Mind the Gap:␣A Closer Look at Tokenization for Multiple-Choice Question Answering with LLMs

Mario Sanz-Guerrero[▽] **Minh Duc Bui**[▽] **Katharina von der Wense**[▽♠]
[▽]Johannes Gutenberg University Mainz, Germany
[♠]University of Colorado Boulder, USA
{msanz, minhducbui, k.vonderwense}@uni-mainz.de

Abstract

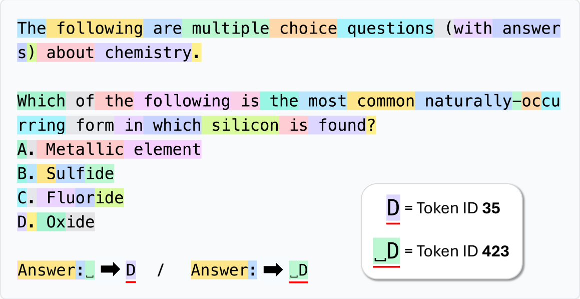
When evaluating large language models (LLMs) with multiple-choice question answering (MCQA), it is common to end the prompt with the string "Answer:" to facilitate automated answer extraction via next-token probabilities. However, there is no consensus on how to tokenize the space following the colon, often overlooked as a trivial choice. In this paper, we uncover accuracy differences of up to 11% due to this (seemingly irrelevant) tokenization variation as well as reshuffled model rankings, raising concerns about the reliability of LLM comparisons in prior work. Surprisingly, we are able to recommend one specific strategy – tokenizing the space *together* with the answer letter – as we observe consistent and statistically significant performance improvements. Additionally, it improves model calibration, enhancing the reliability of the model's confidence estimates. Our findings underscore the importance of careful evaluation design and highlight the need for standardized, transparent evaluation protocols to ensure reliable and comparable results.

1 Introduction

Leaderboards for evaluating large language models (LLMs) often include multiple-choice question answering (MCQA) tasks: the model is shown a question together with several candidate answers and must pick the correct one. To make automatic answer extraction easier, a widely used convention is to end the prompt with the literal string "Answer:" and look at the next-token probabilities for the option labels (usually: A/B/C/D). This seemingly trivial formatting decision immediately poses another: *should there be a space after the colon in the prompt, or should the space be included as part of the answer option token?*

Recent studies have highlighted the significant performance variation that can arise from minor changes in prompt design (Zheng et al., 2024; Pezeshkpour and Hruschka, 2024). However, little attention has been given to the role of tokenization, particularly the tokenization of the empty space character immediately preceding the answer label – after the string "Answer:␣" (see Figure 1). More importantly, we note that *practice is currently split*: some recent papers include the leading space in the prompt (Santurkar et al., 2023; Wang et al., 2024a,b), while others omit it and tokenize it together with the letter (Zheng et al., 2024; Hendrycks et al., 2021), and no community-wide convention has emerged. Even widely used evaluation frameworks differ in their convention (Habib et al., 2023; Gao et al., 2024).

Surprisingly, we find significant differences in performance depending on the choice of the leading space tokenization. When the **leading space is tokenized together with the label letter, we observe consistent, statistically significant gains in both accuracy and calibration** across a wide range of LLMs and datasets. This seemingly irrelevant tokenization choice alone shifts accuracy by as much as 11% – a larger effect than previously observed prompt formatting variations such as option order permutation. Moreover, we find that **the choice of tokenization convention even**



```
The following are multiple choice questions (with answer
s) about chemistry.

Which of the following is the most common naturally-occu
rring form in which silicon is found?
A. Metallic element
B. Sulfide
C. Fluoride
D. Oxide

Answer:␣ ➡ D  /  Answer: ➡ ␣D
```

Figure 1: Illustration of a prompt tokenized with Llama 3.1. The final token representing the prediction depends on the tokenization of the space ("D", without space; or "␣D", with space).

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 19573–19583
November 4-9, 2025 ©2025 Association for Computational Linguistics

alters model rankings. When the space is tokenized before the letter, Llama 3.1 70B Instruct tops our leaderboard; when the space is tokenized together with the letter, Qwen 2.5 72B moves into first place.

Our experiments result in a clear recommendation: tokenize the space together with the letter, and observe model rankings exclusively for this configuration to ensure fair comparisons. More generally, these findings underscore the need for unified evaluation frameworks and greater transparency, particularly for closed-source models, so LLM comparisons remain fair and meaningful.

2 Related Work

LLM Evaluation with MCQA Evaluating generative LLMs presents a significant challenge due to the open-ended nature of their outputs. Recent approaches have explored human evaluation and LLM-as-a-judge methods (Chiang and Lee, 2023; Chen et al., 2024), but these techniques are highly subjective and unreliable. To address this, multiple-choice question answering has been widely adopted, as it enables automated, quantitative assessment of LLM capabilities.

There are multiple ways of automatically extracting an LLM's answer in MCQA tasks. Previous work have shown that better performance can be achieved by allowing the model to generate a free-form answer, followed by using a secondary LLM to extract the final choice (Wang et al., 2024b; Lyu et al., 2024). However, this approach is computationally expensive and can yield inconsistent results across different secondary models. Given that answer options are identified by letters (or labels), one of the most commonly used methodologies is to compute the model probabilities for the next token and get the highest label as the predicted choice (Hendrycks et al., 2021; Santurkar et al., 2023).

Sensitivity to Prompt Details Recent studies have demonstrated that LLM performance in MCQA is highly sensitive to prompt details, often showing biases toward certain labels and answer order (Pezeshkpour and Hruschka, 2024; Zheng et al., 2024; Alzahrani et al., 2024). However, little attention has been given to the tokenization of the space character preceding the answer label, and there are discrepancies in the literature. Some studies tokenize this space as an individual token (e.g., Santurkar et al., 2023; Wang et al., 2024a,b; Pal and Sankarasubbu, 2024), while others tokenize it together with the answer label (e.g., Zheng et al., 2024; Hendrycks et al., 2021).

Complementarily to this body of research, we focus on a largely overlooked and *apparently* irrelevant factor: the tokenization of the space character immediately preceding the answer label.

3 Space or No Space?

The literature has yet to converge on a single convention: even widely used evaluation frameworks such as Lighteval (Habib et al., 2023) from Hugging Face exhibit inconsistencies in how the leading space in the prompt is tokenized across different datasets. We begin by presenting the key arguments supporting each approach and identifying the prior works that have adopted them.

General Setting A MCQA prompt consists of a question and a set of answer choices, each associated with a distinct letter label as in Figure 1. It finishes with the string "`Answer:`", and the LLM prediction is obtained as $\hat{t} = \arg\max_{t \in \{\mathrm{A,B,...}\}} P(t \mid S)$, i.e., selecting the choice whose label token t has the highest next-token conditional probability on the prompt S. This allows for an efficient and automated extraction of model answers for performance assessment.

Letter Token Without Space Given that the t label tokens presented in the list of options in the prompt are tokenized as a single letter (without the leading space; see Figure 1), it seems plausible to analyze the probability of the next token as a single letter as well (i.e., tokenizing the leading space as "`Answer:␣`", *before* the actual token of the letter label). This tokenization represents the exact same token as the one in the corresponding option in the prompt (more details in Appendix A), and there is a body of research that tokenizes this way (Santurkar et al., 2023; Wang et al., 2024a,b; Pal and Sankarasubbu, 2024).

Letter Token With Space However, the previous is not the default tokenization: if we include the final answer letter in the prompt and tokenize it, the last token would be "␣t". Thus, tokenizing the space *together* with the letter also seems a reasonable approach. This convention is also used in prior work (Zheng et al., 2024; Hendrycks et al., 2021).

4 Experimental Setup

The goal of our experiments is to analyze potential differences when tokenizing the option labels as single letters versus as letters preceded by a space (i.e., getting the model predictions from the "X"[1] or "␣X" tokens, respectively). We look at it from two different perspectives: (1) performance, where we evaluate how accurate the model is in its predictions; and (2) calibration, where we assess how reliable the model predictions are.

Datasets Our main experiments are conducted on MMLU (Hendrycks et al., 2021), one of the most widely used benchmarks for LLM evaluation (OpenAI et al., 2024; Grattafiori et al., 2024). MMLU contains multiple-choice questions from 57 different fields, providing a comprehensive set for interdisciplinary knowledge assessment. To ensure our findings are not specific to a single benchmark, we additionally evaluate on five other commonly used MCQA datasets (listed in Appendix B.1).

Models We evaluate 15 LLMs from various families, sizes, and capabilities (listed in Appendix B.2). All models are run with random sampling disabled (i.e., greedy decoding) for deterministic outputs and reproducibility.

Prompts To ensure our findings are robust and not limited to a single prompt template, we experiment with a variety of prompt formulations. These include zero-shot and few-shot settings, chain-of-thought (CoT) prompting, alternative formats for multiple-choice options, and prompts in different languages. Further details are provided in Appendix B.3.

Evaluation For measuring performance, we report accuracy. As for calibration, we report the expected calibration error (ECE; Pakdaman Naeini et al., 2015), which measures the weighted average discrepancy between a model's prediction confidence and its actual accuracy across confidence bins. The formula of ECE is as follows:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left| \underbrace{\frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}\{\hat{y}_i = y_i\}}_{\mathrm{acc}(B_m)} - \underbrace{\frac{1}{|B_m|} \sum_{i \in B_m} p_i}_{\mathrm{conf}(B_m)} \right|,$$

where M is the number of confidence bins, N is the total number of instances, B_m is the set of instances whose predicted confidence falls into bin m, $\hat{y}_i$ and y_i are the predicted and true labels for instance i, p_i is the model's confidence for its predicted label on instance i, $\mathrm{acc}(B_m)$ is the empirical accuracy in bin m, and $\mathrm{conf}(B_m)$ is the average confidence in bin m. In our experiments, $M = 10$ bins (i.e., we have 10 bins, comprising 10% accuracy each).

Statistical Significance To assess whether the choice of space tokenization strategy leads to statistically meaningful differences in performance, we conduct statistical tests comparing results from both setups. For accuracy, we use McNemar's test (McNemar, 1947) and, for calibration, we apply a paired bootstrap resampling test on the ECE (more details in Appendix B.4). In both cases, we consider differences significant when $p < 0.05$.

Probability Extraction To obtain the model's predicted probabilities, we pass the prompt through the LLM and extract the next-token logits of the options letters. The logits are then converted into normalized probabilities, producing a probability distribution over all possible answers. This allows us to analyze not only the most likely answer but also how the model distributes probability mass (confidence) across all options, which is important for evaluating calibration.

5 Results

We evaluate our two tokenization schemes, which we denote: (1) **Letter token** (i.e., last line of the prompt is tokenized as `["Answer",":","␣","X"]`); and (2) **Space–Letter token** (i.e., last line of the prompt is tokenized as `["Answer",":","␣X"]`). We analyze the probability of the tokens underlined in red. Table 1 shows the zero-shot results for all models on the MMLU dataset.

Performance While the place where the space preceding the answer letter is tokenized might seem completely irrelevant, we observe noteworthy **accuracy gains in all models when tokenizing the space *within the same token* as the actual letter**. These improvements are statistically significant (except for Gemma 3 12B and Mistral 7B). This might seem counterintuitive, as these tokens are not the same as the ones in the options list of the prompt.

These performance differences have crucial practical implications: even by evaluating only a handful of models, we show noticeable changes in a hypothetical leaderboard. **By only changing where the space is tokenized, the top-performing model**

[1] Where "X" is one of the option letters (A, B, C, D).

Model	Accuracy (↑)		ECE (↓)	
	“X”	“␣X”	“X”	“␣X”
Llama 2 7B	37.25	**38.88***	2.16	**1.15***
Llama 3.1 8B	61.47	**63.93***	2.58	**0.50***
Llama 3.1 8B Inst	67.28	**68.73***	4.19	**3.77***
Llama 3.1 70B	76.16	**76.64***	1.47	**1.16***
Llama 3.1 70B Inst	82.31	**82.60***	**3.91**	4.87
Gemma 3 4B	56.25	**57.95***	7.40	**1.74***
Gemma 3 4B Inst	57.43	**57.77***	**20.34**	20.36
Gemma 3 12B	71.17	**71.31**	2.18	**0.91***
Mistral 7B v0.3	60.17	**60.28**	1.40	**0.51***
Mistral 7B Inst v0.3	59.70	**60.05***	12.83	**11.98***
Mistral Small 24B	77.28	**77.66***	0.79	**0.74**
Qwen 2.5 7B	69.38	**70.99***	**2.88**	3.05
Qwen 2.5 72B	81.93	**83.24***	1.10	**0.72**
Qwen 3 8B	72.82	**74.62***	2.95	**1.95***
GPT Neo 2.7B	23.65	**24.39***	12.00	**4.05***

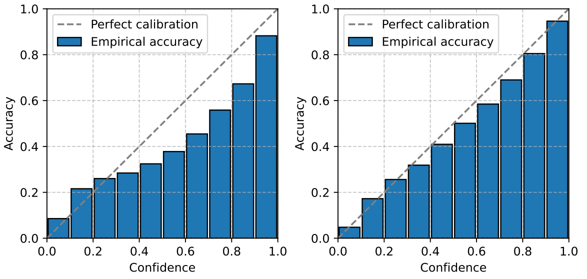
Table 1: Zero-shot performance on MMLU when tokenizing the answer letter as either a single letter (“X”) or as a space plus letter (“␣X”). * indicates a statistically significant improvement ($p < 0.05$). The top-performing model for each tokenization is underlined, and the top-performing tokenization strategy for each model is bolded.

changes (from Llama 3.1 70B Instruct to Qwen 2.5 72B). This indicates that such a subtle tweak could significantly alter LLM leaderboards.

Calibration Additionally, the ECE is lower for the large majority of models when tokenizing the space with the letter, with many of the differences being significant under the paired bootstrap resampling test. We find **model answers being up to 4 times more reliable by only changing the tokenization of the leading space** (see Gemma 3 4B). For illustrating the calibration of the models, Figure 2 shows the reliability diagrams for the Gemma 3 model. The main calibration gains come from the 30% confidence bin onwards – which, after tweaking the tokenization, become closer to the perfect calibration. These improvements are crucial for enhanced performance, as even minor changes near the model's decision boundary can greatly affect predictions.

5.1 Few-Shot and Chain-of-Thought Results

Few-Shot In the few-shot scenario, we include 5 example questions and answers in the prompt before the target question, using the same tokenization for every answer as in the final (evaluated) token. This approach is widely used to help models better understand the task format and expected output (OpenAI et al., 2024; Grattafiori et al., 2024).



(a) Letter token (“X”). (b) Space–Letter token (“␣X”).

Figure 2: Reliability diagrams for Gemma 3 4B.

Setting	Accuracy (↑)		ECE (↓)	
	“X”	“␣X”	“X”	“␣X”
Zero-shot	61.47	**63.93***	2.58	**0.50***
Few-shot	63.90	**65.78***	2.24	**0.37***
Chain-of-Thought	69.64	**70.11**	6.36	**3.75***

Table 2: Performance of Llama 3.1 8B on MMLU under different prompt settings.

The results in Table 2 show that the accuracy and calibration improvements from space–letter tokenization persist in the few-shot setting, confirming that the impact of space tokenization is robust even when the model is provided with explicit demonstrations of the answer format.

Chain-of-Thought We further test the effect of space tokenization under CoT prompting, where the model is encouraged to reason step-by-step before providing its answer. Table 2 shows that, while calibration still improves significantly, the absolute accuracy gains are not – this is reasonable since, after the reasoning chain, extracting the answer label is more straightforward and thus less sensitive to the empty space tokenization.

5.2 Robustness to Prompt Variations

Recent work has demonstrated that LLMs are highly sensitive to subtle changes in prompt phrasing and structure. To evaluate the robustness of our findings, we experiment with a range of prompt formulations (see Appendix B.3.2 for details on these perturbations). Table 3 shows that the impact of empty space tokenization is consistent across all prompt variations and in fact exceeds the effects of other prompt modifications such as changing option labels or their order.[2] Furthermore, Appendix C.2 presents results for MMLU in five dif-

[2] Numeric labels yield identical results because “␣n” is tokenized as [“␣”, “n”], so the final token is the same for both tokenization strategies.

Prompt Template	Accuracy (↑) “X”	Accuracy (↑) “␣X”	ECE (↓) “X”	ECE (↓) “␣X”
Original	61.47	**63.93***	2.58	**0.50***
Parentheses (“␣(A)”)	62.07	**64.18***	1.92	**1.07***
Numbers (“␣1”)	62.21	62.21	1.94	1.94
Space in option list	61.89	**63.25***	1.86	**0.74***
Choices before question	44.52	**48.02***	6.82	**2.74***
Permutations (avg.)	61.53	**63.37***	2.98	**0.56***

Table 3: Performance of Llama 3.1 8B on the MMLU benchmark with different prompt templates.

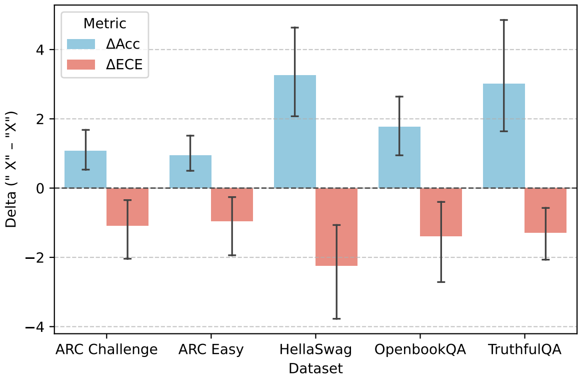


Figure 3: Mean accuracy improvement (left) and ECE reduction (right) from tokenizing the space with the answer letter, averaged across all models for each dataset. Error bars represent 95% confidence intervals.

ferent languages – including some not natively supported by the models – and our findings remain robust in all cases.

5.3 Results on Other Datasets

We further validate the generality of the effect by evaluating all models on five additional widely used MCQA datasets (ARC Challenge (Clark et al., 2018), ARC Easy (Clark et al., 2018), HellaSwag (Zellers et al., 2019), OpenbookQA (Mihaylov et al., 2018), and TruthfulQA (Lin et al., 2022)). Figure 3 summarizes the average accuracy improvement and ECE reduction (delta aggregated across models) for each dataset, while the full per-model results are reported in Appendix C.3. The trends are consistent with the previous findings: tokenizing the space together with the answer letter systematically increases accuracy while (in most cases) lowering calibration error. Notably, even the largest model in our study (Qwen 2.5 72B) exhibits a very substantial accuracy gain of over 11% on the HellaSwag dataset under the space–letter tokenization (“␣X”), underscoring that the effect is not confined to smaller or less capable models.

6 Conclusion

In this work, we uncover a subtle yet impactful detail in the MCQA evaluation of LLMs: the tokenization of the space preceding the answer letter. Despite the lack of a standardized convention for this tokenization – often dismissed as an irrelevant choice – we show that it has significant implications for both model performance and reliability. Our experiments reveal that tokenizing the space *together with the option letter* leads to consistent improvements in accuracy and calibration, with performance gains reaching up to 11%. More strikingly, this minor tokenization change is sufficient to alter the relative rankings of models on leaderboards, raising important concerns about the comparability of prior LLM evaluation results. We encourage future work to consider these low-level details carefully to ensure fair and meaningful model comparisons.

Limitations

Our evaluation focuses on open-weight models, as we require access to all next-token logits, which are not provided for proprietary, API-based models. To allow for extensive experimentation with different models under our computational constraints, we use small- to medium-sized LLMs (up to 72B) and observe similar trends across all of them. Testing our findings with large-scale LLMs remains for future work.

Ethics Statement

Our work highlights a noteworthy discrepancy in the current literature on LLM evaluation with MCQA and demonstrates significant performance improvements by tokenizing the empty space together with the subsequent answer letter token. However, these improvements do not fully eliminate inherent risks, and LLMs remain susceptible to errors. Therefore, we caution against relying solely on LLMs in critical settings, such as medical advice, without appropriate human oversight and domain-specific validation.

Acknowledgments

This work was supported by the Carl Zeiss Foundation through the MAINCE and TOPML projects (grant numbers P2022-08-009 and P2021-02-014).

References

Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, Bangkok, Thailand. Association for Computational Linguistics.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA. Association for Computational Linguistics.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. 2025. OLMES: A standard for language model evaluations. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5005–5033, Albuquerque, New Mexico. Association for Computational Linguistics.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. Lighteval: A lightweight framework for llm evaluation.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

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. *Preprint*, arXiv:2310.06825.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Chenyang Lyu, Minghao Wu, and Alham Aji. 2024. Beyond probabilities: Unveiling the misalignment in evaluating large language models. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 109–131, Bangkok, Thailand. Association for Computational Linguistics.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Mistral AI. 2025. Mistral Small 3 | Mistral AI.

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, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Ankit Pal and Malaikannan Sankarasubbu. 2024. Gemini goes to Med school: Exploring the capabilities of multimodal large language models on medical challenge problems & hallucinations. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 21–46, Mexico City, Mexico. Association for Computational Linguistics.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017, Mexico City, Mexico. Association for Computational Linguistics.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29971–30004. PMLR.

Mario Sanz-Guerrero and Katharina von der Wense. 2025. Corrective in-context learning: Evaluating self-correction in large language models. In *The Sixth Workshop on Insights from Negative Results in NLP*, pages 24–33, Albuquerque, New Mexico. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Xinpeng Wang, Chengzhi Hu, Bolei Ma, Paul Rottger, and Barbara Plank. 2024a. Look at the text: Instruction-tuned language models are more robust multiple choice selectors than you think. In *First Conference on Language Modeling*.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024b. "My answer is C": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7407–7416, Bangkok, Thailand. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

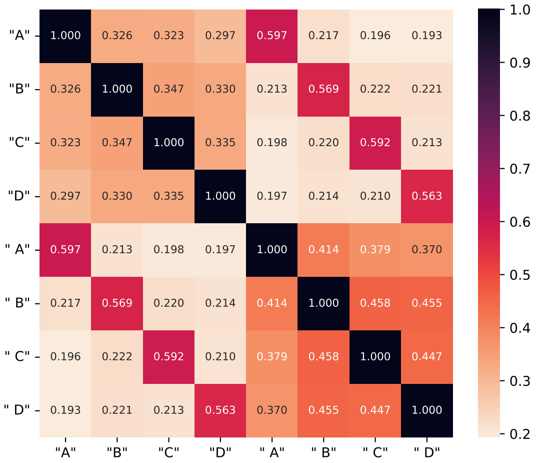
A Space or No Space? Detailed Motivation of Both Approaches

In Section 3 we discuss the two current approaches in the literature for the tokenization of the empty space preceding the answer letter. Here, we provide a more in-depth analysis of the rationale for both approaches.

A.1 Letter Token Without Space: Token Similarity

This approach involves tokenizing the space independently in the prompt template, and looking at the probability of the model generating the label tokens "*t*" (without leading space). Many studies tokenize this way (Santurkar et al., 2023; Wang et al., 2024b,a; Pal and Sankarasubbu, 2024). The main potential reason is that the "*t*" tokens are exactly the same as the ones representing the corresponding options in the prompt (see Figure 1). A recent study highlights this exact match between the tokens in the options list and in the final answer as an important aspect (Gu et al., 2025). To quantify the strength of this argument, Figure 4 shows the token embedding similarity of the option letters.

Consider a MCQA task whose correct choice is "X". When computing the model probabilities for each option (final token), the embeddings of the tokens "␣X" (correct option with space) and "␣Y" (incorrect option with space) compared to "X" (ground truth token) are much more similar among them ($\approx$ 0.6 vs. $\approx$ 0.2, respectively) than if we compare the tokens "X" (correct option without space) and "Y" (incorrect option without space) to "X" (1.0 vs. $\approx$ 0.3, respectively). Therefore, it



	"A"	"B"	"C"	"D"	" A"	" B"	" C"	" D"
"A"	1.000	0.326	0.323	0.297	0.597	0.217	0.196	0.193
"B"	0.326	1.000	0.347	0.330	0.213	0.569	0.222	0.221
"C"	0.323	0.347	1.000	0.335	0.198	0.220	0.592	0.213
"D"	0.297	0.330	0.335	1.000	0.197	0.214	0.210	0.563
" A"	0.597	0.213	0.198	0.197	1.000	0.414	0.379	0.370
" B"	0.217	0.569	0.220	0.214	0.414	1.000	0.458	0.455
" C"	0.196	0.222	0.592	0.210	0.379	0.458	1.000	0.447
" D"	0.193	0.221	0.213	0.563	0.370	0.455	0.447	1.000

Figure 4: Cosine similarity of the Llama 3.1 token embeddings of the options, with and without space tokenization.

seems reasonable to use the same token as in the list of options in the prompt (i.e., "*t*" tokens) since the embeddings of the letter labels are more easily distinguishable. In a situation where the model is in doubt between the two options, this could allow clearer decision boundaries among the choices.

A.2 Letter Token With Space: Model's Default Tokenization

On the other hand, this other approach involves tokenizing the leading space together with the option letter, extracting the model predictions from the probabilities of the "␣*t*" tokens. The rationale for this choice is that it represents the default tokenization of the model after including the letter in the prompt – if we tokenize the string "`Answer: X`", the last token would be "␣X". Thus, this tokenization aligns better with what the model would expect to see, so it seems a plausible approach as well. This convention is also used in prior studies (Zheng et al., 2024; Hendrycks et al., 2021), and has been adopted for other tasks beyond MCQA, such as classification (Sanz-Guerrero and von der Wense, 2025).

B Detailed Experimental Setup

B.1 Datasets

Table 4 contains the list of datasets used in this study. All of them are evaluated using the default test set from Hugging Face[3], the size of which is specified in the table.

Dataset	|Test|
MMLU (Hendrycks et al., 2021)	14,042
AI2 ARC Easy (Clark et al., 2018)	2,365
AI2 ARC Challenge (Clark et al., 2018)	1,172
HellaSwag (Zellers et al., 2019)	10,003
OpenbookQA (Mihaylov et al., 2018)	500
TruthfulQA (Lin et al., 2022)	817

Table 4: Datasets (and their sizes) used in this paper.

Model
Llama 2 7B (Touvron et al., 2023)
Llama 3.1 8B (Grattafiori et al., 2024)
Llama 3.1 8B Instruct (Grattafiori et al., 2024)
Llama 3.1 70B (Grattafiori et al., 2024)
Llama 3.1 70B Instruct (Grattafiori et al., 2024)
Gemma 3 4B (Gemma Team et al., 2025)
Gemma 3 4B Instruct (Gemma Team et al., 2025)
Gemma 3 12B (Gemma Team et al., 2025)
Mistral 7B v0.3 (Jiang et al., 2023)
Mistral 7B Instruct v0.3 (Jiang et al., 2023)
Mistral Small 24B (Mistral AI, 2025)
Qwen 2.5 7B (Qwen et al., 2025)
Qwen 2.5 72B (Qwen et al., 2025)
Qwen 3 8B (Yang et al., 2025)
GPT Neo 2.7B (Black et al., 2021)

Table 5: LLMs evaluated in this paper.

B.2 Models

Table 5 contains the list of models used in this study. All of them are downloaded from the Hugging Face model hub[4], and their size is specified in the table.

B.3 Prompts

B.3.1 Main Prompt Templates

Figures 5 and 6 show the prompts used for base and instruction-tuned models, respectively. The difference between the two prompts is the inclusion of special tokens ({system token}, {user token}, and {assistant token}) in the instruction-tuned models, which are model-specific and represent the expected usage of these models, aligning with conversational interactions. In the figures, the relevant space tokens are marked as ␣. The last line of the prompt is the one that changes between the two tokenization strategies. The probability for each option is extracted from the next-token logits of the tokens after the arrow (→), which are underlined in red.

B.3.2 Prompt Variations

Recent work has shown that LLMs are highly sensitive to subtle changes in prompt phrasing and structure (Pezeshkpour and Hruschka, 2024; Zheng

[3] https://huggingface.co/datasets

[4] https://huggingface.co/models

MCQA Main Prompt (Base models)

```
“The following are multiple choice questions
(with answers).
Question: {question}
A. {option A}
B. {option B}
C. {option C}
D. {option D}
Answer:␣” → “X” // “Answer:” → “␣X”
```

Figure 5: Prompt used for base models. The relevant space tokens are marked as ␣. We analyze the probabilities of the tokens after the arrow (→), which are underlined in red. “X” denotes the option label (A/B/C/D).

MCQA Main Prompt (Instruction models)

```
{system token}
“You are a helpful assistant for
multiple-choice questions. Always answer
strictly in the format “Answer: X”, where X
is the letter of the chosen answer (A, B, C,
or D). Do not include any other text or
explanation.”
{user token}
“Question: {question}
A. {option A}
B. {option B}
C. {option C}
D. {option D}”
{assistant token}
“Answer:␣” → “X” // “Answer:” → “␣X”
```

Figure 6: Prompt used for instruction-tuned models. The relevant space tokens are marked as ␣. We analyze the probabilities of the tokens after the arrow (→), which are underlined in red. “X” denotes the option label (A/B/C/D). The {system token}, {user token}, and {assistant token} are model-specific special tokens.

et al., 2024; Alzahrani et al., 2024). To ensure that our findings are robust to such variations, we experiment with a range of prompt formulations, as analyzed in Section 5.2 (Table 3). Here, we provide the exact prompts used for each variation.

Figure 7 shows the prompt variation with a space before each option in the list. This modification ensures that the final answer token (“␣X”) matches the format of the options in the list (“␣A”, “␣B”, etc.). Figure 8 shows the prompt variation with parentheses around the option labels. Figure 9 shows the prompt variation with numeric option labels (1/2/3/4). Figure 10 shows the prompt variation with the list of options before the question.

Prompt Variation: Space in Option List

```
“The following are multiple choice questions
(with answers).
Question: {question}
␣A. {option A}
␣B. {option B}
␣C. {option C}
␣D. {option D}
Answer:␣” → “X” // “Answer:” → “␣X”
```

Figure 7: Prompt variation with a space before each option in the list. The relevant space tokens are marked as ␣. We analyze the probabilities of the tokens after the arrow (→), which are underlined in red. “X” denotes the option label (A/B/C/D).

Prompt Variation: Parentheses

```
“The following are multiple choice questions
(with answers).
Question: {question}
(A) {option A}
(B) {option B}
(C) {option C}
(D) {option D}
Answer:␣” → “(X)” // “Answer:” → “␣(X)”
```

Figure 8: Prompt variation with parentheses around the option labels. The relevant space tokens are marked as ␣. We analyze the probabilities of the tokens after the arrow (→), which are underlined in red. “X” denotes the option label (A/B/C/D).

Prompt Variation: Numbers

```
“The following are multiple choice questions
(with answers).
Question: {question}
1. {option A}
2. {option B}
3. {option C}
4. {option D}
Answer:␣” → “n” // “Answer:” → “␣n”
```

Figure 9: Prompt variation with numeric option labels. The relevant space tokens are marked as ␣. We analyze the probabilities of the tokens after the arrow (→), which are underlined in red. “n” denotes the option label (1/2/3/4).

Prompt Variation: Choices Before Question

```
"The following are multiple choice questions
(with answers).
A. {option A}
B. {option B}
C. {option C}
D. {option D}
Question: {question}
Answer:␣" → "X" // "Answer:" → "␣X"
```

Figure 10: Prompt variation with the list of options before the question. The relevant space tokens are marked as ␣. We analyze the probabilities of the tokens after the arrow (→), which are underlined in red. "X" denotes the option label (A/B/C/D).

B.4 Statistical Tests

Accuracy: McNemar's Test For accuracy, we use McNemar's test (McNemar, 1947), which assesses whether the number of examples correctly answered only under the "␣X" tokenization is significantly greater than those only correct under the "X" tokenization.

Calibration: Paired Bootstrap Resampling Test For calibration, we apply a paired bootstrap resampling test on the ECE. We resample the evaluation examples with replacement and recompute the ECE difference for each bootstrap sample, estimating the probability that one tokenization strategy leads to significantly lower ECE.

C Additional Results

C.1 Effect of Option Order Permutations

Motivated by recent works showing that LLMs are sensitive to the order of options in MCQA prompts (Zheng et al., 2024; Pezeshkpour and Hruschka, 2024), we experiment with 5 different random shufflings of the options in the prompt. These results are averaged in Table 3 under "Permutations (avg.)", and the individual results for each shuffling are reported in Table 6. We observe that not only the improvements from tokenizing the space with the letter (compared across columns) are consistent, but also that this effect is larger than the differences caused by changing the order of the options (compared across rows).

C.2 Performance across Languages

Table 7 demonstrates that the accuracy and calibration improvements from space–letter tokenization ("␣X") are robust across multiple languages. For all tested languages, including Spanish, German, French, Hindi, and Chinese, tokenizing the space together with the answer letter consistently yields higher accuracy and lower ECE. Notably, even in Chinese – a language not natively supported by Llama 3.1 (Grattafiori et al., 2024) – we observe a substantial gain of over 4 accuracy points and a reduction of 5 ECE points. This confirms that the effect is not limited to English prompts and generalizes to multilingual settings, regardless of the model's native language capabilities.

	Accuracy (↑)		ECE (↓)	
Option order	"X"	"␣X"	"X"	"␣X"
Original	61.47	**63.93***	2.58	**0.50***
Permutation 1	61.78	**63.86***	2.86	**0.64***
Permutation 2	61.43	**62.89***	3.29	**0.41***
Permutation 3	62.11	**63.95***	2.51	**0.61***
Permutation 4	61.28	**63.29***	2.93	**0.52***
Permutation 5	61.07	**62.85***	3.32	**0.61***

Table 6: Performance of Llama 3.1 8B on the MMLU benchmark with different random option orders.

	Accuracy (↑)		ECE (↓)	
Language	"X"	"␣X"	"X"	"␣X"
Spanish	54.0	**56.5***	3.7	**1.1***
German	52.9	**55.2***	3.4	**1.2***
French	53.2	**56.3***	3.9	**1.1***
Hindi	41.8	**45.5***	6.4	**2.1***
Chinese	47.6	**51.9***	7.2	**2.1***

Table 7: Performance of Llama 3.1 8B on the MMLU benchmark in different languages.

C.3 Performance across Datasets

Below we provide the complete per-model, per-dataset results that complement the aggregated deltas shown in Figure 3 of the main paper. We observe a consistent trend in favor of tokenizing the space together with the answer letter across datasets and model families. Even our largest evaluated model, Qwen 2.5 72B, shows a substantial gap of 11.7 accuracy points (on HellaSwag), indicating that larger models are also susceptible to such tokenization effects.

		Acc (↑)		ECE (↓)	
Model	Dataset	“X”	“␣X”	“X”	“␣X”
Llama 2 7B	ARC Challenge	43.9	**46.0***	2.6	**2.2**
	ARC Easy	57.6	**61.5***	**7.9**	9.0
	HellaSwag	26.8	**29.4***	8.2	**3.7***
	OpenbookQA	36.6	**39.4***	3.3	**2.9**
	TruthfulQA	23.3	**24.7**	4.9	**3.7***
Llama 3.1 8B	ARC Challenge	75.8	**78.8***	1.3	1.3
	ARC Easy	90.4	**91.6***	2.1	**1.9**
	HellaSwag	46.8	**52.8***	4.9	**2.6***
	OpenbookQA	73.2	**77.4***	2.4	**1.0***
	TruthfulQA	43.9	**46.0***	7.8	**7.1**
Llama 3.1 8B Inst	ARC Challenge	82.1	**82.7**	1.8	**1.5**
	ARC Easy	93.2	**93.5**	0.9	**0.2***
	HellaSwag	51.4	**59.3***	6.4	**1.9***
	OpenbookQA	81.4	**84.4***	**1.9**	2.4
	TruthfulQA	56.1	**57.8***	8.7	**8.3**
Llama 3.1 70B	ARC Challenge	91.8	**91.9**	1.4	**1.2**
	ARC Easy	97.2	**97.5**	1.5	**1.3**
	HellaSwag	65.4	**68.5***	**2.7**	3.0
	OpenbookQA	89.4	**90.8**	3.0	**2.7**
	TruthfulQA	57.6	**67.4***	4.6	**2.1***
Llama 3.1 70B Inst	ARC Challenge	93.0	**94.2***	2.0	**0.3***
	ARC Easy	97.7	**98.1**	2.0	**0.8***
	HellaSwag	64.9	**68.1***	3.1	**1.2***
	OpenbookQA	93.6	**94.2**	3.8	**2.1***
	TruthfulQA	73.6	**75.8***	**2.6**	**2.6**
Gemma 3 4B	ARC Challenge	70.6	**74.5***	5.2	**1.5***
	ARC Easy	86.2	**88.6***	2.9	**0.8***
	HellaSwag	37.3	**46.9***	8.9	**3.9***
	OpenbookQA	58.8	**65.4***	6.0	**2.3***
	TruthfulQA	31.0	**32.9**	15.0	**10.1***
Gemma 3 4B Inst	ARC Challenge	76.9	**77.0**	10.7	**10.6**
	ARC Easy	89.6	**89.9**	**4.7**	4.8
	HellaSwag	49.9	**50.4**	23.2	**22.9**
	OpenbookQA	74.0	**74.2**	12.3	**11.9**
	TruthfulQA	46.4	**47.0**	20.4	**19.8**
Gemma 3 12B	ARC Challenge	89.0	**89.3**	**0.9**	1.4
	ARC Easy	95.5	**96.0**	1.6	**0.6***
	HellaSwag	50.8	**53.9***	4.7	**1.8***
	OpenbookQA	83.4	**84.8***	4.7	**3.4**
	TruthfulQA	51.4	**55.0***	6.8	**3.1***
Mistral 7B v0.3	ARC Challenge	75.6	**76.7**	1.9	**1.8**
	ARC Easy	88.5	**88.7**	2.1	**2.0**
	HellaSwag	46.7	**47.8**	**2.6**	3.3
	OpenbookQA	72.4	**73.4**	3.0	**2.9**
	TruthfulQA	45.0	**45.4**	5.3	**3.9***
Mistral 7B Inst v0.3	ARC Challenge	78.4	**78.6**	6.3	**5.7**
	ARC Easy	88.3	**88.9**	3.1	**2.7**
	HellaSwag	60.0	**61.1**	9.6	**8.9**
	OpenbookQA	77.2	**77.6**	5.9	**5.0**
	TruthfulQA	48.6	**49.9***	13.1	**13.0**
Mistral Small 24B	ARC Challenge	92.2	**92.7**	1.4	**1.2**
	ARC Easy	97.7	**97.8**	1.9	**1.7**
	HellaSwag	55.4	**56.0**	4.1	**3.9**
	OpenbookQA	86.2	**86.4**	**3.3**	4.4
	TruthfulQA	67.3	**68.2**	**1.5**	**1.5**
Qwen 2.5 7B	ARC Challenge	88.3	**88.7**	1.4	**0.4***
	ARC Easy	95.4	**96.7***	1.6	**0.9***
	HellaSwag	59.6	**63.0***	1.2	**0.6***
	OpenbookQA	86.2	**89.0***	2.5	**1.7***
	TruthfulQA	60.8	**63.3***	**2.4**	3.2
Qwen 2.5 72B	ARC Challenge	95.6	**95.8**	2.3	**0.4***
	ARC Easy	98.5	**98.8**	1.8	**0.4***
	HellaSwag	76.7	**78.0***	3.7	**2.7***
	OpenbookQA	95.8	**96.4**	4.5	**2.0***
	TruthfulQA	63.3	**75.0***	3.2	**1.9***
Qwen 3 8B	ARC Challenge	90.5	**92.1***	1.1	**1.0**
	ARC Easy	97.3	**97.6**	1.5	**1.3**
	HellaSwag	63.4	**68.0***	3.5	**2.1***
	OpenbookQA	85.6	**85.6**	**2.3**	2.8
	TruthfulQA	60.2	**64.9***	3.2	**3.0**
GPT Neo 2.7B	ARC Challenge	23.8	**24.7**	9.4	**3.0***
	ARC Easy	24.6	**26.7***	10.0	**2.8***
	HellaSwag	26.2	**27.0***	14.1	**4.8***
	OpenbookQA	24.8	**26.2***	12.3	**2.9***
	TruthfulQA	22.4	**22.8**	7.5	**4.4***

Table 8: Full results of all models on all datasets tokenizing the space before (Letter token; “X”) or together with the letter (Space–Letter token; “␣X”). * means significantly better (higher for accuracy; lower for ECE). Top-performing tokenization strategy for each model is bolded.