Question Difficulty Estimation for Large Language Models via Answer Plausibility Scoring

Jamshid Mozafari*, Bhawna Piryani, Adam Jatowt
University of Innsbruck, Innsbruck, Austria
{jamshid.mozafari, bhawna.piryani, adam.jatowt}@uibk.ac.at

Abstract

Estimating question difficulty is a critical component in evaluating and improving large language models (LLMs) for question answering (QA). Existing approaches often rely on readability formulas, retrieval-based signals, or popularity statistics, which may not fully capture the reasoning challenges posed to modern LLMs. In this paper, we introduce Q-DAPS (**Q**uestion **D**ifficulty based on **A**nswer **P**lausibility **S**cores) method, a novel approach that estimates question difficulty by computing the entropy of plausibility scores over candidate answers. We systematically evaluate Q-DAPS across four prominent QA datasets—TriviaQA, NQ, MuSiQue, and QASC—demonstrating that it consistently outperforms baselines. Moreover, Q-DAPS shows strong robustness across hyperparameter variations and question types. Extensive ablation studies further show that Q-DAPS remains robust across different plausibility estimation paradigms, model sizes, and realistic settings. Human evaluations further confirm strong alignment between Q-DAPS 's difficulty estimates and human judgments of question difficulty. Overall, Q-DAPS provides an interpretable, scalable, and bias-resilient approach to question difficulty estimation in modern QA systems.

https://github.com/DataScienceUIBK/Q-DAPS

1 Introduction

Questions are a fundamental means by which users express their information needs in Information Retrieval (IR) and Natural Language Processing (NLP) systems. They span a wide range of types—factoid, definition, and yes/no (Pandya and Bhatt, 2021)—and can also be classified by difficulty (Benedetto et al., 2023), such as easy, medium, or hard (Raina and Gales, 2024), or as simple vs. complex (Gabburo et al., 2024).

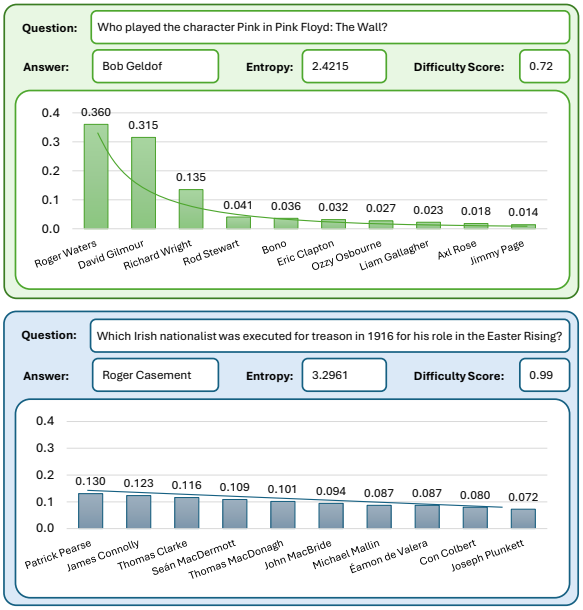

Figure 1: Two examples from the NQ (green) and TriviaQA (blue) datasets are presented, each showing the correct answer, 10 candidate answers (selected from 20 generated candidates), their normalized plausibility scores, and the computed difficulty score. The green colored example illustrates low entropy (an easier question), while the blue colored example demonstrates high entropy (a harder question).

Question difficulty reflects how challenging it is to answer a given question. Prior work has explored difficulty estimation using features such as readability (Naous et al., 2024), retrieval-based signals (Gabburo et al., 2024), and metrics derived from large language models (LLMs) (Dutulescu et al., 2024). However, most of these studies are primarily focused on estimating difficulty from the perspective of human readers or retrieval systems.

Our work differs in two key ways: (1) it provides an LLM-oriented difficulty measure grounded in how convincing incorrect answers appear to the LLM, rather than in surface-level text properties

*Corresponding Author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11124–11151
July 2-7, 2026 ©2026 Association for Computational Linguistics

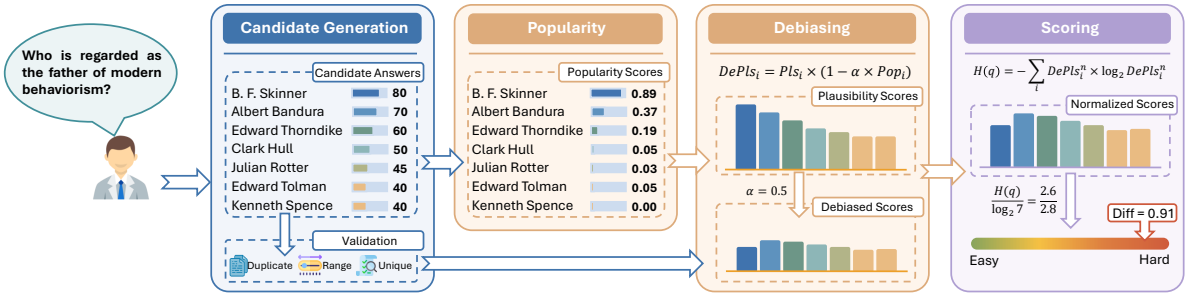

Figure 2: The Q-DAPS method comprises three stages: **Candidate Generation**, which produces candidate answers and their plausibility scores; **Popularity Debiasing**, which adjusts these plausibility scores based on candidate popularity; and **Scoring**, which computes the final difficulty score for the given question.

designed for humans or retrieval statistics tailored to IR systems, and (2) it directly links difficulty estimation to a critical application—hallucination risk detection—where harder questions are more likely to elicit fabricated or incorrect answers. This operational perspective motivates our method as a practical tool in high-stakes deployments: for model selection (e.g., choosing a stronger LLM if most domain questions are difficult), question routing (e.g., sending high-difficulty questions to a human reviewer in a company knowledge base), and safeguard triggering (e.g., requiring citations or user confirmation for exam questions).[1]

In this paper, we introduce ***Q**uestion **D**ifficulty based on **A**nswer **P**lausibility **S**cores* (Q-DAPS), a new method for estimating question difficulty tailored to LLMs. The core idea is to leverage candidate answers[2] and their associated plausibility scores (Mozafari et al., 2025a), which reflect how convincing each incorrect answer is. For example, for the question *What is the capital of China?*, the correct answer is *Beijing*, but two example candidates such as *Shanghai* and *Shenzhen* might also seem plausible, even though they are incorrect. However, *Shanghai* may appear more convincing due to its global prominence as China's largest city, and should therefore receive a higher plausibility score than *Shenzhen*.

We hypothesize that the entropy (Shannon, 1948) of normalized plausibility scores provides an effective signal for estimating question difficulty. High entropy suggests that many candidate answers are similarly plausible, indicating a harder question. In contrast, low entropy reflects a skewed distribution, where only a few candidates are highly plausible—indicating an easier question. Figure 1 shows two examples from Natural Questions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) datasets, with their normalized plausibility scores of candidate answers and the estimated difficulty scores.

An additional challenge in this setting is popularity bias (Klimashevskaia et al., 2024), which is well-known in recommender systems and has also been observed in LLM-generated outputs (Ni et al., 2025; Abe et al., 2025; Mallen et al., 2023). Yet, its impact on candidate answers and their plausibility scores for difficulty estimation remains unexplored. We address this gap by explicitly modeling and reducing popularity bias using Wikipedia page view statistics, to improve accuracy of Q-DAPS method.

As shown in Figure 2, the Q-DAPS proceeds in three stages. First, we generate candidate answers and assign plausibility scores[3] and validate them. Second, we measure their popularity using Wikipedia page view counts and adjust the plausibility scores to mitigate popularity bias. Finally, we compute the entropy of the debiased scores and normalize it to $[0, 1]$, producing an interpretable difficulty estimate. The estimate is interpretable because, in addition to the difficulty score, we return the candidate answers together with their plausibility scores, allowing users to understand how the difficulty was derived.

We evaluate Q-DAPS on a diverse set of datasets, covering simple question answering benchmarks such as TriviaQA (Joshi et al., 2017) and Natural Questions (Kwiatkowski et al., 2019), as

[1] In Appendix A, we provide a more detailed discussion of related works and compare our method with other works.

[2] Candidate answers refer to plausible but incorrect answers for a given question.

[3] Standard distractor generation for multiple-choice QA (Alhazmi et al., 2024) is unsuitable here, since it labels distractors as correct/incorrect without considering plausibility (Raina et al., 2023).

Candidate Answer Generation

Assume that you do not know that the answer to <question> is <ground_truth>. Generate a list of <N> unique candidate answers, excluding <ground_truth>. A plausibility score reflects how reasonable, credible, or contextually appropriate a candidate answer is with respect to the question.

For each candidate answer, provide:

1. a non-zero plausibility score between 0 and 100; and
2. a justification explaining the assigned plausibility score.

Return the output as a valid JSON list in the following format:

```
[
  {
    "Candidate Answer": "<candidate_answer>",
    "PlausibilityScore": <plausibility_score>,
    "Justification": "<justification>"
  }
]
```

The output must contain the JSON list only.

Figure 3: Prompt used for listwise candidate answer generation. <question> denotes the input question, <ground_truth> the correct answer, and <N> the number of candidates. Each <candidate_answer> is associated with a plausibility score (<plausibility_score>) and a justification (<justification>).

well as more complex reasoning benchmarks including MuSiQue (Trivedi et al., 2022) and QASC (Khot et al., 2020). We explore multiple scoring paradigms—pointwise, pairwise, and listwise—and compare Q-DAPS against a broad range of baseline methods. We also perform extensive ablation studies to examine the effects of different LLMs used for candidate answer generation and plausibility scoring, the role of the *Popularity Debiasing* component, and the impact of providing the *gold answer* during inference. The results demonstrate that the central hypothesis of Q-DAPS is robust: it performs effectively even without access to the gold answer, without popularity debiasing, and across different underlying LLMs. We further support our conclusions with two human evaluation studies reported in Section 4.4. Finally, we present an error analysis (Appendix H) and a frequently asked questions (FAQ) section (Appendix I) to address common reader concerns.

Our contributions are as follows:

1. We propose Q-DAPS, a novel LLM-oriented question difficulty metric based on the entropy of answer plausibility scores.
2. We show that popularity bias affects candidate answers and introduce a lightweight debiasing strategy that improves difficulty estimation.
3. We validate Q-DAPS through extensive experiments across multiple datasets, scoring settings, models, and human evaluations.

2 Q-DAPS Method

2.1 Candidate Generation

In the first stage, we prompt an LLM[4] to generate N candidate answers[5] along with their plausibility scores for a question. To generate candidate answers and their plausibility scores (Mozafari et al., 2025a), we use the prompt shown in Figure 3. We pass both the question and its gold answer to the *Candidate Generation* component, which is responsible for candidate generation. In this component, we also request justifications for the plausibility scores, as providing explanations has been shown to improve the reliability of generated outputs (Huang et al., 2023).

We pass the generated candidate answers and their plausibility scores to *Validation* component, which ensures their quality through the following steps: (1) ensuring there are no duplicate candidate answers, (2) verifying that plausibility scores fall within the valid range of 0 to 100, and (3) confirming that exactly N unique candidate answers are produced. Duplicates are identified using semantic similarity using the BEM method (Bulian et al., 2022). If any validation check fails, we increment the LLM's temperature by 0.1 and prompt it to regenerate the list. This process is repeated iteratively until all validation criteria are satisfied.

2.2 Popularity Debiasing

In this stage, we extract monthly Wikipedia page view counts for the generated candidate answers. These candidate answers are passed to the *Popularity* component, which computes their popularity based on the number of views of the corresponding Wikipedia page[6]. We use page view data spanning from January 1, 2015, to December 31, 2024, providing a fixed and consistent basis for computing popularity values.

[4]We use LLaMA 3.3 (Grattafiori et al., 2024) as the default generation core; however, the method is not heavily dependent on the choice of LLM, as demonstrated in the ablation study in the Section 4.5.

[5]We selected 20 as an upper limit because, in practice, LLMs are rarely able to generate more than 20 reasonable candidates for most questions.

[6]If a candidate answer does not have an associated Wikipedia page, its popularity is set to zero.

Given the high variability in Wikipedia page view counts, we normalize popularity scores to the $[0, 1]$ range. Outliers are removed using the interquartile range (IQR) method, following the approach of Mozafari et al. (2024b). To efficiently scale this computation, we leverage the HintEval (Mozafari et al., 2025c) toolkit, which enables parallel processing and significantly improves the runtime performance of Q-DAPS.

After computing the popularity scores, each candidate answer—along with its plausibility score and popularity value—is passed to the *Debiasing* component. This module adjusts the plausibility score of every candidate answer based on its popularity, following Equation 1:

$$\begin{aligned} DePls_i &= Pls_i - \alpha \times Pop_i \times Pls_i \\ &= Pls_i \times (1 - \alpha \times Pop_i) \end{aligned} \tag{1}$$

where Pls_i denotes the plausibility score of the i^{th} candidate answer, and Pop_i is its popularity. The hyperparameter α controls the strength of the debiasing adjustment. We multiply Pop_i by Pls_i to ensure that the debiasing effect is relative to the candidate's plausibility—since popularity is a fixed attribute, while plausibility can vary across questions. This computation is also parallelized to further increase the efficiency of Q-DAPS.

2.3 Score

In the final stage, we compute the entropy of the debiased plausibility scores ($DePls$) for all candidate answers. First, we normalize the scores to form a valid probability distribution:

$$DePls_i^{\text{norm}} = \frac{DePls_i}{\sum_i DePls_i} \tag{2}$$

to later compute the entropy:

$$H(q) = -\sum_i DePls_i^{\text{norm}} \times \log_2 DePls_i^{\text{norm}} \tag{3}$$

where $DePls_i^{\text{norm}}$ denotes the normalized debiased plausibility score of the i^{th} candidate answer. Finally, the computed entropy is passed to the *Normalization* component, which scales it to the range $[0, 1]$:

$$Diff_q = \frac{H(q)}{\log_2 N} \tag{4}$$

where N is the total number of candidate answers. For examples of the approach, we refer readers to Appendix B.

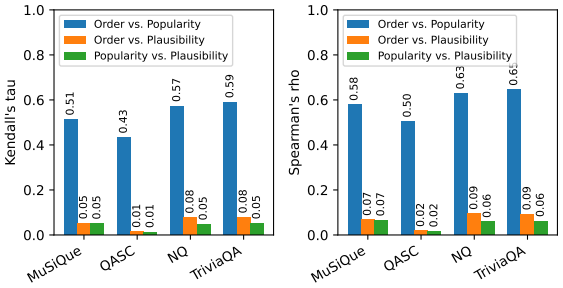

Figure 4: Kendall's τ (left chart) and Spearman's ρ (right chart) correlation coefficients across datasets comparing candidate answer ordering with Popularity, Plausibility scores, and their interplay.

3 Experimental Setup

3.1 Datasets

We use TriviaQA (Joshi et al., 2017) and Natural Questions (NQ) (Kwiatkowski et al., 2019) datasets for simple questions, as well as MuSiQue (Trivedi et al., 2022) and QASC (Khot et al., 2020) for complex questions. From each dataset, we sampled 2,000 questions using stratified sampling (Arnab, 2017) based on question types, which we identified following the method of Tayyar Madabushi and Lee (2016) to have a fair and balanced contribution from each dataset. We use only the questions and their corresponding gold answers from these datasets. Additional details are provided in Appendix C.

3.2 Models

To evaluate the effectiveness of the Q-DAPS method, we conduct experiments using ten different LLMs[7], grouped into five categories based on their parameter sizes. This grouping ensures that our results are not biased toward any particular LLM family or model scale. For a detailed description of the models and their categories, we refer the reader to Appendix D.

3.3 Metrics

3.3.1 Spearman's ρ Correlation

We categorize questions by the number of LLMs that successfully answer them. For each category, we compute the average difficulty score of its questions and then calculate Spearman's ρ correlation coefficient between the number of successful LLMs and the corresponding average

[7]We utilize the Together AI platform to access these LLMs through an API.

Scenario	MuSiQue		QASC		NQ		TriviaQA	
	d	ρ	d	ρ	d	ρ	d	ρ
Pointwise	0.0023	-0.1272	0.6214	-0.4848	1.098	-0.9	0.5557	-0.309
Pairwise	1.1072	-0.7727	0.3708	-0.4272	0.7808	-0.5454	0.3625	-0.2727
Listwise	**1.4335**	**-0.8909**	**1.1978**	**-0.6909**	**1.1486**	**-0.9636**	**0.9072**	**-0.6090**

Table 1: Performance of Q-DAPS across datasets under Pointwise, Pairwise, and Listwise settings. Columns report Cohen's d and Spearman's ρ. **Bold** values indicate the best performance based on the metrics. Full results appear in Appendix J, Tables 17–18.

difficulty. A more negative coefficient indicates stronger alignment between the estimated difficulty scores and actual model performance: as difficulty increases, the number of models able to answer the questions should decrease.

3.3.2 Cohen's d

We first define the median of the computed difficulty scores as a threshold τ, partitioning the set of questions Q into two equal-sized groups:

$$Q_{\text{Easy}} = \{q \in Q \mid \text{Diff}(q) \leq \tau\}, \quad Q_{\text{Hard}} = \{q \in Q \mid \text{Diff}(q) > \tau\} \tag{5}$$

Here, $\text{Diff}(q)$ denotes the difficulty score assigned to a question q. The group Q_{Easy} consists of lower-entropy (easier) questions, while Q_{Hard} consists of higher-entropy (harder) questions.

Next, for each question in both groups, we prompt an LLM (selected from the models described in Section 3.2), denoted as LLM_m, using the prompt shown in Figure 9 in Appendix E. The answer generated by LLM_m for a given question q is denoted as:

$$a_m(q) = \text{LLM}_m(q) \tag{6}$$

These generated answers are subsequently evaluated for correctness using the `GPT-Eval` method (Kamalloo et al., 2023), which leverages strong LLMs as the verification system. To mitigate potential evaluator bias arising from reliance on a single model, we employ three distinct LLMs as judges to improve reliability. Specifically, we use GPT-4 (OpenAI et al., 2024), Gemini 2.5 Flash (Comanici et al., 2025), and Claude Sonnet 4.5 (Anthropic, 2024). The final correctness label is determined via majority voting across the three judgments. Appendix E.1 details our motivation for adopting `GPT-Eval`, the evaluation procedure, and the exact prompt used. Formally, for a question q, an LLM-generated answer $a_m(q)$, and the gold answer gt_q, we define the semantic correctness indicator as:

$$\text{Eval}(q, a_m(q), gt_q) = \begin{cases} 1 & \text{if GPT-Eval is "Yes"} \\ 0 & \text{if GPT-Eval is "No"} \end{cases} \tag{7}$$

For each group $G \in \{Q_{\text{Easy}}, Q_{\text{Hard}}\}$, we then define the accuracy of LLM_m as the average `Eval` correctness score across its answers:

$$\text{Acc}_m(G) = \frac{1}{|G|} \sum_{q \in G} \text{Eval}\left(q, a_m(q)\right) \tag{8}$$

Finally, we use Cohen's d (Cohen, 2013) to measure the standardized difference between the two groups, providing an interpretable estimate of how clearly the method separates questions based on their difficulties:

$$d = \frac{\mu_{\text{Easy}} - \mu_{\text{Hard}}}{\sqrt{\frac{\sigma^2_{\text{Easy}} + \sigma^2_{\text{Hard}}}{2}}} \tag{9}$$

where

$$\mu_{\text{G}} = \frac{1}{M} \sum_{m}^{M} \text{Acc}_m(Q_{\text{G}}) \tag{10}$$

$$\sigma_{\text{G}} = \sqrt{\frac{1}{M} \sum_{m}^{M} \left(\text{Acc}_m(Q_{\text{G}}) - \mu_{\text{G}}\right)^2}$$

Where M refers to the set of LLMs described in Section 3.2. By definition, if the mean accuracy for the easy group exceeds that of the hard group, then $d > 0$; otherwise, $d < 0$. In other words, the larger Cohen's d is, the better Q-DAPS method separates easy and hard questions and vice versa. For additional clarity, Appendix E.2 includes a worked example, and Table 12 summarizes the standard interpretation of Cohen's d ranges.

4 Experiments and Results

In this section, we present a series of key experiments to evaluate the performance of Q-DAPS under various settings and to compare it against

Category	Method	MuSiQue		QASC		NQ		TriviaQA	
		d	ρ	d	ρ	d	ρ	d	ρ
Readability	Flesch-Kincaid (Flesch, 1948)	-0.543	0.5545	0.1496	0.1909	-0.424	0.6363	-0.2689	0.5181
	Gunning-Fog (Gunning, 1952)	-0.3947	0.7181	-0.0944	-0.0636	-0.5775	0.6272	-0.0963	0.2090
Prompt-based	LLaMA 3.1 8b (Grattafiori et al., 2024)	-0.535	0.2636	0.1065	-0.1272	0.0762	0.0818	0.361	-0.2545
	LLaMA 3.3 70b (Grattafiori et al., 2024)	0.2453	-0.109	0.2032	-0.2909	0.0307	-0.3363	0.4566	-0.4272
Popularity	PopQA (Mallen et al., 2023)	-0.0275	0.2818	-0.3206	0.2727	0.1535	0.2636	-0.2702	0.1727
Retriever-based	Retrieval Complexity (Gabburo et al., 2024)	0.1284	-0.3451	0.2225	-0.3126	0.2781	-0.4518	0.4394	-0.5129
Uncertainty-based	LLaMA 3.1 8b (Dutulescu et al., 2024)	0.1365	-0.3815	0.1543	-0.3926	0.1556	-0.5025	0.2211	-0.3121
	LLaMA 3.3 70b (Dutulescu et al., 2024)	0.4219	-0.5518	0.2119	-0.5621	0.3265	-0.5071	0.4823	-0.452
Q-DAPS	Avg-Plausibility	-0.2242	0.0272	0.4784	-0.3	0.1869	-0.2545	0.564	-0.509
	Entropy-Plausibility	**1.0888**	**-0.9001**	**0.803**	**-0.6181**	**0.9448**	**-0.9636**	**0.7498**	**-0.8818**

Table 2: Comparison of baselines and Q-DAPS across datasets. **Bold** indicates the best scores for each dataset. Full results are provided in Appendix J, Tables 21–22.

baseline methods, demonstrating its overall effectiveness. Additional experiments—including generalization (Appendix F.4), and α-robustness (Appendix F.5)—are also conducted and detailed in Appendix F. Error analysis is also detailed in Appendix H.

For experiments, we use the optimal values of α and the number of candidate answers, selected via grid search. Specifically, α is searched over the range $[0, 1]$ with a step size of 0.01, and the number of candidate answers is searched over the range $[1, 20]$ in increments of 1.

4.1 Analyzing Popularity Bias

To understand the influence of popularity on LLM-generated candidate answers, we examine three key quantities: (1) the popularity of each candidate answer, (2) the order in which LLMs produce candidate answers, and (3) their plausibility scores. Prior work (Mallen et al., 2023), has shown that LLMs exhibit popularity bias in their final answers, but it remains unclear whether such bias also appears during the generation of candidate answers.

We evaluate these relationships using Kendall's τ (Kendall, 1938) and Spearman's ρ (Spearman, 1904). Specifically, we measure the pairwise correlations between (1) popularity and plausibility scores, (2) the order of candidate answers and plausibility scores, and (3) the order of candidate answers and popularity.

The results, shown in Figure 4, reveal no significant correlation between plausibility and either popularity or generation order. This confirms that popularity and plausibility reflect fundamentally different properties and should not be used interchangeably. However, we observe a consistent correlation between popularity and the order of generated candidate answers, indicating that LLMs tend to generate more popular answers earlier. This provides clear evidence of a popularity bias already present at the candidate generation stage, which motivates our approach to correcting this effect when computing plausibility scores.

4.2 Answer Plausibility Estimation Methods

We investigate three approaches for estimating plausibility scores of candidate answers: *Pointwise*, *Pairwise*, and *Listwise*. We consider these approaches because they are three principled ways to elicit plausibility scores for a pool of candidate answers, each with a different characteristic. *Pointwise* prompts the model separately for each candidate answer, *Pairwise* compares candidates in pairs and aggregates preferences with the Bradley–Terry model (Bradley and Terry, 1952), and *Listwise* generates candidates and their plausibility scores jointly. A detailed description of these methods, along with prompts and formal definitions, is provided in Appendices F.1, F.2, and F.3.

For this experiment *only*, we sample 250 questions from each dataset using stratified sampling (Arnab, 2017) based on question types. Running this experiment on the full datasets would be computationally prohibitive; for instance, the *Pairwise* approach alone would require approximately 3.2 million prompts. Sampling 250 questions per dataset therefore enables a fair and tractable comparison of the different approaches under a fixed computational budget.

As shown in Table 1, among the three estimation strategies, the *Listwise* approach achieves the strongest overall performance. Beyond accuracy, it is also the most efficient, both in terms of the number of prompts and the average output length, re-

Evaluator	Age	Education Level	Easy	Hard
Evaluator 1	18	High School	0.65	0.68
Evaluator 2	36	High School	0.72	0.70
Evaluator 3	30	Bachelor's Degree	0.68	0.82
Evaluator 4	33	PhD	0.72	0.70
Evaluator 5	40	MSc	0.68	0.75
Evaluator 6	35	PhD	0.70	0.80
Mean Accuracy			**0.68**	**0.74**

Table 3: Per-evaluator accuracy in detecting the correct difficulty.

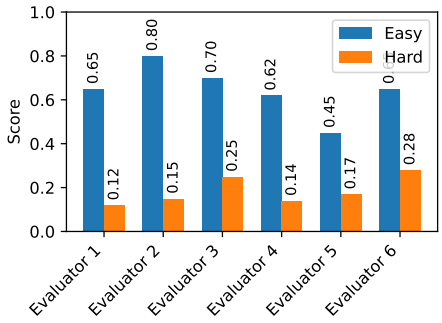

Figure 5: Results of human evaluation. The blue columns indicate each evaluator's accuracy on easy questions, while the orange columns represent their accuracy on hard questions.

quiring only a *single* prompt with complexity $\mathcal{O}(1)$, compared to $\mathcal{O}(n)$ for *Pointwise* and $\mathcal{O}(n^2)$ for *Pairwise*, as reported in Table 14 in Appendix F.1. Taken together, these findings establish *Listwise* as the most effective and practical method for plausibility-based difficulty estimation.

4.3 Q-DAPS Performance

Based on the closed relevant work discussed in Appendix A, we apply the following baselines:

Readability We include two traditional readability measures: Flesch-Kincaid (Flesch, 1948), based on average sentence length and syllable count, and Gunning-Fog (Gunning, 1952), which emphasizes complex word frequency and sentence length.

Prompt-based We prompt LLaMA 3.3 70B and LLaMA 3.1 8B with: *Please rate the difficulty of the question from 0 to 100, and respond with a number only*, to directly estimate difficulty.

Popularity We adopt PopQA (Mallen et al., 2023), which uses normalized Wikipedia page views as a proxy for question difficulty.

Retriever-based We include the method from Gabburo et al. (2024), which estimates

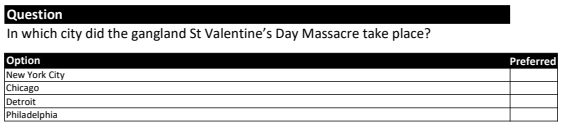

Figure 6: The template of the Excel sheet of questions for MCQA evaluation

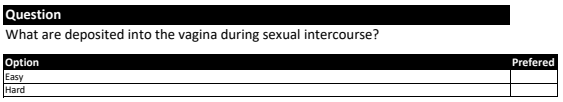

Figure 7: The template of the Excel sheet of questions for the Difficulty Ratings evaluation

difficulty based on answerability and completeness of passages retrieved by a retriever.

Uncertainty-based We include the method from Dutulescu et al. (2024), which estimates difficulty from the QA loss of LLMs, capturing model uncertainty when generating the correct answer.

Q-DAPS (ours) We consider two variants of our approach: *Avg-Plausibility*, which computes the difficulty score as the mean of the plausibility scores across candidate answers, and *Entropy-Plausibility*, which defines the difficulty score as the entropy of the plausibility distribution. We adopt the *Listwise* scenario as the primary configuration for Q-DAPS, as Section 4.2 shows that this scenario achieves the best performance.

Table 2 shows that estimating difficulty using the entropy of plausibility outperforms using the average plausibility. This indicates that entropy better captures the notion of difficulty by reflecting the uncertainty in the plausibility distribution. Furthermore, compared with all baselines, Q-DAPS achieves the highest performance, demonstrating its effectiveness in accurately modeling question difficulty. The second-best performance is obtained by the uncertainty-based method, followed by the retriever-based, which outperforms the prompt-based approach. These results indicate that estimating question difficulty is a non-trivial problem: directly querying an LLM or relying on simple heuristic computations is insufficient. Instead, more principled and novel modeling approaches are required to achieve strong performance.

4.4 Human Evaluation

To assess the alignment of Q-DAPS with human judgments, we conduct two complementary experi-

Category	Method	MuSiQue	QASC	NQ	TriviaQA
Readability	Flesch-Kincaid	-0.543	0.1496	-0.424	-0.2689
	Gunning-Fog	-0.3947	-0.0944	-0.5775	-0.0963
Prompt-based	LLaMA 3.1 8b	-0.535	0.1065	0.0762	0.361
	LLaMA 3.3 70b	0.2453	0.2032	0.0307	0.4566
Popularity	PopQA	-0.0275	-0.3206	0.1535	-0.2702
Retriever-based	Retrieval Complexity	0.1284	0.2225	0.2781	0.4394
Uncertainty-based	LLaMA 3.1 8b	0.1365	0.1543	0.1556	0.2211
	LLaMA 3.3 70b	0.4219	0.2119	0.3265	0.4823
Q-DAPS	Without gold answer	0.8325	0.5144	0.6319	0.6647
	With gold answer	**1.0888**	**0.803**	**0.9448**	**0.7498**

Table 4: Performance of Q-DAPS with and without providing gold answer as input, evaluated using Cohen's d and compared with baseline models. **Bold** values indicate the best performance for each dataset, and underlined values indicate the second-best.

Category	Method	MuSiQue	QASC	NQ	TriviaQA
Readability	Flesch-Kincaid	-0.543	0.1496	-0.424	-0.2689
	Gunning-Fog	-0.3947	-0.0944	-0.5775	-0.0963
Prompt-based	LLaMA 3.1 8b	-0.535	0.1065	0.0762	0.361
	LLaMA 3.3 70b	0.2453	0.2032	0.0307	0.4566
Popularity	PopQA	-0.0275	-0.3206	0.1535	-0.2702
Retriever-based	Retrieval Complexity	0.1284	0.2225	0.2781	0.4394
Uncertainty-based	LLaMA 3.1 8b	0.1365	0.1543	0.1556	0.2211
	LLaMA 3.3 70b	0.4219	0.2119	0.3265	0.4823
Q-DAPS	Without debiasing	0.894	0.5614	0.88	0.6511
	With debiasing	**1.0888**	**0.803**	**0.9448**	**0.7498**

Table 5: Performance of Q-DAPS with and without popularity debiasing, evaluated using Cohen's d and compared with baseline models. **Bold** values indicate the best performance for each dataset, and underlined values indicate the second-best.

ments. First, we convert a sampled set of questions into a multiple-choice QA format and measure human accuracy to examine whether questions labeled as difficult by Q-DAPS are also harder for humans. Second, we ask human annotators to directly judge question difficulty by choosing whether each question is easy or hard, allowing us to compare Q-DAPS 's difficulty labels with human binary difficulty judgments.

To prepare the evaluation set, we label all questions in each dataset using Q-DAPS and then sample 30 questions from each difficulty group (Easy and Hard) per dataset, resulting in 60 questions per dataset and 240 questions in total. Six human participants with diverse backgrounds took part in the evaluation.

4.4.1 Multiple-Choice QA

In this experiment, each sampled question is converted into a multiple-choice format with four answer options presented in random order. One option is the ground-truth answer, and the remaining three are selected from the most plausible candidate answers generated for that question. Figure 6 illustrates the Excel template used for collecting annotators' responses.

Figure 5 shows the results of the human MCQA evaluation. As expected, all evaluators achieved higher accuracy on questions labeled as Easy than on those labeled as Hard. This demonstrates that Q-DAPS effectively captures question difficulty in a manner consistent with human performance.

4.4.2 Question Difficulty Ratings

In the second experiment, we evaluate the extent to which human annotators agree with the difficulty labels assigned by Q-DAPS. Human evaluators are then asked to judge the difficulty of each sampled question by choosing whether it is easy or hard. Figure 7 illustrates the annotation template used to collect these judgments.

To quantify how well evaluators identify the intended difficulty level, we compute accuracy by comparing human judgments against the Q-DAPS-assigned labels for easy and hard questions. The results are summarized in Table 3. As shown, evaluators achieve higher accuracy on hard questions, indicating that harder questions are more consistently recognized by humans, whereas easy questions exhibit greater subjectivity in perceived difficulty.

Overall, the per-difficulty accuracy analysis results confirm that Q-DAPS produces difficulty estimates that align well with human intuition and are robust across different types of questions.

4.5 Ablation Study

In this experiment, we evaluate the performance of Q-DAPS by examining three factors: (1) the effect of removing the gold answer, (2) the impact of excluding the debiasing component, and (3) the influence of using different LLMs.

Without Gold Answers We evaluate Q-DAPS without providing the gold answer as input. Figure 16 in Appendix G shows the prompt used in this setting. Here, the LLM generates candidate answers without knowing the gold answer, reflecting realistic scenarios where gold answers are unavailable. As shown in Table 4, Q-DAPS still outperforms all baselines, although performance decreases compared to the setting with a gold answer. This is expected, as access to the gold answer allows the LLM to generate higher-quality candidate answers. Appendix G provides a detailed explanation of why including the gold answer leads to better performance. Overall, these results demon-

Category	Method	MuSiQue	QASC	NQ	TriviaQA
Readability	Flesch-Kincaid	-0.543	0.1496	-0.424	-0.2689
	Gunning-Fog	-0.3947	-0.0944	-0.5775	-0.0963
Prompt-based	LLaMA 3.1 8b	-0.535	0.1065	0.0762	0.361
	LLaMA 3.3 70b	0.2453	0.2032	0.0307	0.4566
Popularity	PopQA	-0.0275	-0.3206	0.1535	-0.2702
Retriever-based	Retrieval Complexity	0.1284	0.2225	0.2781	0.4394
Uncertainty-based	LLaMA 3.1 8b	0.1365	0.1543	0.1556	0.2211
	LLaMA 3.3 70b	0.4219	0.2119	0.3265	0.4823
Q-DAPS	Qwen 2.5 7b	0.8434	0.1465	0.2465	0.3162
	LLaMA 3.1 8b	0.5467	0.2484	0.3886	0.3481
	LLaMA 3.3 70b	**1.0888**	**0.803**	**0.9448**	**0.7498**

Table 6: Performance of different LLMs as the core of Q-DAPS, evaluated using Cohen's d and compared with baseline models. **Bold** values indicate the best performance for each dataset, and underlined values indicate the second-best. Full results in Appendix J, Tables 19–20.

strate that Q-DAPS remains effective even when gold answers are missing and does not overly rely on them.

Without Debiasing We evaluate Q-DAPS without the popularity debiasing component. In this variant, we compute the difficulty score directly from the entropy of the raw plausibility distribution. Table 5 shows that Q-DAPS continues to outperform all baselines even without debiasing, though performance is lower than the full method. This aligns with our earlier findings in Section 4.1, where we observed a correlation between candidate answer order and popularity; removing the debiasing step can therefore lead to less reliable plausibility estimates. Nonetheless, the strong performance of this variant indicates that Q-DAPS remains robust in domains where Wikipedia page views may not accurately capture popularity (e.g., medicine, finance, or specialized technical fields).

Influence of LLM Choice We evaluate how different LLMs influence the performance of Q-DAPS by using Qwen 2.5 7B (Qwen et al., 2025), LLaMA 3.1 8B (Grattafiori et al., 2024), and LLaMA 3.3 70B (Grattafiori et al., 2024) as the generation core. Table 6 shows that Q-DAPS performs strongly even with smaller models, consistently surpassing baseline methods. The only exception is the TriviaQA dataset, where Uncertainty-based scoring with LLaMA 3.3 70B slightly outperforms Q-DAPS —an expected outcome given its use of a significantly larger and more capable model. For the other datasets, LLaMA 3.1 8B and Qwen 2.5 7b achieve the second-best results across baselines, demonstrating that Q-DAPS remains effective even with relatively small LLMs. As anticipated, larger and more capable models yield stronger results overall, reflecting their superior generation quality—a trend well documented across NLP tasks. These findings indicate that Q-DAPS is not only effective with powerful LLMs but also practical in resource-constrained settings, supporting its usability and reproducibility in real-world applications.

Summary of Ablation Findings. Overall, Q-DAPS remains robust across all ablation settings. While larger LLMs and access to gold answers improve performance, the method continues to outperform all baselines even with smaller models and without gold answers, confirming its practicality in realistic scenarios. Popularity debiasing further enhances performance, but its removal does not break the method, indicating that debiasing is a beneficial refinement rather than a strict requirement. A paired t-test comparing the *With-Debiasing* and *Without-Debiasing* variants shows statistically significant improvements in difficulty separation ($p = 0.0356$, $t = 3.6474$, Cohen's d), reinforcing the effectiveness of the debiasing component.

5 Conclusion

We introduced Q-DAPS, an LLM-oriented method for estimating question difficulty based on the entropy of answer plausibility scores. By modeling how plausible competing candidate answers appear to an LLM and mitigating popularity bias, Q-DAPS captures difficulty in a way that reflects underlying reasoning uncertainty rather than surface-level textual features. Experiments across four QA benchmarks show that Q-DAPS consistently outperforms readability-based, popularity-based, retriever-based, prompt-based, and uncertainty-based baselines in separating easy and hard questions. Ablation studies indicate that the method remains robust across plausibility estimation paradigms, model sizes, and realistic settings without gold answers or popularity debiasing. Human evaluations further confirm strong alignment with human judgments of difficulty. Overall, Q-DAPS provides an interpretable and scalable method for question difficulty estimation in LLM-based QA systems, supporting applications such as hallucination risk estimation, adaptive model selection, question routing, and safeguard triggering. Future work will explore extensions to multilingual and multimodal settings, as well as tighter integration of difficulty estimates into adaptive and safety-aware QA pipelines.

Limitations

The Q-DAPS approach and its associated experiments have some limitations:

- **Question Type:** Q-DAPS is especially suited to question types for which a compact set of plausible candidates can be constructed, including entity, temporal, numeric, boolean, categorical/label selection, and multiple-choice (native or synthesized). These formats are common in enterprise and education contexts, making Q-DAPS broadly applicable without requiring extensive labeled data.

- **Language Scope:** Our study focuses solely on English-language questions. As a result, the methods and findings may not directly transfer to other languages. Further work is needed to assess the approach's effectiveness in multilingual or low-resource language settings.

- **Model Dependency:** The pipeline depends on LLMs, whose behavior, performance, and biases influence the outcomes. These models may encode societal or cultural biases and may behave inconsistently across different contexts.

Ethical Considerations

This study employs GPT models licensed under OpenAI and Apache 2.0, as well as LLaMA models governed by Meta's LLAMA Community License Agreement. We fully comply with all licensing terms. The datasets used are sourced from platforms permitting academic use. To support reproducibility and further research, we release all study materials under the MIT license. Throughout the project, we have ensured that data handling, model usage, and result dissemination conform to relevant ethical standards and legal requirements.

Acknowledgments

The computational results presented here have been achieved (in part) using the LEO HPC infrastructure of the University of Innsbruck and the Austrian Scientific Computing (ASC) infrastructure.

References

Kenya Abe, Kunihiro Takeoka, Makoto P. Kato, and Masafumi Oyamada. 2025. Llm-based query expansion fails for unfamiliar and ambiguous queries. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 3035–3039, New York, NY, USA. Association for Computing Machinery.

Abdalghani Abujabal, Rishiraj Saha Roy, Mohamed Yahya, and Gerhard Weikum. 2019. ComQA: A community-sourced dataset for complex factoid question answering with paraphrase clusters. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 307–317, Minneapolis, Minnesota. Association for Computational Linguistics.

Elaf Alhazmi, Quan Z. Sheng, Wei Emma Zhang, Munazza Zaib, and Ahoud Alhazmi. 2024. Distractor generation in multiple-choice tasks: A survey of methods, datasets, and evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14437–14458, Miami, Florida, USA. Association for Computational Linguistics.

Anthropic. 2024. Claude 3.5 sonnet model card. https://www.anthropic.com.

Raghunath Arnab. 2017. Chapter 7 - stratified sampling. In Raghunath Arnab, editor, *Survey Sampling Theory and Applications*, pages 213–256. Academic Press.

Luca Benedetto, Paolo Cremonesi, Andrew Caines, Paula Buttery, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2023. A survey on recent approaches to question difficulty estimation from text. 55(9).

Alexander Bondarenko, Magdalena Wolska, Stefan Heindorf, Lukas Blübaum, Axel-Cyrille Ngonga Ngomo, Benno Stein, Pavel Braslavski, Matthias Hagen, and Martin Potthast. 2022. CausalQA: A benchmark for causal question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3296–3308, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Manvi Breja and Sanjay Kumar Jain. 2022. A survey on non-factoid question answering systems. *International Journal of Computers and Applications*, 44(9):830–837.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. 2022. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 291–305, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yizhou Chi, Jessy Lin, Kevin Lin, and Dan Klein. 2024. Clarinet: Augmenting language models to ask clarification questions for retrieval.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Cohen. 2013. *Statistical power analysis for the behavioral sciences*, 2 edition. Routledge, London, England.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, and Others. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv e-prints*, page arXiv:2507.06261.

Andreea Dutulescu, Stefan Ruseti, Mihai Dascalu, and Danielle Mcnamara. 2024. How hard can this question be? an exploratory analysis of features assessing question difficulty using llms. In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 802–808, Atlanta, Georgia, USA. International Educational Data Mining Society.

Rudolf Franz Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233.

Matteo Gabburo, Nicolaas Paul Jedema, Siddhant Garg, Leonardo F. R. Ribeiro, and Alessandro Moschitti. 2024. Measuring retrieval complexity in question answering systems. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14636–14650, Bangkok, Thailand. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Others. 2024. The llama 3 herd of models.

Robert Gunning. 1952. *The Technique of Clear Writing*. McGraw–Hill, New York.

Wesley Hildebrandt, Boris Katz, and Jimmy Lin. 2004. Answering definition questions with multiple knowledge sources. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 49–56, Boston, Massachusetts, USA. Association for Computational Linguistics.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.

Yoshee Jain, John Hollander, Amber He, Sunny Tang, Liang Zhang, and John Sabatini. 2025. Exploring the potential of large language models for estimating the reading comprehension question difficulty.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Lance Johnson. 2025. Entropy increasing for nlp: Understanding its impact. BytePlus. Accessed: 2025-07-08.

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

Emil Kalbaliyev and Kairit Sirts. 2022. Narrative why-question answering: A review of challenges and datasets. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 520–530, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Aikaterini-Lida Kalouli, Rebecca Kehlbeck, Rita Sevastjanova, Oliver Deussen, Daniel Keim, and Miriam Butt. 2021. Is that really a question? going beyond factoid questions in NLP. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 132–143, Groningen, The Netherlands (online). Association for Computational Linguistics.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8082–8090.

Alka Khurana and Vasudha Bhatnagar. 2022. Investigating entropy for extractive document summarization. *Expert Systems with Applications*, 187:115820.

Anastasiia Klimashevskaia, Dietmar Jannach, Mehdi Elahi, and Christoph Trattner. 2024. A survey on popularity bias in recommender systems. *User Modeling and User-Adapted Interaction*, 34(5):1777–1834.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Yucheng Li. 2023. Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering.

Fengkai Liu and John Lee. 2023. Hybrid models for sentence readability assessment. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 448–454, Toronto, Canada. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. MQAG: Multiple-choice question answering and generation for assessing information consistency in summarization. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 39–53, Nusa Dua, Bali. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Jamshid Mozafari, Abdelrahman Abdallah, Bhawna Piryani, and Adam Jatowt. 2024a. Exploring hint generation approaches for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9327–9352, Miami, Florida, USA. Association for Computational Linguistics.

Jamshid Mozafari, Abdelrahman Abdallah, Bhawna Piryani, and Adam Jatowt. 2025a. Wrong answers can also be useful: Plausibleqa - a large-scale qa dataset with answer plausibility scores. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 3832–3842, New York, NY, USA. Association for Computing Machinery.

Jamshid Mozafari, Florian Gerhold, and Adam Jatowt. 2025b. Wikihint: A human-annotated dataset for hint ranking and generation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 3821–3831, New York, NY, USA. Association for Computing Machinery.

Jamshid Mozafari, Anubhav Jangra, and Adam Jatowt. 2024b. Triviahg: A dataset for automatic hint generation from factoid questions. SIGIR '24, page 2060–2070, New York, NY, USA. Association for Computing Machinery.

Jamshid Mozafari, Bhawna Piryani, Abdelrahman Abdallah, and Adam Jatowt. 2025c. Hinteval: A comprehensive framework for hint generation and evaluation for questions.

Tarek Naous, Michael J Ryan, Anton Lavrouk, Mohit Chandra, and Wei Xu. 2024. ReadMe++: Benchmarking multilingual language models for multi-domain readability assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12230–12266, Miami, Florida, USA. Association for Computational Linguistics.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2025. How knowledge popularity influences and enhances llm knowledge boundary perception.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, and Others. 2024. Gpt-4 technical report.

Ulrike Padó. 2017. Question difficulty – how to estimate without norming, how to use for automated grading. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–10, Copenhagen, Denmark. Association for Computational Linguistics.

Hariom A. Pandya and Brijesh S. Bhatt. 2021. Question answering survey: Directions, challenges, datasets, evaluation matrices.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and Others. 2025. Qwen2.5 technical report.

Vatsal Raina and Mark Gales. 2024. Question difficulty ranking for multiple-choice reading comprehension.

Vatsal Raina, Adian Liusie, and Mark Gales. 2023. Assessing distractors in multiple-choice tests. In *Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems*, pages 12–22, Bali, Indonesia. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Salomé A. Sepúlveda-Fontaine and José M. Amigó. 2024. Applications of entropy in data analysis and machine learning: A review. *Entropy*, 26(12).

C. E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423.

C. Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

Shuchang Tao, Liuyi Yao, Hanxing Ding, Yuexiang Xie, Qi Cao, Fei Sun, Jinyang Gao, Huawei Shen, and Bolin Ding. 2024. When to trust LLMs: Aligning confidence with response quality. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5984–5996, Bangkok, Thailand. Association for Computational Linguistics.

Harish Tayyar Madabushi and Mark Lee. 2016. High accuracy rule-based question classification using question syntax and semantics. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1220–1230, Osaka, Japan. The COLING 2016 Organizing Committee.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, and Others. 2025. Gemma 3 technical report.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, and Others. 2024. Gemma 2: Improving open language models at a practical size.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

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.

Mayi Xu, Yongqi Li, Ke Sun, and Tieyun Qian. 2024. Adaption-of-thought: Learning question difficulty improves large language models for reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5468–5495, Miami, Florida, USA. Association for Computational Linguistics.

Emre Yalcin and Alper Bilge. 2022. Evaluating unfairness of popularity bias in recommender systems: A comprehensive user-centric analysis. *Information Processing & Management*, 59(6):103100.

Leonidas Zotos, Hedderik van Rijn, and Malvina Nissim. 2025. Are you doubtful? oh, it might be difficult then! exploring the use of model uncertainty for question difficulty estimation.

A Related Works

Questions can be typically classified according to various criteria, including *Answer Type*, *Semantic Type*, *Question Structure*, *Intent*, and *Difficulty*.

Answer Type refers to the nature of the expected response, such as factoid (Kalouli et al., 2021) or non-factoid (Breja and Jain, 2022). **Semantic Type** considers the kind of information sought, for example, entities (Sciavolino et al., 2021), definitions (Hildebrandt et al., 2004), or causal explanations (Bondarenko et al., 2022). **Question Structure** describes the grammatical form of the question, including yes/no (Clark et al., 2019), wh-questions (Kalbaliyev and Sirts, 2022), and multiple-choice questions (Manakul et al., 2023). **Intent** captures the purpose of the question, for example to seek factual information (Abujabal et al., 2019) or to request clarification (Chi et al., 2024). Finally, **Difficulty** measures how challenging a question is to answer, ranging from straightforward fact-based queries (Padó, 2017) to complex reasoning problems (Xu et al., 2024).

A.1 Difficulty

Question difficulty can be assessed from different perspectives. One common perspective focuses on *readability*. Traditional readability-based approaches rely on lexical and syntactic features (Gunning, 1952; Flesch, 1948) but often fail to capture deeper semantic aspects. To address this, some studies apply machine learning techniques (Liu and Lee, 2023), while others leverage LLMs with prompt-based methods to estimate difficulty (Naous et al., 2024).

Another perspective involves *retrieval-based* methods, which evaluate difficulty according to how easily relevant passages can be retrieved. For instance, Mozafari et al. (2025b, 2024a) use BM25 (Robertson and Zaragoza, 2009) to retrieve the top-10 documents and classify a question as easy if one of these documents contained a correct answer. Similarly, Gabburo et al. (2024) expand this idea by incorporating notions of answerability and completeness.

A further line of work focuses on *LLM-based* metrics, such as model confidence and uncertainty (Dutulescu et al., 2024; Jain et al., 2025). For example, Zotos et al. (2025) use model uncertainty to estimate item difficulty for multiple-choice questions. However, these approaches have largely targeted question difficulty from a human perspective, rather than explicitly addressing how difficult a question is for an LLM to answer.

Table 7 summarizes existing perspectives on question difficulty estimation, representative approaches, and their limitations.

A.2 Entropy in NLP

Entropy has been applied in various NLP and Data Analysis tasks (Johnson, 2025; Sepúlveda-Fontaine and Amigó, 2024). For example, Khurana and Bhatnagar (2022) used entropy to measure sentence informativeness in extractive summarization, employing Non-negative Matrix Factorization to derive probability distributions over terms, topics, and sentences. Likewise, Li (2023) introduced Selective Context, a technique that filters out less informative content based on entropy-derived self-information, improving the efficiency of LLMs.

Recent research has also examined plausibility-based scoring of LLM responses, such as Verbalized Confidence (Tian et al., 2023) and CONQORD (Tao et al., 2024), but these approaches focus on confidence estimates for correct answers only. None of them examine candidate answers and their plausibility scores.

A.3 Popularity Bias

Popularity bias is a well-documented challenge in recommender systems, where algorithms tend to overrepresent highly popular items while underrepresenting niche or less-known items (Klimashevskaia et al., 2024; Yalcin and Bilge, 2022). In recommender systems, this can lead to unfair or skewed exposure of content.

In the context of LLMs, Mallen et al. (2023) showed that these models often struggle with less popular factual knowledge, indicating that popularity bias can influence their generated outputs. Similarly, Ni et al. (2025) demonstrated that LLMs tend to perform better, exhibit higher confidence, and more accurately perceive their knowledge boundaries when dealing with more popular knowledge, highlighting a strong correlation between knowledge popularity and QA performance. However, no previous work has investigated how popularity bias impacts question difficulty estimation for LLMs or proposed debiasing methods in this context.

Perspective	Representative Approaches	Surface-level	Retriever-based	Uncertainty-based	LLM-oriented	Popularity Debiasing	Interpretable
Readability	Traditional readability formulas (Gunning, 1952; Flesch, 1948) Hybrid ML-based methods (Liu and Lee, 2023) Prompt-based estimation (Naous et al., 2024)	✓	×	×	×	×	×
Retrieval-based	BM25 retrieval with top-10 documents (Mozafari et al., 2025b) Answerability and completeness metrics (Gabburo et al., 2024)	×	✓	×	×	×	×
LLM-based	Model confidence and uncertainty (Dutulescu et al., 2024) Uncertainty for MCQ difficulty (Zotos et al., 2025)	×	×	✓	×	×	×
Q-DAPS (ours)	Comparative plausibility over candidate answers with popularity debiasing	×	×	✓	✓	✓	✓

Table 7: Different perspectives on question difficulty estimation, with representative approaches, feature coverage, and our proposed Q-DAPS, which is the most complete.

B Case Study

In this section, we present an example that illustrates how the Q-DAPS method works by following the stages explained in Section 2. The question we use is *Who is regarded as the father of modern behaviorism?* from NQ (Kwiatkowski et al., 2019) dataset that its answer is *John B. Watson*. So, in the following, we explain each stage separately.

B.1 Candidate Generation

In this stage, we prompt LLaMA 3.3 (Grattafiori et al., 2024) using the template shown in Figure 3 to generate 20 candidate answers along with plausibility scores in the *Candidate Generation* component. The resulting outputs are presented in Table 8, showing all 20 candidate answers and their plausibility scores. After generation, these candidates are passed to the *Validation* component to ensure their quality through the following checks:

1. **No Duplicates:** We compare the generated candidate answers using a semantic equality function. As shown in Table 8, there are no duplicate answers.

2. **Range of Plausibility Scores:** We verify that all plausibility scores fall within the valid range of 0 to 100. Table 8 confirms this condition is satisfied.

3. **Number of Candidate Answers:** We check that the LLM produced exactly 20 candidate answers as requested. Table 8 confirms this requirement is met.

B.2 Popularity Debiasing

In this stage, we employ the HintEval toolkit (Mozafari et al., 2025c) to compute the popularity[8] of each candidate answer. The *Popularity* component retrieves the popularity scores from Wikipedia page view statistics, as shown in Table 8. Next, in the *Debiasing* component, we adjust the plausibility scores using these popularity scores according to Equation 1, with $\alpha = 0.5$[9].

For example, for the candidate answer *Ernst Hilgard*, we compute the debiased plausibility score as follows:

$$\begin{aligned} DePls_4 &= Pls_4 \times (1 - \alpha \times Pop_4) \\ &= 25 \times (1 - 0.5 \times 0.176) \\ &= 25 \times 0.912 \\ &= 22.800 \end{aligned} \tag{11}$$

Table 8 shows all resulting debiased plausibility scores.

B.3 Scoring

In the *Entropy* component, we first normalize the debiased plausibility scores to a probability distribution using Equation 2. Table 8 reports these normalized values. For example, for the candidate answer *Ernst Hilgard*, normalization proceeds as follows:

$$\begin{aligned} DePls_4^{\text{norm}} &= \frac{DePls_4}{\sum_i DePls_i} \\ &= \frac{22.800}{532.715} \\ &= 0.043 \end{aligned} \tag{12}$$

We then compute the entropy (Shannon, 1948) of the normalized debiased plausibility scores using Equation 3, obtaining $H(q) = 3.9702$. Finally, the *Normalization* component scales this entropy to produce the final difficulty score:

$$\begin{aligned} Diff_q &= \frac{H(q)}{\log_2 N} \\ &= \frac{3.9702}{4.3219} \\ &= 0.91 \end{aligned} \tag{13}$$

[8] HintEval refers to this as *Familiarity*.

[9] This hyperparameter is tunable; here we fix it at 0.5 for illustration.

#	Candidate Answer	Plausibility Score	Wikipedia Popularity	Debiased Plausibility	Normalized Debiased Plausibility
1	Sigmund Freud	10	1.000	5.000	0.009
2	Neal Miller	20	0.029	19.710	0.037
3	Jean Piaget	5	0.785	3.038	0.006
4	Ernst Hilgard	25	0.176	22.800	0.043
5	Ivan Pavlov	20	0.637	13.630	0.026
6	Carl Rogers	10	0.462	7.690	0.014
7	Lev Vygotsky	5	0.444	3.890	0.007
8	Albert Bandura	70	0.373	56.945	0.107
9	Harry Harlow	25	0.275	21.563	0.040
10	Edward Thorndike	60	0.198	54.060	0.101
11	Konrad Lorenz	30	0.180	27.300	0.051
12	B.F. Skinner	80	0.890	44.400	0.083
13	Gordon Allport	15	0.134	13.995	0.026
14	Walter Mischel	35	0.062	33.915	0.064
15	Edward Tolman	40	0.056	38.880	0.073
16	Clark Hull	50	0.050	48.750	0.092
17	Julian Rotter	45	0.030	44.325	0.083
18	Abraham Maslow	5	0.730	3.175	0.006
19	O. Hobart Mowrer	30	0.014	29.790	0.056
20	Kenneth Spence	40	0.007	39.860	0.075

Table 8: Candidate answers with their plausibility scores, Wikipedia popularity, debiased plausibility, and normalized debiased plausibility for the question *Who is regarded as the father of modern behaviorism?*. The candidate answer indicates the example being used in the case study.

Dataset	Train	Validation	Test
Simple			
TriviaQA	78,785	8,837	11,313
Natural Questions (NQ)	79,168	8,757	3,610
Complex			
MuSiQue	19,938	2,417	2,459
QASC	8,134	926	920

Table 9: Dataset statistics with standard train/validation/test splits.

C Dataset Details

In this section, we provide detailed descriptions and key statistics for the datasets used in our experiments. These resources represent a diverse mix of simple and complex QA benchmarks, covering a wide range of domains and question types. Together, they establish a robust evaluation suite for analyzing the effectiveness and generalizability of Q-DAPS method. The standard train/validation/test statistics for these datasets are summarized in Table 9.

TriviaQA TriviaQA (Joshi et al., 2017) is a large-scale QA dataset containing over 650,000 question–answer pairs collected from trivia websites. The questions are primarily factoid, requiring the identification of named entities or short factual responses. TriviaQA also includes approximately 95,000 questions that have been verified with supporting evidence from Wikipedia and the web.

Natural Questions (NQ) Natural Questions (Kwiatkowski et al., 2019) consists of around 300,000 real user queries submitted to Google Search, with annotations for both long and short answers over Wikipedia articles. Roughly 100,000 of these questions are categorized as factoid. In this work, we only focus on the subset of NQ questions included in AmbigQA (Min et al., 2020), which contains ambiguous questions paired with multiple valid factoid-style answers.

MuSiQue MuSiQue (Trivedi et al., 2022) (Multi-hop Structured Questions) is a multi-hop QA dataset with about 25,000 questions designed to require reasoning over multiple supporting facts. These questions are generally non-factoid and involve synthesizing evidence from different sources to reach the correct answer.

QASC QASC (Khot et al., 2020) is a multiple-choice QA benchmark focused on elementary science, featuring 9,980 questions that require reasoning over simple scientific facts. Each question has eight candidate answer choices, and answering often demands combining multiple facts.

Model	Category	Params	Context	Instr. Tuned	Weights Public	Provider	Year
LLaMA 3.2	Small	3B	8K	Yes	Yes	Meta	2024
Gemma 3	Small	4B	8K	Yes	Yes	Google	2024
Mistral 7B	Medium	7B	8K	Yes	Yes	Mistral AI	2023
Qwen 2.5	Medium	7B	32K	Yes	Yes	Alibaba	2024
LLaMA 3.1	Medium	8B	8K	Yes	Yes	Meta	2024
Mistral Small	Large	24B	32K	Yes	Yes	Mistral AI	2024
Gemma 2	Large	27B	8K	Yes	Yes	Google	2024
LLaMA 3.1	Very-Large	70B	8K	Yes	Yes	Meta	2024
Qwen 2.5	Very-Large	72B	32K	Yes	Yes	Alibaba	2024
GPT-4	Ultra-Large	N/A	N/A	Yes	No	OpenAI	2023

Table 10: Characteristics of the LMs evaluated in our experiments, including architectural scale, context window size, instruction tuning, providers, and availability to support reproducibility.

D Models Details

In this section, we provide more detailed information about the LMs used in our experiments. We categorize these LMs into five groups: *Small LMs*, *Medium LMs*, *Large LMs*, *Very-Large LMs*, and *Ultra-Large LMs*. Table 10 summarizes each LM along with its category and parameter size.

Small LMs *LLaMA 3.2 3B* (Grattafiori et al., 2024) is a compact yet capable model optimized for fast inference and moderate computational efficiency. *Gemma 3 4B* (Team et al., 2025) is a lightweight transformer model designed for efficient language processing while maintaining competitive performance on standard NLP benchmarks.

Medium LMs *Mistral 7B* (Jiang et al., 2023) is a high-performance, decoder-only model known for balanced efficiency and effectiveness in generative tasks. *Qwen 2.5 7B* (Qwen et al., 2025) is a 7-billion parameter model trained for robust text generation, excelling in multilingual and knowledge-intensive tasks. *LLaMA 3.1 8B* (Grattafiori et al., 2024) is an improved version of LLaMA designed for enhanced reasoning and generalization in language tasks.

Large LMs *Mistral 24B* (Jiang et al., 2023) is a larger variant designed for high-capacity reasoning and advanced generative performance, with a strong ability to follow complex instructions. *Gemma 2 27B* (Team et al., 2024) is a mid-scale model offering an excellent trade-off between capacity and computational cost, well-suited for general-purpose language tasks.

Very-Large LMs *LLaMA 3.1 70B* (Grattafiori et al., 2024) is a large-scale transformer model optimized for complex reasoning, long-form generation, and open-domain QA. *Qwen 2.5 72B* (Qwen et al., 2025) is a state-of-the-art multilingual model with extensive parameter tuning for diverse NLP applications.

Ultra-Large LMs *GPT-4* (OpenAI et al., 2024) is a highly sophisticated language model with superior reasoning, comprehension, and problem-solving abilities across multiple domains, representing a leading frontier in LLM research.

#	Question	Ground Truth	Difficulty Score	Group
1	Which is the most powerful chess piece?	Queen	$0.2 \leq 0.69$	Easy
2	What product of photosynthesis, a carbohydrate occurring in the cells of plants, can be changed into glucose or dextrine?	Starch	$0.12 \leq 0.69$	Easy
3	Which state did frontiersman Davy Crockett represent in the US House of Representatives?	Tennessee	$0.15 \leq 0.69$	Easy
4	In which James Bond film does actress Jane Seymour play Solitaire?	Live and Let Die	$0.19 \leq 0.69$	Easy
5	'Sufferin' succotash' is a catchphrase of which cartoon cat?	Sylvester	$0.43 \leq 0.69$	Easy
6	What county is Moran located in the state where Konza Prairie Biological Station is located?	Kansas	$0.97 > 0.69$	Hard
7	Who is the spouse of the director of Jump for Glory?	Miriam Cooper	$0.99 > 0.69$	Hard
8	The artist adding backing vocals to You're So Vain attended what grammar school?	Mick Jagger, Dartford	$0.98 > 0.69$	Hard
9	What is the name of the airport in the city where WILM is licensed to broadcast?	Wilmington International Airport	$0.95 > 0.69$	Hard
10	What gun was used by Pollack's director in Westworld?	LeMat revolver	$0.99 > 0.69$	Hard

Table 11: List of questions sampled from the TriviaQA and MuSiQue datasets, along with their ground truths, difficulty scores, and group labels. The green rows indicate easy questions, where the difficulty score is below the threshold, while the red rows indicate hard questions, where the difficulty score exceeds the threshold.

Evaluating the Generated Answer using LLM

Question: <question>
Answer: <ground_truth>
Candidate: <candidate>
Is the candidate correct? Choose between "Yes" or "No"

Figure 8: The placeholder `<question>` represents the question, `<ground_truth>` indicates the correct answer, and `<candidate>` shows the answer generated by different LLMs.

Cohen's d	Interpretation
$d < 0.00$	Very Weak
$0.00 \leq d < 0.20$	Weak
$0.20 \leq d < 0.50$	Modest
$0.50 \leq d < 0.80$	Moderate
$0.80 \leq d < 1.20$	Strong
$1.20 \leq d$	Very Strong

Table 12: Interpretation of Cohen's d thresholds

E Metrics Details

In this section, we provide detailed descriptions of the evaluation metrics used throughout the paper, including both `GPT-Eval` and Cohen's d.

E.1 GPT-Eval

In this section, we describe the motivation of using `GPT-Eval` instead of token-based metrics and then how it works.

In the era of LLMs, traditional metrics such as Exact Match (EM) or token-based *Contains* measures are often insufficient, since LLMs often generate correct answers in varied phrasings. For example, for the question *What is the term for a trained professional responsible for diagnosing and treating illnesses?*, answers like *Doctor*, *Medic*, or *Practitioner* are all acceptable, but EM or token-overlap metrics would fail to recognize their equivalence. LLM-based evaluators can handle such semantic equivalence, motivating our choice of `GPT-Eval` over traditional metrics.

In the `GPT-Eval` framework, we prompt an LLM using the template shown in Figure 8, asking it to determine whether a candidate answer is correct for a given question, with access to the corresponding ground-truth answer. The model outputs a binary judgment: *Yes* or *No*.

E.2 Cohen's d

In this section, we present an example to clarify the use of Cohen's d (Cohen, 2013) and illustrate our evaluation procedure. We follow the steps described in Section 3.3.2 to explain this example in detail. Specifically, we sampled 10 questions from the TriviaQA (Joshi et al., 2017) and MuSiQue (Trivedi et al., 2022) datasets, with $\alpha = 0.5$. Table 11 shows the sampled questions along with their gold answers, computed difficulty scores, and assigned groups. The median of these difficulty scores is 0.69, so we categorize questions with a difficulty score below 0.69 as *Easy* and those above 0.69 as *Hard*. As shown in the *Group* column of Table 11, five questions fall into the *Easy* group and five into the *Hard* group.

Next, we prompted the ten LLMs described in Section 3.2 to answer each question in both groups

Easy Questions						
LLM	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Acc**
LLaMA 3.2 3b	Queen	Starch	Tennessee	Live and Let Die	Popeye	0.8
LLaMA 3.1 8b	Queen	Starch	Tennessee	Live and Let Die	Sylvester the Cat	1.0
Mistral 7B	KNIGHT	STARCH	Tennessee	LIVE AND LET DIE	Garfield	0.6
Qwen 2.5 7b	Queen	Starch	Tennessee	Licence to Kill	Tom	0.8
LLaMA 3.1 70b	Queen	Starch	Tennessee	Live and Let Die	Sylvester	1.0
Qwen 2.5 72b	Queen	Starch	Tennessee	Live and Let Die	Sylvester	1.0
Gemma 2.0 27b	Queen	Starch	Tennessee	Live and Let Die	Sylvester	1.0
Mistral 24b	Queen	Starch	Tennessee	Live and Let Die	Sylvester	1.0
Gemma 3.0 4b	Queen	Starch	Texas	A View to a Kill	Tom Cat	0.4
GPT 4	Queen	Starch	Tennessee	Live and Let Die	Sylvester	1.0
Hard Questions						
LLM	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Acc**
LLaMA 3.2 3b	Kansas	Linda Gray	Charterhouse	Wilmington	Smith & Wesson	0.4
LLaMA 3.1 8b	Wabaunsee	Michael Jordan	Bedales	Wilmington	Colt Python	0.2
Mistral 7B	Kansas	NO ANSWER	Beverly Hills High School	NO ANSWER	GUNSLINGER	0.2
Qwen 2.5 7b	Douglas	Lynne Ramsay	Westminster School	Salisbury Airport	Smith & Wesson Model 29	0.0
LLaMA 3.1 70b	Geary	Michael Powell	Mick Jagger	Wilmington International Airport	Colt Peacemaker	0.4
Qwen 2.5 72b	Riley	Lena Waithe	Radley College	Newark Liberty International Airport	Colt Python	0.0
Gemma 2.0 27b	Riley	Jennifer Lopez	Miss Porter's School	Wilmington International Airport	Mauser C96	0.2
Mistral 24b	Riley	Sally Phillips	Radley College	New Castle	Colt Single Action Army	0.0
Gemma 3.0 4b	Finney	Lara	St. Paul's	Philadelphia International	Revolver	0.2
GPT 4	Riley	Tom Hanks	Mick Jagger	Wilmington International Airport	Colt Single Action Army	0.4

Table 13: LLM responses on both Easy and Hard questions with their accuracy scores. Green colored cells indicate correct answers.

using the prompt illustrated in Figure 9. Table 13 presents the answers generated by each LLM for both the *Easy* and *Hard* groups. As shown in this table, some generated answers are not lexically identical to the ground truth but are still considered correct, thanks to the use of `GPT-Eval` (Kamalloo et al., 2023) for semantic evaluation. For instance, for the question *'Sufferin' succotash' is a catchphrase of which cartoon cat?*, LLaMA 3.1 8B produces *Sylvester the Cat*, which is not lexically identical to the ground truth *Sylvester*, but is semantically equivalent and marked as correct by `GPT-Eval`. The *Acc* column shows the accuracy of each LLM as a QA system, calculated using Equation 8. As expected, the accuracy on easy questions is higher than on hard questions for each LLM, confirming that harder questions are indeed more challenging in practice.

We then computed the mean and standard deviation of the accuracy for each group separately. For the easy group, the mean accuracy is 0.86 with a standard deviation of 0.2; for the hard group, the mean is 0.2 with a standard deviation of 0.154. Finally, we calculate Cohen's d using Equation 9:

$$d = \frac{0.86 - 0.2}{\sqrt{\frac{0.2^2 + 0.154^2}{2}}} \quad (14)$$
$$= 3.697$$

According to Table 12, which shows the interpretation of Cohen's d values, $d = 3.697$ is considered **very strong**, indicating that our method computes the difficulty score reliably and effectively separates questions into *Easy* and *Hard* groups.

Answer Generation

System:
You are an assistant that answers questions. You just answer questions with exact and short answers. You do not use sentences as the response.

Shot (1):
Answer the question under conditions: 1) Answer should not be sentences. It should be some words. 2) Do not generate "sorry" or "I cannot ..." sentences. 3) Do not generate explanations, reasoning, or full sentences—only provide the exact answer.
Question: Who won the Nobel Peace Prize in 2009?
Answer: Barack Obama

Shot (2):
Answer the question under conditions: 1) Answer should not be sentences. It should be some words. 2) Do not generate "sorry" or "I cannot ..." sentences. 3) Do not generate explanations, reasoning, or full sentences—only provide the exact answer.
Question: Edouard Daladier became Prime Minister of which country in 1933?
Answer: France

Shot (3): ...
Shot (4): ...
Shot (5): ...

User:
Answer the question under conditions: 1) Answer should not be sentences. It should be some words. 2) Do not generate "sorry" or "I cannot ..." sentences. 3) Do not generate explanations, reasoning, or full sentences—only provide the exact answer.
Question: Who played the character Pink in Pink Floyd: The Wall?
Answer: <ANSWER>

Figure 9: This prompt uses five-shot examples to guide the LLM in answering questions effectively. The placeholder `<ANSWER>` represents the answer generated by the LLM for the user's question.

Plausibility Score Estimation for Pointwise Scenario

Assume that you are unaware that the answer to <question> is <ground_truth>. A plausibility score evaluates how reasonable, credible, or contextually appropriate the candidate answer is in relation to the given question.
For the candidate answer <candidate_answer>, provide:

1. A non-zero plausibility score as a number between 0 and 100.
2. A detailed explanation of the reasoning behind the plausibility score.

Format your response as a JSON object, where the candidate is represented as:

```
[
  {
    "Candidate Answer": "<candidate_answer>",
    "PlausibilityScore": <plausibility_score>,
    "Justification": "<justification>"
  }
]
```

The output must be a valid JSON object only.

Figure 10: The placeholder <question> represents the given question, while <ground_truth> denotes its correct answer. Candidate answer is represented by <candidate_answer>, along with a plausibility score (<plausibility_score>) and a justification (<justification>) explaining both the answer choice and the reasoning behind its assigned score.

F Additional Experiments

In this section, we present additional experiments and supplementary materials that support and extend our main findings.

F.1 Answer Plausibility Estimation Methods

To estimate the plausibility scores for candidate answers, we investigate three different methods: *Pointwise*, *Pairwise*, and *Listwise*. For the *Pointwise* scenario, we use the candidate answers generated using the prompt shown in Figure 3 and then prompt the LLM separately for each candidate answer to estimate its plausibility score, using the prompt shown in Figure 10. For the *Pairwise* scenario, we estimate plausibility scores by comparing pairs of candidate answers to determine which is more likely correct for a given question. Formally, given the set of candidate answers $\mathcal{C}_q$ for a question q, we construct the set of pairs as $\{(x_1, x_2) \mid x_1, x_2 \in \mathcal{C}_q,\ x_1 \neq x_2\}$. For each pair (x_1, x_2), we prompt the LLM using the prompt shown in Figure 12 in Appendix F.3 to compare them. After collecting preferences across all pairs for a given question, we apply the Bradley-Terry model (Bradley and Terry, 1952) to convert the pairwise comparisons into final plausibility scores.

Scenario	Runtime Complexity	Avg. Length
Pointwise	$\mathcal{O}(n)$	60.21
Pairwise	$\mathcal{O}(n^2)$	295.99
Listwise	$\mathcal{O}(1)$	55.91

Table 14: Comparison of runtime complexity and average output length (measured in words) across different scenarios for each question.

For the *Listwise* scenario, we use the prompt shown in Figure 3 to directly generate candidate answers along with their plausibility scores.

To compare the runtime complexity of each scenario, we consider each prompt as having complexity $\mathcal{O}(1)$. Under this assumption, the runtime complexity for the *Listwise* scenario is $\mathcal{O}(1)$ as it requires only a single prompt, while the *Pointwise* and *Pairwise* scenarios have complexities of $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$, respectively. This indicates that *Listwise* is also more efficient than the other scenarios in terms of runtime complexity.

Regarding the output length, the *Pointwise* and *Listwise* scenarios have approximately the same overall output length, since they ultimately produce the same candidate answer data, but *Pointwise* does so across n separate prompts, whereas *Listwise* produces it in a single prompt. In contrast, the output length for the *Pairwise* scenario is significantly larger because each pairwise comparison includes a separate justification. Table 14 summarizes this comparison between the scenarios.

In Appendix F.2 and F.3, we provide examples illustrating why *Pointwise* and *Pairwise* methods may not estimate plausibility scores effectively.

F.2 Pointwise Estimation

To estimate plausibility scores in the pointwise scenario, we use the prompt shown in Figure 10. In this prompt, the LLM is instructed to estimate a plausibility score for a single candidate answer without knowledge of the other candidate answers. As a result, the LLM evaluates each candidate independently.

Figure 11 presents an example illustrating the plausibility scores generated for the question *Which snooker player was simply known as 'The Grinder'?* from TriviaQA using the *Pointwise* and *Listwise* scenarios. As shown, most candidate answers in the pointwise scenario receive a plausibility score of one. This pattern appears frequently, indicating that the pointwise scenario struggles to estimate plausibility scores effectively and tends to as-

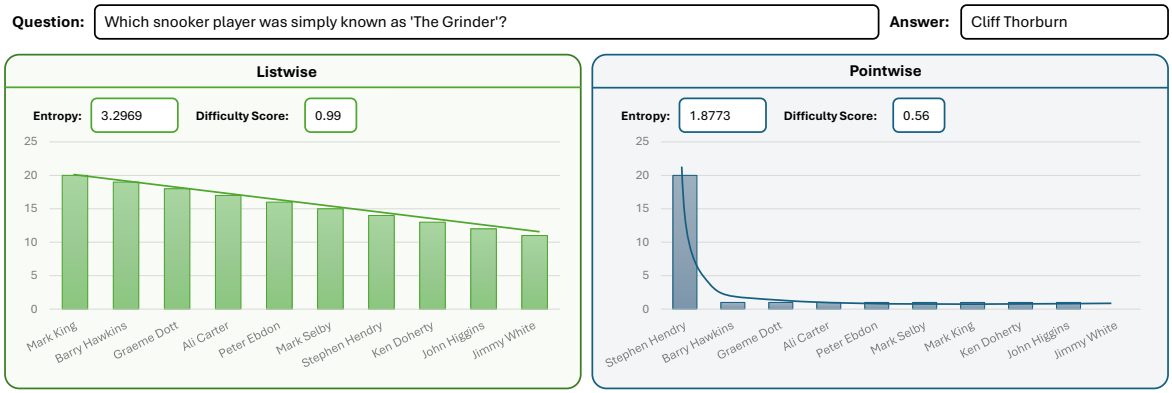

Figure 11: An example from TriviaQA of the pointwise scenario showing that most plausibility scores for the candidate answers are one, while in the listwise scenario, the candidate answers receive more diverse scores.

Plausibility Comparison in the Pairwise Scenario

Choose which of the two candidate answers is more likely to be correct based on the given question. A 'better candidate answer' is the one with a higher probability of being correct. Assume you do not know that the correct answer is <ground_truth>.

Question: <question>
Candidate Answer 1: <candidate_1>
Candidate Answer 2: <candidate_2>

Which is the better candidate answer? Respond with only "1" if `Candidate Answer 1` is better, or "2" if `Candidate Answer 2` is better. Provide a justification, and your final response must be a single character: "1" or "2".

Figure 12: The placeholder `<question>` represents the given question, while `<ground_truth>` denotes its correct answer. Two candidate answers, `<candidate_1>` and `<candidate_2>`, are compared, and the LLM is asked to select which one is more likely to be correct, along with a justification.

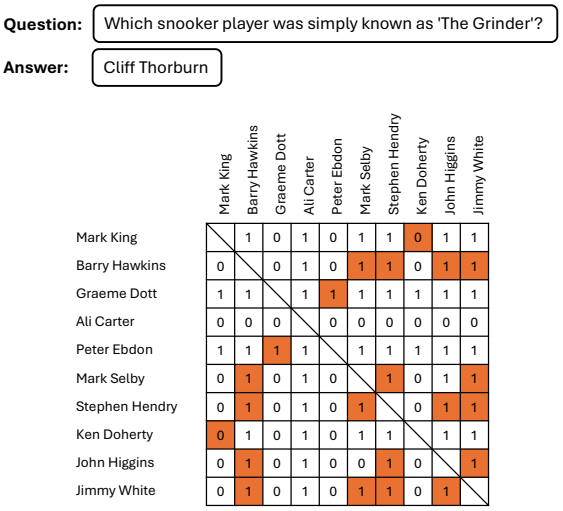

Figure 13: An example from TriviaQA of the pairwise scenario showing the pairwise comparison matrix. A value of 1 indicates that the candidate answer in the row is preferred over the candidate answer in the column, while a value of 0 indicates the opposite. Red cells highlight inconsistencies in the pairwise preferences.

sign a score of one to most candidates. In contrast, the listwise scenario produces a more diverse range of plausibility scores. Additionally, this figure demonstrates how the limitations of the pointwise scenario can affect the entropy and the final difficulty score. For the same question, the difficulty score under the pointwise scenario is 0.56, whereas under the listwise scenario it is 0.99. This discrepancy further motivates our preference for the listwise scenario beyond its computational efficiency.

F.3 Pairwise Estimation

To estimate plausibility scores in the pairwise scenario, we use the prompt shown in Figure 12, which presents a pair of candidate answers and asks the LLM to identify the more likely correct one.

Figure 13 presents an example illustrating the preferences generated for the question *Which snooker player was simply known as 'The Grinder'?* from TriviaQA using the *Pairwise* scenario. In this figure, a value of 1 indicates that the candidate answer in the row is preferred over the candidate answer in the column, while 0 indicates the opposite. As shown, there are many inconsistencies in the pairwise comparison matrix, highlighted in red. These inconsistencies may arise from the hallucination problem in LLMs. Furthermore, since the pairwise scenario requires a large number of prompts per question, the likelihood of hallucinations increases, which can directly affect the final pairwise scores by introducing more inconsistencies into the matrix. This discrepancy further

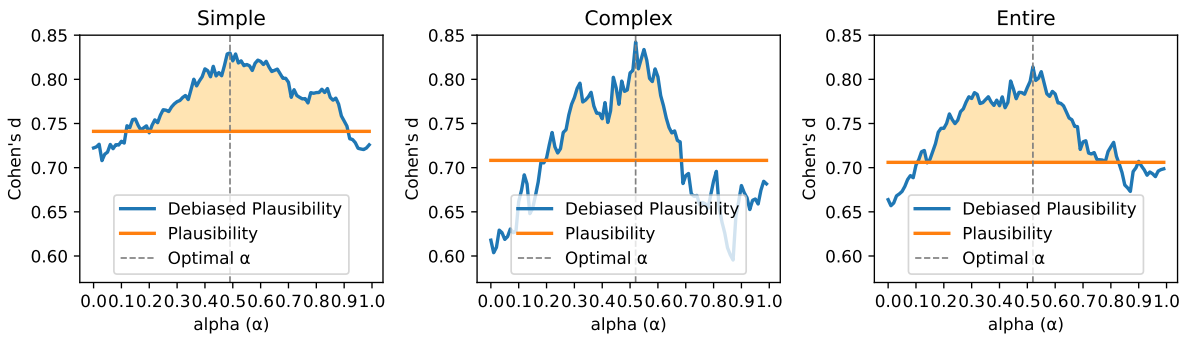

Figure 14: α robustness across different values from 0 to 1 on various question categories, including Simple, Complex, and Entire question types.

Question Type	Optimal α	Optimal # of Cans	d
Simple	0.49	7	0.8299
Complex	0.52	8	0.8423
Entire	0.52	8	0.8142

Table 15: Summary of optimal α, optimal number of candidates, and Cohen's d for simple, complex, and entire cases.

motivates our preference for the listwise scenario beyond its computational efficiency.

F.4 Generalization

In this experiment, we measure the most optimal number of candidate answers and the best α by grouping questions according to broader categories, rather than by dataset. Specifically, we compute these optimal values for question categories such as *Simple* (TriviaQA and NQ datasets) and *Complex* (QASC and MuSiQue datasets), which can provide more practical guidance for reproducibility and for users selecting settings based on their question types. Additionally, we determine the optimal values across the entire dataset (combining simple and complex questions) to provide general-purpose recommendations that can be applied regardless of question type.

Table 15 shows that the optimal α values for all cases are very close to each other and cluster around 0.5, suggesting that the method is not dependent on question type. Similarly, the optimal number of candidate answers is consistent across cases and is significantly smaller than the initial maximum of 20, indicating that generating a high number of candidates is not necessary to achieve strong results. Finally, the reported Cohen's d values show that, with these configurations, question difficulty separation remains very strong, highlighting the independence and robustness of the Q-DAPS method across different question types.

F.5 α Robustness

In this experiment, we demonstrate that the superiority of *Debiased-Plausibility* scenario is not limited to a specific configuration, particularly the parameter α. We categorize the datasets and questions into three main groups based on question type: *Simple* (TriviaQA and NQ datasets), *Complex* (QASC and MuSiQue datasets), and *Entire* (all datasets combined). We then compute the Cohen's d values for various α values ranging from 0 to 1, and compare them with the best configuration of the *Plausibility* scenario.

Figure 14 illustrates the Cohen's d values for these different scenarios. The results indicate that, across all cases and for most α values, the Cohen's d for *Debiased-Plausibility* exceeds that of the *Plausibility* scenario. This suggests that the Q-DAPS approach does not rely on a narrow configuration to achieve strong results, but rather outperforms the second-best method across a broad range of settings.

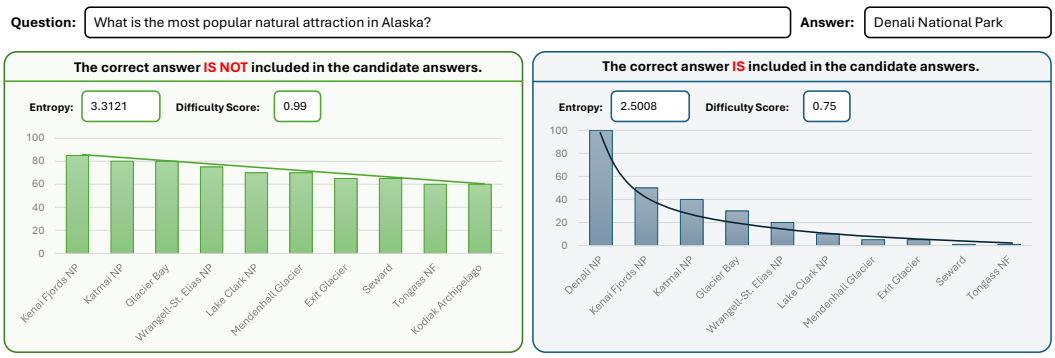

Figure 15: An example from the TriviaQA dataset illustrating how including the correct answer among the candidate answers can affect plausibility scores. The green box shows the case where the correct answer is excluded, while the blue box shows how including it can influence the plausibility distribution.

Candidate Answer Generation with Excluding the Correct Answer

Generate a list of 20 unique candidate answers. A plausibility score evaluates how reasonable, credible, or contextually appropriate each candidate answer is in relation to the given question. For each candidate, provide:

1. A non-zero plausibility score as a number between 0 and 100.
2. A detailed explanation of the reasoning behind the plausibility score.

Format your response as a JSON list, where each candidate is represented as:

```
[
  {
    "Candidate Answer": "<candidate_answer>",
    "PlausibilityScore": <plausibility_score>,
    "Justification": "<justification>"
  }
]
```

The output must be a valid JSON list only.

Figure 16: The placeholder `<question>` represents the given question. Each candidate answer is denoted by `<candidate_answer>`, accompanied by an initial plausibility score (`<plausibility_score>`) and a justification (`<justification>`) explaining both the answer choice and the rationale behind its plausibility score.

G Gold Answer Inclusion

In this section, we show the impact of including the correct answer among candidate answers using a motivating example. We prompt the LLaMA 3.3 (Grattafiori et al., 2024) model to generate 10 candidate answers[10] to the question *What is the most popular attraction in Alaska?*, whose answer is *Denali National Park*, under two scenarios:

1. We use our standard prompt for generating candidate answers and plausibility scores, as shown in Figure 3. This prompt explicitly instructs the LLM not to include the correct answer as a candidate, while still providing it in the prompt to guide plausibility estimates.
2. We modify the prompt to remove the instruction excluding the correct answer, allowing the LLM to rely on its knowledge to identify and potentially include the correct answer among the candidates. The modified prompt used for this setting is shown in Figure 16.

To ensure consistency and reproducibility, we set the temperature to zero in both scenarios, so the same candidate answers could be generated, with the only difference being whether the LLM was allowed to include the correct answer. Figure 15 shows the plausibility scores for both conditions.

As shown, including the correct answer in scenario 2 distorts the plausibility distribution: the correct answer receives a plausibility score of 100, while scores for other candidate answers drop significantly compared to scenario 1. This happens because the LLM assigns plausibility scores to incorrect candidates relative to the correct answer, leading to very low plausibility for all candidates. In contrast, scenario 1, where the model is explicitly instructed to ignore the correct answer as a candidate, allows the LLM to assign more balanced plausibility scores among the candidate answers.

This issue directly affects the entropy of the plausibility scores, and consequently the final difficulty score. For example, as shown in Figure 15, the entropy in scenario 1 is 3.3121, resulting in a normalized difficulty score of 0.99. However, for the same question and candidate answers in scenario 2, the entropy decreases to 2.5008, yielding a lower difficulty score of 0.75.

[10] We use 10 candidates for clearer visualization and analysis compared to all 20.

Pattern	Observed Behavior	Primary Cause	Implication
Entropy-overestimated difficulty	High entropy; QA models answer correctly	Dense set of semantically related plausible answers (e.g., list-based factual questions)	Entropy reflects answer-space richness rather than model uncertainty
Entropy-underestimated difficulty	Low entropy; QA models fail exact-match evaluation	Ambiguous or underspecified gold annotations (e.g., nationality, definitions)	Low entropy does not guarantee agreement with dataset labels

Table 16: Systematic divergence patterns between Q-DAPS entropy-based difficulty estimates and QA evaluation outcomes.

H Error Analysis

While Q-DAPS shows strong overall alignment with QA difficulty across datasets, its entropy-based estimates can occasionally diverge from downstream QA evaluation outcomes. This divergence reflects a fundamental distinction between measuring ambiguity in the plausible answer space and measuring success under a specific QA evaluation protocol. In this section, we analyze representative divergence patterns to clarify the conditions under which such misalignments arise.

We refer to these patterns as *entropy-overestimated difficulty* and *entropy-underestimated difficulty*. The former describes cases where Q-DAPS assigns a high difficulty score (high entropy) despite all QA models answering correctly, while the latter describes cases where Q-DAPS assigns a low difficulty score (low entropy) despite all QA models failing under exact-match evaluation.

H.1 Entropy-Overestimated Difficulty

Question. Types of skiing in the Winter Olympics 2018?

Gold Answer. Freestyle Skiing.

Analysis. For this question (Natural Questions ID: `nq_2902`), Q-DAPS assigns a high difficulty score due to elevated entropy in the candidate plausibility distribution. Multiple skiing-related disciplines receive high plausibility scores, including *Downhill* (0.9442), *Alpine Skiing* (0.9355), *Nordic Combined* (0.925), *Ski Jumping* (0.908), *Cross-Country* (0.901), and *Biathlon* (0.8765), resulting in a broad and competitive plausible answer space.

Despite this high-entropy signal, all evaluated QA models produced correct answers. Model predictions consistently enumerated the appropriate set of Olympic skiing disciplines, differing only in formatting or ordering.

This case exemplifies *entropy-overestimated difficulty*: high entropy reflects the richness of semantically related plausible answers rather than genuine difficulty for QA models. List-based factual questions naturally admit many closely related alternatives, which can inflate entropy even when the task is straightforward for modern QA systems.

H.2 Entropy-Underestimated Difficulty

Question. What nationality was Aristotle Onassis originally?

Gold Answer. Turkish.

Analysis. For this question (TriviaQA ID: `trivia_10433`), the candidate plausibility distribution is sharply peaked. *Greek* receives a plausibility score of 0.7402, while all other candidates receive substantially lower scores, resulting in low entropy and a low difficulty estimate.

Consistent with this signal, all evaluated QA models unanimously predicted *Greek*. However, the dataset gold answer specifies *Turkish*, reflecting birthplace-based nationality rather than ethnicity or cultural identity. As a result, all model predictions are penalized under evaluation.

This case illustrates *entropy-underestimated difficulty*: Q-DAPS correctly captures low ambiguity in the plausible answer space, but low entropy does not guarantee agreement with dataset annotations when gold labels encode implicit definitional or historical assumptions.

H.3 Summary of Divergence Patterns

Table 16 summarizes the two systematic divergence patterns observed between entropy-based difficulty estimation and downstream QA evaluation outcomes.

H.4 What This Means

These error patterns clarify the scope and intended interpretation of entropy-based difficulty estimation. Q-DAPS measures ambiguity in the plausible

answer space as perceived by LLMs, rather than the likelihood of producing a response that matches a specific gold annotation under exact-match evaluation. As a result, divergence from QA accuracy is expected in settings with semantically dense answer spaces or annotation-dependent definitions.

Importantly, these cases do not indicate failure of Q-DAPS; instead, they delineate the boundary between intrinsic question ambiguity and evaluation-specific success criteria. This distinction is particularly relevant for applications such as hallucination risk assessment, question routing, and model selection, where understanding uncertainty in the answer space is more informative than binary correctness.

I FAQ

This section addresses common questions readers may have regarding Q-DAPS and its design choices.

Is this method limited to factoid questions? No. As described in Section 3.1, Q-DAPS is evaluated on a diverse set of QA benchmarks. In addition to factoid datasets such as TriviaQA and Natural Questions, we include reasoning-oriented datasets such as QASC and multi-hop reasoning datasets like MuSiQue. The results reported in Section 4.3 and Table 2 show that Q-DAPS performs consistently across both simple and complex question types, indicating that it is not restricted to factoid questions.

Does Q-DAPS heavily rely on the generation quality of the LLM? While the quality of the underlying LLM affects candidate generation—as is common in LLM-based approaches—Q-DAPS does not rely on it exclusively. The ablation study in Table 6 demonstrates that Q-DAPS maintains strong performance even when using smaller models such as LLaMA 3.1 8B and Qwen 2.5 7B. Across most datasets, Q-DAPS consistently outperforms baselines regardless of model size, indicating that its effectiveness stems from the plausibility–entropy formulation rather than from a specific high-capacity LLM.

Does the difficulty assessment require significant computational cost? No. As analyzed in Section 4.2, the best-performing configuration adopts the *Listwise* plausibility estimation strategy. This strategy requires only a single prompt per question and therefore has $\mathcal{O}(1)$ prompt complexity. A direct comparison of computational complexity and output length across estimation strategies is provided in Table 14 (Appendix F.1), showing that Q-DAPS is computationally efficient and scalable.

Does the difficulty estimation require access to the correct answer? No. Q-DAPS can operate without access to the gold answer. As shown in the ablation study in Table 4, removing the gold answer leads to a moderate performance drop but Q-DAPS still outperforms all baseline methods. This setting reflects realistic scenarios where correct answers are unavailable, confirming that Q-DAPS does not rely on gold answers to function effectively (see also Appendix G for further analysis).

Does the debiasing step restrict applicability to answers with Wikipedia page views? No. Popularity debiasing is an optional refinement rather than a strict requirement. Table 5 shows that Q-DAPS remains competitive and continues to outperform baselines even when the debiasing component is removed. Section 4.1 further clarifies that debiasing addresses a specific popularity bias observed during candidate generation. In domains where Wikipedia page views are unavailable or inappropriate, the debiasing step can be safely omitted without compromising the core effectiveness of Q-DAPS.

What notion of question difficulty does Q-DAPS capture? Q-DAPS estimates *LLM-oriented difficulty*, defined as the degree of uncertainty exhibited by an LLM when multiple candidate answers appear similarly plausible. This notion is formalized in Section 2 and operationalized via entropy in Section 2.3. Unlike readability-based or retrieval-based difficulty, this definition directly reflects reasoning uncertainty from the model's perspective.

Why is entropy used instead of average plausibility? Entropy captures the distributional spread of plausibility scores rather than collapsing them into a single scalar. As shown in Table 2, entropy-based difficulty estimation consistently outperforms average plausibility across datasets. Section 2.3 explains how entropy reflects competition among plausible answers, while Appendix B provides a qualitative example motivating this choice.

How stable are difficulty scores across runs and configurations? Although candidate generation involves stochasticity, difficulty estimates are stable in aggregate. The robustness of Q-DAPS across different values of α and different numbers of candidate answers is demonstrated in Appendix F.5 and Appendix F.4. These experiments show consistent performance across a broad range of configurations.

How interpretable are the difficulty scores produced by Q-DAPS? Q-DAPS is inherently interpretable, as it exposes the candidate answers and their (debiased) plausibility scores that contribute to the final difficulty estimate. A concrete example illustrating this interpretability is provided in Appendix B, where each stage of the pipeline and its effect on the final difficulty score is shown step by step.

J Detailed Results

This section presents extended results that support and complement the analyses in the main part of the paper. In particular, we provide additional details for Section 4.2 through Table 17 (Cohen's d) and Table 18 (Spearman's ρ). Similarly, Section 4.5 is supplemented with Table 19 (Cohen's d) and Table 20 (Spearman's ρ).

We also include comprehensive performance breakdowns for a range of baseline approaches, as discussed in Section 4.3. These results are reported in Table 21 (Cohen's d) and Table 22 (Spearman's ρ). Each table additionally specifies the corresponding α and sample size N alongside the performance metrics.

Scenario	Method	MuSiQue			QASC			NQ			TriviaQA		
		α	N	d	α	N	d	α	N	d	α	N	d
Pointwise	Plausibility	–	15	-0.0639	–	6	**0.7836**	–	13	0.9824	–	14	0.1786
	Debiased-Plausibility	0.25	15	**0.0023**	0.80	14	0.6214	0.95	12	**1.098**	0.82	12	**0.5557**
Pairwise	Plausibility	–	6	0.8915	–	6	0.1504	–	4	0.6016	–	19	0.2405
	Debiased-Plausibility	0.12	6	**1.1072**	0.99	13	**0.3708**	0.77	19	**0.7808**	0.74	9	**0.3625**
Listwise	Plausibility	–	11	1.362	–	10	0.93	–	15	0.9607	–	5	0.7775
	Debiased-Plausibility	0.18	9	**1.4335**	0.56	14	**1.1978**	0.91	20	**1.1486**	0.22	4	**0.9072**

Table 17: Comparison of plausibility and debiased-plausibility methods across datasets using the Pointwise, Pairwise, and Listwise scenarios. The parameter α indicates the optimal weight used for debiasing popularity, N denotes the optimal number of candidate answers, and d represents the computed Cohen's d value. Gray cells indicate the best value for each dataset, while **bold** values highlight the best value for each scenario.

Scenario	Method	MuSiQue			QASC			NQ			TriviaQA		
		α	N	ρ	α	N	ρ	α	N	ρ	α	N	ρ
Pointwise	Plausibility	–	4	-0.0545	–	17	**-0.5**	–	13	-0.8909	–	17	-0.0181
	Debiased-Plausibility	0.39	4	**-0.1272**	0.99	14	-0.4848	0.10	15	**-0.9**	0.98	4	**-0.309**
Pairwise	Plausibility	–	9	-0.5818	–	4	-0.4272	–	17	-0.2818	–	17	-0.2727
	Debiased-Plausibility	0.89	6	**-0.7727**	0.00	4	**-0.4272**	0.99	17	**-0.5454**	0.00	17	**-0.2727**
Listwise	Plausibility	–	19	-0.6454	–	18	-0.6	–	20	-0.509	–	19	-0.5545
	Debiased-Plausibility	0.93	19	**-0.8909**	0.39	15	**-0.6909**	0.85	19	**-0.9636**	0.75	18	**-0.6090**

Table 18: Comparison of plausibility and debiased-plausibility methods across datasets using the Pointwise, Pairwise, and Listwise scenarios. The parameter α indicates the debiasing weight, N the number of candidate answers, and ρ the Spearman rank correlation. Gray cells indicate the best ρ for each dataset, while **bold** values highlight the best value within each scenario.

Model	Method	MuSiQue			QASC			NQ			TriviaQA		
		α	N	d	α	N	d	α	N	d	α	N	d
Qwen 2.5 7b	Plausibility	–	5	0.6193	–	4	-0.1797	–	6	-0.0191	–	7	0.1845
	Debiased-Plausibility	0.36	4	**0.8434**	0.36	6	**0.1465**	0.93	7	**0.2465**	0.29	6	**0.3162**
LLaMA 3.1 8b	Plausibility	–	10	0.0622	–	19	0.144	–	14	0.136	–	6	0.1754
	Debiased-Plausibility	0.51	20	**0.5467**	0.92	18	**0.2484**	0.01	18	**0.3886**	0.04	19	**0.3481**
LLaMA 3.3 70b	Plausibility	–	9	0.894	–	10	0.5614	–	6	0.88	–	4	0.6511
	Debiased-Plausibility	0.17	6	**1.0888**	0.61	12	**0.803**	0.21	6	**0.9448**	0.88	6	**0.7498**

Table 19: Comparison of plausibility vs. debiased plausibility methods across datasets for Qwen 2.5 7B, LLaMA 3.1 8b, and LLaMA 3.3 70b used as cores. The parameter α indicates the optimal weight used for debiasing popularity, N denotes the optimal number of candidate answers, and d represents the computed Cohen's d value. Gray cells indicate the best value for each dataset, while **bold** values highlight the best value for each core.

Model	Method	MuSiQue			QASC			NQ			TriviaQA		
		α	N	ρ	α	N	ρ	α	N	ρ	α	N	ρ
Qwen 2.5 7B	Plausibility	–	4	-0.29	–	20	0.3818	–	8	-0.3181	–	16	-0.509
	Debiased-Plausibility	0.98	4	**-0.7181**	0.98	11	**0.0363**	0.07	6	**-0.9636**	0.28	19	**-0.7636**
LLaMA 3.1 8B	Plausibility	–	18	0.7545	–	5	-0.4263	–	18	0.4545	–	7	0.2
	Debiased-Plausibility	0.22	20	**-0.7272**	0.69	19	**-0.59**	0.95	19	**-0.8454**	0.41	10	**-0.8181**
LLaMA 3.3 70B	Plausibility	–	5	-0.8363	–	15	-0.4545	–	19	-0.8909	–	14	-0.7272
	Debiased-Plausibility	0.39	6	**-0.9001**	0.54	12	**-0.6181**	0.61	12	**-0.9636**	0.48	12	**-0.8818**

Table 20: Comparison of plausibility vs. debiased plausibility methods across datasets for Qwen 2.5 7B, LLaMA 3.1 8B, and LLaMA 3.3 70B used as cores. The parameter α indicates the optimal weight used for debiasing popularity, N denotes the optimal number of candidate answers, and ρ the Spearman rank correlation. Gray cells highlight the best ρ for each dataset, while **bold** values mark the best within each core.

Category / Method	MuSiQue			QASC			NQ			TriviaQA		
	α	N	d	α	N	d	α	N	d	α	N	d
Readability												
Flesch-Kincaid (Flesch, 1948)	–	–	-0.543	–	–	0.1496	–	–	-0.424	–	–	-0.2689
Gunning-Fog (Gunning, 1952)	–	–	-0.3947	–	–	-0.0944	–	–	-0.5775	–	–	-0.0963
Prompt-based												
LLaMA 3.1 8b (Grattafiori et al., 2024)	–	–	-0.535	–	–	0.1065	–	–	0.0762	–	–	0.361
LLaMA 3.3 70b (Grattafiori et al., 2024)	–	–	0.2453	–	–	0.2032	–	–	0.0307	–	–	0.4566
Popularity												
PopQA (Mallen et al., 2023)	–	20	-0.0275	–	4	-0.3206	–	4	0.1535	–	16	-0.2702
Retriever-based												
Retrieval Complexity (Gabburo et al., 2024)	–	–	0.1284	–	–	0.2225	–	–	0.2781	–	–	0.4394
Uncertainty-based												
LLaMA 3.1 8b (Dutulescu et al., 2024)	–	–	0.1365	–	–	0.1543	–	–	0.1556	–	–	0.2211
LLaMA 3.3 70b (Dutulescu et al., 2024)	–	–	0.4219	–	–	0.2119	–	–	0.3265	–	–	0.4823
Avg-Plausibility												
Plausibility	–	18	-0.4173	–	10	0.2724	–	19	0.1663	–	16	0.5061
Debiased-Plausibility	0.82	20	-0.2242	0.53	13	0.4784	0.11	19	0.1869	0.57	16	0.564
Entropy-Plausibility												
Plausibility	–	9	**0.894**	–	10	**0.5614**	–	6	**0.88**	–	4	**0.6511**
Debiased-Plausibility	0.17	6	**1.0888**	0.61	12	**0.803**	0.21	6	**0.9448**	0.88	6	**0.7498**

Table 21: Comparison of various baselines and the Q-DAPS approach across datasets based on the Cohen's d. Gray cells indicate the highest score in each dataset, while **bold** values highlight the second-highest score.

Category / Method	MuSiQue			QASC			NQ			TriviaQA		
	α	N	ρ	α	N	ρ	α	N	ρ	α	N	ρ
Readability												
Flesch-Kincaid (Flesch, 1948)	–	–	0.5545	–	–	0.1909	–	–	0.6363	–	–	0.5181
Gunning-Fog (Gunning, 1952)	–	–	0.7181	–	–	-0.0636	–	–	0.6272	–	–	0.2090
Prompt-based												
LLaMA 3.1 8b (Grattafiori et al., 2024)	–	–	0.2636	–	–	-0.1272	–	–	0.0818	–	–	-0.5545
LLaMA 3.3 70b (Grattafiori et al., 2024)	–	–	0.109	–	–	-0.2909	–	–	-0.3363	–	–	-0.4272
Popularity												
PopQA (Mallen et al., 2023)	–	19	0.2818	–	4	0.2727	–	18	0.2636	–	6	0.1727
Retriever-based												
Retrieval Complexity (Gabburo et al., 2024)	–	–	-0.3451	–	–	-0.3126	–	–	-0.4518	–	–	-0.5129
Uncertainty-based												
LLaMA 3.1 8b (Dutulescu et al., 2024)	–	–	-0.3815	–	–	-0.3926	–	–	-0.5025	–	–	-0.3121
LLaMA 3.3 70b (Dutulescu et al., 2024)	–	–	-0.5518	–	–	-0.5621	–	–	-0.5071	–	–	-0.452
Avg-Plausibility												
Plausibility	–	4	0.0818	–	14	0.0909	–	20	-0.2545	–	16	-0.509
Debiased-Plausibility	0.02	10	0.0272	0.98	10	-0.3	0	20	-0.2545	0	16	-0.509
Entropy-Plausibility												
Plausibility	–	5	**-0.8363**	–	15	**-0.4545**	–	19	**-0.8909**	–	14	**-0.7272**
Debiased-Plausibility	0.39	6	**-0.9001**	0.54	12	**-0.6181**	0.61	12	**-0.9636**	0.48	12	**-0.8818**

Table 22: Comparison of various baselines and the Q-DAPS approach across datasets based on the Spearman's ρ. Gray cells indicate the lowest score in each dataset, while **bold** values highlight the second-lowest score.