A Systematic Assessment of Weak-to-Strong Confidence Prediction in Large Language Models

Anonymous authors
Paper under double-blind review

Abstract

As large language models (LLMs) are deployed in increasingly diverse applications, understanding their capacity through uncertainty quantification (UQ) is crucial for ensuring safe and reliable behavior. Reliable uncertainty estimates that accompany the text generated by an LLM can signal when a response is likely to be incorrect and thus serve as an effective fail-safe mechanism against hallucinations. In this paper, we explore the extent to which the probability of a frontier model answering a query correctly can be predicted by smaller, weaker models with publicly available embeddings using a simple probe. We show that this probability can be predicted effectively, and the probes are easy to train, making oversight of large proprietary models more widely accessible. Leveraging embeddings from models as small as Llama3-8b, our predictor achieves 83.4% AUROC on TriviaQA and 64.3% on MMLU, and improves selective generator accuracy by up to 17.9%. We then carefully analyze the factors that affect the probe performance across six benchmarks and fifteen weak predictors. The performance does not simply improve with predictor model size, and that the weak-to-strong signal is robust to decoding configurations, label imbalance, and embedding aggregation choices. These findings support the view that representational compatibility between weak-model embeddings and strong model behavior matters more than model size alone. Overall, our results advance the understanding of weak-to-strong generalization and provide a simple, scalable framework for building more trustworthy LLMs.

1 Introduction

To what extent can a weaker language model predict a stronger language model's ability to answer a given question correctly? In this paper, we explore this question and find that a frozen open-access backbone paired with a two-layer probe on question-only embeddings can predict correctness of frontier models such as GPT-4o (OpenAI, 2024a) across diverse question-answering benchmarks.

Prior work on uncertainty quantification in large language models (LLMs) has largely focused on estimating a model's uncertainty from the model itself via self-reported probabilities (Kadavath et al., 2022), LLM-as-a-judge scoring (Liu et al., 2023; Zheng et al., 2023), or sampling-based disagreement (Wang et al., 2023), often without external evidence. These approaches are fundamentally introspective, probing a model's *internal* latent beliefs and decision process rather than using an *external* signal. As a result, they do not directly apply to strong, closed-weight models such as GPT, where we only observe the final answers. Moreover, recent work shows that such introspective signals can be highly sensitive to task design and prompt formats, leading to brittle calibration even when internal access is available (Dorner et al., 2025; Gu et al., 2024).

By contrast, it remains unclear when a *weaker* model, using for example question-only embeddings, can reliably predict a stronger black-box model's correctness across diverse tasks and generators (Burns et al., 2023). We address this gap with a systematic analysis of weak-to-strong confidence prediction in language models across six benchmarks and five groups of open-access models. Concretely, we study: given a question posed to one LLM (the *generator*), can a *weaker* LLM (the *predictor*) estimate the likelihood that the generator's answer is correct? We show that this is possible with a minimal evaluation setup: we apply a small probe on top of frozen embeddings from an open-access predictor to estimate the generator answer's

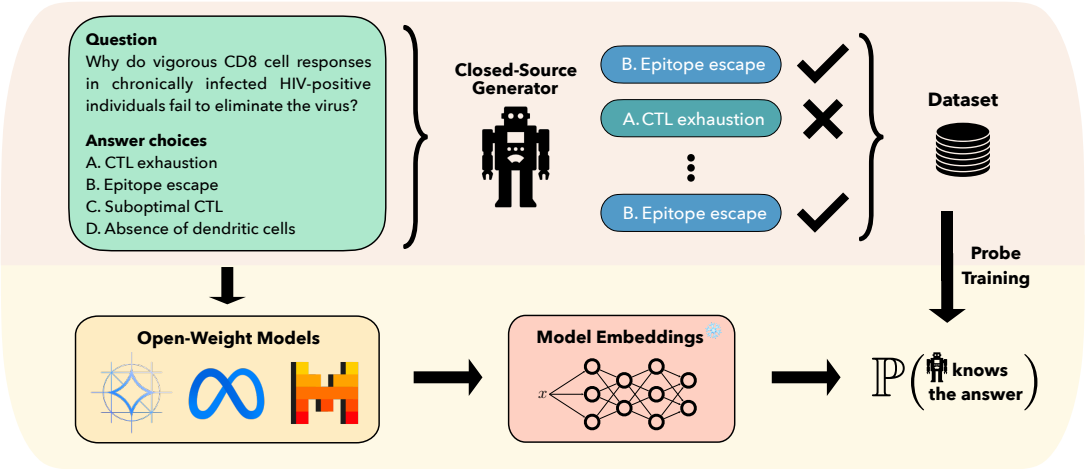

Figure 1: A schematic representation of weak-to-strong confidence prediction. First, given a dataset of questions, answers from a "strong" generator model are collected and labeled to indicate whether the generated answer is correct or incorrect. The question-answer pairs are aggregated into a weak-predictor training dataset. Using this dataset, a probe is then trained on top of a "weak," frozen open-weight or open-access model, with the goal of predicting whether the generator model is able to answer a given question correctly. After training is completed, the weak predictor model can then be used to produce confidence scores that predict whether the strong generator model will be able to answer a new, previously unseen question correctly.

correctness. The predictor can be orders of magnitude smaller than the generator, and only the simple probe is trained. This design makes it easy to analyze when such oversight works and what signals it relies on. We illustrate our settings in Figure 1. This yields an *external*, data-driven confidence signal that does not rely on the black-box model's own probabilities or hidden states and is computationally efficient and easy to deploy, making the framework broadly applicable in practice.

In our evaluation, we treat GPT-3.5-turbo and GPT-4o as black-box generators and use 15 open-access predictors: 13 open-weight LLM backbones drawn from the Llama (Meta, 2024), Mistral (Jiang et al., 2023), Gemma (Gemma Team, 2024; Gemma Team and Google DeepMind, 2025), and GPT-oss(OpenAI et al., 2025) models, plus two embedding models from OpenAI (OpenAI, 2024b) and DeepMind (Google DeepMind, 2025), with sizes ranging from 7B to 120B. We train a shallow MLP probe on the predictor's question-only embeddings to predict a binary correctness label, outputting the prediction via a softmax over logits. Notably, our setup does **not** require the *generators*' log-probabilities – the generators are used solely to produce answers. Our evaluations span 6 benchmarks, and we report both per-dataset and macro-aggregated results.

We position this setup as a pragmatic baseline for uncertainty quantification in LLMs, providing a standardized point of comparison for future work. The approach is broadly applicable and requires only question embeddings and generator responses. Although we find limited zero-shot transfer across datasets, this does not reduce practical utility, since the probe is lightweight and inexpensive to train per target distribution once generator answers can be sampled, and we envision this simple approach can be particularly useful in narrowly defined settings such as medical question answering. In addition, our analysis raises several open questions: *Can predictor predictions enhance reliability of generator outputs? Does predictor performance improve with predictor model size? Are predictors just learning the label distributions? Do different embedding aggregations impact predictor performance?* To facilitate further research, we will release all datasets.

To summarize, our key contributions are:

1. We examined the framework of weak-to-strong confidence prediction for LLMs and empirically showed that a "stronger" language model can be predicted using embeddings from "weaker" open-access models, thereby improving generator reliability in selective prediction (See Table 1).

2. We analyzed the key determinants of successful weak-to-strong confidence prediction, revealing that performance depends more on alignment between weak-model embedding and strong-model decision boundaries than on model scale alone, highlighting the importance of representational compatibility.
3. Through extensive ablations, we show the robustness of our evaluation: predictor performance remains stable across label distributions, embedding aggregation strategies, suggesting that the weak-to-strong signal is a fundamental property of model representations rather than an artifact of data or methods.
4. We release benchmark datasets to facilitate systematic study of weak-to-strong confidence prediction across future LLM generations.

2 Related Work

Weak-to-strong generalization. As large frontier models are outperforming humans on various benchmarks (Wei et al., 2022; Kaplan et al., 2020; Brown et al., 2020), we might be concerned that human intelligence is not strong enough to reliably supervise future models. But can we use a source that is weaker to supervise the training of a stronger model? This is the motivation of weak-to-strong generalization (Burns et al., 2023). Researchers have developed different frameworks to study this question. Hase et al. (2024) find that large pretrained models can generalize well from easy to hard data using in-context learning, linear probe heads, and other methods on various datasets. Liu & Alahi (2024) improves such generalization with a hierarchical mixture of experts (MoE) and Du et al. (2023) asks multiple LLM agents to debate with each other to achieve better generalization. weak-to-strong generalization has also been studied theoretically: Lang et al. (2024) shows that the student model can achieve better performance on the actual task than a weak teacher model under certain robustness conditions. Our work focuses on showing the weak-to-strong trend on model uncertainty prediction, an important step towards building more interpretable frontier models.

Uncertainty quantification in LLMs. Much research into uncertainty quantification in LLMs has had the goal of mitigating problems stemming from hallucination (Yadkori et al., 2024). Many works develop novel external metrics to compute model uncertainty or confidence in an answer generated to a given question (Kuhn et al., 2023; Duan et al., 2024). Another popular approach is to prompt LLMs to explicitly express their uncertainty (Lin et al., 2022; Tanneru et al., 2023), and other works explicitly train LLMs to identify which questions they do not know the answer to(Amayuelas et al., 2024; Yin et al., 2023). In this work, we will instead directly *learn* the probability that a model may be correct or incorrect about a specific question it is asked using an external, other model, which may be different from the question-answering one.

Selective prediction. Given the nature of our problem of interest (predicting how likely the answer of an LLM to a given question is to be incorrect) it is crucial that our evaluation models are not only accurate when their predicted probabilities are binarized into yes/no prediction labels, but also that they are well-calibrated, i.e., that their predicted probabilities match the *frequency* with which a given generator LLM is correct or incorrect in answering a question. This is crucial as this predicted probability is likely of more interest to an end-user than a simple yes/no judgment would be. This directly links our work to existing literature on model calibration. To compute model calibration, we introduce a rejection class (El-Yaniv & Wiener, 2010a): if the predictor's uncertainty is beyond a given threshold, the model abstains from making a prediction. By evaluating model outputs on a variety of thresholds, we can compute selective metrics like selective (balanced) accuracy (Fisch et al., 2024). By comparing the selective metrics with the non-selective ones, we can better understand if our model is well-calibrated (Rudner et al., 2024; Varshney et al., 2022).

3 Experimental Design

We employ a large and more capable pretrained language model, denoted as **the generator**, f_G, to generate responses to questions given in our evaluation datasets. Then we use a smaller and weaker open-access model (an open-weight LLM or an embedding model via API) with a linear probe as **the predictor**, f_P. Our goal is to train f_P to learn the uncertainty of f_G from distributions of correctness labels y of the responses and the embeddings X from the dataset $\mathcal{D}$.

Table 1: Accuracy and selective accuracy of two strong generator models (GPT-3.5-turbo and GPT-4o) across datasets. For each rejection rate, the selective generator accuracy obtained with a Llama3-8b backbone is reported. Open-access weak predictor models were used for selective prediction, showing significant improvements in performance even at a 30% rejection rate. These results provide evidence that weak predictor models can serve as effective oracles to perform scalable oversight of stronger models.

	GPT-3.5-turbo				GPT-4o			
	Accuracy	Selective Accuracy (with different rejection rates)			Accuracy	Selective Accuracy (with different rejection rates)		
Dataset		10%	30%	50%		10%	30%	50%
TriviaQA	74.33	80.63	87.35	92.26	81.47	86.73	91.51	93.77
CounterFact	94.42	96.79	98.23	99.09	95.68	97.42	98.45	98.96
Winogrande	72.70	74.97	78.31	83.16	90.00	91.05	92.66	93.57
MMLU	69.78	72.67	76.69	80.24	85.93	87.28	89.00	89.90
MedQA	59.95	61.77	66.48	69.97	88.30	89.66	90.51	91.92
SuperGPQA	32.37	33.89	36.56	40.54	44.98	46.85	51.65	57.91

3.1 Dataset Construction

For a given dataset $\mathcal{D}$, we collect n questions from $\mathcal{D}$ and query f_G. For each question x_i, we sample k (default 10) answers from f_G. We then compare each generated answer to the true answer provided in $\mathcal{D}$ and obtain $Y = \{1, 0\}$, indicating whether f_G answers correctly. By averaging the k samples for each question, we compute a probability label y, representing the likelihood that an answer from f_G is correct.

$$y_{\text{true},i} = \frac{1}{k} \sum_{j=1}^{k} \mathbb{1}\{f_G(x_i)_j = \text{ground truth}(x_i)\} \tag{1}$$

$$y_{\text{pred},i} = f_P(x_i). \tag{2}$$

This probability serves as the prediction target for the predictor, that is, $P(f_G \text{ knows the answer})$. We use a cross entropy loss function:

$$-\sum_i \Big[y_{\text{true},i} \log\big(y_{\text{pred},i}\big) + (1 - y_{\text{true},i}) \log\big(1 - y_{\text{pred},i}\big) \Big]. \tag{3}$$

We evaluate with questions from six datasets, three knowledge recall datasets: TriviaQA (Joshi et al., 2017), CounterFact (Meng et al., 2023), and Winogrande (Sakaguchi et al., 2019), and three logical reasoning datasets: MMLU (Hendrycks et al., 2021), MedQA (Jin et al., 2020), and SuperGPQA (M-A-P Team, 2025). For open-ended TriviaQA, only the original question is given to f_G, whereas for the multiple-choice datasets we provide the question along with up to four choices. We select the first 10,000 questions from chosen MMLU categories and the first 10,000 from the other datasets. More details on how our evaluation data was selected can be found in Appendix A. We provide the generator accuracy in answering each dataset in Table 1.

3.2 Evaluation Setup

To train the predictor f_P, we leverage the latent representations of the input questions produced by a "backbone" model, which may be either a generator LLM or an embedding-only model. These representations are inputs to a simple probe that predicts the correctness labels. We freeze the representations and only train the probe.

Probe head classifier setup. We implement a probe head as a two-layer neural network that takes as input the high-dimensional text representations generated by an open-access fixed "backbone" model. Following Kadavath et al. (2022), we use only the representation of the last non-padding token of each prompt, resulting in an input $X \in \mathbb{R}^{n \times d}$, where d is the output dimension of the last layer of the "backbone" model, before the final linear head. We normalize the output $y \in \mathbb{R}^2$ with a softmax function, so that y becomes two probabilities that sum to 1. We train the probe head using cross-entropy loss.

For f_G, we collected answers from GPT-3.5-turbo and GPT-4o. For f_P, we use a total of 15 open-access models: five Llama models, three Mistral models, three Gemma models, two GPT-oss models, and two embedding-only models (Text-Embedding-3-Small and Gemini-Embedding-001) via API. We use these models as the backbone to obtain embeddings. Our results show that open-access models as small as Llama3-8b can be used to predict uncertainty of much larger black-box models like GPT-4o.

Evaluation metrics. We evaluate the performance of our predictors using a range of metrics. Since y represents a probability, we discretize both y_{true} and y_{pred} with threshold $= 0.5$ to compute metrics including accuracy and F1 score. For metrics like AUROC and AUPRC, we keep y_{pred} as a probability, discretizing only y_{true} to treat it as a classification task. To address data imbalance, we subsample the data to create a class-balanced test set, which we use to compute balanced accuracy. Additionally, we compute selective metrics as an area under the prediction curve with different thresholds of binary categorical entropy. Unless stated otherwise, all metrics in the plots and tables are presented in percentages.

Datasets. We evaluate the predictive performance on across six benchmarks spanning both knowledge retrieval and logical reasoning. TriviaQA (Joshi et al., 2017) is an open-domain question-answering dataset that covers factual knowledge in various fields. CounterFact (Meng et al., 2023) aims at factual consistency, which requires models to identify the true statements over the counterfactual one. Winogrande (Sakaguchi et al., 2019) asks models to fill in pronoun blanks and requires commonsense knowledge. MMLU (Hendrycks et al., 2021) contains questions from various professional and academic domains. MedQA (Jin et al., 2020) consists of medical licensing exam questions and assesses models' medical reasoning ability. Lastly, SuperGPQA (M-A-P Team, 2025) is a challenging benchmark with graduate-level scientific questions.

3.3 Selective Prediction

We use selective metrics to further assess the models' predictive uncertainty. Selective prediction modifies the standard prediction pipeline by introducing a rejection class (El-Yaniv & Wiener, 2010b). Given a classifier f where $f(x) = \text{argmax}_{k \in \mathcal{Y}} f(x|k)$, the selective prediction model introduces a selection function s which determines whether a prediction should be made. This selection function can be based on the outputs of f, such as $s(x) = \max_{k \in \mathcal{Y}} f(x|k)$. The selective prediction model f_s is defined as

$$f(x; \tau) = \begin{cases} f(x) & \text{if } s(x) \geq \tau \\ \perp & \text{otherwise } s(x) < \tau, \end{cases} \tag{4}$$

where τ represents the rejection threshold.

Predictive uncertainty is a natural choice for a selection function. If a model's predictive uncertainty is above a certain threshold, the selection function will decline to make a prediction. In many real-world applications, especially those where the cost of an incorrect prediction is high (e.g. medical LLM making a hallucinated diagnosis), it is valuable for a model to abstain from making a prediction when it is uncertain.

Formally, given predicted probabilities $p \in [0, 1]^N$ and true labels $Y \in \{0, 1\}^N$, let π be the permutation that sorts p in an ascending order such that $p_{\pi(1)} \leq p_{\pi(2)} \leq \cdots \leq p_{\pi(N)}$. For a rejection rate τ (where $0 \leq \tau \leq 1$), we reject the lowest $\lfloor \tau N \rfloor$ predictions and define the selective generator accuracy as

$$S(\tau) = \frac{1}{N - \lfloor \tau N \rfloor} \sum_{i=\lfloor \tau N \rfloor + 1}^{N} \mathbb{1}\{Y_{\pi(i)} = 1\}, \tag{5}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. We compute $S(\tau)$ from $\tau = 0\%$ to 99%, and then use the area under $S(\tau)$ to yield the average selective generator accuracy.

4 When Can Weak Predictor Models Serve As Oracles?

In this section, we empirically analyze whether weaker predictor models are able to perform scalable oversight of stronger models by producing confidence scores that predict when the stronger generator model will give

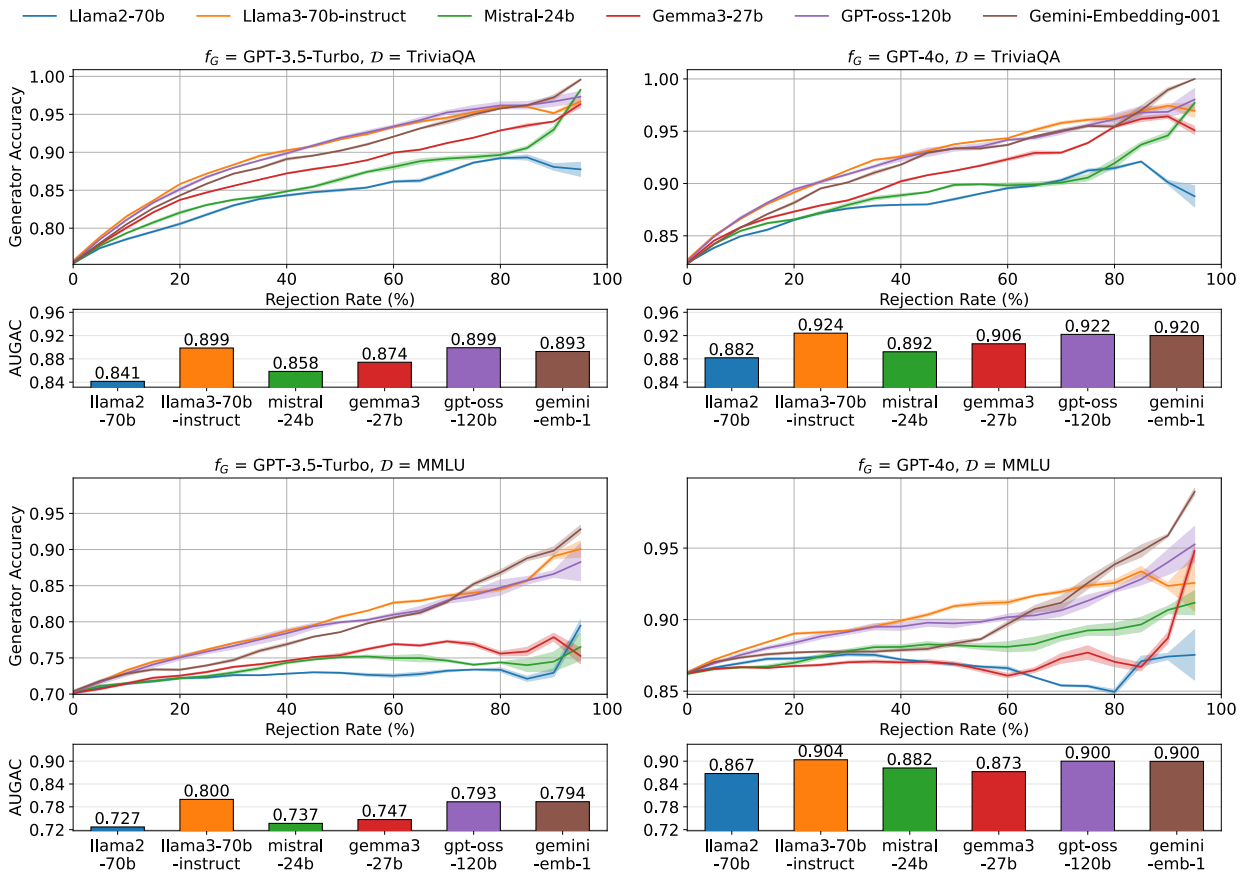

Figure 2: Upper: Selective generator accuracy on TriviaQA and MMLU for GPT-3.5-turbo and GPT-4o under different rejection thresholds. Each curve corresponds to a single representative "best" predictor, chosen as the largest or most recent model within each predictor family or generation. Generators can often achieve about 90% accuracy with a rejection rate less than 40%. Lower: Area under the selective generator accuracy curve (AUGAC) for the same set of predictors. The largest gains come from Llama3-70b-Instruct compared to Llama2-70b, while differences across the other groups of models are more modest. Generator accuracy and AUGAC are reported in proportion units. Full results for all 15 predictors are shown in Appendix B.

an incorrect answer. To do so, we consider a selective prediction setting, in which a "weak" predictor model is used to decide when not to trust a "strong" generator model. This setting is practically appealing as such weak-to-strong confidence prediction would allow users to improve the reliability of strong generator models and exercise effective scalable oversight.

First, we investigate whether effective weak-to-strong confidence prediction is possible using a simple probe as an oracle for strong generator correctness. We then examine which factors influence a predictor model's effectiveness in providing accurate confidence scores and improving generator selective prediction performance. Specifically, we investigate the role of model size (Section 4.2), the effect of label distribution (Section 4.3), and the contribution of different embedding aggregations (Section 4.4).

4.1 Does Weak-to-Strong Confidence Prediction Improve Generator Reliability?

To assess predictor performance, we use its predictions to filter out generator answers it identifies as likely incorrect. We assess performance in terms of the generator model's selective accuracy, where a certain threshold of low-confidence answers are rejected based on predictor confidence scores. We plot the selective generator accuracy curve of TriviaQA and MMLU under different thresholds in Figure 2. The curves show that selective generator accuracy increases with higher rejection rates.

To obtain a single evaluation metric that reflects the generator models' selective prediction performance using weak-predictor confidence scores, we compute the area under the selective generator accuracy curve (AUGAC). While absolute AUGAC values vary depending on each task's baseline accuracy, AUGAC is consistently higher than standard generator accuracy for all representative weak predictors shown in Figure 2. AUGAC for all the other models are reported in Tables 6 and 7, and the best improvements over the initial generator accuracy can be found in Table 1.

We present results on additional datasets and all 15 models in Appendix B. These results demonstrate substantial gains from weak predictors in selective prediction on both factual retrieval and logical reasoning tasks. The largest improvements are observed on TriviaQA, where the strongest predictors achieve AUGAC exceeding 90% with an initial generator accuracy of only 70%. In contrast, weaker backbones like Llama2-70b yield more limited gains, resulting in noticeable gaps between the selective accuracy curves across predictors.

Next, we compare our method against simple black-box baselines based on multiple sampled outputs. Self-consistency (SC) (Wang et al., 2023) is the fraction of samples assigned to the majority answer, while disagreement (DA) (Kuhn et al., 2023) is the negative entropy of the resulting empirical answer distribution. For these two baselines, we first normalize the sampled responses and group identical normalized strings as the same answer. We additionally consider confidence prompting (CP), following Kadavath et al. (2022), in which the generator performs a post-hoc true or false self-evaluation of a proposed answer and this judgment is used as a confidence score. Unlike our probes, all these baselines require k generator calls at inference time. Results for GPT-4o are shown in Table 2, with full results for both generators in Appendix H.

Table 2: AUROC comparison between weak-to-strong probe predictors and black-box multi-sample baselines, with (f_G = GPT-4o). SC, DA, and CP require repeated generator queries at inference time, whereas probe-based predictors use only question embeddings at test time.

Baseline Method / f_p	TriviaQA	CounterFact	Winogrande	MMLU	MedQA	SuperGPQA
Self-consistency (SC)	92.42	83.50	60.29	71.95	62.54	67.52
Disagreement (DA)	92.37	83.46	61.48	71.76	64.51	68.62
Confidence Prompting (CP)	56.12	68.03	64.41	57.95	55.89	55.16
Llama3-8b	80.06	81.95	59.56	62.84	62.06	59.90
Mistral-22b	67.53	72.68	56.19	57.79	58.03	57.46
Gemma3-12b	76.99	77.15	66.11	55.78	59.92	61.85
GPT-oss-20b	79.69	78.97	60.93	64.77	63.55	66.99
Gemini-Embedding-001	76.42	68.47	65.06	59.53	62.52	66.93

Overall, the trained probes outperform SC and DA on most datasets, while on MMLU and SuperGPQA their performance is comparable, with SC or DA sometimes slightly higher despite requiring repeated generator queries. CP is generally weaker than the probe baselines. TriviaQA is the main exception: SC and DA are unusually strong there because our naive string matching groups many incorrect answers into small separate clusters, which can artificially inflate both metrics. These comparisons show that the probe remains competitive with simple multi-sample black-box baselines while being substantially more efficient at test time.

Takeaway: Weak-to-strong confidence prediction improves strong generator reliability in selective prediction by providing accurate confidence scores for generated answers.

4.2 Does Weak-to-Strong Confidence Prediction Improve with Predictor Model Size?

Figure 3 shows that our predictors perform better on factual retrieval datasets than on logical reasoning datasets. We believe there are several plausible reasons for this. Question-only embeddings may capture topical content and coarse difficulty, but they do not condition on latent intermediate steps that determine correctness in multi step reasoning. In addition, there may be multiple failure modes in the latent reasoning path (Song et al., 2026; Yang et al., 2024; Li et al., 2024), increasing difficulty in separability for probes. Finally, latent reasoning chains may lead to more stochastic generations, producing noisier generator answers.

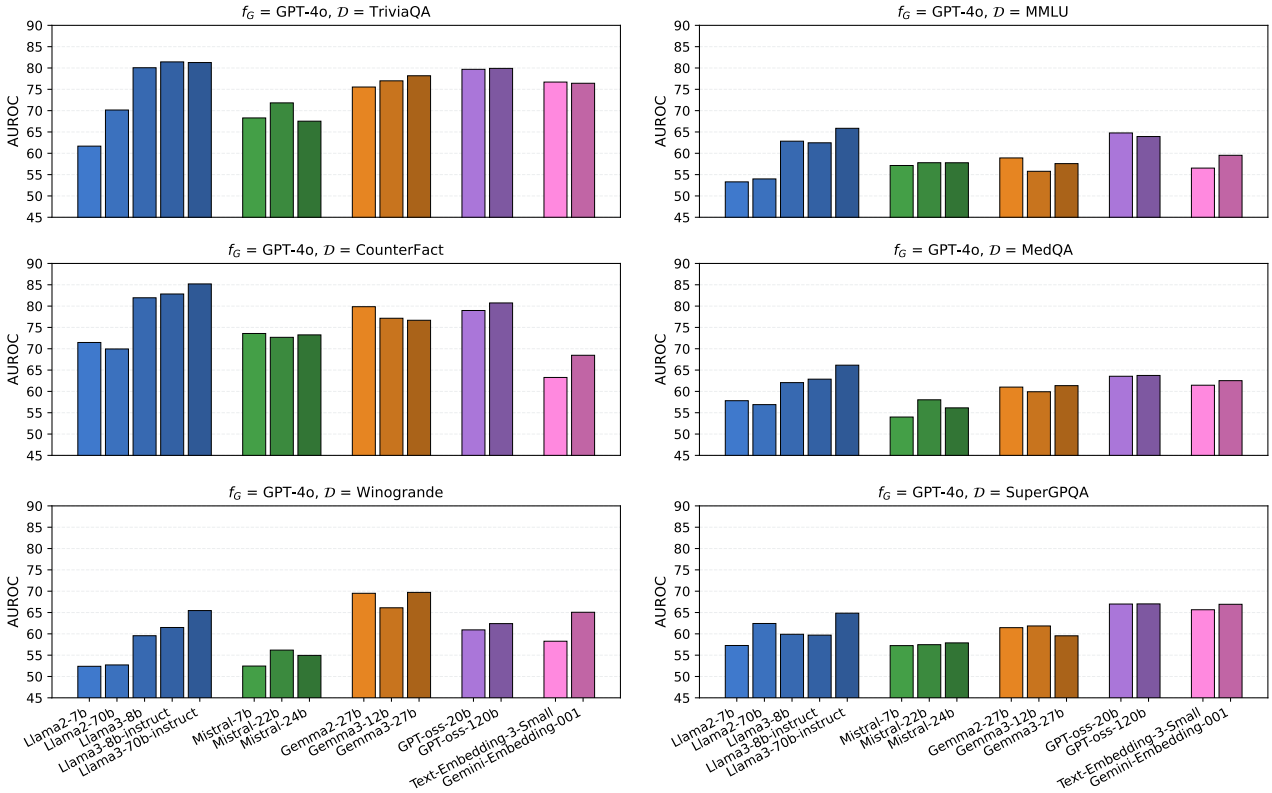

Figure 3: Evaluations across all six datasets with different predictors. For Llama models (blue), as the model size of the weak predictor model, f_P, increases, a significant improvement in AUROC can be observed. However, this is generally not true for Mistral and Gemma models. It is also hard to find a consistent scaling trend between different model groups within each task. Similar results can be observed on other metrics like balanced accuracy. For more visualized results, please refer to Figures 13 to 18 in Appendix C.

We examined the stochasticity hypothesis and quantified the effect using empirical answer disagreement entropy. The results, included in Appendix D, demonstrate that generator answers on reasoning datasets are more stochastic and less identifiable from question-only features.

A central question in our analysis is whether increasing the size of the predictor backbone improves performance. Within the Llama family, more capable backbones result in higher AUROC. Notably, Llama 3 embeddings provide a clear boost over Llama 2, even though Llama 3's generation accuracy improves only modestly (Meta, 2024). Other metrics, such as class-balanced accuracy, exhibit similar scaling trends as the predictor backbone becomes larger and more capable (see Figure 13 to Figure 15 for more details.)

To test whether this scaling effect appears specific to Llama or universal, we experiment with the Mistral, Gemma, and GPT-oss families, and additionally the two embedding-only backbones. We select Mistral and Gemma models with ascending model size and different generations, where their generation performance also improves with size (Gemma Team and Google DeepMind, 2025), but we do not observe the same consistent improvements (see Figures 3 and 13 to Figure 18). In fact, scaling trends vary across tasks; for example, the larger and more recent Gemma sometimes underperforms its smaller counterparts, while in other cases it even surpasses Llama3-70b.

This raises a broader concern: while the LLMs are trained to be increasingly capable at answering the questions, the representations they learn to *map questions to answers* are not necessarily aligned with the task of *mapping questions to probabilities of correctness*. This may explain why the scaling behavior observed in generation ability does not consistently transfer to probes built on model embeddings. To induce a predictor

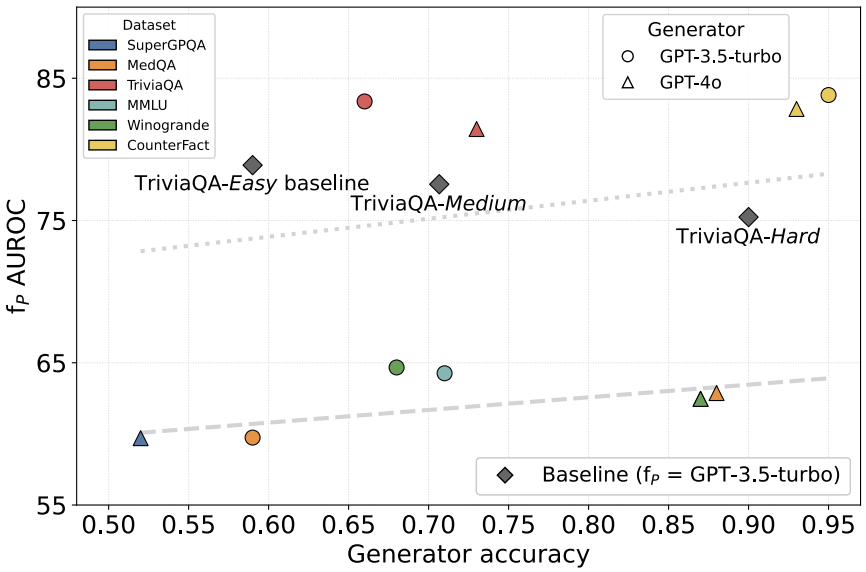

Figure 4: Predictor AUROC (%) using Llama-3-8B-Instruct backbone on six datasets. Generator accuracy is displayed in proportion units. Circle markers denote GPT-3.5-turbo, and triangles denote GPT-4o. Each marker is compared to the nearest TriviaQA-*Easy*/*Medium*/*Hard* baseline in terms of generator accuracy. Dotted and dashed lines fit knowledge recall and reasoning datasets. No statistically significant trend is observed. Complete baseline results are provided in Appendix F.

scaling behavior that mirrors LLM scaling, one would likely need to train probes on richer, model-internal signals that underlie an LLM's own reasoning, rather than relying solely on binary correctness labels.

Moreover, even within the Llama family, the benefits of scaling are modest. *Upgrading from sub-10B to 70B-parameter Llama backbones yields only incremental predictor improvements.* This suggests that, for the evaluation task that is central to our paper, smaller models may be nearly as effective in the setting. We hypothesize that less capable Llama backbones produce representation distributions well aligned with GPT outputs, leaving limited room for further gains.

Finally, the training method of the backbone LLM contributes only marginally to predictor improvements. For instance, Llama3-8b-instruct, which is finetuned via instruction tuning, brings more than 3% improvement than Llama3-8b in AUROC and balanced accuracy on the Winogrande dataset, but is only slightly better on the TriviaQA dataset, where AUROC already exceeds 80%. Similar patterns hold across other metrics, as shown in Table 8 - Table 13, underscoring that predictor performance is not strongly determined by the same factors that drive generation performance.

Takeaway: Unlike in generation tasks, model size alone is not a reliable determinant of predictor performance across architectures.

4.3 Are Weak-to-Strong Confidence Predictors Just Learning Label Distributions?

We next test whether predictors truly learn generator- and question-specific features, or whether they simply learn to mimic the overall distribution of correctness labels. To disentangle these effects, we designed two ablation experiments described below.

Controlling "Difficulty" Levels. Imbalanced datasets pose a greater challenge for predictors. Predictors may struggle because of the dataset intrinsic difficulty or the skewed label distribution, which often arises because generation quality is influenced by that same intrinsic difficulty.

To isolate the effect of dataset intrinsic difficulty, we collected additional GPT-3.5-turbo responses to construct three TriviaQA subsets with controlled generator accuracies: 60% (*Easy*), 75% (*Medium*), and 90% (*Hard*).

By fixing the proportion of correct answers, we disentangle label imbalance effects from dataset-specific features and can better assess their impact on training.

Results are shown in Figure 4. The three gray diamonds mark predictor performance on TriviaQA-*Easy*, *Medium*, and *Hard*. Notably, the *Hard* baseline – where incorrect responses are scarce – remains the most challenging. This setting enables comparisons across datasets with matched "difficulty" level, as each marker can be compared to the nearest TriviaQA-*Easy*/*Medium*/*Hard* baseline in terms of generator accuracy. For instance, MedQA with GPT-3.5-turbo achieves 58.91% generator accuracy, close to TriviaQA-*Easy*, yet predictors perform worse on MedQA. This suggests that embeddings from TriviaQA questions are more predictive of correctness than those from MedQA. A similar pattern appears in the *Hard* cluster: datasets such as MMLU, MedQA, and Winogrande (all near 90% generator accuracy) yield predictor AUROCs of only around 65%, far below CounterFact or TriviaQA-*Hard*.

Controlling Exact Label Ratios. We also tested three variants of TriviaQA and MMLU: a balanced 50/50 subset (*50/50*), one matching the training label ratio (*Exact ratio*), and the full test set (*Original*). The original training set had 8,000 samples, and the test set 2,000. This removes natural imbalance while preserving dataset content.

This experiment probes a different aspect than the difficulty study. While the earlier experiment varied dataset difficulty across tasks, here we fix the dataset (TriviaQA or MMLU) and vary only the input-output mapping via generator labels (GPT-3.5-turbo vs GPT-4o). Thus, inputs are drawn from the same distribution, and differences reflect whether the predictor learns meaningful associations between embeddings and correctness.

Results (Tables 21 and 23) show that predictors behave differently for GPT-4o versus GPT-3.5-turbo labels, confirming that they exploit input features rather than only memorizing label priors. Moreover, performance on the *Original* and *Exact ratio* test sets is comparable, with the *Original* sometimes slightly worse. This indicates that predictors trained on balanced data still generalize well to natural, imbalanced distributions. We refer readers to Table 21–Table 27 for full results and label distribution details.

Takeaway: Predictors learn generator- and question-specific features rather than simply imitating the label distributions.

4.4 Does Weak-to-Strong Confidence Prediction Depend on Embedding Choice?

We have been using the embedding of the last non-padding token of each question as inputs for the probe layers. While this is common practice in autoregressive models (Kadavath et al., 2022), when the context becomes long it may not always capture the most relevant information. Ahdritz et al. (2024) studies how embeddings from different layers may impact the performance of downstream layers on different tasks. Following this example, we used Llama models to experiment and compare middle layer embeddings with last layer embeddings as inputs to our predictors.

We also experimented with different pooling strategies to aggregate the embeddings, including average pooling and max pooling. We find that only average pooling and intermediate embeddings shows a minor improvement, and results are given in Appendix G. Overall, the choice of aggregation strategy makes only a small difference. Crucially, the same scaling trends observed earlier persist across all aggregation methods, suggesting that backbone model quality remains a more important factor.

Takeaway: Different embedding aggregations produce broadly consistent results, suggesting that predictor performance is robust to the choice of aggregation method.

4.5 Does Predictor Performance Depend on Sampling and Decoding Choices?

Because y_{true} is constructed from multiple sampled generator outputs, its value can vary with the decoding strategy and the sampling budget. To address this concern, we conduct a comprehensive sensitivity analysis over the sampling temperature $\in \{0.2, 0.8, 1.0, 1.2\}$, the nucleus sampling threshold (Holtzman et al., 2020) top_$p \in \{0.9, 1.0\}$, and the sampling budget $k \in \{5, 10, 20\}$. Across all datasets, generators and predictors,

the main qualitative findings remain consistent, indicating that our conclusions are broadly stable and do not rely on specific sampling hyperparameter choices. The complete results are provided in Appendix I.

Overall, we find that smaller predictor backbones such as Llama3-8B still provide effective weak-to-strong confidence signals, and that knowledge retrieval tasks are consistently easier to predict than reasoning benchmarks. Reducing the sampling budget from 10 to 5 typically leads to a small decrease in AUROC, while increasing it from 10 to 20 often produces modest improvements. However, neither change affects the overall benefits of selective prediction or alters the main conclusions regarding representational compatibility.

Takeaway: Predictor performance is driven primarily by the quality of the underlying representations, rather than by any particular choice of decoding hyperparameters or sampling budget.

5 Discussion and Limitations

First, because the predictor only sees the question, it often underperforms on reasoning tasks where crucial evidence arises in intermediate solution steps. Second, while we observe a slight scaling trend with Llama models, this trend does not generalize across broader model families. Finally, our analysis showed that there was no single, clear determinant of weak-to-strong confidence prediction capabilities: variations in model size, label balance, embedding aggregation, and hyperparameter selection each only have minor effects on predictor performance.

The main weakness of our analysis is its conservatism: By choosing frozen open-access models and only learning a probe classifier, we considered a highly restrictive setting. While we were able to demonstrate meaningful weak-to-strong predictive capabilities, it is likely that learning tailored weak-to-strong confidence prediction models will result in quantitatively and qualitatively distinct findings.

These considerations suggest several future research directions. On the architecture side, the choice of probe can be refined. Replacing the MLP with more complicated structure has shown promise in recent works (CH-Wang et al., 2024; Psomas et al., 2025), but with the increased parameter count, it is unclear whether our dataset is large enough to fit such models. Additionally, *answer-aware* weak-to-strong confidence prediction in transfer learning across different tasks (e.g. TriviaQA to CounterFact) and categories (TriviaQA to SuperGPQA) has not been explored. Because a probe is easy to train, demonstrating such generalization to unseen benchmarks would be practically valuable (see Appendix J for out-of-distribution results). We will release our datasets to support these investigations and to provide a clean baseline for research into weak-to-strong confidence prediction.

6 Conclusion

In this paper, we studied weak-to-strong confidence prediction—the task of using a weak predictor model to produce confidence scores that can be used to predict whether a stronger generator model will answer a given question correctly. In our empirical evaluation, we found that weak-to-strong confidence prediction can significantly improve strong generator reliability in selective prediction settings, suggesting that it is a useful tool for exercising effective scalable oversight. We hope this work will encourage future research into weak-to-strong confidence prediction as a step towards building safer and more reliable frontier models.

References

Gustaf Ahdritz, Tian Qin, Nikhil Vyas, Boaz Barak, and Benjamin L. Edelman. Distinguishing the knowable from the unknowable with language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 503–549. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ahdritz24a.html.

Alfonso Amayuelas, Kyle Wong, Liangming Pan, Wenhu Chen, and William Wang. Knowledge of knowledge: Exploring known-unknowns uncertainty with large language models, 2024. URL https://arxiv.org/abs/2305.13712.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision, 2023. URL https://arxiv.org/abs/2312.09390.

Sky CH-Wang, Benjamin Van Durme, Jason Eisner, and Chris Kedzie. Do androids know they're only dreaming of electric sheep?, 2024. URL https://arxiv.org/abs/2312.17249.

Florian E. Dorner, Vivian Y. Nastl, and Moritz Hardt. Limits to scalable evaluation at the frontier: Llm as judge won't beat twice the data. *arXiv preprint arXiv:2410.13341*, 2025. doi: 10.48550/arXiv.2410.13341. URL https://arxiv.org/abs/2410.13341.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023. URL https://arxiv.org/abs/2305.14325.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models, 2024. URL https://arxiv.org/abs/2307.01379.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010a. URL http://jmlr.org/papers/v11/el-yaniv10a.html.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *J. Mach. Learn. Res.*, 11:1605–1641, August 2010b. ISSN 1532-4435.

Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Calibrated selective classification, 2024. URL https://arxiv.org/abs/2208.12084.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Gemma Team and Google DeepMind. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Google DeepMind. Gemini embedding now generally available in the gemini api and vertex ai. https://developers.googleblog.com/en/gemini-embedding-available-gemini-api/, 2025. Announces the gemini-embedding-001 model.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024. URL https://arxiv.org/abs/2411.15594.

Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegreffe. The unreasonable effectiveness of easy training data for hard tasks, 2024. URL https://arxiv.org/abs/2401.06751.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. URL https://arxiv.org/abs/1904.09751.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams, 2020. URL https://arxiv.org/abs/2009.13081.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-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, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

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

Hunter Lang, David Sontag, and Aravindan Vijayaraghavan. Theoretical analysis of weak-to-strong generalization, 2024. URL https://arxiv.org/abs/2405.16043.

Xiaoyuan Li, Wenjie Wang, Moxin Li, Junrong Guo, Yang Zhang, and Fuli Feng. Evaluating mathematical reasoning of large language models: A focus on error identification and correction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11316–11360, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words, 2022. URL https://arxiv.org/abs/2205.14334.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.153. EMNLP 2023.

Yuejiang Liu and Alexandre Alahi. Co-supervised learning: Improving weak-to-strong generalization with hierarchical mixture of experts, 2024. URL https://arxiv.org/abs/2402.15505.

M-A-P Team. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt, 2023. URL https://arxiv.org/abs/2202.05262.

Meta. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

OpenAI. Gpt-4 technical report, 2024a. URL https://arxiv.org/abs/2303.08774.

OpenAI. text-embedding-3-small model. https://platform.openai.com/docs/models/text-embedding-3-small, 2024b. OpenAI text embedding model.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Bill Psomas, Dionysis Christopoulos, Eirini Baltzi, Ioannis Kakogeorgiou, Tilemachos Aravanis, Nikos Komodakis, Konstantinos Karantzalos, Yannis Avrithis, and Giorgos Tolias. Attention, please! revisiting attentive probing for masked image modeling, 2025. URL https://arxiv.org/abs/2506.10178.

Tim G. J. Rudner, Xiang Pan, Yucen Lily Li, Ravid Shwartz-Ziv, and Andrew Gordon Wilson. Fine-tuning with uncertainty-aware priors makes vision and language foundation models more reliable. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024. URL https://openreview.net/forum?id=37fM2QEBSE.

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

Peiyang Song, Pengrui Han, and Noah Goodman. Large language model reasoning failures, 2026.

Sree Harsha Tanneru, Chirag Agarwal, and Himabindu Lakkaraju. Quantifying uncertainty in natural language explanations of large language models, 2023. URL https://arxiv.org/abs/2311.03533.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. Investigating selective prediction approaches across several tasks in IID, OOD, and adversarial settings. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1995–2002, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.158. URL https://aclanthology.org/2022.findings-acl.158.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022. URL https://arxiv.org/abs/2206.07682.

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. To believe or not to believe your llm, 2024. URL https://arxiv.org/abs/2406.02543.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10210–10229, 2024.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. Do large language models know what they don't know?, 2023. URL https://arxiv.org/abs/2305.18153.

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. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023. URL https://arxiv.org/abs/2306.05685.