Large Language Model Cascades with Mixture of Thoughts Representations for Cost-efficient Reasoning

Murong Yue[1], Jie Zhao[2], Min Zhang[3], Liang Du[2], Ziyu Yao[1]

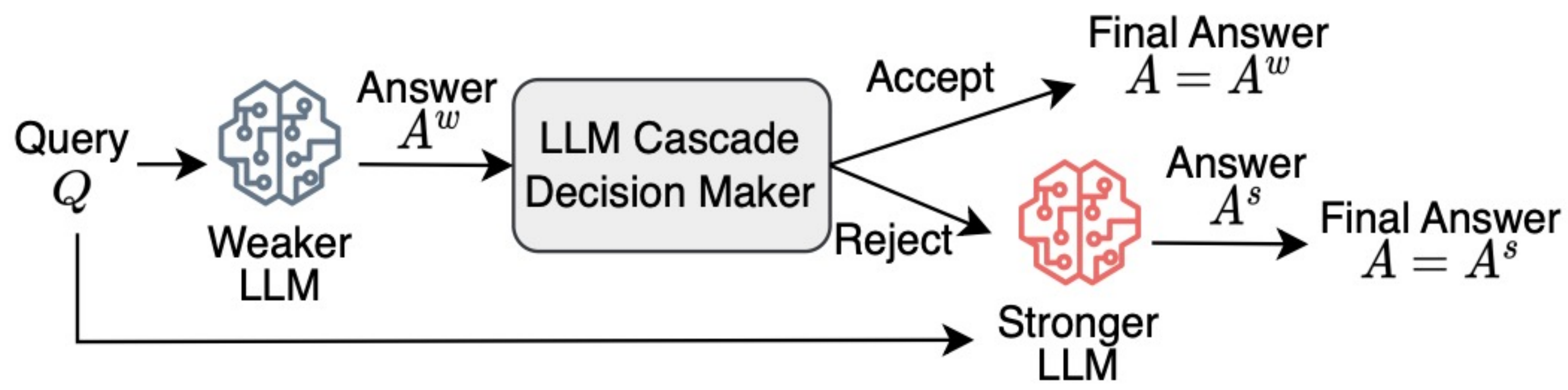
[1]George Mason University, [2]Microsoft, [3]Virginia Tech

Paper & Code

Overview

LLMs exhibit remarkable performance but come with high expense. We are motivated to design a cascade following the intuition that **simple questions** could be answered by **weaker LLM**, whereas only the **challenging questions** necessitate the **stronger LLM**.

We leverage a cascade to save the cost. Given the question, the cascade first leverage the weaker LLM to get an answer and then decide to accept or reject the answer. The key component is the decision maker. In our work, we propose to make the decision based on the "**answer consistency**" with a **mixture of two thought representations** (i.e., CoT [1] and PoT [2]).



LLM Cascades for Cost-efficient Reasoning

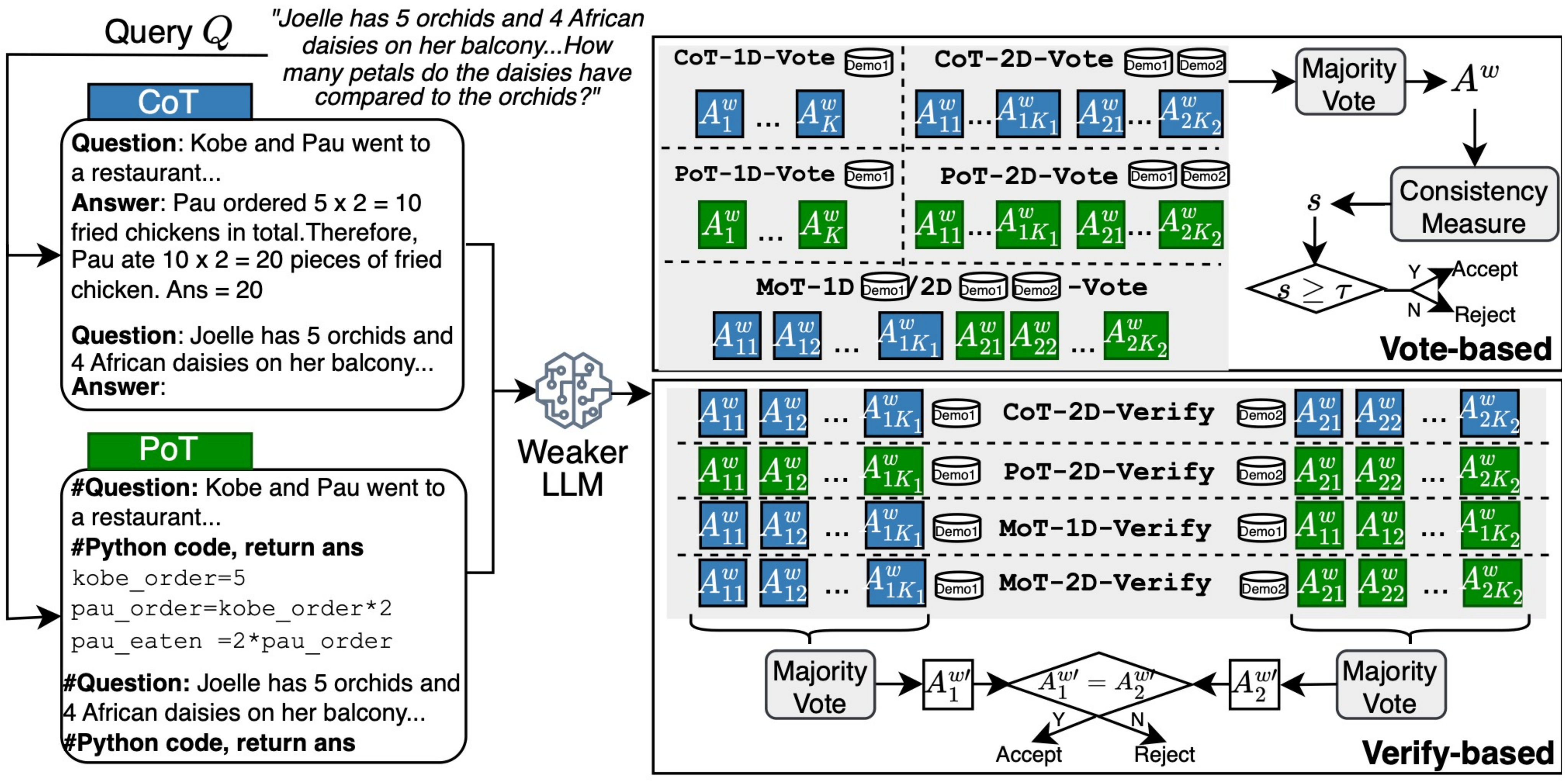
We set a non-zero temperature and have different sampling strategies:

- **Sampling with the same prompt (self-consistency):** Sampling multiple answers given the same prompt input [3].
- **Sampling with different demonstrations:** Sampling answers with prompts that have different in-context demonstration examples.
- **Sampling with different representations:** Sampling answers with prompts that have different representation of intermediate steps.

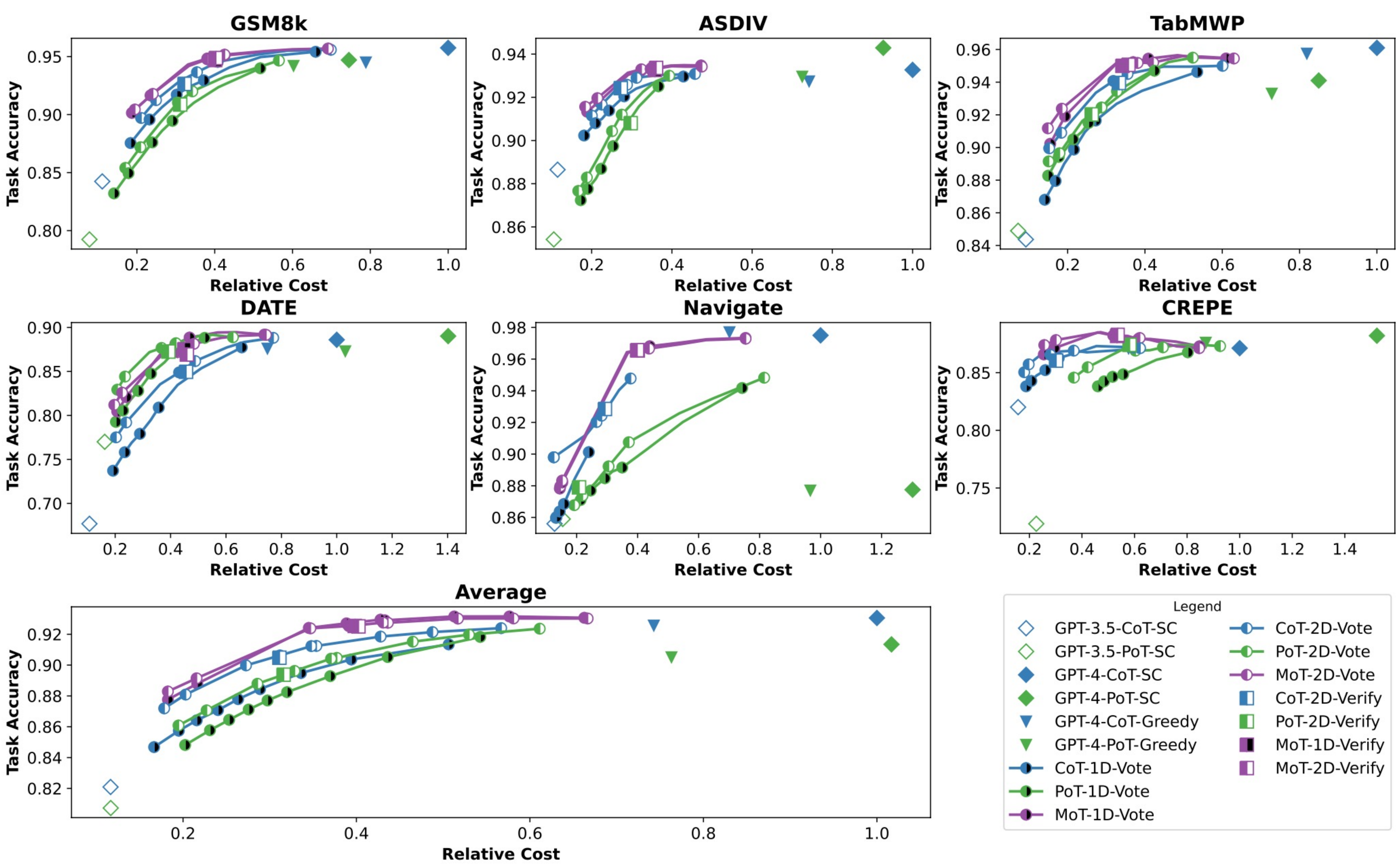
	Same Examples	*Same Representation*	*Different Examples*	*Different Representations*
CoT-1D	√	√		
PoT-1D	√	√		
MoT-1D	√			√
CoT-2D		√	√	
PoT-2D		√	√	
MoT-2D			√	√

Vote-based: Examining if the agreement score of the majority voted answer is larger than a pre-defined threshold.

Verification-based: Checking if the majority voted answers sampled from different prompts are consistent.



Experimental Results



1. Our pipeline achieves comparable task performance with GPT-4 but requires only 40% of costs.
2. Sampling from the mixture of thought representations (MoT) is particularly effective.
3. Increasing the threshold yields marginal benefits for the vote-based approach.

Gold

Black runs 200 yards for each back and forth because...He runs 3000 yards in total because...Kelly runs 200 yards at first because 100 x 2 = 200. She runs 80 yards each time she does a back and forth to the 40-yard line because... She runs 2720 from these back and forth. She runs 2920 in total because...The winner ran 80 yards more because 3000 - 2920=80.

CoT 1

Answer: [...(Ignored)] Kelly runs to the 40-yard line and back 34 times, so she runs 34*80=2720 yards. Thus, Blake runs 3000-2720=280 yards farther than Kelly.

CoT 2

Answer: [...(Ignored)] Kelly runs 34 x (40 x 2) = 2720 yards. Blake runs 3000 - 2720 = 280 yards farther than Kelly.

PoT 1

```
# Python code, return ans
[...(Ignored)]
distance_covered_by_kelly =
(length_of_football_field *
num_of_laps_for_kelly * 2) + (40 *
num_of_laps_for_kelly * 2)
ans = abs(distance_covered_by_blake
- distance_covered_by_kelly)
```

(Answer via Python execution: 6520.0)

Logic Generation Error   Value Grounding Error

MoT could introduce more "opinions" in hard questions.

Further Analysis

Can we fine-tune a small LM [4] or directly use the weaker LLM as the verifier?

They cannot yield satisfying results in complex reasoning tasks, which can be due to the intrinsic challenge of deciding question difficulty and answer correctness solely based on their textual descriptions.

How weak can the weaker LLM be?

It cannot be too weak. When it struggles with the task, our cascade doesn't work.

Is our method robust to hyperparameters?

MoT is always better with different sampling temperatures and sample sizes.

Can stronger LLM learn from weaker LLM?

Leveraging answers from the weaker LLM may mislead the stronger LLM.

Future Work

Can our method be prompted to more general tasks without a specific answer? E.g., Universal Self-Consistency [5].

Can our method be applied to multiple (>2) LLMs or to reduce the hallucination of LLMs?

Can we distill knowledge from different representations?

Reference

[1] Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." Advances in Neural Information Processing Systems 35 (2022): 24824-24837.
[2] Chen, Wenhu, et al. "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks." arXiv preprint arXiv:2211.12588 (2022).
[3] Wang, Xuezhi, et al. "Self-consistency improves chain of thought reasoning in language models." International Conference on Learning Representations. ICLR, 2023
[4] Chen, Lingjiao, Matei Zaharia, and James Zou. "FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance." arXiv preprint arXiv:2305.05176 (2023).
[5] Chen, Xinyun, et al. "Universal self-consistency for large language model generation." arXiv preprint arXiv:2311.17311 (2023).