Can We Truly Measure Uncertainty in Language Models?

Xueyan Li, Dr Jonas Geiping

2nd Supervisor:
Prof Mrinmaya Sachan

ETH zürich

Token Level Uncertainty-Aware Chain-of-Thought Reasoning

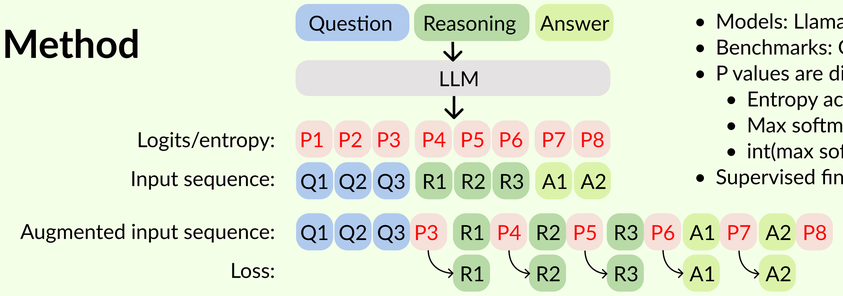
Motivation:

- Step-wise token probabilities are discarded in future generations → future generations condition on previous tokens only, their uncertainties are lost
- Could be valuable to propagate current uncertainty to future steps as a signal to reflect or reason more around past uncertainty

Identifying errors in reasoning chains is a very hard task!

Model	Acc
GPT-4-Turbo	38.33
GPT-4	44.00
GPT-3.5-Turbo	20.00
Gemini Pro	21.67
PaLM 2 Unicorn	22.00
Llama-3.1-8B	20.00
Llama-3.1-8B uncertainty aware finetuned	23.33

Method

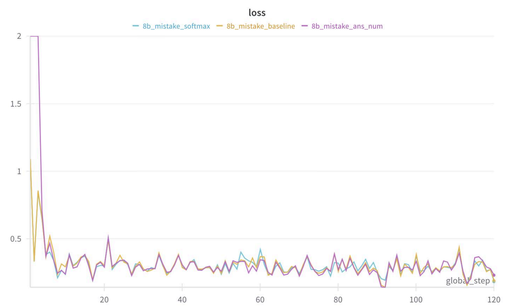


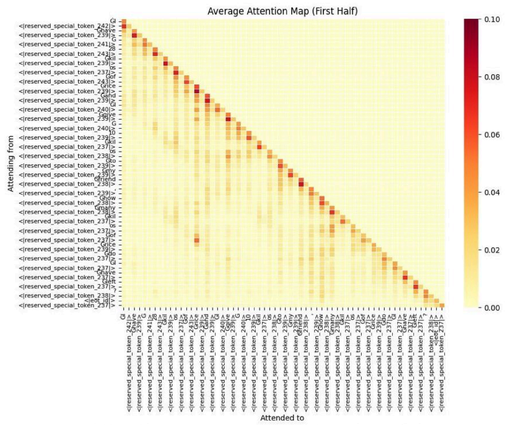
- Models: Llama3.1-1B and 8B
- Benchmarks: GSM8K, big-bench-mistake
- P values are discretized into 10 bins
  - Entropy across vocabulary at each step
  - Max softmax value
  - int(max softmax value *10)
- Supervised fine-tuning

Results

- No improvement in performance after SFT. Similar or deteriorated performance
- Attention map for text and confidence tokens show vertical strips with gaps where the gaps are the lack of attention to special reserved tokens → special confidence tokens are ignored
- Baseline and interleaving setup have nearly identical loss curves
- Various P values result in different start loss, but converge to around the same point

→ not learning anything from uncertainty values!!!





Attention map across text and confidence tokens for GSM8K

user
If I have 28 kilos of rice and I give 10 kilos to my friend, how many kilos of rice do I have left?assistant

If I give 10 kilos of rice to my friend, I have 28 - 10 = <<28-10=18>>18 kilos of rice left. #### 18

Are these model uncertainties actually truthful?

Future improvement

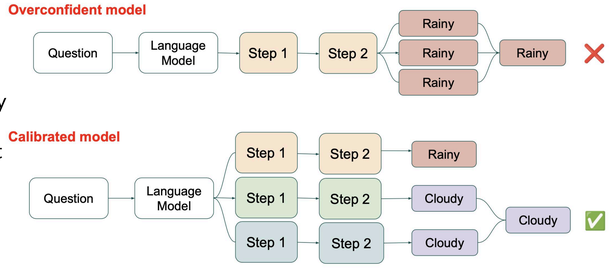
- RL to discover reasoning paths that utilize uncertainties and eventually leads to the correct final answer.
- Different ways to insert uncertainty values e.g. in the embedding dimension
- Calibrate the model uncertainties first, so they more truthfully reflect the underlying correctness.

References:
Tyen, G., Mansoor, H., Cărbune, V., Chen, P., & Mak, T. (2023, November 14). LLMs cannot find reasoning errors, but can correct them given the error location. arXiv.org. https://arxiv.org/abs/2311.08516
Xie, J., Chen, A. S., Lee, Y., Mitchell, E., & Finn, C. (2024, September 29). Calibrating Language Models with Adaptive Temperature Scaling. arXiv.org. https://arxiv.org/abs/2409.19817

Uncertainty-Aware Temperature Adaptation for Better Calibration

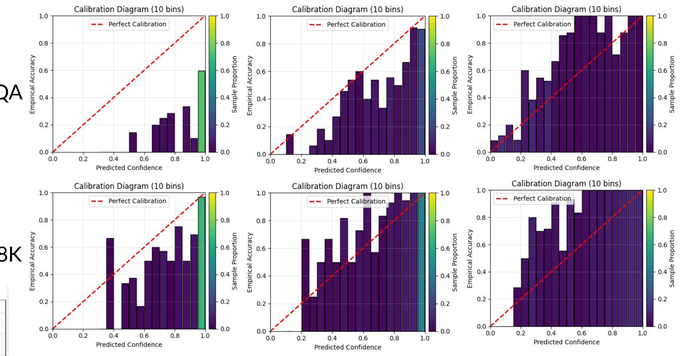
Motivation

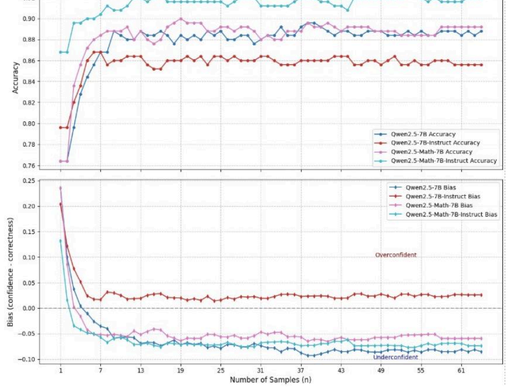
- Modern LLMs that have undergone instruction finetuning and RLHF are usually overconfident → logit values are peaked and do not represent the true uncertainty → poorly calibrated
- How to better calibrate models by adapting temperature at each generation step?
- Does better calibration help self-consistency by majority voting over diverse reasoning paths?



$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)|$$

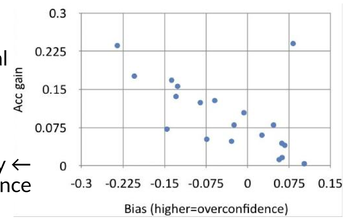
Overconfidence can be alleviated by sampling with high temperatures. However, there are trade-offs of worsening performance at high temperatures and much larger number of samples needed



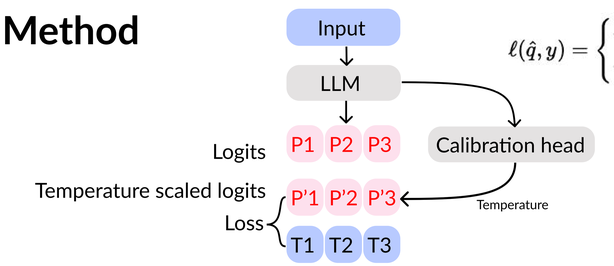


→ Instruct model is the most overconfident, and reaches peak performance at 7 samples. Additional samples cannot be utilized

Accuracy gain from self-consistency ← decreases with overconfidence



Method



$$\ell(\hat{q}, y) = \begin{cases} -(1-\alpha)\log(\sigma_{SM}(\hat{q})_y) & \text{if } \arg\max(\hat{q}) = y \quad \text{(correct)} \\ -\frac{\alpha}{|V|}\sum_{i=1}^{|V|}\log(\sigma_{SM}(\hat{q})_i) & \text{if } \arg\max(\hat{q}) \neq y \quad \text{(wrong)} \end{cases}$$

- Correct Prediction: regular cross-entropy - low T
- Incorrect Prediction: encourages the softmax distribution to be closer to uniform - high T

Future work:

- Evaluate performance of self-consistency after adaptive temperature scaling. Explore the effect of smoothing and bias required to minimize "careless mistakes" caused by sampling
- Does big-bench-mistake become easier after calibrating uncertainties?