Calibrating Large Language Models Using Their Generations Only

Dennis Ulmer[1, 2, 3] **Martin Gubri**[1] **Hwaran Lee**[4] **Sangdoo Yun**[4] **Seong Joon Oh**[1, 5, 6]
[1]Parameter Lab [2]IT University of Copenhagen [3]Pioneer Centre for Artificial Intelligence
[4]NAVER AI Lab [5]University of Tübingen [6]Tübingen AI Center
dennis.ulmer@mailbox.org

Abstract

As large language models (LLMs) are increasingly deployed in user-facing applications, building trust and maintaining safety by accurately quantifying a model's confidence in its prediction becomes even more important. However, finding effective ways to calibrate LLMs—especially when the only interface to the models is their generated text—remains a challenge. We propose APRICOT 🍑 (Auxiliary prediction of confidence targets): A method to set confidence targets and train an additional model that predicts an LLM's confidence based on its textual input and output alone. This approach has several advantages: It is conceptually simple, does not require access to the target model beyond its output, does not interfere with the language generation, and has a multitude of potential usages, for instance by verbalizing the predicted confidence or adjusting the given answer based on the confidence. We show how our approach performs competitively in terms of calibration error for white-box and black-box LLMs on closed-book question-answering to detect incorrect LLM answers.

1 Introduction

When given a case description of "*A man superglued his face to a piano and he says it's making it hard to get a full night of sleep*", a recently released medical LLM was found to list unrelated potential causes in its diagnosis, including narcolepsy, sleep apnea and others.[1] This, of course, ignores the seemingly obvious reason for the patient's complaints. While humorous, this example illustrates the pitfalls of practical LLM applications: Despite often looking convincing on the surface—especially to non-experts—model responses can be wrong or unreliable, leading to potentially harmful outcomes or a loss of trust in the system, foregoing its benefits. Indeed, consistent behavior (imagine

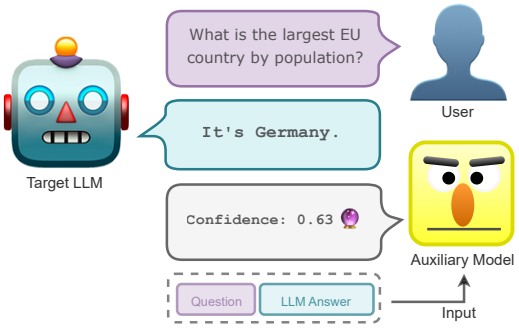


Figure 1: Illustration of APRICOT 🍑: We train an auxiliary model to predict a target LLM's confidence based on its input and the generated answer.

e.g. reliably indicating a lack of confidence for unsure responses) has been argued as one way to build trust in automated systems (Jacovi et al., 2021), while misleading predictions have been empirically shown to lead to a loss of trust that can be hard to recover from (Dhuliawala et al., 2023).

We introduce APRICOT 🍑, a method to use an auxiliary model to infer a LLM's confidence in an open-ended question-answering setting. The auxiliary model does so based on the given input to and generated output text from the LLM alone. The model is trained by using these two parts as input and predicting calibration targets. The latter are obtained without access to the LLM's sequence likelihoods or internal states by clustering input representations produced by an additional embedding model, and thus only require black-box access. This is especially relevant since an increasing number of LLM providers safeguard their model behind black-box APIs. This approach is conceptually straightforward, easy to optimize, and opens up a large number of possible applications, for instance, verbalizing uncertainty—a recently popular way to communicate uncertainty for LLMs (Lin et al., 2022; Xiong et al., 2023a)—or adjusting a model's response with linguistic markers of confidence.

Our contributions are as follows: We propose to

[1]https://x.com/spiantado/status/1620459270180569090 (last accessed Nov. 7, 2023).

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15440–15459

obtain calibration targets without requiring any additional information about LLM internals or question metadata. We show that using auxiliary models on the target LLM's input and output is sufficient to predict a useful notion of confidence. We also perform additional studies to identify which parts of the LLM's output are most useful to predict confidence. All the code is openly available.[2]

2 Related Work

Trustworthiness in ML systems. Trustworthiness has been identified as a key challenge to exploit the potential applications of automated systems (Marcus, 2020; Goldblum et al., 2023). This trait is seen as especially under risk in the presence of out-of-distribution examples or distributional shift (Snoek et al., 2019; D'Amour et al., 2022). As illustrated by the introductory example, addressing these problems is paramount in high-stakes domains such as healthcare (He et al., 2019; Ulmer et al., 2020; van der Meijden et al., 2023), analyzing asylum cases (Nalbandian, 2022), or legal deliberations (Chalkidis, 2023; Dahl et al., 2024). Jacovi et al. (2021) argue that trust in automated systems can for instance be built extrinsically, e.g. through consistent and predictable behavior. One such a way is *uncertainty quantification*, i.e. by supplying a user with scores reflecting the reliability of a model prediction (Bhatt et al., 2021; Liao and Sundar, 2022; Hasan et al., 2023). However, the lab experiments of Dhuliawala et al. (2023) also demonstrate the flip side of this: When exposing human participants to unreliable confidence estimates, they measure a decrease in trust and participant outcomes alike. Therefore, a method like ours can help to build trust in LLMs and pave the way for reaping their benefits.

Uncertainty Quantification for LLMs. While methods for predictive uncertainty quantification have already been explored in NLP for classification (Ulmer et al., 2022a; Van Landeghem et al., 2022; Vazhentsev et al., 2023), regression (Beck et al., 2016; Glushkova et al., 2021; Zerva et al., 2022a) and language generation tasks (Xiao et al., 2020; Malinin and Gales, 2021), their application to LLMs has posed novel challenges. Besides the different kinds of uncertainty due to the nature of language itself (Baan et al., 2023), LLMs are usually too large for Bayesian methods and are more expensive to be finetuned (unless one resorts to low-rank approximations; Yang et al., 2023). Furthermore, the generation process is often shielded behind black-box APIs for commercial models, only leaving access to the predicted probabilities, or—in the worst case—the generated text. Existing approaches operate on the model's confidence and improve it through (re-)calibration (Tian et al., 2023; Chen et al., 2024; Bakman et al., 2024), the ensembling of prompts (Jiang et al., 2023; Hou et al., 2023), asking the model to rate its own uncertainty (Lin et al., 2022; Chen and Mueller, 2023; Tian et al., 2023) or analyzing its use of linguistic markers (Zhou et al., 2023), computing the entropy over sets of generations with similar meaning (Kuhn et al., 2023) or comparing the textual similarity of generations for the same input (Lin et al., 2023). The most similar work to ours comes from Mielke et al. (2022), who train a calibrator on the target model's hidden states to predict whether the answer is correct or incorrect. They then finetune the target model using control tokens that are based on the calibrator's prediction and indicate the desired confidence level. In comparison, our method has multiple advantages: We only assume access to the input question and text outputs, not requiring white-box model access or finetuning of the target model. We further demonstrate in our experiments that using more fine-grained targets than the simple binary yields better calibration overall.

Predicting properties of generated text. Instead of trying to obtain a piece of information of interest—e.g. truthfulness or predictive confidence—from the original model, other works have investigated whether this can be done so through secondary (neural) models. For instance, Lahlou et al. (2023); Mukhoti et al. (2023) fit density estimators on internal model representations. In NLP, Pacchiardi et al. (2023) demonstrated that wrong statements by LLMs can be detected by collecting yes / no answers for a set of questions, storing them in a vector and fitting a logistic regression model. In general, using secondary neural models to predict properties of the generated text also has connections to other tasks such as translation quality estimation (Blatz et al., 2004; Quirk, 2004; Wang et al., 2019; Glushkova et al., 2021; Zerva et al., 2022b), toxicity classification (Masle-Krešňáková et al., 2020) or fine-grained reward modeling (Wu et al., 2023).

[2] https://github.com/parameterlab/apricot.

3 Methodology

Method	Black-box LLM?	Consistent?	Calibrated?
Seq. likelihoods	✗	✔	✗
Verb. uncertainty	✔	✗	✗
APRICOT 🍑 (ours)	✔	✔	✔

Table 1: Comparison of appealing attributes that LLM confidence quantification techniques should fulfil. They should ideally be applicable to black-box LLMs, be consistent (i.e., always elicit a response), and produce calibrated estimates of confidence.

Estimating the confidence of a LLM can be challenging, since their size rules out many traditional techniques that require finetuning or access to model parameters. In this light, using the likelihood of the generated sequence might seem like an appealing alternative; however, it might not actually reflect the reliability of the model's answer and often cannot be retrieved when using black-box models, where the only output is the generated text. Verbalized uncertainty, i.e. prompting the LLM to express its uncertainty in words, can be a solution when the model is powerful enough. But as we later show in Section 4, the generated confidence expressions are not very diverse, and results are not always *consistent*, meaning that the model does not always generate a desired confidence self-assessment. As we illustrate in Table 1, our method, APRICOT 🍑, fulfills all of these criteria: Through a one-time finetuning procedure of an auxiliary model on the target LLMs outputs, we have full control over a calibrated model that gives consistent and precise confidence estimates.

In Figure 2 we give an overview of APRICOT 🍑, which consists of three main steps: Firstly, we prompt the target LLM to generate training data for our auxiliary model (Section 3.1). Secondly, we set calibration targets in a way that does not require access to the target LLM beyond its generated outputs (Section 3.2). Lastly, we train the auxiliary calibrator to predict the target LLM's confidence for a given question (Section 3.3). Thereby, we contribute two parts that are agnostic to the LLM in question: The creation of calibration targets and their prediction through the auxiliary model. Note that we will use the terms auxiliary model or calibrator interchangeably in the following sections.

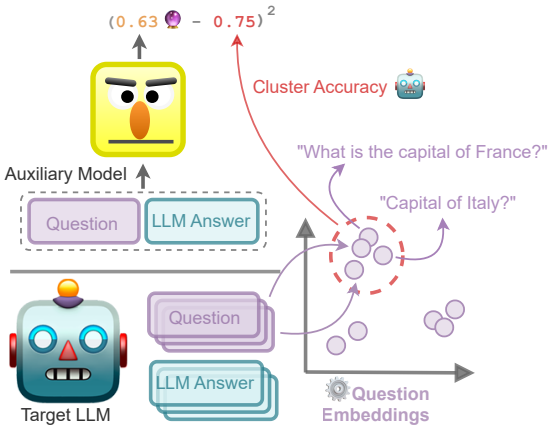


Figure 2: Full overview of APRICOT 🍑. We collect a LLM's answer to a set of questions and embed the latter using an embedding model. After clustering similar questions and identifying the LLM's accuracy on them, we can use this value as reference when training to predict the confidence from a question-answer pair.

(a) Default prompting.

(b) Chain-of-though prompting.

(c) Prompting with verbalized confidence.

Figure 3: Illustration of the prompting strategies used to generate the input data for the auxiliary calibrator. Note that (c) can also involve confidence expressed in words ("My confidence level is low") and that (b) and (c) can be combined. The exact prompts are listed in Appendix A.1.

3.1 Prompting the Target LLM

In this step, we generate finetuning data for the auxiliary model by prompting the target LLM on the given task. Here, we explore different varia-

tions to see which model response might provide the best training signal for the auxiliary calibrator. More concretely, while the original prompt and model generation might already suffice to predict the model's confidence, we also ask the model to elaborate on its answer using chain-of-thought prompting (Wei et al., 2022). We hypothesize that including additional reasoning steps could expose signals that are useful for the calibrator.[3] We furthermore take a model's assessment of its confidence into account, too. Recent works on *verbalized uncertainty* (Lin et al., 2022; Tian et al., 2023) investigated how to elicit such an assessment as a percentage value, e.g. "I am 95 % confident in my answer", or using linguistic expressions such as "My confidence is somewhat low". While previous studies like Zhou et al. (2024) have demonstrated the difficulty in obtaining reliable self-assessments, we can just treat them as additional input features, and let their importance be determined through the auxiliary model training. We illustrate the different prompting strategies in Figure 3 and list the exact prompts in Appendix A.1.

3.2 Setting Calibration Targets

After explaining the inputs to the auxiliary model, the question naturally arises about what the calibrator should be trained to predict. The work by Mielke et al. (2022) introduces an additional model that simply predicts individual answer correctness (and does so by using the target model's internal hidden states, which is not possible for black-box models). While we also check test this type of output in Section 4.2), we show that we can produce better calibration targets through clustering.

Background. We consider the notion of calibration by Guo et al. (2017): We define a predictor as *calibrated* when given some predicted outcome $\hat{y} \in \mathcal{Y}$ and some associated probability $\hat{p} \in [0, 1]$, we have

$$\mathbb{P}\big(Y = \hat{y} \mid P = \hat{p}\big) = \hat{p}, \tag{1}$$

i.e. the probability $\hat{p}$ corresponding to the actual relative frequency of correct predictions. This quantity is often approximated empirically through means such as the expected calibration error (ECE; Naeini et al., 2015). It evaluates the error

$$\mathbb{E}\Big[\big|\mathbb{P}\big(Y = \hat{y} \mid P = \hat{p}\big) - \hat{p}\big|\Big], \tag{2}$$

where in the case of the ECE, the expectation is computed by grouping N test predictions into M equally wide bins. Defining $\mathcal{B}_m$ as the set indices that belong to bin m, we can write the ECE as

$$\sum_{m=1}^{M} \frac{|\mathcal{B}_m|}{N} \Big| \underbrace{\frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} \mathbf{1}(\hat{y}_i = y_i)}_{\text{Bin accuracy (target)}} - \underbrace{\frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} \hat{p}_i}_{\text{Avg. bin confidence}} \Big|, \tag{3}$$

where $\mathbf{1}(\hat{y}_i = y_i)$ is the indicator function showing whether the prediction was correct. As Guo et al. (2017) note, both terms in the difference approximate the left-hand and right-hand side in Equation (1) per bin, respectively.

Contribution. Our key insight is here that we can optimize an objective similar to Equation (2), inspired by the way that the ECE in Equation (3) aggregates samples in homogeneous groups and measures the group-wise accuracy. This done without changing the LLM's original answers or access to token probabilities. Instead of creating bins $\mathcal{B}_m$ by confidence, which is not possible in a black-box setting, we create clustered sets $\mathcal{C}_m$ of inputs with similar sentence embeddings. Calibration targets are then obtained by using the observed accuracy per set $\mathcal{C}_m$. This is similar to Lin et al. (2022), who consider the accuracy per question category. Yet in the absence of such metadata, we expect good embedding and clustering algorithms to roughly group inputs by category. Höltgen and Williamson (2023) also echo a similar sentiment, describing how ECE's grouping by confidence can be abstracted to other kinds of similarities. They also provide a proof that the calibration error of a predictor based on a k-nearest neighbor clustering tends to zero in the infinite data limit.

Practically, we embed questions into a latent space using a light-weight model such as Sentence-BERT (Reimers and Gurevych, 2019), normalize the embeddings along the feature dimension (Timkey and van Schijndel, 2021), and then use HDBSCAN (Campello et al., 2013) to cluster them into questions of similar topic. The use of HDBSCAN has multiple advantages: Compared to e.g. k-means, we do not have to determine the numbers of clusters in advance, and since the clustering is conducted bottom-up, clusters are not constrained to a spherical shape. Furthermore, compared to its predecessor DBSCAN (Ester et al., 1996), HDBSCAN does not require one to determine the minimum distance between points for clustering man-

[3] We do this while acknowledging evidence by Turpin et al. (2023) that shows that any chain-of-thought reasoning might not reflect the actual reasons for a specific model response.

ually. We evaluate this procedure in Section 4.1 and Appendix A.4.

3.3 Training the Auxiliary Model

After determining the input and the training targets for the auxiliary model in the previous sections, we can now describe the actual training procedure that makes it predict the target LLM's confidence. To start, we feed the questions alongside some in-context samples into our target LLM. We retain the generated answers and create a dataset that combines the question (without in-context samples) and the target model's answers. These are used to train the auxiliary calibrator to predict the calibration targets obtained by the clustering procedure above. In our experiments, we use DeBERTaV3 (He et al., 2023), an improvement on the original DeBERTa model (He et al., 2021) using ELECTRA-style pretraining (Clark et al., 2020) and other improvements. We then finetune it using the AdamW optimizer (Loshchilov and Hutter, 2018) in combination with a cosine learning rate schedule. We minimize the following mean squared error, where $\hat{p}_i$ is the predicted confidence, $\mathcal{C}(i)$ the cluster that the input question with index i belongs to and $\hat{a}_j$ an answer given by the target LLM:

$$\Big(\hat{p}_i - \underbrace{\frac{1}{|\mathcal{C}(i)|}\sum_{j\in\mathcal{C}(i)} \mathbf{1}(\hat{a}_j \text{ is correct})}_{\text{Cluster accuracy (target)}}\Big)^2. \quad (4)$$

We also explore a variant that simply predicts whether the LLM's answer is expected to be correct or incorrect. In this case, we optimize a binary cross-entropy loss with loss weights.[4] Finally, we select the final model via the best loss on the validation set. We determine the learning rate and weight decay term through Bayesian hyperparameter search (Snoek et al., 2012), picking the best configuration by validation loss. We detail search ranges and found values in Appendix A.3. Training hardware and the environmental impact are discussed in Appendix A.6.

4 Experiments

We now demonstrate how APRICOT 🍑 provides a simple yet effective solution to calibrate LLMs. Before assessing the quality of the unsupervised clustering to determine calibration targets from Section 3, we first introduce the dataset and models.

Datasets. We employ TriviaQA (Joshi et al., 2017), a common (closed-book) question-answering dataset. Open-ended question answering is an ideal testbed for natural language generation tasks, since it is comparatively easy to check whether an answer is correct or not, so calibration has an intuitive interpretation. To preprocess TriviaQA, we create a training set of 12k examples and choose another 1.5k samples as a validation and test split, respectively.[5] Secondly, we run experiments on CoQA (Reddy et al., 2019), a conversational question-answering dataset in which the model is quizzed about the information in a passage of text. We treat the dataset as an open-book dataset, where the model is shown the passage and then asked one of the corresponding questions at a time. We extract a subset of the dataset to match the split sizes of TriviaQA.

Models. For our white-box model experiments, we choose a 7 billion parameter variant of the Vicuna v1.5 model (Zheng et al., 2023),[6] an instruction-finetuned model originating from Llama 2 (Touvron et al., 2023). For the black-box model, we opt for OpenAI's GPT-3.5 (OpenAI, 2022).[7] Despite recent API changes granting access to token probabilities,[8] creating methods for black-box confidence estimation is still relevant for multiple reasons: Token probabilities are not available for most black-box models, they might be removed again to defend against potential security issues; and they are not always a reliable proxy for confidence.

4.1 Setting Calibration Targets by Clustering

Before beginning our main experiments, we would like to verify that our proposed methodology in Section 3.2 is sound. In particular, clustering the embeddings of questions and computing the calibration confidence targets rests on the assumption that similar questions are collected in the same

[4] The loss weights are based on `scikit-learn`'s implementation using the "balanced" mode, see https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html.

[5] Since the original test split does not include answers, we generate the validation and test split from the original validation split.

[6] https://huggingface.co/lmsys/vicuna-7b-v1.5.

[7] Specifically, using version `gpt-3.5-turbo-0125`.

[8] https://x.com/OpenAIDevs/status/1735730662362189872 (last accessed on 16.01.24).

	TriviaQA		CoQA	
	Textual	Semantic	Textual	Semantic
Random	.11 ±.08	.00 ±.08	.08 ±.12	.00 ±.12
Clustering	.39 ±.28	.60 ±.14	.47 ±.25	.70 ±.17

Table 2: Results of evaluation of found clusters on TriviaQA and CoQA, including one standard deviation.

cluster. Ideally, we would like to check this using metadata, which however is usually not available.

Setup. Instead, we evaluate this through different means: We first use the `all-mpnet-base-v2` model from the sentence transformers package (Reimers and Gurevych, 2019) and HDBSCAN with a minimum cluster size of 3 to cluster questions. We then analyze the textual and semantic similarity of questions in a cluster by computing the average pair-wise ROUGE-L score (*semantic*; Lin, 2004)[9] between questions and cosine similarities between question *embeddings* of the same cluster (*semantic*). Since performing this evaluation on the entire dataset is computationally expensive, we approximate the score by using 5 pairwise comparisons per cluster, with 200 comparisons for ROUGE-L and 1000 for cosine similarity in total, respectively. As a control for our method (*clustering*), we also compute values between unrelated questions that are not in the same cluster (*random*).

Results. We show the results of this analysis in Table 2. We observe noticeable differences between the random baseline and the similarity for the clustering scores, both on a textual and semantic level. While there is smaller difference on a textual level due to the relatively similar wording of questions, the semantic similarity based on the encoded questions is very notable. We provide deeper analyses of this part in Appendix A.4, showing that this method creates diverse ranges of calibration confidence targets. This suggests two things: On the one hand, our proposed methodology is able to identify fine-grained categories of questions. On the other hand, the diversity in calibration targets indicates that we detect sets of questions on which the LLM's accuracy varies—and that this variety should be reflected. We test the ability of different methods to do exactly this next.

[9] As implemented by the `evaluate` package, see https://huggingface.co/docs/evaluate/index.

4.2 Calibrating White- and Black-Box Models

We are interested to see whether auxiliary models can reliably predict the target LLM's confidence. We describe our experimental conditions below.

Evaluation metrics. Aside from reporting the accuracy on the question answering task, we also report several calibration metrics, including the expected calibration error (ECE; Naeini et al., 2015) using 10 bins. In order to address any distortion of results introduced by the binning procedure, we use smooth ECE (smECE; Błasiok and Nakkiran, 2023), which avoids the binning altogether by smoothing observations using a RBF kernel. We also consider Brier score (Brier, 1950), which can be interpreted as mean-squared error for probabilistic predictions. We further show how indicative the predicted confidence is for answering a question incorrectly by measuring the AUROC. The AUROC treats the problem as a binary misprediction detection task based on the confidence scores, aggregating the results over all possible decision thresholds. In each case, we report the result alongside a bootstrap estimate of the standard error (Efron and Tibshirani, 1994) estimated from 100 samples and test for significance using the ASO test (Del Barrio et al., 2018; Dror et al., 2019; Ulmer et al., 2022b) with $\tau = 0.35$ and a confidence level of $\alpha = 0.1$.

Baselines. To contextualize the auxiliary calibrator results, we consider the following baselines: We consider the raw (length-normalized) sequence likelihoods (Seq. likelihood) as well as variant using Platt scaling (Platt et al., 1999): Using the raw likelihood $\hat{p} \in [0, 1]$ and the sigmoid function σ, we fit two additional scalars $a, b \in \mathbb{R}$ to minimize the mean squared error on the validation set to produce a calibrated likelihood $\hat{q} = \sigma(a\hat{p} + b)$ while keeping all other calibrator parameters fixed. We also compare it to the recent method of *verbalized* uncertainty (Lin et al., 2022; Tian et al., 2023; Xiong et al., 2023b), where we ask the model to assess its confidence directly. We do this by asking for confidence in percent (Verbalized %) and using a seven-point scale from "very low" to "very high" and which is mapped back to numeric confidence scores (Verbalized Qual.). Where applicable, we also distinguish between baselines with and without chain-of-thought prompting (CoT; Wei et al., 2022). We detail this scale and corresponding prompts in Appendix A.1. For our approach, we distinguish between confidence targets obtained

through the procedure in Section 3.2 (clustering) and simply predicting whether the given answer is correct or incorrect (binary).

Results. Vicuna v1.5 7B achieves 58% accuracy on TriviaQA and 44% on CoQA, while GPT-3.5 obtains 85% and 55% accuracy, respectively.[10] We present the calibration results in Table 3. APRICOT 🍑 hereby achieves the highest AUROC in all settings and among the lowest Brier scores and calibration errors. On the latter metric, verbalized confidence beats our method, but often at the cost of a higher worst-case calibration error and lower AUROC. In addition, the qualitative verbalized uncertainty does not work reliably for the smaller Vicuna v1.5 model. CoT prompting seems to increase the success rate of verbalized uncertainty. The additional results on GPT-3.5 suggests that this ability might also be dependent on model size. The effect of CoT prompting on calibration, however, remains inconsistent across different baselines. While verbalized uncertainties often perform well according to calibration error, these results have to be taken with a grain of salt: Especially for the relatively small 7B Vicuna v1.5 model, the generations do not always contain the desired confidence expression, as visible by the low success rate. And even when taking the generated confidence expression, their ability to distinguish potentially correct from incorrect LLM responses remains at or close to random level. Lastly, APRICOT 🍑 with clustering beats the use of binary targets for Vicuna v1.5 and GPT-3.5 on both TriviaQA and CoQA. We also juxtapose reliability diagrams for the different methods for Vicuna v1.5 on TriviaQA in Figure 4 (we show the other reliability diagrams, including for GPT-3.5, in Appendix A.5). Here it becomes clear that verbalized uncertainties approaches usually do not emit a wide variety of confidence scores. This is in line with observations by Zhou et al. (2023), who hypothesize the distribution of expressions generated by verbalized uncertainty heavily depend on the mention of e.g. percentage values in the model's training data. While Figure 10 shows that GPT-3.5 provides more variety in this regard, the overall phenomenon persists. We now conduct some additional analyses based on the clustering-based variant of our method.

[10]We describe the evaluation protocol in Appendix A.2. Since GPT-3.5 is a closed-source model, it is hard to say whether the higher accuracy scores are due to better model quality, test data leakage, or overlap in questions in the case of TriviaQA (Lewis et al., 2021).

4.3 What does the calibrator learn from?

The previous results pose the question of which parts of input the auxiliary model actually learns from. So, analogous to the different prompting strategies in Figure 3, we explore different input variants: First, we test question-only, where the target LLM's answer is omitted completely. We also test the performance of the calibrator when given more information, for instance the model answer with and without chain-of-thought prompting, which could potentially expose flaws in the LLM's response.[11] Finally, we also expose the verbalized uncertainty of the LLM to the calibrator.

Results. We show these results in Table 8 in Appendix A.5. Interestingly, we can observe that even based on the question to the LLM alone, APRICOT 🍑 can already achieve respectable performance across all metrics. This suggests that the calibrator at least partially learns to infer the difficulty of the LLM answering a question from the type of question alone. Nevertheless, we also find that adding the LLM's actual answer further improves results, with additional gain when using CoT prompting. In some cases, the calibration error can be improved when using the LLM's verbalized uncertainties; in this sense, we can interpret the role of the calibrator as mapping the model's own assessment to a calibrated confidence score.

5 Discussion

Despite the difficulty of predicting the LLM's confidence from its generated text alone, our experiments have shown that APRICOT 🍑 can be used to produce reasonable scores even under these strict constraints. We showed in the past sections that the auxiliary model can be finetuned to learn from multiple signals. On the one hand, the auxiliary calibrator learns a mapping from a latent category of question to the expected difficulty for a target LLM. On the other hand, including the answer given through CoT prompting and including the LLM's own assessment of its uncertainty helped to further improve results. While sometimes beaten in terms of calibration error, our method consistently outperforms our baselines in misprediction AUROC, meaning that it can provide the best signal to detect wrong LLM answers. Compared to

[11]Based on the recent study by Turpin et al. (2023), we assume that CoT does *not* expose the LLM's actual reasoning. Nevertheless, it provides more context about the given answer.

	Method	TriviaQA Success	TriviaQA Brier↓	TriviaQA ECE↓	TriviaQA smECE↓	TriviaQA AUROC↑	CoQA Success	CoQA Brier↓	CoQA ECE↓	CoQA smECE↓	CoQA AUROC↑
Vicuna v1.5 (white-box)	Seq. likelihood	-	.22 ±.01	.05 ±.00	.03 ±.00	.79 ±.01	-	.32 ±.01	.08 ±.00	.08 ±.00	.69 ±.01
	Seq. likelihood (CoT)	-	.25 ±.01	.04 ±.00	.04 ±.00	.70 ±.01	-	.35 ±.01	.04 ±.00	.05 ±.00	.61 ±.01
	Platt scaling	-	.24 ±.00	.08 ±.00	.07 ±.00	.70 ±.01	-	.30 ±.00	.03 ±.00	.03 ±.00	.69 ±.01
	Platt scaling (CoT)	-	.24 ±.00	.12 ±.00	.11 ±.00	.79 ±.01	-	.30 ±.00	.02 ±.00	.02 ±.00	.61 ±.01
	Verbalized Qual.	0.19	.38 ±.03	.02 ±.00	.02 ±.00	.62 ±.03	0.66	.45 ±.01	**.00** ±.00	**.00** ±.00	.48 ±.01
	Verbalized Qual. (CoT)	0.25	.39 ±.02	**.01** ±.00	**.01** ±.00	.60 ±.02	0.73	.45 ±.01	**.00** ±.00	**.00** ±.00	.48 ±.01
	Verbalized %	1.00	.39 ±.01	.38 ±.00	.27 ±.00	.52 ±.01	0.99	.49 ±.01	.48 ±.00	.32 ±.00	.53 ±.01
	Verbalized % (CoT)	1.00	.39 ±.01	.38 ±.00	.26 ±.00	.49 ±.01	0.99	.48 ±.01	.06 ±.00	.06 ±.00	.55 ±.01
	Auxiliary (binary)	-	.20 ±.01	.16 ±.01	.15 ±.01	.81 ±.01	-	.20 ±.01	.16 ±.01	.15 ±.01	**.82** ±.01
	Auxiliary (clustering)	-	**.18** ±.00	.09 ±.01	.09 ±.01	**.83** ±.01	-	**.18** ±.00	.04 ±.01	.04 ±.01	**.82** ±.01
GPT-3.5 (black-box)	Seq. likelihood	-	.15 ±.01	.04 ±.00	.04 ±.00	.69 ±.02	-	.29 ±.01	.11 ±.00	.11 ±.00	.70 ±.01
	Seq. likelihood (CoT)	-	.14 ±.00	.05 ±.00	.05 ±.00	.60 ±.02	-	.25 ±.00	**.01** ±.00	**.02** ±.00	.52 ±.02
	Platt scaling	-	.15 ±.00	.04 ±.00	.04 ±.00	.69 ±.02	-	.26 ±.01	.03 ±.00	.03 ±.00	.70 ±.01
	Platt scaling (CoT)	-	.15 ±.00	.12 ±.00	.12 ±.00	.60 ±.02	-	.25 ±.00	.06 ±.00	.06 ±.00	.52 ±.02
	Verbalized Qual.	1.00	.14 ±.01	.07 ±.00	.04 ±.00	.61 ±.02	1.00	.27 ±.00	.07 ±.00	.05 ±.00	.52 ±.01
	Verbalized Qual. (CoT)	1.00	.15 ±.00	.04 ±.00	.03 ±.00	.63 ±.02	1.00	.30 ±.01	.08 ±.01	.04 ±.00	.50 ±.01
	Verbalized %	1.00	.13 ±.01	.01 ±.00	**.01** ±.00	.63 ±.02	1.00	.34 ±.01	.25 ±.00	.22 ±.00	.54 ±.01
	Verbalized % (CoT)	0.99	.13 ±.01	**.00** ±.00	**.01** ±.00	.63 ±.02	0.58	.37 ±.01	.09 ±.01	.06 ±.00	.49 ±.02
	Auxiliary (binary)	-	.14 ±.00	.14 ±.01	.14 ±.01	.65 ±.02	-	.19 ±.01	.13 ±.01	.13 ±.01	**.81** ±.01
	Auxiliary (clustering)	-	**.12** ±.01	.06 ±.01	.06 ±.01	**.72** ±.02	-	**.18** ±.00	.02 ±.01	**.02** ±.00	**.81** ±.01

Table 3: Calibration results for Vicuna v1.5 and GPT-3.5 on TriviaQA and CoQA. We bold the best results per dataset and model, and underline those that are statistically significant compared to all other results assessed via the ASO test. Results are reported along with a bootstrap estimate of the standard error.

other approaches, this yields some desirable properties: APRICOT 🍑 is available when sequence likelihood is not; it is more reliable than verbalized uncertainty; and it only needs a light finetuning once, adding negligible inference overhead. Compared to other methods such as Kuhn et al. (2023); Lin et al. (2023), it also does not require more generations for the same input, reducing the more expensive LLM inference costs.

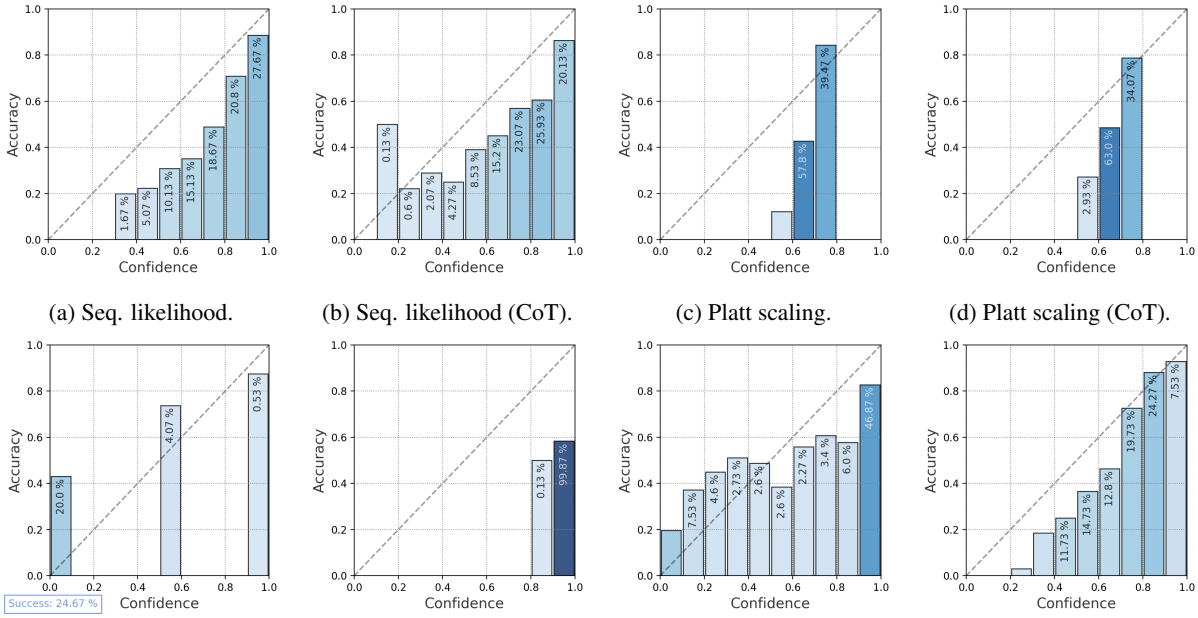
6 Conclusion

We presented APRICOT 🍑, a general method to obtain confidence scores from any language model on the input and text output alone. We showed that it is possible to compute calibration targets through the clustering of question embeddings. Through the subsequent finetuning of a smaller language model, we then outperform other methods to distinguish incorrect from correct answers with competitive calibration scores, on different models and datasets. While we only presented a first, more fundamental version of this approach in this work, it lends itself naturally to a whole body of research that aims to improve the calibration of pretrained language models (Desai and Durrett, 2020; Jiang et al., 2021; Chen et al., 2023). Lastly, future studies might also investigate the uncertainty of the auxiliary model itself and use techniques such as conformal prediction (Vovk et al., 2005; Papadopoulos et al., 2002; Angelopoulos and Bates, 2021) to produce estimates of LLM confidence *intervals*.

Limitations

Clustering. While yielding generally positive results in our case, the clustering methodology from Section 3.2 requires access to a sufficiently expressive sentence embedding model and a large enough number of data points. When this is not given, we show that the binary approach—tuning the auxiliary model to predict misprediction—is a viable alternative that can even outperform the clustering variant in some settings.

Distributional shift. As any neural model, the auxiliary calibrator might be prone to distributional shift and out-of-distribution data. Further research could help to understand how this issue can be reduced and which parts of the input the model



(a) Seq. likelihood. (b) Seq. likelihood (CoT). (c) Platt scaling. (d) Platt scaling (CoT).

(e) Verbalized Qual. (f) Verbalized %. (g) Auxiliary (binary). (h) Auxiliary (clustering).

Figure 4: Reliability diagrams for our different methods using 10 bins each for Vicuna v1.5 on TriviaQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.

identifies to predict confidence scores in order to unveil potential shortcut learning (Du et al., 2023).

Other types of language generation. Our experiments focused on open-ended question answering tasks, which provide an easy way to check answer correctness. In other types of language generation such as summarization, translation or open text generation, this notion is not given. However, we point out the relation of our approach to other NLP tasks in Section 2, which might provide an opening for future research.

Ethical Considerations

We mostly see any ethical considerations with our work arising in the general nature of neural models, which therefore also affects our auxiliary calibrator. The efficacy of neural models might vary on out-of-distribution data. In safety-sensitive applications, this also means that its predictions might be less trustworthy on certain sub-populations. In this case, explicit validation of the LLM's answers and the auxiliary calibrators corresponding predictions is necessary and further finetuning might be necessary. In these cases, reporting the confidence score alongside an answer might also be replaced by withholding the LLM's response, entirely.

Acknowledgements

This work was generously supported by the NAVER corporation.

References

Anastasios N Angelopoulos and Stephen Bates. 2021. A Gentle Introduction To Conformal Prediction And Distribution-Free Uncertainty Quantification. *ArXiv preprint*, abs/2107.07511.

Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. 2023. Uncertainty In Natural Language Generation: From Theory To Applications. *ArXiv preprint*, abs/2307.15703.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. Mars: Meaning-Aware Response Scoring For Uncertainty Estimation In Generative Llms. *ArXiv preprint*, abs/2402.11756.

Daniel Beck, Lucia Specia, and Trevor Cohn. 2016. Exploring Prediction Uncertainty In Machine Translation Quality Estimation. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 208–218.

Umang Bhatt, Javier Antorán, Yunfeng Zhang, Q Vera Liao, Prasanna Sattigeri, Riccardo Fogliato, Gabrielle Melançon, Ranganath Krishnan, Jason Stanley, Omesh Tickoo, et al. 2021. Uncertainty As A Form Of Transparency: Measuring, Communicating, And

Using Uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 401–413.

Lukas Biewald. 2020. Experiment Tracking With Weights And Biases. Software available from wandb.com.

Jarosław Błasiok and Preetum Nakkiran. 2023. Smooth Ece: Principled Reliability Diagrams Via Kernel Smoothing. *ArXiv preprint*, abs/2309.12236.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation For Machine Translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321.

Glenn W Brier. 1950. Verification Of Forecasts Expressed In Terms Of Probability. *Monthly weather review*, 78(1):1–3.

Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. 2013. Density-Based Clustering Based On Hierarchical Density Estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

Ilias Chalkidis. 2023. Chatgpt May Pass The Bar Exam Soon, But Has A Long Way To Go For The Lexglue Benchmark. *ArXiv preprint*, abs/2304.12202.

Jiuhai Chen and Jonas Mueller. 2023. Quantifying Uncertainty In Answers From Any Language Model Via Intrinsic And Extrinsic Confidence Assessment. *ArXiv preprint*, abs/2308.16175.

Lihu Chen, Alexandre Perez-Lebel, Fabian M Suchanek, and Gaël Varoquaux. 2024. Reconfidencing Llms From The Grouping Loss Perspective. *ArXiv preprint*, abs/2402.04957.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. 2023. A Close Look Into The Calibration Of Pre-Trained Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1343–1367.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-Training Text Encoders As Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large Legal Fictions: Profiling Legal Hallucinations In Large Language Models. *ArXiv preprint*, abs/2401.01301.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. 2022. Underspecification Presents Challenges For Credibility In Modern Machine Learning. *The Journal of Machine Learning Research*, 23(1):10237–10297.

Eustasio Del Barrio, Juan A Cuesta-Albertos, and Carlos Matrán. 2018. An Optimal Transportation Approach For Assessing Almost Stochastic Order. In *The Mathematics of the Uncertain*, pages 33–44.

Shrey Desai and Greg Durrett. 2020. Calibration Of Pre-Trained Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302.

Shehzaad Dhuliawala, Vilém Zouhar, Mennatallah El-Assady, and Mrinmaya Sachan. 2023. A Diachronic Perspective On User Trust In Ai Under Uncertainty. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5567–5580. Association for Computational Linguistics.

Rotem Dror, Segev Shlomov, and Roi Reichart. 2019. Deep Dominance - How To Properly Compare Deep Neural Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2785.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut Learning Of Large Language Models In Natural Language Understanding. *Communications of the ACM*, 67(1):110–120.

Bradley Efron and Robert J Tibshirani. 1994. *An Introduction To The Bootstrap*.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm For Discovering Clusters In Large Spatial Databases With Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-Aware Machine Translation Evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3920–3938.

Micah Goldblum, Anima Anandkumar, Richard Baraniuk, Tom Goldstein, Kyunghyun Cho, Zachary C Lipton, Melanie Mitchell, Preetum Nakkiran, Max Welling, and Andrew Gordon Wilson. 2023. Perspectives On The State And Future Of Deep Learning–2023. *ArXiv preprint*, abs/2312.09323.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration Of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330.

Mehedi Hasan, Moloud Abdar, Abbas Khosravi, Uwe Aickelin, Pietro Lio, Ibrahim Hossain, Ashikur Rahman, and Saeid Nahavandi. 2023. Survey On Leveraging Uncertainty Estimation Towards Trustworthy Deep Neural Networks: The Case Of Reject Option And Post-Training Processing. *ArXiv preprint*, abs/2304.04906.

Jianxing He, Sally L Baxter, Jie Xu, Jiming Xu, Xingtao Zhou, and Kang Zhang. 2019. The Practical Implementation Of Artificial Intelligence Technologies In Medicine. *Nature medicine*, 25(1):30–36.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving Deberta Using Electra-Style Pre-Training With Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-Enhanced Bert With Disentangled Attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Benedikt Höltgen and Robert C Williamson. 2023. On The Richness Of Calibration. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1124–1138.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2023. Decomposing Uncertainty For Large Language Models Through Input Clarification Ensembling. *ArXiv preprint*, abs/2311.08718.

Alon Jacovi, Ana Marasović, Tim Miller, and Yoav Goldberg. 2021. Formalizing Trust In Artificial Intelligence: Prerequisites, Causes And Goals Of Human Trust In Ai. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 624–635.

Mingjian Jiang, Yangjun Ruan, Sicong Huang, Saifei Liao, Silviu Pitis, Roger Baker Grosse, and Jimmy Ba. 2023. Calibrating Language Models Via Augmented Prompt Ensembles.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How Can We Know *When* Language Models Know? On The Calibration Of Language Models For Question Answering. *Trans. Assoc. Comput. Linguistics*, 9:962–977.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TRiviaQa: A Large Scale Distantly Supervised Challenge Dataset For Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic Uncertainty: Linguistic Invariances For Uncertainty Estimation In Natural Language Generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying The Carbon Emissions Of Machine Learning. *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019*.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. 2023. Deup: Direct Epistemic Uncertainty Prediction. *Trans. Mach. Learn. Res.*, 2023.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question And Answer Test-Train Overlap In Open-Domain Question Answering Datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008.

Q Vera Liao and S Shyam Sundar. 2022. Designing For Responsible Trust In Ai Systems: A Communication Perspective. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1257–1268.

Chin-Yew Lin. 2004. Rouge: A Package For Automatic Evaluation Of Summaries. In *Text Summarization Branches Out*, pages 74–81.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching Models To Express Their Uncertainty In Words. *Trans. Mach. Learn. Res.*, 2022.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating With Confidence: Uncertainty Quantification For Black-Box Large Language Models. *ArXiv preprint*, abs/2305.19187.

Ilya Loshchilov and Frank Hutter. 2018. Fixing Weight Decay Regularization In Adam.

Kadan Lottick, Silvia Susai, Sorelle A. Friedler, and Jonathan P. Wilson. 2019. Energy Usage Reports: Environmental Awareness As Part Of Algorithmic Accountability. *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019*.

Andrey Malinin and Mark J. F. Gales. 2021. Uncertainty Estimation In Autoregressive Structured Prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Gary Marcus. 2020. The Next Decade In Ai: Four Steps Towards Robust Artificial Intelligence. *ArXiv preprint*, abs/2002.06177.

Viera Maslej-Krešňáková, Martin Sarnovskỳ, Peter Butka, and Kristína Machová. 2020. Comparison Of Deep Learning Models And Various Text Pre-Processing Techniques For The Toxic Comments Classification. *Applied Sciences*, 10(23):8631.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing Conversational Agents' Overconfidence Through Linguistic Calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip HS Torr, and Yarin Gal. 2023. Deep Deterministic Uncertainty: A New Simple Baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24384–24394.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2901–2907.

Lucia Nalbandian. 2022. An Eye For An 'I:'A Critical Assessment Of Artificial Intelligence Tools In Migration And Asylum Management. *Comparative Migration Studies*, 10(1):1–23.

OpenAI. 2022. Introducing Chatgpt.

Lorenzo Pacchiardi, Alex J Chan, Sören Mindermann, Ilan Moscovitz, Alexa Y Pan, Yarin Gal, Owain Evans, and Jan Brauner. 2023. How To Catch An Ai Liar: Lie Detection In Black-Box Llms By Asking Unrelated Questions. *ArXiv preprint*, abs/2309.15840.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. 2002. Inductive Confidence Machines For Regression. In *Machine Learning: ECML 2002: 13th European Conference on Machine Learning Helsinki, Finland, August 19–23, 2002 Proceedings 13*, pages 345–356. Springer.

John Platt et al. 1999. Probabilistic Outputs For Support Vector Machines And Comparisons To Regularized Likelihood Methods. *Advances in large margin classifiers*, 10(3):61–74.

Christopher B. Quirk. 2004. Training A Sentence-Level Machine Translation Confidence Measure. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. COQa: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Nils Reimers and Iryna Gurevych. 2019. Sentence-Bert: Sentence Embeddings Using SIamese Bert-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni. 2021. Codecarbon: Estimate And Track Carbon Emissions From Machine Learning Computing.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian Optimization Of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 2960–2968.

Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua V. Dillon, Jie Ren, and Zachary Nado. 2019. Can You Trust Your Model'S Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13969–13980.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just Ask For Calibration: Strategies For Eliciting Calibrated Confidence Scores From Language Models Fine-Tuned With Human Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5433–5442.

William Timkey and Marten van Schijndel. 2021. All Bark And No Bite: Rogue Dimensions In Transformer Language Models Obscure Representational Quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open Foundation And Fine-Tuned Chat Models. *ArXiv preprint*, abs/2307.09288.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language Models Don't Always Say What They Think: Unfaithful Explanations In Chain-Of-Thought Prompting. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Dennis Ulmer, Jes Frellsen, and Christian Hardmeier. 2022a. Exploring Predictive Uncertainty And Calibration In Nlp: A Study On The Impact Of Method & Data Scarcity. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2707–2735.

Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. 2022b. Deep-Significance: Easy And Meaningful Signifcance Testing In The Age Of Neural Networks. In *ML Evaluation Standards Workshop at the Tenth International Conference on Learning Representations*.

Dennis Ulmer, Lotta Meijerink, and Giovanni Cinà. 2020. Trust Issues: Uncertainty Estimation Does Not Enable Reliable Ood Detection On Medical Tabular Data. In *Machine Learning for Health*, pages 341–354. PMLR.

Siri L van der Meijden, Anne AH de Hond, Patrick J Thoral, Ewout W Steyerberg, Ilse MJ Kant, Giovanni Cinà, and M Sesmu Arbous. 2023. Intensive Care Unit Physicians' Perspectives On Artificial Intelligence–Based Clinical Decision Support Tools: Preimplementation Survey Study. *JMIR Human Factors*, 10:e39114.

Jordy Van Landeghem, Matthew Blaschko, Bertrand Anckaert, and Marie-Francine Moens. 2022. Benchmarking Scalable Predictive Uncertainty In Text Classification. *IEEE Access*.

Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023. Hybrid Uncertainty Quantification For Selective Text Classification In Ambiguous Tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 11659–11681.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic Learning In A Random World*, volume 29.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019. Improving Back-Translation With Uncertainty-Based Confidence Estimation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 791–802.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. 2021. Understanding How Dimension Reduction Tools Work: An Empirical Approach To Deciphering T-Sne, Umap, Trimap, And Pacmap For Data Visualization. *J. Mach. Learn. Res.*, 22:201:1–201:73.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-Grained Human Feedback Gives Better Rewards For Language Model Training. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Tim Z Xiao, Aidan N Gomez, and Yarin Gal. 2020. Wat Zei Je? Detecting Out-Of-Distribution Translations With Variational Transformers. *ArXiv preprint*, abs/2006.08344.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023a. Can Llms Express Their Uncertainty? An Empirical Evaluation Of Confidence Elicitation In Llms. *ArXiv preprint*, abs/2306.13063.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023b. Can Llms Express Their Uncertainty? An Empirical Evaluation Of Confidence Elicitation In Llms. *arXiv preprint arXiv:2306.13063*.

Adam X Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. 2023. Bayesian Low-Rank Adaptation For Large Language Models. *ArXiv preprint*, abs/2308.13111.

Chrysoula Zerva, Taisiya Glushkova, Ricardo Rei, and André F. T. Martins. 2022a. Disentangling Uncertainty In Machine Translation Evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8622–8641.

Chrysoula Zerva, Taisiya Glushkova, Ricardo Rei, and André FT Martins. 2022b. Better Uncertainty Quantification For Machine Translation Evaluation. *arXiv e-prints*, pages arXiv–2204.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging Llm-As-A-Judge With Mt-Bench And Chatbot Arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Kaitlyn Zhou, Jena D Hwang, Xiang Ren, and Maarten Sap. 2024. Relying On The Unreliable: The Impact Of Language Models' Reluctance To Express Uncertainty. *ArXiv preprint*, abs/2401.06730.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating The Grey Area: Expressions Of Overconfidence And Uncertainty In Language Models. *ArXiv preprint*, abs/2302.13439.

A Appendix

This appendix is structured as follows: We list all used prompts in Appendix A.1, briefly explore the evaluation protocal for the QA task in Appendix A.2 and list hyperparameter search details for replicability in Appendix A.3. The rest of the appendix is dedicated to additional results, namely for the analysis of the clustering step in Appendix A.4 or for the calibration experiments in Appendix A.5. We also list our compute hardware and its environmental impact in Appendix A.6.

A.1 Prompting Setup

In the following we elaborate on the prompts used in this work. In general, we use a very simple prompt for question answering, where we fill in a template of the form "Question: {Question} Answer:". For in-context samples, we prepend the demonstrations to the input, using the sample template as above. In the case of chain-of-thought prompting, we use the prompting below:

QA Chain-of-thought prompt

Briefly answer the following question by thinking step by step. Question: {Question} Answer:

In the case of CoQA, we slightly adjust the prompt template to the following:

CoQA Chain-of-thought prompt

Context: {Context}
Instruction: Briefly answer the following question by thinking step by step.
Question: {Question}
Answer:

Note that here the passage that questions are based on is given first, and chain-of-thought prompting is signaled through the "Instruction" field. When no chain-of-thought prompting is used, this field is omitted. For the verbalized uncertainty, we use the following prompts, in which case we omit any in-context samples:

Expression	Value
Very low	0
Low	0.3
Somewhat low	0.45
Medium	0.5
Somewhat high	0.65
High	0.7
Very high	1

Table 4: Mappings between confidence expressions for qualitative uncertainty and numerical confidence scores.

Verbalized uncertainty prompt (quantitative)

{Question} {Model answer} Please provide your confidence in the answer only as one of 'Very Low' / 'Low' / 'Somewhat Low' / 'Medium' / 'Somewhat High' / 'High' / 'Very High':

When mapping these expressions back to probabilities using the mapping in Table 4.

Verbalized uncertainty prompt (qualitative)

{Question} {Model answer} Please provide your confidence in the answer only in percent (0-100 %):

We follow Kuhn et al. (2023) and use 10 in-context samples for the original answer, which are randomly sampled from the training set (but in contrast to Kuhn et al., we sample different examples for each instance). When prompting for verbalized uncertainty, we remove these in-context samples.

A.2 Question-Answering Evaluation Protocol

In order to check model answers automatically, we take inspiration from the evaluation protocol by Kuhn et al. (2023). They use ROUGE-L (Lin, 2004) to compare the LLM's answer against the reference answer, and consider the answer as correct if the resulting score surpasses 0.3. We improve this protocol by adding the following condition: If the gold answer can be found verbatim in the generated answer, the answer is also considered correct.

A.3 Finetuning Hyperparameters

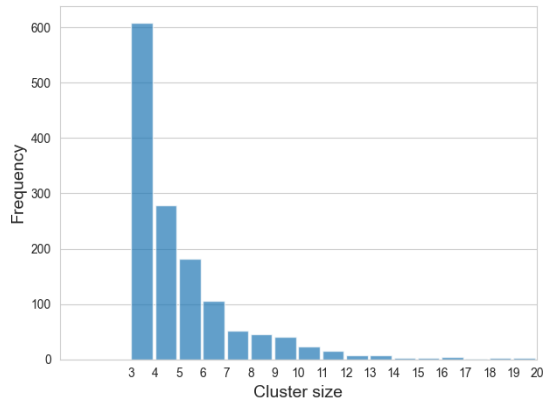
We conduct suites of hyperparameter searches per target LLM, dataset and type of calibration targets (binary and clustering) corresponding to the results in Table 3, resulting in eight different suites. We then use these found hyperparameters for the results in Table 8.

Search method and ranges. For the search, we opt for Bayesian hyperparameter search (Snoek et al., 2012) as implemented by Weights & Biases (Biewald, 2020). We optimize only two hyperparameters: Learning rate and weight decay. The learning rate is samples from a log-uniform distribution $\log \mathcal{U}[1 \times 10^{-5}, 0.01]$ and weight decay from a uniform distribution $\mathcal{U}[1 \times 10^{-4}, 0.05]$ for a total of 50 runs and 250 training steps each. The final hyperparameters selected are given in Table 5.

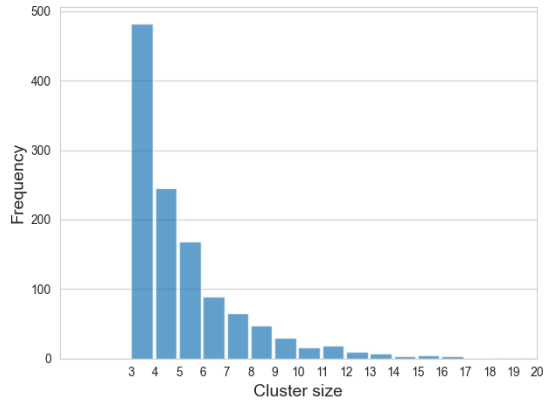
Other hyperparameters. When obtaining the responses from Vicuna v1.5 7B, we use a batch size of 4 and generate for a maximum of 50 tokens and stop generation when the model tries to generate parts of the prompt, such as "Question:" / "Q:" or "Answer:" / "A:". We also use 10 in-context samples for TriviaQA, but no in-context samples for

		TriviaQA		CoQA	
		Binary	Clustering	Binary	Clustering
Vicuna v1.5	learning rate	1.4×10^{-5}	3.37×10^{-5}	9.58×10^{-5}	8.84×10^{-5}
	weight decay	0.03184	0.008936	0.005793	7.42×10^{-4}
GPT-3.5	learning rate	2.96×10^{-5}	1.62×10^{-5}	5.12×10^{-5}	5.59×10^{-5}
	weight decay	0.01932	0.01362	0.03327	0.03495

Table 5: Chosen hyperparameters for our model on different datasets and for different calibration targets.



(a) Cluster sizes on TriviaQA.



(b) Cluster sizes on CoQA.

Figure 5: Bar plot of cluster sizes found. The plot is truncated at size 20.

CoQA. For the auxiliary calibrator, we use a context size of 512 tokens, batch size of 32, gradient clipping with a maximum norm of 10.

A.4 Additional Clustering Results

In this section we take a closer look at the results of the clustering procedure described in Section 3.2. In our experiments, we run HDBSCAN using a minimum cluster size of three, since preliminary experiments showed this number to produce the best trade-off between the coherence of cluster contents (as evaluated in Table 2) and a diversity in cluster targets. This setting yields a distribution of cluster sizes shown in Figure 5. We can see that the majority of cluster sizes are rather small, including questions on specific topics, some of which we display in Tables 6 and 7. Not shown are cluster sizes over 20 since the distribution quickly levels off, as well the set of all points that could not be sorted into any cluster.

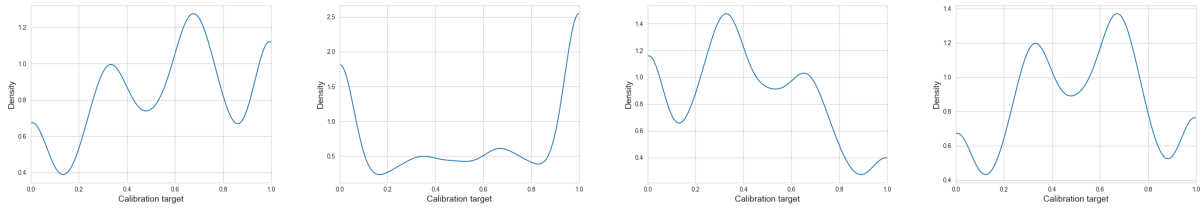
After clustering and computing the average accuracy per cluster, we obtain a distribution over calibration targets, which we show with density plots in Figure 6. Since most clusters are of size three, we can see clear modes around 0, 0.33, 0.66 and 1 for Vicuna v1.5 in Figure 6a. For GPT-3.5 in Figure 6b these are however less pronounced: We see that targets are often concentrated on 0 or 1, respectively. Similar spikes like in Figure 6a are observable for both models on CoQA in Figures 6c and 6d. This trend is also visible when plotting the assigned calibration targets per datapoint in Figure 7: While we can spot more transitionary colors between the blue and red extremes in the manifold for Figure 7a, the colors tend more to either of the options Figure 7b. These mode trends continue for CoQA in Figure 7c and Figure 7d.

A.5 Additional Calibration Results

Additional reliability plots. We show the all available reliability diagrams for Vicuna-v1.5 for TriviaQA in Figure 4 and CoQA in Figure 9, as well as the corresponding plots for GPT-3.5 in Figures 10 and 11. We can summarize some general trends: Sequence likelihood *can* be well-calibrated already, but this fact depends strongly on the dataset in question. And while our version of Platt scaling can improve results, it also narrows the range of confidence values to a narrow window. Verbalized uncertainty in both of variants also is not able to produce a wide variety of responses,

Cluster 1120
How many fluid ounces are in one quarter of an imperial pint?
How many fluid ounces in one Imperial pint?
How many fluid ounces in half an Imperial pint?
Cluster 920
Which famous US outlaw shot the cashier of a savings bank in Gallatin Missouri in 1869?
What famous outlaw committed the Wild West's first train robbery on July 21, 1873 in Adair, Iowa?
On July 21, 1873, Jesse James and the James-Younger gang pulled off the first successful what of the American West, in Adair Iowa?
Cluster 984
In what country was the game of golf invented?
Which ball game was invented by Dr James Naismith in Massachusetts USA in 1891?
It is generally accepted that the game of golf originated in what country?
What's a country where most people love playing rugby?
What's a country where most people love playing golf?
Cluster 64
Which alternative word for the Devil is a Hebrew word with translates as Lord Of The Flies?
Beelzebub is Hebrew for what phrase, which is also the title of a famous novel?
Diablo is another name for who?
Cluster 811
How many colors are there in the spectrum when white light is separated?
Which part of the eye contains about 137 million light-sensitive cells in one square inch?
Which of the retina's cells can distinguish between different wavelengths of light?
In four colour process printing, which is also known as CMYK, which are the only four colours that are used?
How many colours are in the rainbow?
In art, what are the three primary colours?
What color consists of the longest wavelengths of lights visible by the human eye?
What are the three primary colours of light?
Cluster 74
What was the name of the computer in the movie 2001: A Space Odyssey?
What was the name of the computer in Blake's Seven?
Developed by IBM, Deep Blue was a computer that played what?
What was the name of the PDA produced by Apple, most famous for its handwriting recognition software turning Random House into Condom Nose during a major presentation?

Table 6: Contents of some randomly sampled cluster that result from the clustering procedure for TriviaQA.

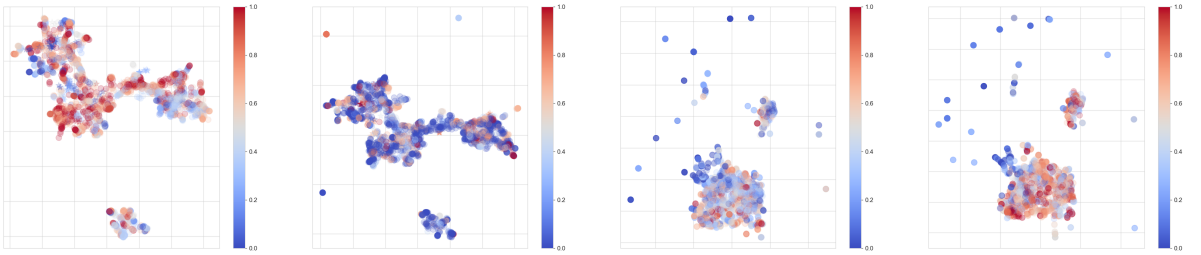


(a) Vicuna v1.5 on TriviaQA. (b) GPT-3.5 on TriviaQA. (c) Vicuna v1.5 on CoQA. (d) GPT-3.5 on CoQA.

Figure 6: Density plot of calibration targets generated through the clustering procedure for the two LLMs and TriviaQA / CoQA.

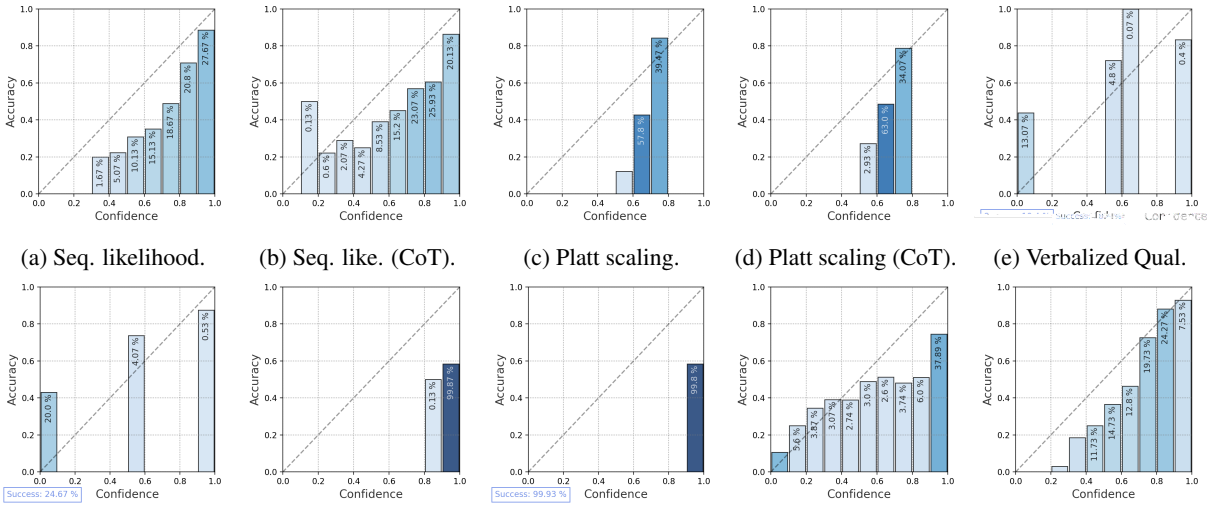
Cluster 823
Where in Europe is it located?
Is it in the European Plain?
Which region of Europe is it in?
Cluster 1176
Did she have children?
Does she have any children?
Did she have any children?
Did she have any other children?
Cluster 2244
Who won the Kentucky Derby?
as he won the Derby before?
Has he raced in the Derby before?
What were the winning horse's odds?
How many Derbys have their been?
Cluster 11
Are they densities of everything the same?
What is the densest elements at regular conditions?
What is density of a substance?
What is another symbol for density?
Who gives weight per unit volume as the definition?
Where is density the same value as it's mass concentration?
To make comparisons easier what stands in for density?
What is the relative density of something that floats?
Cluster 1081
Who was murdered?
who was killed?
Who committed this murder?
who was killed?
Who was killed?
who was killed?
Cluster 579
Did it succeed?
DId they continue to try?
did they succeed?
Was it successful?
Did they expect that success?

Table 7: Contents of some randomly sampled cluster that result from the clustering procedure for CoQA.



(a) Vicuna v1.5 on TriviaQA. (b) GPT-3.5 on TriviaQA. (c) Vicuna v1.5 on CoQA. (d) GPT-3.5 on CoQA.

Figure 7: Illustrating questions from TriviaQA along with their assigned confidence targets for the two LLMs, signified through their color from dark blue (0) to dark red (1). To avoid clutter, we subsampled 40% of the combined datasets to be shown here and used PaCMAP (Wang et al., 2021) to transform their sentence embeddings into 2D space.



(a) Seq. likelihood. (b) Seq. like. (CoT). (c) Platt scaling. (d) Platt scaling (CoT). (e) Verbalized Qual.

(f) Verb. Qual. (CoT). (g) Verbalized Quant. (h) Verb. Quant. (CoT). (i) Auxiliary (binary). (j) Aux. (clustering).

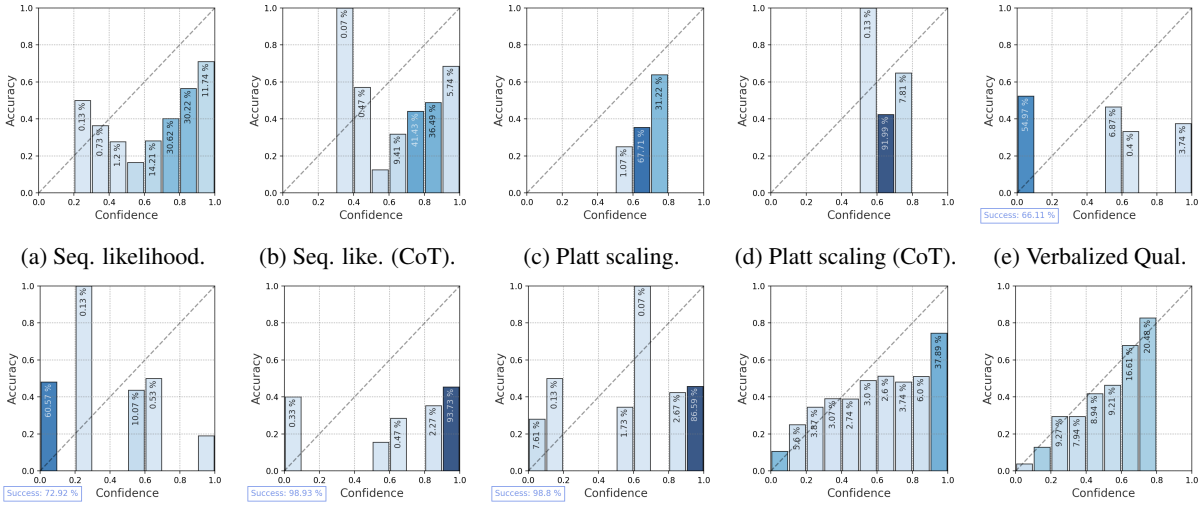
Figure 8: Reliability diagrams for our different methods using 10 bins each for Vicuna v1.5 7B on TriviaQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.

even though this effect is slightly less pronounced for GPT-3.5. This could be explained by previous observations by Zhou et al. (2023) that certain mentions of percentage values are over-represented in model training data. The auxiliary model is able to predict a wider array of confidence values in all settings, with the clustering variant achieving better calibration overall.

Ablation results. We show the results of the different ablations in Table 8. For both LLMs, we can see that the auxiliary model is able to achieve already decent scores by using the question as input alone. This suggest that the calibrator is learning about the general difficulty of questions for the LLM. However, both cases also show that including more information—the LLM's answer, CoT reasoning or verbalized uncertainties—can help to improve the auxiliaries model calibration and misprediction AUROC even further, even though the effect remains somewhat inconsistent across models.

A.6 Environmental Impact

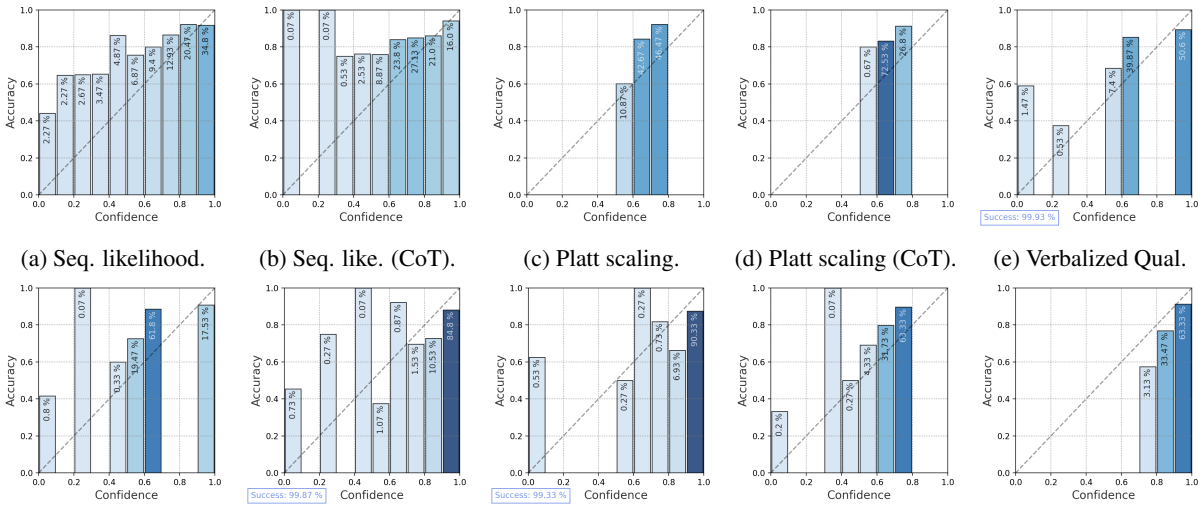
All experiments are run on a single V100 NVIDIA GPU. Using `codecarbon` (Schmidt et al., 2021; Lacoste et al., 2019; Lottick et al., 2019), we estimate finetuning the auxiliary calibrator on it to amount to 0.05 $kgCO_2eq$ of emission with an estimated carbon efficiency of 0.46 $kgCO_2eq$ / kWH. Therefore, we estimate total emissions of around 1 $kgCO_2eq$ to replicate all the experiments in our work.



(a) Seq. likelihood. (b) Seq. like. (CoT). (c) Platt scaling. (d) Platt scaling (CoT). (e) Verbalized Qual.

(f) Verb. Qual. (CoT). (g) Verbalized %. (h) Verb. % (CoT). (i) Auxiliary (binary). (j) Aux. (clustering).

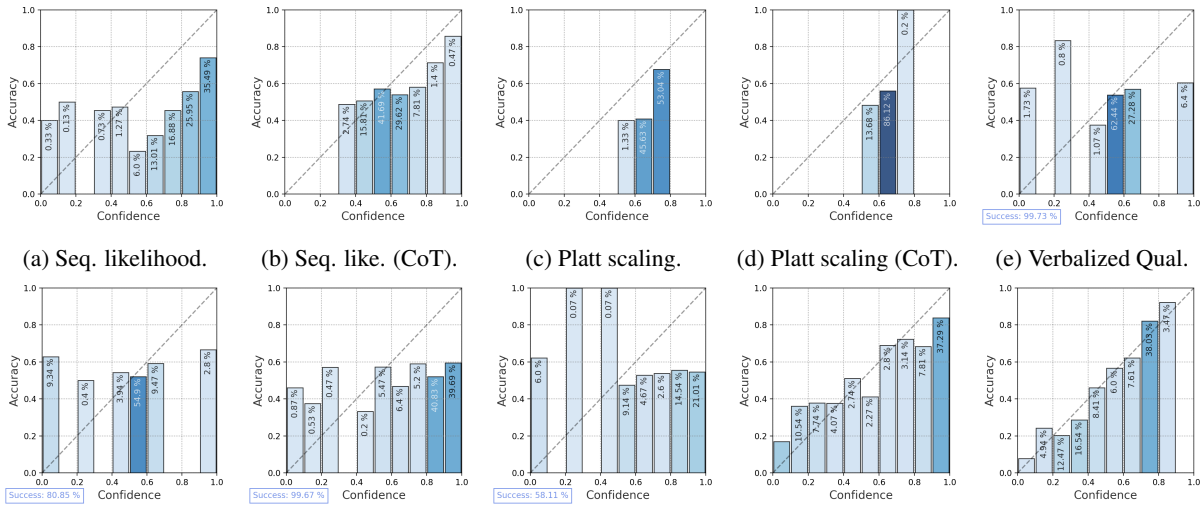
Figure 9: Reliability diagrams for our different methods using 10 bins each for Vicuna v1.5 7B on CoQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.



(a) Seq. likelihood. (b) Seq. like. (CoT). (c) Platt scaling. (d) Platt scaling (CoT). (e) Verbalized Qual.

(f) Verb. Qual. (CoT). (g) Verbalized %. (h) Verb. % (CoT). (i) Auxiliary (binary). (j) Aux. (clustering).

Figure 10: Reliability diagrams for our different methods using 10 bins each for GPT-3.5 on TriviaQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.



(a) Seq. likelihood. (b) Seq. like. (CoT). (c) Platt scaling. (d) Platt scaling (CoT). (e) Verbalized Qual.

(f) Verb. Qual. (CoT). (g) Verbalized %. (h) Verb. % (CoT). (i) Auxiliary (binary). (j) Aux. (clustering).

Figure 11: Reliability diagrams for our different methods using 10 bins each for GPT-3.5 on CoQA. The color as well as the percentage number within each bar indicate the proportion of total points contained in each bin.

	Auxiliary Model Input				TriviaQA				CoQA			
	Quest.	Ans.	CoT	Verb.	Brier↓	ECE↓	smECE↓	AUROC↑	Brier↓	ECE↓	smECE↓	AUROC↑
Vicuna v1.5 (white-box)	✔	✗	✗	✗	.21 ±.00	**.07** ±.01	**.06** ±.01	.74 ±.01	.22 ±.00	**.03** ±.01	**.03** ±.00	.70 ±.01
	✔	✔	✗	✗	**.18** ±.00	.09 ±.01	.09 ±.01	**.83** ±.01	**.18** ±.00	.04 ±.01	.04 ±.01	**.82** ±.01
	✔	✔	✗	**Qual.**	**.18** ±.00	.08 ±.01	.08 ±.01	.82 ±.01	.19 ±.00	.04 ±.01	.04 ±.01	.79 ±.01
	✔	✔	✗	%	**.18** ±.00	**.07** ±.01	.07 ±.01	.82 ±.01	**.18** ±.00	**.03** ±.01	**.03** ±.01	.80 ±.01
	✔	✔	✔	✗	.19 ±.01	**.07** ±.01	.07 ±.01	.80 ±.01	.21 ±.00	.04 ±.01	**.03** ±.01	.74 ±.01
	✔	✔	✔	**Qual.**	.19 ±.00	.08 ±.01	.08 ±.01	.80 ±.01	.22 ±.00	**.03** ±.01	**.03** ±.01	.70 ±.01
	✔	✔	✔	%	**.18** ±.00	**.07** ±.01	.07 ±.01	.81 ±.01	.20 ±.00	**.03** ±.01	**.03** ±.00	.75 ±.01
GPT-3.5 (black-box)	✔	✗	✗	✗	**.12** ±.01	.05 ±.01	.05 ±.01	.71 ±.03	.21 ±.00	.03 ±.01	.03 ±.01	.72 ±.01
	✔	✔	✗	✗	**.12** ±.01	.06 ±.01	.06 ±.01	.72 ±.02	**.18** ±.01	.04 ±.02	.04 ±.02	**.82** ±.02
	✔	✔	✗	**Qual.**	**.12** ±.01	**.03** ±.01	**.03** ±.01	.72 ±.03	**.18** ±.01	**.02** ±.01	**.02** ±.00	.80 ±.01
	✔	✔	✗	%	**.12** ±.01	**.03** ±.01	**.03** ±.01	.72 ±.02	**.18** ±.00	.04 ±.01	.03 ±.00	.80 ±.01
	✔	✔	✔	✗	**.12** ±.01	.06 ±.01	.06 ±.01	.72 ±.02	.21 ±.00	.03 ±.01	.03 ±.01	.72 ±.01
	✔	✔	✔	**Qual.**	**.12** ±.01	.04 ±.01	.04 ±.01	**.73** ±.02	.21 ±.00	.04 ±.01	.04 ±.01	.72 ±.01
	✔	✔	✔	%	**.12** ±.01	.04 ±.01	.04 ±.01	.64 ±.02	.21 ±.00	**.02** ±.01	**.02** ±.00	.72 ±.01

Table 8: Calibration results for Vicuna v1.5 and GPT-3.5 on TriviaQA and CoQA using the auxiliary (clustering) method. We bold the best results per dataset, method and model.