Project update

July 15 2025

Xueyan Li

- Input sequence: $Q_1Q_2Q_3\ R_1R_2R_3\ A_1A_2$
- Logits: $P_1P_2P_3\ \ P_4P_5P_6\ \ P_7P_8$
- Augmented input sequence: $Q_1Q_2Q_3P_3\ R_1P_4R_2P_5R_3P_6\ A_1P_7A_2P_8$
- Labels: $R_1\quad R_2\quad R_3\quad A_1\quad A_2$

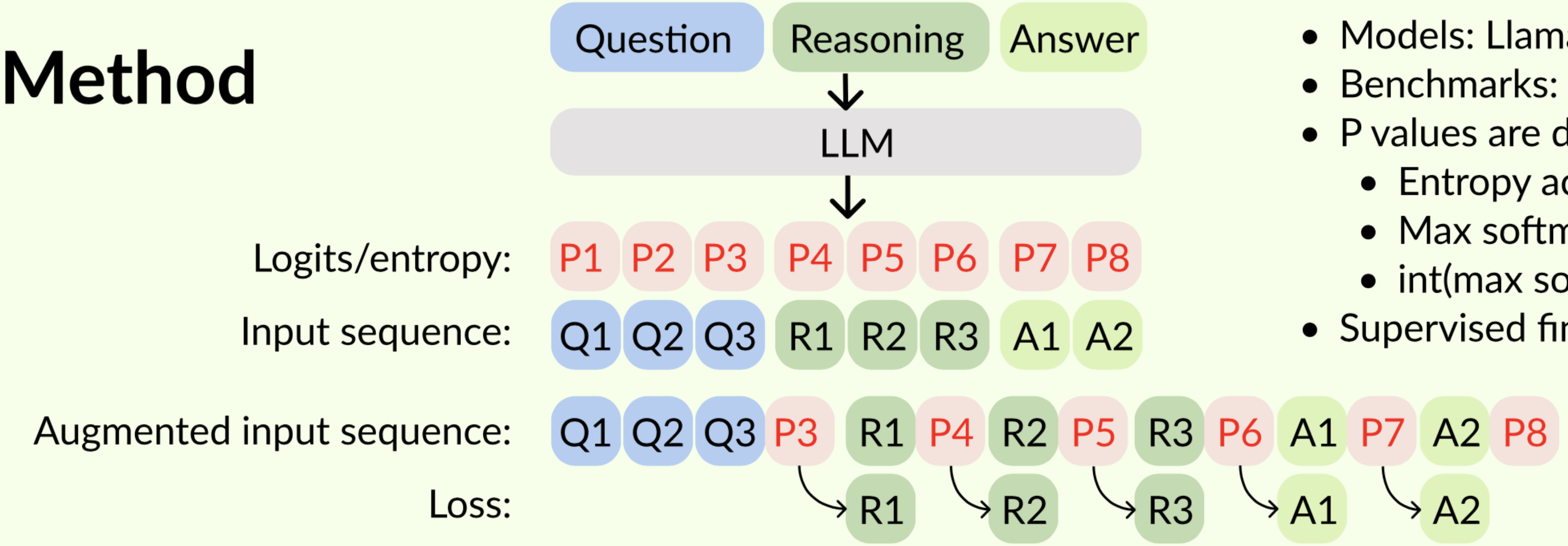
- Models: Llama3.1-1B and 8B
- Benchmarks: gsm8k, big-bench-mistake
- P are discretized entropy or max softmax values using reserved special tokens

First project – Token Level Uncertainty-Aware COT reasoning

Motivation:

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

Method


Question
Reasoning
Answer
LLM
Logits/entropy: P1 P2 P3 P4 P5 P6 P7 P8
Input sequence: Q1 Q2 Q3 R1 R2 R3 A1 A2
Augmented input sequence: Q1 Q2 Q3 P3 R1 P4 R2 P5 R3 P6 A1 P7 A2 P8
Loss: R1 R2 R3 A1 A2

- 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

GSM8K

Results:

- no consistent improvement in performance
- Special confidence tokens are ignored

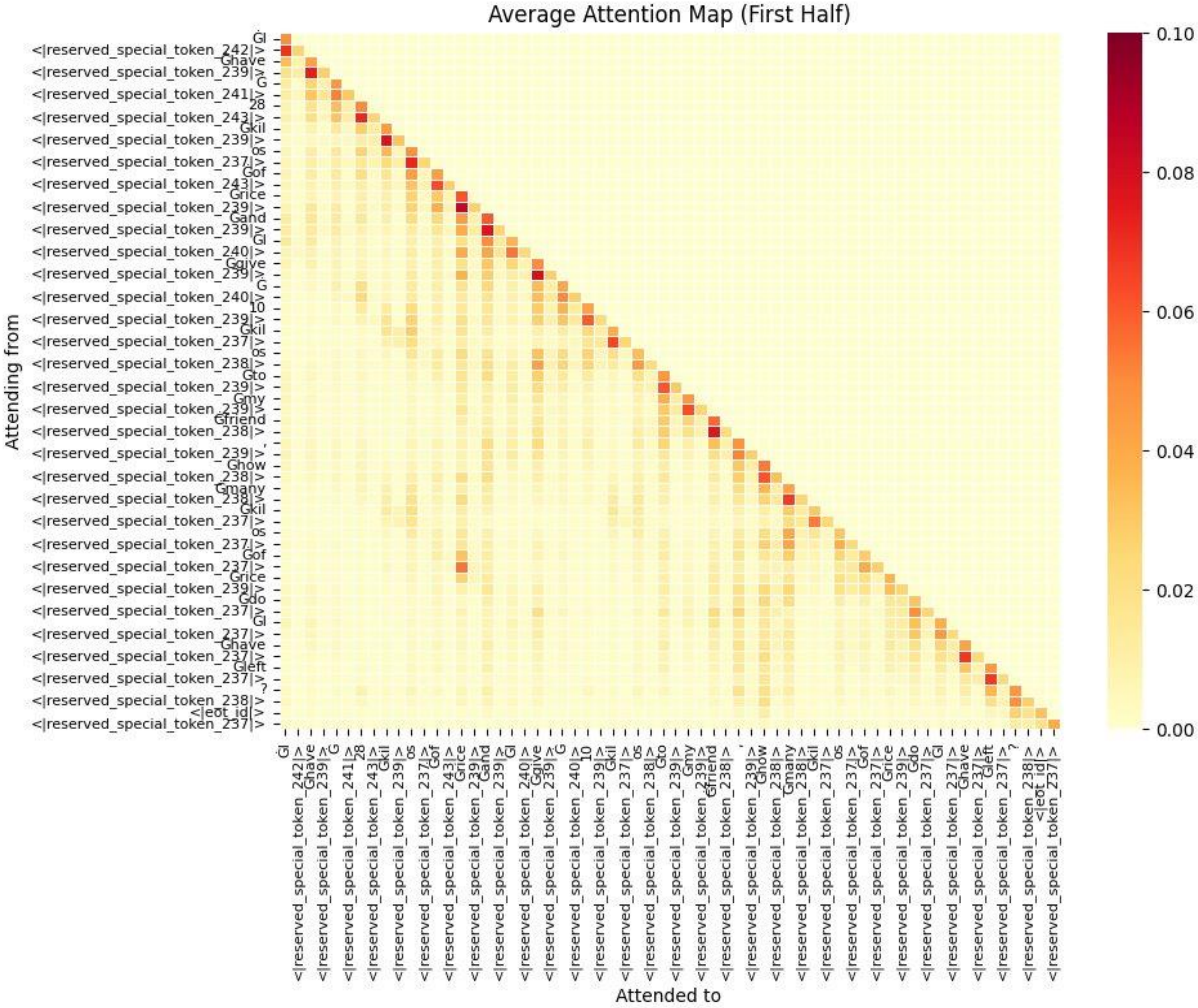
system

Cutting Knowledge Date: December 2023
Today Date: 26 Jul 2024

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

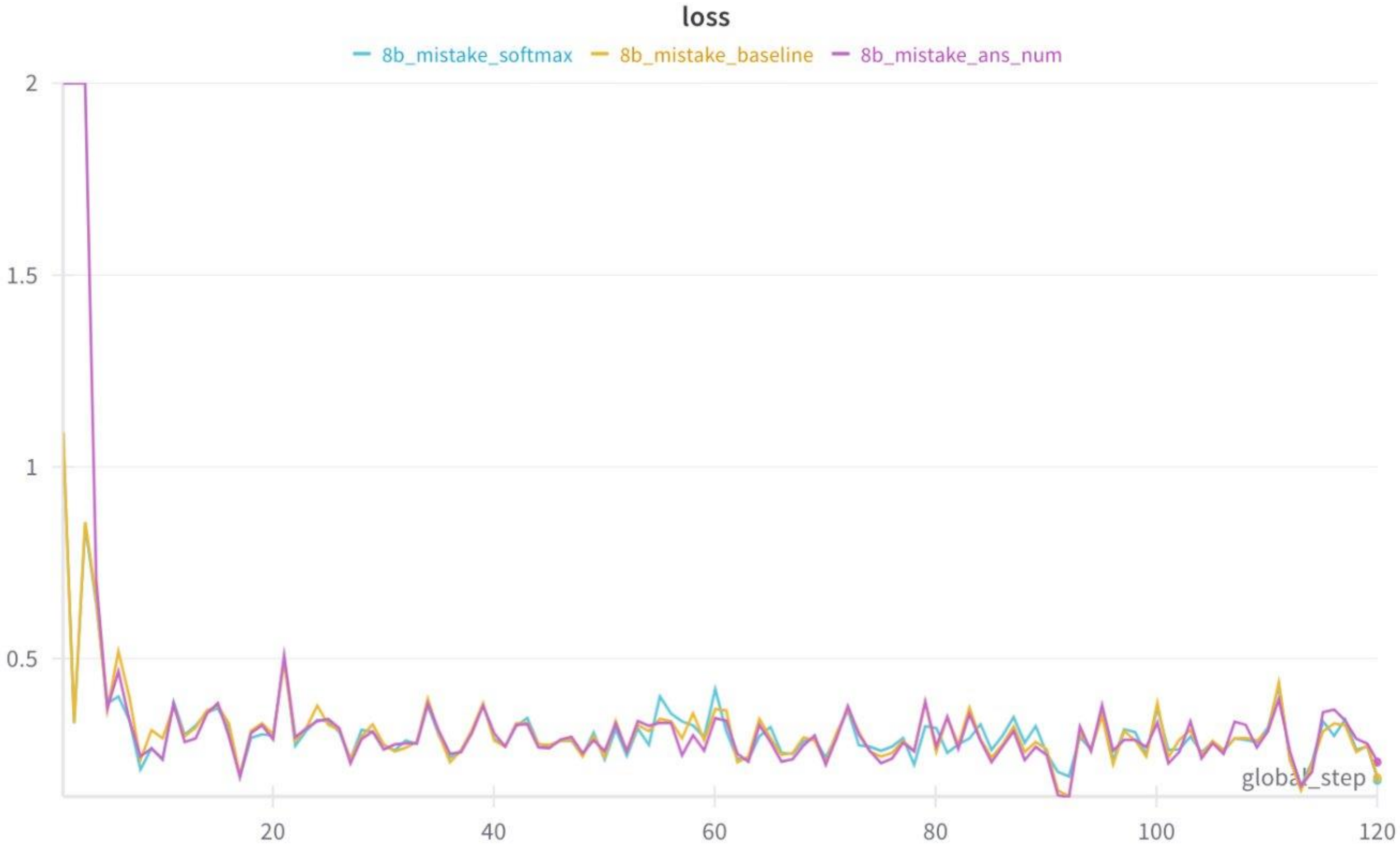
If I give 10 kilos of rice to my friend, I have 28 - 10 = <<28-10=18>>18 kilos of rice left. #### 18



Big-bench mistake

Baseline and interleaving setup have nearly identical loss curves -> not learning anything from uncertainty values

Are these model uncertainties actually truthful?



Uncertinaty-Aware Temperature Adaptation

Starting motivation:

- Modern LLMs are overconfident -> logits values are not truthful to the actual uncertainty/correctnes
- Can we improve self-consistency by improving calibration?
- Calibrating large language models with sample consistency

Preliminary results:

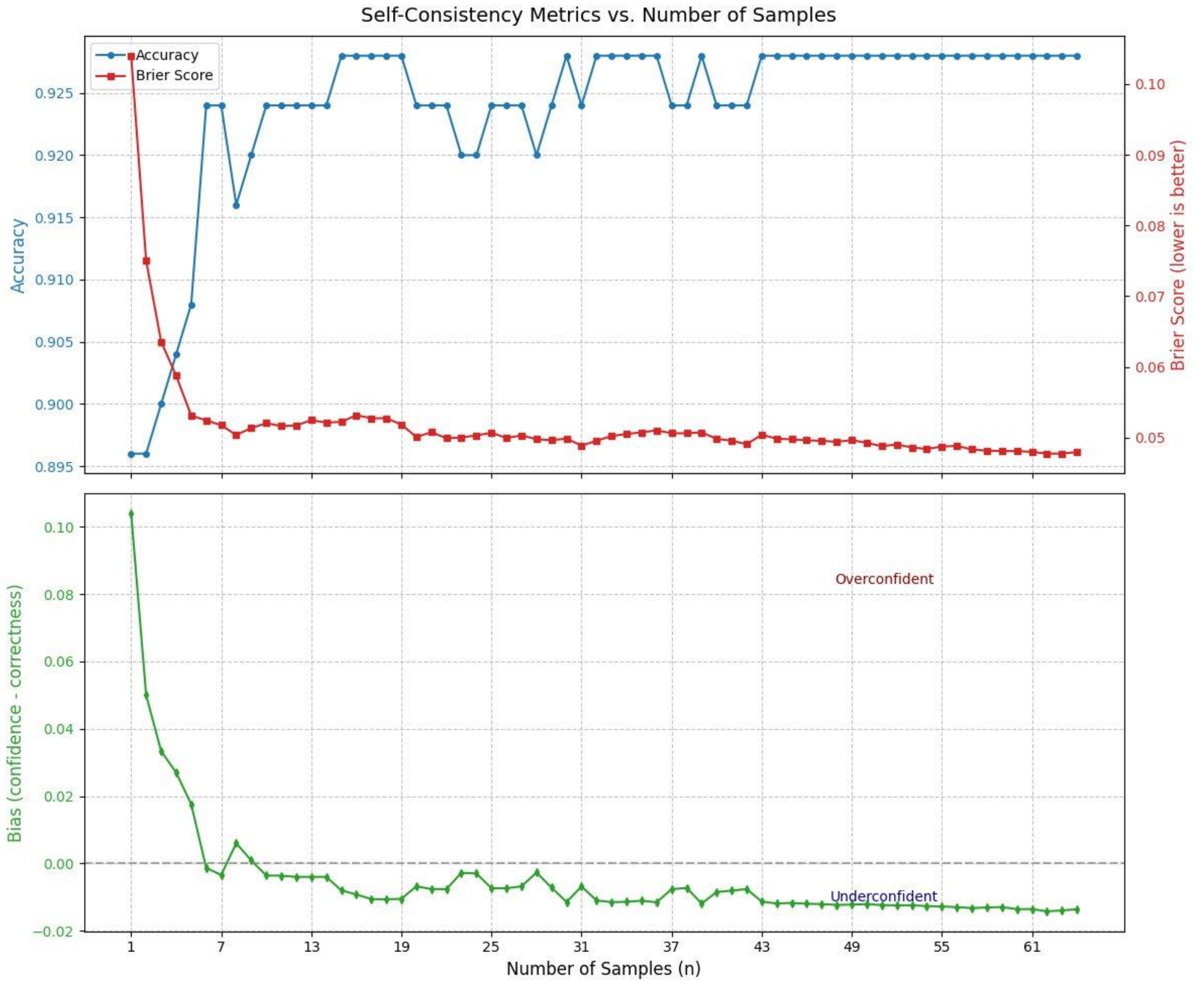
- Simple temperature scaling can improve calibration but does not necessarily lead to better self-consistency

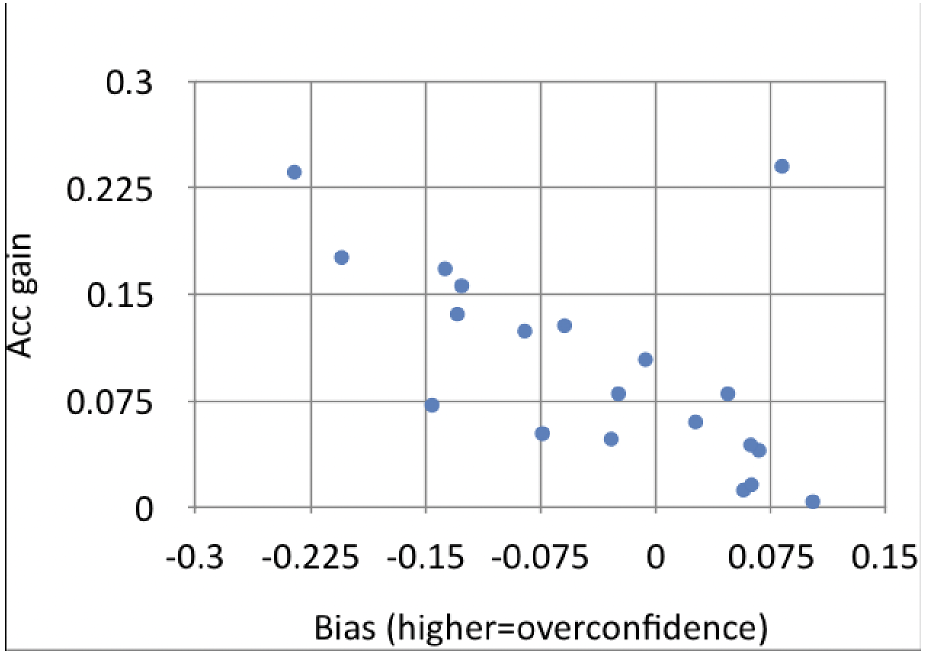
Uncertainty quantification by majority voting

- Gap in how calibration affects self-consistency performance
  - Lots of studies on how to improve calibration with temperature/token probabilities.
  - No direct link between calibration -> self-consistency
- Hypothesis: overconfident models reduce the effectiveness of self-consistency methods

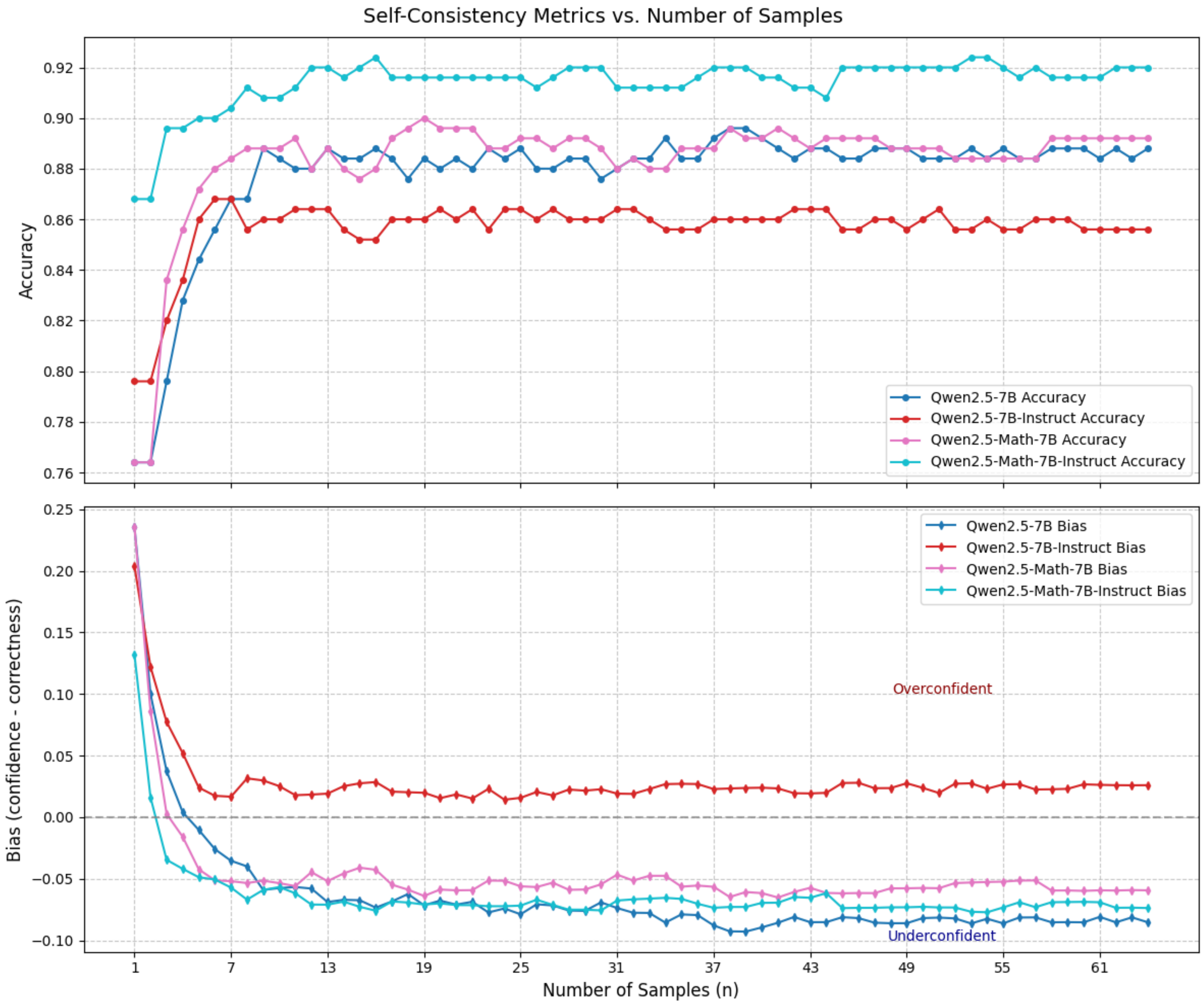
$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|$$

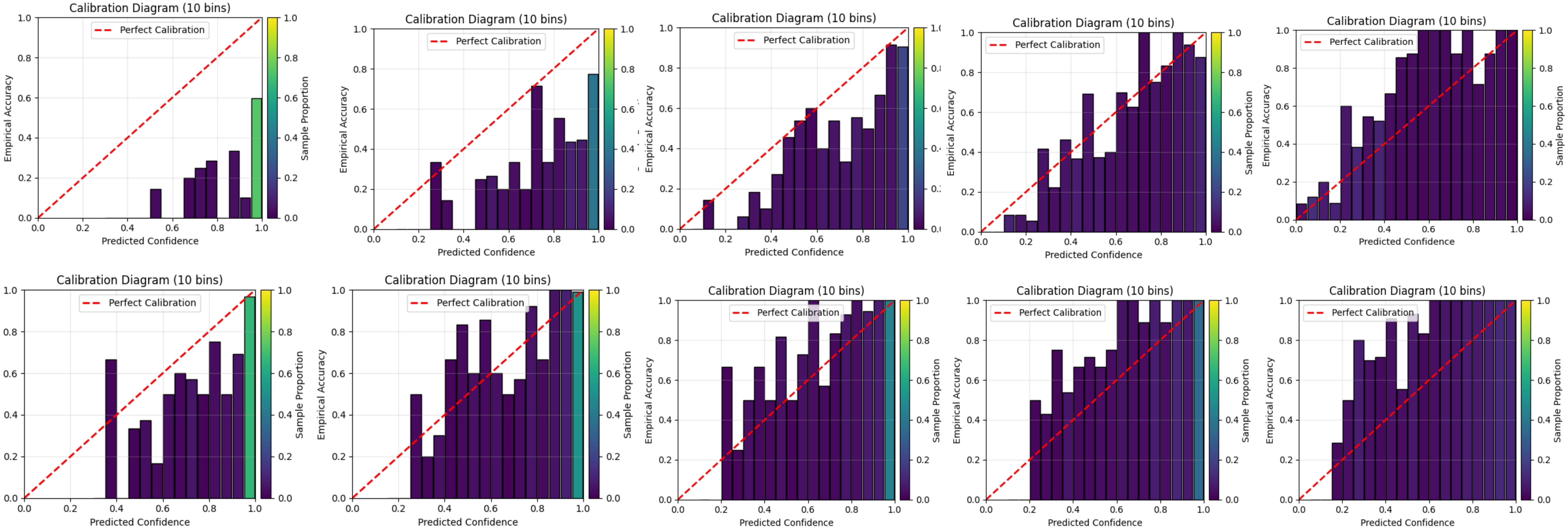
$$BS = \frac{1}{N} \sum_{i=1}^{N} \left(\mathrm{conf}(x_j, \hat{y}_j) - \mathbb{I}(\hat{y}_j = y_j)\right)^2$$

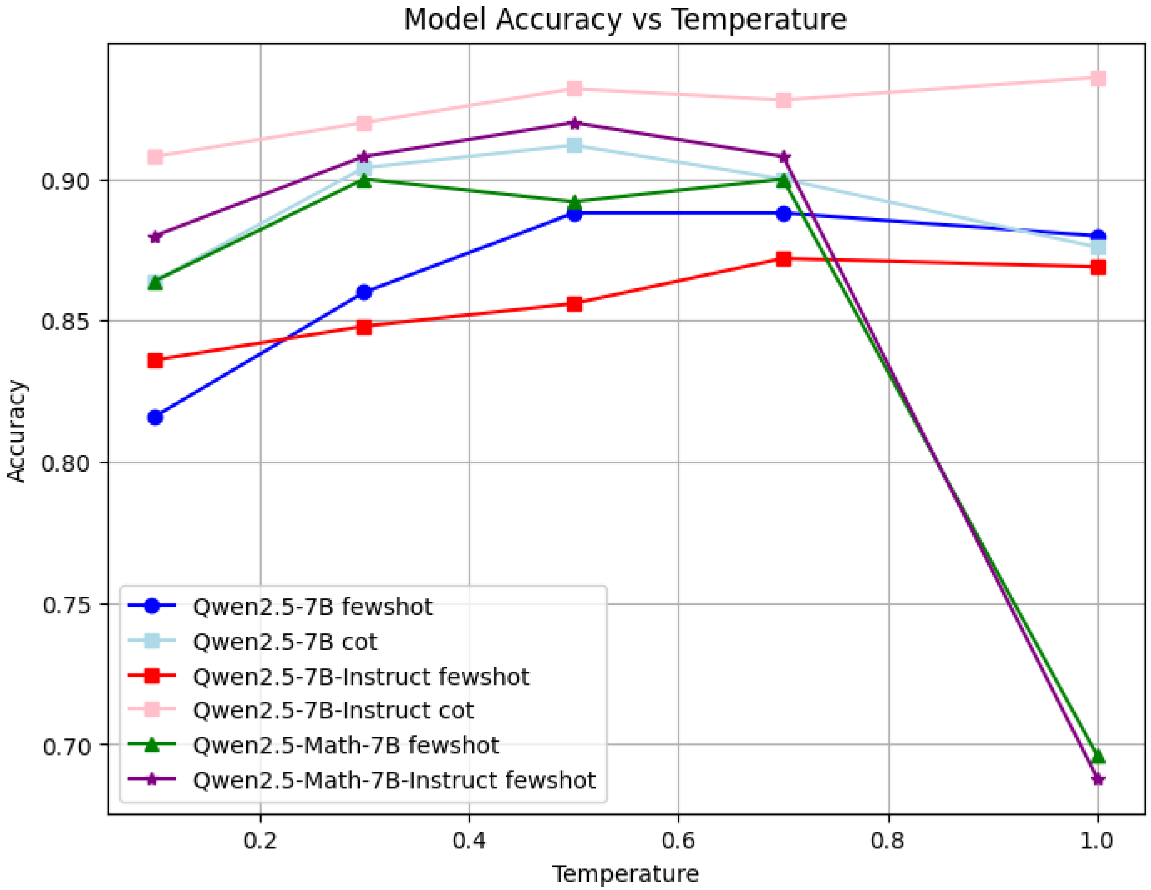


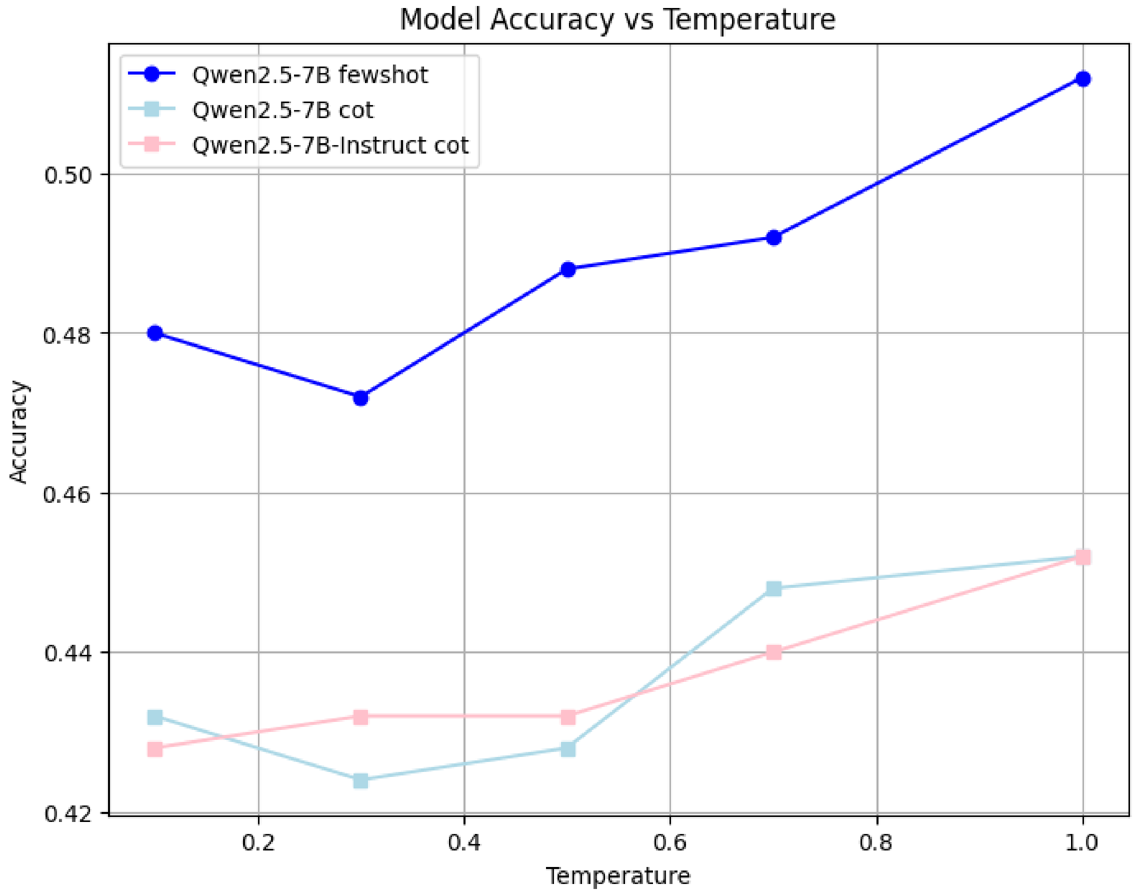


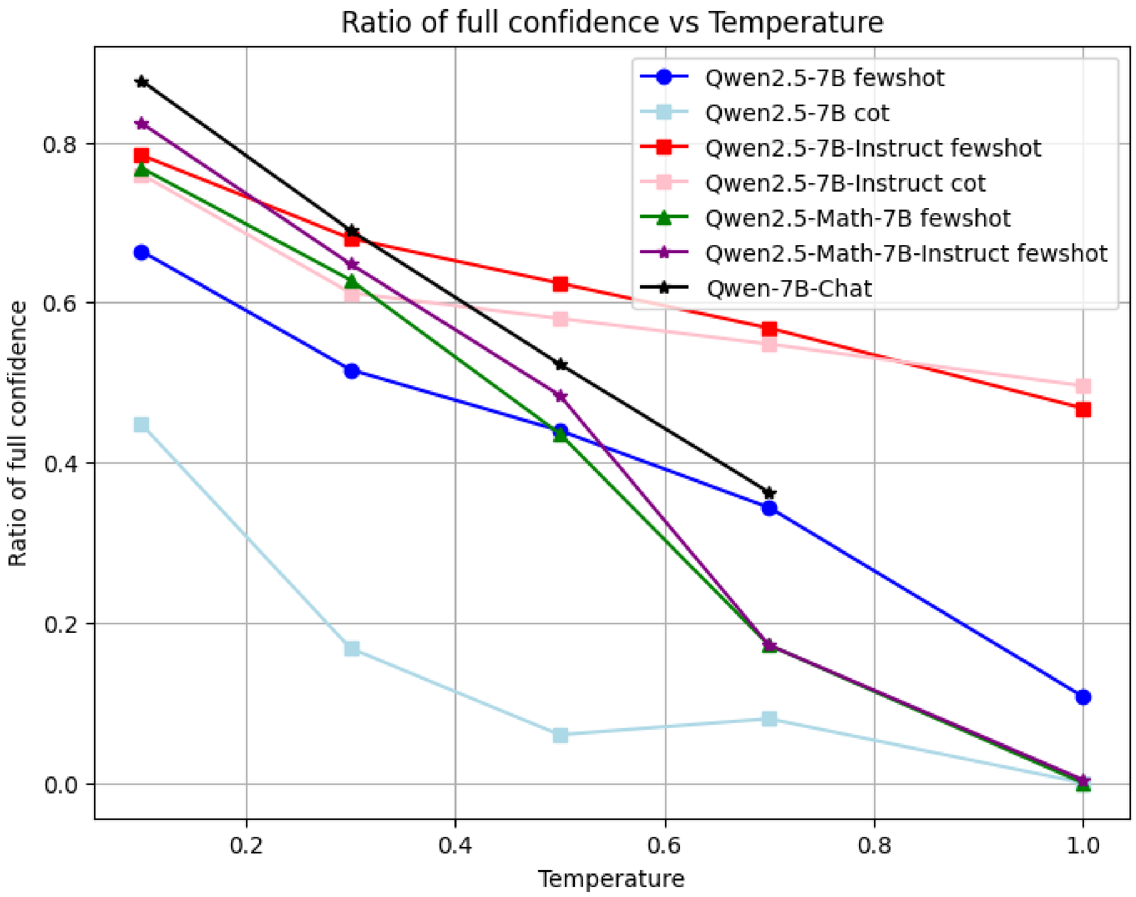
Overconfidence does reduce the benefit from self-consistency, if you ignore greedy performance

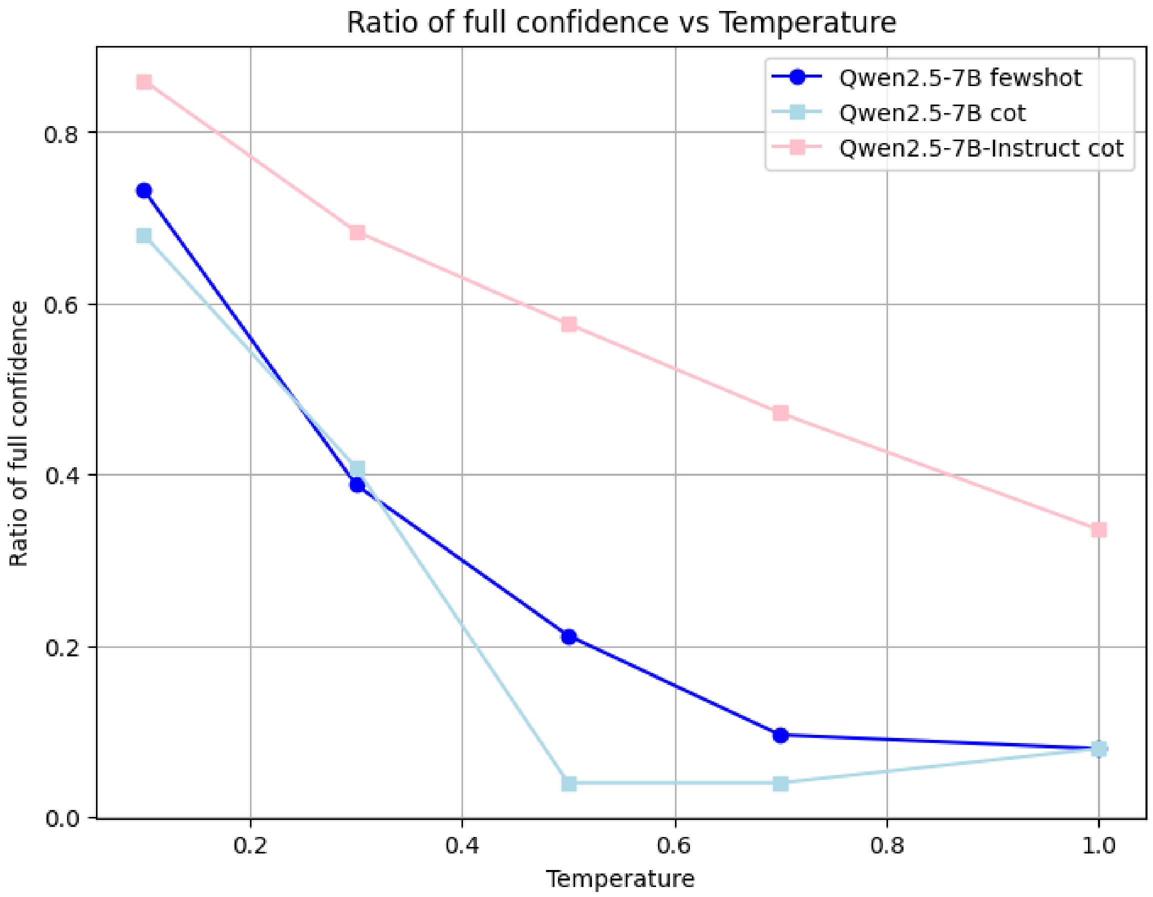



Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Sample Proportion
0.0
0.2
0.4
0.6
0.8
1.0
Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Sample Proportion
Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Sample Proportion
Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Sample Proportion
Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Sample Proportion
Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Sample Proportion
Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Sample Proportion
Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Sample Proportion
Calibration Diagram (10 bins)
Perfect Calibration
Empirical Accuracy
Predicted Confidence
Sample Proportion


Model Accuracy vs Temperature
Accuracy
Temperature
0.70
0.75
0.80
0.85
0.90
0.2
0.4
0.6
0.8
1.0
Qwen2.5-7B fewshot
Qwen2.5-7B cot
Qwen2.5-7B-Instruct fewshot
Qwen2.5-7B-Instruct cot
Qwen2.5-Math-7B fewshot
Qwen2.5-Math-7B-Instruct fewshot


Model Accuracy vs Temperature
Accuracy
Temperature
0.42
0.44
0.46
0.48
0.50
0.2
0.4
0.6
0.8
1.0
Qwen2.5-7B fewshot
Qwen2.5-7B cot
Qwen2.5-7B-Instruct cot

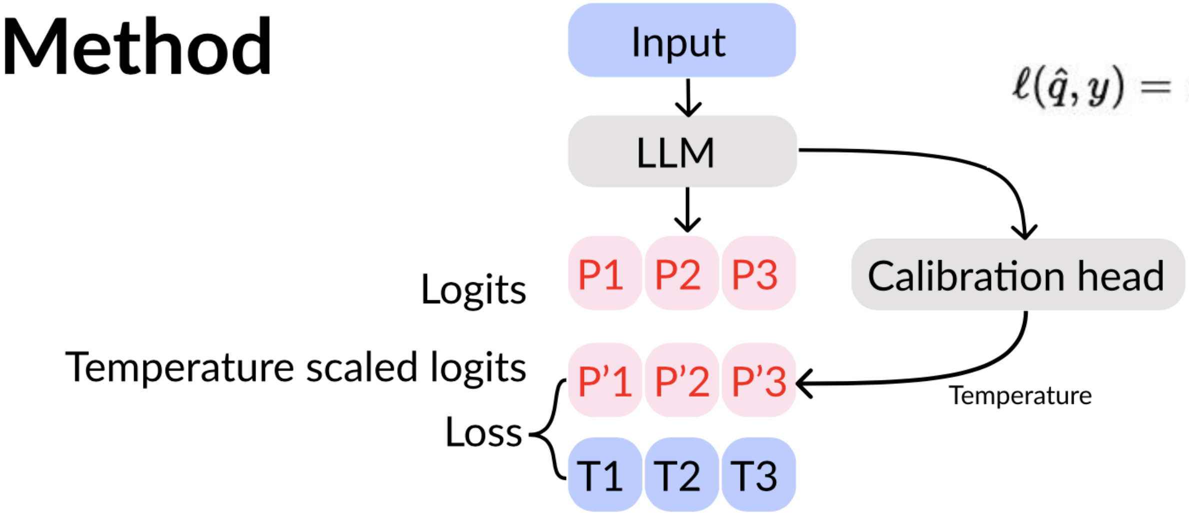

Ratio of full confidence vs Temperature
Ratio of full confidence
Temperature
0.0
0.2
0.4
0.6
0.8
0.2
0.4
0.6
0.8
1.0
Qwen2.5-7B fewshot
Qwen2.5-7B cot
Qwen2.5-7B-Instruct fewshot
Qwen2.5-7B-Instruct cot
Qwen2.5-Math-7B fewshot
Qwen2.5-Math-7B-Instruct fewshot
Qwen-7B-Chat


Ratio of full confidence vs Temperature
Ratio of full confidence
Temperature
0.0
0.2
0.4
0.6
0.8
0.2
0.4
0.6
0.8
1.0
Qwen2.5-7B fewshot
Qwen2.5-7B cot
Qwen2.5-7B-Instruct cot

Token-Wise Adaptive Temperature

- Calibrating Language Models with Adaptive Temperature Scaling
  - Flawed implementation: no cross-attention, wrong temperature calculation, only short form generation

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

	Qwen-7B-Chat	Llama-2-7b-chat-hf
Baseline greedy	0.512	0.236
Baseline sampled T=1.0	0.584	0.280
Baseline sampled T=1.3	0.608	0.260
Adaptive temperature tuned	**0.656**	**0.388**

Training dataset: alpaca
Eval dataset: GSM8K first 250 questions
Number of samples: 80
All have very bad calibration, severely under-confident

Current work: scale to other models outside current code-base

- Related works:
- A Head to Predict and a Head to Question: Pre-trained Uncertainty Quantification Heads for Hallucination Detection in LLM Outputs
- Uncertainty-Aware Attention Heads: Efficient Unsupervised Uncertainty Quantification for LLMs