🍔 MAC-Tuning: LLM Multi-Compositional Problem Reasoning with Enhanced Knowledge Boundary Awareness

Anonymous ACL submission

Abstract

With the widespread application of large language models (LLMs), the issue of generating non-existing facts, known as hallucination, has garnered increasing attention. Previous research in enhancing LLM confidence estimation mainly focuses on the single problem setting. However, LLM awareness of its internal parameterized knowledge boundary under the more challenging multi-problem setting, which requires answering multiple problems accurately simultaneously, remains underexplored. To bridge this gap, we introduce a novel method, **M**ultiple **A**nswers and **C**onfidence Stepwise **Tuning** (**MAC-Tuning**), that separates the learning of answer prediction and confidence estimation during fine-tuning. Extensive experiments across various base models and different model sizes demonstrate that our method proposed outperforms baselines by up to 25% in average precision.[1]

1 Introduction

Large language models (LLMs) are widely used in knowledge-intensive scenarios, such as question answering (Gu et al., 2023), information retrieval (Ren et al., 2023), and recommendation systems (Liu et al., 2023). Yet, they often produce non-existing facts when faced with questions outside their parametric knowledge, which undermines their reliability (Maynez et al., 2020). Many efforts have been dedicated to mitigating LLM hallucination, such as leveraging knowledge boundaries to constrain the reasoning scope of LLMs to help them better distinguish between reliable and unreliable information (Chen et al., 2024; Liang et al., 2024a; Zhang et al., 2024). Notably, these work mainly focus on the **single-problem setting**, where users repeatedly input questions and context for models to answer one by one.

[1]Our code and resource will be released upon publication.

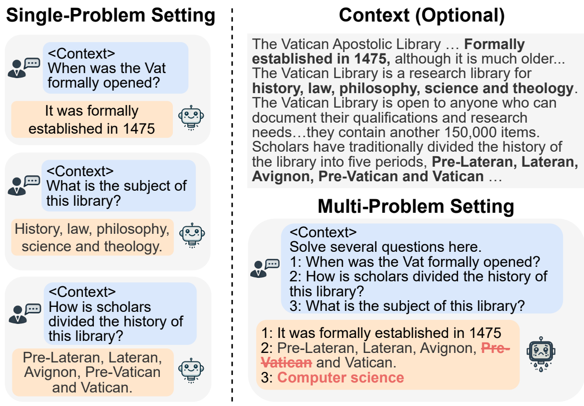

Figure 1: An illustration of the multi-problem setting. *Red* indicates that the LLM's output is inaccurate.

LLM hallucination in the **multi-problem setting** — in which a single input contains multiple distinct sub-questions with optional context for the model to extract and address — remains relatively underexplored. As seen in Figure 1, this is a fundamentally challenging setting because the model must distinguish each sub-question, reason over different knowledge, and synthesize results cohesively. Undesirable overshadowing of context from one sub-question with another, and propagation of reasoning confusion, may compromise the reliability of LLMs in multi-problem answering (Cheng et al., 2023a, Wang et al., 2024, Son et al., 2024,Li et al., 2024). As LLM-based multi-problem reasoning becomes increasingly widespread due to its efficiency benefits in scenarios involving extensive shared contexts (e.g., task instructions, exemplars), reduced model access, and lower API costs, enhancing model confidence estimation calibration for this emerging class of reasoning demands growing attention and effort as well.

In this paper, we investigate the hallucinations in LLMs within the multi-problem setting and propose leveraging the knowledge boundary to simultaneously handle the composition of multiple problems. Inspired by Zhang et al. (2024), which advocates for encouraging the LLM to express confidence to reduce hallucinations, we introduce **M**ultiple **A**nswers and **C**onfidence Stepwise **Tuning** (**MAC-Tuning**) under multi-problem setting. Our approach involves several key steps. First, we identify the knowledge boundary between parametric knowledge and the multi-problem dataset to extract uncertain questions. Next, we automatically label the model's confidence for both certain and uncertain data. These labeled data are then used to create multiple question-answer data and multiple QA-Confidence data so we can train the original model by separating the learning process of ground-truth answers and confidence, which enhances performance and reliability.

Our contributions can be summarized as follows:

- We are the first to explore LLM confidence estimation under the more challenging multi-problem setting, where LLMs must handle multiple problems simultaneously.
- We propose MAC-Tuning, which separates the learning process of answer and confidence predictions for enhancing knowledge boundary awareness and reducing hallucination.
- Through extensive experiments with different base models of varying sizes and various datasets, MAC-Tuning achieves an AP score gain of up to 25% over baselines in LLM multi-problem reasoning. Finally, we share our insights discovered to motivate future work.

2 Methodology

Figure 2 shows the data construction process for **M**ultiple **A**nswers and **C**onfidence **S**tepwise **T**uning (**MAC-Tuning**).

2.1 Multi-Problem Tuning Data Construction

First, we combine ***n*** single problems from original datasets to construct our initial Multi-Problem dataset. We utilize this to compare LLMs' outputs with ground-truth answers, for distinguishing the knowledge boundary between LLM parameters and instruction data. Specifically, for each individual problem in the multi-problem pair, we assign: "*I am sure*" if the output aligns with ground-truth answer; "*I am unsure*" elsewise (e.g., Step 2 in Figure 2). With the assigned confidence labels, we construct Multi-Problem Tuning data as follows:

Multiple QA pair D_{MultQA}: We directly combine the questions and answers together, with *Question* q_i as input and *Answer* a_i as output label, to form $D_{MultQA} = [(q_1, a_1)...(q_i, a_i)...(q_n, a_n)]$.

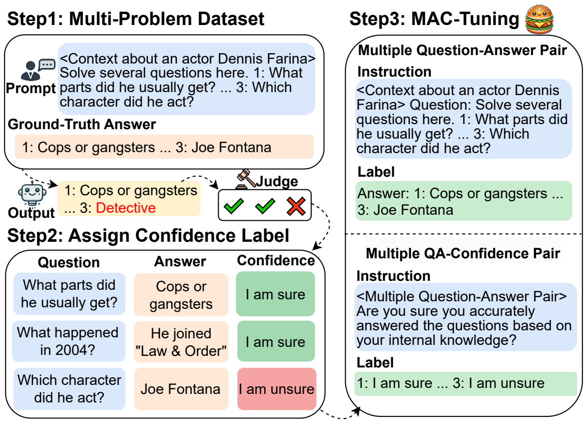

Figure 2: We first construct the Multi-Problem dataset, and then use it to generate Multi-Problem Tuning data.

Multiple QA-Confidence pair $D_{MultQA,C}$: The input consists of an instruction for the LLM to express its confidence (*i.e*, certainty in correctness) for a given question-answer pair, while the output is the confidence level in linguistic form[2].

2.2 Training and Inference

Using the Multi-Problem Tuning data, we conduct a two-step supervised fine-tuning process to train the model to answer questions and express confidence in a multi-problem setting. The objective for the first step, in answering question, is:

$$\max_{\Theta_0} \sum_{(Q,A) \in D_{MultQA}} \log P(A|Q; \Theta_0) \quad (1)$$

The objective for the second step, in expressing confidence, is:

$$\max_{\Theta_1} \sum_{(Q,A,C) \in D_{MultQA,C}} \log P(C|Q, A; \Theta_1) \quad (2)$$

where Q, A, and C represent the sets of multiple questions, multiple answers, and multiple confidence levels, respectively. Θ_0 and Θ_1 represent the parameters of the base model and the model after the first step of fine-tuning, respectively.

3 Experiment

3.1 Dataset

We validate the effectiveness of our method across different problem settings and datasets: for the *Independent* setting, where the questions are not related to each other, we use the **CoQA** (Reddy et al., 2019), **GSM** (Cobbe et al., 2021), **MMLU** (Hendrycks et al., 2021), and **ParaRel** (Elazar et al.,

[2]The template is in Appendix A.3

Model	Independent								Sequential			
	CoQA		ParaRel		GSM		MMLU		MTI-Bench		SQA	
	AP	ECE	AP	ECE	AP	ECE	AP	ECE	AP	ECE	AP	ECE
LLaMA3	54.6	22.6	45.1	40.8	79.3	52.8	50.3	43.8	37.4	17.7	44.9	35.4
QA-Only	66.3	15.1	53.7	12.6	75.3	36.1	58.5	17.9	45.0	16.9	56.6	21.0
Single-QA	65.5	28.9	73.5	10.7	56.6	44.5	58.3	25.7	N/A	N/A	N/A	N/A
Merge-AC	67.4	17.0	73.0	65.3	75.1	44.8	58.5	18.3	38.3	33.7	49.2	31.7
MAC-Tuning	**69.8**	**7.33**	**76.1**	**3.61**	**79.9**	**3.16**	**63.1**	**12.5**	**64.0**	**13.4**	**65.0**	**14.6**

Table 1: This is the confidence calibration result (%). We use one-shot CoT for GSM results. **Bold** font highlights the best performance for the dataset across different methods. We don't apply Single-QA to the *Sequential* setting dataset, as doing so would disrupt the logical connections among the questions.

2021) datasets; for the *Sequential* setting, where the questions are logically related to each other, we use the **MTI-Bench** (Son et al., 2024) and **SQA** (Iyyer et al., 2017) datasets. These datasets are either Question Answer (QA) or Multiple Choice (MC) formats. Table 2 shows the details of the dataset. Further information on the distribution of certain and uncertain data among the training set across different datasets is detailed in Appendix A.4.

	Independent				Sequential	
	CoQA	**ParaRel**	**GSM**	**MMLU**	**MTI-Bench**	**SQA**
Train	5006	7500	7468	2448	2400	3985
Test	5011	5584	1319	2439	600	925
Type	QA	QA	QA	MC	QA	QA

Table 2: Statistics of the datasets.

3.2 Evaluation Metrics

We directly compare the LLM generation to the ground-truth answer for the Question-Answer format. For Multiple-Choice format, we check the choice (A, B, C, D) and the option in the LLM generation. Across both types of answer generation tasks, we consider three evaluation metrics: (1) **Average Precision (AP)**: We use AP to measure the precision in identifying and ranking relevant predictions. A higher AP score means the model has high certainty about correct answers and high uncertainty about wrong answers. (2) **Expected Calibrated Error (ECE)**: We use ECE to measure how closely the predicted certainty reflects the true certainty of LLM (Chen et al., 2023). Low ECE indicates better-calibrated predictions. (3) **Accuracy**: We compute accuracy as the fraction of correct responses amongst questions in which LLMs expressed certainty towards their answers.

3.3 Baselines

We compare MAC-Tuning with the base model and its variants in the multi-problem settings. We use LLaMA3-8B-Instruct (**LLaMA3**) (Dubey et al., 2024) as the backbone. For baseline **QA-Only**, we fine-tune the base model directly using the Multiple Question-Answer pairs to evaluate the effectiveness of the traditional instruction tuning method under the multi-problem setting. For baseline **Single-QA**, we use single-problem data to fine-tune and directly apply it to the multi-problem setting. For baseline **Merge-AC**, instead of separating the learning process of ground-truth answers and confidence, we directly let the model learn multiple answers along with their corresponding confidence levels[3].

3.4 Overall Performance

In Table 1, we report the results on multi-problem setting from three single questions combined together. MAC-Tuning achieves the best AP score across all datasets, showing up to a 15% improvement, along with a lower ECE. This suggests that after MAC-Tuning, the model becomes more adept at distinguishing between certain and uncertain questions, delivering more reliable results through improved confidence estimation in answer prediction. We also evaluate each model's accuracy on every dataset. MAC-Tuning consistently outperforms the base model in accuracy by up to 45.8% and, on average, 23.7%. The reason is that we separate the tasks of learning correct answers and confidence within a multi-problem setting. After learning the ground-truth answer, the LLM can better understand confidence, while still retaining its ability to extract information, respond accurately, and address multiple problems simultaneously.

Ablation on Different Component We further test three variants of the MAC-Tuning method in the multi-problem setting: **QA-Only**, which is

[3]Baseline examples are in Appendix A.8. Implementation details are in Appendix A.6.

MAC-Tuning without the confidence component; **Single-QA**, where we evaluate MAC-Tuning with single problem data; and **Merge-AC**, where we evaluate MAC-Tuning without separating the learning process of ground-truth answers and confidence. As seen from the results in Table 1, **MAC-Tuning** has up to 25% and, on average, 11% AP improvement compared with **Merge-AC**, reflecting that separating the learning process of ground-truth answers and confidence is crucial in multi-problem setting, as LLM cannot learn both in one time. The performance of **Single-QA** is better than the base model but worse than **QA-Only** in most cases, showing that LLM can aware the knowledge boundary under single-problem setting and transfer it to multi-problem setting, but it is not sufficient for LLM to answer multiple problems simultaneously.

3.5 Investigation on Out of Domain Settings

We perform MAC-Tuning on base model with *Sequential* setting dataset SQA and test it on other datasets, with the results as presented in Table 3. Even on out-of-domain datasets, MAC-Tuning still outperforms the base model, showing that it can effectively learn the multi-problem setting and generalize across different domains.

Metric	CoQA	Pararel	MMLU	MTI-Bench
Accuracy	59.3	70.3	52.6	57.8
AP score	62.2	58.7	53.8	81.7
ECE	10.4	9.64	8.95	16.1

Table 3: The result (%) for MAC-Tuning on SQA dataset and test on other datasets.

3.6 Analysis on Various Number of Questions

We explore different numbers of questions in the multi-problem setting to investigate how this varies the accuracy. We only do this for three *Independent* setting datasets, and the results are reported in Figure 3. MAC-Tuning consistently outperforms the base model in accuracy by at least 10.0% and, on average, 26.8%. For easy tasks like ParaRel, The ability of the base model to handle multiple problems simultaneously is even higher when compared with the traditional single-problem setting, indicating that LLM could leverage in-context learning and focus on relevant knowledge better under multi-problem setting. However, for other datasets like MMLU, MAC-Tuning performs slightly worse as the question number increases. A reasonable explanation is that it is out of the base model's ability to learn too many hard tasks together but within effective scope to learn several easy tasks at the same time. Further studies are detailed in A.10.

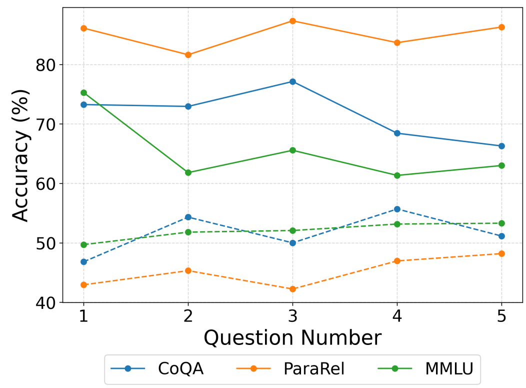

Figure 3: Accuracy for combining different number (***n***) of single problem together. Solid lines represent MAC Tuning, while dashed lines represent LLaMA3.

3.7 Analysis on Different Base Model

Table 4 shows the result from changing the base model to Qwen2-7B-Instruct (Yang et al., 2024). We observe that the performance trends remain consistent even with a different base model. MAC-Tuning continues to demonstrate an average precision (AP) gain of up to near 24% with a lower ECE, showcasing the effectiveness of learning ground-truth answers and confidence separately. Results for Llama-3.2-3B and Phi-3.5-mini-Instruct is shown in in Appendix A.9.

Approach	Independent				Sequential			
	ParaRel		MMLU		MTI-Bench		SQA	
	AP	ECE	AP	ECE	AP	ECE	AP	ECE
Vanilla	54.3	37.8	68.1	25.3	48.8	31.3	30.3	54.6
MAC-Tuning	**78.7**	**9.59**	**73.0**	**17.1**	**53.3**	**18.6**	**47.7**	**29.2**

Table 4: Confidence calibration result (%) for Qwen2-7B-Instruct, with **bold** denoting the top performance.

4 Conclusion

In this paper, we introduce a novel method, MAC-Tuning, to enhance large language model (LLM) confidence calibration and reasoning robustness in the challenging yet underexplored multi-problem scenario. Our proposed approach automatically constructs multi-problem setting question-answer pairs with confidence annotations for identifying the intrinsic knowledge gap between parametric knowledge and instructional data. With this data constructed, we guide the LLM to better reason on answer prediction and confidence estimation separately, in multi-problem setting. Extensive experiments across different datasets show that our method significantly improves performance in areas where the original LLM struggles.

Limitation

While our work provides valuable insight on the new Multiple Question setting and introduces an innovative fine-tuning method, there are several limitations to acknowledge. First, although we experimented with various prompts, as is typical in prompt-based LLM studies, we cannot ensure that slight changes in prompts would not significantly alter the results. Second, due to constraints of cost, time, and computational resources, we selected a subset of experiments that we believe to be informative and representative. However, additional experiments across a wider range of datasets and LLMs might provide further insights. Lastly, in this new setting, there may be other underlying reasons for the experimental results. Future work will aim to address these limitations by expanding datasets and conducting new experiments to explore other potential factors affecting performance.

References

Lida Chen, Zujie Liang, Xintao Wang, Jiaqing Liang, Yanghua Xiao, Feng Wei, Jinglei Chen, Zhenghong Hao, Bing Han, and Wei Wang. 2024. Teaching large language models to express knowledge boundary from their own signals.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. 2023. A close look into the calibration of pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1343–1367, Toronto, Canada. Association for Computational Linguistics.

Zhoujun Cheng, Jungo Kasai, and Tao Yu. 2023a. Batch prompting: Efficient inference with large language model APIs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 792–810, Singapore. Association for Computational Linguistics.

Zhoujun Cheng, Jungo Kasai, and Tao Yu. 2023b. Batch prompting: Efficient inference with large language model apis.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Ed Hovy, Hinrich Schutze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *ArXiv*, abs/2102.01017.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey.

Yu Gu, Xiang Deng, and Yu Su. 2023. Don‘t generate, discriminate: A proposal for grounding language models to real-world environments. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4928–4949, Toronto, Canada. Association for Computational Linguistics.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Shengding Hu, Yifan Luo, Huadong Wang, Xingyi Cheng, Zhiyuan Liu, and Maosong Sun. 2023. Won't get fooled again: Answering questions with false premises. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5626–5643, Toronto, Canada. Association for Computational Linguistics.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada. Association for Computational Linguistics.

Ming Li, Pei Chen, Chenguang Wang, Hongyu Zhao, Yijun Liang, Yupeng Hou, Fuxiao Liu, and Tianyi Zhou. 2024. Mosaic-it: Free compositional data augmentation improves instruction tuning.

Qiuyu Liang, Weihua Wang, Feilong Bao, and Guanglai Gao. 2024a. L^2GC:lorentzian linear graph convolutional networks for node classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9988–9998, Torino, Italia. ELRA and ICCL.

Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. 2024b. Learning to trust your feelings: Leveraging self-awareness in LLMs for hallucination mitigation. In *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, pages 44–58, Bangkok, Thailand. Association for Computational Linguistics.

Jianzhe Lin, Maurice Diesendruck, Liang Du, and Robin Abraham. 2024. Batchprompt: Accomplish more with less.

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2023. A first look at llm-powered generative news recommendation. *ArXiv*, abs/2305.06566.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Ruiyang Ren, Wayne Xin Zhao, Jing Liu, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Tome: A two-stage approach for model-based retrieval.

Guijin Son, Sangwon Baek, Sangdae Nam, Ilgyun Jeong, and Seungone Kim. 2024. Multi-task inference: Can large language models follow multiple instructions at once?

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision.

Zhengxiang Wang, Jordan Kodner, and Owen Rambow. 2024. Exploring the zero-shot capabilities of llms handling multiple problems at once.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2024. R-tuning: Instructing large language models to say 'I don't know'. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7113–7139, Mexico City, Mexico. Association for Computational Linguistics.

A Appendix

A.1 Full Case for Examples of Introduction

Full case for the examples in introduction can be found in Figure 4.

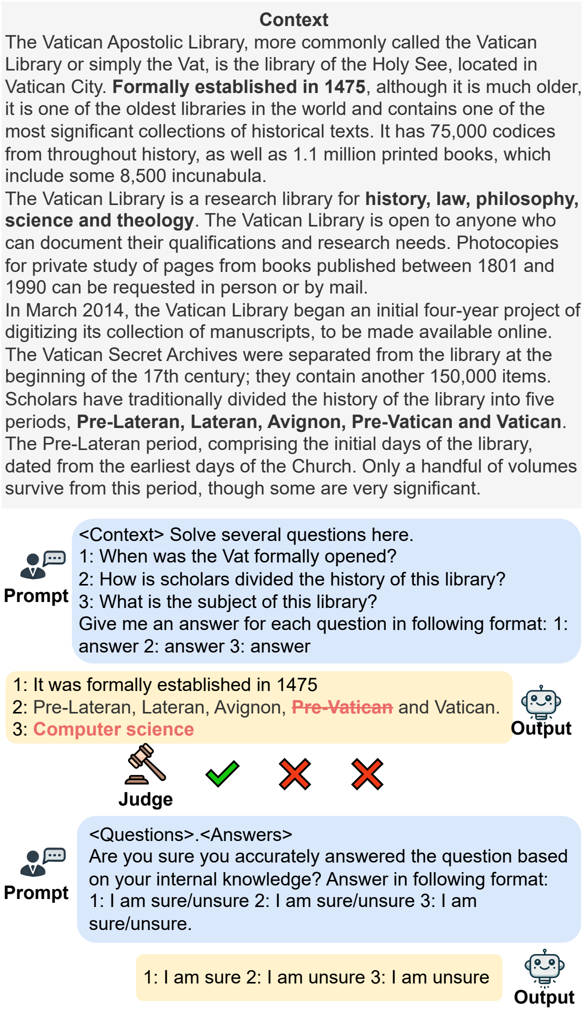

Figure 4: The full case of examples in introduction in Multiple Problem setting. Red context indicates that LLM's output is inaccurate. The second answer lacks the information of "Pre-Vatican" and the third answer contains a completely factual error. After MAC-Tuning, LLM show uncertainty towards answering this two previously incorrect questions.

A.2 Related Work

Hallucination: Large language models (LLMs) are widely used in knowledge-intensive scenarios, such as question answering (Gu et al., 2023), information retrieval (Ren et al., 2023) and recommendation systems (Liu et al., 2023). However, LLMs have tendency to generate non-existing facts when faced with questions that are out of their parametric knowledge (Maynez et al., 2020). Many efforts are dedicated to mitigating hallucinations in LLMs, such as retrieval-augmented generation (Gao et al., 2024,Peng et al., 2023), multi-agent debate (Du et al., 2023,Sun et al., 2023), and model confidence calibration (Zhang et al., 2024, Hu et al., 2023).
Knowledge Boundary: There are many different ways to utilize knowledge boundary to reduce LLM hallucination. (Liang et al., 2024b)'s work uses merged knowledge probing and consistency checking methods to help LLM express their internal knowledge. (Chen et al., 2024)'s work leverages LLM internal signals to let LLM know their unknowns. (Zhang et al., 2024) utilize knowledge boundary to instruct LLM say "I don't know". It is a popular way to use confidence to express knowledge boundary of LLMs and we also follow this.

Multiple Problem Setting: Current LLM research has predominantly focused on single problem setting. There are only a few works focusing on this new setting. (Cheng et al., 2023a) propose batch prompting that prompts LLMs with single independent problems batched together following few-shot exemplars together. Son et al. (2024) goes further by researching sequential datasets and develops the first multi-task benchmark (MTI-Bench). Wang et al. (2024) pays attention to zero-shot cases of multi-problem setting and design a new benchmark ZeMPEB. Li et al. (2024) analyze different strategy under independent setting, where single questions are combined into various constraint formats without sharing context between them. Despite these efforts, the multi-problem setting presents significant challenges. For example, Wang et al. (2024) shows that in zero-shot setting, LLMs consistently perform worse when selecting indices of texts for a given class label with multiple mixed-source reasoning problems. Similarly, for few-shot setting, Cheng et al. (2023b) and Lin et al. (2024) have found that the overall accuracy decreases with the increase in batch size. Notably, this setting is also meaningful in real-world applications: for independent scenario, batching unrelated queries can reduce model calls and API costs; for sequential senario, where questions share context—such as in math problem solving, data processing, or software debugging—the correctness of each intermediate reasoning step is critical. Overall, hallucination and performance instability under the multi-problem setting are still under-explored and present significant challenges for current LLMs.

A.3 Template for QA-Confidence pair

> Question: <*Question*>. Answer: <*Answer*>. Are you sure you accurately answered the question based on your internal knowledge?
> 1: <*Confidence*> 2: <*Confidence*> 3: <*Confidence*>

A.4 Dataset Details

We carry out our experiments across six datasets, described as follows.

- **GSM** (Cobbe et al., 2021): a dataset containing high-quality grade school math problems created by the OpenAI group. These problems require between 2 and 8 steps to solve, primarily involving a sequence of elementary calculations with basic arithmetic operations such as addition, subtraction, multiplication, and division to arrive at the final answer. We directly use 7.5k training data and 1k testing data in our Question Answer setting.

- **Pararel** (Elazar et al., 2021): a dataset containing factual knowledge with a variety of prompts and relationships, originally created for mask prediction. In Question Answer setting, we employ the modified dataset from Zhang et al. (2024).

- **MMLU** (Hendrycks et al., 2021) a dataset covering different subjects and difficulty. It tests both world knowledge and problem solving ability, which has good granularity and breath. We directly use the modified dataset from Zhang et al. (2024) in our Multiple Choice setting.

- **CoQA** (Reddy et al., 2019) a dataset designed to evaluate the ability of models to understand and generate answers in a conversational setting. We randomly pick 5k training dataset from theirs. In Question Answer setting, we combine multiple questions together under the same "story" category in the dataset.

- **MTI Bench** (Son et al., 2024) a comprehensive evaluation benchmark encompassing 5,000 instances across 25 tasks. We pick the sequential part of this benchmark and divide it into 800 training data and 200 test data.

- **SQA** (Iyyer et al., 2017) a dataset designed to explore the task of answering sequences of inter-related questions on HTML tables. We pick 5 sequential questions for each HTML table and have 3985 training data.

A.5 Formula and Calculation Details

Average Precision (AP) Score measures the performance of a binary classifier's confidence rankings. It corresponds to the area under the Precision-Recall curve. It is calculated as follows:

$$\text{AP} = \sum_{k=1}^{n} \Big(R_k - R_{k-1} \Big) \times P_k$$

where k is the number of data at current thread with precision P_k and recall R_k. n is the total data number. The confidence is the weighted average of certain prediction probability and uncertain prediction probability.

Expected Calibrated Error (ECE) indicates how well a model's predicted probabilities match the true likelihood of an event. We split the predictions into 10 bins based on the certain prediction probability, then compare the average predicted probability to the actual proportion of positive samples (correct cases) in each bin. It is calculated as follows:

$$\text{ECE} = \sum_{m=1}^{10} \frac{|B_m|}{n} \left| \overline{p}_m - \overline{y}_m \right|$$

where m is the bin number with corresponding average predicted probability $\overline{p}_m$ and actual proportion of positive samples $\overline{y}_m$.

A.6 Implementation

We use HuggingFace PEFT (Mangrulkar et al., 2022) to conduct LoRA fine-tuning (Hu et al., 2021). We set the training epoch to 3, learning rate to $1\mathrm{e}^{-5}$, LoRa rank to 8, and LoRa scaling factor to 32. The batch size is 1 and the temperature is 0. All experiments are implemented on Nvidia A100-40GB GPUs.

A.7 Case Study

We show two specific cases for MAC-Tuning under the multiple problem setting with question number $n = 3$ in Figure 5. The example on the left is from the SQA dataset, in which a table context is given and the LLM need to answer sequential questions based on the table. LLM answers correctly and shows certainty to first two questions, so these two questions will be counted into accuracy calculation. It answers wrong and shows uncertainty to the third question, which achieves the refusal behavior that we aim to see. The example on the right is from the GSM dataset. The LLM gives wrong answers to the second question but indicates certainty, which means this is a failure case.

A.8 Detailed Information for Variant Methods

The detailed example for different baseline methods is shown in Figure 6.

A.9 Impact of Model Size on Performance

The confidence calibration results for Llama-3.2-3B and Phi-3.5-mini-Instruct is shown in table 5 and table 6 respectively. Despite using different base models of varying sizes, the results follow the same trend as reported in the main paper. For smaller models, the accuracy improvement after **MAC-Tuning** is more evident, indicating enhanced ability to differentiate between certain and uncertain questions.

Approach	**Independent**				**Sequential**			
	Pararel		**CoQA**		**MTI-Bench**		**SQA**	
	AP	ECE	AP	ECE	AP	ECE	AP	ECE
Vanilla	30.7	70.3	46.0	45.0	34.3	70.3	35.6	45.0
MAC-Tuning	**55.5**	**33.5**	**62.4**	**33.5**	**35.5**	**33.5**	**44.3**	**33.5**

Table 5: Confidence calibration result (%) for Llama-3.2-3B, with **bold** denoting the top performance across different methods.

Approach	**Independent**				**Sequential**			
	Pararel		**CoQA**		**MTI-Bench**		**SQA**	
	AP	ECE	AP	ECE	AP	ECE	AP	ECE
Vanilla	58.0	22.8	56.0	32.9	21.4	29.1	96.6	33.7
MAC-Tuning	**70.2**	**14.2**	**68.2**	**29.0**	**68.7**	**22.6**	**52.3**	**23.9**

Table 6: Confidence calibration result (%) for Phi-3.5-mini-Instruct, with **bold** denoting the top performance across different methods.

A.10 Cross Task Transfer Study

We fine-tune the model with question number $\boldsymbol{n}$ = 3 and let it response to question number $\boldsymbol{n}$ = 1 (which is Single Problem setting) and question number $\boldsymbol{n}$ = 5. For the former one, we want to test if the model understands the single problem. For the latter one, we want to test if MAC-Tuning can generalize across different question number $\boldsymbol{n}$. The results are reported in Table 7.

Question Number	**CoQA**	**ParaRel**	**GSM**	**MMLU**
n = 1	78.78	84.24	71.12	54.60
n = 5	79.12	86.16	67.67	63.69

Table 7: Accuracy (%) for MAC-Tuning with question number $\boldsymbol{n}$ = 3 transferring to question number $\boldsymbol{n}$ = 1 (which is single-problem setting) and question number $\boldsymbol{n}$ = 5. We use one-shot CoT for GSM results.

From the result of $\boldsymbol{n}$ = 1, we observe that accuracy improves in easy dataset like CoQA but decreases in difficult dataset like GSM, comparing with using question number $\boldsymbol{n}$ = 1 to fine-tune. This indicates that LLM learn knowledge during fine-tuning rather than just memorizing the patterns for multi-problem setting. With $\boldsymbol{n}$ = 5, the model performs about the same or even better than using question number $\boldsymbol{n}$ = 5 to fine-tune. This strengthen the statement we make in Section 3.6 that LLMs are more efficient at learning multiple easy tasks but struggle with several difficult tasks simultaneously.

A.11 Certainty Distribution of the Training Dataset

We demonstrate the certainty distribution of the training dataset under Multiple Problem setting with question number n = 3 in Figure 7:

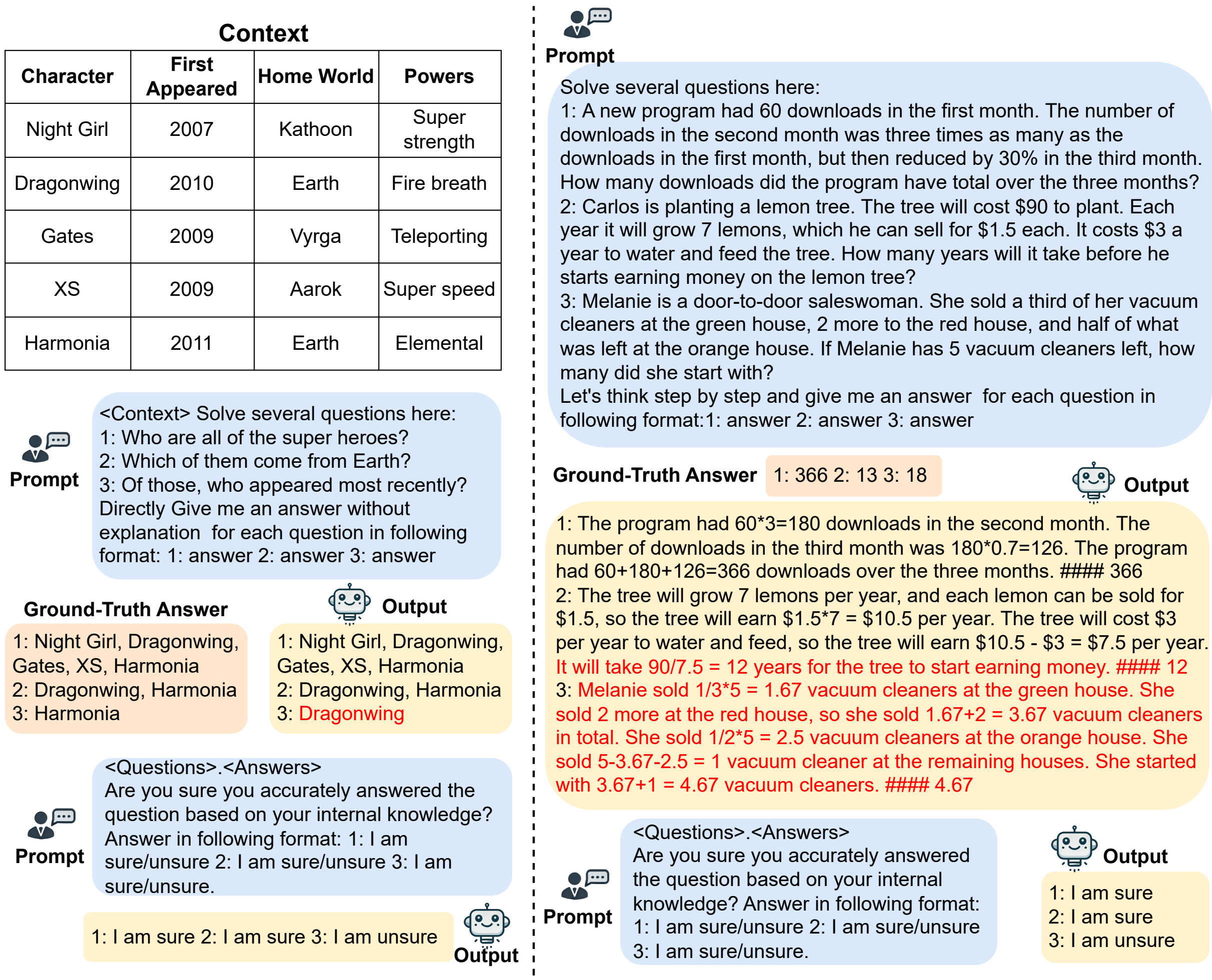

Figure 5: Specific cases for MAC-Tuning under the multiple problem setting with question number $n = 3$. Red-highlighted context indicates inaccuracies in the LLM's output. The left example is drawn from the *Sequential* setting dataset (SQA), while the right example is from the *Independent* setting dataset (GSM), with one-shot context omitted for conciseness.

QA-Only

Input Instruction

<Shared Context> Question: Solve several questions here.
1: Did Jane think she could control Lassiter? 2: Who is Bells?
3: How did he get his name?
Directly Give me an answer without explanation for each question in following format: 1: answer 2: answer 3: answer

Label

Answer:1: No 2: One of her safe racers 3: because of the way he struck his iron shoes on the stones.

Merge-AC

Input Instruction

<Shared Context> Question: Solve several questions here.
1: Did Jane think she could control Lassiter? 2: Who is Bells?
3: How did he get his name?
Directly Give me an answer without explanation for each question in following format: 1: answer 2: answer 3: answer

Label

Answer: 1: No 2: One of her safe racers 3: because of the way he struck his iron shoes on the stones.
Are you accurately answered the question based on your internal knowledge? 1: I am sure 2: I am unsure 3: I am sure

Single-QA

Input Instruction

<Shared Context> Question: Solve several questions here.
1: Did Jane think she could control Lassiter?
Directly Give me an answer without explanation for each question in following format: 1: answer

Label

Answer: 1: No
Are you accurately answered the question based on your internal knowledge? 1: I am sure

Shared Context

CHAPTER VII. THE DAUGHTER OF WITHERSTEEN
"Lassiter, will you be my rider?" Jane had asked him. "I reckon so," he had replied.
Few as the words were, Jane knew how infinitely much they implied. She wanted him to take charge of her cattle and horse and ranges, and save them if that were possible. Yet, though she could not have spoken aloud all she meant, she was perfectly honest with herself. Whatever the price to be paid, she must keep Lassiter close to her; she must shield from him the man who had led Milly Erne to Cottonwoods. In her fear she so controlled her mind that she did not whisper this Mormon's name to her own soul, she did not even think it. Besides, beyond this thing she regarded as a sacred obligation thrust upon her, was the need of a helper, of a friend, of a champion in this critical time. **If she could rule this gun-man, as Venters had called him, if she could even keep him from shedding blood, what strategy to play his flame and his presence against the game of oppression her churchmen were waging against her?** Never would she forget the effect on Tull and his men when Venters shouted Lassiter's name. If she could not wholly control Lassiter, then what she could do might put off the fatal day.
One of her safe racers was a dark bay, and she called him Bells because of the way he struck his iron shoes on the stones. When Jerd led out this slender, beautifully built horse Lassiter suddenly became all eyes. A rider's love of a thoroughbred shone in them. Round and round Bells he walked, plainly weakening all the time in his determination not to take one of Jane's favorite racers.

Figure 6: A specific case to show how baseline methods are doing the fine-tuning. The answers are derived from the highlighted portions of the context. In QA-Only, the input is the Question instruction, and the output is the Answer. In Merge-AC, the output includes both the Answer and its Confidence. Single-QA is the single-problem variant of Merge-AC.

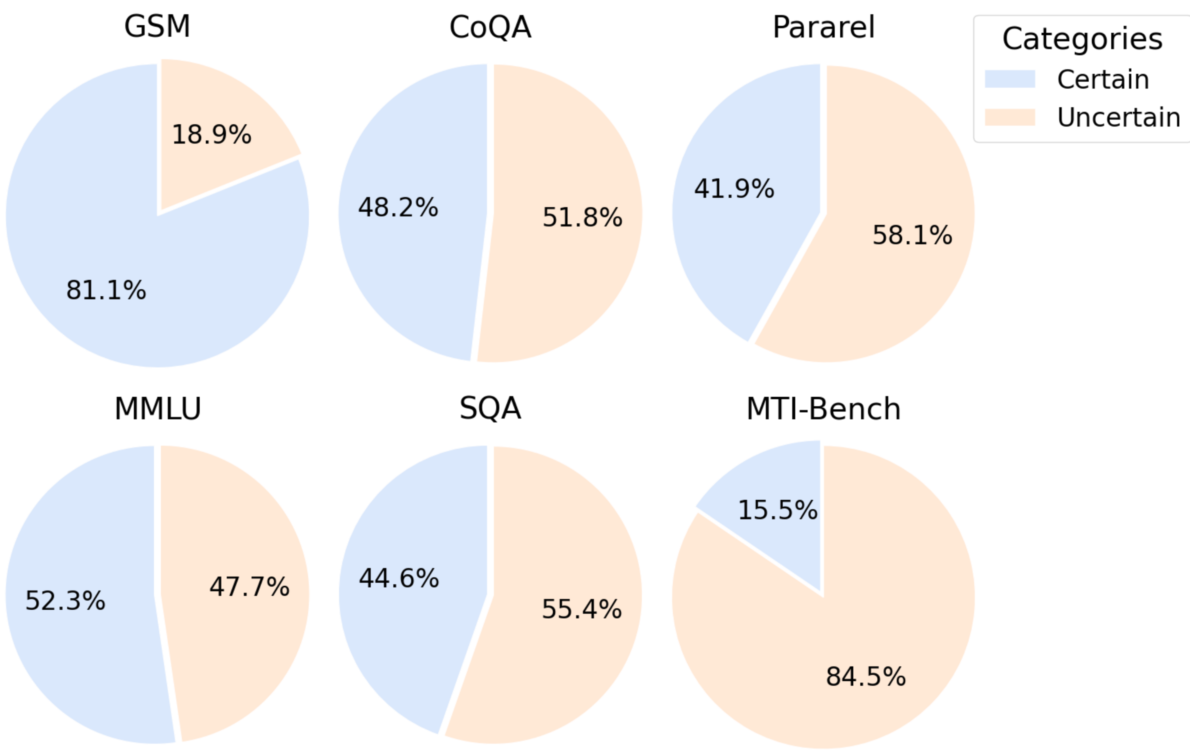

Figure 7: Certainty distribution of the training set under multi-problem setting with $n = 3$