated answer is faithful to the retrieved context via several invocations to a powerful LLM. We consider this technique as well as other NLI models in our paper.

Wu et al. (2024) studied the inclination of RAG models to prefer their parametric memory over the provided context, and vice versa. They provide a test for faithfulness in which they compare the perplexity of an answer generated by an LLM with and without retrieved content. We make use of this technique in our paper.

Uncertainty estimation. Outside the RAG scenario, a related line of work concerns uncertainty estimation in LLMs. Estimating uncertainty/confidence is crucial for assessing the reliability of LLMs (Geng et al., 2024). Earlier studies (Murray and Chiang, 2018; Malinin and Gales, 2020; Jiang et al., 2021) estimated model confidence by computing the marginal probability of the generated sequences based on the language model's token probabilities. Other works directly prompted LLMs to generate their confidence (Mielke et al., 2022; Lin et al., 2022; Tian et al., 2023; Zhou et al., 2023). Another line of works (Si et al., 2023; Lin et al., 2023; Nikitin et al., 2024) used sampling decoding to generate multiple answers to the same question and considered semantically different answers as a proxy for uncertainty. All these works are general-purpose and do not specifically address our scenario: LLMs can generate responses with high confidence even when the retrieved context doesn't actually support their claims.

3 Problem Definition & Metrics

We provide a formal definition of the AwF problem and then show how a line of methods fits into this framework. The input to the AwF problem consists of a question q, and a collection of passages P, typically obtained via retrieval. Our goal is building an *AwF method* M that computes

$$M(q, P) = (a, v),$$

where a is the generated answer, and $v \in \{0, 1\}$ is its predicted faithfulness indicator. The faithfulness indicator aims to predict the *true* faithfulness of an answer given the passages:

$$V_{q,P}(a) = \begin{cases} 1 & P \text{ supports the statement:} \\ & \text{"the answer to } q \text{ is } a\text{"}, \\ 0 & \text{otherwise.} \end{cases}$$

$V_{q,P}(a)$ can be estimated by human annotators, a judge LLM (Chiang and Lee, 2023; Zheng et al., 2023; Es et al., 2024), or comparison with a given ground truth answer known to be faithful to P.

The predictions a and v are highly related, and their quality should be evaluated as a whole. In particular, the metrics measuring the performance of M should capture the fact that when $v = 0$, the quality of a is not important. Indeed, one can think of making use of v as a gating mechanism to invoke a different generation process when $v = 0$ thereby ignoring a in this case. Similarly, when M fails to produce a faithful answer, v should be 0 even if a supported answer can be generated from the passages. Moreover, note that the cost of providing a wrong answer vs. the cost of not providing an answer when a proper answer can be inferred from the passages, depends on the specific use case. Thus, we want to maximize two competing objectives that capture this tradeoff. To that end, we introduce a tailored notion of precision and recall, defined below.

Assume we are given a set of question and passage pairs $\{(q_i, P_i)\}_{i=1}^N$, and M is used to append to each such pair its predictions a_i, v_i. We define our metrics w.r.t. to the set of tuples $\{(q_i, P_i, a_i, v_i)\}_{i=1}^N$. Throughout, all sums are over these N tuples, and we denote their corresponding ground truth labels as $V_i = V_{q_i,P_i}(a_i)$.

AwF Precision is similar to the standard classification precision - the fraction of answers the generator correctly deemed faithful, out of the total number of faithful answers. The number of correctly classified faithful answers (True Positives) is $\text{True-Pos} = \sum_i v_i \cdot V_i$, and the total number of answers that were classified as faithful (Predicted Positive) is $\text{Pred-Pos} = \sum_i v_i$. The *answering faithfulness precision* is therefore

$$\text{AwF-Precision} = \frac{\text{True-Pos}}{\text{Pred-Pos}}$$

We note that even though the precision appears identical to the standard classifier precision at first glance, it also depends on the generated answers as well, since the ground truth label V_i depends on the answer a_i.

AwF Recall is the fraction of answers correctly deemed faithful, out of the total number of *faithfully answerable* questions, meaning questions that have a faithful answer w.r.t. the passages. Formally, the number of faithfully answerable questions is $\text{F-Answerable} = |\{(q_i, P_i) : \exists a^* \text{ such that } V_{q_i,P_i}(a^*) = 1\}|$, and the answering faithfulness recall is

$$\text{AwF-Recall} = \frac{\text{True-Pos}}{\text{F-Answerable}}$$

A connection to the classical notion of classifier recall can be obtained from a simple reformulation. Denoting by Faithful the number of faithful generated answers, $\text{Faithful} = \sum_i V_i$, the recall can be reformulated as

$$\text{AwF-Recall} = \underbrace{\frac{\text{True-Pos}}{\text{Faithful}}}_{\text{classifier recall}} \cdot \underbrace{\frac{\text{Faithful}}{\text{F-Answerable}}}_{\text{answering recall}}$$

Thus, our notion of recall is the classifier recall given the answers, multiplied by the ability of the generator to produce faithful answers whenever a faithful answer exists.

Connection to Post-Retrieval QPP We note that AwF is similar to QPP, with the distinction that the predicted faithfulness indicator V evaluates whether P supports the correct answer a_q^*, rather than the generated answer a. Due to their similarity, techniques originally designed for QPP are evaluated as AwF and vice versa. In what follows we consider QPP-Precision and QPP-Recall, defined analogously to AwF-Precision and AwF-Recall, but w.r.t. the QPP variant of V.

4 Methods

We consider various methods that fit within the AwF framework, demonstrating how our formulation unifies approaches originally designed for different problems, such as answer generation. In some cases, we make slight adaptations to align these approaches with AwF (e.g., pairing answer generation with a simple faithfulness prediction that always sets $v = 1$). Some of the methods we consider provide a hard classification result, i.e., $v \in \{0, 1\}$, whereas others provide a continuous decision function that can be thresholded to obtain $v \in \{0, 1\}$. We first present *unified* methods that simultaneously output both an answer and its faithfulness indicator. Then, we provide *composed* methods, that combine answering modules with faithfulness prediction ones. The exact LLM prompts we used in the following methods are available in Appendix A.4.

4.1 Unified Methods

Intrinsic Abstention. A straightforward technique where we prompt an LLM to answer only if the answer appears in the context and reply with "DONT KNOW" when it does not. We set $v = 1$ if and only if the answer is not "DONT KNOW".

CoT few-shot Hybrid. A variant of the Intrinsic Abstention method using both chain-of-thought and few-shot examples. It is inspired by the method described in (Wei et al., 2024), where the LLM is instructed to reason about the relevance of the passages before answering and is given two examples comprising a question, passages, and the reasoning. We adapt the original method by prompting the LLM to answer "DONT KNOW" if an answer cannot be deduced from the passages ($v = 0$).

Dual Generation. A method proposed by Wu et al. (2024). The idea is to generate an answer both with and without P, then compare the (normalized) perplexity percentiles of both answers in order to choose one. We define a continuous decision function for v as the difference between the perplexities.

4.2 Composed Methods

We consider methods that compose two components for producing the AwF output (a, v): an answer generation method to generate a, and a faithfulness prediction method to produce v. Below we describe concrete answer generation and faithfulness prediction methods we consider in this paper.

4.2.1 Answer Generation

Vanilla. The straightforward approach for answering questions. Here, we instruct the LLM to answer the question given the passages.

InstructRAG. This is a variant of the Vanilla method using both chain-of-thought and few-shot examples proposed by Wei et al. (2024). We

slightly modified the in-context examples and instructions to enable a structured response, from which we can extract only the final answer.

4.2.2 Faithfulness Prediction

Trivial. A simple baseline that always predicts $v = 1$, meaning that it believes the answer from the generation method is always faithful.

Pre-Answering Prediction. A method originally designed for Post-Retrieval QPP. Given q, P we ask the LLM to evaluate whether P contains an answer to q. We ask for a single yes/no answer given all the passages and obtain a continuous decision function for v by inspecting the logits of the generated response. We use the prompt given in (Thakur et al., 2024).

Post-Answering NLI. A faithfulness prediction method mimicking $V_{q,P}(a)$. Here, we first invoke one of the answering methods described above to generate the response a, then use the question, passages, and the generated answer to decide whether the question-answer pair is faithful to the passages. We use a DeBERTa-based NLI model[1] (Laurer et al., 2024) by feeding it the hypothesis and premise as described in the definition of V. We chose a DeBERTa-based model due to it being lightweight ($< 1B$ parameters), and having adequate quality. Further details about considered alternatives such as TRUE and RAGAs and the implementation can be found in Appendix A.1.

5 Empirical Investigation

We conduct a series of experiments to evaluate the performance of different AwF methods in terms of precision and recall.

5.1 Experimental Setup

For our experiments, we use question-answering benchmarks where each entry consists of a question, one or more retrieved passages, a reference answer, and a binary relevance label indicating whether the answer can be inferred from the passages. We focus on single-hop questions, where the answer is fully contained within a single passage. To compute precision and recall, as defined in Section 3, we estimate $V_{q,P}(a)$ as follows. We consider $V_{q,P}(a)$ to be 1 if: (1) a is equivalent to the reference answer, as judged by a strong language model (Claude 3.5 Sonnet), and (2) the reference answer is supported by at least one passage.

We evaluate our methods on three public benchmarks: NQ (Kwiatkowski et al., 2019), NoMIRACL (Thakur et al., 2024), and BioASQ (Krithara et al., 2023). NQ consists of real-user queries with answers retrieved from Wikipedia. NoMIRACL is a benchmark based on real-user queries, used to assess whether LLMs have the ability to abstain when retrieval fails. BioASQ focuses on biomedical questions from PubMed abstracts. Further details on the benchmarks collection and pre-processing are provided in Appendix A.2.

Table 1 provides benchmark statistics: the sizes of our datasets (number of entries), the average number of passages per question, and the percentage of questions that are answerable by their associated passages.

Benchmark	size	% of answerable questions	passages per question
NQ	5K	82%	5
NoMIRACL	3.2K	81%	10.1
BioASQ	2.9K	50%	6.5

Table 1: Benchmarks Statistics.

For each benchmark, we test the unified and composed methods for the AwF task, as presented in Section 4. For the composed methods, we test all combinations of answer generation and faithfulness prediction methods. Since AwF methods rely on instruction-tuned generative models, we conduct experiments using Llama 3 Instruct (3B, 8B, 70B), Falcon 3 Instruct (3B, 10B), and Qwen 2.5 Instruct (72B). Models are referred to by their first letter and size, e.g., F10B.

5.2 Results

For each AwF method, LLM, and dataset, we compute the AwF precision, AwF recall, and their F1 score. For the methods outputting a continuous score (e.g., Post-Answering NLI), we evaluate their F1 across all thresholds and report the max value. We used the Bootstrap method to compute 95% confidence intervals. Table 2 presents the average F1 score obtained by each of the methods over our three benchmarks. When using names of answer generation methods, we implicitly refer to those methods composed with the Trivial faithfulness prediction method. Elaborated tables including all benchmarks of both F1 scores and area under the

[1] https://huggingface.co/MoritzLaurer/deberta-v3-large-zeroshot-v2.0

Method	F3B	F10B	L3B	L8B	L70B	Q72B
Intrinsic	0.53	0.61	0.29	0.55	0.70	0.72
Trivial Vanilla	0.56	0.64	0.37	0.58	0.66	0.68
CoT	0.61	0.67	0.55	0.64	0.69	0.72
Trivial InstRAG	0.62	0.66	0.57	0.62	0.66	0.68
Pre-Ans Vanilla	0.56	0.65	0.37	0.59	0.69	0.68
Pre-Ans InstRAG	0.62	0.67	0.57	0.63	0.70	0.68
NLI Vanilla	0.60	0.66	0.42	0.61	0.67	0.69
NLI InstRAG	0.64	0.68	0.59	0.64	0.68	0.70
Dual Gen	0.56	0.64	0.37	0.58	0.66	0.68

Table 2: Average F1, defined by the harmonic mean of the average precision and recall over the datasets of every method and model. The results of each dataset appear in Appendix A.5.

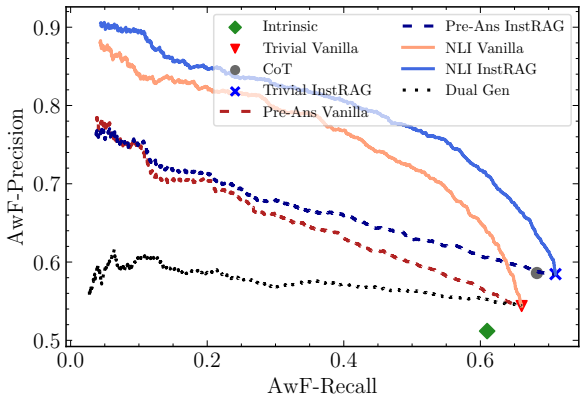


Figure 1: AwF-Precision and AwF-Recall of AwF methods using F3B on NQ benchmark.

curve (AUC), together with a 95% confidence interval appear in Appendix A.5 (Tables 6 and 7).

Using large-scale LLMs. We observe an inherently different behavior between AwF methods using medium-scale and large-scale LLMs. In particular, for large models, a simple method such as Intrinsic Abstention performs very well, achieving an F1 score that is either higher or on par compared to other methods. Notably, Intrinsic Abstention with large LLMs outperforms all methods with medium-scale models, highlighting the advantage of model size, where no sophisticated AwF method is necessarily required. However, since Intrinsic Abstention lacks a decision function, it produces fixed precision-recall values. Thus, in scenarios requiring higher precision or recall, alternative methods may be preferable.

Using medium-scale LLMs. We turn to examine the behavior of AwF methods for the case of medium-scale LLMs, and present a representative example in Figure 1. Other LLMs and benchmarks exhibit similar trends; full results are available in Appendix A.6 (Figure 5). Notably, chain-of-thought improves performance: InstructRAG consistently outperforms Vanilla (as it is a variant of Vanilla with CoT), and CoT few-shot Hybrid outperforms Intrinsic Abstention (as it is a variant of Intrinsic Abstention using CoT).

Moreover, looking at Table 2, as well as at Figure 1, which is representative of the overall trends observed across all configurations for the case of medium LLMs, we see a clear hierarchy between faithfulness prediction methods. Across all medium scale models, benchmarks, and answering generation methods, the curve resulting from the composition of Post-Answering NLI fully dominates the curve resulting from the composition of Pre-Answering Prediction on the same answering method. This reinforces the intuition that considering the generated answer improves faithfulness prediction. Although Dual Generation is not a composed method, as it is tailored to predict the faithfulness of Vanilla, we see that it behaves similarly to Pre-Answering Prediction composed over Vanilla, and consistently underperforms compared to Post-Answering NLI.

Consistently, we observe that composition preserves the ranking of answering methods. That is, across all tested medium scale models and benchmarks, when one answering method outperforms another (InstructRAG consistently outperforms Vanilla), this ordering remains unchanged after their composition with any faithfulness prediction method, resulting in a fully dominant curve[2].

Finally, composing a faithfulness prediction method with an answer generation method yields a balanced tradeoff between recall and precision. This allows for significant precision gains, often by dozens of percentage points, by adjusting recall. This flexibility makes composition crucial for applications requiring higher precision, such as medical queries, where fixed-precision methods (e.g., answer generation methods) may fall short.

Relation to QPP. Consider QPP-Precision and QPP-Recall as defined with respect to post-retrieval QPP in Section 3. We present the evaluation of those metrics using a representative ex-

[2]The only exception is the case of Post-Answering NLI with Falcon10B on BioASQ, in which the domination is not complete. Moreover, for L8B on BioASQ, InstructRAG is on par with Vanilla, thus it is not expected that their composition with faithfulness prediction methods will yield curves with a clear hierarchy.

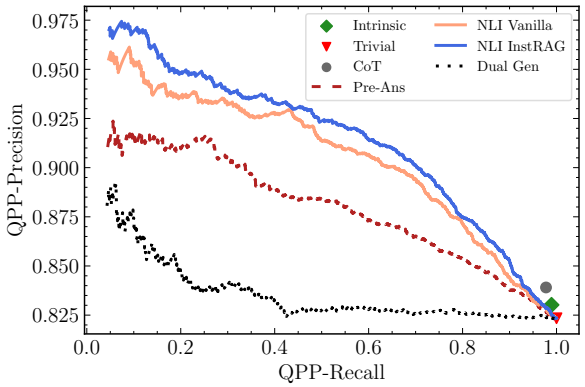


Figure 2: QPP-Precision and QPP-Recall of AwF methods using F3B on NQ benchmark.

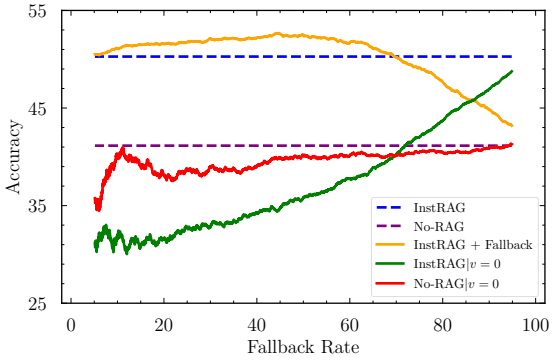


Figure 3: No-RAG fallback. Accuracies of different types of answers as a function of the fallback rate. Orange: InstructRAG accuracy with No-RAG fallback when NLI predicts $v = 0$. Green: Avg. accuracy of InstructRAG answers predicted as unfaithful. Red: Avg. accuracy of No-RAG answers for unfaithful InstructRAG cases.

ample of a medium-scale LLM (F3B) in Figure 2[3] (a full visual description appears in Figure 6 of Appendix A.6). Recall that Pre-Answering Prediction is designed to predict the QPP objective whereas Post-Answering NLI is designed for the AwF objective. Nevertheless, the same trends as before remain, in particular the superior performance of Post-Answering NLI. This is somewhat surprising and could bring insights into future solutions for the QPP problem.

6 Applications

We present two applications of AwF, each employing a distinct strategy for handling instances where the generated answer is predicted unfaithful ($v = 0$). We demonstrate how utilizing AwF methods with better AwF-Precision/AwF-Recall curves in these applications improves system performance.

6.1 No-RAG Fallback

This strategy falls back to generating a response without RAG whenever $v = 0$, using the same LLM but relying only on its parametric memory. Indeed when $v = 0$, the retrieved content is likely to be irrelevant and consequently it might only distract the LLM, hurting its answer quality. Therefore, it could be beneficial to try to generate the answer without using the retrieved content. Figure 3 illustrates the No-RAG strategy for Llama3B on BioASQ questions, when using the composition of InstructRAG and Post-Answering NLI. The figure presents overall answer accuracy (i.e., the percentage of generated answers that match the reference) as a function of the fallback rate which can be controlled via different thresholding of the soft score Post-Answering NLI generates for v. Incorporating fallback improves accuracy over InstructRAG for fallback rates up to 70%, peaking around 50% rate before declining. These results are not surprising, since 50% of BioASQ questions are not answerable from the passages; this is a demonstration of AwF ability to detect those cases. This can be further explained by comparing InstructRAG and No-RAG answers when $v = 0$ (w.r.t. InstructRAG answer): in low fallback rates No-RAG outperforms InstructRAG, so replacing the answers enhances overall accuracy. However, as fallback increases, the accuracy gap between the two narrows, and beyond 70%, InstructRAG surpasses No-RAG, making further fallback detrimental.

In Table 3 we compare Pre-Answering Prediction and Post-Answering NLI (both composed with InstructRAG) for this application[4]. We present here results only for BioASQ, since for NQ and NoMIRACL we observe little to no improvement in overall system accuracy for most LLMs, likely due to them having mostly (∼82%) questions with relevant passages. In BioASQ however, only 50% of the questions contain relevant context and the overall improvement is significant for most LLMs. The results for all benchmarks can be found in Ap-

[3]Since QPP-Precision and QPP-Recall are independent of the generated answer, all methods that estimate faithfulness without considering the answer produce identical results. In particular, this applies to all methods based on Trivial (which always predicts $v = 1$) and on Pre-Answering Prediction (where faithfulness prediction is performed before answer generation). The respective composed methods are referred shortly as Trivial and Pre-Answering Prediction in Figure 2.

[4]We use 5-fold cross-validation, optimizing the threshold on four folds and evaluating performance on the fifth.

LLM	Pre-Ans	NLI
Q72B	4.26%	6.51%
L70B	7.39%	8.72%
L8B	1.33%	3.13%
L3B	-0.10%	2.32%
F10B	0.51%	0.20%
F3B	0.03%	0.07%

Table 3: Accuracy improvement with No-RAG fallback over InstructRAG answers, using Pre-Answering Prediction or Post-Answering NLI for faithfulness prediction on BioASQ.

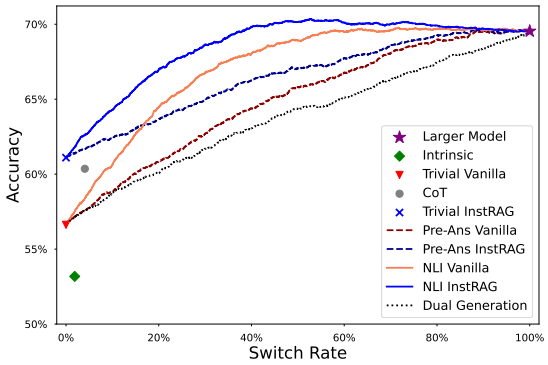


Figure 4: Switching to a larger model. Accuracy vs Switch Rate, when using InstructRAG and replacing F3B answers with L70B for cases where $v = 0$ on NQ.

pendix A.3.1, along with an analysis showing that improvements occur mainly for questions without relevant context. In most cases, Post-Answering NLI outperforms Pre-Answering Prediction in accuracy improvement. This is consistent with Section 5.2, where Post-Answering NLI compositions achieve better AwF-Precision/AwF-Recall curves. These findings reinforce the value of selecting the best AwF method and being able to tune its faithfulness threshold (and resulting fallback rate) for achieving a maximal accuracy for the No-RAG fallback application.

6.2 Switching to a Larger Model

This strategy matches a scenario where the RAG system primarily uses a small and cheap LLM, but when $v = 0$, switches to a larger, more expensive model. The system balances two competing objectives: (i) quality, measured by accuracy, and (ii) cost, measured by switch rate, i.e., the proportion of answers replaced by the larger model. Figure 4 illustrates the trade-off between accuracy and switch rate for Falcon3B and Llama70B on the NQ benchmark. The ranking of the faithfulness methods from Section 5.2 remains consistent, showing that better AwF-Precision/AwF-Recall curves lead to a more favorable trade-off. Note that a baseline that switches the answer randomly would have a linear trade-off curve, similar to the Dual Generation one. This same trend persists across the other benchmarks and LLM choices (full results can be found in Appendix A.3.2).

7 Discussion

Our work introduces the Answering with Faithfulness problem along with tailored precision and recall metrics, providing a unified framework for its evaluation. By making faithfulness prediction an explicit output, we generalize prior approaches that implicitly address answer faithfulness, enabling direct comparisons across methods. Evaluating all methods on a common scale facilitates informed trade-offs, allowing applications to balance them based on their specific requirements.

The trends observed across the diverse AwF methods we consider remain consistent across configurations, problems (AwF and QPP), and applications explored in this work, demonstrating the robustness of our framework. Beyond their performance for solving the AwF problem, we find that AwF methods that rely on a generated answer are also highly effective for solving the QPP problem. In particular, we see that those methods achieve superior results to QPP solutions, despite their inherent bias of solving a slightly different problem. The same performance trends also persist in the applications we consider, where AwF methods are used together with different fallback strategies for handling unfaithful answers. This reinforces the practical utility of AwF methods, allowing for informed selection and tuning based on specific application needs.

Our findings also show that applications can select AwF methods solely based on their performance, without needing to assess the quality of specific fallback or gating strategies when handling unfaithful answers. In some cases, this distinction is less critical, such as trivial fallbacks like abstaining or high-cost alternatives like switching to a larger model. However, a promising direction for future work is to extend the AwF framework to incorporate fallback performance, enabling a more comprehensive evaluation of downstream corrective actions.

8 Limitations

The AwF problem applies to any benchmark where RAG provides a suitable solution. In this study, we focused on question-answering benchmarks, specifically those with factoid questions. We focused our attention on these benchmarks since other types would admit additional technical challenges that are outside the scope of our study, making it difficult to understand the core problem and the analysis of our results. For example, with long-form answers, faithfulness ceases to become a binary score since an answer can be partially supported by the documents. An additional limitation to our study is the language: We restricted our focus to English benchmarks and corpora and left the analysis over additional languages to future work.

Finally, our focus was on methods that do not require fine-tuning an LLM. This choice is due to two reasons: (1) The popularity of such choices in real settings, indeed it is much more convenient to use an off-the-shelf LLM as opposed to fine-tuning one. (2) The added technical challenges related to such methods, such as searching for the right hyper-parameters for training, the cost of training, and the complexity related to in-distribution vs out-of-distribution performance.

References

Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi, and Ebrahim Bagheri. 2024. Query performance prediction: From fundamentals to advanced techniques. In *European Conference on Information Retrieval*, pages 381–388. Springer.

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

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. True: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. 2024. Long-context llms meet rag: Overcoming challenges for long inputs in rag. *arXiv preprint arXiv:2410.05983*.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170.

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.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2024. Less annotating, more classifying: Addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli. *Political Analysis*, 32(1):84–100.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

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

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

Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi, and Maarten de Rijke. 2024. Query performance prediction using relevance judgments generated by large language models. *arXiv preprint arXiv:2404.01012*.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223, Brussels, Belgium. Association for Computational Linguistics.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *arXiv preprint arXiv:2405.20003*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2023. Prompting gpt-3 to be reliable. In *International Conference on Learning Representations (ICLR)*.

Nandan Thakur, Luiz Bonifacio, Crystina Zhang, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, et al. 2024. "knowing when you don't know": A multilingual relevance assessment dataset for robust retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12508–12526.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Xi Wang, Procheta Sen, Ruizhe Li, and Emine Yilmaz. 2024a. Adaptive retrieval-augmented generation for conversational systems. *arXiv preprint arXiv:2407.21712*.

Yuxia Wang, Revanth Gangi Reddy, Zain Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, et al. 2024b. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14199–14230.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. Instructrag: Instructing retrieval augmented generation via self-synthesized rationales. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*.

Kevin Wu, Eric Wu, and James Zou. 2024. Clasheval: Quantifying the tug-of-war between an LLM's internal prior and external evidence. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. *arXiv preprint arXiv:2402.13178*.

Xi Ye, Ruoxi Sun, Sercan Arik, and Tomas Pfister. 2024. Effective large language model adaptation for improved grounding and citation generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6237–6251.

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

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: Expressions of overconfidence and uncertainty in language models. *arXiv preprint arXiv:2302.13439*.

A Appendix

A.1 NLI Model

A.1.1 Implementation details

For using the NLI model to predict whether the answer question-answer pair is faithful to the pas-

sages, we created the hypothesis using this template: `The answer to the question "{q}" is: {a}`, while each passage serves as an independent premise (in preliminary experiments, we explored rephrasing the question-answer pair into its declarative form using an LLM, but it did not yield an additional advantage). In case the passage and the hypothesis together exceed the context window of the NLI model, we split the passage into chunks with an overlap of 20 words. We then use the maximum score of the NLI model over all premises as the decision function for v.

A.1.2 Model selection

To select the NLI model, we conducted a preliminary experiment evaluating the performance of different models on our task using 700 questions from NQ. We considered four bert-based models, each with fewer than 1 billion parameters:

- MoritzLaurer/DeBERTa-v3-large-mnli-fever-anli-ling-wanli
- MoritzLaurer/deberta-v3-large-zeroshot-v2.0
- MoritzLaurer/ModernBERT-large-zeroshot-v2.0
- MoritzLaurer/bge-m3-zeroshot-v2.0

Additionally, we tested TRUE (Honovich et al., 2022), a T5-XXL-based model with 7 billion parameters. Among the BERT-based models, *MoritzLaurer/deberta-v3-large-zeroshot-v2.0* performed best, achieving results comparable to TRUE. Given its significantly smaller size, we selected it as our NLI model.

We also conducted preliminary experiments with RAGAS faithfulness (Es et al., 2024), using Claude 3.5 Sonnet. However, the observed improvements over the DeBERTa-based model were negligible, and we determined that the additional computational cost of a larger model was not justified.

A.2 Benchmarks

- **NQ** (Kwiatkowski et al., 2019) is a general knowledge question answering benchmark based on queries of real users. The dataset consists of questions and ground truth answers. Specifically, we sampled, uniformly at random, 5K question-answer pairs. For each question, we retrieved 5 passages from Wikipedia, using E5-base-v2 (Wang et al., 2022) dense retrieval. Each passage was then labeled as relevant if it contains the answer as a (normalized) substring, or according to the TRUE NLI(Honovich et al., 2022) [5].

- **NoMIRACL** (Thakur et al., 2024) is a public benchmark testing whether LLMs have the ability to abstain. Each entry contains a question, passages, and relevance labels for the passages. The original dataset does not have a ground truth answer. To obtain one, we prompted Claude 3.5 Sonnet based only on the passages that were annotated as containing the answers. In addition, in the original dataset, the relevant passages are separated from the non-relevant ones. We shuffle relevant and non-relevant passages together in a random order. We consider only the English part of this dataset, as all language and NLI models we employed, support this language.

- **BioASQ** (Krithara et al., 2023) is a manually generated question-answer dataset based on abstracts of biological academic papers available in the Pubmed corpus (we used the snapshot published by (Xiong et al., 2024)). We used the BioASQ12 training set, out of which we collected the questions labeled as factoid questions, resulting in a collection of 1.48K entries. Each entry contains a question, a ground truth answer, and a list of relevant passages. To obtain irrelevant passages we used BM-25 to extract the top-10 related passages from PubMed and discard those containing the ground truth answer. Finally, we considered each question twice, using two different passage lists: once with only irrelevant passages and once with the same set, but with one randomly selected irrelevant passage replaced by a randomly chosen relevant one.

A.3 Applications supplementary material

A.3.1 No-RAG fallback

Table 4 includes the comparison between Pre-Answering Prediction and Post-Answering NLI for No-RAG fallback across all benchmarks and LLMs.

[5] A manual inspection showed this strategy to be near perfect in the setting of NQ where the answers are very short and contain only a single fact.

Benchmark	LLM	Pre-Ans (all)	Pre-Ans (relevant)	Pre-Ans (irrelevant)	NLI (all)	NLI (relevant)	NLI (irrelevant)
NQ	F3B	-0.06%	-0.07%	0.00%	0.00%	0.00%	0.00%
	F10B	-0.02%	-0.02%	0.00%	-0.04%	-0.02%	-0.23%
	L3B	-0.02%	-0.02%	0.00%	-0.04%	0.02%	-0.36%
	L8B	-0.08%	-0.07%	-0.13%	0.50%	0.39%	1.00%
	L70B	0.28%	-0.54%	4.15%	1.02%	0.18%	5.02%
	Q72B	-0.04%	-0.05%	0.00%	-0.32%	-0.19%	-0.89%
NoMIRACL	F3B	-0.03%	-0.04%	0.00%	-0.06%	-0.04%	-0.37%
	F10B	0.00%	0.00%	0.05%	-0.09%	-0.15%	0.14%
	L3B	-0.13%	-0.08%	-0.16%	-0.16%	-0.42%	0.74%
	L8B	0.00%	0.00%	0.00%	0.25%	-0.15%	2.06%
	L70B	-0.03%	0.00%	-0.20%	0.06%	-0.00%	0.55%
	Q72B	-0.13%	-0.12%	-0.13%	0.34%	-1.41%	8.22%
BioASQ	F3B	0.03%	0.00%	0.07%	0.07%	-0.26%	0.36%
	F10B	0.51%	-0.48%	1.52%	0.20%	-0.80%	1.27%
	L3B	-0.10%	0.00%	-0.20%	2.32%	0.68%	3.97%
	L8B	1.33%	-3.88%	6.52%	3.13%	-1.48%	7.75%
	L70B	7.39%	-2.75%	17.63%	8.72%	0.49%	16.99%
	Q72B	4.26%	-5.25%	13.76%	6.51%	1.60%	11.27%

Table 4: Application #1 - No-RAG fallback. The improvement in Accuracy when using No-RAG fallback over the original answers generated with InstructRAG prompt, and using Pre-Answering Prediction or Post-Answering NLI to predict faithfulness. For each method, results are shown for (all): all questions, (relevant): only questions with relevant retrieved passages, and (irrelevant): only questions with irrelevant retrieved passages.

A.3.2 Switch to a larger model

Table 5 extends the analysis of Section 6.2 across all medium-sized LLMs and datasets. We evaluated all methods with continuous decision functions, which allow control over the switch rate. Accuracy is reported at a fixed 20% switch rate, simulating a scenario with a constrained budget for expensive LLM calls. As shown, accuracy rankings at a 20% switch rate align with F1 rankings from Section 5.2, reinforcing trend consistency.

A.4 Method prompts

Below are the prompts to the Vanilla, Intrinsic Abstention, and No Context methods.

Vanilla

```
system: You are a helpful assistant that answers a question based on the context provided. Please be as concise as possible, do not add any additional information, and do not refer to the context in anyway.
---
user: Read the following context carefully and answer the question below.
Question:
<Question>
Context:
<Passage 1>

<Passage 2>

⋮

<Passage n>
```

Intrinsic Abstention

```
system: You are a helpful assistant that answers a question based on the context provided. Please be as concise as possible, do not add any additional information, and do not refer to the context in anyway. If the answer does not exist in the context, you should output the special string __DONT_KNOW__ .
---
user: Read the following context carefully and answer the question below only if the answer is supported by the context.
Question:
<Question>
Context:
<Passage 1>

<Passage 2>

⋮

<Passage n>
```

Benchmark	LLM	Dual Gen	Random Vanilla	Pre-Ans Vanilla	NLI Vanilla	Pre-Ans InstRAG	NLI InstRAG	L70B
NQ	F3B	60.04%	59.20%	60.84%	64.46%	63.66%	66.98%	69.54%
	F10B	65.90%	65.16%	67.18%	68.24%	68.02%	69.02%	69.54%
	L3B	57.74%	56.26%	57.36%	60.88%	62.78%	65.04%	69.54%
	L8B	63.28%	61.81%	62.86%	65.30%	66.58%	67.92%	69.54%
NoMIRACL	F3B	63.96%	63.74%	64.62%	68.44%	70.41%	72.98%	76.14%
	F10B	71.16%	70.43%	71.79%	73.67%	75.92%	77.77%	76.14%
	L3B	52.85%	49.24%	49.41%	53.10%	69.32%	71.35%	76.14%
	L8B	64.37%	64.55%	64.65%	67.56%	72.29%	74.48%	76.14%
BioASQ	F3B	56.68%	55.80%	56.54%	58.28%	61.00%	62.47%	67.17%
	F10B	64.07%	63.16%	64.37%	64.20%	66.72%	66.49%	67.17%
	L3B	30.31%	27.44%	27.55%	29.02%	54.56%	55.69%	67.17%
	L8B	58.38%	57.17%	58.17%	58.92%	59.98%	60.83%	67.17%

Table 5: Application #2 - switch to a larger model. Accuracy of different methods where the switch rate is fixed at 20%. The Random Vanilla method switches to a bigger LLM uniformly at random, and serves as a baseline.

No context

system: You are a helpful assistant that answers a question based on your knowledge. Please be concise as possible.

user: <Question>

Below are the prompts of the InstructRAG and CoT few-shot Hybrid methods. We note that each dataset has its own set of example questions and "rationales" for analyzing them. Below is the structure of the prompts.

InstructRAG

user: Your task is to analyze the provided documents and answer the given question. Please generate a brief explanation of how the contents of these documents lead to your answer. If the provided information is not helpful in answering the question, you only need to respond based on your own knowledge, without referring to the documents. After your analysis, give the final answer in a self-contained manner after a "Response: " prefix.

Below are some examples of how to answer the question:

###

Example 1

Question: <Example question 1>?

Answer: <Rationale 1>

###

Example 2

Question: <Example question 2>?

Answer: <Rationale 2>

###

Now it is your turn to analyze the following documents and answer the given question.

Document 1: <Passage 1>

Document 2: <Passage 2>

⋮

Document 4: <Passage n>

Based on your knowledge and the provided information, answer the question: <Question>?

CoT few-shot Hybrid

user: Your task is to analyze the provided documents and answer the given question. Please generate a brief explanation of how the contents of these documents lead to your answer. If the provided information is not helpful in answering the question, you need to respond __DONT_KNOW__. After your analysis, give the final answer in a self-contained manner after a "Response: " prefix.

Below are some examples of how to answer the question:

###

Example 1

Question: <Example question 1>?

Answer: <Rationale 1 with instruction to abstain>

```
###

Example 2

Question: <Example question 2>?

Answer: <Rationale 1 with instruction to abstain>

###

Now it is your turn to analyze the following documents and answer the given question.

Document 1: <Passage 1>

Document 2: <Passage 2>

⋮

Document 4: <Passage n>

Either answer the following question based on the provided information, or reply __DONT_KNOW__:
<Question>?
```

Here is an example of a question and the corresponding rationale for NQ:

Question: who won season 13 so you think you can dance?

Rationale: After analyzing the provided documents, I found that none of them directly mention the winner of Season 13 of "So You Think You Can Dance". However, I can use my own knowledge to answer the question.

According to various online sources, including Wikipedia and other reputable dance websites, the winner of Season 13 of "So You Think You Can Dance" is indeed Leon "Kida" Burns.

To deduce this answer, I used my knowledge of the show's history and its format. The show typically features a new season every year, and each season has a different winner. By analyzing the provided documents, I noticed that they only mention winners from previous seasons (Seasons 8, 1, and no mention of Season 13). This led me to conclude that the documents are not relevant to the question.

Therefore, I relied on my own knowledge to answer the question, which is that Leon "Kida" Burns won Season 13 of "So You Think You Can Dance".
Response: Leon "Kida" Burns won Season 13 of "So You Think You Can Dance".

Rationale with instruction to abstain: After analyzing the provided documents, I found that none of them directly mention the winner of Season 13 of "So You Think You Can Dance". However, I can use my own knowledge to answer the question.

According to various online sources, including Wikipedia and other reputable dance websites, the winner of Season 13 of "So You Think You Can Dance" is indeed Leon "Kida" Burns.

To deduce this answer, I used my knowledge of the show's history and its format. The show typically features a new season every year, and each season has a different winner. By analyzing the provided documents, I noticed that they only mention winners from previous seasons (Seasons 8, 1, and no mention of Season 13). This led me to conclude that the documents are not relevant to the question.
Response: __DONT_KNOW__

A.5 Full F1 and PR-AUC tables

Table 6 shows the best achievable F1 score, whereas Table 7 shows the precision-recall AUC, for every AwF method, benchmark, and LLM.

A.6 Graphic description of AwF methods

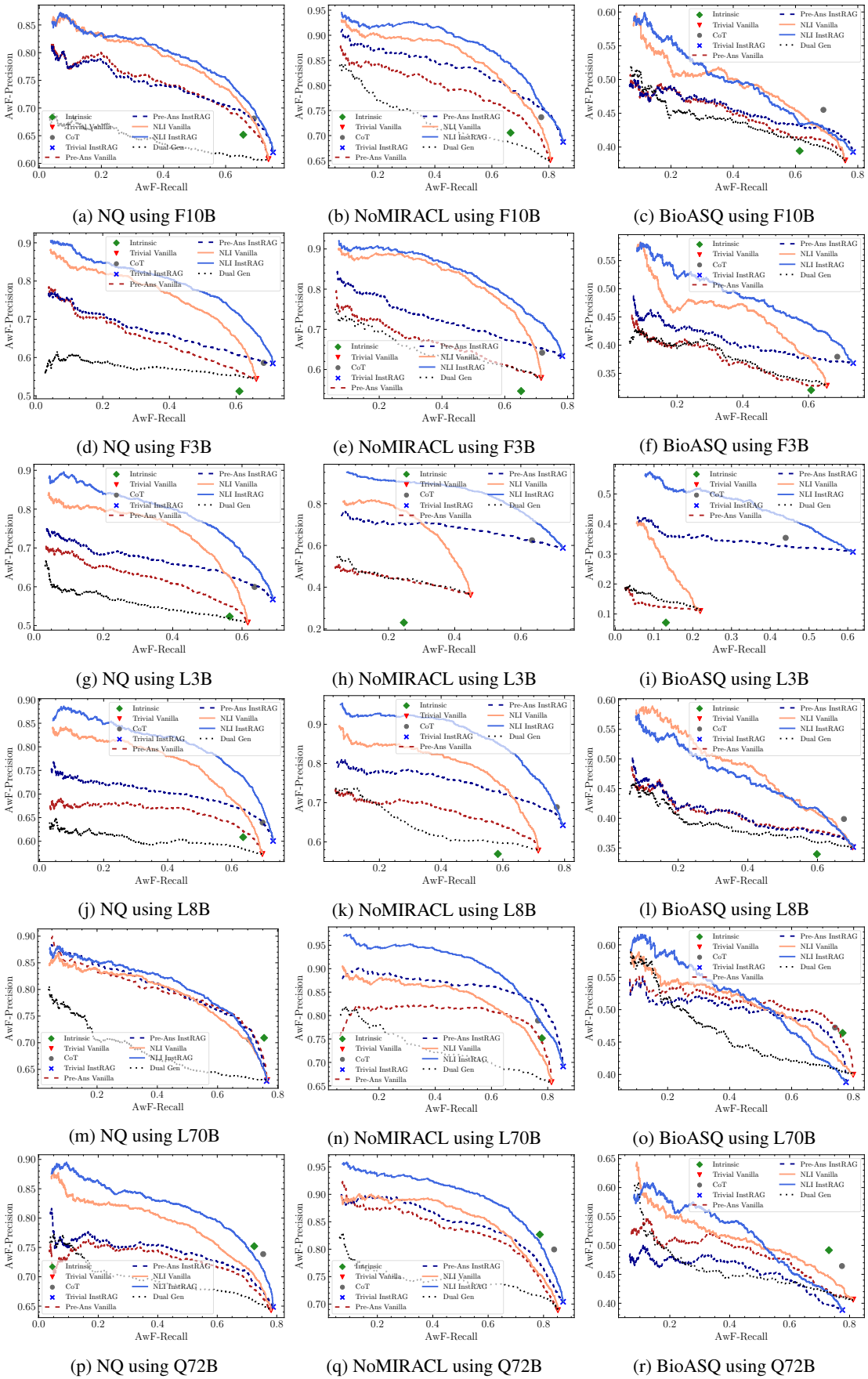
Figures 5 and 6 present the AwF precision-recall curves and QPP precision-recall curves of all AwF methods, on all LLMs and benchmarks.

Model	Benchmark	Intrinsic	Trivial Vanilla	CoT	Trivial InstRAG	Pre-Ans Vanilla	Pre-Ans InstRAG	NLI Vanilla	NLI InstRAG	Dual Gen
F3B	NQ	$55_{\pm 1.4}$	$59_{\pm 1.6}$	$63_{\pm 1.4}$	$64_{\pm 1.4}$	$59_{\pm 1.5}$	$64_{\pm 1.4}$	$62_{\pm 1.3}$	$66_{\pm 1.4}$	$59_{\pm 1.6}$
	NoMIRACL	$59_{\pm 2.2}$	$64_{\pm 1.8}$	$67_{\pm 1.7}$	$70_{\pm 1.7}$	$64_{\pm 1.8}$	$70_{\pm 1.7}$	$68_{\pm 1.9}$	$71_{\pm 1.6}$	$64_{\pm 1.8}$
	BioASQ	$41_{\pm 1.8}$	$43_{\pm 1.9}$	$48_{\pm 2.0}$	$49_{\pm 2.2}$	$43_{\pm 2.1}$	$49_{\pm 2.2}$	$46_{\pm 2.5}$	$51_{\pm 2.0}$	$43_{\pm 2.1}$
F10B	NQ	$65_{\pm 1.4}$	$66_{\pm 1.3}$	$68_{\pm 1.4}$	$67_{\pm 1.4}$	$67_{\pm 1.5}$	$69_{\pm 1.4}$	$68_{\pm 1.4}$	$69_{\pm 1.4}$	$66_{\pm 1.3}$
	NoMIRACL	$68_{\pm 1.8}$	$71_{\pm 1.6}$	$75_{\pm 1.7}$	$75_{\pm 1.5}$	$73_{\pm 1.6}$	$77_{\pm 1.5}$	$74_{\pm 1.8}$	$77_{\pm 1.7}$	$71_{\pm 1.6}$
	BioASQ	$48_{\pm 2.3}$	$50_{\pm 2.0}$	$54_{\pm 2.0}$	$52_{\pm 1.8}$	$51_{\pm 2.3}$	$53_{\pm 1.9}$	$52_{\pm 2.3}$	$53_{\pm 2.4}$	$51_{\pm 2.2}$
L3B	NQ	$54_{\pm 1.6}$	$55_{\pm 1.5}$	$61_{\pm 1.5}$	$62_{\pm 1.3}$	$56_{\pm 1.5}$	$62_{\pm 1.3}$	$59_{\pm 1.6}$	$64_{\pm 1.5}$	$55_{\pm 1.5}$
	NoMIRACL	$23_{\pm 1.5}$	$40_{\pm 1.8}$	$62_{\pm 1.9}$	$65_{\pm 1.9}$	$40_{\pm 1.8}$	$65_{\pm 1.9}$	$46_{\pm 2.2}$	$67_{\pm 2.0}$	$40_{\pm 1.7}$
	BioASQ	$09_{\pm 1.2}$	$14_{\pm 1.7}$	$39_{\pm 2.6}$	$40_{\pm 1.9}$	$14_{\pm 1.8}$	$41_{\pm 1.8}$	$18_{\pm 2.4}$	$43_{\pm 2.4}$	$15_{\pm 1.6}$
L8B	NQ	$62_{\pm 1.6}$	$62_{\pm 1.3}$	$66_{\pm 1.6}$	$65_{\pm 1.5}$	$63_{\pm 1.4}$	$66_{\pm 1.5}$	$65_{\pm 1.4}$	$68_{\pm 1.3}$	$62_{\pm 1.3}$
	NoMIRACL	$57_{\pm 1.9}$	$63_{\pm 1.9}$	$72_{\pm 1.6}$	$71_{\pm 1.9}$	$64_{\pm 2.0}$	$71_{\pm 1.8}$	$67_{\pm 2.0}$	$73_{\pm 1.8}$	$63_{\pm 1.9}$
	BioASQ	$43_{\pm 2.1}$	$46_{\pm 2.0}$	$50_{\pm 2.4}$	$46_{\pm 1.7}$	$47_{\pm 2.1}$	$47_{\pm 1.9}$	$49_{\pm 2.0}$	$49_{\pm 2.1}$	$46_{\pm 2.0}$
L70B	NQ	$73_{\pm 1.3}$	$69_{\pm 1.2}$	$70_{\pm 1.3}$	$68_{\pm 1.5}$	$71_{\pm 1.3}$	$71_{\pm 1.3}$	$70_{\pm 1.3}$	$70_{\pm 1.2}$	$69_{\pm 1.2}$
	NoMIRACL	$76_{\pm 1.9}$	$72_{\pm 1.6}$	$77_{\pm 1.6}$	$76_{\pm 1.5}$	$76_{\pm 1.7}$	$80_{\pm 1.5}$	$74_{\pm 1.6}$	$78_{\pm 1.7}$	$72_{\pm 1.6}$
	BioASQ	$57_{\pm 2.0}$	$53_{\pm 2.1}$	$57_{\pm 1.9}$	$51_{\pm 2.1}$	$58_{\pm 2.2}$	$56_{\pm 2.1}$	$54_{\pm 2.3}$	$53_{\pm 2.6}$	$53_{\pm 2.1}$
Q72B	NQ	$73_{\pm 1.4}$	$70_{\pm 1.2}$	$74_{\pm 1.3}$	$71_{\pm 1.4}$	$70_{\pm 1.4}$	$71_{\pm 1.5}$	$71_{\pm 1.4}$	$73_{\pm 1.5}$	$70_{\pm 1.3}$
	NoMIRACL	$80_{\pm 1.6}$	$76_{\pm 1.6}$	$81_{\pm 1.4}$	$77_{\pm 1.5}$	$76_{\pm 1.8}$	$78_{\pm 1.7}$	$77_{\pm 1.5}$	$80_{\pm 1.6}$	$76_{\pm 1.6}$
	BioASQ	$58_{\pm 2.2}$	$54_{\pm 1.9}$	$58_{\pm 2.0}$	$51_{\pm 2.0}$	$54_{\pm 2.1}$	$51_{\pm 2.0}$	$55_{\pm 2.3}$	$53_{\pm 2.2}$	$54_{\pm 2.0}$

Table 6: Maximum achievable AwF-F1 score, normalized to $[0, 100]$, of each method, benchmark, and LLM, with 95% bootstrap confidence intervals in subscripts.

Model	Benchmark	Intrinsic	Trivial Vanilla	CoT	Trivial InstRAG	Pre-Ans Vanilla	Pre-Ans InstRAG	NLI Vanilla	NLI InstRAG	Dual Gen
F3B	NQ	$31_{\pm 1.6}$	$35_{\pm 1.8}$	$40_{\pm 1.8}$	$41_{\pm 1.8}$	$43_{\pm 2.4}$	$47_{\pm 2.1}$	$50_{\pm 1.9}$	$56_{\pm 2.2}$	$37_{\pm 2.4}$
	NoMIRACL	$35_{\pm 2.6}$	$41_{\pm 2.3}$	$46_{\pm 2.3}$	$49_{\pm 2.3}$	$48_{\pm 2.8}$	$57_{\pm 2.7}$	$59_{\pm 2.6}$	$65_{\pm 2.4}$	$47_{\pm 2.8}$
	BioASQ	$19_{\pm 1.7}$	$21_{\pm 1.7}$	$26_{\pm 2.1}$	$27_{\pm 2.1}$	$25_{\pm 2.3}$	$30_{\pm 2.7}$	$30_{\pm 3.0}$	$36_{\pm 3.3}$	$24_{\pm 2.6}$
F10B	NQ	$42_{\pm 1.8}$	$44_{\pm 1.7}$	$47_{\pm 2.0}$	$46_{\pm 1.9}$	$54_{\pm 2.4}$	$55_{\pm 2.2}$	$58_{\pm 1.9}$	$59_{\pm 2.3}$	$47_{\pm 2.0}$
	NoMIRACL	$46_{\pm 2.4}$	$52_{\pm 2.3}$	$56_{\pm 2.6}$	$58_{\pm 2.2}$	$64_{\pm 2.7}$	$70_{\pm 2.6}$	$69_{\pm 2.4}$	$74_{\pm 2.2}$	$59_{\pm 2.6}$
	BioASQ	$24_{\pm 2.4}$	$28_{\pm 2.1}$	$31_{\pm 2.3}$	$30_{\pm 2.0}$	$34_{\pm 3.0}$	$35_{\pm 2.8}$	$37_{\pm 3.5}$	$39_{\pm 3.5}$	$33_{\pm 3.5}$
L3B	NQ	$29_{\pm 1.7}$	$31_{\pm 1.7}$	$38_{\pm 1.8}$	$39_{\pm 1.6}$	$38_{\pm 2.1}$	$46_{\pm 1.9}$	$46_{\pm 2.0}$	$54_{\pm 1.8}$	$34_{\pm 2.0}$
	NoMIRACL	$05_{\pm 0.7}$	$16_{\pm 1.5}$	$39_{\pm 2.4}$	$42_{\pm 2.5}$	$19_{\pm 2.3}$	$49_{\pm 2.9}$	$32_{\pm 2.6}$	$62_{\pm 2.4}$	$20_{\pm 2.2}$
	BioASQ	$00_{\pm 0.3}$	$02_{\pm 0.6}$	$15_{\pm 2.2}$	$18_{\pm 1.6}$	$02_{\pm 0.8}$	$21_{\pm 2.4}$	$06_{\pm 1.5}$	$28_{\pm 2.8}$	$03_{\pm 0.8}$
L8B	NQ	$38_{\pm 2.0}$	$39_{\pm 1.7}$	$44_{\pm 2.1}$	$43_{\pm 2.0}$	$46_{\pm 2.2}$	$51_{\pm 2.0}$	$53_{\pm 1.9}$	$58_{\pm 1.8}$	$42_{\pm 2.2}$
	NoMIRACL	$33_{\pm 2.2}$	$41_{\pm 2.4}$	$53_{\pm 2.4}$	$51_{\pm 2.7}$	$48_{\pm 2.8}$	$59_{\pm 2.7}$	$57_{\pm 2.6}$	$69_{\pm 2.3}$	$46_{\pm 2.6}$
	BioASQ	$20_{\pm 1.9}$	$24_{\pm 1.9}$	$26_{\pm 2.4}$	$24_{\pm 1.7}$	$28_{\pm 2.8}$	$29_{\pm 2.4}$	$34_{\pm 3.4}$	$33_{\pm 2.9}$	$27_{\pm 2.8}$
L70B	NQ	$53_{\pm 2.0}$	$48_{\pm 1.7}$	$49_{\pm 1.8}$	$47_{\pm 2.0}$	$61_{\pm 2.0}$	$61_{\pm 2.1}$	$60_{\pm 2.1}$	$61_{\pm 1.9}$	$52_{\pm 2.4}$
	NoMIRACL	$58_{\pm 2.9}$	$53_{\pm 2.3}$	$60_{\pm 2.6}$	$59_{\pm 2.4}$	$65_{\pm 2.8}$	$73_{\pm 2.5}$	$68_{\pm 2.6}$	$77_{\pm 2.0}$	$59_{\pm 2.6}$
	BioASQ	$35_{\pm 2.3}$	$31_{\pm 2.2}$	$34_{\pm 2.2}$	$30_{\pm 2.2}$	$41_{\pm 2.9}$	$38_{\pm 2.9}$	$41_{\pm 3.0}$	$40_{\pm 3.6}$	$37_{\pm 3.4}$
Q72B	NQ	$54_{\pm 2.0}$	$50_{\pm 1.8}$	$55_{\pm 1.9}$	$51_{\pm 2.0}$	$56_{\pm 2.2}$	$58_{\pm 2.1}$	$61_{\pm 2.0}$	$64_{\pm 2.3}$	$54_{\pm 2.1}$
	NoMIRACL	$65_{\pm 2.5}$	$58_{\pm 2.4}$	$67_{\pm 2.3}$	$61_{\pm 2.3}$	$71_{\pm 2.6}$	$73_{\pm 2.6}$	$72_{\pm 2.3}$	$78_{\pm 2.4}$	$63_{\pm 2.7}$
	BioASQ	$35_{\pm 2.8}$	$32_{\pm 2.2}$	$35_{\pm 2.5}$	$30_{\pm 2.1}$	$39_{\pm 2.9}$	$35_{\pm 3.0}$	$42_{\pm 3.6}$	$40_{\pm 3.3}$	$38_{\pm 3.1}$

Table 7: The AwF-Precision-AwF-Recall AUC, normalized to $[0, 100]$, of each method, benchmark, and LLM, with 95% bootstrap confidence intervals in subscripts. The AUC of methods producing a hard label is defined as the product of the precision and the recall.



(a) NQ using F10B (b) NoMIRACL using F10B (c) BioASQ using F10B

(d) NQ using F3B (e) NoMIRACL using F3B (f) BioASQ using F3B

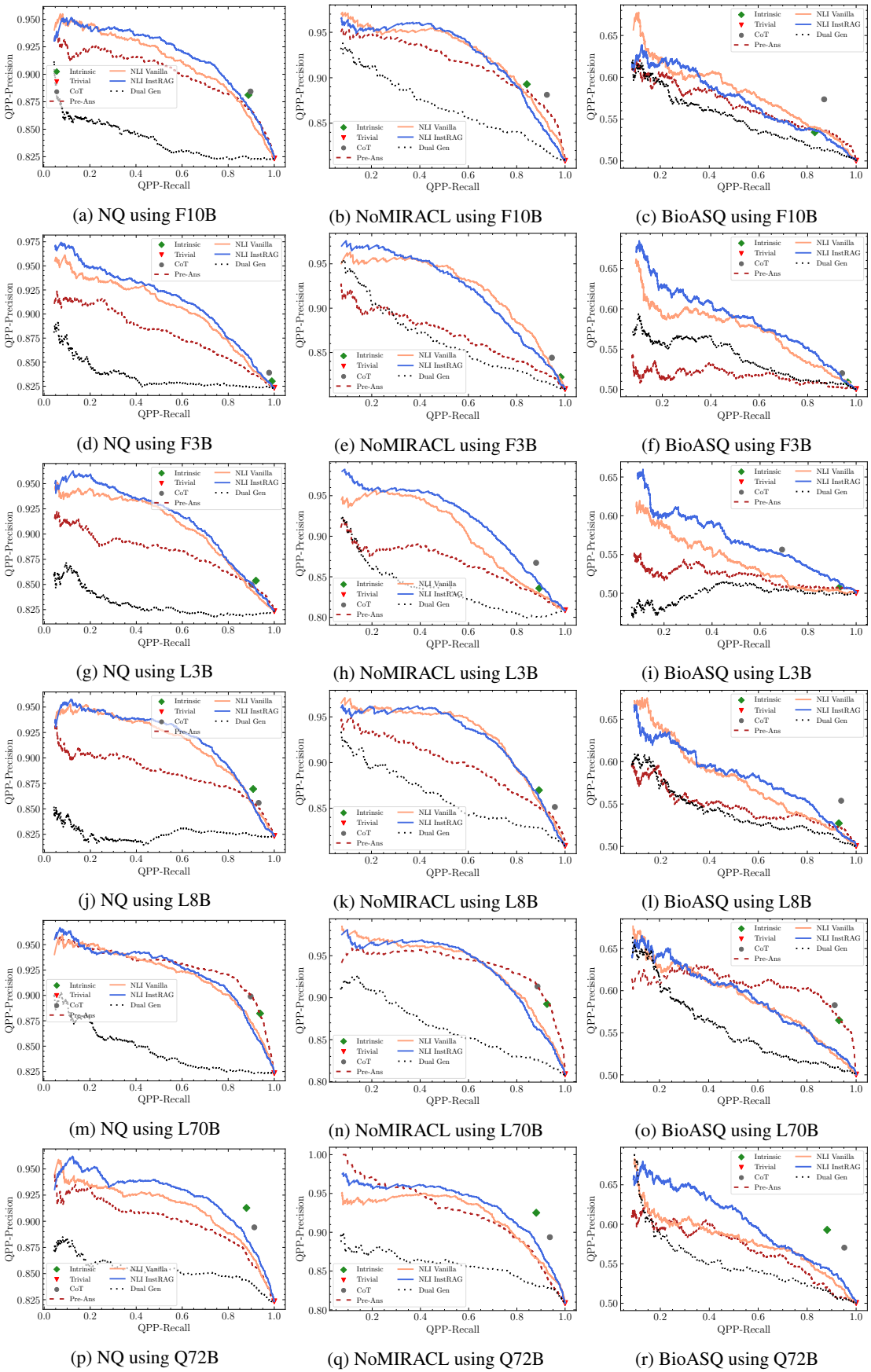
(g) NQ using L3B (h) NoMIRACL using L3B (i) BioASQ using L3B

(j) NQ using L8B (k) NoMIRACL using L8B (l) BioASQ using L8B

(m) NQ using L70B (n) NoMIRACL using L70B (o) BioASQ using L70B

(p) NQ using Q72B (q) NoMIRACL using Q72B (r) BioASQ using Q72B

Figure 5: AwF-Precision and AwF-Recall of AwF methods over different benchmark using different LLMs.



(a) NQ using F10B (b) NoMIRACL using F10B (c) BioASQ using F10B

(d) NQ using F3B (e) NoMIRACL using F3B (f) BioASQ using F3B

(g) NQ using L3B (h) NoMIRACL using L3B (i) BioASQ using L3B

(j) NQ using L8B (k) NoMIRACL using L8B (l) BioASQ using L8B

(m) NQ using L70B (n) NoMIRACL using L70B (o) BioASQ using L70B

(p) NQ using Q72B (q) NoMIRACL using Q72B (r) BioASQ using Q72B

Figure 6: QPP-Precision and QPP-Recall of AwF methods over different benchmark using different LLMs.