Enhancing Zero-shot Chain of Thought Prompting via Uncertainty-Guided Strategy Selection

Shanu Kumar Saish Mendke Karody Lubna Abdul Rahman
Santosh Kurasa Parag Agrawal Sandipan Dandapat
Microsoft Corporation, India
{shankum,saishmendke,lubnakarody,skurasa,paragag,sadandap}@microsoft.com

Abstract

Chain-of-thought (CoT) prompting has significantly enhanced the capability of large language models (LLMs) by structuring their reasoning processes. However, existing methods face critical limitations: handcrafted demonstrations require extensive human expertise, while trigger phrases are prone to inaccuracies. In this paper, we propose the Zero-shot Uncertainty-based Selection (*ZEUS*) method, a novel approach that improves CoT prompting by utilizing uncertainty estimates to select effective demonstrations without needing access to model parameters. Unlike traditional methods, *ZEUS* offers high sensitivity in distinguishing between helpful and ineffective questions, ensuring more precise and reliable selection. Our extensive evaluation shows that *ZEUS* consistently outperforms existing CoT strategies across four challenging reasoning benchmarks, demonstrating its robustness and scalability.

1 Introduction

Large Language Models (LLMs) have achieved remarkable performance in a wide range of natural language processing tasks (Brown et al., 2020; Touvron et al., 2023; Thoppilan et al., 2022). However, they often struggle with tasks that require complex reasoning (Rae et al., 2021; Liang et al., 2022). The "chain-of-thought" (CoT) prompting technique (Wei et al., 2022; Feng et al., 2024) has been proposed to address this limitation by generating intermediate rationales (r) along with the final answer (a) for a given question (q). In this context, few-shot in-context examples, referred to as demonstrations $D = (q_j, r_j, a_j)_{j=1}^{k}$, consist of k example questions q_j, manually crafted rationales r_j, and answers a_j. This approach, known as *Manual-CoT*, relies on handcrafted rationales to guide the model.

Building on *Manual-CoT*, *Zero-Shot-CoT* (Kojima et al., 2022) presents a novel prompting method where LLMs generate rationales using a trigger phrase t (e.g., *"Let's think step by step"*) appended to the input question q, without requiring manually crafted demonstrations. While Zero-Shot-CoT is cost-effective, its performance often falls short compared to *Manual-CoT* due to the absence of effective demonstrations.

Crafting rationales manually is typically labor-intensive and time-consuming, particularly for tasks demanding intricate reasoning. To mitigate this, Auto-CoT (Zhang et al., 2022) combines Manual-CoT and Zero-Shot-CoT, thereby reducing the performance gap while minimizing manual effort. Auto-CoT employs self-supervised learning on a set of unlabeled questions $Q = \{q_j\}_{j=1}^{m}$ to generate rationales and answers. Demonstrations are created by clustering Q into k groups and selecting a representative question, rationale, and answer from each cluster. This clustering approach aims to maintain diversity in the demonstrations, which can help mitigate the impact of any errors in the generated rationales.

In this work, we seek to enhance the creation of demonstrations that improve LLM performance solely using unlabeled questions Q without any rationale and answer. The selection process of examples q_j in demonstrations D has been to significantly influence LLM performance (Wan et al., 2023), and generating consistent rationales (Wang et al., 2022) is crucial. Recent CoT prompting methods (Diao et al., 2024; Bayer and Reuter, 2024) have utilized Active Learning (AL) (Fu et al., 2013; Settles and Craven, 2008; Rotman and Reichart, 2022; Kumar et al., 2022) to identify examples for human annotation, showing that annotating the most uncertain examples yields the best performance. Drawing on these principles, we propose several selection strategies based on the uncertainty of unlabeled questions.

To estimate uncertainty, we adopt perturbation-based methods (Ribeiro et al., 2020; Kuhn et al., 2023; Gao et al., 2024; Tomani et al., 2024), which

Proceedings of the 31st International Conference on Computational Linguistics, pages 2003–2025
January 19–24, 2025. ©2025 Association for Computational Linguistics

operate on the principle that incorrect predictions can be detected by resampling rationales through perturbations, such as temperature adjustments. If the LLM is confident in its prediction, perturbations are unlikely to affect the outcome. However, if the LLM's prediction is uncertain, different perturbations can lead to varied responses. Our initial experiments reveal that while temperature-based perturbation estimates are well-calibrated, they lack sufficient sensitivity.[1] To address this, we propose a robust method for estimating uncertainty that exhibits near-ideal linearity with accuracy.

Our primary contributions are threefold: i) We present *ZEUS*,[2] a method for estimating LLM uncertainty that is both well-calibrated and sensitive. ii) We leverage these uncertainty estimates to guide the selection of most informative demonstrations and show that these strategies outperform existing prompting methods across four challenging reasoning tasks. iii) We demonstrate that the performance of *ZEUS* correlates strongly with few-shot uncertainty estimates on the unlabeled set, providing actionable recommendations for creating effective demonstrations.

2 Related Work

Chain-of-Thought (CoT) prompting has significantly influenced various advanced techniques designed to enhance reasoning capabilities. These include Tree of Thoughts (Yao et al., 2023), Role Play (Kong et al., 2024), and Collaborative Prompting (Zhu et al., 2023; Yin et al., 2023; Liang et al., 2023; Wang et al., 2023), each building on the CoT methodology to improve model performance in complex reasoning tasks. Concurrently, Active Learning (AL)-based methods have gained traction in few-shot prompting scenarios. Diao et al. (2023) enhance CoT prompting within an AL framework by actively selecting questions based on an uncertainty metric and manually constructing demonstrations. Shum et al. (2023) work with labeled questions devoid of rationales, generating rationales through pruning and using an AL-inspired variance-reduced policy gradient strategy to select the most informative examples. Similarly, Bayer and Reuter (2024) apply uncertainty-based AL methods to identify the most valuable questions for annotation. Unlike these studies, our work addresses a more challenging scenario where neither human-annotated labels nor rationales are available.

Our method relies on accurate uncertainty estimation due to the lack of human supervision. Estimating uncertainty is a well-explored challenge, with methods ranging from Bayesian approaches and ensemble methods to more recent perturbation-based techniques (Hendrycks and Gimpel, 2016; Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Guo et al., 2017; Van Amersfoort et al., 2020; Ovadia et al., 2019). Perturbation-based methods, which include techniques like temperature adjustments and question rephrasing, have shown promise in recent studies (Gao et al., 2024). These methods, while effective, may not be universally applicable to LLMs due to their generative nature (Vashurin et al., 2024). Other recent work has explored uncertainty estimation for specific tasks, such as hallucination detection in LLMs (Kuhn et al., 2023; Tomani et al., 2024). We extend these perturbation techniques by enhancing their sensitivity to capture finer distinctions between questions, thereby improving uncertainty estimation in LLMs.

3 *ZEUS*: Zero-shot Uncertainty-based Selection

We propose the *ZEUS* method, which aims to construct useful demonstrations containing a specific level of required uncertainty. It is comprised of three stages: (i) uncertainty estimation, (ii) uncertainty-based question selection, and (iii) demonstration construction. We have illustrated all the stages of ZEUS in Figure 1.

3.1 Uncertainty Estimation (Stage 1)

In the *ZEUS* method, uncertainty estimation is a critical step and is performed using perturbation. We exploit three distinct types of perturbations to estimate uncertainty for each unlabeled question in the set Q. These perturbations include temperature adjustments, trigger phrase variations, and question rephrasing.

Temperature Perturbation: This perturbation technique is based on the principle that a question can be answered in multiple ways, and these variations can be explored by adjusting the temperature parameter during decoding. Temperature perturbation helps in simulating different reasoning paths within the LLM. When the temperature is set to a higher value, the model's outputs become more diverse, while a lower temperature typically

[1] Sensitivity is a measure of the degree of change in accuracy by a unit change in the confidence score.

[2] We have uploaded code and datasets for reproducibility.

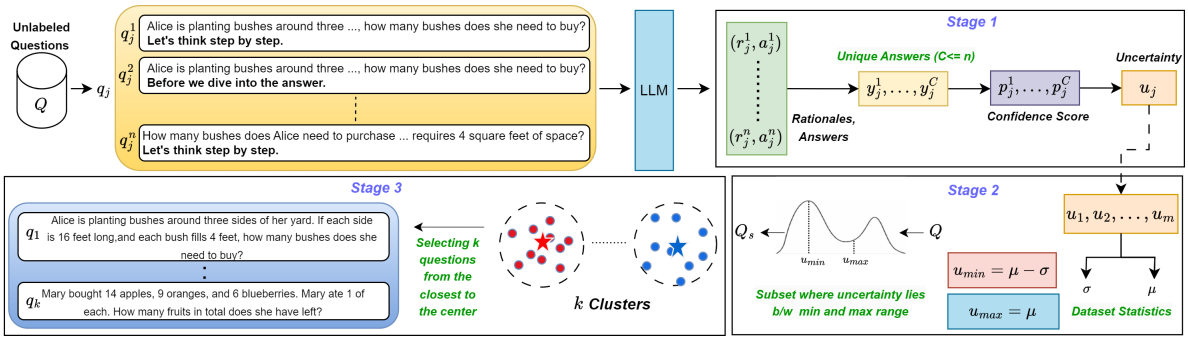


Figure 1: Overview of *ZEUS*: Uncertainty for a question q_j is calculated using a pool of answers generated using various prompts, including trigger phrases, non-zero temperature-based decoding, and rephrasing of q_j. Subsequently, questions with uncertainty within a certain range are selected and used for constructing demonstrations.

results in more confident and consistent responses (Koehn, 2009). According to Wang et al. (2022), if an LLM is confident in its answer to a question q_j, the responses generated at various temperatures should reach the same answer. In contrast, if the LLM is uncertain, different temperatures will yield a range of potentially inconsistent answers. To estimate uncertainty using this property, we generate n responses for a question q_j by using the highest temperature (=1). These responses $\{r_j^l\}_{l=1}^n$ form the basis for our temperature-based uncertainty estimation.

Trigger Phrase Perturbation: This factor leverages the sensitivity of LLM performance to trigger phrases. Kojima et al. (2022) demonstrated that appending different trigger phrases to a question can affect the LLM's output. By introducing variations in trigger phrases, we can assess whether the LLM's responses remain consistent. If the LLM provides the same answer across different trigger phrases, it suggests a high level of confidence in its response. Conversely, varying answers across trigger phrases indicate that the question q_j is challenging or that the LLM is uncertain. To apply this perturbation, we append a set of t different trigger phrases to the original question q_j and generate a corresponding set of responses $\{r_j^l\}_{l=1}^t$.

Rephrasing Perturbation: The third technique utilizes rephrasing of the input question to explore the impact on the LLM's responses. We hypothesize that if the LLM is confident about its answer, rephrasing the question should not significantly alter the generated answer. On the other hand, if the LLM's answer is influenced by specific biases or ambiguities in the original question, rephrasing may lead to a different response. To estimate uncertainty using rephrasing, we generate v rephrased versions of the question q_j and obtain the sets of responses $\{r_j^l\}_{l=1}^v$.

By integrating these three types of perturbations—temperature adjustment, trigger phrase variation, and question rephrasing, we generate a diverse set of responses for each question q_j. Specifically, we produce a total of $n \times t \times v$ responses. This pool of answers reflects variations due to different decoding settings, trigger phrases, and question rephrasing, serving as Monte Carlo samples from the LLM's likelihood distribution. From these responses, we identify C $(\leq n)$ unique answers $y_j^1, \ldots, y_j^C$ for the question q_j. The confidence score $p(y_j^c|q_j)$ for each unique answer y_j^c is computed based on the consistency of responses across the different perturbations. This score quantifies the degree of certainty associated with each answer and serves as a basis for selecting informative demonstrations in subsequent stages of the *ZEUS* method. The confidence score $p(y_j^c|q_j)$ for a unique answer y_j^c is defined as:

$$p(y_j^c|q_j) = \frac{1}{n}\sum_{l=1}^{n} 1(y_j^c = a_j^l) \qquad (1)$$

where $1(\cdot)$ is the indicator function that evaluates to 1 if y_j^c matches a_j^l and 0 otherwise.

To represent the uncertainty of the LLM regarding the question q_j, we use predictive entropy (*PE*) (Kumar et al., 2022). *PE* is maximized when confidence scores are uniformly distributed across many unique answers and increases as the number of unique answers grows. It reaches zero when all answers are identical. The *PE* for the question q_j is computed as follows:

Strategy	$u_{\min}$	$u_{\max}$
Trivial	0	μ - σ
Very Easy	0	μ
Easy	0	$\mu + \sigma$
Moderate	μ - σ	μ
Challenging	μ - σ	$\mu + \sigma$
Hard	μ - σ	∞
Very Hard	μ	∞

Table 1: Selection Strategies used in *ZEUS* with their minimum $\mu_{\min}$ and maximum $\mu_{\max}$ range.

$$u_j = -\sum_{c=1}^{C} p(y_j^c|q_j) \cdot \log(p(y_j^c|q_j)) \quad (2)$$

where u_j measures the degree of uncertainty by quantifying the diversity of the answers.

3.2 Uncertainty-based Selection (Stage 2)

We define the LLM's overall understanding of the task using the mean uncertainty μ and the standard deviation σ of the uncertainty estimates from the unlabeled set Q. A higher mean μ indicates a more challenging task for the LLM, while a higher standard deviation σ reflects greater variability in question difficulty within Q. These two parameters provide insight into the usefulness of a question for improving the LLM's performance.

For instance, we hypothesize that when the mean uncertainty μ is low (indicating the LLM is performing well on the task), selecting questions with uncertainties lower than μ would not contribute valuable information. On the other hand, when the mean μ is high (suggesting the LLM struggles with the task), selecting questions with uncertainties significantly higher than μ may lead to less informative or erroneous rationales.

Based on these assumptions, we propose selecting a subset of questions Q_s that fall within a specific uncertainty range, as defined by the following condition:

$$Q_s \subset Q = \{q_j \mid u_{\min} \leq u_j < u_{\max}\} \quad (3)$$

Here, $u_{\min}$ and $u_{\max}$ represent the minimum and maximum uncertainty thresholds used to select questions. In the subsequent section, we will detail the specific ranges (cf. Table 1) based on μ and σ for constructing demonstrations.

3.3 Demonstration Construction (Stage 3)

We adopt the demonstration construction methodology from Auto-CoT, which emphasizes diversity to mitigate the influence of incorrect rationales generated by the Zero-Shot-CoT method. The selected questions Q_s are first encoded into vector representations using Sentence Transformers (Reimers and Gurevych, 2019). These vectors are then clustered using k-Means++ (Arthur and Vassilvitskii, 2007), forming k distinct clusters. From each cluster, the question closest to the cluster centroid is selected. The associated rationale and answer, generated by the Zero-Shot-CoT method, are then combined to form the demonstration set D. During inference, a test question q is appended to the constructed demonstration D and passed to the LLM for final predictions.

4 Experimental Setup

Datasets: We evaluate our proposed method on four challenging reasoning datasets. **GSM8K** (Cobbe et al., 2021) comprises arithmetic reasoning problems. **StrategyQA** (Geva et al., 2021) is a question-answering benchmark requiring implicit multi-hop reasoning. **Logical Fallacy** (referred to as *Fallacy*) (Jin et al., 2022) involves reasoning about arguments and detecting formal and informal fallacies. **Epistemic Reasoning** (*EPR*) (Sileo and Lernould, 2023) is a natural language inference task that challenges LLMs to reason about human mental states. For a fair comparison, we split all datasets, except GSM8K, into two sets using stratified sampling: (i) an unlabeled set (70%) for demonstration creation, and (ii) a test set (30%) for zero-shot performance evaluation. GSM8K already contains train and test sets, so no further split was needed.

Implementation: We conduct experiments using five LLMs: GPT-4o (OpenAI, 2024), *Mistral-7B-Instruct-v0.2* (Mistral) (Jiang et al., 2023), *Phi-3-mini-4k-instruct* (Phi3) (Abdin et al., 2024), *text-davinci-002* (GPT3-XL), and *text-davinci-003* (GPT3.5) (Brown et al., 2020). Note that this models are including both open-source (Phi3, Mistral) and proprietary models (GPT-4o, GPT3.5, GPT3-XL). To ensure consistency with prior work such as Auto-CoT, we use $k = 8$ demonstrations for all datasets, except for StrategyQA, where we use $k = 6$. Additionally, during the evaluation of the LLMs, we set the temperature to 0 to ensure deterministic outputs, and report the average perfor-

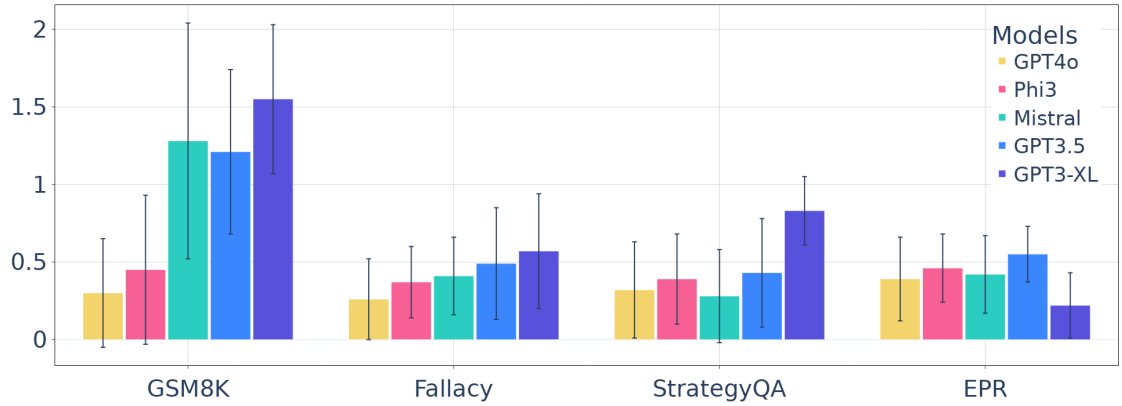


Figure 2: Mean and standard deviation of uncertainty values as error graph -specific statistics across models.

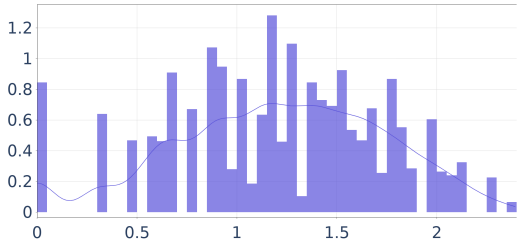


Figure 3: Probability density function of uncertainty estimates of our method using GPT3.5 on GSM8K.

mance across three runs to maintain consistency in predictions.

Uncertainty Estimation in *ZEUS*: Uncertainty in *ZEUS* is estimated using a combination of three perturbation methods: (1) non-zero temperature decoding, (2) trigger phrase variation, and (3) question rephrasing. We use five trigger phrases: " " (Empty), "*Let's think step by step.*" (SS), "*Let's think about this logically step by step.*" (LSS), "*Before we dive into the answer,*" (BDA), and "*Before answering the question, let's understand the input.*" (BQU). For each question, we generate two rationale-answer pairs per trigger phrase at a temperature of 1, producing 10 rationale-answer pairs.

Each question is also rephrased using the instruction "*Rephrase the below passage*" with GPT4o.[3] We then generate five additional rationale-answer pairs using these rephrased questions with trigger phrases at a temperature of 0 to ensure precise responses. Thus, a total of 15 rationale-answer pairs are generated for each question to estimate uncertainty.

Selection Strategies in *ZEUS*: We define seven selection strategies based on the mean μ and standard deviation σ of uncertainty values across the unlabeled set, detailed in Table 1. These strategies include: ***Trivial***, ***Very Easy***, and ***Easy*** (selecting the lowest uncertainty demonstrations), ***Challenging***, ***Hard***, and ***Very Hard*** (focusing on high uncertainty values), and ***Moderate*** (selecting demonstrations from a range of uncertainty levels around μ).

Baselines: We compare *ZEUS* against five baseline methods: Zero-Shot, Few-Shot,[4] Zero-Shot-CoT (Kojima et al., 2022), Manual-CoT (Few-Shot-CoT) (Wei et al., 2022), and Auto-CoT.

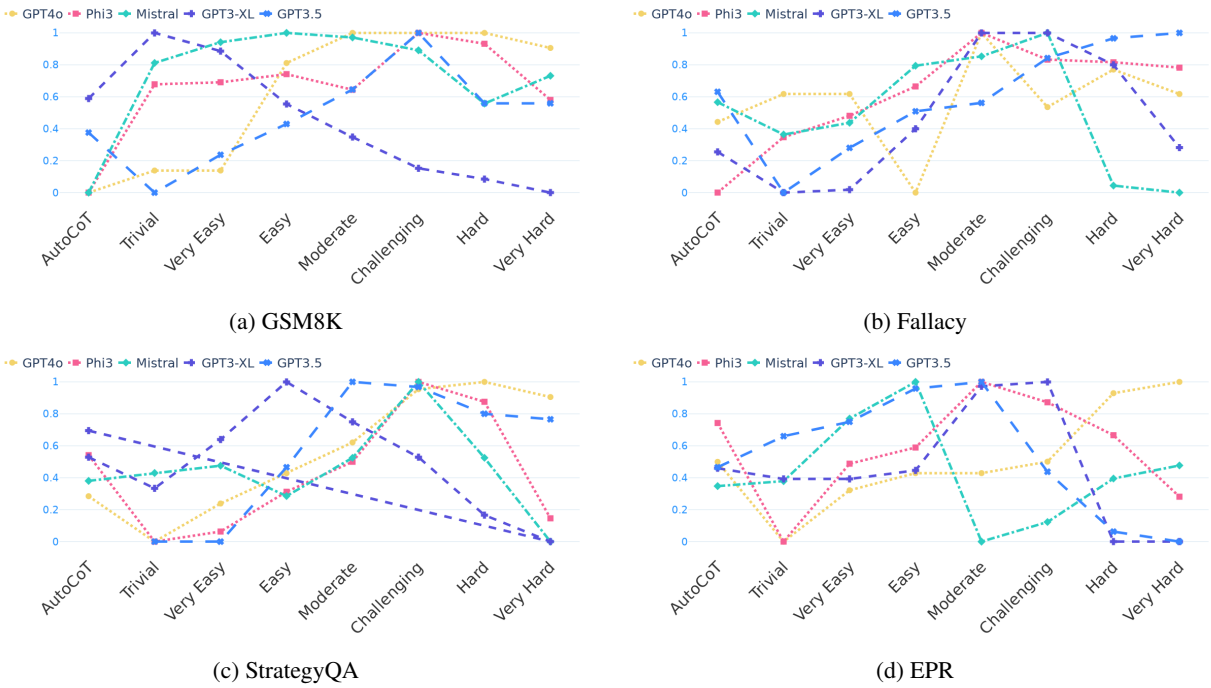
5 Results & Qualitative Analysis

5.1 Uncertainty Distribution Analysis

In this subsection, we present an analysis of the mean (μ) and standard deviation (σ) of uncertainty estimates for different LLMs across various reasoning datasets. In Figure 3, we illustrate the distribution of uncertainty estimates for GPT-3.5 on the GSM8K dataset. We have provided the comprehensive plots of the distributions in the appendix (see Figures 7 –11). The mean μ and standard deviation σ of the uncertainty estimates using the unlabeled set Q has been shown through an error bar graph in Figure 2. Notably, LLMs such as GPT3-XL and Mistral show higher uncertainty in GSM8K, particularly with a larger deviation, whereas for tasks like StrategyQA and EPR, the uncertainty is generally more consistent across models, with GPT4o displaying the lowest variation. The trend highlights that model uncertainty is highly task-dependent, with complex reasoning tasks eliciting higher variability in predictions.

[3] In general, rephrasing ensures that the intent of the question does not change.

[4] Zero-Shot and Few-Shot baselines do not use rationales or trigger phrases, instead utilizing either zero or a few examples.



(a) GSM8K (b) Fallacy

(c) StrategyQA (d) EPR

Figure 4: Normalized values of accuracy for various selection strategies using multiple LLMs.

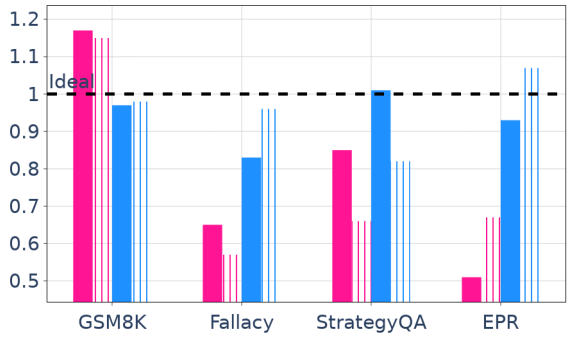


Figure 5: Sensitivity coefficient of confidence score wrt accuracy. Blue indicates *ZEUS* and Magenta for *Temp-Perb*. Solid for GPT3-XL and Dashed for GPT3.5. Coefficient using *ZEUS* is closest to ideal coefficient.

5.2 Sensitivity of Uncertainty Estimates

To assess the sensitivity of uncertainty estimates in distinguishing between helpful and redundant questions, we investigate the relationship between confidence scores and accuracy. This is done by fitting a linear regression (LR) model between the confidence score of the most common answer and its corresponding accuracy. In an ideally sensitive model, the slope coefficient of the LR would be one, indicating that a unit change in confidence directly corresponds to a unit change in accuracy. We compare our confidence scoring method against a temperature-based perturbation approach (Wan et al., 2023; Diao et al., 2023; Gao et al., 2024), referred to as *Temp-Perb*. This comparison is carried out using Zero-Shot-CoT prompting with 15 distinct temperature perturbations.

Figure 5 shows the slope coefficients for both *ZEUS* and *Temp-Perb*. Our results demonstrate that *ZEUS* consistently produces slope coefficients closer to the ideal sensitivity compared to *Temp-Perb*. Interestingly, *Temp-Perb* shows notably low sensitivity in the Logical Fallacy and EPR datasets, indicating a lack of reliability. In contrast, for GSM8K, *Temp-Perb* exhibits a coefficient exceeding 1, reflecting excessive sensitivity in this task.

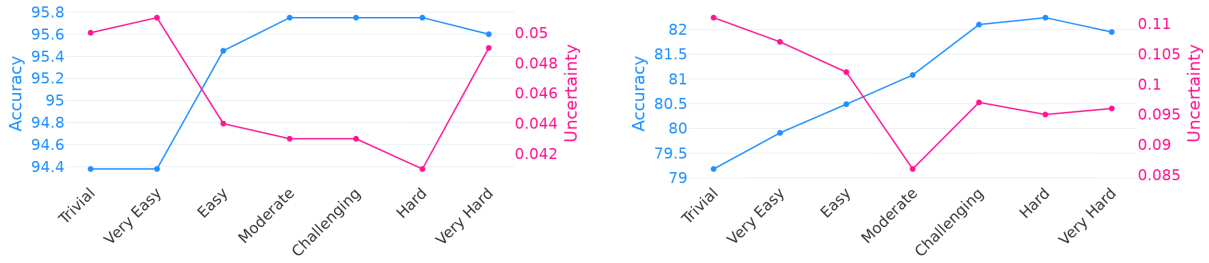
5.3 Analysis of Selection Strategies

We present the normalized accuracy values for all selection strategies, including AutoCoT, in Figure 4. Our analysis reveals that AutoCoT was consistently outperformed by at least one other strategy across all LLMs and datasets. This indicates that leveraging uncertainty-based demonstration creation can more effectively identify valuable questions that enhance model performance. To provide a clearer perspective, Table 2 details the best and worst selection strategies for each model and dataset.

LLMs exhibit distinct performance patterns across varying levels of question difficulty, which allows us to categorize them into two broad groups: advanced models (GPT-4o, Phi3, GPT-3.5) and

	GSM8K		Fallacy		StrategyQA		EPR	
Model	**Best**	**Worst**	**Best**	**Worst**	**Best**	**Worst**	**Best**	**Worst**
GPT4o	Hard	Trivial	Moderate	Easy	Hard	Trivial	Very Hard	Trivial
Phi3	Challenging	Very Hard	Moderate	Trivial	Challenging	Trivial	Moderate	Trivial
Mistral	Easy	Hard	Challenging	Very Hard	Challenging	Very Hard	Easy	Moderate
GPT3.5	Challenging	Trivial	Very Hard	Trivial	Moderate	Trivial	Moderate	Very Hard
GPT3-XL	Trivial	Very Hard	Challenging	Trivial	Easy	Very Hard	Challenging	Very Hard

Table 2: Best and worst-performing strategies across tasks for each model, indicating that GPT4o requires harder strategies for optimal performance, while GPT3-XL shows improved results with easier strategies.



(a) GSM8K (b) StrategyQA

Figure 6: Accuracy vs *Temp-Perb* Uncertainty trend across all selection strategies for **GPT4o**.

simpler models (Mistral, GPT-3 XL). This classification is based on observed performance trends rather than model size or architecture alone. Advanced models excel in handling *Hard* and *Very Hard* questions due to their superior reasoning capabilities, but they show limited gains when engaging with *Trivial* or *Very Easy* strategies, where their advanced abilities remain underutilized. On the other hand, simpler models perform better with *Trivial* and *Easy* strategies, as these align well with their baseline capabilities. However, they struggle considerably with *Hard* and *Very Hard* questions, where errors and uninformative outputs become more prevalent.

To capture general trends, we analyzed performance across the best and worst strategies for each model. Our findings highlight that while *Trivial* and *Very Easy* strategies consistently yield the lowest performance for advanced models, simpler models face significant challenges with *Hard* and *Very Hard* strategies. Notably, our categorization focuses on overall performance trends rather than model size, which places models like GPT-4o and Phi3 in the same group.

Among the selection strategies, *Trivial* and *Very Hard* tend to yield poorer performance across most models. This suggests that extremes in task difficulty—whether too easy or too hard—are generally detrimental to model accuracy. The *Hard* strategy generally improves performance for GPT-4o, whereas the *Challenging* strategy appears to be optimal for Phi3, Mistral, and GPT-3.5. These findings align with the overall performance trends observed for these models.

However, performance variations still exist across tasks and models. For instance, the Mistral model's performance declines with *Moderate* and harder strategies on the EPR task, while it improves with higher uncertainty estimates on other tasks. This indicates that selecting the optimal strategy can be complex and task-dependent. To address this, the next subsection will explore methods for determining the most effective selection strategy.

5.4 Choosing Optimal Selection Strategy

Upon constructing the demonstration for each strategy, we need to identify the optimal strategy for a given task and model. We calculate the average uncertainty on the unlabelled set Q while keeping the demonstration unchanged. The optimal strategy is the one with the lowest entropy, as this tends to strongly correlate with higher accuracy. *Temp-Perb* provides well-calibrated uncertainty estimates, although it lacks the sensitivity required to effectively differentiate between similar questions. Despite this limitation, its well calibration makes *Temp-Perb* suitable for selecting the best-performing strategy based on uncertainty estimates.

Method	GSM8K					Fallacy				
	GPT4o	Phi3	Mistral	GPT3.5	GPT3-XL	GPT4o	Phi3	Mistral	GPT3.5	GPT3-XL
Zero-Shot	49.4	50.7	45.3	12.6	10.7	80.5	81.8	71.1	63.9	48.3
Few-Shot	84.0	50.7	45.3	16.5	14.4	92.5	90.4	62.9	76.9	**79.8**
Zero-Shot-CoT	94.8	85.9	51.8	60.4	44.7	84.8	87.5	67.1	67.7	61.7
Manual-CoT	89.3	81.9	42.4	56.4	43.9	90.1	90.1	64.3	-	-
Auto-CoT	94.2	87.6	47.2	58.5	44.6	97.0	85.6	74.4	76.9	66.7
ZEUS (LU)	**95.8**	**89.9**	57.3	**62.9**	**51.9**	98.0	94.0	**78.5**	**79.4**	76.4
ZEUS (HA)	**95.8**	**89.9**	**57.6**	**62.9**	**51.9**	**98.0**	**94.0**	**78.5**	**79.4**	76.4

	StrategyQA					EPR				
	GPT4o	Phi3	Mistral	GPT3.5	GPT3-XL	GPT4o	Phi3	Mistral	GPT3.5	GPT3-XL
Zero-Shot	65.2	56.6	59.8	54.4	16.6	61.2	72.2	45.2	60.0	61.5
Few-Shot	77.6	65.8	61.1	66.2	64.8	83.0	64.0	55.2	**75.6**	58.2
Zero-Shot-CoT	70.7	67.5	59.0	57.4	51.2	64.7	**79.8**	65.7	60.2	59.3
Manual-CoT	81.1	**68.9**	**63.8**	**68.6**	57.6	**84.2**	64.0	57.7	-	-
Auto-CoT	80.1	64.5	57.9	64.9	64.1	68.2	75.3	52.5	52.5	59.5
ZEUS (LU)	81.1	67.7	59.8	66.8	**66.5**	72.8	76.2	**68.5**	65.3	**66.2**
ZEUS (HA)	**82.2**	67.7	59.8	66.8	**66.5**	72.8	77.0	**68.5**	65.3	**66.2**

Table 3: Accuracy on various datasets. *ZEUS (HA)* chooses the best performing strategy for each dataset while *ZEUS (LU)* chooses the strategies having lowest *Temp-Perb* uncertainty estimates.

Therefore, we use *Temp-Perb* for uncertainty estimation to determine the optimal selection strategy for a given model and task.

In Figure 6, we illustrate the accuracy of various selection strategies for GPT-4o in relation to *Temp-Perb* based uncertainty estimates. The data indicates that the accuracy is inversely correlated with uncertainty across all four datasets. This inverse relationship allows us to identify the optimal selection strategy as the one associated with the lowest uncertainty. We have included similar analyses for other models in the appendix (cf. Figures 12 – 16).

5.5 Comparison with Baselines

The selection strategy with the lowest uncertainty is denoted as *ZEUS (LU)*, while the strategy with the highest accuracy is represented by *ZEUS (HA)*. Table 3 demonstrates that *ZEUS (LU)* and *ZEUS (HA)* yield nearly identical performance, underscoring the robustness of the *Temp-Perb* uncertainty estimates. In general, the optimally selected *ZEUS(LU)* either outperforms all baseline methods or comes in a close second to in a few cases across three datasets (GSM8K, Fallacy, and Strategy QA), with only a few exceptions. *ZEUS* methods consistently outperform all baseline strategies on the GSM8K and Fallacy datasets, with the exception of GPT-3 XL on the Fallacy dataset. For the StrategyQA dataset, Manual-CoT achieves the highest accuracy for most models, highlighting the effectiveness of human-crafted demonstrations. On the EPR dataset, *ZEUS* surpasses Zero-Shot, Zero-Shot-CoT, and Auto-CoT methods across most models. Overall, *ZEUS* methods either match or exceed the accuracy of these baseline strategies without requiring manual annotations.

6 Conclusion

This paper introduces the zero-shot uncertainty-based *ZEUS* method for evaluating and selecting optimal strategies based on uncertainty estimates. Our analysis reveals that *ZEUS* provides highly sensitive and reliable uncertainty estimates, outperforming temperature-based perturbation approaches (*Temp-Perb*) in distinguishing between helpful and redundant questions.

Our findings classify models into two groups based on their optimal strategies. Advanced models like GPT-4o, Phi3, and GPT3.5 perform best with *Hard* and *Challenging* example selection strategies, effectively leveraging their greater capabilities to tackle complex queries. In contrast, simpler models such as Mistral and GPT3-XL benefit more from *Trivial* and *Easy* strategies, where even low-uncertainty questions yield valuable information. By selecting the strategy with the lowest uncertainty estimates, *ZEUS(LU)* (recommended) achieves performance comparable to the best-performing strategies *ZEUS(HA)*, without requiring manual annotations. Overall, *ZEUS* con-

sistently matches or surpasses baseline accuracy, demonstrating its robustness and sensitivity in improving model performance.

7 Limitation

While our work demonstrates the effectiveness of the *ZEUS* method, there are several limitations and avenues for future research. First, the selection strategies in our current approach require exhaustive exploration to find the optimal strategy, which can be time-consuming and computationally expensive. This process could be automated by incorporating a greedy search algorithm based on uncertainty estimates, allowing for more efficient strategy selection. Another limitation is our reliance on uncertainty estimates from unlabeled questions, without examining the impact of dataset attributes like diversity or size. These factors could affect the estimates and lead to suboptimal strategy selection. Future work should explore these effects to improve robustness.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

David Arthur and Sergei Vassilvitskii. 2007. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.

Markus Bayer and Christian Reuter. 2024. Activellm: Large language model-based active learning for textual few-shot scenarios. *arXiv preprint arXiv:2405.10808*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. 2024. Active prompting with chain-of-thought for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1350, Bangkok, Thailand. Association for Computational Linguistics.

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2024. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36.

Yifan Fu, Xingquan Zhu, and Bin Li. 2013. A survey on instance selection for active learning. *Knowledge and information systems*, 35(2):249–283.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Tnternational Conference on Machine Learning*, pages 1050–1059.

Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, and Kamalika Das. 2024. Spuq: Perturbation-based uncertainty quantification for large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2336–2346.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

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*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, and Bernhard Schoelkopf. 2022.

Logical fallacy detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7180–7198.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113, Mexico City, Mexico. Association for Computational Linguistics.

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

Shanu Kumar, Sandipan Dandapat, and Monojit Choudhury. 2022. " diversity and uncertainty in moderation" are the key to data selection for multilingual few-shot transfer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1042–1055.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

OpenAI. 2024. Introducing gpt-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-09-16.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

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, Hong Kong, China. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Guy Rotman and Roi Reichart. 2022. Multi-task active learning for pre-trained transformer-based models. *Transactions of the Association for Computational Linguistics*, 10:1209–1228.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079.

Kashun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic prompt augmentation and selection with chain-of-thought from labeled data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12113–12139, Singapore. Association for Computational Linguistics.

Damien Sileo and Antoine Lernould. 2023. Mindgames: Targeting theory of mind in large language models with dynamic epistemic modal logic. *arXiv preprint arXiv:2305.03353*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Christian Tomani, Kamalika Chaudhuri, Ivan Evtimov, Daniel Cremers, and Mark Ibrahim. 2024. Uncertainty-based abstention in llms improves safety and reduces hallucinations. *arXiv preprint arXiv:2404.10960*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. 2020. Uncertainty estimation using a

single deep deterministic neural network. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9690–9700. PMLR.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Akim Tsvigun, Daniil Vasilev, Rui Xing, Abdelrahman Boda Sadallah, Lyudmila Rvanova, Sergey Petrakov, Alexander Panchenko, et al. 2024. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *arXiv preprint arXiv:2406.15627*.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. 2023. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3493–3514, Toronto, Canada. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona selfcollaboration. *arXiv preprint arXiv:2307.05300*, 1(2):3.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

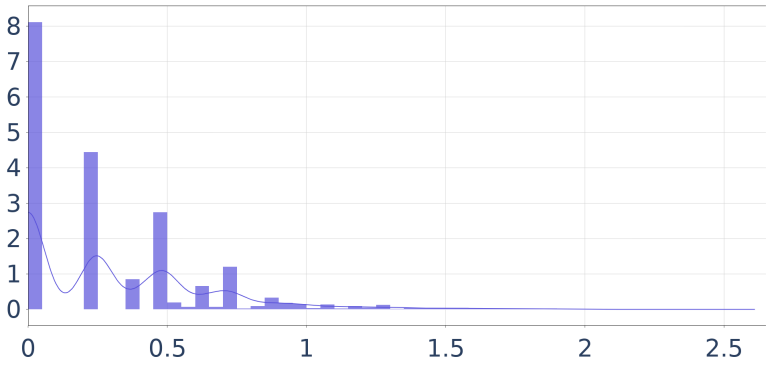
Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. 2023. Exchange-of-thought: Enhancing large language model capabilities through cross-model communication. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15135–15153, Singapore. Association for Computational Linguistics.

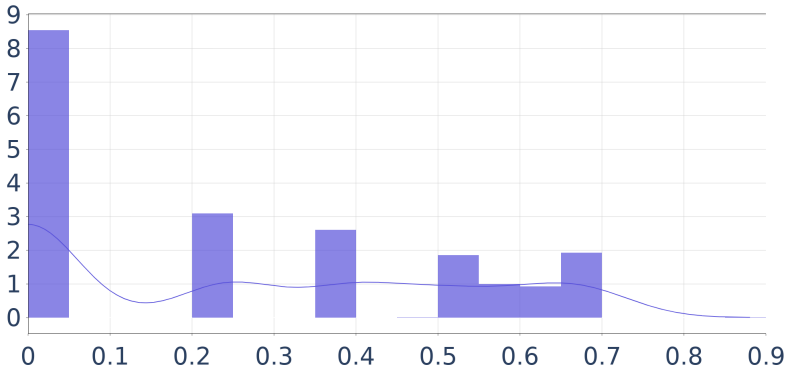
Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Yongfeng Huang, Ruyi Gan, Jiaxing Zhang, and Yujiu Yang. 2023. Solving math word problems via cooperative reasoning induced language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4471–4485, Toronto, Canada. Association for Computational Linguistics.

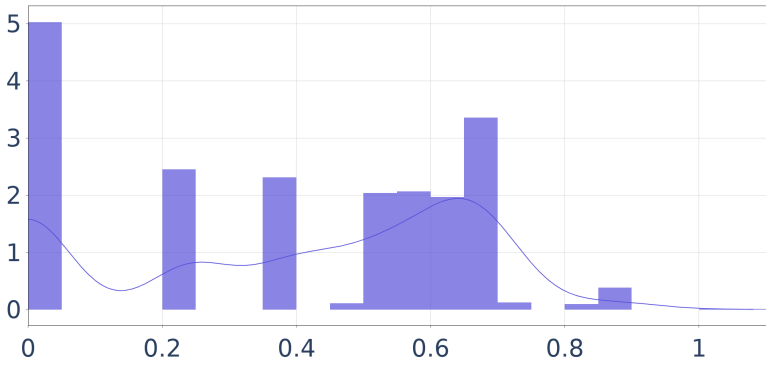
A Appendix



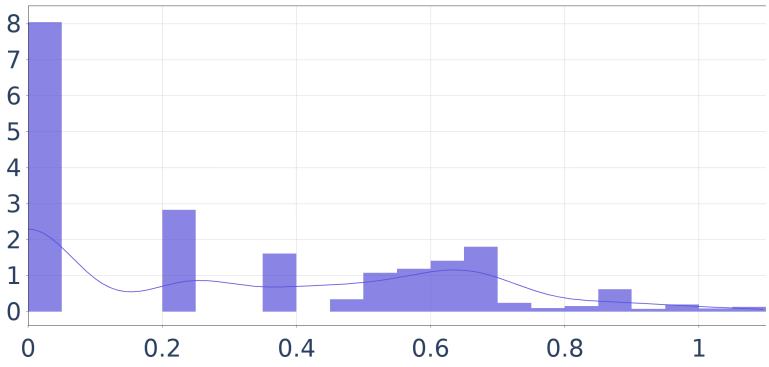
(a) GSM8K



(b) Fallacy

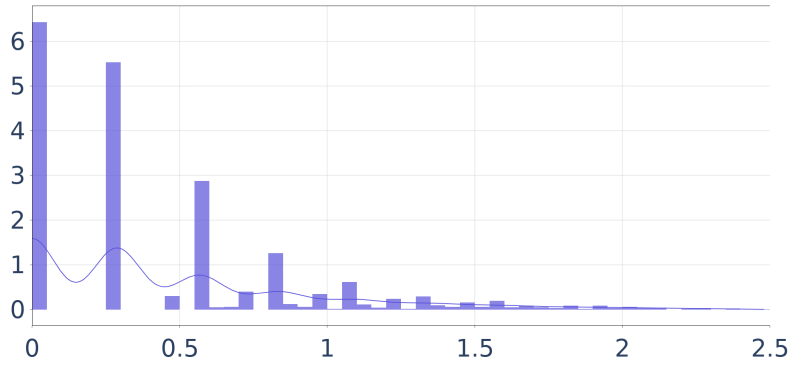


(c) EPR

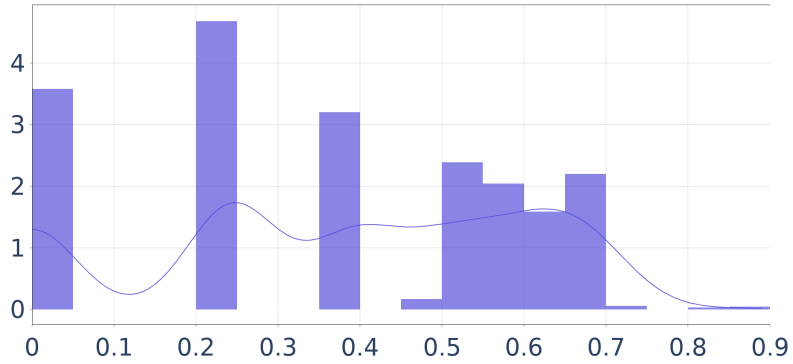


(d) StrategyQA

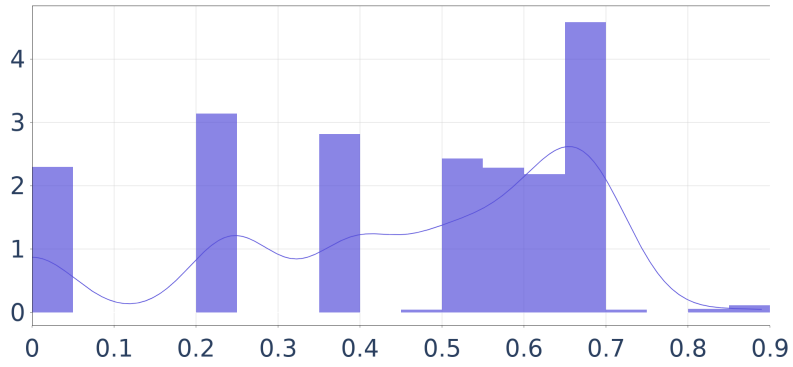
Figure 7: Probability density function of uncertainty estimates of our method using **GPT4o**.



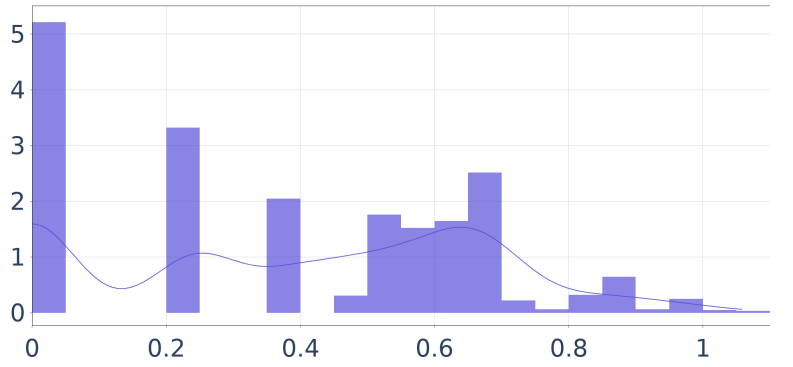
(a) GSM8K



(b) Fallacy

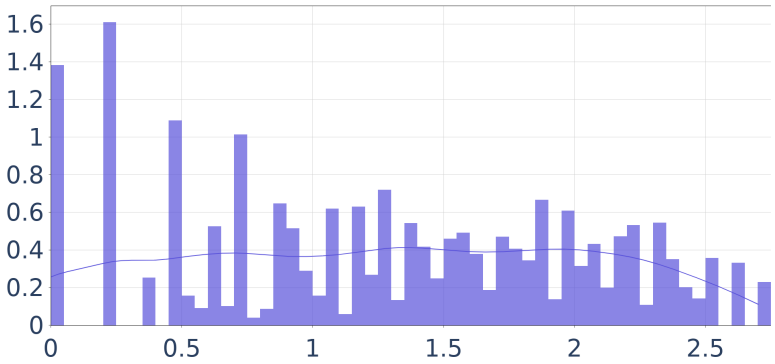


(c) EPR

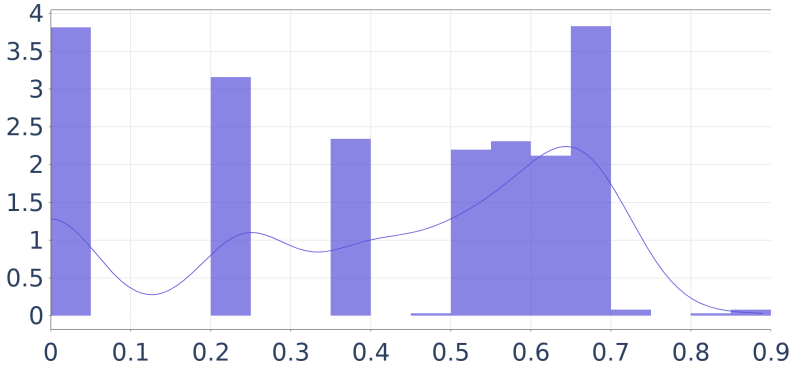


(d) StrategyQA

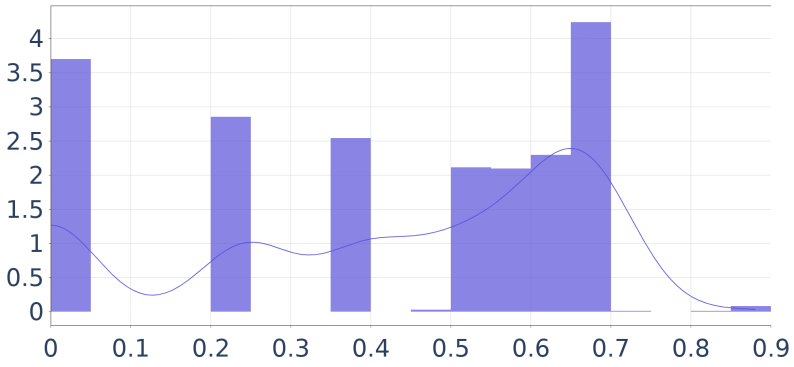
Figure 8: Probability density function of uncertainty estimates of our method using **Phi3**.



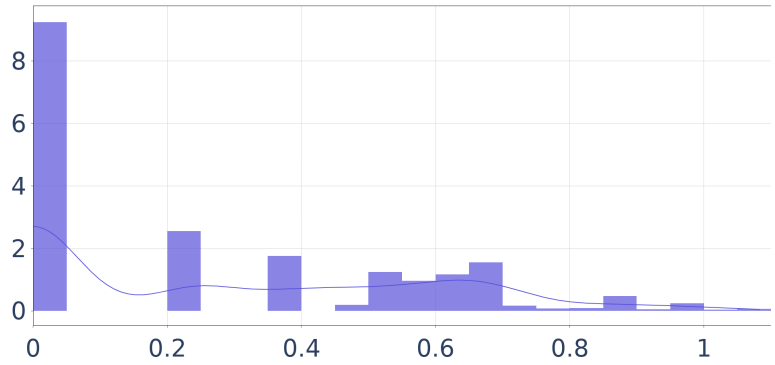
(a) GSM8K



(b) Fallacy

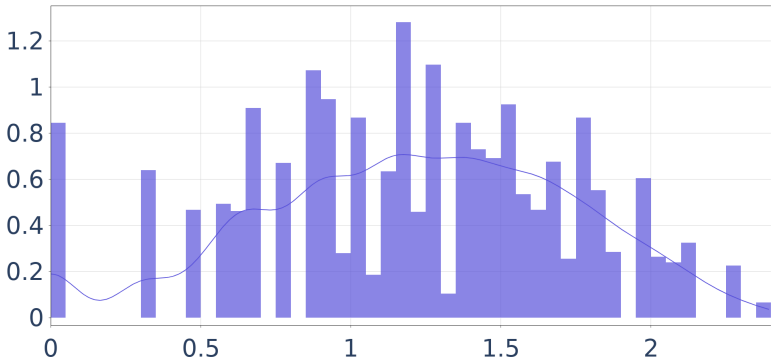


(c) EPR

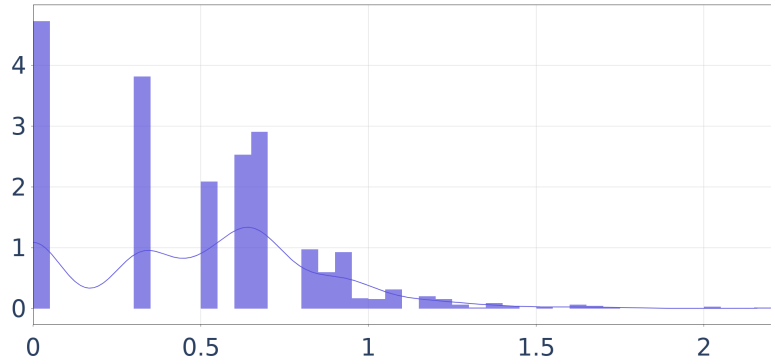


(d) StrategyQA

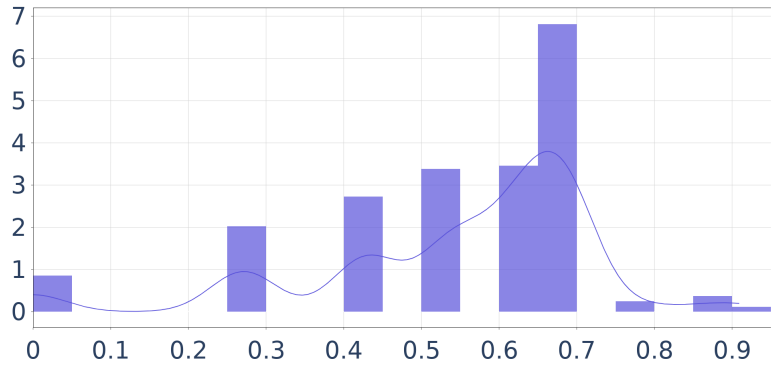
Figure 9: Probability density function of uncertainty estimates of our method using **Mistral**.



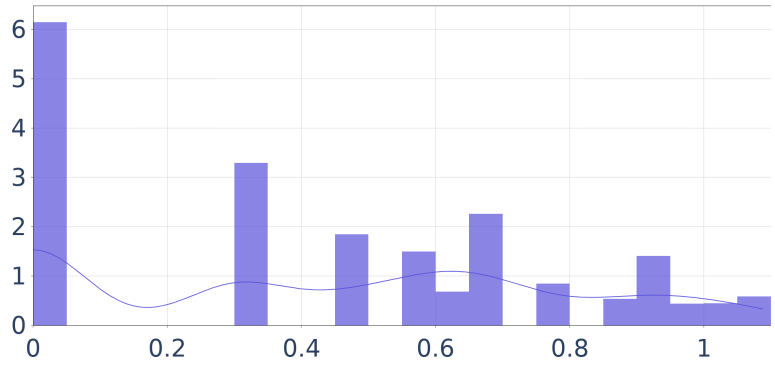
(a) GSM8K



(b) Fallacy

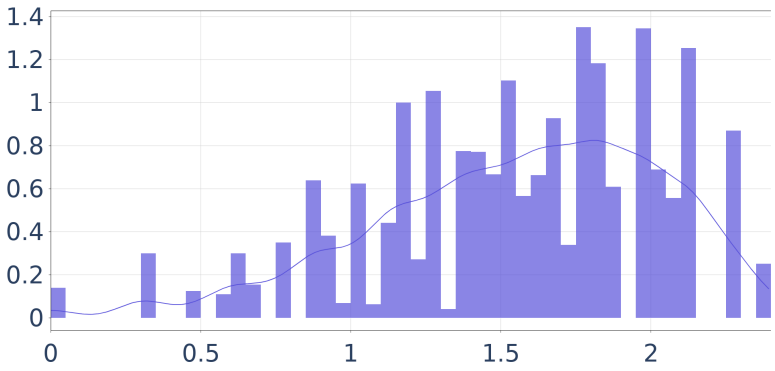


(c) EPR

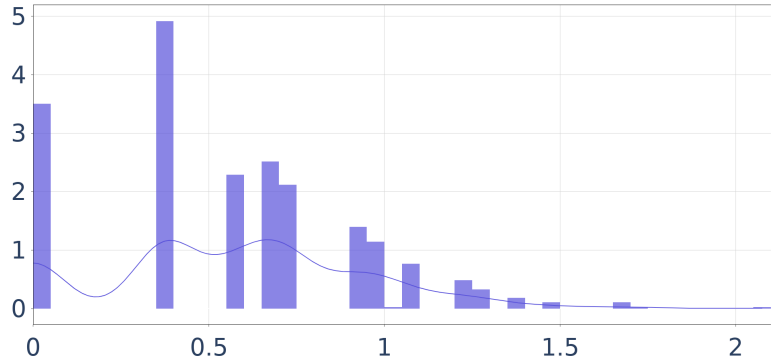


(d) StrategyQA

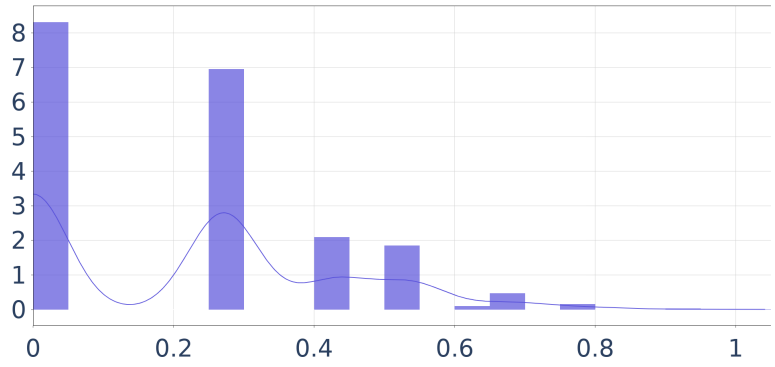
Figure 10: Probability density function of uncertainty estimates of our method using **GPT3.5**.



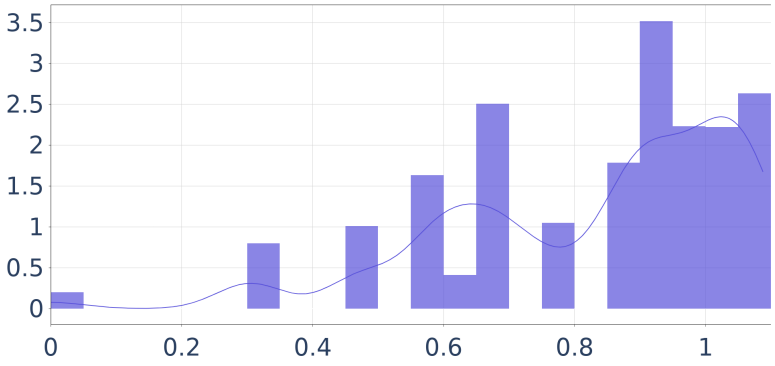
(a) GSM8K



(b) Fallacy

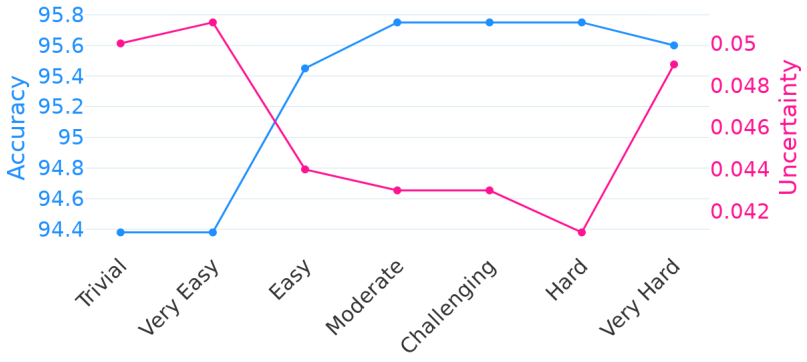


(c) EPR

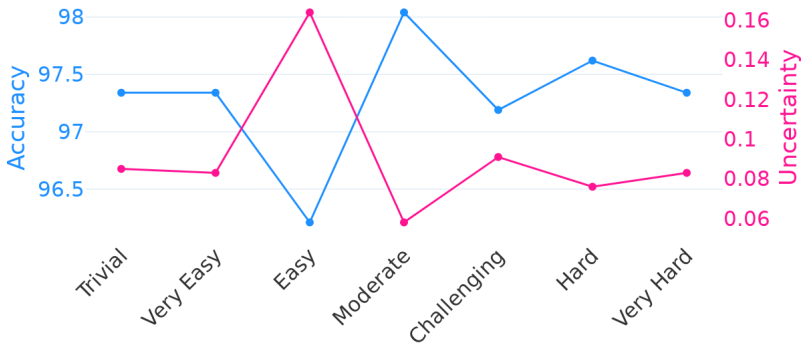


(d) StrategyQA

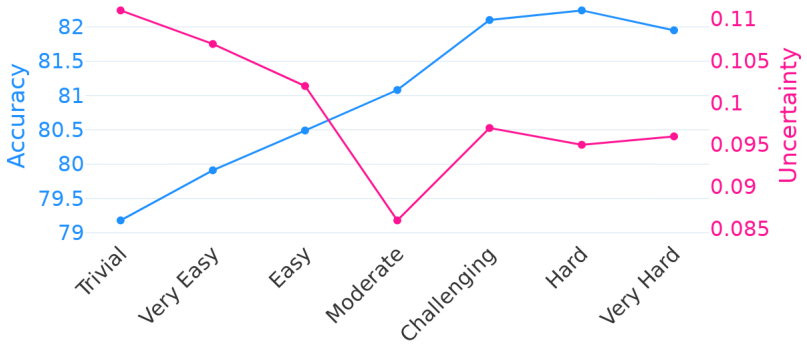
Figure 11: Probability density function of uncertainty estimates of our method using **GPT3-XL**.



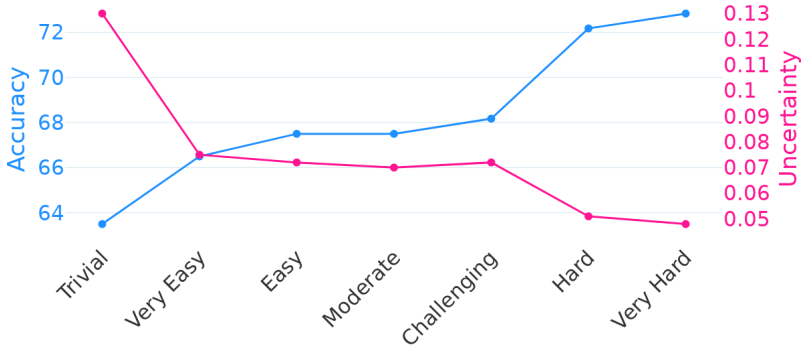
(a) GSM8K



(b) Fallacy

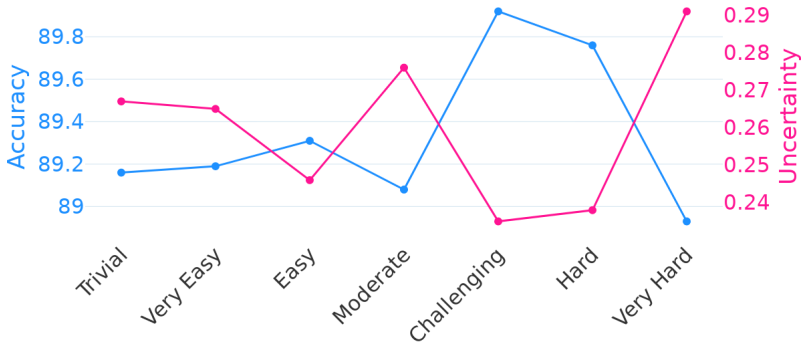


(c) StrategyQA

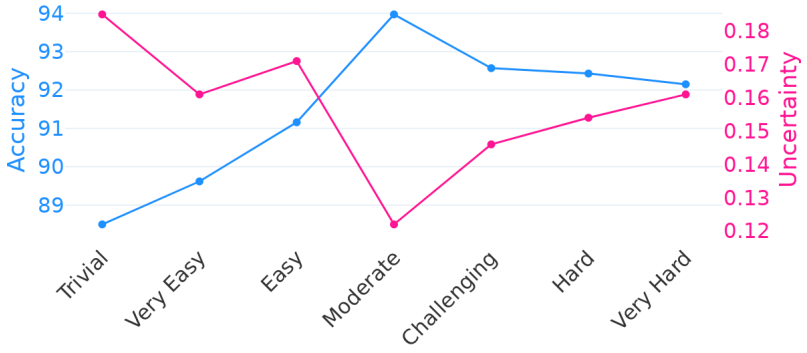


(d) EPR

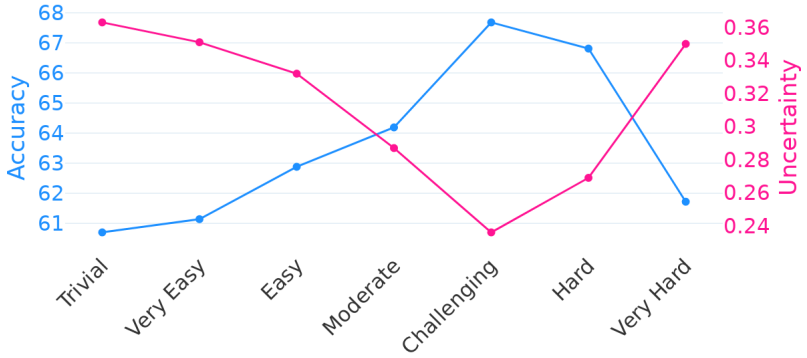
Figure 12: Accuracy vs *Temp-Perb* Uncertainty trend across all selection strategies for**GPT4o**



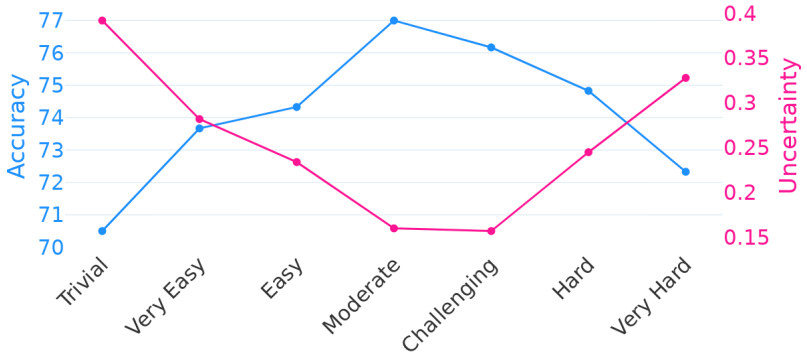
(a) GSM8K



(b) Fallacy

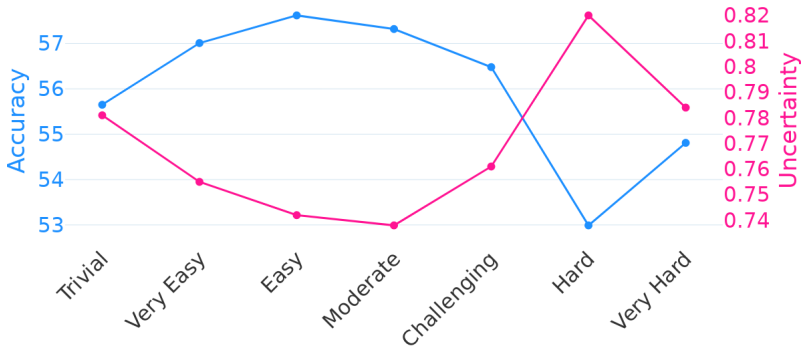


(c) StrategyQA

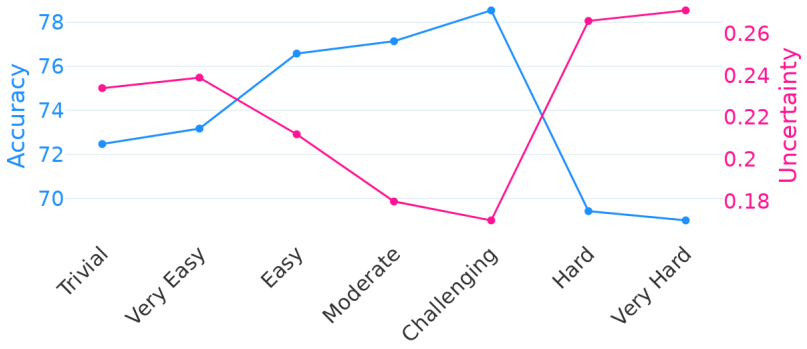


(d) EPR

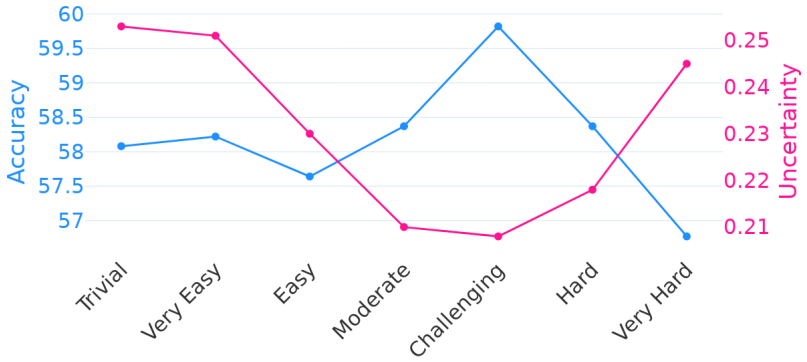
Figure 13: Accuracy vs *Temp-Perb* Uncertainty trend across all selection strategies for**Phi3**



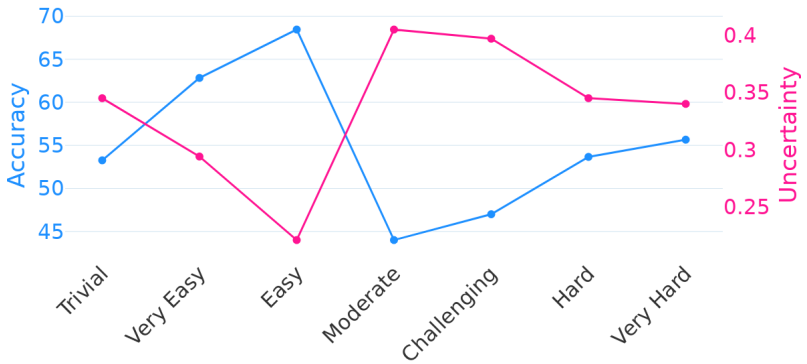
(a) GSM8K



(b) Fallacy

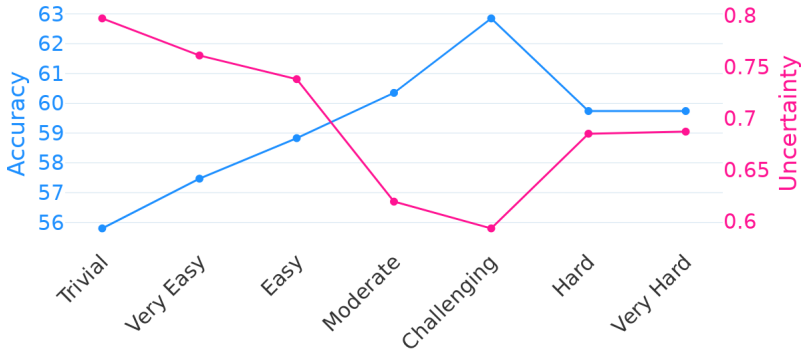


(c) StrategyQA

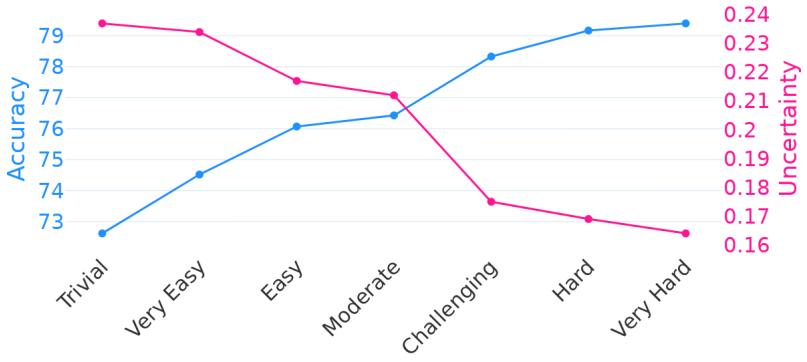


(d) EPR

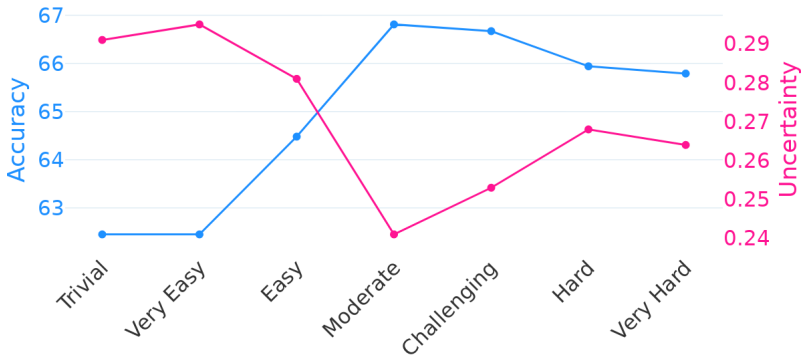
Figure 14: Accuracy vs *Temp-Perb* Uncertainty trend across all selection strategies for**Mistral**



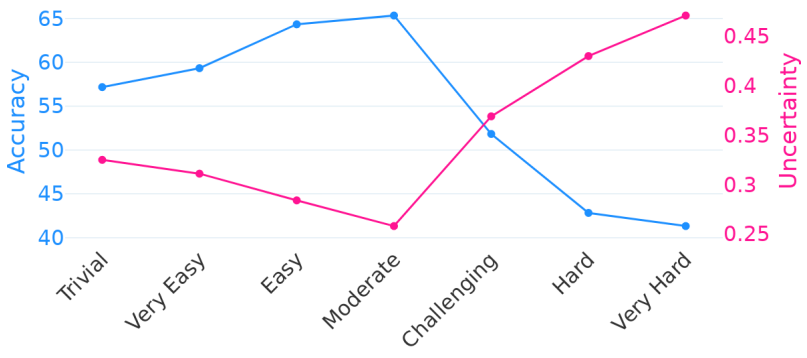
(a) GSM8K



(b) Fallacy

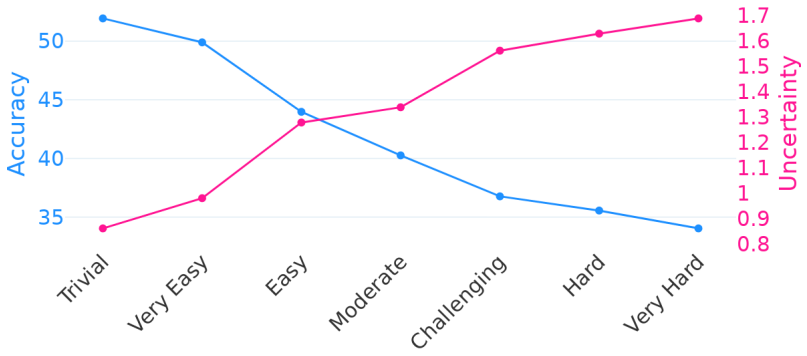


(c) StrategyQA

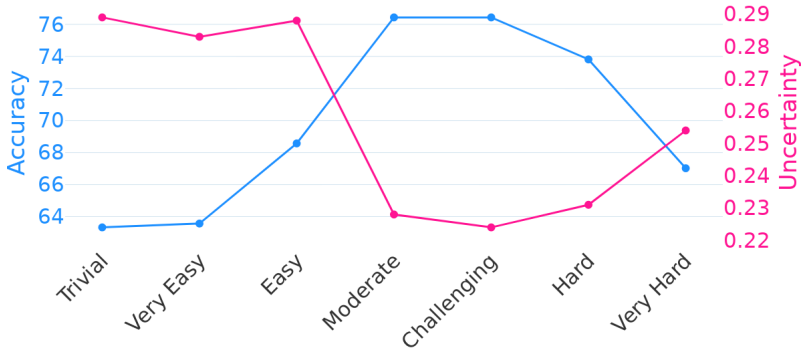


(d) EPR

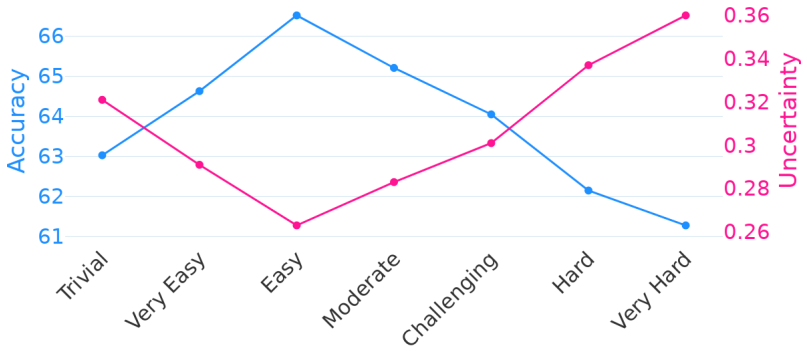
Figure 15: Accuracy vs *Temp-Perb* Uncertainty trend across all selection strategies for**GPT3.5**



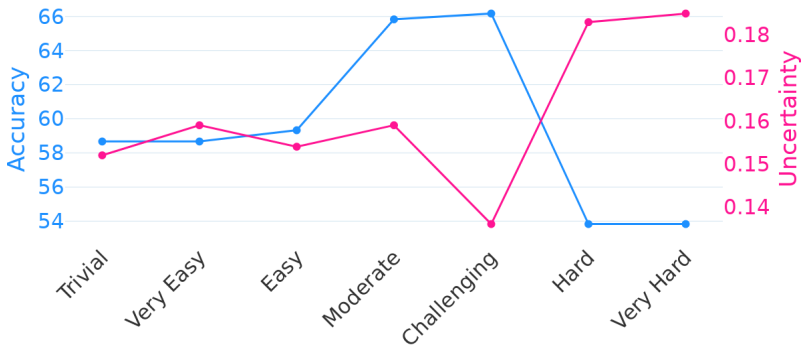
(a) GSM8K



(b) Fallacy



(c) StrategyQA



(d) EPR

Figure 16: Accuracy vs *Temp-Perb* Uncertainty trend across all selection strategies for**GPT3-XL**

	GPT3.5		GPT3-XL		GPT4O		Phi3		Mistral	
Dataset	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
GSM8K	1.21	0.53	1.55	0.48	0.30	0.35	0.45	0.48	1.28	0.76
Fallacy	0.49	0.36	0.57	0.37	0.26	0.26	0.37	0.23	0.41	0.25
EPR	0.55	0.18	0.22	0.21	0.39	0.27	0.46	0.22	0.42	0.25
StrategyQA	0.43	0.35	0.83	0.22	0.32	0.31	0.39	0.29	0.28	0.30

Table 4: Mean and standard deviation of uncertainty values as error graph -specific statistics across models.