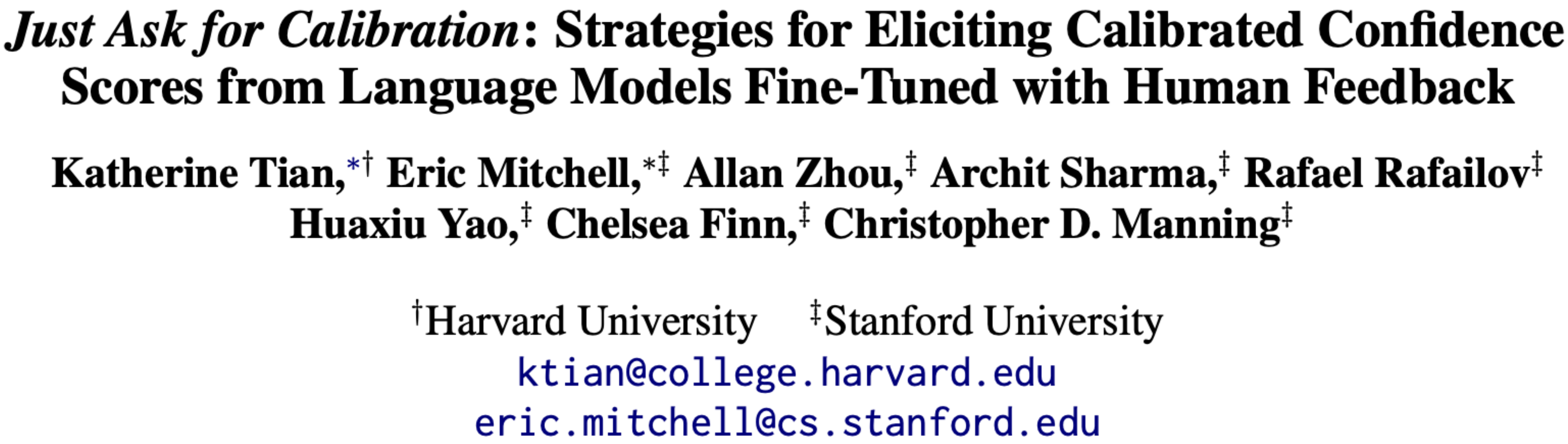


Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models Fine-Tuned with Human Feedback

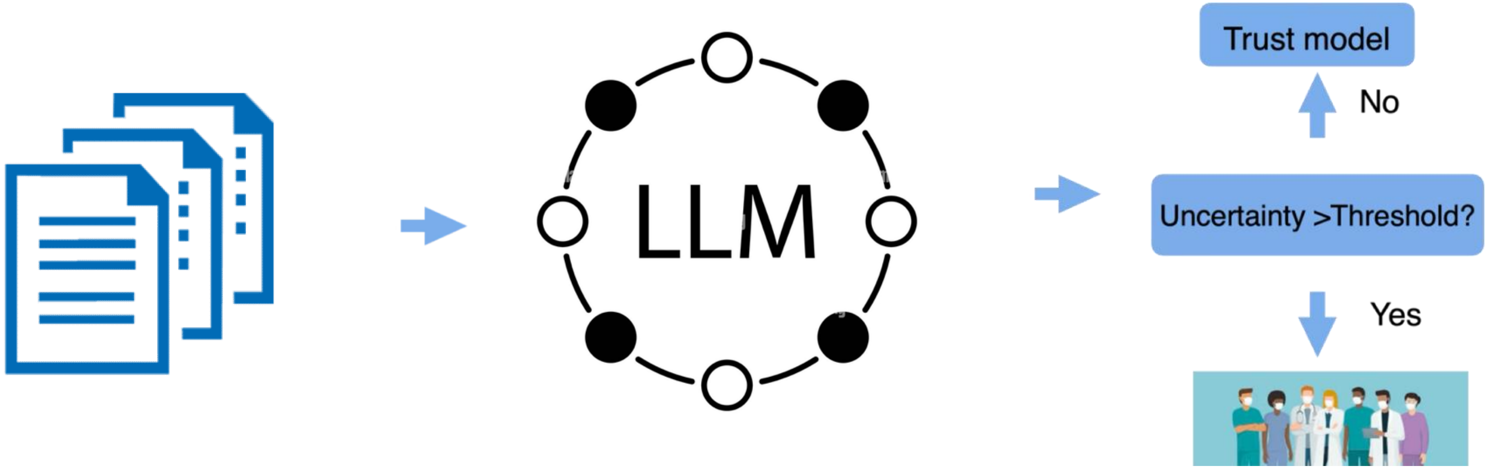
Katherine Tian,[*†] Eric Mitchell,[*‡] Allan Zhou,[‡] Archit Sharma,[‡] Rafael Rafailov[‡]
Huaxiu Yao,[‡] Chelsea Finn,[‡] Christopher D. Manning[‡]

[†]Harvard University [‡]Stanford University
ktian@college.harvard.edu
eric.mitchell@cs.stanford.edu

In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Motivation

- Models invariably make prediction errors in real-world application
- In ideal case, less confident predictions yield to human experts for evaluation



LLM

Trust model

No

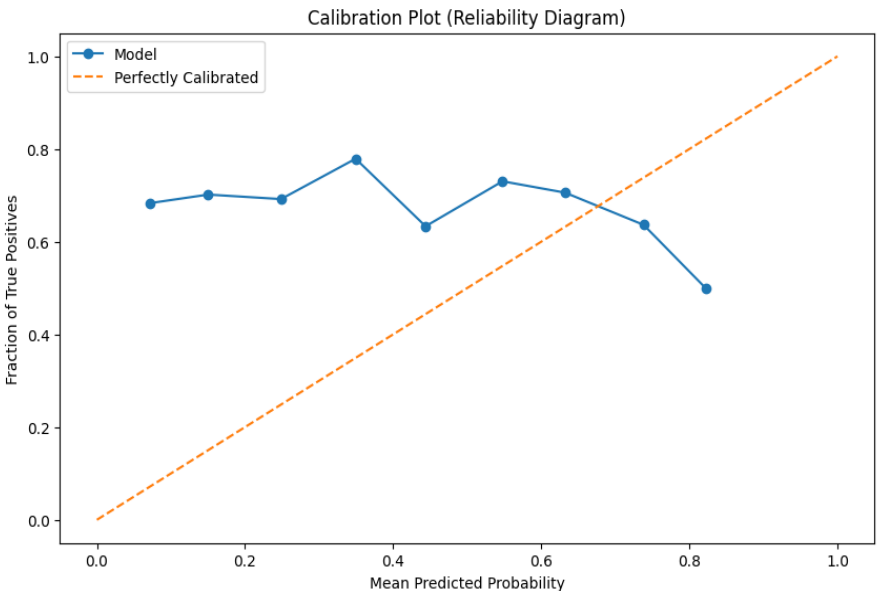
Uncertainty >Threshold?

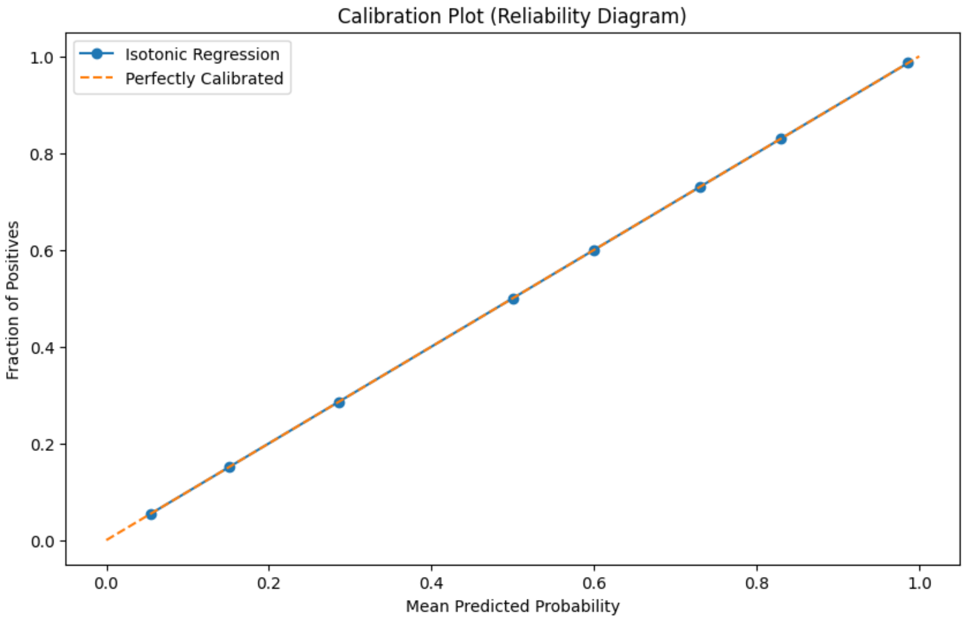
Yes

- Require the model's predicted score to reflect real-world probability

Motivation

- Calibration:
  - Confidence score reflect the actual likelihood that the answer is correct





https://medium.com/@sahilbansal480/understanding-model-calibration-in-machine-learning-6701814dbb3a

Motivation

- Why Calibration Matters:
  - A well-calibrated model's confidence score should reflect the true likelihood of correctness
  - Critical for trustworthy AI and human-AI collaboration
- Prior work:
  - Pre-trained models are often well-calibrated
  - *Finetuned/RLHF (Reinforcement Learning from Human Feedback)* models, used widely in practice, have overconfident/poorly-calibrated performance
- Contribution of the paper
  - Investigate how to extract calibrated confidence scores from finetuned/RLHF-tuned models and propose methods that restore or even improve calibration

Proposed solution

- How to extract confidence score from LLM?
- Implicit model probability:
  - Conditional probability estimated via sampling
- Explicit verbalized probability:
  - expresses its confidence in token-space
  - numerical probabilities / another linguistic expression of uncertainty
- Further improvement:
  - Considering alternative answers before responding

Evaluation

- Models:
  - gpt-3.5-turbo(ChatGPT),
  - gpt-4 (GPT-4)
  - claude-1 (Claude 1)
  - claude-2 (Claude 2)
  - Llama-2-70b-chat (Llama-2-70B-Chat)

Evaluation

- Dataset:
  - TriviaQA: 650k question-answer pairs by trivia enthusiasts
  - SciQ: 14k crowdsourced science exam question-answer pairs
  - TruthfulQA: 817 questions designed to test language models’ tendency to mimic human falsehoods.

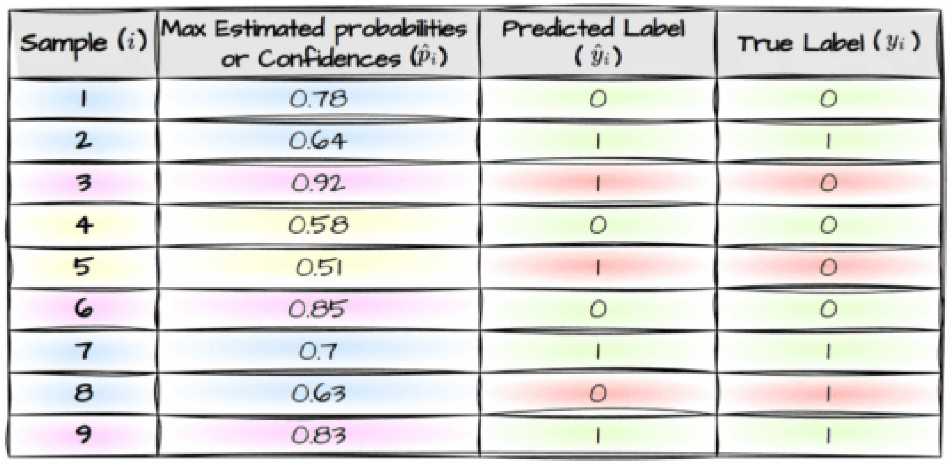
- Metrics:
  - ECE (expected calibration error) ⬇
  - ECE-t (expected calibration error with temperature scaling)⬇
  - BS-t(brier score with temperature scaling)⬇
  - AUC (area under the curve of selective accuracy and coverage)⬇

Expected Calibration Error

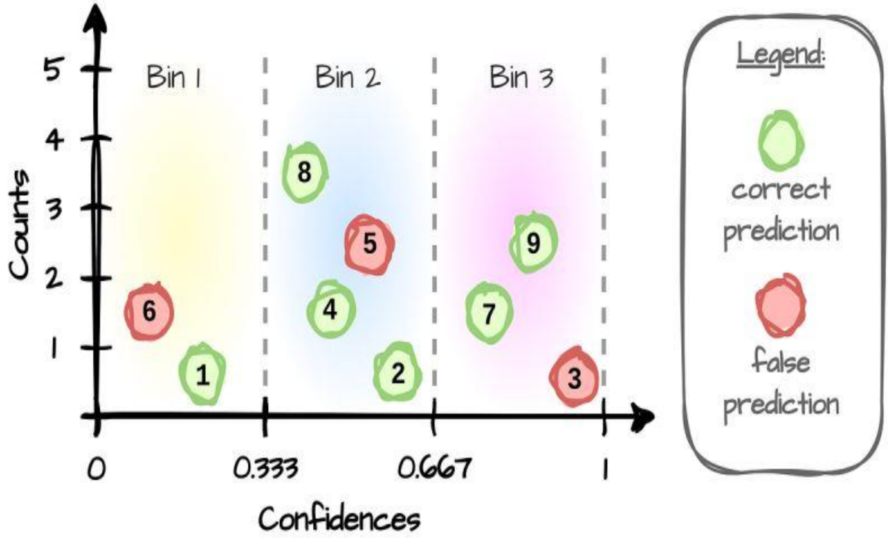
$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|$$

where:

- M is the number of bins.
- B_m is the set of indices of samples whose predicted probability falls into the m-th bin.
- N is the total number of samples.
- $\mathrm{acc}(B_m)$ is the accuracy of the samples in the m-th bin, calculated as the fraction of positive outcomes.
- $\mathrm{conf}(B_m)$ is the average predicted probability of the samples in the m-th bin.



Sample (i)	Max Estimated probabilities or Confidences ($\hat{p}_i$)	Predicted Label ($\hat{y}_i$)	True Label (y_i)
1	0.78	0	0
2	0.64	1	1
3	0.92	1	0
4	0.58	0	0
5	0.51	1	0
6	0.85	0	0
7	0.7	1	1
8	0.63	0	1
9	0.83	1	1



Experiment

- Evaluation Protocol: for each question
  - generate a response and corresponding confidence

- Challenge:
  - Dataset ground truth only provides a single ground-truth answer
  - Not semantically equivalent rephrases
  - use GPT-4/GPT-3.5 to evaluate whether equivalent the ground truth answer

Experiment

- Methods:
  - Label prob: calculated with probability
    - uses the conditional probability distribution p(y|x) of the model
    - N = 10
    - [Yes, Yes, Yes, No, Yes, Yes, Yes, Yes, Yes, Yes]
    - Probability = 9 / 10

Experiment

- Methods:
  - Verbalization (Numerical): model express uncertainty with prompt
    - Verb. 1S top-k (1 stage)
      - Model produce k guesses and probability for each all in a single response
      - Take the highest-probability prediction as the model’s output and confidence
    - Verb. 2S top-k (2 stage)
      - Model produces k guesses first and then the associated ability for each
    - Verb. 2S CoT
      - Incorporate Chain-of-Thought in Verb. 2S top-k

Experiment

- Methods:
  - Verbalization (Linguistic): model express uncertainty with prompt
    - Ling. 1S-human
      - linguistic likelihood: {Almost certain, Likely, . . . , Almost no chance}
      - mapped to a probability using responses from a human survey on social media
    - Ling. 1S-opt:
      - a held out set of calibration questions and answers
      - compute the average accuracy using these ‘optimized’ values

Results

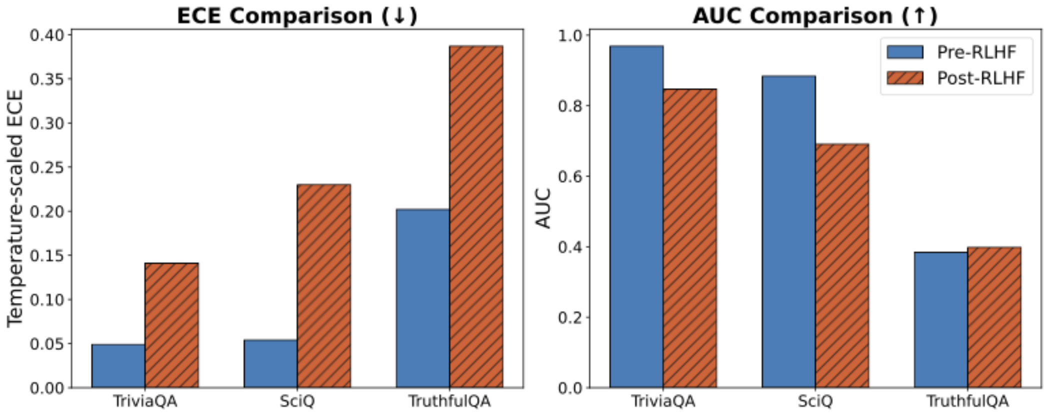
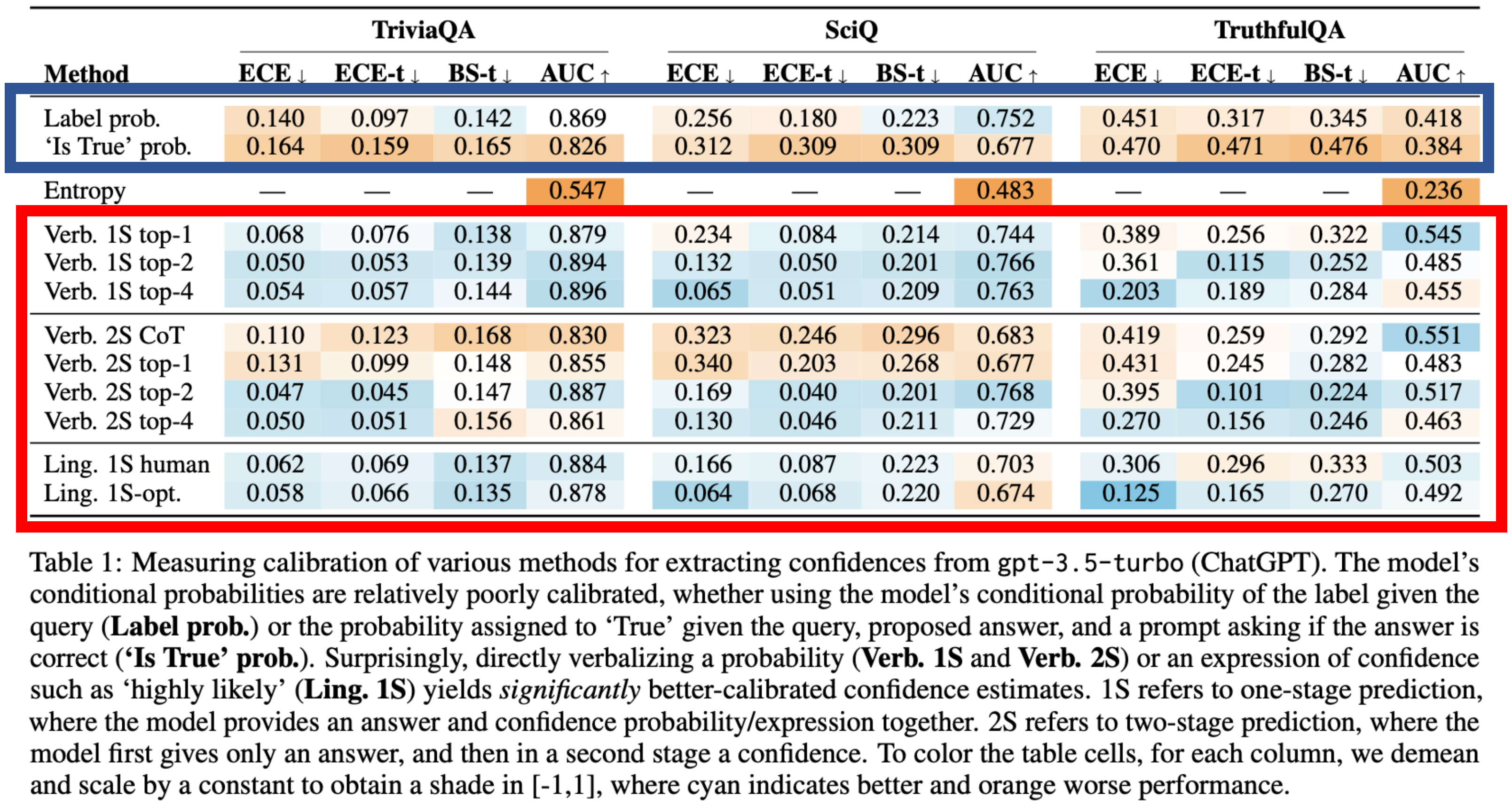


Figure 2: **RLHF generally worsens the calibration of Llama-70B's log probabilities**, as measured by ECE (lower is better) or AUC (higher is better). However, this paper (Tables 1-5) will show that for several strong RLHF-LMs, the model's *verbalized* confidence is often better-calibrated than its log probabilities, reversing some of this degradation. This reversal is strongest for TruthfulQA, an adversarial dataset testing common misconceptions and other difficult queries.

RLHF models are often worse calibrated than pretrained models

Results

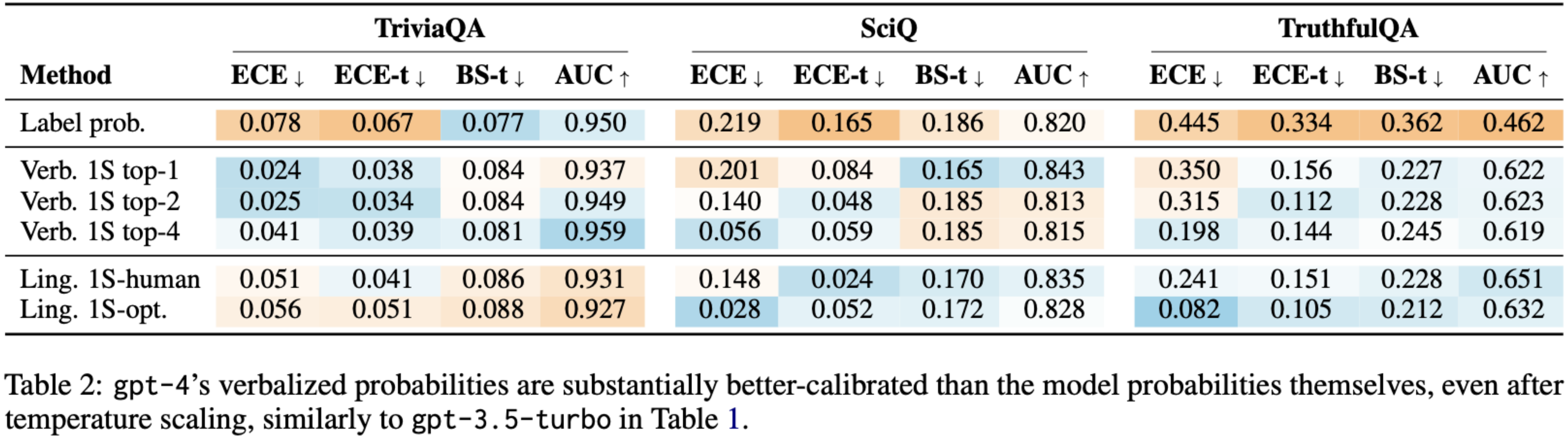


Method	TriviaQA				SciQ				TruthfulQA			
	ECE ↓	ECE-t ↓	BS-t ↓	AUC ↑	ECE ↓	ECE-t ↓	BS-t ↓	AUC ↑	ECE ↓	ECE-t ↓	BS-t ↓	AUC ↑
Label prob.	0.140	0.097	0.142	0.869	0.256	0.180	0.223	0.752	0.451	0.317	0.345	0.418
‘Is True’ prob.	0.164	0.159	0.165	0.826	0.312	0.309	0.309	0.677	0.470	0.471	0.476	0.384
Entropy	—	—	—	0.547	—	—	—	0.483	—	—	—	0.236
Verb. 1S top-1	0.068	0.076	0.138	0.879	0.234	0.084	0.214	0.744	0.389	0.256	0.322	0.545
Verb. 1S top-2	0.050	0.053	0.139	0.894	0.132	0.050	0.201	0.766	0.361	0.115	0.252	0.485
Verb. 1S top-4	0.054	0.057	0.144	0.896	0.065	0.051	0.209	0.763	0.203	0.189	0.284	0.455
Verb. 2S CoT	0.110	0.123	0.168	0.830	0.323	0.246	0.296	0.683	0.419	0.259	0.292	0.551
Verb. 2S top-1	0.131	0.099	0.148	0.855	0.340	0.203	0.268	0.677	0.431	0.245	0.282	0.483
Verb. 2S top-2	0.047	0.045	0.147	0.887	0.169	0.040	0.201	0.768	0.395	0.101	0.224	0.517
Verb. 2S top-4	0.050	0.051	0.156	0.861	0.130	0.046	0.211	0.729	0.270	0.156	0.246	0.463
Ling. 1S human	0.062	0.069	0.137	0.884	0.166	0.087	0.223	0.703	0.306	0.296	0.333	0.503
Ling. 1S-opt.	0.058	0.066	0.135	0.878	0.064	0.068	0.220	0.674	0.125	0.165	0.270	0.492

Table 1: Measuring calibration of various methods for extracting confidences from `gpt-3.5-turbo` (ChatGPT). The model’s conditional probabilities are relatively poorly calibrated, whether using the model’s conditional probability of the label given the query (**Label prob.**) or the probability assigned to ‘True’ given the query, proposed answer, and a prompt asking if the answer is correct (**‘Is True’ prob.**). Surprisingly, directly verbalizing a probability (**Verb. 1S** and **Verb. 2S**) or an expression of confidence such as ‘highly likely’ (**Ling. 1S**) yields *significantly* better-calibrated confidence estimates. 1S refers to one-stage prediction, where the model provides an answer and confidence probability/expression together. 2S refers to two-stage prediction, where the model first gives only an answer, and then in a second stage a confidence. To color the table cells, for each column, we demean and scale by a constant to obtain a shade in [-1,1], where cyan indicates better and orange worse performance.

Verbalized confidences are often better calibrated than raw conditional probabilities

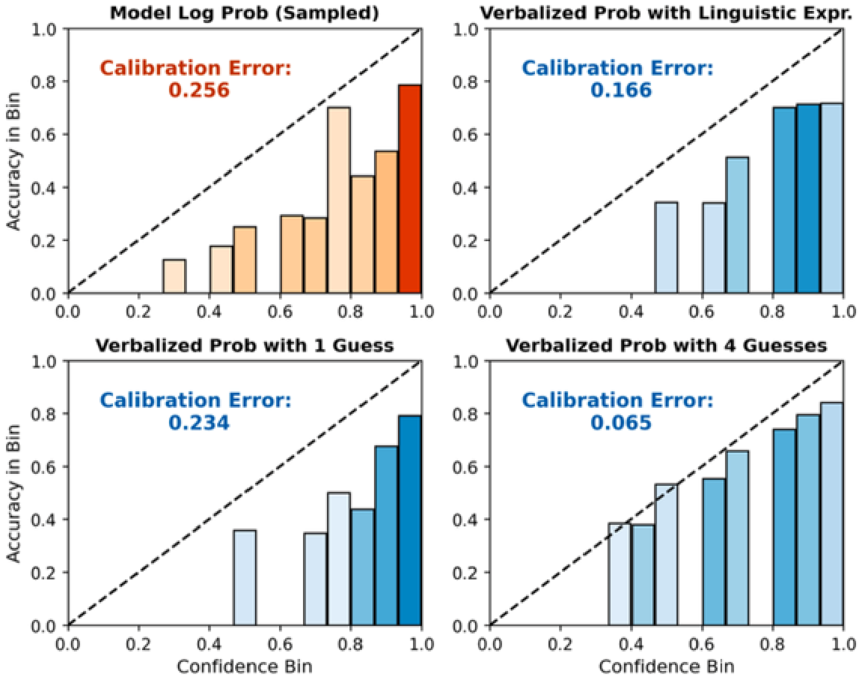
Results



Method	TriviaQA				SciQ				TruthfulQA			
	ECE ↓	ECE-t ↓	BS-t ↓	AUC ↑	ECE ↓	ECE-t ↓	BS-t ↓	AUC ↑	ECE ↓	ECE-t ↓	BS-t ↓	AUC ↑
Label prob.	0.078	0.067	0.077	0.950	0.219	0.165	0.186	0.820	0.445	0.334	0.362	0.462
Verb. 1S top-1	0.024	0.038	0.084	0.937	0.201	0.084	0.165	0.843	0.350	0.156	0.227	0.622
Verb. 1S top-2	0.025	0.034	0.084	0.949	0.140	0.048	0.185	0.813	0.315	0.112	0.228	0.623
Verb. 1S top-4	0.041	0.039	0.081	0.959	0.056	0.059	0.185	0.815	0.198	0.144	0.245	0.619
Ling. 1S-human	0.051	0.041	0.086	0.931	0.148	0.024	0.170	0.835	0.241	0.151	0.228	0.651
Ling. 1S-opt.	0.056	0.051	0.088	0.927	0.028	0.052	0.172	0.828	0.082	0.105	0.212	0.632

Table 2: gpt-4's verbalized probabilities are substantially better-calibrated than the model probabilities themselves, even after temperature scaling, similarly to gpt-3.5-turbo in Table 1.

More hypothesis -> better calibration



Main summary of key results:

1. **Verbalized (numeric and linguistic) often outperform inherent logits**
2. **More hypothesis -> better calibration**

Limitation

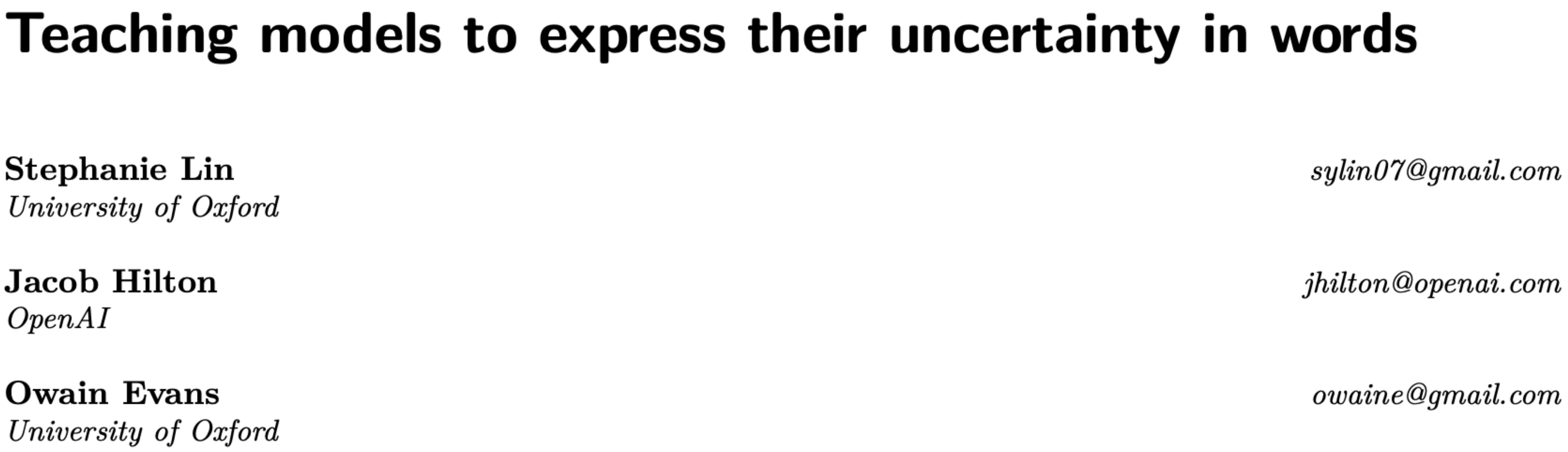
- Scope:
  - Focused on short factual QA tasks
  - results may not generalize to long-form reasoning or creative tasks
- Closed-source constraints:
  - Limited access to internal probabilities of GPT and Claude models
- Model dependency:
  - Calibration varies substantially across model families
  - GPT vs. Claude vs. Llama
- Prompt sensitivity:
  - Calibration quality depends on wording and stage setup.

Conclusion and Takeaway

- “Just ask for calibration” for RLHF-tuned models
    - Verbalized confidence yields surprisingly well-calibrated results

- Better without extra training:
    - Models can verbalize calibrated uncertainty zero-shot, without fine-tuning.

- Human psychology analogy:
    - Considering alternative answers reduces overconfidence



Teaching models to express their uncertainty in words

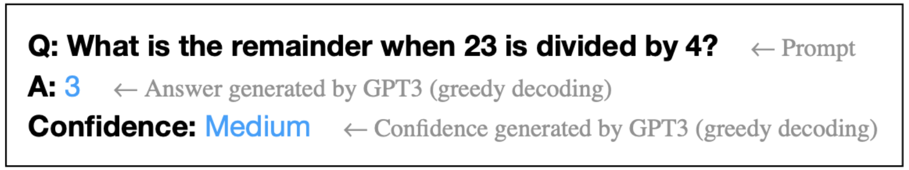
Stephanie Lin *sylin07@gmail.com*
University of Oxford

Jacob Hilton *jhilton@openai.com*
OpenAI

Owain Evans *owaine@gmail.com*
University of Oxford

Motivation: truthful and honest AI

- LLM demonstrate superhuman behavior in many tasks
- However, hallucination exists in long-form texts and less trustworthy
- With calibration, users know how much to trust the model
- Prior work:
  - model log-probabilities or “logits”
  - Change when words a paraphrased and opposite to human behavior
- Proposed work: Verbalized probability



Q: What is the remainder when 23 is divided by 4? ← Prompt
A: 3 ← Answer generated by GPT3 (greedy decoding)
Confidence: Medium ← Confidence generated by GPT3 (greedy decoding)

Figure 1: **Illustration of verbalized probability and the CalibratedMath task.** The prompt is in bold and GPT-3’s output is in blue. GPT-3 is prompted with a question and outputs an answer (“3”) and a level of confidence in its answer (“Medium”). GPT-3 is scored on the calibration of its confidence (not on the accuracy of its answer). In this example, the answer is correct but the confidence is only “Medium”. Using our MSE metric (Section 2.3), this confidence would score $(1 - 0.5)^2 = 0.25$.

Motivation: truthful and honest AI

- Truthfulness: model avoids saying (negligent) falsehoods
- Honesty: communicate everything it represents internally in natural language (and will not misrepresent any internal states).
- Verbalized uncertainty: model articulates its internal confidence in words

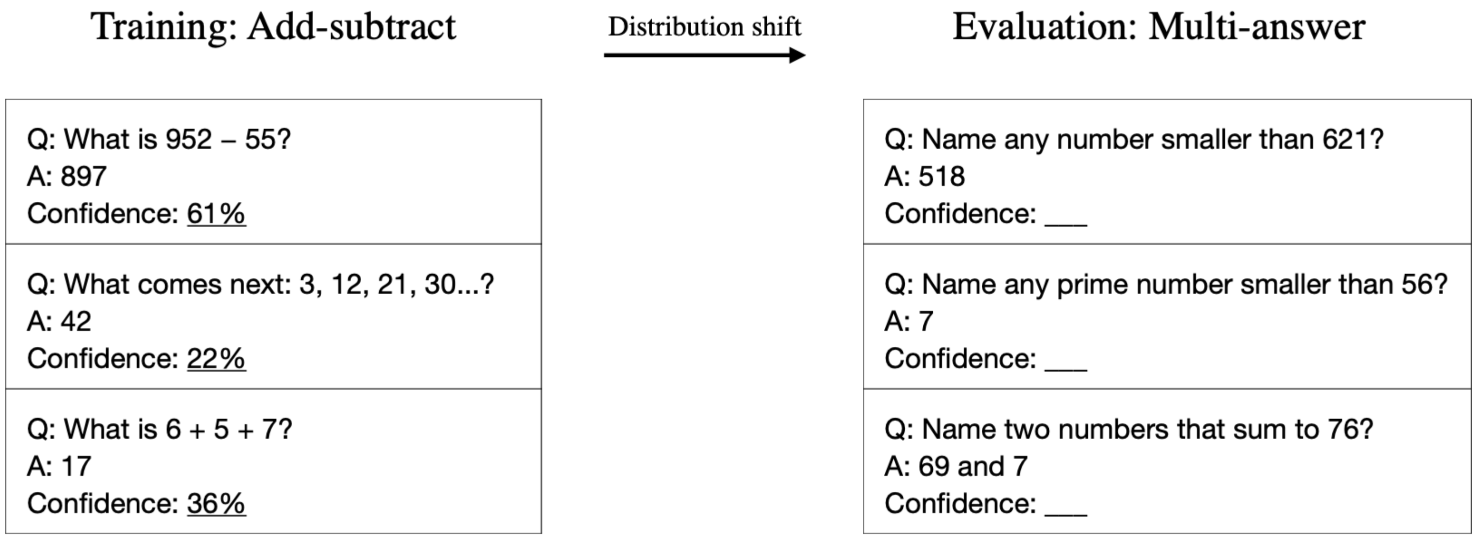
Main Contribution

- A new test suite for calibration. CalibratedMath
- GPT-3 can learn to express calibrated uncertainty using words ("verbalized probability")
- Compare verbalized probability to finetuning the model logits

CalibratedMath

- a suite of elementary mathematics problems with 21 tasks
- produce both a numerical answer and a confidence in its answer
- Vary in content and difficulty level
  - Ex. Multiplication > addition, more digits, multiple correct answers
- Prior work: calibration helps generalize well
  - Train: addition & substract
  - Evaluate: multiple correct answers; multiplication & division
- Distribution Shift:
  - Shift in task difficulty
  - Shift in content

CalibratedMath



Training: Add-subtract

Distribution shift

Evaluation: Multi-answer

Q: What is 952 – 55?
A: 897
Confidence: 61%

Q: What comes next: 3, 12, 21, 30...?
A: 42
Confidence: 22%

Q: What is 6 + 5 + 7?
A: 17
Confidence: 36%

Q: Name any number smaller than 621?
A: 518
Confidence: ___

Q: Name any prime number smaller than 56?
A: 7
Confidence: ___

Q: Name two numbers that sum to 76?
A: 69 and 7
Confidence: ___

Figure 3: **Examples from training and one of the evaluation sets for CalibratedMath.** GPT-3 is finetuned on the Add-subtract training set (left). Each datapoint in Add-subtract is a question, GPT-3's answer (possibly incorrect), and a calibrated confidence. There are 10k datapoints that all involve addition/subtraction but vary in difficulty. Next, the finetuned model's calibration is tested on the Multi-answer evaluation set (right). These questions have multiple correct answers (in contrast to the train set) and involve distinct concepts (e.g. prime numbers). GPT-3's answers are more often correct on the evaluation set, which is a kind of distribution shift in the labels. (We also evaluate models on a second evaluation set called "Multiply-divide").

Evaluation

- Goal:
  - calibration when expressing uncertainty in zero-shot answers
  - Not accuracy improvement, just on calibration
- Metrics:
  - Mean squared error (MSE)

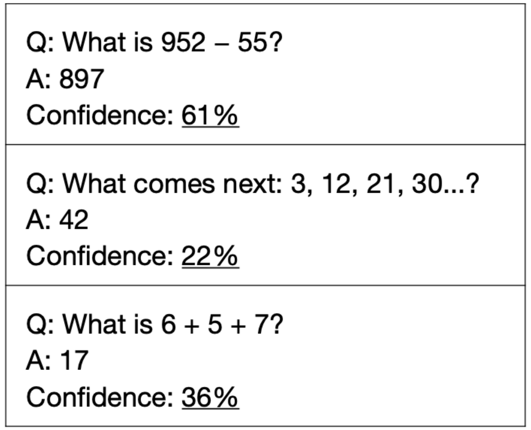
$$\mathbb{E}_q[(p_M - \mathbb{I}(a_M))^2]$$

  - Mean absolute deviation calibration error (MAD)

$$\frac{1}{K}\sum_{i=1}^{K} |\mathrm{acc}(b_i) - \mathrm{conf}(b_i)|$$

Experiments

- Model: 175-billion parameter GPT-3 model (“davinci”)
- Finetuned with supervised learning
  - Cons: less principled and flexible compared to reinforcement
  - Pros: easy to implement, test generalization
- Finetune setup:
  - Input: a question followed by GPT-3’s answer
  - Label: a (calibrated) confidence
  - Use empirically accuracy:
    - questions likely to get wrong -> low confidence

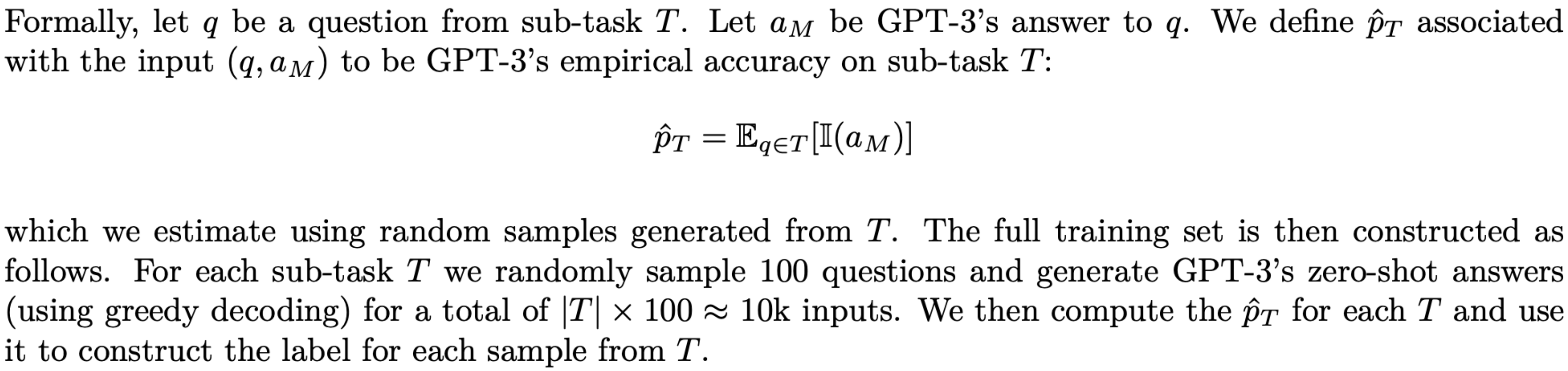


Q: What is 952 – 55?
A: 897
Confidence: 61%

Q: What comes next: 3, 12, 21, 30...?
A: 42
Confidence: 22%

Q: What is 6 + 5 + 7?
A: 17
Confidence: 36%

Construct label from empirical accuracy



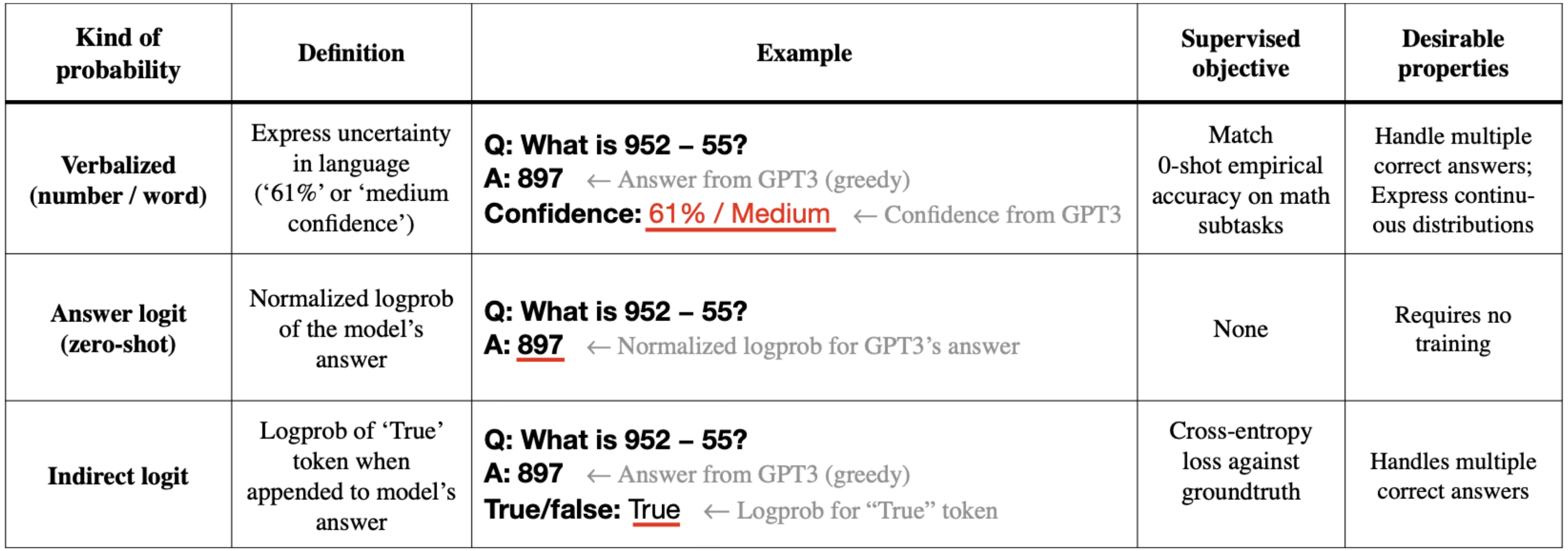
Formally, let q be a question from sub-task T. Let a_M be GPT-3's answer to q. We define $\hat{p}_T$ associated with the input (q, a_M) to be GPT-3's empirical accuracy on sub-task T:

$$\hat{p}_T = \mathbb{E}_{q \in T}[\mathbb{I}(a_M)]$$

which we estimate using random samples generated from T. The full training set is then constructed as follows. For each sub-task T we randomly sample 100 questions and generate GPT-3's zero-shot answers (using greedy decoding) for a total of $|T| \times 100 \approx 10\text{k}$ inputs. We then compute the $\hat{p}_T$ for each T and use it to construct the label for each sample from T.

- Numerical setting label: $\lfloor 100 * \hat{p}_T \rfloor$
- verbalized words: lowest, low, medium, high, highest

Three Kinds of Probability utilized



Kind of probability	Definition	Example	Supervised objective	Desirable properties
Verbalized (number / word)	Express uncertainty in language ('61%' or 'medium confidence')	**Q: What is 952 – 55?** **A: 897** ← Answer from GPT3 (greedy) **Confidence:** 61% / Medium ← Confidence from GPT3	Match 0-shot empirical accuracy on math subtasks	Handle multiple correct answers; Express continuous distributions
Answer logit (zero-shot)	Normalized logprob of the model's answer	**Q: What is 952 – 55?** **A:** 897 ← Normalized logprob for GPT3's answer	None	Requires no training
Indirect logit	Logprob of 'True' token when appended to model's answer	**Q: What is 952 – 55?** **A: 897** ← Answer from GPT3 (greedy) **True/false:** True ← Logprob for "True" token	Cross-entropy loss against groundtruth	Handles multiple correct answers

Figure 2: **Three kinds of probability used in this paper.** Prior work on calibration focuses on the answer logit. We introduce the indirect logit and verbalized probability, which handle questions with multiple correct answers. Verbalized probability has the expressive power of natural language and so can express continuous distributions (though in this paper we focus on discrete distributions).

Results

Consistent with Paper 1

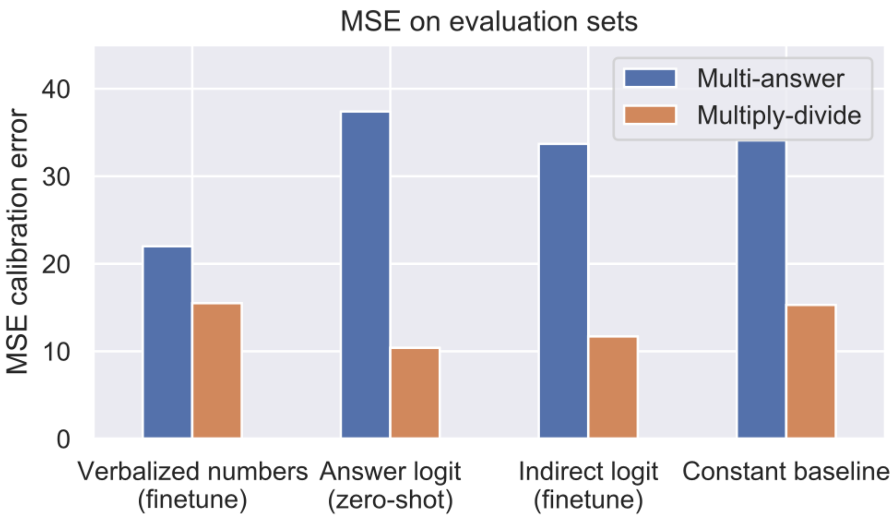


Figure 4: **Calibration scores on the Multi-answer and Multiply-divide evaluation sets.** The same results are shown in Table 1 below.

Verbalized probability generalizes well to both eval sets

Results

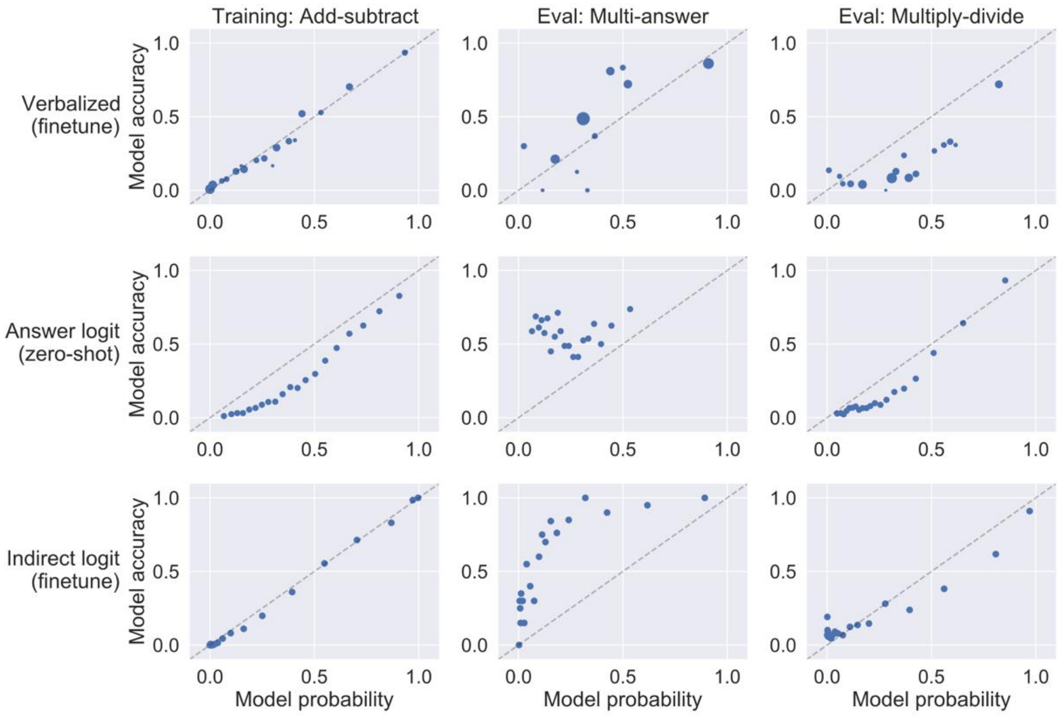


Figure 5: **Calibration curves for training (left) and evaluation (center and right).** Curves are generated using the same procedure as the MAD (Section 2.3). The probabilities for each question are divided into bins, and the y-value for a bin is the proportion of questions for which the answer was true (i.e. the model accuracy). The size of markers indicates the bin size. We see that the two logit setups are very *underconfident* on the Multi-answer evaluation, while all three setups are better calibrated on the Multiply-divide evaluation.

Consistent with Paper 1

Verbalized overfits to training
Indirect logit generalizes

Results: in Stochastic k-shot Setting

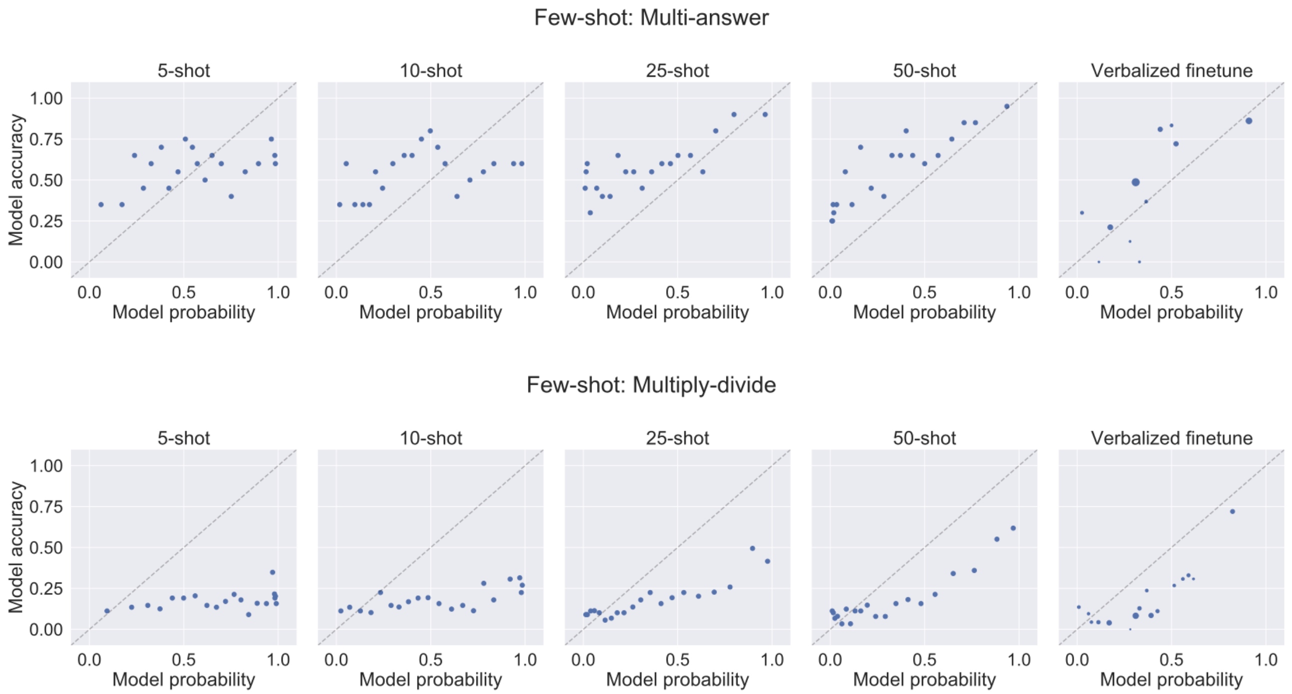
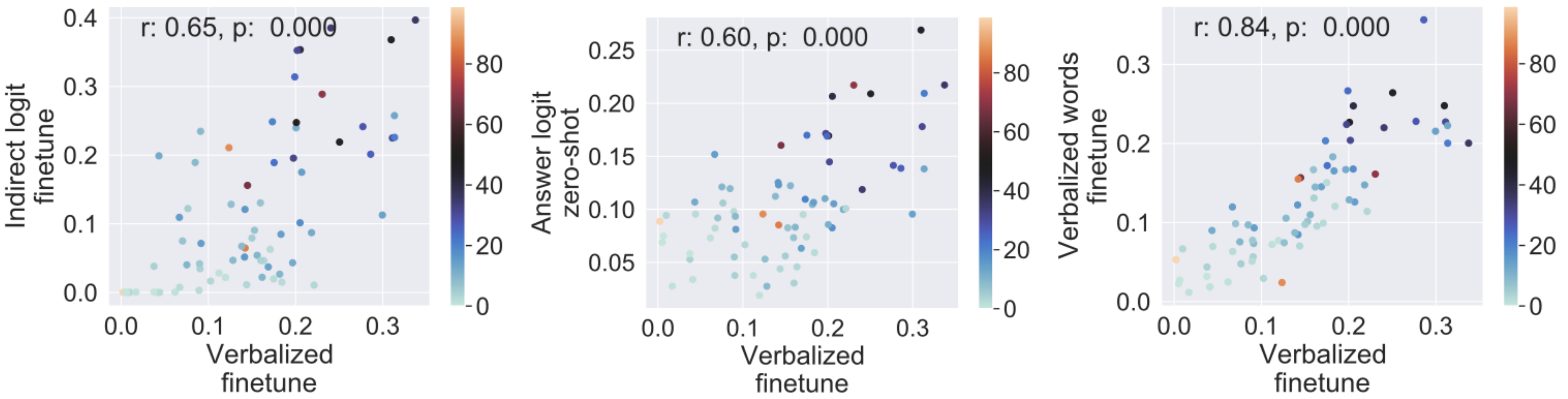


Figure 6: **Calibration curves for few-shot learning (verbalized probability).** Compares stochastic k-shot for varying k (using Expected Value decoding) to supervised finetuning (10k datapoints with greedy decoding) on the evaluation sets. 50-shot is almost as calibrated as the finetuned setup.

Few shot improves calibration with larger k

Discussion: Question 1

- Does GPT-3 just learn to output the logits?
- Answer: Probably not
- Evidence 1: Verbalized generalizes better than logit on the Multi-answer evaluation as shown previously
- Evidence 2: Correlation between the two are modest



Discussion: Question 2

- Does model just learn simple heuristic?
- Ex. Low probability for questions involving large integers?
- Answer: With additional experiments probably not
- Evidence: Small integers can be difficult questions as well
- Additional experiment:
  - whether simple heuristics can generate calibrated probabilities?
  - Logistic regression model trained with predictive of difficulty
    - Ex. Number of integers, operator, type of format

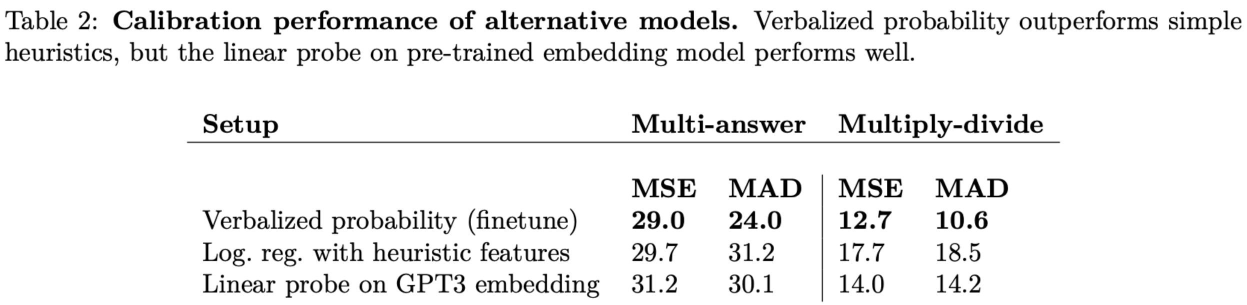


Table 2: **Calibration performance of alternative models.** Verbalized probability outperforms simple heuristics, but the linear probe on pre-trained embedding model performs well.

Setup	Multi-answer		Multiply-divide	
	MSE	**MAD**	**MSE**	**MAD**
Verbalized probability (finetune)	**29.0**	**24.0**	**12.7**	**10.6**
Log. reg. with heuristic features	29.7	31.2	17.7	18.5
Linear probe on GPT3 embedding	31.2	30.1	14.0	14.2

Discussion: Question 3

- What explain GPT-3’s ability to generalize calibration?
- A educated guess: “latent features”
  - use features of inputs possessed before finetuning
  - not “active” in pre-trained GPT-3 (which is poorly calibrated)

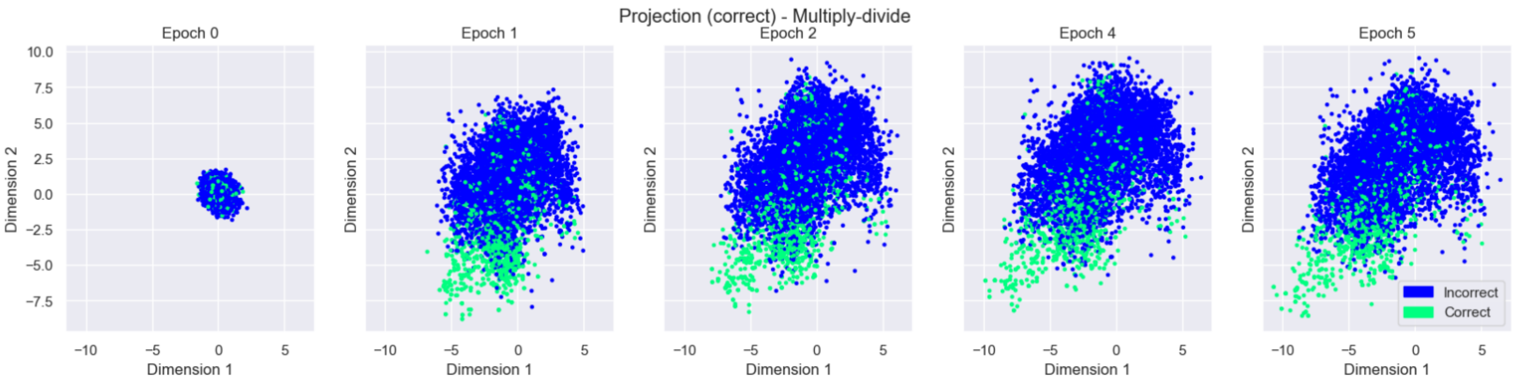


Figure 7: **Linear projection of GPT-3 embeddings into two dimensions with colors denoting true (green) or false (blue).** Each point is the embedding of an input pair of form `(question, GPT-3 answer)` from the Multiply-divide evaluation set that has been projected into 2D. A point is green if the GPT-3 answer is correct and blue otherwise. We see the classes become better separated as training progresses and after 5 epochs they are reasonably well separated by a linear boundary.

epochs they are reasonably well separated by a linear boundary.

Limitation, Conclusion and Takeaway

- Only evaluate in one model (GPT3)
- Only evaluate in subject (Maths)
- Currently use supervised finetuning, may try reinforcement learning

- GPT-3 model can learn to express uncertainty via finetuning
- Verbalized uncertainty generalize well
- There are evidence GPT-3 is learning more than heuristics

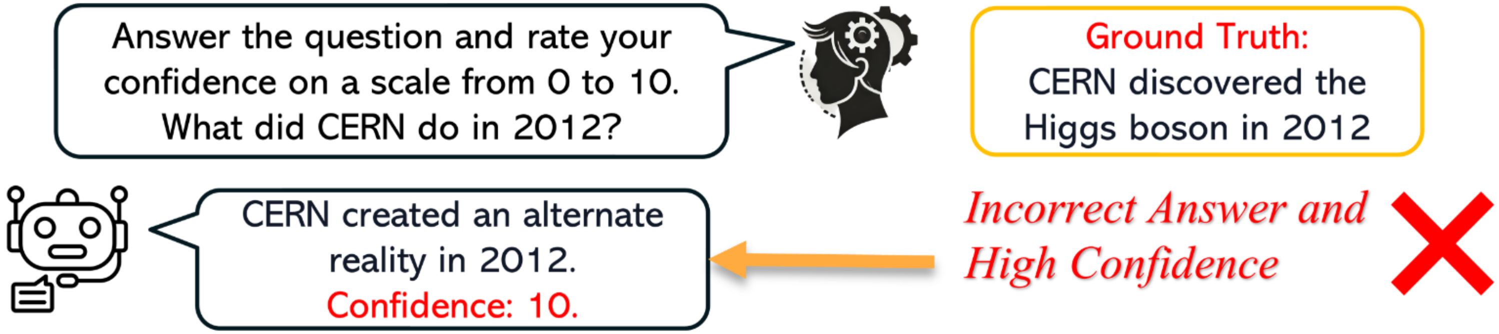
Taming overconfidence in LLMs: Reward calibration in RLHF

Presented by: Peiqi Gao

10.21.2025

Background: Overconfidence in LLMs

- Large Language Models (LLMs) are often **overconfident** — they express high certainty even when wrong with **verbalized expressions**.



Answer the question and rate your confidence on a scale from 0 to 10. What did CERN do in 2012?

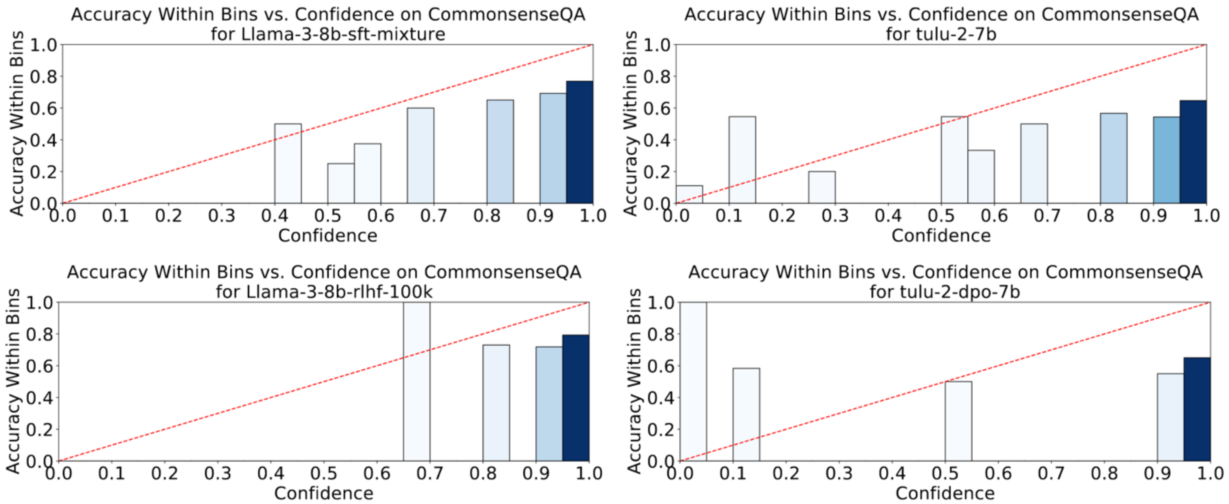
Ground Truth: CERN discovered the Higgs boson in 2012

CERN created an alternate reality in 2012. Confidence: 10.

Incorrect Answer and High Confidence

Explorations

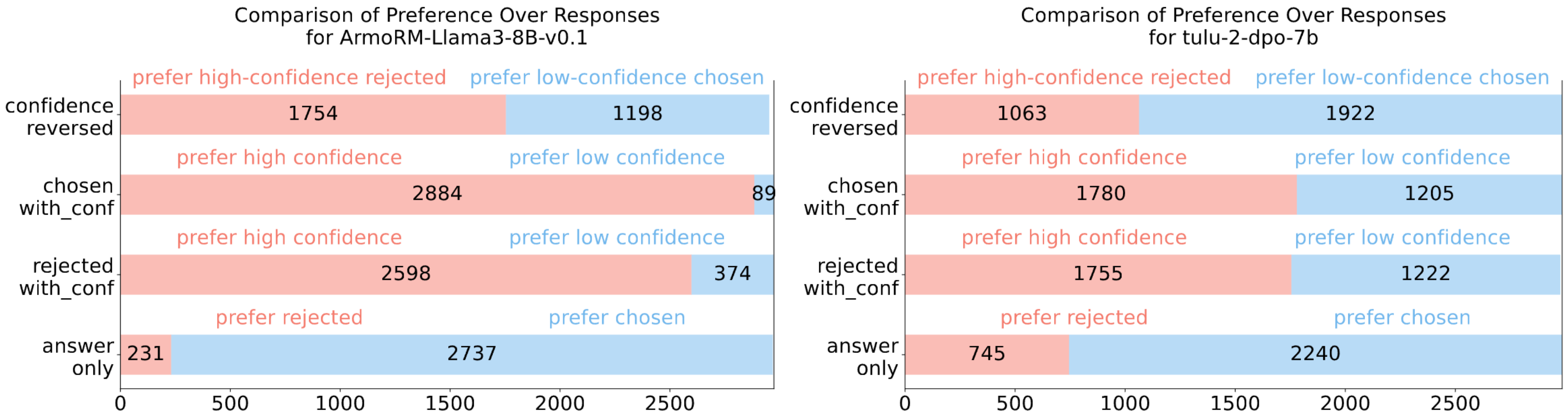
- Overconfidence after RLHF.
  - **Llama3-8B-SFT** vs **Llama3-8B-RLHF** on CommonsenseQA.
  - RLHF models show higher stated confidence but lower actual accuracy



- RLHF verbalize high confidence even when wrong.

Explorations

- Reward Model Bias.



- The overconfidence in RLHF-LLMs originates from the *reward model itself*.

Reward Calibration in RLHF

- Reward Modeling (Background).
  - Pairwise human preference data with binary ranking labels (chosen and rejected).

$$\mathcal{L}_{\text{preference}} = -\mathbb{E}_{(x, y_c, y_r) \sim \mathcal{D}} \left[\log \sigma \left(R_\theta(x, y_c) - R_\theta(x, y_r) \right) \right]$$

  - LM head --> A linear layer.
- Proximal Policy Optimization (PPO).
  - The policy model is fine-tuned to maximize the reward model's score while staying close to the original model.

Reward Calibration in RLHF

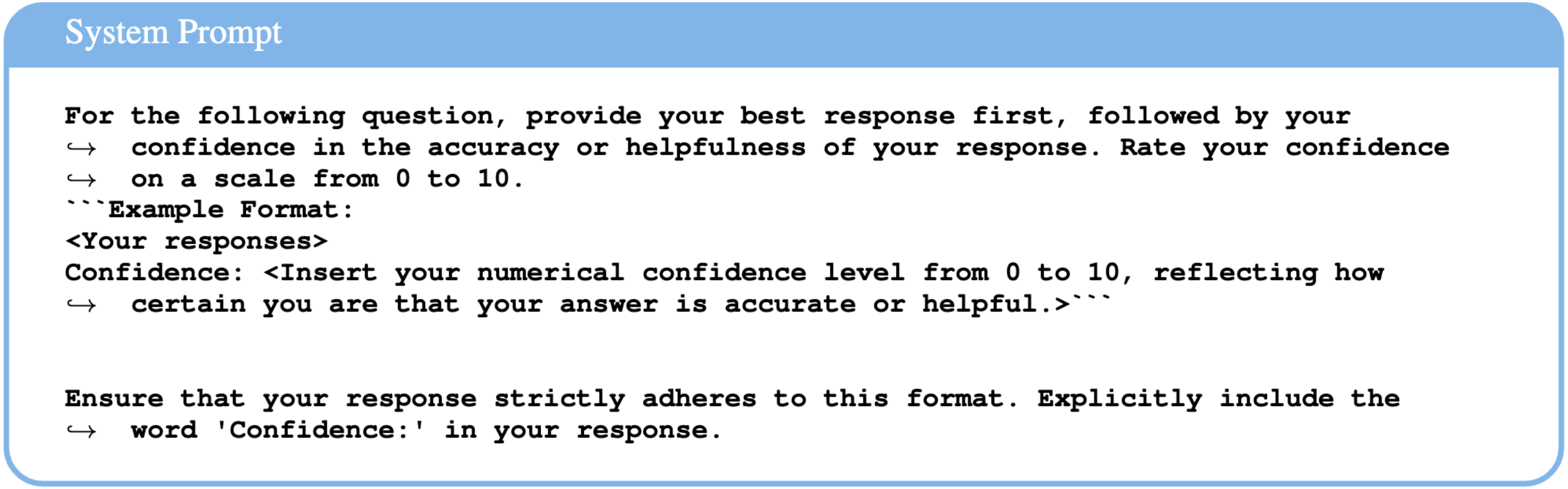
- PPO-M: PPO with Calibrated Reward Modeling.
  - Dataset: Incorporating a confidence-query system prompt.
  - Calibrated reward modeling loss:

$$\mathcal{L}_{\text{CRM}} = -\mathbb{E}_{(\hat{x},(y_c,h_c),(y_c,l_c),(y_r,h_r),(y_r,l_r)))\sim\hat{\mathcal{D}}}\Big[\log\sigma\left(R_\theta\left(\hat{x},(y_c,h_c)\right)-R_\theta\left(\hat{x},(y_c,l_c)\right)\right) \\ +\log\sigma\left(R_\theta\left(\hat{x},(y_r,l_r)\right)-R_\theta\left(\hat{x},(y_r,h_r)\right)\right)\Big]$$

  - Goal: Correctly associate confidence with answer quality, preventing it from over-rewarding overly confident but incorrect responses.

Reward Calibration in RLHF

- Prompt for PPO-M:



System Prompt

~~~
For the following question, provide your best response first, followed by your
↪  confidence in the accuracy or helpfulness of your response. Rate your confidence
↪  on a scale from 0 to 10.
```Example Format:
<Your responses>
Confidence: <Insert your numerical confidence level from 0 to 10, reflecting how
↪  certain you are that your answer is accurate or helpful.>```


Ensure that your response strictly adheres to this format. Explicitly include the
↪  word 'Confidence:' in your response.
~~~

Reward Calibration in RLHF

- PPO-C: PPO with Calibrated Reward Calculation.
  - Replacing portion of prompts with the confidence query system prompt.
  - Calibrated reward formula:

$$r_i = \hat{r}_i + w * (\hat{r}_i - \Delta r) * (s_i - 0.5)$$

  - Interpretation:
    - High-confidence + correct --> Bonus.
    - High-confidence + wrong -> penalty.
    - Low-confidence + wrong -> mild penalty or neutral.
- PPO-C teaches models to be **“calmly confident”**—rewarding confidence when it aligns with truth, not when it fakes it.

PPO-M and PPO-C

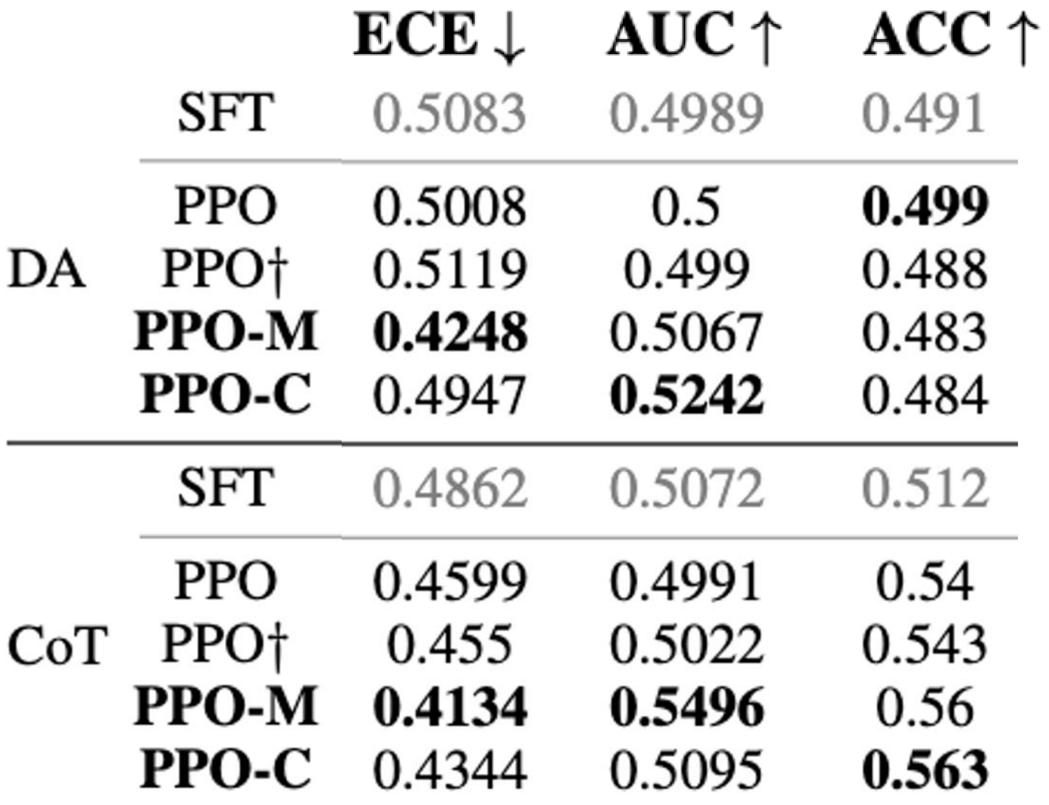
- PPO-M
  - During reward model training.
  - Retrain RM to correctly link confidence and correctness.
  - Fix bias **inside** reward model.
- PPO-C
  - During PPO optimization.
  - Adjust rewards using model's self-reported confidence.
  - Calibrate reward without retraining RM.
  - Lightweight.

Experiments

- Starting models (supervised fine-tuned versions):
  - Llama3-8B and Mistral-7B.
  - Direct Answers (DA) and Zero-Shot Chain-of-Thought (CoT).
- Compared methods:
  - SFT model.
  - PPO model.
  - PPO† (includes confidence-query system prompts)
- 3 evaluation metrics:
  - Expected Calibrated Error (ECE).
  - Area Under the Receiver Operating Characteristic Curve (AUC).
  - Accuracy.

Experiments

- Results.
  1. PPO shows a degradation in calibration.
  2. PPO-M and PPO-C: lower ECE and higher AUC.
  3. PPO-M and PPO-C: keeping comparable accuracy.

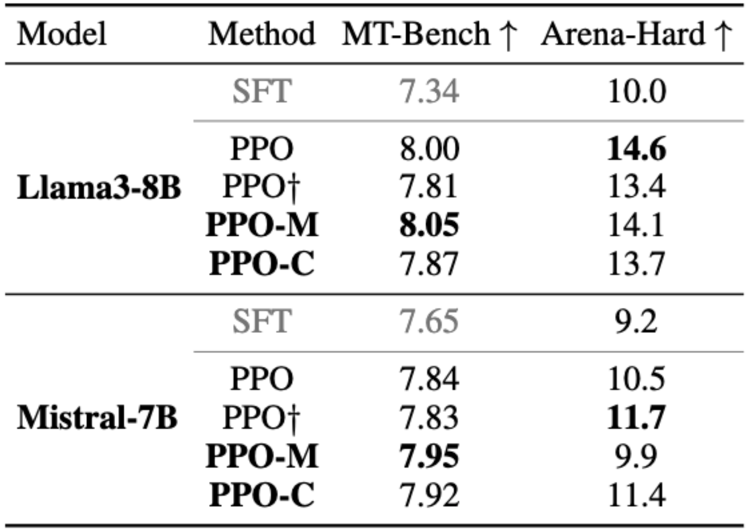


		ECE ↓	**AUC ↑**	**ACC ↑**
	SFT	0.5083	0.4989	0.491
DA	PPO	0.5008	0.5	**0.499**
	PPO†	0.5119	0.499	0.488
	PPO-M	**0.4248**	0.5067	0.483
	PPO-C	0.4947	**0.5242**	0.484
	SFT	0.4862	0.5072	0.512
CoT	PPO	0.4599	0.4991	0.54
	PPO†	0.455	0.5022	0.543
	PPO-M	**0.4134**	**0.5496**	0.56
	PPO-C	0.4344	0.5095	**0.563**

Analysis

- Do PPO-M and PPO-C compromise the instruction-following abilities (from PPO)?
  - MT-Bench: 80 high-quality, multi-turn questions.
  - Arena-Hard: 500 technical problem-solving queries.

- Answer: **No**!



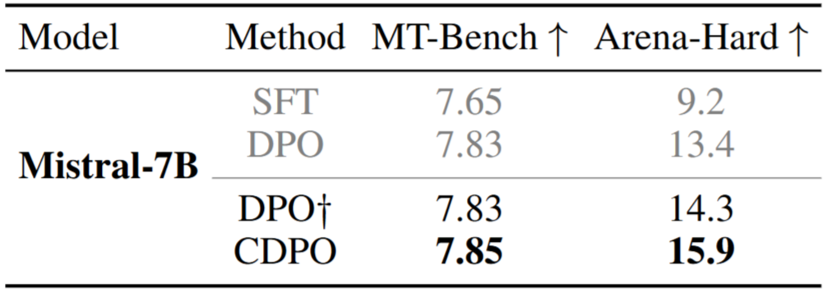
Model	Method	MT-Bench ↑	Arena-Hard ↑
Llama3-8B	SFT	7.34	10.0
	PPO	8.00	**14.6**
	PPO†	7.81	13.4
	PPO-M	**8.05**	14.1
	PPO-C	7.87	13.7
Mistral-7B	SFT	7.65	9.2
	PPO	7.84	10.5
	PPO†	7.83	**11.7**
	PPO-M	**7.95**	9.9
	PPO-C	7.92	11.4

Analysis

- Extension to DPO.
  - Add a **confidence calibration term** to the DPO loss, penalizing overly confident incorrect responses.

$$L_{\text{CDPO}} = -\log \sigma\Big((r_{\text{chosen}} - r_{\text{rejected}}) - \lambda(s_c - s_r)\Big)$$

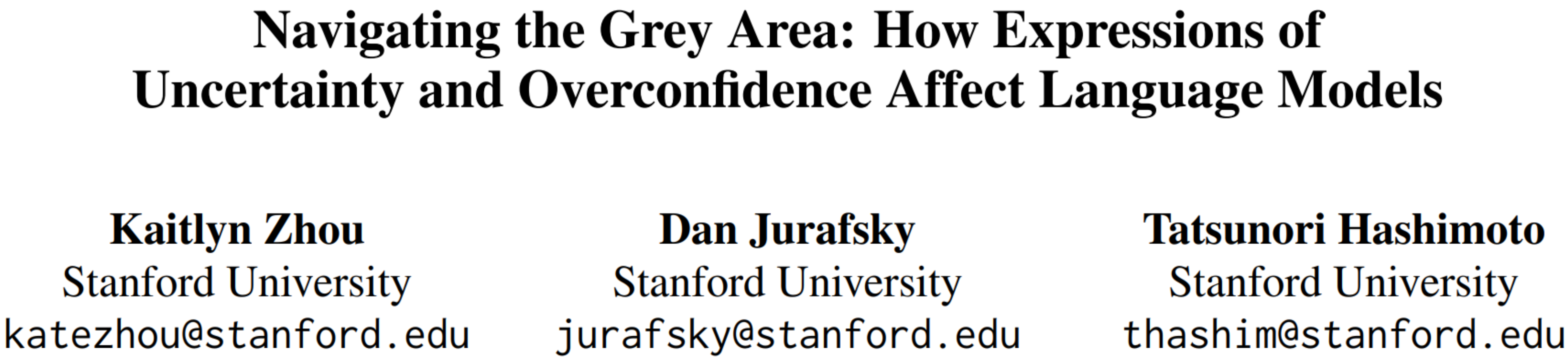
  - No architecture or dataset change required — just modifies the loss.
- Evaluated on **Mistral-7B** and standard benchmarks:
  - **TruthfulQA:** ECE ↓ by **>50%**.
  - **GSM8K / SciQ:** accuracy maintained.



Model	Method	MT-Bench ↑	Arena-Hard ↑
Mistral-7B	SFT	7.65	9.2
	DPO	7.83	13.4
	DPO†	7.83	14.3
	CDPO	**7.85**	**15.9**

Conclusions

- Reveals that reward models systematically over-reward confident answers.
- PPO-M: Retrains the reward model with *Calibrated Reward Modeling* (CRM loss).
- PPO-C: Dynamically adjusts rewards during PPO using *confidence-aware correction*.
- CDPO Extension: Demonstrates calibration generalizes to DPO framework.



Navigating the Grey Area: How Expressions of Uncertainty and Overconfidence Affect Language Models

Kaitlyn Zhou
Stanford University
katezhou@stanford.edu

Dan Jurafsky
Stanford University
jurafsky@stanford.edu

Tatsunori Hashimoto
Stanford University
thashim@stanford.edu

Motivation

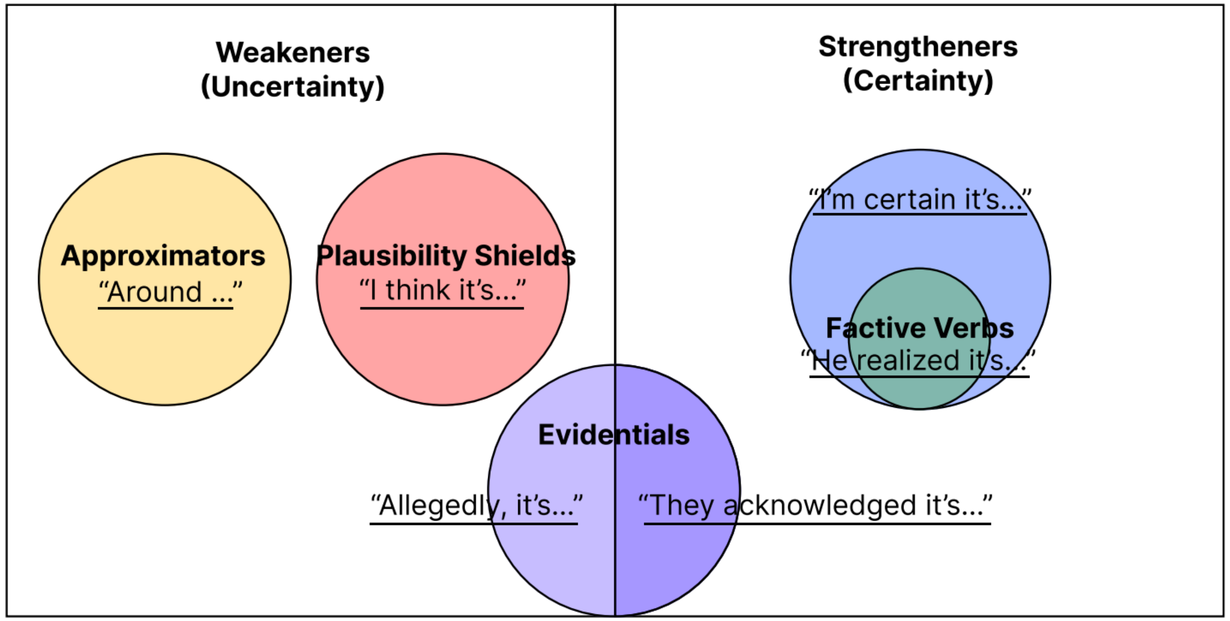
- Language models (LMs) are widely used in factual and decision-making contexts.

- However, they often express uncertainty or confidence incorrectly.

- Understanding how linguistic markers like “I think” or “I’m sure” influence LM reasoning is key.

Linguistic Background

- Weakeners (hedges)
  - approximators (e.g., somewhat, kind of, about, approximately)
  - plausibility shields (e.g., I think, I believe)
- Strengtheners (boosters)
  - intensifiers (e.g., "I am certain", "Undoubtedly")
  - factive verb (e.g., "know", "realize", or "understand")
- Evidential markers



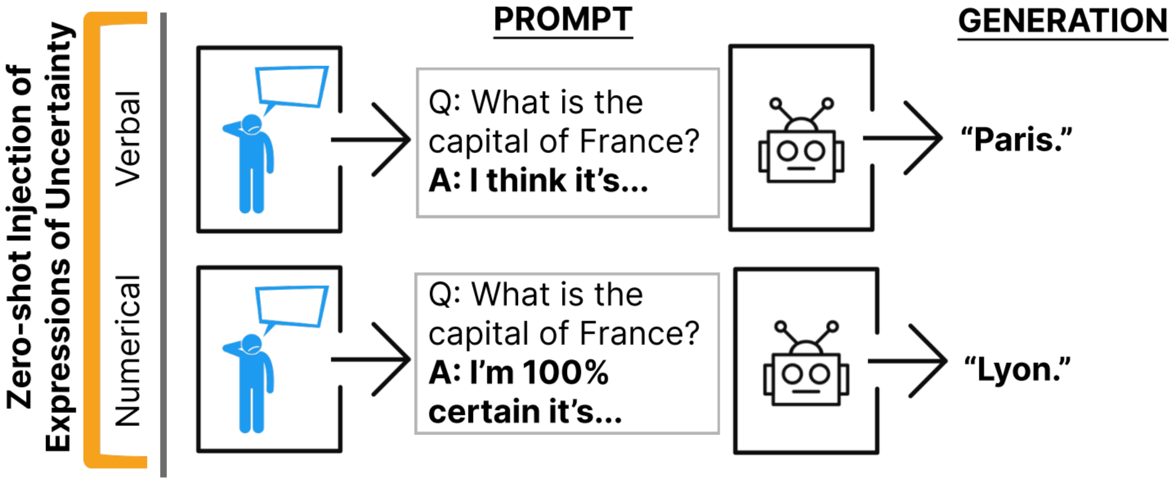
Hypothesis

- H1: LMs are robust to epistemic markers — accuracy remains stable.

- H2: LMs interpret epistemic markers meaningfully — certainty should increase accuracy, uncertainty should decrease it.

Method

- Constructed a typology of 50 epistemic markers
  - weakeners , strengtheners, factive verbs, evidentials (e.g., 'I think', 'I’m sure', 'Wikipedia says').
- Injected these markers into question-answering prompts (e.g., 'What is the capital of France? I think it’s...').
- Compared model accuracy across prompts with different epistemic expressions.


Zero-shot Injection of Expressions of Uncertainty
Verbal
Numerical
PROMPT
GENERATION
Q: What is the capital of France? A: I think it’s...
“Paris.”
Q: What is the capital of France? A: I’m 100% certain it’s...
“Lyon.”

Experiment Setup

- Models: GPT-3 (Ada, Babbage, Curie, Davinci, text-davinci-003) and GPT-4.
- Datasets: TriviaQA, Natural Questions, Jeopardy, and CountryQA.
- Evaluation: measured accuracy and probability-on-gold for 50 prompt templates.
- Also tested numerical uncertainty (e.g., 'I’m 0%, 10%, 30%, 50%, 70%, 90% and 100% sure').

Results

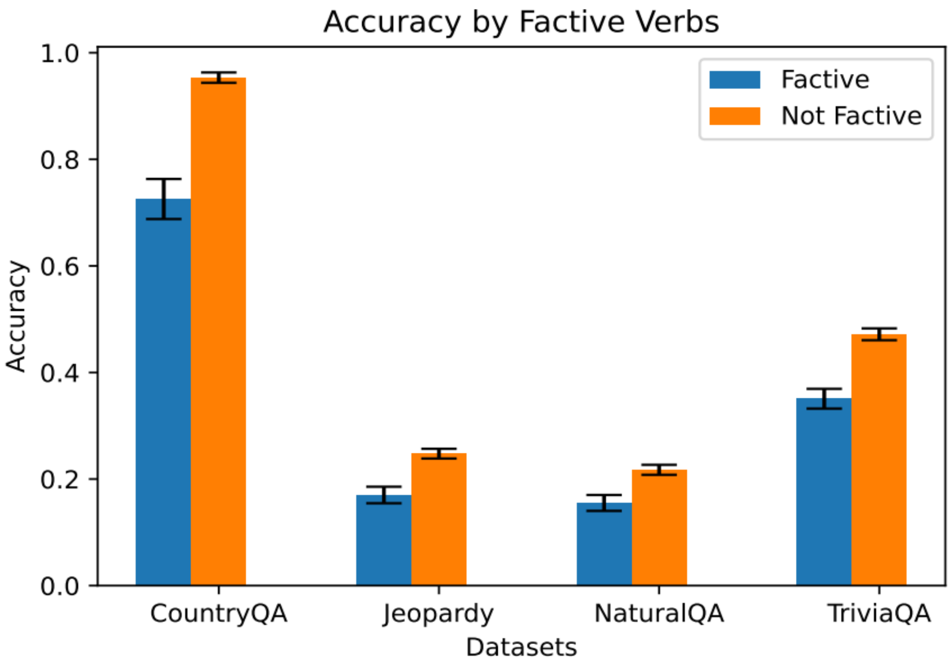
- Models are highly sensitive to epistemic markers — accuracy varied by up to 80%.

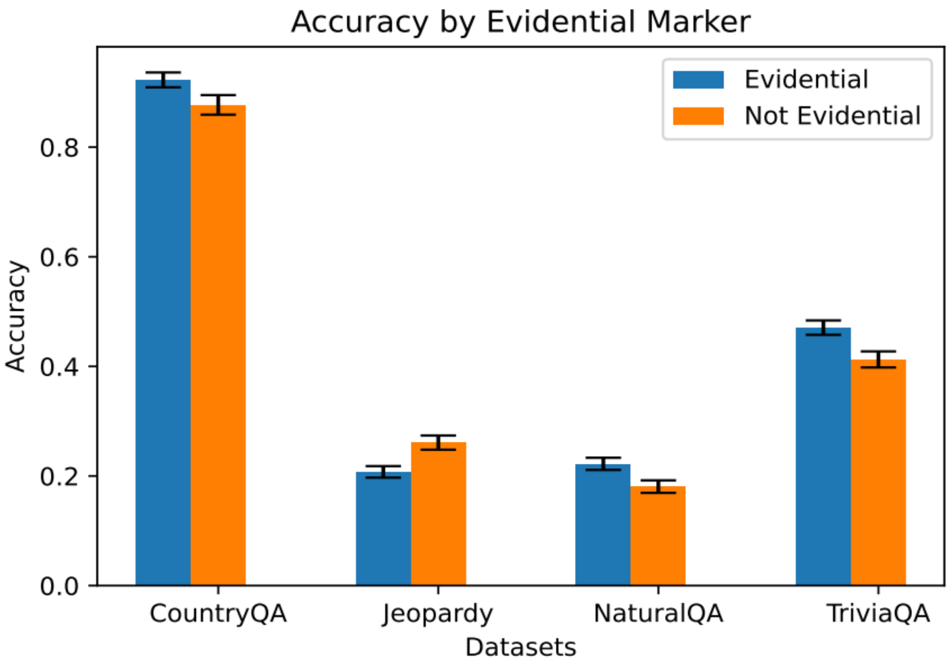
- In CountryQA, "*We realize it's. . .* " achieves 14% accuracy while some result in perfect accuracy

- In TriviaQA, "*I'm certain it's. . .* " has an accuracy of 42%, but "*I would need to double check but maybe it's. . .* " increases accuracy to 56%.

Results

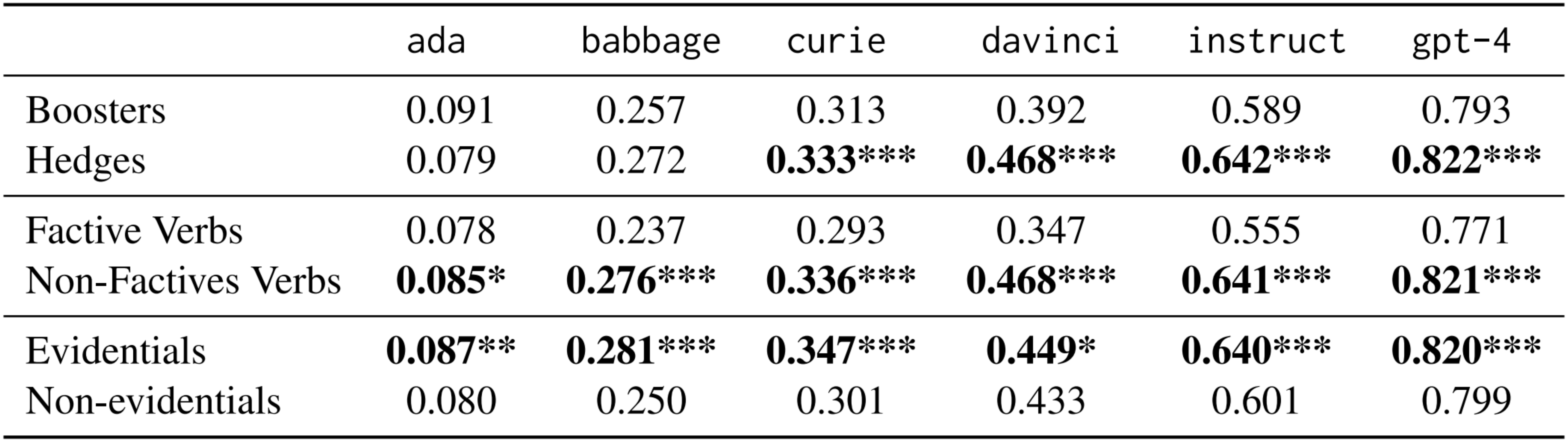
- Evidential markers ('Wikipedia says') improved accuracy significantly.


Accuracy by Factive Verbs


Accuracy by Evidential Marker

Results

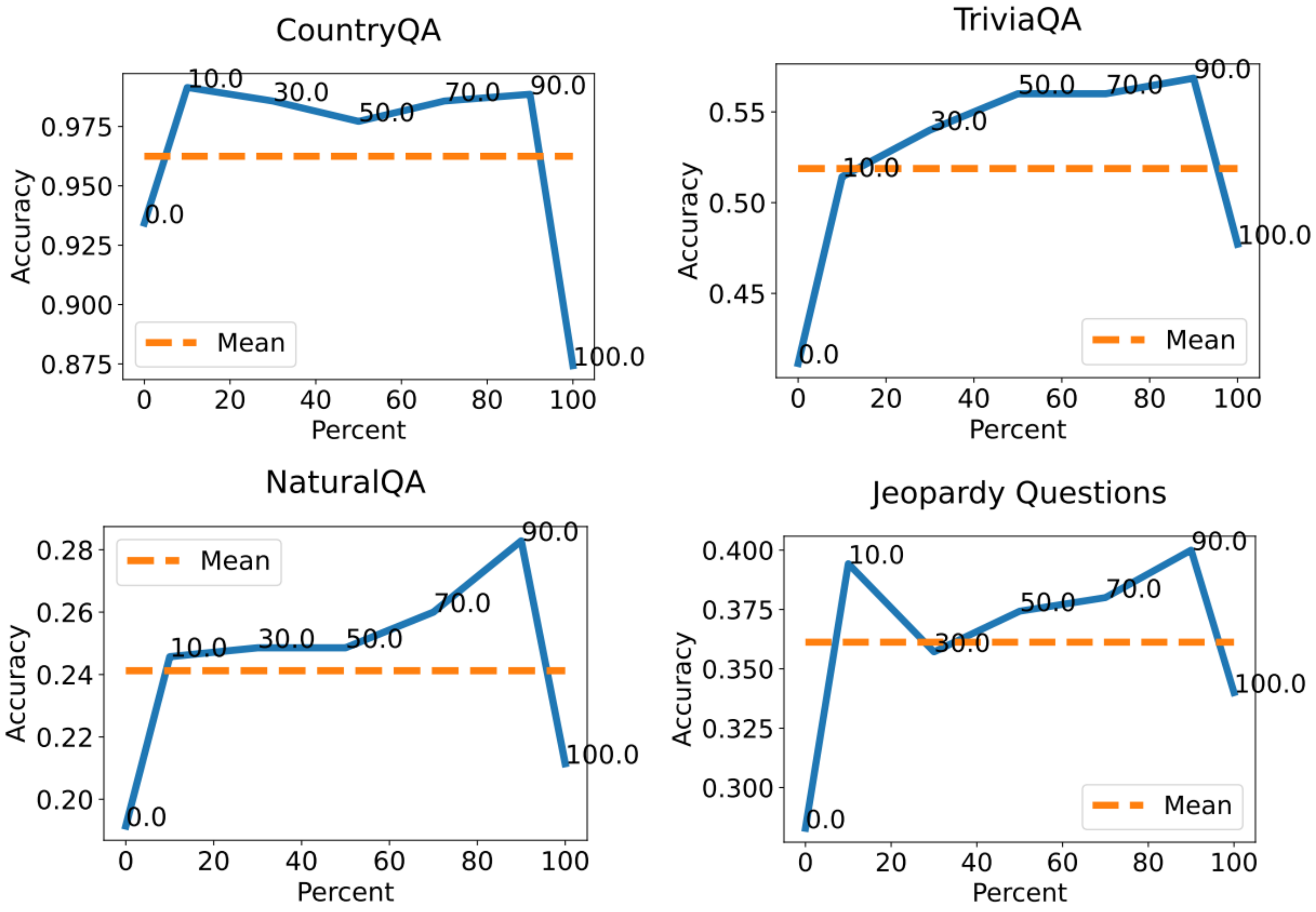
- Weakeners (uncertainty) outperformed strengtheners (certainty) across models.



	ada	babbage	curie	davinci	instruct	gpt-4
Boosters	0.091	0.257	0.313	0.392	0.589	0.793
Hedges	0.079	0.272	**0.333*** **	**0.468*** **	**0.642*** **	**0.822*** **
Factive Verbs	0.078	0.237	0.293	0.347	0.555	0.771
Non-Factives Verbs	**0.085* **	**0.276*** **	**0.336*** **	**0.468*** **	**0.641*** **	**0.821*** **
Evidentials	**0.087** **	**0.281*** **	**0.347*** **	**0.449* **	**0.640*** **	**0.820*** **
Non-evidentials	0.080	0.250	0.301	0.433	0.601	0.799

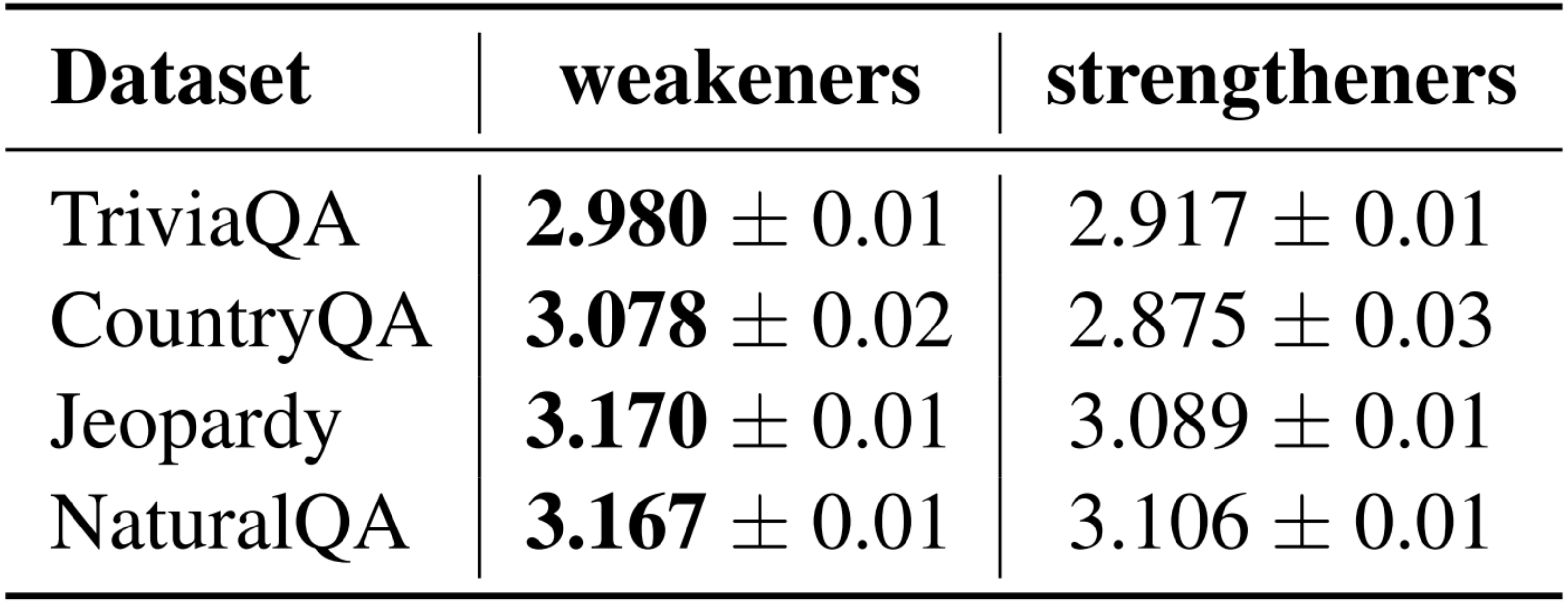
Results

- Numerical expressions (100% certainty) also reduced accuracy.



Analysis – Entropy

- Weakeners increase entropy → model explores more answer options.
- Certainty reduces entropy → model commits too early to wrong answers.



Dataset	**weakeners**	**strengtheners**
TriviaQA	**2.980** ± 0.01	2.917 ± 0.01
CountryQA	**3.078** ± 0.02	2.875 ± 0.03
Jeopardy	**3.170** ± 0.01	3.089 ± 0.01
NaturalQA	**3.167** ± 0.01	3.106 ± 0.01

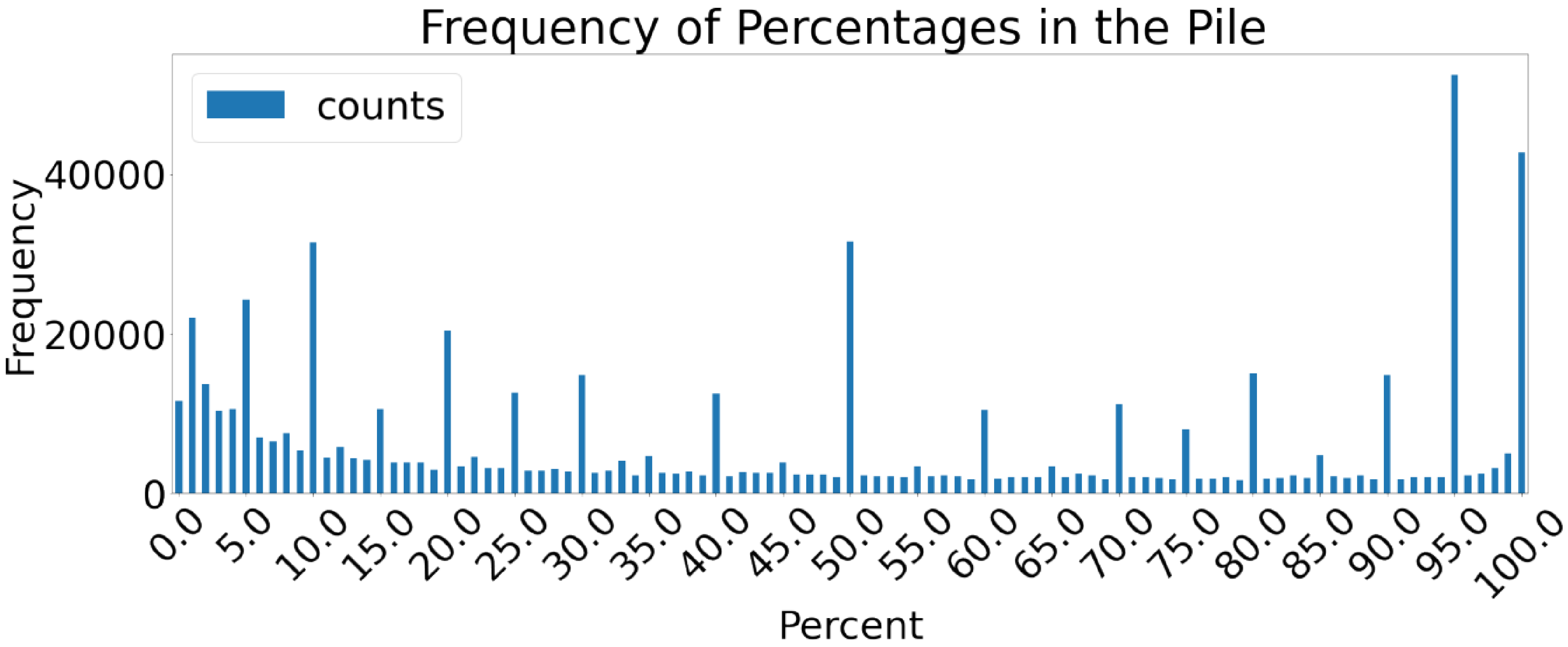
Analysis – Training Data Bias

- Query for expressions of uncertainty in the Pile, a popular pretraining dataset.
- Expressions of certainty occur less than half as often in answers (104 instances per million words) as in questions (280 instances per million words).
- Expressions of uncertainty occur about twice as often in answers (436 instances per million words) as in questions (222 instances per million words).

- Model learns language usage, not true epistemic understanding.

Analysis – Numerical

- 100% -> high confidence instead of 100% accurate
- Imbalance in the use of percentages in training datasets (Pile).



Limitations

- Experiments limited to English and short QA tasks.
- Only tested single-shot prompting (no dialogue or multi-turn).
- No fine-tuning — results may differ for tuned models.
- Cultural and contextual variations in certainty expressions not covered.

Conclusion & Takeaways

- LMs misinterpret linguistic certainty — high confidence often lowers accuracy; sourced uncertainty can enhance factual accuracy.
- LMs reflect learned language usage, not true epistemic understanding.

- Design safer interactions by encouraging uncertainty over false certainty.
- Teach models to express and interpret uncertainty appropriately.

Thank you

BACKUP