Calibration-Tuning: Teaching Large Language Models to Know What They Don't Know

Sanyam Kapoor sanyam@nyu.edu
Nate Gruver nvg7279@nyu.edu
Manley Roberts manley@abacus.ai
Arka Pal arka@abacus.ai
Samuel Dooley samuel@abacus.ai
Micah Goldblum goldblum@nyu.edu
Andrew Gordon Wilson andrewgw@cims.nyu.edu

Abstract

Large language models are increasingly deployed for high-stakes decision making, for example in financial and medical applications. In such applications, it is imperative that we be able to estimate our confidence in the answers output by a language model in order to assess risks. Although we can easily compute the probability assigned by a language model to the sequence of tokens that make up an answer, we cannot easily compute the probability of the answer itself, which could be phrased in numerous ways. While other works have engineered ways of assigning such probabilities to LLM outputs, a key problem remains: existing language models are poorly calibrated, often confident when they are wrong or unsure when they are correct. In this work, we devise a protocol called *calibration tuning* for finetuning LLMs to output calibrated probabilities. Calibration-tuned models demonstrate superior calibration performance compared to existing language models on a variety of question-answering tasks, including open-ended generation, without affecting accuracy. We further show that this ability transfers to new domains outside of the calibration-tuning train set.

1 Introduction

Whereas early successes of large language models (LLMs) highlighted their fluency and vast knowledge (Radford et al., 2019), they still lack many necessary capabilities, particularly as they are used and interpreted by a general audience. One such desiderata of LLMs is the ability to answer factually-based questions with factually correct answers. Further, it is desirable that LLMs be able to respond with a well-calibrated confidence, corresponding to a probability of correctness, when responding to such fact-based questions.

Autoregressive language models (Touvron et al., 2023b; OpenAI, 2023) allow us to compute the probability of a particular sequence of tokens they output by utilising the chain rule of probability to multiply the conditional probabilities of each generated token. For example, a model performing medical diagnosis may output "`This patient experienced numbness on one side of the body and degraded vision, so they likely suffered a stroke`" with associated probability 0.2%; however, there are innumerable ways to phrase this diagnosis, and we need to add up all their corresponding probabilities to measure the *concept-level* probability of the stroke diagnosis. This calculation is infeasible since there are simply too many such sequences. Thus, the token-level probabilities of existing language models do not allow for useful confidences for open-ended generation, limiting their value for decision making beyond multiple-choice scenarios.

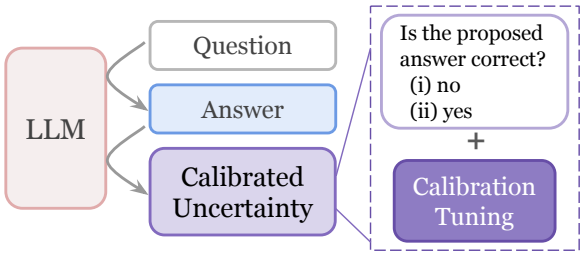


Figure 1: We propose calibration tuning (see Section 4) as a method for deriving calibrated uncertainty estimates from language models on question answering tasks (multiple choice or open-ended). Uncertainty estimates come from prompting the language model for its correctness and fine-tuning directly on this task. Our approach outperforms common baselines, including temperature scaling and probing methods (see Section 5).

While a number of works propose methods for extracting probabilities from language models, many of these methods are inapplicable to open-ended generation (Jiang et al., 2020; Zhao et al., 2021) or are prohibitively expensive to compute (Kuhn et al., 2023). Moreover, existing language

Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024), pages 1–14

models are simply miscalibrated (Chen et al., 2022; Zhang et al., 2023).

In this work, we propose an instruction tuning-inspired method for LLMs to output well-calibrated concept-level uncertainty estimates which are useful for both multiple-choice question answering and open-ended generation alike.

Our method, *calibration tuning*, produces superior calibration to existing approaches across question answering tasks, including out-of-distribution tasks. While we perform calibration tuning on questions phrased as multiple-choice with known answers, we show that our method generalizes to open-ended evaluations too. Calibration tuning is easy to implement, cheap to deploy for inference, and does not impact model performance.

2 Related Work

Calibration. A model is well-calibrated when an outcome predicted with probability p does occur p fraction of the time in reality. This alignment between predictions and reality is measured using the expected calibration error (ECE) via empirical binning (Naeini et al., 2015), such that an ECE of 0 corresponds to perfect calibration, i.e. a model knows when its wrong. Having well-calibrated probabilities is crucial for effective downstream decision-making.

Guo et al. (2017) reignited the discussion of calibration for neural network classifiers by demonstrating that many modern neural networks are poorly calibrated, and post-hoc temperature scaling is an effective method for calibration of pretrained models. Subsequent literature, however, shows that calibration can be directly improved at train time via improved learning objectives (Minderer et al., 2021; Mukhoti et al., 2020; Müller et al., 2019; Tran et al., 2022). In similar spirit, our work shows that well-calibrated large language models (LLMs) are indeed possible by careful modification of the language modeling objective during fine-tuning.

Calibration in LLMs. Defining calibration for language models is challenging, especially for variable length response sequences. In one instance, Braverman et al. (2019) define calibration in terms of the entropy of distribution of fixed-length sequences. Under this definition, the entropy rates of generation dramatically drift upwards as the sequence lengths increase, hinting severe miscalibration of language models. Our framework, building on ECE, allows for a general definition of calibration of language models that is broadly applicable.

Contrary to prior observations, Chen et al. (2022) suggest that language models do not necessarily learn to become better calibrated by pretraining longer. Following this observation, we show that a carefully devised fine-tuning objective can significantly improve calibration of large language models.

For auto-regressive LLM generation, Jiang et al. (2020) provide an early investigation (e.g. T5 (Raffel et al., 2019), BART (Lewis et al., 2019), GPT-2 (Radford et al., 2019)) and report very poor out-of-the-box calibration for question-answering tasks. Such miscalibration is then shown to improve via logit temperature scaling. This approach, however, is limited by the requirement of candidate answers at both train and test time. Calibration tuning instead does not rely on a pre-existing set of candidate answers.

Calibration tuning is most closely inspired by Kadavath et al. (2022), that shows evidence that LLMs can in fact be well-calibrated for question-answering tasks when the answers are provided as choices, or true/false statements. Yin et al. (2023) take a step back to evaluate an LLMs ability to identify whether a question is answerable or not. Their focus, however, remains on evaluating *self-knowledge*. The uncertainty of an answer is evaluated via representational similarity to a set of reference sentences that encompass uncertain meanings, a restriction that limits broader applicability. Calibration tuning on the other hand only requires a subset of existing training data for instruction-tuning, without needing a reference set, while providing a framework to directly *improve* calibration.

An alternative class of approaches to estimate confidence rely on linguistic features — Kuhn et al. (2023) propose *semantic uncertainty* which clusters generated sequences via bi-directional entailment to account for semantic similarity among multiple candidate answer sequences. Verbal elicitation approaches ask the model to express its confidence in words (Lin et al., 2022). Zhou et al. (2023) investigate the impact of linguistic features such as hedges or epistemic markers on natural language generation.[1] Such approaches, however, remain out of scope for our work since we aim to modify existing models to improve calibration rather than

[1] See Xiong et al. (2023) for a detailed recent survey on methods for verbal elicitation.

extract better expressions of uncertainty.

In summary, we emphasize that calibration tuning (1) provides a broadly applicable definition of calibration for variable-length language generation, (2) builds on prior literature that suggests fine-tuning is necessary for improving calibration, (3) does not require additional data beyond the instruction-tuning dataset(s), and (4) prescribes a carefully constructed instruction-tuning loss that helps improve calibration.

3 Background

Autoregressive Language Models. LLMs perform next-token prediction over sequences. The model parameters, θ, are trained with cross-entropy loss, and parameterize a conditional distribution

$$p_\theta(w_{t+1}|w_{0:t}), \tag{1}$$

where the prompt $w_{0:t}$, is the input tokens, and w_{t+1} is the next token. In this paper we consider using LLMs for question answering, which involves the following inputs

- $\boldsymbol{P}$: the text prompt used to contextualize the question.
- $\boldsymbol{Q}$: the question, in text.
- $\boldsymbol{A}$: the ground-truth answer in text.
- $\hat{\boldsymbol{A}}$: the language model's answer.

LLM Prompting. LLM generations can be guided by modifying the prompt text that precedes sampled tokens. In question answering tasks, careful prompting (often called "prompt engineering") can be essential for eliciting good performance. A simple form of prompting is providing the language model with examples from a particular task - this is referred to as 'few-shot' prompting (Brown et al., 2020). In multiple-choice question answering, for example, it is common practice to provide the model with multiple question-answer pairs before generating a final answer to a question (Brown et al., 2020). We show this prompting strategy in Figure 2.

LLM Finetuning. From an engineering perspective, prompting is simple and lightweight, as it does not require updating model parameters, but can often be limited in its effectiveness. As a result, many finetuning procedures have also been developed to make LLMs useful for downstream tasks. For example, instruction tuning (Wei et al., 2021) can be used to improve the controllability of a model, or DPO (Rafailov et al., 2023) can be used to align a language model with human preferences. Although prompt-tuning was initially favored because the most powerful models were intractably large or blocked behind APIs, rapid improvements in open source availability, base model sizes, and finetuning procedures (e.g. LoRA with quantized base weights) have made finetuning practical on a more limited compute budget. In our work, we take advantage of these advances to improve the overall calibration of LLMs with a simple finetuning procedure.

Expected Calibration Error (ECE). A model's uncertainties are well calibrated if they align with the empirical probabilities–i.e. an event assigned probability p occurs at rate p in reality. Following (Naeini et al., 2015), we estimate ECE by binning the maximum output probability of each of n samples into b equally-spaced bins $\mathcal{B} = \{B_j\}_{j=1}^{b}$ w.r.t. the prediction confidence estimated for each sample. The empirical ECE estimator is given by,

$$\widehat{ECE} = \sum_{j=1}^{b} \frac{|B_j|}{n} \left|\mathrm{conf}(B_j) - \mathrm{acc}(B_j)\right|, \tag{2}$$

where $\mathrm{conf}(B_j)$ is the average confidence of samples in bin B_j and $\mathrm{acc}(B_j)$ is the corresponding accuracy within the bin. As is typical in literature, we use $b = 10$ bins. An ECE of 0 corresponds to a perfectly calibrated model, i.e. in each bin, the predicted confidence perfectly aligns with the proportion of the correct predictions of the model.

We now describe calibration tuning in detail.

4 Calibration Tuning

To perform calibration tuning (CT), we need ground truth question-answer pairs (Q, A), the language model's generation of the answer, $\hat{A}$, the language model's assessment of the correctness of its generated answer, $\hat{C}$, and whether the answer actually is correct, C. In multiple choice question answering, it easy to ascertain whether $\hat{A}$ is the same as A with exact string matching; $C = \texttt{True}$ if and only if $A = \hat{A}$. In open-ended question answering, we use an auxiliary grading prompt to assign C since the phrasings of A and $\hat{A}$ could be different but semantically the same.

The goal of calibration tuning is to construct an estimate for $p(C = \texttt{True})$ and have that estimate be well calibrated. To obtain this estimate, we

Few-shot prompt $(\boldsymbol{P})$	**Uncertainty query** $(\boldsymbol{U})$	**Grading**
Question: Which of the following represents an accurate statement concerning arthropods? Answer: They possess an open circulatory system with a dorsal heart. ... Question: Which of the following contain DNA sequences required for the segregation of chromosomes in mitosis and meiosis? Answer: Centromeres Question: $\boldsymbol{Q}$ Answer: $\hat{\boldsymbol{A}}$ </s>	$\boldsymbol{P}$ Question: $\boldsymbol{Q}$ Answer: $\hat{\boldsymbol{A}}$ Is the proposed answer correct? (i) no (ii) yes Answer: $\hat{\boldsymbol{C}}$ </s>	*For Multiple choice:* $\boldsymbol{C} =$ `True` iff $\boldsymbol{A} = \hat{\boldsymbol{A}}$ *For Open-ended:* **Grading prompt** $(\boldsymbol{G})$ The problem is: $\boldsymbol{Q}$ The correct answer for this problem is: $\boldsymbol{A}$ A student submitted the answer: $\hat{\boldsymbol{A}}$ The student's answer must be correct and specific but not overcomplete (for example, if they provide two different answers, they did not get the question right). However, small differences in formatting should not be penalized (for example, 'New York City' is equivalent to 'NYC'). Did the student provide an equivalent answer to the ground truth? Please answer yes or no without any explanation: $\boldsymbol{C}$ </s>

Figure 2: For calibration tuning, we use the few-show prompt and uncertainty query to yield the generated answer $\hat{A}$ and the correctness estimate $\hat{C}$. For multiple choice question answering, we grade the answer with an exact-match to the ground truth choice. For open-ended, we use a grading prompt. The token </s> refers to the end of sentence token. **Blue text** is included in the loss function.

follow Kadavath et al. (2022) and use the language model itself, in tandem with an *uncertainty query* U (shown in Figure 2). The language model predicts $\hat{\boldsymbol{C}}$ conditioned on the concatenation $[P, Q, \hat{A}, U]$. The loss for each answered question $(Q, A, \hat{A}, C)$ in the dataset is therefore

$$\widetilde{\mathcal{L}}_{\mathrm{CT}}(\theta) = -\log p_\theta(\hat{C} = C \mid [P, Q, \hat{A}, U]) \quad (3)$$

In order to make predicting $\hat{C}$ easy for a language model, we pose the problem as a multiple-choice response with two values. As shown in Figure 2, the uncertainty query U is restricted to answer with only two possible target tokens "i" and "ii". While we can potentially retain the full vocabulary to compute the language modeling loss for calibration-tuning, restricting to only two tokens prevents logit mass from spreading over tokens which are unrelated to the uncertainty query. Therefore, $\widetilde{\mathcal{L}}_{\mathrm{CT}}(\theta)$ in Eq. (3) is simply a language modeling loss normalized over the restricted set of tokens.

However, modifying the existing model can lead to a drift in the generation distribution of the underlying language model whose uncertainty calibration properties we are trying to improve. To counter such a drift, we regularize the training by matching the generation distribution with a divergence-based regularization term. Specifically, let p_{θ_0} be the language modeling distribution in Eq. (1) of the language model we wish to calibration-tune, and q_θ be the corresponding language modeling distribution as a consequence of calibration-tuning. We then use the Jenson-Shannon Divergence $\mathrm{JSD}(p_{\theta_0} \parallel q_\theta)$ (MacKay, 2004) between the two language modeling distributions as the regularizer, where $\mathrm{JSD}(p \parallel q) \triangleq 1/2(\mathrm{KL}(p \parallel m) + \mathrm{KL}(q \parallel m))$, where $m \triangleq 1/2(p + q)$ is the mixture distribution. JSD regularization is applied only to the logits corresponding to the target sequence A. Denoting the JSD regularization term by $\mathcal{R}_{\mathrm{CT}}(\theta; \theta_0)$, and weighting with a parameter κ for flexibility, the final regularized loss for calibration-tuning is,

$$\mathcal{L}_{\mathrm{CT}}(\theta; \kappa, \theta_0) = \widetilde{\mathcal{L}}_{\mathrm{CT}}(\theta) + \kappa \cdot \mathcal{R}_{\mathrm{CT}}(\theta; \theta_0) \quad (4)$$

4.1 Evaluating Correctness (C)

As stated above, for a given question with known and generated answers $(Q, A, \hat{A})$ the correctness C is True if the generated answer $\hat{A}$ matches the ground truth answer A. For multiple-choice question-answering, the matching process only involves checking the first token generated via greedy decoding.

For open-ended evaluations, determining if the

answer given is correct is more complex. One simple approach is to check if the ground truth answer A appears as a substring of answer $\hat{A}$. However, this does not capture rephrasings that may be essentially equivalent - such as "NYC" for "New York City," or "Daoism" and "Taoism." Conversely, it also has the potential to be overgenerous if the model is particularly verbose and emits many incorrect answers along with the correct string. Given the difficulty involved in writing a rule-based method for evaluating open-ended answer correctness, we use instead a strong auxiliary language model to evaluate correctness. The auxiliary language model is shown the query Q, the ground truth answer A, and the model's output $\hat{A}$, and is prompted to grade the answer whilst tolerating nuance. For full details of the prompt used see (Figure 2). In this paper we utilise GPT 3.5 Turbo as the auxiliary grading model. We conduct a comparison of human grading, substring grading, and GPT 3.5 Turbo grading on select subsets of MMLU in Appendix C. We find that humans and GPT 3.5 Turbo have much greater agreement than humans and the substring method.

4.2 Measuring Calibration

Because we frame correctness as a classification problem, i.e. predicting $\hat{C}$, we now have the ability to assign sequences of variable length a single probability value for its correctness, instead of assigning a probability based on the logits of the primary generation $\hat{A}$. Consequently, we can easily compute the ECE using the normalized probability of the token `ii` (corresponding to choosing the `yes` option) to compute each $\mathrm{conf}(B_b)$ in Eq. (2).

5 Experiments

We now test the effectiveness of calibration tuning (CT) via empirical evaluations.

Models. All our experiments are conducted with decoder-only LLaMA-2 models (Touvron et al., 2023a,b). To make training feasible, we rely on 8-bit quantization of the base models (Dettmers et al., 2022), and use Low-Rank Adapters (LoRA) (Hu et al., 2021) as the only trainable parameters. Note that the usage of 8-bit quantization and LoRA are merely engineering considerations, and the calibration tuning framework remains applicable independently. We use HuggingFace Transformers (Wolf et al., 2020) and PyTorch (Paszke et al., 2019) for the implementation of these models.

Training Datasets. To allow our models to reach reasonable performance for subsequent analysis, we fine-tune on a large collection of commonly used datasets from literature (full list in Appendix A.2). All datasets are formatted as question-answers with the prompt containing multiple choices.

Training. Following the prescription of FLAN (Wei et al., 2021; Chung et al., 2022), we use a diverse combination of the training datasets mentioned above, and follow standard instruction tuning framework. For all our experiments, we use the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of 10^{-4}, a cosine decay schedule, and effective batch size $M = 32$. The training runs for $G = 10000$ with an initial linear warmup schedule for 1000 steps. For LoRA (Hu et al., 2021), we keep the default hyperparameters – rank $r = 8$, $\alpha = 32$, and dropout probability 0.1. For calibration-tuning, we use $\kappa = 1$. Each training run takes approximately 4 GPU days with NVIDIA V100 (32GB).

Evaluation Datasets. For all our evaluations, we use the Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2020) benchmark. MMLU is a suite of tasks that covers 57 subjects including STEM, humanities, and social sciences, providing a diverse test bed for generalization. Similar to training, we provide options in the prompt for the case of multiple-choice question-answering tasks. For the case of open-ended answer generation, we do not provide options in the prompt. As typical in literature, we report results with 5-shot prompting.

5.1 Base Instruction Tuning

Before calibration tuning, we construct instruction-tuned models that are able to generalize well on the MMLU benchmark in terms of accuracy.

We report the average accuracy over all 57 tasks of MMLU in Table 1. In addition, we also compute the LOGITS ECE, which corresponds to the evaluation of ECE in Eq. (2), where the confidence is estimated directly from the logits of the target token prediction. These numbers help us establish a baseline without the calibration-tuning intervention.

Note that our purpose here is not to achieve the state-of-the-art on MMLU, but only serve as a proxy for a reasonable instruction-tuned model that one might want to improve the calibration of.

Table 1: Instruction tuning (IT) on LLaMA-2 7b. BASE refers to the pretrained LLaMA-2 7b weights (Touvron et al., 2023b).

MODEL	ACC. ↑	LOGITS ECE
BASE	31.4%	13.8%
IT	49.3%	22.7%

Further, while the base model's LOGITS ECE may appear significantly better, the accuracy is significantly worse and the ECEs are therefore not directly comparable.

An important detail, that we expand later in Section 5.2 is the choice of training data distribution — we only instruction-tune on a subset of all the training datasets. The remainder of the datasets are used for calibration tuning.

5.2 Calibration Tuning

Building on top of the instruction-tuned (IT) model as summarized in Algorithm 1, we now apply calibration-tuning, while starting with the instruction-tuning checkpoint.

In Section 5.2, we report the query accuracy UQ ACC. which corresponds to the accuracy of the uncertainty query prompt from Figure 2. All subsequent usage of the term ECE corresponds to confidences estimated from the uncertainty query prompt as described in Section 4.2.

MODEL	QUERY ACC. ↑	ECE ↓
IT	53.0%	15.3%
CT	**64.0%**	**12.1%**

Notably, the uncertainty query prompt combined with our computation of the ECE already provides a strong improvement over the LOGITS ECE computation in Table 1. Nevertheless, we are further able to significantly improve the calibration of our instruction-tuned model. Therefore, calibration-tuning acts as a more effective uncertainty estimator. In Figure 6, we show a comparison of the ECE (c.f. Section 4.2) between both IT and CT among all the tasks of MMLU.

Before we compare calibration-tuning to baselines, we highlight two important design considerations when using calibration-tuning:

Choice of Data Distribution. We find that calibration tuning is marginally less effective when trained on the same data distribution as the underlying instruction-tuned model we are trying to improve the calibration of. In Table 2, we show that while the query accuracy is similar, using the same data distribution can lead to a degraded ECE.

Table 2: Calibration Tuning with the same data distribution (DATA DIST.) leads to marginally worse calibration.

DATA DIST.	QUERY ACC. ↑	ECE ↓
DIFFERENT	64.0%	12.1%
SAME	64.2%	13.2%

Restricting the amount of data we instruction-tune on can be marginally detrimental to accuracy, we next show that

Choice of Uncertainty Query Evaluation Model. By design, calibration-tuning modifies the same set of parameters as the starting model parameters, while using JSD regularization to keep the output distribution of the calibration-tuned model Q_θ similar to the starting model P_{θ_0}. As a consequence, the uncertainty estimator built via the uncertainty query is most effective for similar generating distributions as P_{θ_0}. To exploit this fact, we continue using our instruction-tuned model to generate the answers, while the uncertainty estimation is done by the calibration-tuned model. In Table 3, when the calibration-tuned model is used for both answer generation and uncertainty estimation, we denote it by SAME. If the calibration-tuned model only computes uncertainty, we denote it by DIFFERENT.

Table 3: When using CT, using the same model for generation and uncertainty estimation leads to a degraded ECE.

QUERY MODEL	ACC.	QUERY ACC. ↑	ECE ↓
DIFFERENT	49.3%	64.0%	12.1%
SAME	47.0%	64.2%	13.6%

We find that while the query accuracy (QUERY ACC.) remains similar, there is a drop in the calibration (ECE) indicating a drop in the performance of the uncertainty estimator. Subsequently, unless otherwise specified, we *do not* use SAME query model for evaluations.

Using a different model for uncertainty estimation increases the computational requirements. However, since we rely on LoRA (Hu et al., 2021) in practice, we incur only a marginal additional cost compared to using a single model.

We now discuss and compare against baselines that directly attempt to improve the calibration of

LLMs. A summary of numerical comparisons is provided in Table 4.

5.3 Multiple-Choice Question Answering

We now compare with existing baselines in literature. Each of the following methods presented in Table 4 either directly aim to improve calibration in LLMs or can be used towards estimating calibration. Unless otherwise noted, confidence of predictions for computation of calibration is done directly from the answer logits.

Table 4: Comparison with baselines on instruction-tuned LLaMA-2 7b (Touvron et al., 2023b) as discussed in Section 5.3, for multiple-choice question-answering tasks.

METHOD	ECE ↓
CS (JIANG ET AL., 2020)	22.8%
LTS (JIANG ET AL., 2020)	31.0%
LTS-MMLU (JIANG ET AL., 2020)	12.5%
CTX-C (ZHAO ET AL., 2021)	29.9%
CC (AZARIA AND MITCHELL, 2023)	20.3%
IT (WEI ET AL., 2021)	15.3%
CT (OURS)	**12.1**%

Candidate Answer Softmax Score (CS). As in Jiang et al. (2020), among a set of candidate targets $\mathcal{T}$, we pick the highest probability sequence $[S, T]$ for all $T \in \mathcal{T}$. Unlike calibration-tuning which estimates the uncertainty in a single forward pass, this approach requires $|\mathcal{T}|$ number of forward passes. In addition, requiring a set of candidate targets makes such an approach less broadly applicable.

Logit Temperature Scaling (LTS). (Jiang et al., 2020) Using a calibration dataset, we continue instruction-tuning, except only optimizing a single temperature parameter to scale the logits. Using the train set of MMLU for calibration (denoted by LTS-MMLU in Table 4) does indeed prove effective in improving calibration when tested on MMLU. However, when using the held out datasets from our training distribution, a setup closer to real world applications, we find that temperature scaling significantly deteriorates calibration of the model, unlike calibration tuning. This observation is in line with prior work where temperature scaling is not robust to distribution shifts between the calibration set and the test set.

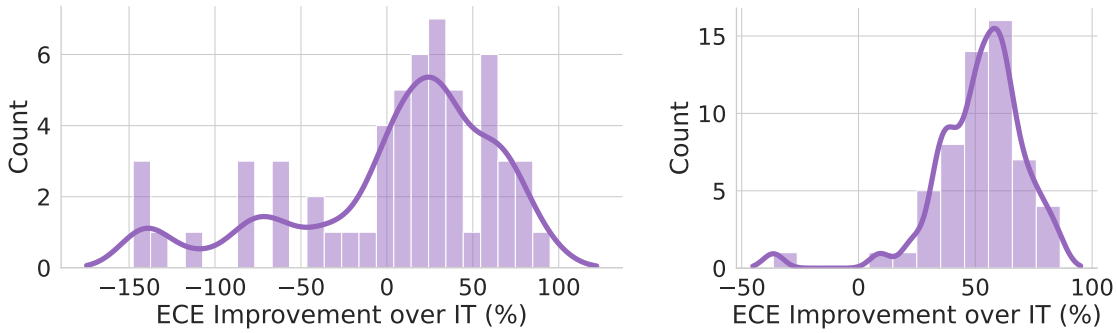
Contextual Calibration (CTX-C). (Zhao et al., 2021) This approach is a generalization of Platt scaling (Platt, 1999) for text inputs, where the scaling parameters are input-dependent. By first replacing an input sequence S, with a context-free input S_ϕ (e.g. the string "N/A"), we find the logit transform such that the target probabilities are uniform (e.g. logits are all zero). Subsequently, the same logit transform is applied to the original context. We construct an ensemble from $E = 3$ such context-free inputs to mitigate sensitivity to context-free inputs. Unlike single-shot estimation with calibration-tuning, this approach requires $E + 1$ forward passes for uncertainty estimation. Such an approach requires prompt engineering to be effective, which is less desirable.

Correctness Classifier (CC). Azaria and Mitchell (2023) use the last-layer features from a held-out set of sequences to train a linear classifier that predicts correctness probabilities, $p(\hat{C} = \texttt{True})$, which are then used to compute ECE. We find that finetuning the existing model is more effective for calibration than training an auxiliary model on top of frozen features, leading to better calibration in terms of ECE in our large-scale evaluation. These results suggest that calibration tuning might generalize more effectively than an auxiliary correctness classifier.

Instruction Tuning (IT). Finally, we also compare calibration tuning to vanilla instruction-tuning, while evaluating with the uncertainty query prompt as in Figure 2. In Figure 3(a), we show the distribution of the relative improvement that calibration tuning achieves over uncertainty tuning in the case of multiple-choice question answering. We see that a bulk of the mass lies to the positive side, indicating improvements across a broad set of MMLU tasks.

5.4 Open-Ended Generation

We additionally test the ability of calibration-tuning on open-ended generation. Notably, we do not explicitly tune the model on open-ended generations but apply the same calibration-tuning procedure on LLaMA-2 13b-chat model (Touvron et al., 2023b) only using multiple-choice question-answering. We perform this task to quantify how well multiple-choice calibration-tuning impacts open-ended performance, given that open-ended is the more widely used application. In future work, we plan to expand to open-ended calibration-tuning.



(a) Multiple Choice Question-Answering (b) Open-Ended Question-Answering

Figure 3: We plot the relative performance of calibration tuning (CT) w.r.t. instruction tuning (IT) in terms of calibration as measured by ECE over all 57 MMLU tasks. A positive relative performance indicates improved (lower) ECE. Noticeably, a bulk of the mass lies to the positive half for both (a) multiple choice question answering and (b) open-ended answer generation, indicating that calibration tuning does generalize effectively. See Appendix B for a precise breakdown comparison of uncertainty query accuracies and ECEs across all tasks.

In Figure 2, we provide an example of the prompt used by GPT 3.5 Turbo for grading the (semantic) similarity between the ground truth answer A and model's generated answer $\hat{A}$.

Surprisingly, calibration-tuning on MCQ QA also improves calibration for open-ended generation without having been explicitly trained for this format. In Figure 3 we visualize the relative calibration improvement over instruction tuning across the MMLU benchmark suite, showing an improvement over all but one of the 57 tasks.

We highlight a small caveat. For some MMLU tasks, the query accuracy in Figure 7 remains low, i.e. the calibration-tuned model still lacks a good understanding of what topics it does not know. Combined with conservative probability estimates as a consequence of calibration tuning, we end up with better calibration. For such tasks, any calibration improvements are less significant, and we hope in future work to address this with calibration-tuning on open-ended answer generation.

6 Discussion

We have setup *calibration tuning* as a general purpose method that relies on existing training data to improve the calibration of LLMs. By using an uncertainty query prompt, we are able to provide a definition of calibration for LLMs that is applicable in broad contexts. For question-answering tasks, we show that calibration tuning is able to generalize out-of-distribution to the MMLU tasks when evaluated with LLaMA-2 7b. Surprisingly, calibration tuning with multiple-choice question-answering also improves the calibration of the base LLaMA-2 13b-chat model for open-ended answer generation.

Future Work. Calibration tuning sets us up for exciting broader scoped future work. While we have shown that calibration tuning on question-answering provides an improved calibration even for open-ended generation, we can improve further by explicitly instruction-tuning our models on open-ended generations. Such tuning will allows us to break away from biases specific to multiple-choice question answering.

RLHF (Ouyang et al., 2022) is now a standard tool to align language generations with human preferences. Prior work (Kadavath et al., 2022) hints that RLHF models may suffer from degraded calibration, and provides us an exciting opportunity to use a general-purpose method that improves calibration of such models too.

Limitations. A key ingredient of calibration tuning is the uncertainty query. While we currently only use one format for the prompt (Section 4.2), it is likely that better prompt engineering allows us to significantly improve model's calibration. Further, despite calibration tuning, we do not necessarily expect the language model to follow the rules of probability across the space of semantically correct answers. We hypothesize that the model can be nudged towards such characteristics by biasing the loss, such as an unsupervised consistency loss (Burns et al., 2022).

Overall, calibration tuning remains a promising framework to develop further.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367. Association for Computational Linguistics.

Amos Azaria and Tom M. Mitchell. 2023. The internal state of an llm knows when its lying. *ArXiv*, abs/2304.13734.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. Piqa: Reasoning about physical commonsense in natural language. *ArXiv*, abs/1911.11641.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing*.

Mark Braverman, Xinyi Chen, Sham M. Kakade, Karthik Narasimhan, Cyril Zhang, and Yi Zhang. 2019. Calibration, entropy rates, and memory in language models. In *International Conference on Machine Learning*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *ArXiv*, abs/2005.14165.

Collin Burns, Hao-Tong Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *ArXiv*, abs/2212.03827.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. 2022. A close look into the calibration of pre-trained language models. *ArXiv*, abs/2211.00151.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *ArXiv*, abs/1905.10044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. *9th International Conference on Learning Representations, ICLR*.

Andrew S. Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2011. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *International Workshop on Semantic Evaluation*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. In *Conference on Empirical Methods in Natural Language Processing*.

Zhengbao Jiang, J. Araki, Haibo Ding, and Graham Neubig. 2020. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Saurav Kadavath, Tom Conerly, Amanda Askell, T. J. Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zachary Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, John Kernion, Shauna Kravec, Liane Lovitt, Kamal

Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom B. Brown, Jack Clark, Nicholas Joseph, Benjamin Mann, Sam McCandlish, Christopher Olah, and Jared Kaplan. 2022. Language Models (Mostly) Know What They Know. *ArXiv*, abs/2207.05221.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *North American Chapter of the Association for Computational Linguistics*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *ArXiv*, abs/2302.09664.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *International Conference on Computational Linguistics*.

Stephanie C. Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

David John Cameron MacKay. 2004. Information theory, inference, and learning algorithms. *IEEE Transactions on Information Theory*, 50:2544–2545.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Ann Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks. *ArXiv*, abs/2106.07998.

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip H. S. Torr, and Puneet Kumar Dokania. 2020. Calibrating deep neural networks using focal loss. *ArXiv*, abs/2002.09437.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? *ArXiv*, abs/1906.02629.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *ArXiv*, abs/1910.14599.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems*.

John Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *ArXiv*, abs/1907.10641.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *ArXiv*, abs/1811.00937.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open

and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Dustin Tran, Jeremiah Zhe Liu, Michael W. Dusenberry, Du Phan, Mark Collier, Jie Jessie Ren, Kehang Han, Z. Wang, Zelda E. Mariet, Huiyi Hu, Neil Band, Tim G. J. Rudner, K. Singhal, Zachary Nado, Joost R. van Amersfoort, Andreas Kirsch, Rodolphe Jenatton, Nithum Thain, Honglin Yuan, E. Kelly Buchanan, Kevin Murphy, D. Sculley, Yarin Gal, Zoubin Ghahramani, Jasper Snoek, and Balaji Lakshminarayanan. 2022. Plex: Towards reliability using pretrained large model extensions. *ArXiv*, abs/2207.07411.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *ArXiv*, abs/1707.06209.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *ArXiv*, abs/2306.13063.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Annual Meeting of the Association for Computational Linguistics*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*.

Hanlin Zhang, Yi-Fan Zhang, Yaodong Yu, Dhruv Madeka, Dean Foster, Eric Xing, Hima Lakkaraju, and Sham Kakade. 2023. A study on the calibration of in-context learning. *arXiv preprint arXiv:2312.04021*.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: Expressions of overconfidence and uncertainty in language models. *ArXiv*, abs/2302.13439.

Appendix

Table of Contents

A Method

A.1 Algorithm

The complete general framework for calibration tuning (CT) is summarized in Algorithm 1.

Algorithm 1: Calibration-Tuning (CT)

Input : Dataset $\mathcal{U}$, Batch size M, Number of gradient steps G, Regularization weight κ

repeat
- Sample minibatch $\mathcal{U}_M = \{[S_j, T_j]\}_{j=1}^{M}$
- Generate outputs $\widehat{\mathcal{T}}_M = \{\widehat{T}_j\}_{j=1}^{M}$
- Construct uncertainty queries $\widehat{\mathcal{U}}_M = \{\widehat{U}_j = [S_j, \widehat{T}_j, Q_j, R_j]\}_{j=1}^{M}$
- Compute loss $\mathcal{L}_{\text{CT}}(\theta; \kappa, \theta_0)$ in Eq. (4)
- Update using gradients $\nabla_\theta \mathcal{L}_{\text{CT}}(\theta; \kappa, \theta_0)$

until *G updates have been completed*

A.2 Training Data

We reserve the following datasets for training.

- AI2 Reasoning Challenge (ARC) (Clark et al., 2018),
- Boolean Questions (BoolQ) (Clark et al., 2019),
- CommonsenseQA (Talmor et al., 2019),
- CosmosQA (Huang et al., 2019),
- HellaSwag (Zellers et al., 2019),
- MathQA (Amini et al., 2019),
- Recognizing Textual Entailment (RTE/SNLI) (Bowman et al., 2015),
- Adversarial NLI (Nie et al., 2019),
- OpenBookQA (Mihaylov et al., 2018),

- PIQA (Bisk et al., 2019),
- SciQ (Welbl et al., 2017),
- The CommitmentBank (CB) (de Marneffe et al., 2019),
- Multi-Sentence Reading Comprehension (MultiRC) (Khashabi et al., 2018),
- Choice of Plausible Alternatives (CoPA) (Gordon et al., 2011),
- TREC (Li and Roth, 2002),
- Adversarial Winograd (Winogrande) (Sakaguchi et al., 2019).

B Additional Results

B.1 MMLU Task Breakdown for Multiple-Choice Question Answering

We report the breakdown of uncertainty query accuracy and ECE on all MMLU tasks in Figures 4 and 5.

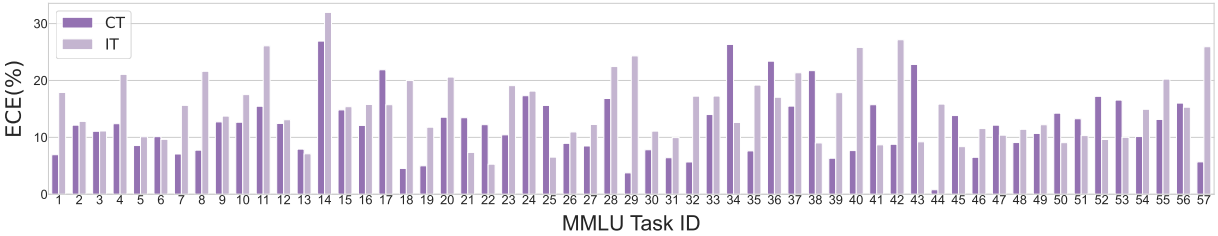


Figure 4: Calibration Tuning (CT) improves ECE (lower is better) on 39 out of 57 tasks from the MMLU benchmark suite (IDs assigned alphabetically for visualization) (Hendrycks et al., 2020), when compared to instruction tuning (IT). This breakdown validates that the calibration improvements we see in Section 5.2 are in fact meaningful. See Figure 5 for the corresponding graph of query accuracies.

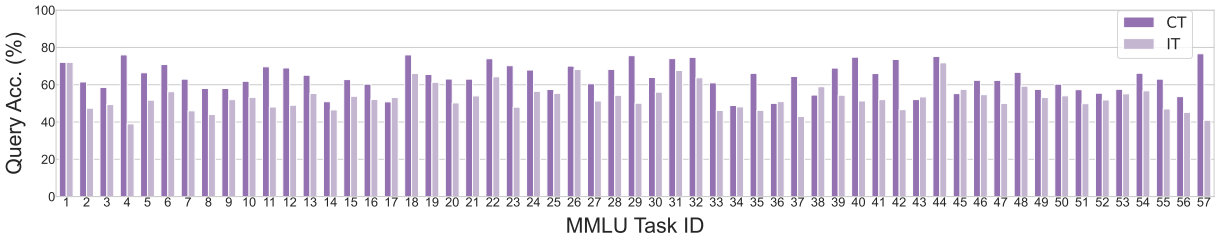


Figure 5: Calibration Tuning (CT) maintains or improves the accuracy of the uncertainty query on 52 out of 57 tasks from the MMLU benchmark suite (IDs assigned alphabetically for visualization) (Hendrycks et al., 2020), when compared to instruction tuning (IT). See Figure 4 for the corresponding graph of query accuracies.

B.2 MMLU Task Breakdown for Open-Ended Answer Generation

We provide a similar breakdown of uncertainty query accuracy and ECEs in Figures 6 and 7 for open-ended answer generation.

C Comparison of Open-Ended Evaluation Grading Techniques

We conducted an analysis of the methods outlined in 4.1 for open-ended evaluation. First, the base LLaMA-2 13b-chat model was prompted with questions from the following test subsets of MMLU: World Religions, Philosophy, Anatomy, High School Chemistry and Elementary School Math. The questions were stripped of their multiple-choice options before being supplied to the model.

A response was generated by the model via greedy decoding and this response was compared to the ground truth answer. The grading methods tested were Human, Substring Match, GPT 3.5 Turbo, and GPT 4.

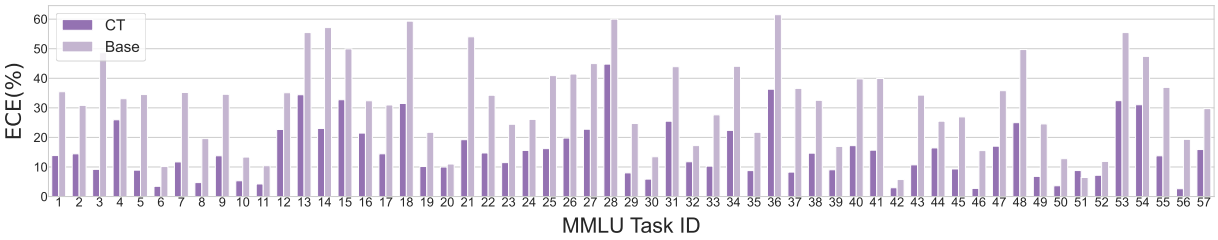


Figure 6: Calibration Tuning (CT) on multiple-choice questions appears to generalize to open-ended evaluations. Calibration improves over the base LLaMA-2 13b-chat model on all but one of the MMLU tasks (Hendrycks et al., 2020). MMLU Task IDs are assigned alphabetically for visualization. See Figure 7 for the corresponding uncertainty query accuracies.

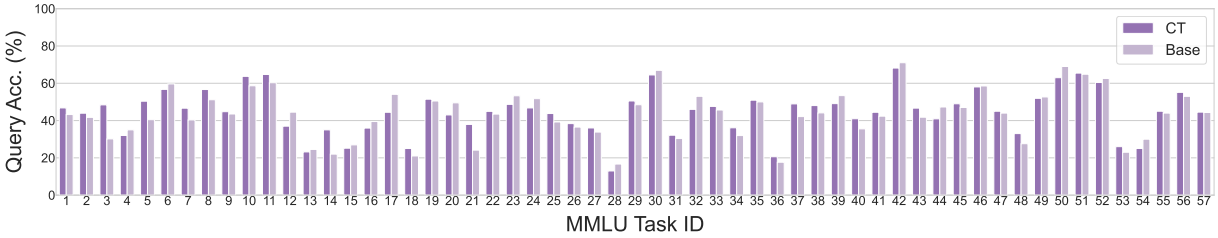


Figure 7: Calibration Tuning (CT) tends to perform slightly worse on some MMLU tasks in terms of the uncertainty query accuracy when evaluating the LLaMA-2 13b-chat model. This degraded performance is partly attributed to the lack of fine-tuning on open-ended answer generation. See Figure 6 for the corresponding ECE.

The humans (a subset of our authors) were tasked to judge if the model response was essentially equivalent to the ground truth. For substring match, equivalence was determined by simply checking whether the ground truth answer existed as a substring within the model response. For GPT 3.5 Turbo and GPT 4, the models were supplied with the question, the ground truth, and the base model response, as well as a prompt indicating they should determine essential equivalence - see Figure 2.

MMLU Subset	Substring Match	GPT3.5	GPT4
World Religions	21.6%	6.4%	1.8%
Philosophy	22.8%	2.3%	14.5%
Anatomy	13.3%	14.8%	1.5%
Chemistry	13.8%	5.4%	1.0%
Math	12.4%	14.8%	3.7%
Average	**16.8%**	**8.7%**	**4.5%**

Table 5: Absolute differences in accuracy % for the different grading methods vs human estimated accuracy. A lower value corresponds to an accuracy estimate closer to the human estimate.

We recorded the binary decision on correctness for each query and response by each of the grading methods above. Taking the human scores as the gold standard of correctness, we computed the model accuracy for each subset, and then derived the absolute error in estimate of model accuracy by each of the other grading methods. These are displayed in Table 5. We see that GPT4 is a better estimator of human-judged correctness than GPT 3.5 Turbo, which in turn is substantially better than substring match; although there is some variance on a per-subset basis. For expediency of processing time and cost, we chose to use GPT 3.5 Turbo in this paper.