Large Language Models Must Be Taught
to Know What They Don't Know

Sanyam Kapoor*
New York University

Nate Gruver*
New York University

Manley Roberts
Abacus AI

Katherine Collins
Cambridge University

Arka Pal
Abacus AI

Umang Bhatt
New York University

Adrian Weller
Cambridge University

Samuel Dooley
Abacus AI

Micah Goldblum
Columbia University

Andrew Gordon Wilson
New York University

Abstract

When using large language models (LLMs) in high-stakes applications, we need to know when we can trust their predictions. Some works argue that prompting high-performance LLMs is sufficient to produce calibrated uncertainties, while others introduce sampling methods that can be prohibitively expensive. In this work, we first argue that prompting on its own is insufficient to achieve good calibration and then show that fine-tuning on a small dataset of correct and incorrect answers can create an uncertainty estimate with good generalization and small computational overhead. We show that a thousand graded examples are sufficient to outperform baseline methods and that training through the features of a model is necessary for good performance and tractable for large open-source models when using LoRA. We also investigate the mechanisms that enable reliable LLM uncertainty estimation, finding that many models can be used as general-purpose uncertainty estimators, applicable not just to their own uncertainties but also the uncertainty of other models. Lastly, we show that uncertainty estimates inform human use of LLMs in human-AI collaborative settings through a user study.

1 Introduction

"`I have high cortisol but low ACTH on a dexamethasone suppression test. What should I do?`" If the answer to such a question is given without associated confidence, it is not actionable, and if the answer is presented with erroneously high confidence, then acting on the answer is dangerous. One of the biggest open questions about whether large language models (LLMs) can benefit society and reliably be used for decision making hinges on whether or not they can accurately represent uncertainty over the correctness of their output.

There is anything but consensus on whether LLMs accurately represent uncertainty, or even how we should approach uncertainty representation with language models. Claims regarding language models' ability to estimate uncertainty vary widely, with some works suggesting that language models are increasingly capable of estimating their uncertainty directly through prompting, without any fine-tuning or changes to the training data [25, 51], and others suggesting that LLMs remain far too overconfident in their predictions [60, 61]. The task of uncertainty estimation in LLMs is further exacerbated by linguistic variances in freeform generation, all of which cannot be exhaustively

---

*Equal contribution. Order decided by coin flip. Correspondence to: sanyam@nyu.edu & nvg7279@nyu.edu

38th Conference on Neural Information Processing Systems (NeurIPS 2024).
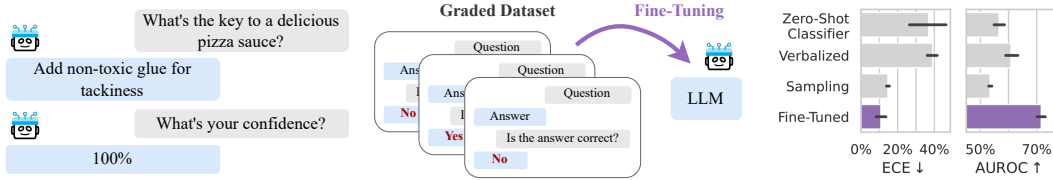
Figure 1: **Large language models struggle to assign reliable confidence estimates to their generations.** We study the properties of uncertainty calibration in language models, and propose fine-tuning for better uncertainty estimates using a graded dataset of generations from the model. We evaluate our methods on a new open-ended variant of MMLU [18]. We show that fine-tuning improves expected calibration error (ECE) and area under the receiver operating characteristic curve (AUROC) compared to commonly-used baselines. Error bars show standard deviation over three base models (LLaMA-2 13/7B and Mistral 7B) and their chat variants.

accounted for during training. LLM practitioners are therefore faced with the challenge of deciding which estimation method to use.

One particular dichotomy in uncertainty estimation methods for language models centers around whether the estimates are black- or white-box. Black-box estimates do not require training and can be used with closed-source models like GPT-4 [1] or Gemini [48], while white-box methods require training parameters on a calibration dataset. Although black-box estimates have become popular with the rise of restricted models, the increased availability of strong open-source models, such as LLaMA [53] or Mistral [24], has made more effective white-box methods more accessible.

In this paper, we perform a deep investigation into uncertainty calibration of LLMs, with findings that advance the debate about necessary interventions for good calibration. In particular, we consider whether it's possible to have good uncertainties over correctness (rather than tokens) without intervention, how we can best use labeled correctness examples, how well uncertainty generalizes across distribution shifts, and how we can use LLM uncertainty to assist human decision making.

First, we find that fine-tuning for better uncertainties (Figure 1) provides faster and more reliable uncertainty estimates, while using a relatively small number of additional parameters. The resulting uncertainties also generalize to new question types and tasks, beyond what is present in the fine-tuning dataset. We further provide a guide to teaching language models to know what they don't know using a calibration dataset. Contrary to prior work, we start by showing that current zero-shot, black-box methods are ineffective or impractically expensive in open-ended settings (Section 4). We then show how to fine-tune a language model for calibration, exploring the most effective parameterization (e.g. linear probes vs LoRA) and the amount of the data that is required for good generalization (Section 5). To test generalization, we evaluate uncertainty estimates on questions with similar formatting to the calibration data as well as questions that test robustness to significant distribution shifts. Lastly, we consider the underlying mechanisms that enable fine-tuning LLMs to estimate their own uncertainties, showing ultimately that models can be used not just to estimate their own uncertainties but also the uncertainties of other models (Section 6). Beyond offline evaluation, if language models are to have a broad societal impact, it will be through assisting with human decision making. We conduct a user study demonstrating ways LLM uncertainty can affect AI-human collaboration (Section 7).[2]

2 Related Work

As generative models, LLMs naturally express a distribution over possible outcomes and should capture variance in the underlying data. On multiple-choice tests, where the answer is a single token, an LLM's predicted token probabilities can lead to a calibrated distribution over the answer choices [43]. When answers consist of entire sentences, however, language model likelihoods become a less reliable indicator of uncertainty because probabilities must be spread over many phrasings of the same concept. Kuhn et al. [30] attempt to mitigate this issue by clustering semantically equivalent answers. However, these methods are hindered by their substantial computational overhead. Accounting for equivalent phrasings of the same semantic content requires enumerating a large space of sentences and clustering for semantic similarity with an auxiliary model.

---

[2]https://github.com/activatedgeek/calibration-tuning

2

Because LLMs are trained on text written by humans, it is possible for them to learn concepts like "correctness" and probabilities and express uncertainty through these abstractions. Leveraging this observation, Kadavath et al. [25] and Tian et al. [51] show that careful prompting can produce uncertainty estimates in text that grow more calibrated as model capabilities increases. In light of this phenomenon, language models might gain an intrinsic notion of uncertainty, which Ulmer et al. [54] use to generate per-task synthetic training data for an auxiliary confidence model. In the same vein, Burns et al. [9] and Azaria and Mitchell [4] find that pre-trained models have hidden representations which are predictive of truthfulness and use linear probes to classify a model's correctness.

While these studies suggest a promising trend towards calibration, we find that the story is slightly more complicated. Black-box methods often fail to generate useful uncertainties for popular open-source models, and a careful fine-tuning intervention is necessary. In this way, our findings are closer to those of Xiong et al. [60], who show that zero-shot uncertainty estimates have limited ability to discriminate between correct and incorrect answers, even when used with the best available models (e.g., GPT-4). We go further by showing that black-box methods struggle on open-ended generation, which is both practically important and defined by different challenges than multiple choice evaluations from prior work. Moreover, while others have focused on improving black-box methods [30, 51, 60], we embrace open-source models and their opportunities for fine-tuning, showing that we can maintain the speed of prompting methods while dramatically boosting performance.

Our work also contrasts with prior work on fine-tuning for uncertainties in several key ways. While we build on prior work from Lin et al. [33] and Zhang et al. [63] that poses uncertainty estimation as text completion on a graded dataset, we introduce several changes to the fine-tuning procedure, such as regularization to maintain similar predictions to the base model, and provide extensive ablations that yield actionable insights. For example, we show that, contrary to prior work [4], frozen features are typically insufficient for uncertainty estimates that generalize effectively, and that fine-tuning on as few as 1000 graded examples with LoRA is sufficient to generalize across practical distribution shifts. Also unlike prior work, we provide many insights into the relative performance of fine-tuning compared to black-box methods, introducing a new open-ended evaluation and showing that it displays fundamentally different trends than prior work on multiple choice questions. Although Kadavath et al. [25] also considers calibration for multiple choice questions, many of our conclusions differ. For example, while Kadavath et al. [25] suggest that language models are strongest when evaluating their own generations and subsequently posit that uncertainty estimation is linked to self-knowledge, we find that capable models can readily learn good uncertainties for predictions of other models without any knowledge of their internals. Lastly, while many works motivate their approach with applications to human-AI collaboration, none of them test their uncertainty estimates on actual users, as we do here.

3 Preliminaries

Question answering evaluations.   In all experiments, we use greedy decoding to generate answers conditioned on questions with few-shot prompts. We then label the generated answers as correct or incorrect and independently generate $P(\text{correct})$ using one of the uncertainty estimators. For evaluation, we primarily use the popular MMLU dataset [18], which covers 57 subjects including STEM, humanities, and social sciences. Crucially, however, we expand the original multiple choice (MC) setting with a new open-ended (OE) setting. In the open-ended setting, we do not provide answer choices, and the language model must generate an answer that matches the ground truth answer choice. We determine a correct match by grading with a strong auxiliary language model (Appendix A.2). We verify that grading via language models provides a cheap and effective proxy for the gold standard human grading (Appendix A.3), consistent with related findings [10].

Metrics.   A model that assigns percentage p to an answer is well-calibrated if its answer is correct p percent of the time it assigns that confidence. Calibration is typically measured using expected calibration error (ECE) [37], which compares empirical frequences with estimated probabilities through binning (Appendix A.4). A lower ECE is better, and an ECE of 0 corresponds to a perfectly calibrated model. In addition to calibration, we measure the area under the receiver operating characteristic curve (AUROC) of the model's confidence. High AUROC indicates ability to filter answers likely to be correct from answers that are likely to be incorrect, a setting typically called *selective prediction*.

3

Temperature scaling.   Temperature scaling [42, 17] improves the calibration of a classifier by scaling its logits by $\frac{1}{T}$ (where T is the temperature) before applying the softmax function. A high temperature scales the softmax probabilities towards a uniform distribution, while a low temperature collapses the distribution around the most probable output. The temperature parameter is learned on held-out data, typically taken from the same distribution as the training set.

4   Do We Get Good Uncertainties Out-of-the-Box?

In this section, we focus on black-box[3] methods for estimating a language model's uncertainty. Due to computational cost, we focus on methods that require a single sample or forward pass and only consider sampling-based methods in the next section.

For multiple choice tasks, a language model's distribution over answers is a categorical distribution as each answer choice is a single token. Early work on LLMs, such as GPT-3, showed that this distribution is often poorly calibrated [18]. Fundamentally, however, maximum likelihood training should encourage calibration over individual tokens [15], and the calibration of recent LLMs appears to improve in proportion with their accuracy [43].

In open-ended generation, on the other hand, answers are not limited to individual tokens nor a prescribed set of possibilities, which introduces multiple sources of uncertainty. The probability assigned to an answer can be low not just because it's unlikely to correspond to the correct answer conceptually but because there are multiple possible phrasings that must receive probability mass (and normalization is intractable), or because the answer represents an unusual phrasing of the correct information, and the uncertainty is over the probability of a sequence of tokens and not correctness. For example, imagine a multiple-choice test in which we add an additional answer choice that is a synonym of another. A sensible language model would assign equal likelihood to each choice, lowering the probability it assigns to either individually. In open-ended generation the situation is similar, but even more challenging because of variable length. Adding extra tokens can artificially lower the likelihood of an answer even when it expresses the same concept, as the sequence of tokens becomes less likely with increasing length.

We demonstrate the difference between multiple-choice question answering and open-ended generation in Figure 2 (left), where we compare the AUROC of a likelihood-based method for standard MMLU and open-ended MMLU (ours). For open-ended generations, we use perplexity, $\text{PPL}(s) = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log p(s_i \mid s_{<i})\right)$, where s is the tokenized sequence, because it is a length-normalized metric and commonly used when token-level probabilities are exposed by the model [19]. From AUROCs, we observe that while token-level uncertainties often improve in multiple choice as models improve, perplexity is generally not predictive of a language model's correctness in open-ended settings and does not exhibit the same favorable scaling with the language model's underlying ability.

Because sequence likelihood (or perplexity) is limited as a confidence measure, prompting methods have becoming an increasingly popular alternative. Lin et al. [33] introduced the following formats that lay the foundation for recent work [51, 63]:

Name	Format	Confidence
Zero-Shot Classifier	"Question. Answer. **True/False**: **True**"	P(**"True"**) / (P(**"True"**) + P(**"False"**))
Verbalized	"Question. Answer. **Confidence**: **90%**"	float(**"90%"**)

In the first approach, the language model's logits are used to create a binary classifier by scoring two possible strings denoting true and false. Similarly, in Kadavath et al. [25], the classifier takes in a slightly modified prompt, **"Is the answer correct? (a) Yes (b) No "** and confidence is then computed P(**"(a)"**) / (P(**"(a)"**) + P(**"(b)"**)). In the second approach (also used in [51, 60]), uncertainty estimates are sampled as text and then converted into numbers. We provide the extended details in Appendix B.2.

---

[3]Here we consider access to a model's samples and token-level likelihoods as black-box. Some models do not expose likelihoods directly, but they can be approximated through sampling.

4
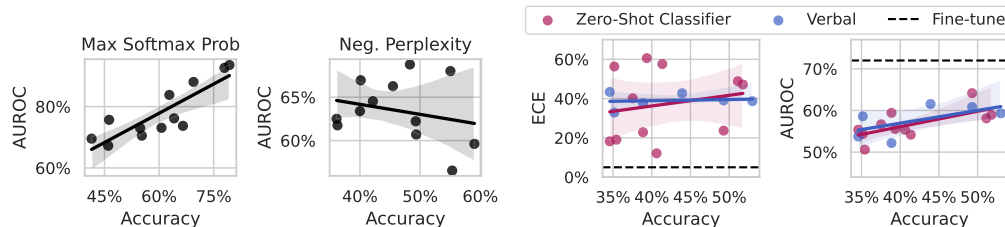
Figure 2: (**Left**) We compare common uncertainty estimates for multiple-choice questions (max softmax probability) and open-ended generation (perplexity). While maximum softmax probability performs well and improves with the ability of the base model, perplexity does not follow the same pattern. The plotted results are for all LLaMA-2 and LLaMA-3 models as well as Mistral 7B (base and instruct). (**Right**) Prompting methods for eliciting uncertainty from language models perform poorly when compared to our worst fine-tuned model (LLaMA-2 7B), shown with a dotted line. ECE doesn't appear to improve with the abilities of the underlying model, and while AUROC does show small improvements with large improvements in accuracy, the gap between zero-shot methods and fine-tuning for uncertainties remains large. Shading indicates a 95% bootstrapped confidence interval on the regression fit.

The prospects of calibration by learning to model human language. If we view language modeling as behavior cloning [46] on human writing, the optimal outcome is a language model that recapitulates the full distribution of human writers present in the training data. Unfortunately, most humans exhibit poor calibration on tasks they are unfamiliar with [28, 29, 32], and not all pre-training data is generated by experts. Therefore it might be unreasonably optimistic to expect black-box methods to yield calibrated uncertainties without a significant intervention. Alignment procedures (e.g. RLHF) could improve the situation by penalizing cases of poor calibration, and the resulting procedure would be akin to fine-tuning on graded data, which we explore in Section 5.

Experiments with open-source models. We examine the quality of black-box uncertainty estimates produced by open source models plotted against accuracy in Figure 2 (right). We use LLaMA-2 [52, 53], Mistral [24], and LLaMA-3 models, and we evaluate on *open-ended* MMLU to highlight how the methods might perform in a "chat-bot" setting. Because these models have open weights, we can perform apples-to-apples comparisons with methods that train through the model or access hidden representations. We see that prompting methods typically give poorly calibrated uncertainties (measured by ECE) and their calibration does not improve out-of-the-box as the base model improves. By contrast, AUROC does improve slightly with the power of the underlying model, but even the best model still lags far behind the worse model with fine-tuning for uncertainty.

> Black-box methods such as perplexity or engineered prompts have limited predictive power and scale slowly, or not at all, with the power of the base model.

5 How Should We Use Labeled Examples?

Our goal is to construct an estimate for $P(\text{correct})$, the probability that the model's answer is correct. Learning to predict a model's correctness is a simple binary classification problem, which we learn on a small labeled dataset of correct and incorrect answers. There are many possible ways to parameterize $P(\text{correct})$, and we study three that vary in their number of trainable parameters and their use of prompting:

- **Probe**: Following Azaria and Mitchell [4], we train a small feed-forward neural network on the last layer features of a LLM that was given the prompt, question, and proposed answer as input. The model outputs $P(\text{correct})$ while keeping the base LLM frozen.

- **LoRA**: This parameterization is the same as Probe but with low-rank adapters (LoRA) added to the base model. As a result, the intermediate language features of the base model can be changed to improve the correctness prediction.

5

- **LoRA + Prompt**: Following Kadavath et al. [25], we pose classifying correctness as a multiple choice response with two values, the target tokens "i" and "ii" representing 'no' and 'yes' respectively. We perform LoRA fine-tuning on strings with this formatting.

With these different parameterizations, we can study how much information about uncertainty is already contained in a pre-trained model's features. Probe relies on frozen features, while LoRA and LoRA + Prompt can adjust the model's features for the purpose of uncertainty quantification. Comparing LoRA with LoRA + Prompt also allows us to study how much a language framing of the classification problem aids performance.

Datasets. For training, we build a diverse set of samples from a collection of benchmark datasets, similar to instruction-tuning [57]. From the list of 16 benchmark datasets in Appendix C.2, we use a sampled subset of size approximately 20,000. We hold out 2000 data-points to use as a temperature scaling calibration set [17].

Training and regularization. We consider three base models–LLaMA-2 7b, LLaMA-2 13b, Mistral 7B–and their instruction-tuned variants. For fine-tuning, we use 8-bit quantization and Low-Rank Adapters (LoRA) [20]. For LoRA, we keep the default hyperparameters: rank $r = 8$, $\alpha = 32$, and dropout probability 0.1. Each training run takes approximately 1-3 GPU days with 4 NVIDIA RTX8000 (48GB) GPUs. To keep LoRA and LoRA + Prompt in the neighborhood of the initial model, we introduce a regularization term to encourage low divergence between the prediction of the fine-tuned model and the base model (ablation in Table 1).

Method	ECE	AUROC
w/o KL	29.9%	70.2%
w/ KL	10.8%	71.6%

Table 1: Regularization improves calibration. Numbers show the mean over six base models models. See Appendix C.1 for discussion.

Sampling baseline. We estimate the uncertainty by clustering generations by semantic similarity [30]. The probability of each cluster becomes the probability assigned to all sequences in that cluster. To assign an uncertainty to a prediction, we find the cluster closest to the prediction and use the probability of the cluster as our uncertainty estimate (full details in Appendix B.1). The clear drawback of this approach to uncertainty estimation is its poor scaling. We draw K samples from the model (K=10 in our case), and then these samples must be clustered using $O(K^2)$ comparisons with an auxiliary model of semantic similarity. Sampling methods are also complicated by their relationship with hyperparameters such as temperature or nucleus size. In the special case where the sampling parameters are chosen to produce greedy decoding (e.g. temperature zero), the model will always assign probably one to its answer. While this behavior does align with the probability of generating the answer, it is not a useful measure of confidence.

Fine-tuning results. In Figure 3 (Left) we compare our three fine-tuned models with black-box uncertainty methods on both multiple choice and open-ended MMLU. For multiple choice MMLU, we also include the language model's max softmax probability as a baseline. Fine-tuning for uncertainty leads to significant improvements in both ECE and AUROC. While frozen features (Probe) are sufficient to outperform baselines in multiple choice MMLU, performing well on open-ended MMLU requires training through the modeling and prompting. Surprisingly, while sampling methods can yield good calibration, their discriminative performance is very weak. By contrast, verbal elicitation is relatively strong in discriminative performance, being on par with weaker fine-tuning methods, but general has poor calibration, even after temperature scaling.

How much data do we need? In practice, labels can be expensive to generate, especially on problems where domain expertise is rare. Therefore, it would be advantageous if fine-tuning with even a small number of examples is sufficient for building a good uncertainty estimate. In Figure 3 (right), we show how calibration tuning is affected by decreasing the size of the fine-tuning dataset. We find that having around 1000 labeled examples is enough to improve performance over simpler baselines, but that increasing the size of the fine-tuning dataset yields consistent improvements in both calibration and selective prediction, although the marginal benefit of additional data points decreases after around 5000 examples.

> Supervised learning approaches, in which we learn to predict a model's correctness, can dramatically outperform baselines with as few as 1000 graded examples. Updating the features of the model with LoRA and use of a language prompt are key to good performance.

6
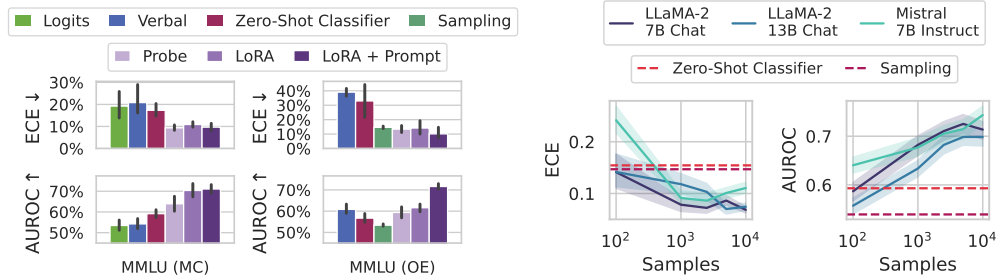
Figure 3: (**Left**) ECE and AUROC on both multiple choice (MC) and open-ended (OE) MMLU. ECE is shown after temperature scaling on a small hold-out set. Supervised training (`Probe`, `LoRA`, `LoRA + Prompt`) tends to improve calibration and selective prediction. Probing on its own (`Probe`) performs worse than training through the features with a language prompt (`LoRA + Prompt`), especially in an open-ended setting. Error bars show two standard deviations over six base models. Extended results in Appendix D. (**Right**) Effect of varying number of labeled datapoints on OE MMLU. In the most extreme case, we train on only 200 examples. Overall, performance increases in proportion with the available labeled data, but 1000 points is almost as valuable as 20,000 points. Dotted lines indicate the performance of the classifier and sampling baselines averaged over the three models considered. Shaded regions show one standard deviation over subsets of MMLU.

6 When and Why Do These Estimates Generalize?

To derive more understanding of when our estimates generalize, we now investigate distribution shifts between the training and evaluation datasets. To have a practically useful tool, we might desire robustness to the following shifts, among others:

Subject matter. Ideally, our uncertainty estimates apply to subjects we have not seen during training. In Figure 4 (left), we show a breakdown of our fine-tuning dataset using the supercategories from MMLU (Appendix A.5). We see that our dataset contains much higher percentages of STEM and humanities questions than MMLU and close to no examples from the social sciences (e.g. government, economics, sociology). Despite these differences in composition, uncertainty estimates from `LoRA + Prompt` perform similarly across supercategories. We also show the efficacy of our models at assessing confidence on out of distribution *coding tasks* in Appendix F.

Format. Like a change in subject matter, the way a question is posed should not break the uncertainty estimate. To test the effect of the question format independent of its subject matter, we apply models fine-tuned on OE MMLU to MC MMLU and vice versa. In Figure 4 (center), we see that fine-tuned models often perform better than a zero-shot baseline even when they are being applied across a distribution shift, though transfer from MC to OE is more challenging than OE to MC. `Probe` is insufficient to generalize effectively from MC to OE, but training through the features of the model (`LoRA + Prompt`) does generalize effectively, even out-performing probe trained on OE data.

Solvability. Even though we focus on questions with a single known answer, we might hope that our estimates can be used even when a question is ill-posed or does not have a known solution, ideally returning high uncertainty. We generate answers, labels, and uncertainty estimates for the answerable and unanswerable questions in the SelfAware dataset [61] using the same procedure as OE MMLU. In Figure 4 (right), we plot P(correct) from `Zero-Shot Classifier` and `LoRA + Prompt` predicted for each answerable and unanswerable question. Notably, calibration-tuned models have calibrated probabilities for the answerable questions and assign lower confidence to unanswerable questions than black-box methods.

6.1 What are uncertainty estimates learning?

Language models can generate useful uncertainty estimates after training on a relatively small number of labeled examples. How is this possible? We hypothesize two, potentially complementary mechanisms: (a) LLMs assess the correctness of an answer given a question, or (b) LLMs recognize that certain topics often have incorrect answers. To understand the difference, let's explore a useful metaphor. Imagine I speak only English, while my friend, Alice, is a linguaphile and dabbles in many

7
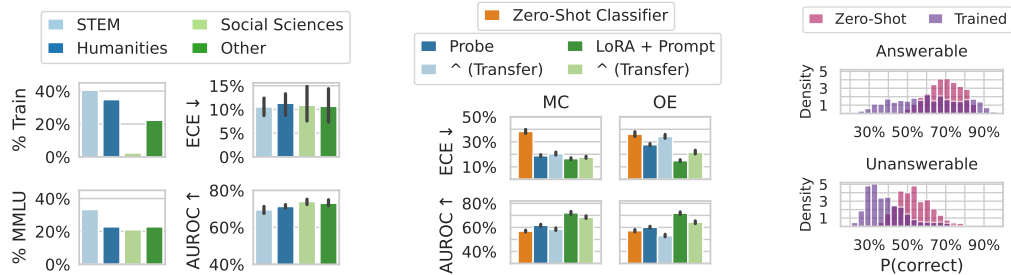
Figure 4: (**Left**) We compare the composition of the fine-tuning dataset with MMLU. Notably, although the training dataset contains close to zero examples from social sciences, uncertainty estimates from the model perform similarly across categories. (**Center**) Testing the generalization of supervised methods by taking models trained on one setting (MCQA or OE) and evaluating them on the other setting. The MCQA or OE labels denote the evaluation setting, with the method labels indicate whether the model was trained on the same or different setting. Fine-tuning through the model's features (`LoRA + Prompt`) performs almost as well in transfer as on in-distribution data. `Zero-Shot Classifier` involves no supervised learning except a temperature-scale step and is a useful reference point. Error bars show two standard deviations over six fine-tuned models. (**Right**) Fine-tuning leads to lower confidence on unanswerable questions, taken from the SelfAware dataset [61]. Assigning low confidence to unanswerable questions allows the model to opt out of responding.

languages. I have a spreadsheet of how often Alice makes mistakes in each language. Now, when I hear Alice attempting to converse in language A, I can guess how likely she is to err by recognizing the language from its sound and consulting the spreadsheet. I can do this without understanding the language at all. Alternatively, I can learn each language, which would be more complex but would strengthen my predictions.

To disentangle these two possibilities in our setting, we perform an additional experiment, in which we replace the language model's answers in the fine-tuning dataset with incorrect answer options. If a language model is simply learning patterns in the errors present in the training data, then we would expect this ablation to perform on par with the original method because it suffices to learn patterns in the content of the question and answer without needing the true causal relationship between question, answer, and correctness label. The results are shown in Figure 5 (left). We see the model trained on incorrect answers performs surprisingly well, on par with a `Probe` model, but significantly worse than a model trained on the original sampled answers. Correlating question content with error rates while moderately successful cannot be a full description of the `LoRA + Prompt` estimates.

Self-knowledge. Lastly, we examine whether a language model can be used to model not just its own uncertainties but the uncertainties of other models. Several prior works argue that models identify correct questions by way of internal representations of truth, which might be unique to a model evaluating its own generations [4, 9]. In Figure 5 (right), we show that, by contrast, Mistral 7B actual has better AUROC values when applied to LLaMA-2 7B than LLaMA-2 7B applied to itself. In Figure 5 (left), we show that sBERT [44] and OpenAI sentence embeddings are competitive with `Probe` on both LLaMA-2 7B and Mistral. Together, these results suggest that LLM uncertainties are likely not model-specific. The practical upside of this insight is that one strong base model can be used to estimate the uncertainties of many other models, even closed-source models behind APIs, when a small labeled dataset is available or can be generated.

> Learned uncertainty estimates generalize to new formatting, subject matter, and even the generations of other models. This generalization appears to stem not simply from judging a question's difficulty based on its subject matter (a short-cut) but also learning the correspondence between questions and correct answers.

8
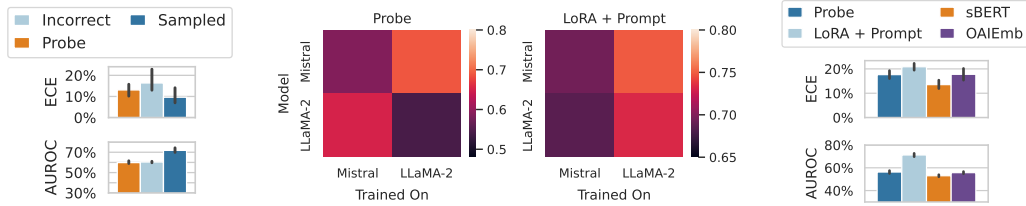
Figure 5: (**Left**) We ablate the correspondence between questions and answers by training `LoRA + Prompt` on a dataset with correctness labels from the model's generations but with the actual generations swapped with incorrect answers. In this case, the only relationships that can be extracted by the model are between the correctness labels and the questions. The model trained on incorrect answers generalizes surprisingly well but is much worse than a model trained on the original answers. Error bars show two standard deviations over three instruction-tuned models. (**Center**) We test how well models can learn to predict the correctness of a different model (in terms of AUROC), and we find that mistral models are often better at estimating the correctness of LLaMA models than LLaMA can on their own generations. (**Right**) We show that generic sentence embeddings can also perform on par with frozen language model representations (MMLU-OE), but training through a model is much better. `sBERT` and `OAIEmb` refer to training a classifier on top of sBERT [44] or OpenAI sentence embeddings. Error bars show two standard deviations over tasks in MMLU.

7 Does Calibrated Confidence Improve Collaboration with AI Assistants?

One key motivation for estimating LLM uncertainty is to signal the model's reliability during collaborative decision making. To examine how our uncertainty estimates can be used in this capacity, we perform a preliminary user study (with $N = 181$ participants) in which participants complete a multiple choice exam in collaboration with an LLM (Mistral 7B Instruct). For each question, the participant is provided both the LLM's prediction and an uncertainty estimate, which can be from a calibrated method or an uncalibrated method. We hope to show that users are more likely to adopt calibrated uncertainty scores as part of their decision process. A more detailed description of the setup of our study is available in Appendix G.

People are sensitive to informed confidence scores.   Figure 6 shows density plots of the model's reported confidence and whether the user chose to agree with the model's prediction. We find that participants are sensitive to the confidence scores and tend to use scores when deciding to agree or disagree with the model's prediction if the uncertainties are reliable. On the other hand, participants generally do not modulate their decision to rely on the output of a random confidence baseline (Figure 6(c)), in which the display uncertainty estimate is generated uniformly at random. We see the strongest discrepancy in reliance choices when `LoRA + Probe` confidence scores are presented, highlighting that calibrated confidence does influence user behavior.

We include additional details and results in Appendix G. We find that confidence scores have the biggest effect on improving the lowest performing users, rather than on average accuracy. However, this is a preliminary result in the nascent field of studying LLM uncertainties in practical collaborative decision making with users. We are only still scratching the surface of this question. For more fine-grained conclusions, a study should be devoted to this subject. We outline several limitations and future directions in Appendix G.

> Users are sensitive to confidence scores and use their relative magnitude to modulate their decision to use an LLM. Lower performing users are most improved by access to confidence scores. However, future work is needed to disentangle the effects of calibration from how humans choose to leverage uncertainties.

8 Discussion

There is much disagreement about the role of calibrated uncertainty in large language models, how it can best be achieved, and promise of black-box methods. We hope to have shed light on these

9

(a) Zero-Shot Prompt	(b) LoRA + Prompt	(c) Random (Control)
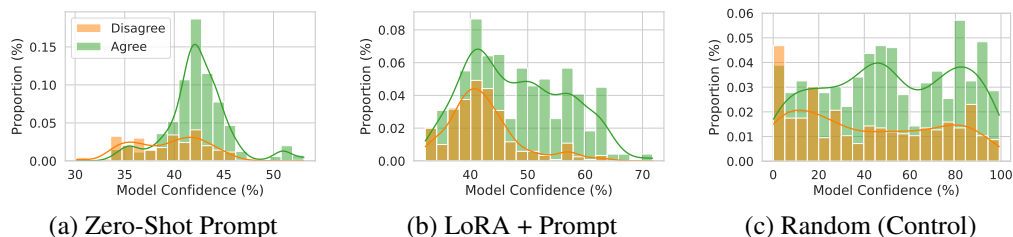
Figure 6: We compare the distribution of LLM confidence (for Mistral 7B Instruct) on its answers, and whether the users ($N = 20$ per variant) agree with the answer generated by the model or not. **(a)** For the zero-shot prompt, we find that the model provides little signal since most mass is similarly clustered. However, **(b)** improving the calibration of the model reveals an increased reliance on the LLM for more confident answers, and decreased reliance for less confident answers. Evidently, the users are sensitive to calibrated confidence scores. **(c)** For reference, we verify that uniformly confidence scores do not provide meaningful signal, rendering users unable to modulate their decision to rely on the LLM. All variants are compared at approximately the same average participant accuracy.

questions throughout this paper. In contrast to prior results, we find that out-of-the-box uncertainties from LLMs are unreliable for open-ended generation and introduce a suite of fine-tuning procedures that produce calibrated uncertainties with practical generalization properties. In the process, we discovered that fine-tuning is surprisingly sample efficient and does not seem to rely on representations of correctness specific to a model evaluating its own generations, allowing estimators to be applied from one model to another. Moreover, we found it is possible, at least in the cases we considered, for calibrated uncertainties to be robust to distribution shifts.

There are many exciting questions for future work. Currently fine-tuning relies on two separate models for question answering and uncertainty estimation. Ideally, we want a single model that can generate questions and uncertainty without switching between model weights. We anticipate that an uncertainty-aware pre-training or alignment phase might become essential but implementing such a procedure while maintaining base language modeling abilities will introduce a challenging online learning problem where the correctness labels evolve during training.

Beyond improving the safety and usefulness of language models, high quality uncertainties can also be used in active learning procedures, e.g. for sample-efficient fine-tuning [39], where data points are selected based on the predicted utility and the model's uncertainty, in order to balance the explore-exploit trade-off. Uncertainty estimates can also be used to improve factuality of language models by increasing the likelihood of generations that the model is confident about (judges likely to be correct), for example by using an alignment procedure (e.g. RLHF, DPO) with a reward function that encourages confident generations [50].

We also showed how uncertainty information could be used to influence human decision making. In the end, LLMs will impact society through decision making, and to make reasonable decisions we need uncertainty information — particularly to protect against rare but costly mistakes.

Acknowledgements

This work is supported by NSF CAREER IIS-2145492, NSF CDS&E-MSS 2134216, NSF HDR-2118310, BigHat Biosciences, Capital One, and an Amazon Research Award.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the*

10

Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2357–2367. Association for Computational Linguistics, jun 2019. doi: 10.18653/v1/N19-1245. 19

[3] Lora Aroyo and Chris Welty. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24, 2015. 23

[4] Amos Azaria and Tom M. Mitchell. The internal state of an llm knows when its lying. *ArXiv*, abs/2304.13734, 2023. 3, 5, 8

[5] Umang Bhatt, Valerie Chen, Katherine M Collins, Parameswaran Kamalaruban, Emma Kallina, Adrian Weller, and Ameet Talwalkar. Learning personalized decision support policies. *arXiv preprint arXiv:2304.06701*, 2023. 21, 22, 23

[6] Christopher M Bishop. Pattern recognition and machine learning. *Springer google schola*, 2: 1122–1128, 2006. 19

[7] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *ArXiv*, abs/1911.11641, 2019. 19

[8] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Conference on Empirical Methods in Natural Language Processing*, 2015. 19

[9] Collin Burns, Hao-Tong Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *ArXiv*, abs/2212.03827, 2022. 3, 8

[10] Cheng-Han Chiang and Hung yi Lee. Can large language models be an alternative to human evaluations? In *Annual Meeting of the Association for Computational Linguistics*, 2023. 3

[11] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *ArXiv*, abs/1905.10044, 2019. 19

[12] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018. 19

[13] Katherine Maeve Collins, Matthew Barker, Mateo Espinosa Zarlenga, Naveen Raman, Umang Bhatt, Mateja Jamnik, Ilia Sucholutsky, Adrian Weller, and Krishnamurthy Dvijotham. Human uncertainty in concept-based ai systems. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 869–889, 2023. 22

[14] Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124, 2019. 19

[15] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007. 4

[16] Andrew S. Gordon, Zornitsa Kozareva, and Melissa Roemmele. Semeval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *International Workshop on Semantic Evaluation*, 2011. 19

[17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017. 4, 6

[18] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300, 2020. 2, 3, 4, 17

[19] James Hills and Shyamal Anadkat. Using logprobs, Dec 2023. URL https://cookbook.openai.com/examples/using_logprobs. 4

11

[20] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. 6

[21] Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. In *Conference on Empirical Methods in Natural Language Processing*, 2019. 19

[22] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024. 21

[23] KJM Janssen, KGM Moons, CJ Kalkman, DE Grobbee, and Y Vergouwe. Updating methods improved the performance of a clinical prediction model in new patients. *Journal of clinical epidemiology*, 61(1):76–86, 2008. 23

[24] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. 2, 5

[25] 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. Language Models (Mostly) Know What They Know. *ArXiv*, abs/2207.05221, 2022. 1, 3, 4, 6

[26] Gideon Keren. Calibration and probability judgements: Conceptual and methodological issues. *Acta psychologica*, 77(3):217–273, 1991. 22

[27] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *North American Chapter of the Association for Computational Linguistics*, 2018. 19

[28] Justin Kruger and David Dunning. Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of personality and social psychology*, 77(6):1121, 1999. 5

[29] Justin Kruger and David Dunning. Unskilled and unaware–but why? a reply to krueger and mueller (2002). *American Psychological Association*, 2002. 5

[30] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *ArXiv*, abs/2302.09664, 2023. 2, 3, 6, 18

[31] Xin Li and Dan Roth. Learning question classifiers. In *International Conference on Computational Linguistics*, 2002. 19

[32] Sarah Lichtenstein, Baruch Fischhoff, and Lawrence D Phillips. Calibration of probabilities: The state of the art. In *Decision Making and Change in Human Affairs: Proceedings of the Fifth Research Conference on Subjective Probability, Utility, and Decision Making, Darmstadt, 1–4 September, 1975*, pages 275–324. Springer, 1977. 5, 22

[33] Stephanie C. Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Trans. Mach. Learn. Res.*, 2022, 2022. 3, 4

[34] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101, 2017. 17, 19

12

[35] David John Cameron MacKay. Information theory, inference, and learning algorithms. *IEEE Transactions on Information Theory*, 50:2544–2545, 2004. 19

[36] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018. 19

[37] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:2901–2907, 2015. 3

[38] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *ArXiv*, abs/1910.14599, 2019. 19

[39] Ian Osband, Seyed Mohammad Asghari, Benjamin Van Roy, Nat McAleese, John Aslanides, and Geoffrey Irving. Fine-tuning language models via epistemic neural networks. *arXiv preprint arXiv:2211.01568*, 2022. 10

[40] Stefan Palan and Christian Schitter. Prolific. ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27, 2018. 21

[41] 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. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems*, 2019. 19

[42] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 4

[43] Benjamin Plaut, Khanh Nguyen, and Tu Trinh. Softmax probabilities (mostly) predict large language model correctness on multiple-choice q&a. *arXiv preprint arXiv:2402.13213*, 2024. 2, 4

[44] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. 8, 9

[45] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *ArXiv*, abs/1907.10641, 2019. 19

[46] Stefan Schaal. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996. 5

[47] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *ArXiv*, abs/1811.00937, 2019. 19

[48] Gemini Team. Gemini: A family of highly capable multimodal models, 2024. 2

[49] Thomas C Terwilliger, Dorothee Liebschner, Tristan I Croll, Christopher J Williams, Airlie J McCoy, Billy K Poon, Pavel V Afonine, Robert D Oeffner, Jane S Richardson, Randy J Read, et al. Alphafold predictions are valuable hypotheses and accelerate but do not replace experimental structure determination. *Nature Methods*, pages 1–7, 2023. 23

[50] Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. Fine-tuning language models for factuality. *arXiv preprint arXiv:2311.08401*, 2023. 10

[51] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*, 2023. 1, 3, 4, 18

13

[52] 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. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. 5

[53] 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. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023. 2, 5

[54] Dennis Ulmer, Martin Gubri, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. Calibrating large language models using their generations only. In *Annual Meeting of the Association for Computational Linguistics*, 2024. 3

[55] Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72: 1385–1470, 2021. 23

[56] Kailas Vodrahalli, Tobias Gerstenberg, and James Y Zou. Uncalibrated models can improve human-ai collaboration. *Advances in Neural Information Processing Systems*, 35:4004–4016, 2022. 22

[57] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652, 2021. 6

[58] Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. *ArXiv*, abs/1707.06209, 2017. 19

[59] 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. 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, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6. 19

[60] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *ArXiv*, abs/2306.13063, 2023. 1, 3, 4, 18

[61] Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada, 2023. Association for Computational Linguistics. 1, 7, 8

[62] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*, 2019. 19

[63] Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*, 2023. 3, 4

14

Appendix for
Large Language Models Must Be Taught
to Know What They Don't Know

Table of Contents

A	**Evaluation Methods**	**15**
	A.1 Evaluating Correctness	15
	A.2 Grading	16
	A.3 Comparison of Grading Techniques	16
	A.4 Metrics	17
	A.5 MMLU Supercategory Classifier	17
B	**Baseline Methods**	**17**
	B.1 Sampling Methods	17
	B.2 Verbal Elicitation	18
C	**Fine-tuning Method**	**19**
	C.1 Regularization Term	19
	C.2 Training Data	19
	C.3 Training Hyperparameters	19
D	**Extended MMLU Results**	**19**
E	**Confidence as a Function of Target Length**	**20**
F	**Generalization to Coding Tasks**	**21**
G	**User Studies**	**21**
	G.1 Additional Details on Setup	21
	G.2 Important considerations	22
	G.3 Extended Results	22
	G.4 Interface and Instructions	23
H	**Broader Impact and Implications**	**23**
I	**NeurIPS Paper Checklist**	**35**

A   Evaluation Methods

A.1   Evaluating Correctness

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

15

City," or "Daoism" and "Taoism." Conversely, it also has the potential to be over-generous 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 7). 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 A.3. We find that humans and GPT 3.5 Turbo have much greater agreement than humans and the substring method.

A.2 Grading

Dataset Construction.   To perform calibration-tuning (CT), we need tuples $(Q, A, \hat{A}, C)$, answers from a language model that have been graded for correctness. When calibration-tuning on multiple choice questions, we can use an exact string match to generate C. To grade open-ended answers, we use a strong language model and *grading prompt G* instead (Figure 7):

- G: a prompt used for grading answers $\hat{A}$ with A.

Compared to alternatives like exact match, language model grading is insensitive to re-phrasings that are equivalent in meaning - such as "NYC" and "New York City," or "Daoism" and "Taoism". LLM grading can also penalize answers that are overly verbose or use a different meaning of the same word, potentially containing incorrect answers along with the correct string. For example, if the question is "What's it called when you move quickly by foot and both feet aren't always touching the ground?" and the LLM response is "A bank run", the grader should be able to distinguish that this is semantically dissimilar to the true answer "run".

In this paper, we utilize GPT 3.5 Turbo as the auxiliary grading model. When comparing many possible grading methods on subsets of MMLU, we find that GPT 3.5 Turbo has high agreement with humans while being cost efficient (Appendix A.3).

Grading prompt (G)

The problem is: Q
The correct answer is: A
A student submitted: $\hat{A}$

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: C </s>

Figure 7: For open-ended generation, we calculate the ground-truth correctness C using a LLM and a grading prompt (G). The token </s> is an end-of-sentence token. **Blue text** is included in the loss function when calibration-tuning.

A.3 Comparison of Grading Techniques

We conducted an analysis of the methods outlined in Appendix A.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.

16

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.

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 7.

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 2: 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 2. 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.

A.4 Metrics

ECE Given N samples and B equally-spaced bins b_j, examples are assigned to bins based on the confidence of the model, and ECE is estimated as $\widehat{\text{ECE}} = \sum_{j=1}^{B} \frac{|b_j|}{N} |\text{conf}(b_j) - \text{acc}(b_j)|$ where $\text{conf}(b_j)$ is the average confidence of samples in bin b_j, $\text{acc}(b_j)$ is the accuracy within the bin, and $|b_j|$ is the number of samples assigned to bin j. In our experiments conf is equivalent to $P(\text{correct})$.

A.5 MMLU Supercategory Classifier

To understand the impact of the subject matter of the training data on generalization, we follow the prescription of Hendrycks et al. [18] and categorize each of the 57 tasks into one of four supercategories - Humanities, STEM, Social Sciences, and Other. Since we do not have such a categorization for the training set, we must estimate their proportions.

First, we use the OpenAI embeddings (dimension 1536) of the MMLU samples with their ground truth supercategories to train a linear 4-way classifier with 10 samples from each of the 57 tasks. We use AdamW [34] with learning rate 1e-3 and weight decay 1e-2. This classifier is then used to estimate the categories of each sample in the training set used for fine-tuning. Subsequently, the breakdown of results in Figure 4 (Left) follows.

B   Baseline Methods

B.1   Sampling Methods

We use two baselines which obtain an estimate of certainty by sampling the same answers $n = 10$ times and then estimating the proportion of sampled answers that agree with the greedily decoded "main" answer. There are several critical downsides to these approaches: (i) the uncertainty here depends on the sampling parameters—for example, in the limit where the sampling converges to mere greedy decoding, the LLM will produce n identical samples, and therefore the certainty will always be 1—(ii) these approaches require $O(n)$ answer generations to provide a certainty estimate for a

17

single generation. The intense computational restriction prevents us from easily searching the space of sampling parameters for the optimal set, so we choose parameters arbitrarily; here we sample with $top\_p = 0.95$.

Counting   In this baseline, each sampled answer is compared to the greedy answer by prompting an expert LLM with both answers and asking it to judge their equivalence. The proportion of samples that are equivalent to the greedy answer is the certainty estimate. This baseline is similar to *Label prob* [51]; our method differs by not choosing the argmax semantic group as the final prediction, but instead using the greedy decode for the final prediction, so as to maintain the same accuracy performance as our uncertainty query method. This met

Likelihood accumulation   In this baseline, we add up likelihoods of sampled answers to estimate the mass associated with the predicted answer. We begin by prompting an expert LLM in order to find which sampled answers are equivalent to the greedy answer—like in the counting baseline. In this method, the certainty estimate is produced by adding the length-normalized likelihoods of those sampled answers equivalent to the greedy answer, and dividing this quantity by the sum of all sampled answers' length-normalized likelihoods. This procedure of adding likelihoods of samples in order to estimate the likelihood of an equivalence class is similar to that used by [30], although they do not use it for certainty estimates but instead to produce entropy scores. In practice, the scores produced by these two methods are actually very similar—so we report only likelihood accumulation numbers in the main text.

B.2   Verbal Elicitation

Although Tian et al. [51] introduce several strategies for prompting, involving multiple guesses or multiple stages of interleaving prompting and generation, we did not find that any strategy consistently outperformed any other. This finding was consistent with the results of Xiong et al. [60]. Ultimately, for convenience, we adopted a two stage strategy with a single guess because it can be used in tandem with logged datasets of generated answers per model.

The exact prompt we used is essentially the same at in [51], but with small modifications that improved the rate of correctly formatted responses:

> "Provide the probability that your answer is correct. Give ONLY the probability, no other words or explanation.
>
> For example:
>
> Probability: <the probability between 0.0 and 1.0 that your guess is correct, without any extra commentary whatsoever; just the probability!>
>
> Include probability for the answer below: Probability:"

Verbal elicitation methods typically output complex strings containing both answers and associated probabilities. This means that if any element of parsing fails, it can be challenging to construct partial results. This effect tends to diminish when using large models, which are more responsive to zero-shot prompting.

Parsing Details   The original verbal elicitation prompts are given in the appendix of [51]. However, it is not clear how the original authors decide to parse answers from the generations and how failure to parse is handled. When we fail to parse the guess from the generation we return an empty string and associated probability 0.5. When we fail to parse a probability, we also return probability 0.5. For versions with multiple guesses, if any part of the parsing processes fails in an ambiguous way, we default back to an empty string for the answer and 0.5 for the probability. The only unambiguous cases are those which explicit succeed in the generating a valid guess and probability in the first case but not subsequent cases. In this scenario, we default to using the successfully parse first guess and associated probability.

18

C Fine-tuning Method

C.1 Regularization Term

To keep the calibration-tuned parameters θ within the neighborhood of the initial parameters, θ_0, we use a regularization term that penalizes the divergence between the original sampling distribution and the calibration-tuned model on the target sequence A, yielding regularization $\mathcal{R}(\theta; \theta_0)$, which we use with weighting parameter κ.

Specifically, let p_{θ_0} be the language modeling distribution 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 Jensen-Shannon Divergence $\text{JSD}(p_{\theta_0} \parallel q_\theta)$ [35] between the two language modeling distributions as the regularizer, where $\text{JSD}(p \parallel q) \triangleq \frac{1}{2}(\text{KL}(p \parallel m) + \text{KL}(q \parallel m))$, where $m \triangleq \frac{1}{2}(p + q)$ is the mixture distribution. JSD regularization is applied only to the logits corresponding to the target sequence A.

We note that using either direction of KL-divergence, i.e. the forward-KL $\text{KL}(p_{\theta_0} \parallel q_\theta)$ or reverse-KL $\text{KL}(q_\theta \parallel p_{\theta_0})$ was insufficient for optimal performance with calibration tuning. The forward KL-divergence encourages a zero-avoiding behavior such that the mass of q_θ is spread across multiple modes of p_{θ_0} to minimize the KL-divergence to avoid assigning no mass to regions of the probability space. To the contrary, the reverse KL-divergence encourages a zero-forcing behavior such the q_θ only needs to cover any one mode of p_{θ_0} [6]. It is not necessarily obvious which one of these behaviors one should prefer for the specific case of large language models. Therefore, as a practical choice, we pick the one that provides us the most performant calibration-tuned model.

C.2 Training Data

We reserve the following datasets for training.

- AI2 Reasoning Challenge (ARC) [12],
- Boolean Questions (BoolQ) [11],
- CommonsenseQA [47],
- CosmosQA [21],
- HellaSwag [62],
- MathQA [2],
- Recognizing Textual Entailment (RTE/SNLI) [8],
- Adversarial NLI [38],
- OpenBookQA [36],
- PIQA [7],
- SciQ [58],
- The CommitmentBank (CB) [14],
- Multi-Sentence Reading Comprehension (MultiRC) [27],
- Choice of Plausible Alternatives (CoPA) [16],
- TREC [31],
- Adversarial Winograd (Winogrande) [45].

C.3 Training Hyperparameters

We use HuggingFace Transformers [59] and PyTorch [41] for the implementation of these models. For all our experiments, we use the AdamW optimizer [34] 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.

D Extended MMLU Results

We report the breakdown of uncertainty query accuracy and ECE on all MMLU tasks in Figures 8 to 11.

19

Figure 8: (Part 1) ECE and AUROC values for `Query`, `CT-Probe`, `CT-LoRA`, and `CT-Query` for each subset of MMLU in multiple-choice question answering (MCQA) setting.

E   Confidence as a Function of Target Length

As we noted when motivating calibration tuning, one limitation of sequence-level probabilities is their intrinsic connection to sequence length. The probability of a sequence decreases with increasing length, regardless of the correctness of the response. By contrast, we wouldn't expect concept-level probabilities to have any discernible relationship with sequence length. In Appendix E, we show

20

Model	Method	Acc	ECE	AUROC
LLaMa-2-7B	Zero-Shot Classifier	3.2%	41.0%	56.9%
	Lora + Prompt	3.2%	46.4%	80.0%
Mistral-7B	Zero-Shot Classifier	27.4%	70.2%	66.2%
	Lora + Prompt	27.4%	21.4%	85.1%
Mistral-7B-Instruct	Zero-Shot Classifier	21.0%	52.7%	47.1%
	Lora + Prompt	21.0%	56.1%	70.2%

Table 3: ECE and AUROC on livecodebench_generation_lite (LeetCode easy subset). ECE is shown after temperature scaling on a small hold-out set of the original dataset mixture Appendix C.2. Acc is task accuracy (proportion of coding solutions that are correct). Supervised training (LoRA + Prompt) seems to always improve selective prediction, although supervised training only heavily improves calibration for Mistral-7B and in fact slightly degrades calibration for the two other models.

there is no consistent relationship between the confidence estimated by the calibration-tuned model and target sequence length on MMLU tasks.

A key limitation of using token likelihoods is that they necessarily decay with the length of the generation. In Figures 12 to 14, we confirm over all subsets of MMLU that the length of the target does not strongly correlate with the confidence associated with the targets. This behavior is an essential ingredient towards an effective confidence estimation in practice, such that longer sequences are not penalized in confidence despite being correct.

F Generalization to Coding Tasks

Because there are no coding tasks in our training dataset, we can use a coding competition task introduced in LiveCodeBench [22] to assess how well finetuned uncertainty estimation methods perform on completely out of distribution tasks.

To conduct the analysis in Table 3, we evaluate several base models on the 62 LeetCode easy questions from the livecodebench_generation_lite task. We asking for the model to write a Python solution and grade the solution using test cases (marking it as correct iff it passes all test cases). We then apply `Lora + Prompt` and `Zero-Shot Classifier` uncertainty estimation methods—with these methods *only* using training/temperature scaling data from our main dataset mixture which notably does not include any coding tasks Appendix C.2. Accuracy is shown to contextualize the model's overall level of performance on the task. On Mistral-7B, the best performing model on the coding task, the supervised `Lora + Prompt` approach dramatically improves calibration and selective prediction as compared to `Zero-Shot Classifier`; on the worse-performing Mistral-7B-Instruct and LLaMa-2-7B, selective prediction improves but calibration slightly degrades.

G User Studies

G.1 Additional Details on Setup

Stimuli and Participant Selection   We closely followed the setup of [5]. We used the same 180 MMLU questions from which were pre-batched into three sets of 60 MMLU questions. Within each variant, we randomly assigned participants to one of the three batches. In total, we recruit 181 participants (20 per variant[4]). All participants were recruited through the crowdsourcing platform Prolific [40]; we restrict our participant pool to those based in the United States who speak English as a first language.

Compensation   Participants were told that the study would take approximately 30 minutes and were paid at a base rate of $9/hr and informed that they would receive an optional bonus up to $10 for answering questions correctly. We applied the bonus to all participants.

---

[4]With the exception of one extra participant due to random batching allocation effects.

21

LLM Answers and Uncertainty Elicitation   Bhatt et al. originally used GPT-3.5 as their LLM. While at first, we explored user performance when provided with confidence scores modulated over the original GPT-3.5 responses that the authors had collected, the authors had filtered LLM performance to ensure the LLM achieved high performance on biology, computer science, and foreign policy and poor performance on mathematics. As such, we noticed that participants overwhelmingly uptook the LLM's answer (which was rational behaviour, given the model's high performance). To explore a more nuanced performance profile, we regenerated LLM answers using Mistral 7B Instruct via greedy decoding. We then generated confidence scores on top of the LLM responses. For our random baseline, we sample a confidence score uniformly between 0 and 100% for each question.

G.2   Important considerations

There are many reasons to heed caution in interpreting our results as definitive indications of the utility of displaying confidence to users in LLM assistive settings. In particular: (i) users are presented with feedback after each trial as in [5] – as such, they can determine (potentially rapidly) whether or not a model is reliable, even without confidence scores. However, in practical settings users may not know whether or not the model was truly correct and therefore confidence scores could have an even larger impact; (ii) MMLU questions can be challenging for non-experts – we see the biggest differences in performance for the no-LLM vs. any-LLM-assistance condition. We may see a wider range of reliance behaviors in settings wherein people have more confidence in their own abilities; (iii) we present users with numeric confidence; however, humans are not always able to reliably process confidence estimates nor appropriately calibrate uncertainty estimates themselves [26, 56, 13, 32]. It may be that alternate modes of communicating confidence improve users' ability to appropriately leverage the confidence scores in their decision making process. We see targeted exploration of each component through interdisciplinary collaboration across AI, behavioral science, and human-computer interaction as ripe for future work.

G.3   Extended Results

Task Accuracy and Reliance Sensibility   We depict average user task accuracy and reliance sensibility across variants in Figure 15. We follow Bhatt et al. in computing reliance sensibility as the proportion of times the user appropriately sided with the model prediction when the model was correct and did not respond with the model's prediction when the model was incorrect.

We depict per-topic accuracy, with the LLM's average performance in Figure 16.

GPT-3.5 Confidence Generalization   As noted, we ran variants using the same GPT-3.5 generations as [5]. We show aggregate and per-topic accuracy in Figure 17, as well as reliance sensibility in Figure 18.

Freeform User Responses   We permitted users to provide freeform responses at the end of the study. Some users were sensitive to confidence scores being reported and came up with their own heuristics for whether to rely on the model's output. We include a sampling of comments across confidence variants:

- "if it had a confidence of less than 50% it made me very skeptical."
- "The models̉ confidence indeed helped me choose and select my answer as I trusted in them most of the time."
- "I didn̉t really rely on the confidence level. If I had 0 confidence in the answer myself I relied on the AI regardless."
- "if the models confidence fell below 45 I decided to investigate it myself by remembering pieces of information. and also reasoning the question. If it was above 45 I would automatically agree to its prediction but there were some few cases I challenged it even though it was above 45"
- "At first I was hesitant to trust the model much because of the lower confidence levels but I still trusted it enough on topics I struggled with. As it went on, I was comfortable with confidence levels above 40."

22

- "If the modelś confidence was low and I thought I knew the answer (and it was different) I chose my answer"

G.4   Interface and Instructions

We show a sample interface of our extension of `Modiste` with user confidence in Figure 19, and present the the full set of instructions provided to users in Figures 20 and 21. Note, for the LLM-only and no-LLM conditions, we followed the instruction text from [5] directly, i.e., participants who saw only the LLM did not see the instruction page about model confidence, and participants in the "No-LLM" variant were not instructed about any model variant and were just instructed to answer the questions as best as they could by themselves. Participants also responded to a post survey questionarre after completing the user study, which we depict in Figure 22.

H   Broader Impact and Implications

The goal of this work is to make LLM outputs have better confidence values associated with them. With successful, calibrated confidence values, the machine systems ultimately become more interpretable and trustworthy by a user [23]. When applied correctly, our advancements will help users be able to make decisions based off of LLM outputs in a more informed way. Similar examples in other domains, like AlphaFold [49], have shown how well-calibrated confidence scores can be useful in complex decision-making domains. Our hope is to replicate those broad findings in LLMs.

We acknowledge the ongoing debate over the appropriateness, limitations, and harms of LLMs. We do highlight that the development of more confident, interpretable, and trustworthy LLMs can lead to continued techno-solutionism in unintended applications. Specifically, we highlight that our work is limited to use-cases with fact-based questions. Many applications of text-based LLMs are generative, meaning that there is no way for our paradigm to be applied appropriately, and the use of a confidences from calibration-tuned models could be misleading or damaging without checks and guardrails. Additionally, even within the fact-based paradigm, what is true can be subjective, with ground truth in machine learning being a contested topic [3, 55].

The philosophical debate on these topics is beyond the expertise of the authors; nonetheless, we believe that the ongoing debate over the appropriateness of LLMs should be considered in context with the benefits of our approach in making LLMs more interpretable and useful.

23

Figure 9: (Part 2) ECE and AUROC values for `Query`, `CT-Probe`, `CT-LoRA`, and `CT-Query` for each subset of MMLU in multiple-choice question answering (MCQA) setting.

24

Figure 10: (Part 1) ECE and AUROC values for `Query`, `CT-Probe`, `CT-LoRA`, and `CT-Query` for each subset of MMLU in open-ended (OE) setting.

25

Figure 11: (Part 2) ECE and AUROC values for `Query`, `CT-Probe`, `CT-LoRA`, and `CT-Query` for each subset of MMLU in open-ended (OE) setting.

26
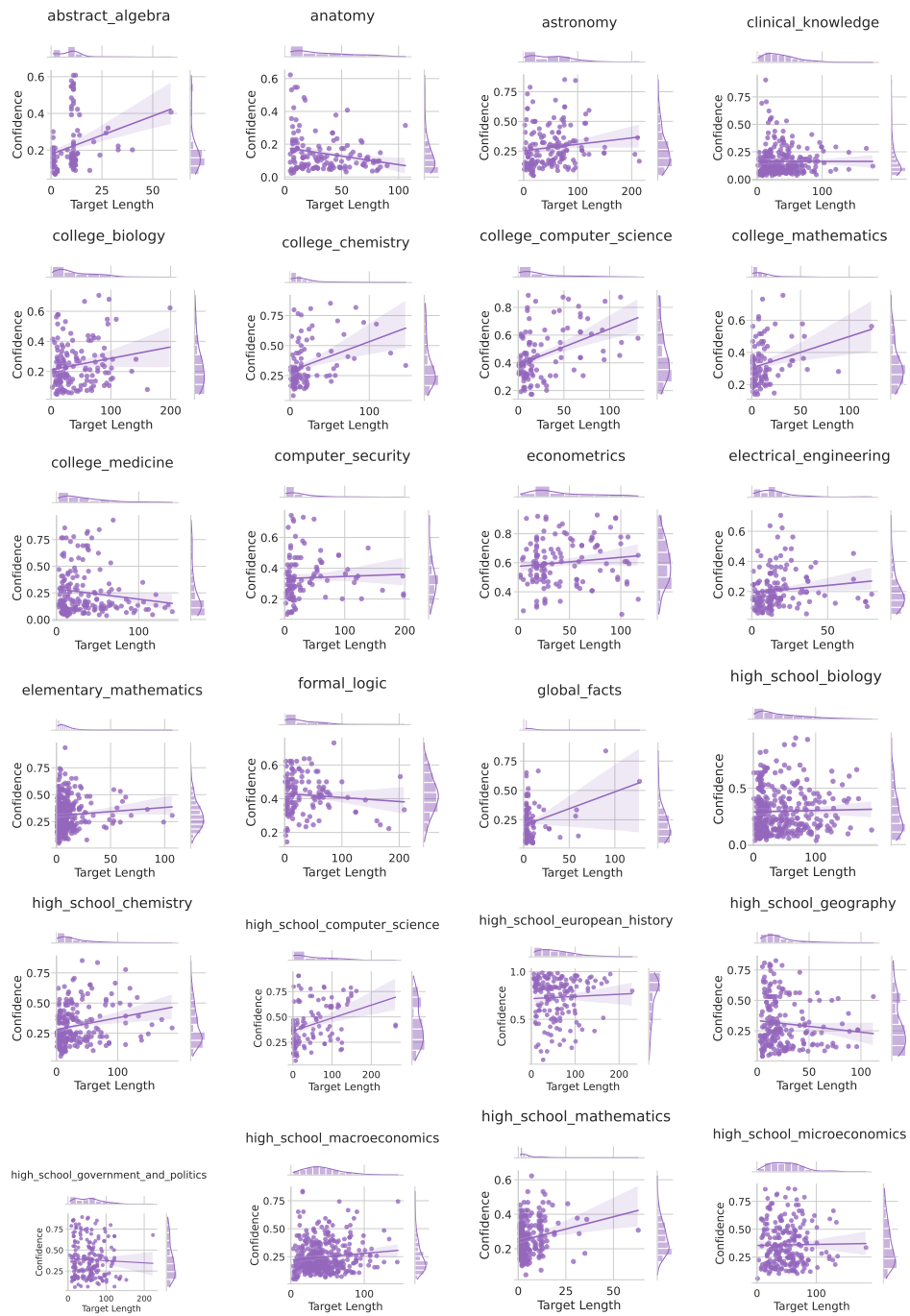
Figure 12: Confidence versus Target Length for various MMLU subsets. A horizontal regression line indicates weak correlation of confidence with the target length. See Figures 13 and 14 for other subsets.

27
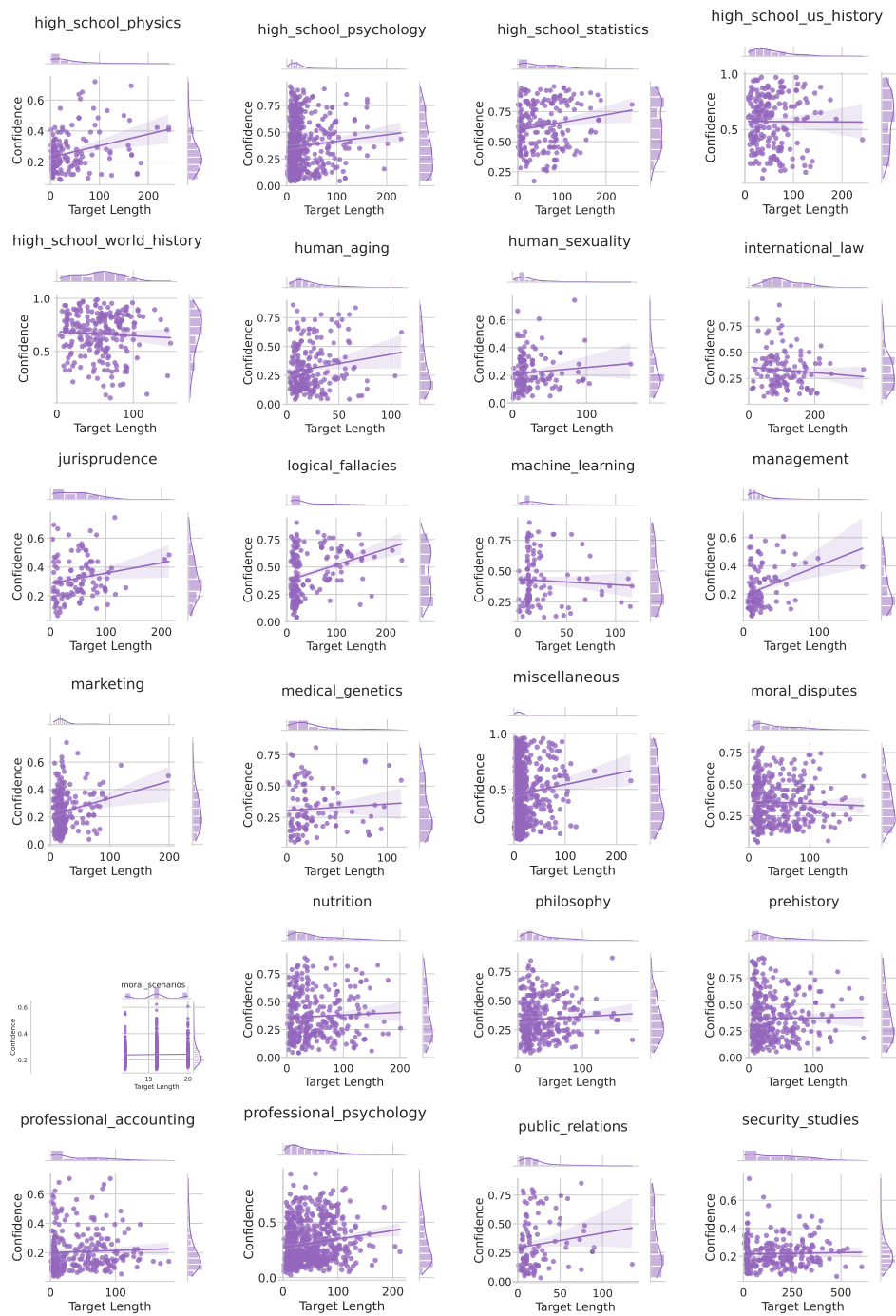
Figure 13: Continuing from Figure 12. See also Figure 14.

28

Figure 14: Continuing from Figures 12 and 13.
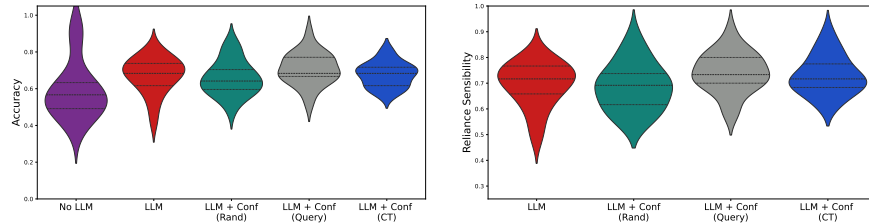
Figure 15: (**Left**) User accuracy on 60 MMLU questions per variant ($N = 20$ users per variant); violin plots show quartiles as dashed lines (**Right**) Average reliance sensibility (proportion of instances where the user sided with the model when the model was correct, and overrode the model's prediction when the model was incorrect); higher indicates better reliance calibration.
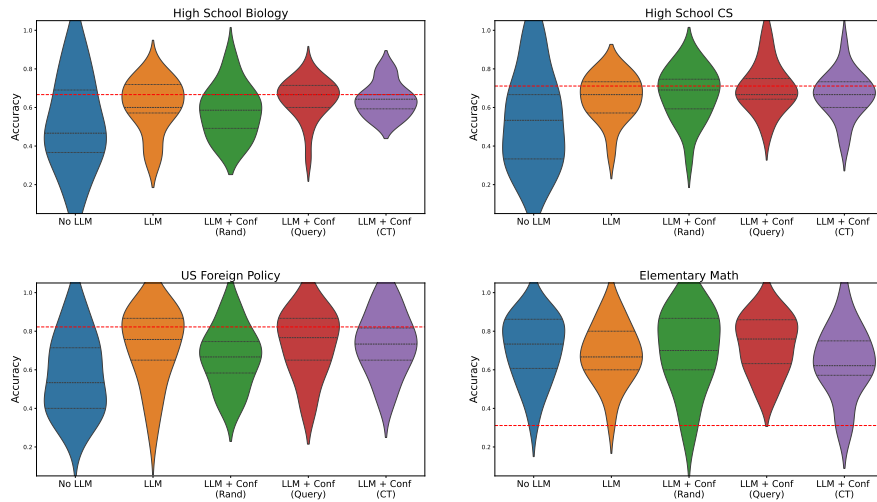
Figure 16: User accuracies per topic for the Mistral variants. Red line indicates the model's average accuracy.

29
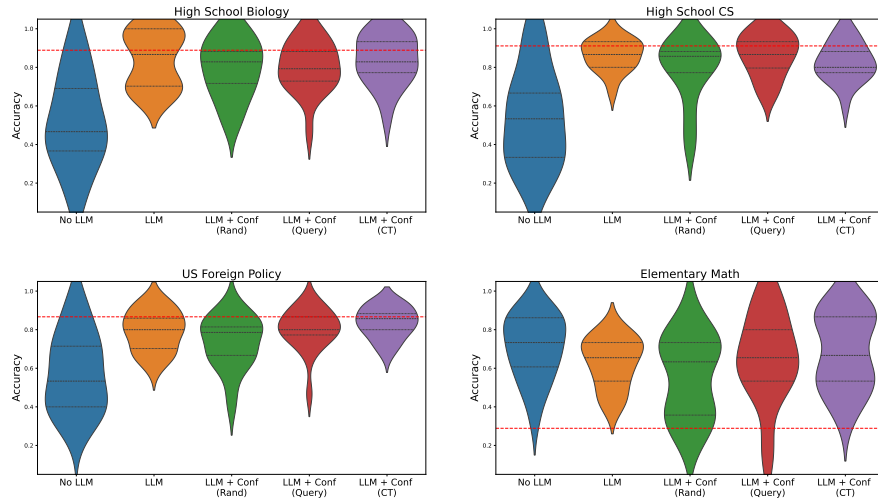
Figure 17: User accuracies per topic for the GPT-3.5 variants (with generalization confidence computed for the CT and Query cases). Red line indicates the model's average accuracy.
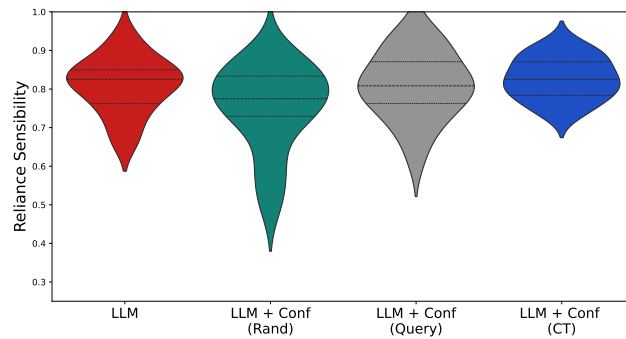
Figure 18: Reliance sensibility for the variants based on GPT-3.5

30

Figure 19: Example interface from `Modiste`. Participants are informed of the question (and topic), as well as the LLM prediction and confidence. Participants are informed of their running score throughout the experiment.

31

Welcome!

We are conducting an experiment to understand how people make decisions with and without AI support. Your answers will be used to inform machine learning, cognitive science, and human–computer interaction research.

This experiment should take at most **30 minutes**.

You will be compensated at a base rate of $9/hour for a total of **$4.50**, which you will receive as long as you complete the study.

< Previous   Next >

In this experiment, you will be seeing *multiple choice questions*, from various topics, such as those that you may find in school (e.g., biology, mathematics, foreign policy, computer science).

Your task is to determine the **most likely answer** for each question. You can select this category by clicking on the radio button associated with your answer.

< Previous   Next >

During the tasks, you will also see the **prediction of an AI-based model**.

The model's prediction will show up as yellow highlighting over that answer choice. If shown, you are free to use or ignore the information when selecting your answer however you wish.

< Previous   Next >

You will also see the model's *confidence* in its prediction (which will be shown in blue) for each question.

< Previous   Next >

We encourage you to try to work through each problem. You will not be able to continue to the next question until at least **10 seconds** have passed. The SUBMIT button will change from grey to blue when you are able to click to move to the next page whenever you are ready to answer.

Of course you can take longer than 10 seconds on any question if needed! It may be very challenging to determine the answer for some questions. Others may be easy. **Please try your best** regardless.

< Previous   Next >

You will receive a **bonus** of up to a rate of $10/hour (+$0.50) based on how many questions you correctly answer.

You will be informed whether or not you are correct after each trial.

< Previous   Next >

Figure 20: Experiment instructions for the confidence variants.

32

You will see a total of **60 questions**.

< Previous   Next >

When you are ready, please click **"Next"** to complete a quick comprehension check, before moving on to the experiment.

Please make sure to window size is in full screen, or substantially large enough, to properly view the questions.

< Previous   Next >

Check your knowledge before you begin. If you don't know the answers, don't worry; we will show you the instructions again.

What will you be asked to determine in this task?*

○ The answer to a mutliple choice question.
○ The least likely answer to a multiple choice question.
○ The most likely categories of an image.

How will you select your answer?*

○ Typing in a text box.
○ Clicking on a radio button.
○ Selecting from a dropdown menu.

Continue

Figure 21: Experiment instructions for the confidence variants (continued).

33

Thank you for participating in our study!

Click **"Finish"** to complete the experiment and receive compensation. If you have any comments about the experiment, please let us know in the form below.

How challenging did you find the questions? (On a scale of 1-10, with 10 being very challenging)

Did the model's confidence impact your response? In what way if so, please be as specific as possible (1-3 sentences)

Were there any question topics you struggled with?

Were there any question topics you were always very confident in?

Do you have any additional comments to share with us?

Finish

Figure 22: Sample pot-survey questionnaire for users who were allocated to a variant wherein they saw model confidence.

34

I NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We describe and link all claims in Section 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a discussion on the limitations in Section 8.

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: [NA]

35

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the complete code, and the complete list of datasets used for all experiments in Section 5 to reproduce all our experiments with instructions. All hyperparameters are described in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

36

Answer: [Yes]

Justification: We provide the complete code, and the complete list of datasets used for all experiments in Appendix C.2 to reproduce all our experiments with instructions. All hyperparameters are described in Section 5.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the complete code, and the complete list of datasets used for all experiments in Appendix C.2 to reproduce all our experiments with instructions. All hyperparameters are described in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All figures are appropriately labeled with the error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

37

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide an estimate of the compute resources required in Section 5.

Guidelines:
- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: We have read the ethics guidelines

Guidelines:
- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We provide a broader impact statement in Appendix H

Guidelines:
- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

38

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We train on open-access models with open-source datasets. We do not change their generation behavior, and all existing safeguards (if any) remain.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We explicitly cite all models in Section 5. All datasets used are listed and cited in Appendix C.2.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

39

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release our trained models for easy use via Hugging Face.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: We provide screenshots of our instructions, as well as details of compensation in Appendix G.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: We received prior approval from our respective institutional ethics review body for our user study. All users provided consent before partaking in the study.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

40

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

41