https://doi.org/10.70844/ijas.2025.2.29

Innovative Journal of Applied Science

Research Article ISSN: 3049-3072

IJAS-25-090

Evaluating Log-Likelihood for Confidence Estimation in LLM-Based Multiple-Choice Question Answering

Christopher Boseak*

Independent Researcher, Dallas, TX, USA

Corresponding Author: Christopher Boseak, Independent Researcher, Dallas, TX, USA, E-mail: cboseak@gmail.com

Received date: 07 July, 2025, **Accepted date:** 21 July, 2025, **Published date:** 28 July, 2025

Citation: Boseak C (2025) Evaluating Log-Likelihood for Confidence Estimation in LLM-Based Multiple-Choice Question Answering. Innov J Appl Sci 2(4): 29.

Abstract

Reliable deployment of Large Language Models (LLMs) in question-answering tasks requires well-calibrated confidence estimates. This work investigates whether token-level log-likelihoods—sums of log-probabilities over answer tokens—can serve as effective confidence signals in Multiple-Choice Question Answering (MCQA). We compare three methods: (1) Raw log-likelihood, (2) length-normalized log-likelihood and (3) conventional softmax-based choice probability. Across four diverse MCQA benchmarks, we find that no single scoring method is universally best. Length normalization can significantly improve calibration but may reduce accuracy, while softmax and raw log-likelihood yield identical predictions. These results highlight important trade-offs between calibration and accuracy and offer insights into selecting or adapting confidence measures for different tasks. Our findings inform the design of more trustworthy LLM-based QA systems and lay groundwork for broader uncertainty quantification efforts.

Keywords: Artificial intelligence, Machine learning, Large Language Models (LLMs), Confidence estimation, Log-likelihood, Calibration, Multiple-Choice Question Answering (MCQA), Softmax, Uncertainty quantification, Model reliability, Answer scoring methods, NLP evaluation

Introduction

Large Language Models (LLMs) exhibit impressive performance in knowledge and reasoning tasks, but assessing the model's confidence in its answers remains challenging. An ideal LLM would not only provide an answer but also a well-calibrated probability of correctness—enabling users to trust or verify the outputs. However, existing LLMs are often poorly calibrated and sometimes express high confidence in incorrect answers and low confidence in correct ones. This miscalibration poses risks in high-stakes applications where decision-making relies on model confidence [1].

Multiple-Choice Question Answering (MCQA) is a setting where LLMs must choose the correct answer from a fixed set of options. MCQA provides a natural way to extract confidence scores: One can compute the probability the model assigns to each option. Typically, this is done by converting the model's token-level probabilities into a normalized softmax score over choices, which serves as the model's confidence in its selected answer. Yet, using model probabilities for scoring often requires task-specific heuristics like answer length normalization or other calibrations. This is because the raw log-likelihood of a sequence is biased by its length — longer answer options tend to have lower total probability (product of token probabilities), even if they might be the correct answer. As a result, practitioners frequently normalize log-likelihoods by the number of tokens to mitigate length bias [2].

In this work, we investigate how the choice of confidence scoring method affects the performance and calibration of LLMs on MCQA. In particular, we evaluate using the raw log-likelihood of the model's predicted answer as a confidence signal, versus a length-normalized log-likelihood and compare these to the conventional softmax-based confidence. We focus on the question: Can token log-likelihood (with or without normalization) serve as a reliable confidence measure in place of the usual softmax probability and what are the trade-offs?

We use the pre-trained Meta-Llama-3.1-8B large language model to score answer options. Specifically, we query the 8B-parameter LLaMA-3.1 model in a zero-shot setting *via* a completion API, extracting token log-probabilities. The model is treated as a black box scoring function that provides P (wt | context) for each next token. We ensured that the model's prompt format was fixed across all methods; each answer is scored independently using the same prompt template to avoid any ordering bias. The model is used as-is without fine-tuning on the target datasets, so its calibration reflects that of the original pre-trained model. While Meta-Llama-3.1-8B may include some instruction tuning, we treat it as a standard language model without explicit alignment or additional calibration.

Contributions

- We provide a head-to-head comparison of raw log-likelihood, normalized log- likelihood and softmax-based confidence as uncertainty measures for LLMs in MCQA.
- We evaluate model calibration under each scoring method using Expected Calibration Error (ECE), revealing how each method may under- or overestimate true correctness likelihood [3].

Copyright © 2025 | ijas.meteorpub.com

https://doi.org/10.70844/ijas.2025.2.29

Innovative Journal of Applied Science

- We analyze the impact on accuracy, highlighting cases where length normalization changes the model's predicted answer and affects overall performance.
- We offer insights into when length normalization helps (e.g., reducing overconfidence on datasets with highly variable answer lengths) and when it hurts (e.g., when correct answers are generally shorter or when the model is already well- calibrated).

The results show that while raw log-likelihood and softmax confidence yield identical predictions (and thus identical accuracy), length normalization can substantially alter model behavior. For instance, on the HellaSwag common-sense dataset, using normalized log-likelihood reduces the model's overconfidence (ECE drops from 37.0% to 4.4%) but also slightly lowers accuracy (from 28.9% to 27.0%). In contrast, on an easier dataset (ARC-Easy), normalized scoring significantly hurts accuracy (dropping from 69.4% to 53.7%) and slightly worsens calibration. These findings suggest that log-likelihood can serve as a viable confidence signal, but its effective use requires careful normalization or calibration strategies depending on the dataset.

Related Work

Confidence Calibration in Language Models. Calibration of neural network pre- dictions has long been studied in classification tasks [3]. Guo et al., showed modern neural networks are often overconfident and proposed metrics such as Expected Calibration Error (ECE) to quantify miscalibration [4]. For pre-trained language models, found that models like BERT are also mis- calibrated on NLP tasks and that calibration can degrade out-of-domain [5]. Recent work has extended calibration studies to large language models. Specifically examined whether language models "know when they don't know" in question answering, exploring methods to make confidence scores correlate better with correctness *via* finetuning and post-hoc calibration [6]. Kadavath et al., showed that large language models (up to 52B parameters) have an internal notion of uncertainty: By probing or fine-tuning, one can get the model to predict when it will likely be wrong [7].

More recently proposed calibration tuning, an instruction-tuning approach to make LLMs output calibrated confidence estimates alongside answers, demonstrating improved ECE without sacrificing accuracy [8]. Our work differs in that we do not finetune the model, but instead analyze inherent confidence signals (log-likelihoods) from a pre-trained model.

Multiple-Choice QA and Length Normalization. For LMs applied to multiple- choice tasks, a known challenge is that candidate answers may differ in length or surface form in ways unrelated to correctness. Prior evaluations (e.g., the EleutherAI LM Evaluation Harness) account for this by dividing each choice's total log-probability by its token length [9]. Highlighted that using raw language model probabilities requires heuristics such as length normalization or bias correction. They proposed an unsupervised Answer-Level Calibration (ALC) method to remove context-independent biases (including those from answer length and frequency), which improved zero-shot answer selection on commonsense QA tasks [2]. Our study complements this line of work by quantifying the effects of the simplest normalization (dividing by length) on both accuracy and calibration.

Another relevant line of research examines how instruction tuning (alignment) affects calibration. He et al., found that alignment (e.g., RLHF-trained models) can make LLMs more overconfident compared to their base models, due to conflating different uncertainty sources [10]. This emphasizes the importance of evaluating calibration for each model and use case. We focus on a single model's behavior under different scoring methods, noting that our findings could vary for aligned *vs.* base LLMs.

Overall, while many works propose methods to adjust model confidence (through tuning or external models), we concentrate on understanding the confidence signals already available from the model's token probabilities. In particular, we shed light on a fundamental question for MCQA: Given an LLM, should one trust the raw log- likelihood or normalized probability as a measure of confidence?

Methodology

In a multiple-choice QA setting, a language model processes a prompt consisting of the question q and a candidate answer option a and computes the probability of the text of that answer. For each answer option a_i (where i indexes the options), the model can assign a probability $P(a_i|q)$, typically factorized over tokens *via* the chain rule:

$$P(a_i|q) = \prod_{t=1}^{T_i} P(w_{i,f}|q, w_{i,<t}),$$

where $w_i, 1, \dots, w_i, T_i$ are the tokens in answer a_i and T_i is the length (number of tokens) of that answer. The model's log-likelihood for a_i is:

$$LL_i = \log P(a_i|\, q) = \sum_{t=1}^{T_i} log\ P\big(w_{i,t}\big|q, w_{i,<t}\big).$$

This LL_i is the unnormalized log-probability of the entire answer sequence. Due to the sum over tokens, LL_i tends to favor shorter a_i (fewer multiplications of probabilities).

A common approach to choose the most likely answer is to pick

$$\hat{\imath} = \arg \overset{max}{i} LL_i\ ,$$

the option with highest log-likelihood. However, LL_i by itself is not a calibrated confidence measure; it is not bounded (often negative and unbounded above for probability 1). Moreover, it is not comparable across different questions because it depends on the length and complexity of the answer.

The usual solution in MCQA is to derive a normalized probability by applying a softmax over the options. We can convert log-likelihoods to a probability distribution over choices for the given question:

$$P_{choice}(a_i|\, q) = \frac{\exp\,(LL_i)}{\sum_{j=1}^{N} \exp\,(LL_j)},$$

where N is the number of options. We refer to this as the softmax confidence for option i. The model's predicted answer $\hat{\imath}$ will be the same arg max under P_{choice} as under LL_i, since softmax is monotonic. Thus, using softmax confidence does not change which answer is chosen, but does provide a probabilistic confidence score between 0 and 1.

Copyright © 2025 | ijas.meteorpub.com

https://doi.org/10.70844/ijas.2025.2.29

Innovative Journal of Applied Science

Alternatively, one can normalize the log-likelihood by length to mitigate the length bias. We define the normalized log-likelihood (NormLL) as:

$$NormLL_i = \frac{1}{T_i} LL_i = \frac{1}{T_i} \sum_{t=1}^{T_i} log\ P(w_{i,t}|q, w_{i,<t}).$$

This effectively computes the average log-probability per token for the answer. Using *NormLL* for scoring means selecting

$$\hat{\imath} = \arg \max_i NormLL_i,$$

i.e., the answer that has the highest average token probability. Note that this can potentially select a different answer than raw LL_i if a longer answer had higher average probability but lower total probability.

We can also obtain a confidence score from NormLLi by exponentiating and nor-malizing:

$$P_{norm}(a_i|q) = \frac{\exp(NormLL_i)}{\sum_{j=1}^{N} \exp(NormLL_j)}.$$

This yields a pseudo-probability distribution over choices based on length-normalized scores. Like softmax confidence, P_{norm} lies in (0, 1) and sums to 1 over options (though it no longer corresponds to actual language-model probabilities without length normalization).

In summary, we study three confidence scoring methods:

1. **LL (Log-Likelihood):** Use LL_i to choose the answer (equivalent to softmax choice) and interpret confidence as $P_{choice}(a_i \mid q)$. In practice, this yields the same prediction as softmax.
2. **NormLL (Normalized Log-Likelihood):** Use $NormLL_i$ to choose the answer and use $P_{norm}(a_i|q)$ as the confidence.
3. **Softmax (Choice Probability):** The standard approach: use LL_i for choice and $P_{choice}(a_i \mid q)$ for confidence.

We evaluate each method in terms of:

- **Accuracy:** The percentage of questions where the model's chosen answer $\hat{\imath}$ is correct.
- **Calibration:** How well the confidence estimates align with reality. We measure this *via* Expected Calibration Error (ECE) [3]. ECE partitions predictions into bins by confidence and computes the difference between mean confidence and accuracy in each bin, averaging these differences weighted by bin size. A perfectly calibrated model would have, for example, 70% accuracy among all predictions to which it assigned 70% confidence. We report ECE as a percentage (lower is better).

Additionally, we use bootstrap resampling to estimate the uncertainty (confidence intervals) for accuracy and ECE on each dataset. This allows us to assess if differences between methods are statistically significant.

Experimental Setup

Model. We use the pre-trained Meta-Llama-3.1-8B large language model to score answer options. Specifically, we query the 8B-parameter LLaMA-3.1 model in a zero-shot setting *via* a completion API, extracting token log-probabilities. The model is treated as a black box scoring function that provides $P(w_t \mid \text{context})$ for each next token. We ensured that the model's prompting format is fixed across all methods; each answer is scored independently using the same prompt template to avoid any ordering bias. The model is used as-is without fine-tuning on the target datasets, so its calibration reflects that of the original pre-trained model. While Meta-Llama-3.1-8B may include some instruction tuning, we treat it as a standard language model without explicit alignment or additional calibration.

Datasets

We evaluate on four multiple-choice QA benchmarks:

- **ARC-Easy and ARC-Challenge:** These are subsets of the AI2 Reasoning Challenge (ARC) dataset of grade-school science questions [11]. Each question has four answer options. ARC-Easy contains questions answerable by retrieval of a single fact, whereas ARC-Challenge contains harder questions requiring reasoning or combining facts. We use the official test sets: 2,376 questions in ARC-Easy and 1,172 in ARC-Challenge.
- **BoolQ:** A yes/no QA dataset with 3,270 questions, each with a relevant passage. We convert it to a binary-choice format (options" Yes" and" No"). This task has only two options, which are single-word answers [12].
- **HellaSwag:** A commonsense reasoning dataset of 10,042 instances. Each instance provides a brief scenario and four possible continuations; the task is to choose the most plausible continuation. HellaSwag is known for adversarial wrong answers that are superficially plausible and often longer or more detailed than the correct answer [13].

These datasets cover a range of domains (science exams, open-domain web passages, video descriptions) and answer lengths (BoolQ answers are very short. HellaSwag answers are full sentences). This diversity lets us examine how length normalization behaves in different scenarios.

Procedure: For each question in each dataset, we obtain LL_i for each answer option by prompting the model and summing token log-probabilities. We then compute the predicted answer and confidence under each of the three methods (LL, NormLL, Softmax) described in the Methodology. Accuracy is computed by comparing the predicted answer to the ground truth. For confidence calibration, we gather all predictions and their confidences to compute ECE (using 10 confidence bins, a common choice following [4]. We also calculate ECE per dataset.

We perform 1000-sample bootstrap resampling on each dataset to compute 95% confidence intervals for accuracy and ECE under each method. This involves sampling questions with replacement and recalculating metrics, which helps determine if differences observed are likely due to chance or represent real performance gaps.

Evaluation Metrics: Our primary metrics are accuracy and ECE as defined above. Additionally, we record auxiliary statistics such as the average confidence of correct *vs.* incorrect predictions to assess if the model tends to be more or less confident on its mistakes. We also examine cases where NormLL and LL disagree on which option is best, to understand the nature of those questions (e.g., are the answers of different lengths?).

Copyright © 2025 | ijas.meteorpub.com

https://doi.org/10.70844/ijas.2025.2.29

Innovative Journal of Applied Science

Results

We first present overall accuracy and calibration results for each method on each dataset, then delve into analysis of the differences (Table 1).

Dataset	Method	Accuracy (%)	Δ Acc.	ECE (%)	Δ ECE
ARC-Easy	Log-Likelihood	69.36	–	1.86	–
	Norm Log-Likelihood	53.7	-15.66	15.05	13.19
	Softmax (Prob)	69.36	0	1.86	0
ARC-Challenge	Log-Likelihood	54.44	–	9.92	–
	Norm Log-Likelihood	46.76	-7.68	14.23	4.31
	Softmax (Prob)	54.44	0	9.92	0
BoolQ	Log-Likelihood	54.83	–	4.57	–
	Norm Log-Likelihood	54.83	0	4.57	0
	Softmax (Prob)	54.83	0	4.57	0
HellaSwag	Log-Likelihood	28.92	–	36.99	–
	Norm Log-Likelihood	27.04	-1.88	4.35	-32.64
	Softmax (Prob)	28.92	0	36.99	0

Table 1: Accuracy and Expected Calibration Error (ECE) of a large language model on four MCQA datasets, using different confidence scoring methods. Bold values indicate the best result for a given dataset. Δ columns reflect differences in accuracy and ECE relative to the Log-Likelihood baseline.

Table 1 summarizes the accuracy and ECE for each dataset-method combination. The softmax and raw log-likelihood methods have identical accuracy in all cases, as expected (since they select the same answer). NormLL sometimes chooses a different answer, leading to differences in accuracy:

- On ARC-Easy and ARC-Challenge, NormLL substantially underperforms raw LL/softmax. Accuracy drops by about 15.7 points on ARC-Easy (from 69.36% to 53.70%) and by 7.7 points on ARC-Challenge (54.44% to 46.76%). These drops are statistically significant (95% confidence intervals do not overlap).
- On BoolQ, all methods yield the same accuracy (54.8%). This implies that, for this dataset, for every question, the answer with the highest total likelihood is also the one with the highest average per-token likelihood. Given BoolQ's yes/no answers are one token each, length normalization has no effect.
- On HellaSwag, NormLL accuracy is slightly lower than raw (27.04% *vs* 28.92%), a drop of 1.9 points. While small in absolute terms, this difference reflects a few changed predictions.

In terms of calibration:

- For ARC-Easy, the model was very well-calibrated under softmax/LL (ECE ≈ 1.86%). NormLL yields a much higher ECE (15.05%), indicating that normalizing by length severely mis calibrated confidence on ARC-Easy. With NormLL the model became underconfident on many easy questions it answered correctly (assigning lower probability than it should have).
- For ARC-Challenge, softmax ECE is 9.92% and NormLL increases it to 14.23%. NormLL also worsened calibration on this dataset, though the effect is smaller than on ARC-Easy.
- For BoolQ, ECE is 4.57% for all methods (since predictions and confidences were identical).
- For HellaSwag, we see a dramatic improvement in ECE with NormLL: From a very high 36.99% under softmax to only 4.35% with NormLL. The raw model was extremely overconfident on HellaSwag (often confidently choosing wrong answers), but normalizing the log-probs by length reduced this overconfidence to near zero. We analyze this case below.

Bootstrap confidence intervals confirm these differences. On ARC-Easy, the raw/softmax accuracy 95% CI (around 67.5–71.1%) *vs* NormLL (51.7–55.8%) show no overlap.

ARC-Challenge raw (51.5–57.3%) *vs* NormLL (43.8–49.5%) also do not overlap. On HellaSwag, the accuracy CIs for raw *vs* NormLL overlap (roughly 27.8–30.0 *vs* 26.2–28.7), indicating the accuracy difference there might not be statistically robust. However, the ECE improvements are clear: HellaSwag NormLL ECE 4.35% (CI ≈3.6–5.1) *vs* raw 36.99% (CI ≈35.0–38.0). Likewise, ARC-Easy NormLL ECE 15.05% *vs* raw 1.86% (no overlap).

Analysis of length normalization effects

The above results beg the question: Why does length normalization help so much on HellaSwag but hurt on ARC? We examine these datasets and the model's behavior:

HellaSwag: Overconfidence on short distractors: In HellaSwag, the model's raw softmax confidence for its chosen answer was frequently very high (e.g., > 90%) even when it was wrong. Often, the wrong answer the model preferred was notice- ably shorter or simpler than the correct answer. By using NormLL, the longer correct answer's average token probability can compete more fairly with the short answer. In essence, NormLL prevented the model from being too biased toward short, high- probability sequences, thus reducing confidence on those wrong answers. This yielded a near elimination of calibration error. Notably, NormLL didn't drastically increase accuracy on HellaSwag; it mainly made the model less sure about its (often wrong) choices, aligning confidence with its low accuracy (29%). This is reflected in ECE dropping to 4%, meaning when the model says "I'm 30% confident", it is correct 30% of the time, which was not the case with raw scoring.

ARC: Correct answers are short: In ARC, the correct answers (especially in ARC-Easy) are often shorter or simpler than the distractors. Many easy science questions have single-word correct answers (e.g., "photosynthesis"), whereas incorrect options might be longer phrases or sentences. The model with raw scoring was already doing well at picking the correct short answer and was relatively well-calibrated. Applying NormLL penalized those correct short answers (removing their length advantage) and in some cases caused a longer, wrong option to score higher. This explains the accuracy drop. In terms of calibration, NormLL made the model hesitate more on

Copyright © 2025 | ijas.meteorpub.com

questions it actually knew, increasing ECE by introducing under confidence. Essentially, length normalization undermined a useful heuristic that aided model performance on ARC-Easy (favoring concise correct answers) that the raw scoring exploited. For ARC-Challenge, the effect was similar but less extreme. Correct answers did not consistently have a length advantage, so NormLL misled the model on some questions without a clear calibration benefit (ECE slightly worsened).

When Do NormLL and Raw Disagree? We inspected cases where NormLL picked a different answer than raw (and softmax). On ARC-Easy/Challenge, these tended to be instances where: The raw top answer was shorter than the NormLL top answer. If the raw top answer was correct and NormLL top was wrong (common on ARC), length norm hurt performance. Conversely, a few cases had the raw top answer longer and wrong and NormLL switched to a shorter correct answer (helping accuracy), but these were rarer.

On HellaSwag, disagreements often involved the correct answer being longer. NormLL sometimes switched to the correct answer if raw had picked a shorter wrong one. It also occasionally picked a different wrong answer (if that wrong option had higher average probability, even though raw had chosen another wrong one by total probability). The net effect was a mix of wins and losses for NormLL in accuracy, but a consistent reduction in confidence for wrong answers, improving calibration.

Discussion

Our findings demonstrate that simple log-likelihood, as a direct output of a language model, carries valuable information about model confidence, but its proper use is nuanced. The raw log-likelihood (when transformed into a probability *via* softmax over options) gave us a baseline calibration of the model. In some cases (ARC-Easy), this baseline was surprisingly good (ECE ≈ 2%). In others (HellaSwag), the baseline was poor, showing the model was grossly overconfident.

Length normalization is a double-edged sword. It addresses one specific issue (length bias) which clearly mattered in HellaSwag, aligning confidence better. But it can interfere with the model's natural scoring in cases where length correlates with correctness (like concise answers in science questions). This suggests that one-size-fits-all normalization might not be optimal. An adaptive approach could be needed: e.g., apply length normalization only when answer lengths vary greatly or when the model is exploiting superficial cues.

Our results also resonate with prior work that introduced more sophisticated calibration techniques. The ALC method by Kumar, et al., for example, can be seen as an extension that not only deals with length but also other biases [2]. Our simpler analysis confirms that even the most basic bias (length) can swing calibration and accuracy significantly.

Notably, raw log-likelihood and softmax probability yielded identical ECE in our evaluations (except when NormLL changed the predictions). This is because we computed ECE after converting raw LL to probabilities *via* softmax, so raw LL and softmax are effectively the same confidence measure in practice. If one were to treat the raw log-likelihood values directly as a confidence score (without conversion to a probability), additional calibration or mapping would be needed.

We used Meta-Llama-3.1-8B, a large instruction-tuned decoder-only language model. Larger models can be better calibrated in some settings than smaller ones, but instruction- tuned models may become less calibrated [10]. If our model was an aligned one, the observed overconfidence on HellaSwag might be partly due to alignment. For a base model or different architecture, the numbers might differ, though the qualitative effects of length likely remain.

Finally, while we focused on multiple-choice format (which confines the probability space to given options), the ultimate goal is calibrating open-ended generation. In open-ended settings, the space of possible answers is vast and length normalization or softmax over options isn't directly applicable. Recent works like have started training models to produce calibrated confidence statements for their out- puts [8]. Our study provides insight into the simplest form of model-internal confidence for a constrained task, as a stepping stone toward understanding confidence in free-form generation.

Conclusion

We presented a comprehensive evaluation of using log-likelihood as a confidence signal for large language models on multiple-choice QA tasks. By comparing raw log-likelihood, length-normalized log-likelihood and the standard softmax-based confidence, we showed that the choice of scoring method can significantly impact both prediction accuracy and confidence calibration. Raw log-likelihood (when appropriately converted to a probability) generally mirrors the softmax method, yielding good performance on tasks where the model is already calibrated. Length normalization can either greatly improve calibration (as seen on HellaSwag) or hurt performance (as on ARC), depending on dataset characteristics.

No single confidence scoring approach is universally best for all scenarios. Practitioners should be aware of potential length biases in model scoring and consider calibration metrics like ECE when deciding how to interpret model confidences. In cases where calibration is critical, a slight loss in accuracy from normalization might be acceptable to obtain reliable probabilities. Conversely, if accuracy is paramount and the model is known to be well-calibrated or if correct answers tend to be short, raw scoring may be preferred.

Future work could explore dynamic strategies that detect when to apply length normalization or other calibration adjustments on a per-question or per-dataset basis. Extending these confidence measures to open-ended generation is another avenue, as quantifying uncertainty in free-form outputs remains challenging. Ultimately, improving LLMs' ability to know when they might be wrong is key to building trustworthy AI systems.

Limitations

This study has several limitations. First, we evaluated only a single instruction-tuned language model (Meta-Llama-3.1-8B). While representative of modern LLMs, our findings may not generalize to other architectures, model sizes, or alignment levels. For example, base models or task-specific fine-tuned models may exhibit different confidence behaviors or length biases.

Second, we focused exclusively on multiple-choice formats, which allow for clear application of log-likelihood and softmax-based confidence. In open-ended or generative QA, these methods are less applicable and adapting them requires further methodological development.

Copyright © 2025 | ijas.meteorpub.com

Third, our analysis primarily used Expected Calibration Error (ECE) to assess confidence alignment. While widely used, ECE overlooks nuances such as confidence distributions among incorrect predictions or overconfidence in specific answer types. Alternative metrics (e.g., Brier score) or post-hoc techniques like temperature scaling were not explored but could improve calibration.

Additionally, we treated the model as a black box and did not probe internal representations or attention patterns. More advanced methods—such as inspecting hidden states or using verifier models—might yield richer confidence signals.

We also did not systematically quantify dataset-level properties such as answer length distributions, distractor quality, or prompt format sensitivity. While our qualitative analysis suggested these factors play a role (e.g., in ARC or HellaSwag), a more rigorous characterization could strengthen these conclusions.

Finally, prompt design was held constant, but its interaction with confidence and calibration was not studied. Subtle changes in phrasing, formatting, or option ordering may influence both predictions and confidence scores.

Despite these limitations, our study provides actionable empirical insights into confidence estimation in LLMs and underscores the need for dataset-aware calibration strategies when deploying these models in QA settings.

Conflict of interest

The author declares no conflict of interest.

References

1. Kapoor S, Gruver N, Roberts M, Pal A, Dooley S, et al. (2024) Calibration-tuning: Teaching large language models to know what they don't know. InProceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024) 1-14. [GoogleScholar]
2. Kumar S (2022) Answer-level calibration for free-form multiple choice question answering. InProceedings of the 60th Annual Meeting of the Association for Computational Linguistics 1: 665-679. [Crossref] [GoogleScholar]
3. Pavlovic M (2025) Understanding model calibration – a gentle introduction and visual exploration of calibration and the expected calibration error (ECE). arXiv Preprint. [Crossref] [GoogleScholar]
4. Guo C, Pleiss G, Sun Y, Weinberger (2017) KQ On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning (ICML). [GoogleScholar]
5. Desai S, Durrett G (2020) Calibration of pre-trained transformers. In Proceedings of EMNLP. [Crossref] [GoogleScholar]
6. Jiang Z, Xia P, Wang X, Eisner J (2021) How can we know when language models know? on the calibration of language models for question answering. Transactions of the ACL (TACL) 9: 962-977. [Crossref] [GoogleScholar]
7. Kadavath B, Askell A, Mishkin P, Henighan T, Drain D, et al. Language models (mostly) know what they know. arXiv preprint. [Crossref] [GoogleScholar]
8. Mielke S, Heinisch P, Strobelt H, et al. (2024) Calibration-tuning: Teaching large language models to know what they don't know. In Proceedings of the 1st Workshop on Uncertainty in NLP (UNCERTAIN), 2024.
9. Cho G, So Y, Lee J (2025) Anpmi: Assessing the true comprehension capabilities of llms for multiple choice questions. arXiv preprint. [Crossref] [GoogleScholar]
10. He G, Cui P, Chen J, Hu W, Zhu J (2024) Investigating uncertainty calibration of aligned language models under the multiple-choice setting. Open Review. [Crossref] [GoogleScholar]
11. Clark P, Cowhey I, Etzioni O, Khot T, Sabharwal A, et al. (2018) Think you have solved question answering? try arc, the AI2 reasoning challenge. arXiv preprint. [Crossref] [GoogleScholar]
12. Clark C, Lee K, Chang MW, Kwiatkowski T, Collins M, et al. (2019) Boolq: Exploring the surprising difficulty of natural yes/no questions. In Proceedings of NAACL. [Crossref] [GoogleScholar]
13. Zellers R, Holtzman A, Bisk Y, Farhadi A, Choi Y (2019) HellaSwag: Can a machine really finish your sentence?. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL). [Crossref] [GoogleScholar]

Copyright © 2025 | ijas.meteorpub.com