Are LLMs Aware that Some Questions are not Open-ended?

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have shown the impressive capability of answering questions in a wide range of scenarios. However, when LLMs face different types of questions, it is worth exploring whether LLMs are aware that some questions have limited answers and have to respond more deterministically but some do not. We refer to the ability as *question awareness* that LLMs know to adjust the determinacy of the answers according to the questions. The lack of question awareness leads to two contradictory issues: (1) Too casual to answer non-open-ended questions. (2) Too boring to answer open-ended questions. In this paper, we first evaluate the question awareness of LLMs. The experimental results show that LLMs have the above issues of lacking the awareness of questions in certain domains, e.g. factual knowledge. To mitigate these issues, we propose a method called Question Awareness Temperature Sampling (QuATS). This method enhances the question awareness of LLMs by dynamically adjusting the output distributions based on question features. The automatic adjustment in QuATS eliminates the need for manual temperature tuning in text generation and improves model performance in various benchmarks.

1 Introduction

Large language models (LLMs) (OpenAI, 2022, 2023; Anthropic, 2023) have emerged as groundbreaking innovations in achieving a remarkable level of fluency and comprehension in question-answering using the human language (Taori et al., 2023; Chiang et al., 2023; Xu et al., 2023). Though LLMs can answer enormous questions with their knowledge base, we are hard to tell if the LLMs are aware of what kinds of questions they are answering. In other words, do LLMs understand that, open-ended questions encourage more casual and creative answers, but non-open-ended questions, e.g. problems about calculations and factual knowledge, need more accurate answers? We refer to this ability *question awareness* that one knows which type of questions requires deterministic answers and which does not.

The question awareness indicates LLMs can identify which questions need more accurate answers and choose to act more deterministic. It is significant to explore that the question awareness of LLMs has a relationship to the model hallucinations and how to improve it because LLMs may be more likely to generate hallucinated answers when they are not sure.

In this paper, we explore whether LLMs have question awareness across different types (open-ended/non-open-ended) of questions. To evaluate the question awareness, we have to first introduce a metric. Because LLMs sample the answer tokens from output distribution, as shown in Figure 1, we can examine the degree of the determinacy of LLMs from the "steepness" of the output distributions. A steeper output distribution means the model has confidence in generating the token with a large probability and a flat one means the models have more choices of generating what token. It corresponds to the degree of question awareness. Therefore, we can investigate question awareness by checking if there is a difference in the output distribution when LLMs are asked different types of questions. We collect different types of non-open-ended questions and open-ended questions for evaluation. The experimental results show that LLMs have a certain degree of question awareness but lack the awareness in some scenarios, e.g., factual knowledge, thus giving more casual and hallucinated answers in some cases.

To alleviate the influence of lacking question awareness, we propose Question Awareness Temperature Sampling (QuATS), a method that enhances question awareness of LLMs and adjusts the output distributions through temperature according to the question type. When facing different

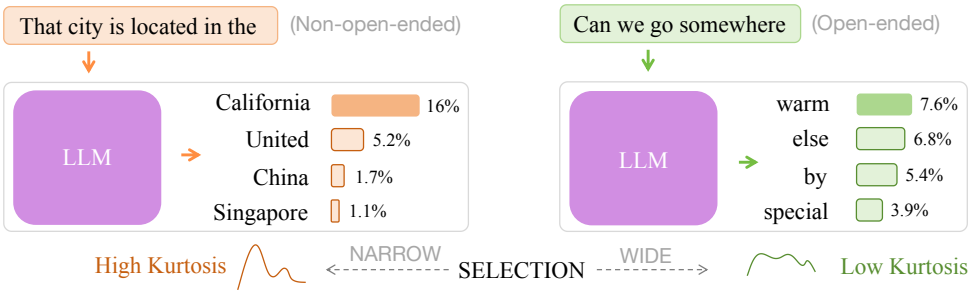

Figure 1: LLM should have question awareness to handle different questions.

questions, LLMs can adaptively choose to be more deterministic or not by themselves avoiding the tedious process of temperature tuning. To sum up, our contributions are stated as follows:

- We evaluate the question awareness of the LLMs and observe that LLMs have the fundamental ability to identify open-ended and non-open-ended questions but lack effective awareness in some domains, e.g., factual knowledge.
- We propose Question Awareness Temperature Sampling (QuATS). It enables LLMs to choose to be deterministic or not when answering different questions by adaptively adjusting the sampling temperature.
- Our experimental results show that the QuATS enhances the question awareness of the LLMs and improves the performance on various benchmarks.

2 Question Awareness Evaluation

In this section, we evaluate the question awareness of LLMs on two widely used open-source LLMs, LLaMA 2 (Touvron et al., 2023) and Falcon (Penedo et al., 2023), on different question types. It is noted that we do not evaluate question awareness on GPT-3.5-turbo (OpenAI, 2022) and GPT-4 (OpenAI, 2023) because we can not obtain the output distributions from the APIs.

2.1 Formulation of Supervised Fine-tuning

To better clarify the evaluation process of the question awareness, we first give a formulation of supervised fine-tuning (SFT) / instruction fine-tuning (Cobbe et al., 2021). Supervised fine-tuning (SFT) serves as a bridge that leverages the foundational language comprehension gained during pre-training and then tailors it for conversational purposes. For an auto-regressive language model, denoted as ϕ, given a joint sequence $s = (x_1, x_2, \ldots, y_1, y_2, \ldots, y_T)$ of a question x and an answer y, we minimize the SFT training objective only on the answer sequence y in the teacher forcing way (Lamb et al., 2016):

$$\mathcal{L}_{SFT}(\phi) = \mathbb{E}\left(-\sum_{t=1}^{T} \log p_\phi(\hat{y}_t|x, y_{<t})\right). \quad (1)$$

During the inference after supervised fine-tuning, we sample token from the output distribution $p_\phi(\hat{y}_t|x, y_{<t})$ to generate the token at step t.

2.2 Metric

The distribution $p_\phi(\hat{y}_t|x, y_{<t}) = (p_1, p_2, \ldots, p_n)$ in Eq 1 indicates how LLMs are confident on the next token to predict among the entire token vocabulary with size n. A steeper distribution demonstrates LLMs are more sure to predict the tokens with larger probabilities, which is crucial for answering the non-open-ended question, as shown in Figure 1. Therefore, we introduce kurtosis to measure the steepness of the distribution. If the distribution is steeper, the kurtosis will get larger. We use the average kurtosis of the distribution over the whole answer to reflect the general determinacy of the answer. We calculate the average kurtosis $\mathcal{K}$ over the entire output distributions as follows:

$$\begin{aligned} \kappa_t &= \frac{\frac{1}{n}\sum_{i=1}^{n}(p_i - \overline{p})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(p_i - \overline{p})^2\right)^2} - 3, \\ \mathcal{K} &= \frac{1}{T}\sum_{t=1}^{T}(\kappa_t/\kappa_{\text{one-hot}}), \end{aligned} \quad (2)$$

where p_i is the probability of the token to predict at step t and κ_t is the kurtosis of the distribution of the token at step t. We normalize the average kurtosis to $(0, 1)$ by dividing the kurtosis of the one-hot distribution $\kappa_{\text{one-hot}}$.

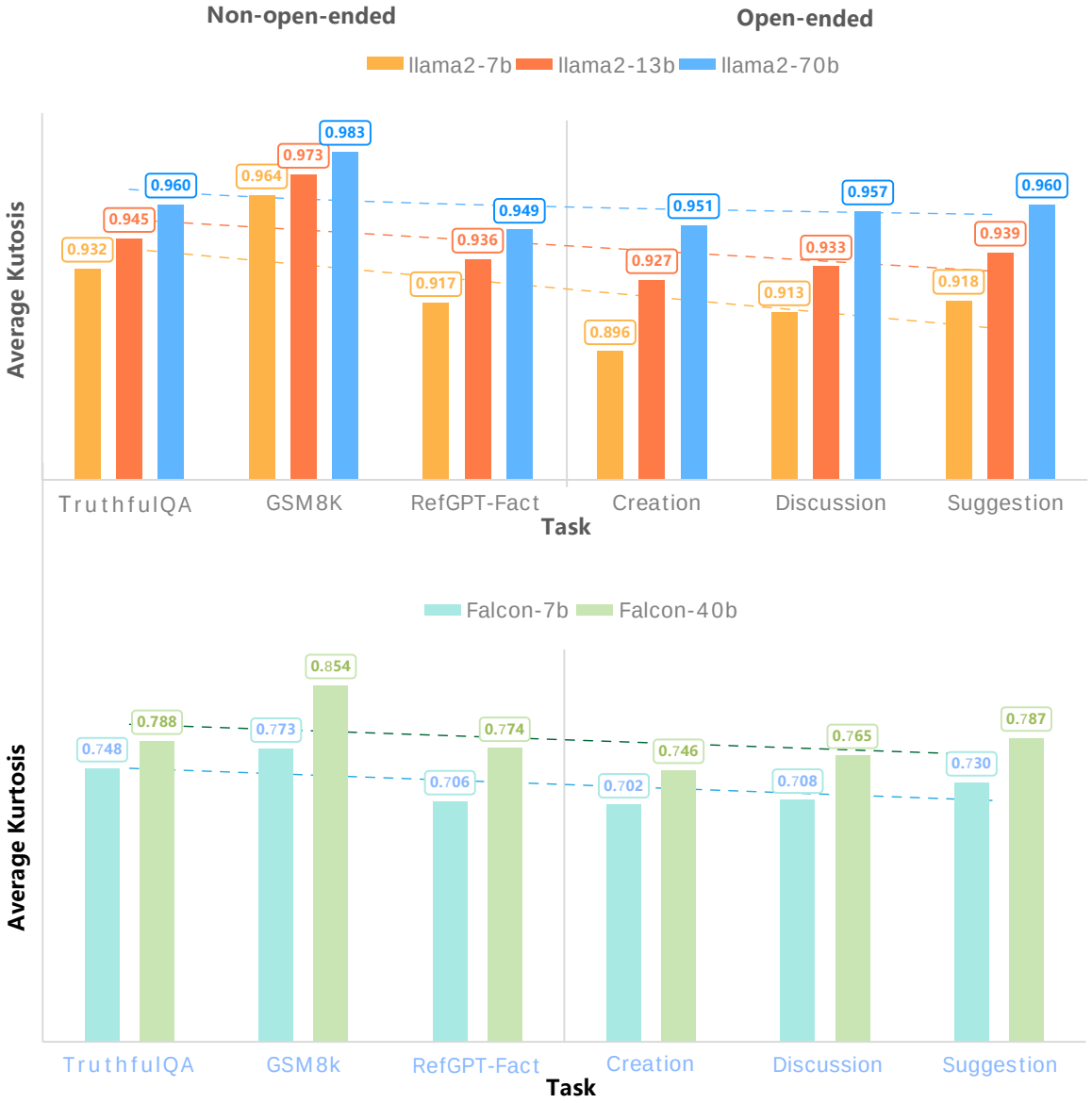

Figure 2: The result of question awareness evaluation. The dotted lines are the trend lines of the kurtosises, which are linearly fitted.

2.3 Evaluation Process

Evaluation Dataset To evaluate question awareness, we need to construct an evaluation dataset where questions have distinctions in terms of the determinacy to answer them. Therefore, we collect the questions of mainly two types, non-open-ended and open-ended questions. We collect three types of non-open-ended questions that have only fixed/limited answers: (1) TruthfulQA: We select 100 hard questions about commonsense knowledge from the TruthfulQA dataset (Lin et al., 2022). (2) GSM8K: We select 100 school math word problems of diverse grades from the GSM8K dataset (Cobbe et al., 2021). (3) RefGPT-Fact: We select 100 questions about world knowledge from the RefGPT-Fact dataset (Yang et al., 2023), which includes factual knowledge of histories, celebrities, places, and so on. We also collect open-ended questions that encourage more creative answers: (1) Creation: content creation including written articles, emails, and so on. (2) Discussion: discussion on a certain topic, (3) Suggestion: offering useful suggestions. All these subsets of non-open-ended type have 100 questions each and are carefully filtered by humans from ShareGPT datasets (Dom Eccleston, 2023).

Setup Using the evaluation dataset, we investigate chat models with different sizes, including LLaMA 2-Chat 7b/13b/70b (Touvron et al., 2023), Falcon-instruct 7b/40b (Penedo et al., 2023). We

calculate the average kurtosises of the output distributions from the models on the evaluation dataset to evaluate the question awareness.

2.4 Results and Analysis

LLMs lack a strong sense of question awareness. In Figure 2, from the trend lines, we observe that the kurtosises of the non-open-ended questions are not significantly higher than the kurtosises of the open-ended questions in most models. For non-open-ended questions, LLMs have fundamental question awareness, especially in answering commonsense knowledge in TruthfulQA and math problems in GSM8K. However, LLMs do not show more determinacy when answering questions about factual knowledge in RefGPT-Fact, where the kurtosises are close to the average of open-ended questions. It shows that LLMs fail to recognize some questions about world knowledge are required to be answered carefully, thus leading to casual and hallucinated answers. For open-ended questions, similar problems can be found: Most LLMs have relatively lower kurtosis in Creation but fail to be more creative and casual in Discussion and Suggestion. It suggests the models may give repetitive answers to these questions if we ask several times.

Larger models have more confidence in text generation. Though we do not observe an emergence of question awareness in larger models, we find that models with larger sizes tend to be more deterministic and focused with higher kurtosis. It means they are more confident in their answer.

3 Question Awareness Temperature Sampling

Based on the findings above, we want to further improve the performance of LLMs by enhancing the question awareness of the LLMs in more scenarios. Therefore, we propose the Question Awareness Temperature Sampling (QuATS), which adaptively adjusts the sampling temperature according to the given questions. We first illustrate how sampling temperature affects the output distributions in text generation. Then we will introduce the mechanism of our QuATS. For simplification, we consider kurtosis and steepness to be the same things.

3.1 Temperature Sampling

In text generation, we can adjust the steepness of the output distribution by setting the temperature in the Softmax function as follows:

$$p_\phi(\hat{y}_t|x, y_{<t}) = \text{Softmax}\left(\frac{l_{\phi,t}(x, y_{<t})}{\mathcal{T}}\right), \quad (3)$$

where the $l_{\phi,t}(x, y_{<t})$ is the output logit of the token at the step t. We can consider the Softmax function without $\mathcal{T}$ as the Softmax function with a temperature of 1. If we sample the next token with a lower temperature, the output distribution will get steeper thus likely sampling the token with a large probability. When the LLMs face different questions, we want the LLMs themselves to decide the determinacy of the answer by adjusting the temperature to change the steepness of output distributions. However, it is a challenge that temperature is a hyperparameter that can not be optimized. We bypass the direct optimization and use the neural network to predict the tendency of how temperature changes according to the determinacy.

3.2 Training DetBlock to Predict Determinacy

We introduce a tiny network called **DetBlock** to predict the determinacy and leverage it to find the optimal temperature for sampling. Before doing inference with QuATS, we train the DetBlock to predict how deterministic and focused they should be based on the given questions. After DetBlock is ready, we convert the predicted determinacy score to the sampling temperature and adaptively adjust the temperature on the fly during inference.

Training Dataset To train the DetBlock, we construct a dataset where questions are rated by a scalar determinacy score. To be specific, we rate the open-ended questions with lower scores and non-open-ended questions with higher scores. We use the questions as the input and the determinacy scores as the training labels.

DetBlock Structure As shown in Figure 3, we design a tiny network to be DetBlock to predict the determinacy score. The backbone of DetBlock is copied from the last decoder layer of the LLM. We add the QuATS head to the end of the backbone to predict a scalar score of determinacy.

Training Process We collect the penultimate hidden states of the question x, denoted as the $h_\phi(x)$. We feed the $h_\phi(x)$ to the DetBlock to predict the determinacy score τ by minimizing the Mean Square Error (MSE) loss as follows:

$$\hat{\tau} = \text{DetBlock}(h_\phi(x)), \quad (4)$$

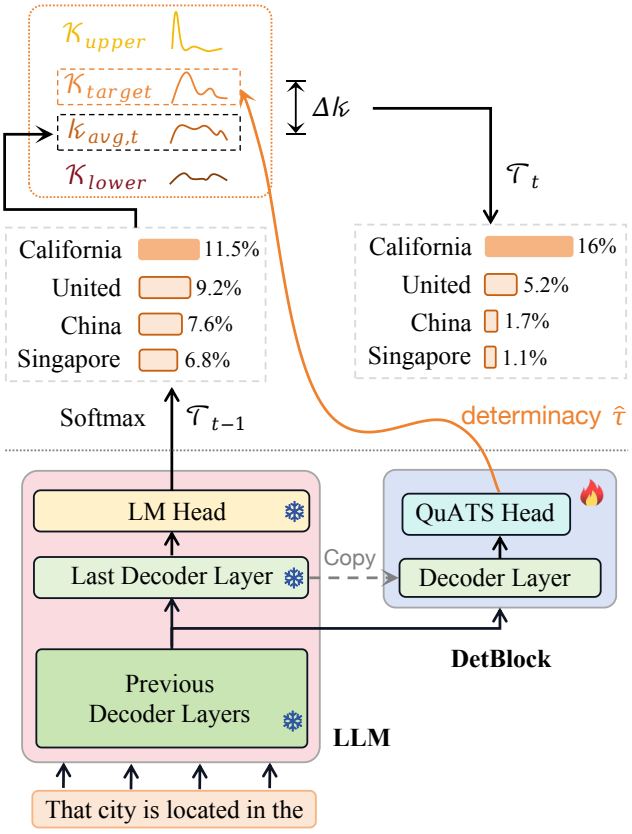

Figure 3: The overview of the QuATS.

Algorithm 1 QuATS in the inference

Input: hidden states $h_\phi(x)$, output logits $l_{\phi,t}(x, y_{<t})$, kurtosis mean $\mathcal{K}_{avg}$ and std $\mathcal{K}_{std}$
Output: answer sequence y

$\hat{\tau} = \text{DetBlock}(h_\phi(x))$
$\mathcal{K}_{upper} = \mathcal{K}_{avg} + \lambda \cdot \mathcal{K}_{std}$,
$\mathcal{K}_{lower} = \mathcal{K}_{avg} - \lambda \cdot \mathcal{K}_{std}$
$\mathcal{K}_{target} = \hat{\tau} \cdot (\mathcal{K}_{upper} - \mathcal{K}_{lower}) + \mathcal{K}_{lower}$
$t = 1, \mathcal{T}_0 = 1.0, y = [\,]$
repeat
  $\hat{p_\phi}(\hat{y}_t|x, y_{<t}) = \text{Softmax}\left(\frac{l_{\phi,t}(x,y_{<t})}{\mathcal{T}_{t-1}}\right)$
  $\kappa_t = \frac{\frac{1}{n}\sum_{i=1}^{n}(p_i-\overline{p})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(p_i-\overline{p})^2\right)^2} - 3$
  $\kappa_{avg,t} = \frac{1}{t}\sum_{i=1}^{t}\kappa_i$
  $\hat{\mathcal{T}}_t = 1 + \eta \cdot (\kappa_{avg,t} - \mathcal{K}_{target})t$
  $\hat{\mathcal{T}}_t = \text{Clamp}(\hat{\mathcal{T}}_t, \mathcal{T}_{min}, \mathcal{T}_{max})$
  $p_\phi(\hat{y}_t|x, y_{<t}) = \text{Softmax}\left(\frac{l_{\phi,t}(x,y_{<t})}{\mathcal{T}_t}\right)$
  $\hat{y}_t = \text{Sample}(p_\phi(\hat{y}_t|x, y_{<t}))$
  $y = \text{Append}(y,\ \hat{y}_t)$
  $t = t + 1$
until $\hat{y}_t == <|\text{endoftext}|>$
return y

$$\mathcal{L}_{QuATS}(\phi) = \frac{1}{2}(\tau - \hat{\tau})^2. \quad (5)$$

During the training of DetBlock, we freeze the weights of the LLM, preventing the original model from being affected.

Besides that, we need to record the mean and standard deviation of the kurtosis of the output distributions during training, denoted as $\mathcal{K}_{avg}$ and $\mathcal{K}_{std}$. We record these values for the inference later. We calculate the $\mathcal{K}_{avg}$ and $\mathcal{K}_{std}$ using the exponential moving average as follows:

$$\begin{aligned} \mathcal{K}_{avg,s} &= \beta \cdot \mathcal{K}_{avg,s-1} + (1-\beta) \cdot \hat{\mathcal{K}}_{avg,s}, \\ \mathcal{K}_{std,s} &= \beta \cdot \mathcal{K}_{std,s-1} + (1-\beta) \cdot \hat{\mathcal{K}}_{std,s}, \end{aligned} \quad (6)$$

where the $\hat{\mathcal{K}}_{avg,s}$ and $\hat{\mathcal{K}}_{std,s}$ are calculated by averaging the means and standard deviations of kurtosis of the whole batch at training step s.

3.3 Inference with QuATS

Before sampling the next token in the inference, we use DetBlock to predict the determinacy score $\hat{\tau}$ in Eq 4 from the input question. If the determinacy score is large, it means the LLMs are required to be more deterministic to answer this question. The prediction of the determinacy score will be done only once at the start of the generation.

Though we can rescale the determinacy score to get the temperature, it is noted that predicting temperature in this way does not take into account the intrinsic question awareness of LLMs. Based on the question awareness evaluation in Sec 2, we observe that LLMs have fundamental question awareness in some cases, which means some output distributions are steep/flat enough to give a deterministic/creative answer. If we directly change the sampling temperature, it may lead to overcorrection. Therefore, to avoid overcorrection, we propose QuATS to dynamically adjust the sampling temperature of every decoded token based on both the determinacy score and output distributions.

To implement QuATS in the inference, we calculate three things step by step: (1) target kurtosis $\mathcal{K}_{target}$, (2) current average kurtosis of the answer κ_{avg}, and finally (3) estimated temperature $\mathcal{T}$. We predict the temperature for every token to be decoded by projecting κ_{avg} to $\mathcal{K}_{target}$.

Target Kurtosis We want to correct the output distribution to be steeper or flatter according to

the question. Therefore, we have to find out the target kurtosis we want the distribution to have. The target kurtosis takes the value from the kurtosis interval $[\mathcal{K}_{lower},\ \mathcal{K}_{upper}]$ as follows:

$$\begin{aligned} \mathcal{K}_{upper} &= \mathcal{K}_{avg} + \lambda \cdot \mathcal{K}_{std}, \\ \mathcal{K}_{lower} &= \mathcal{K}_{avg} - \lambda \cdot \mathcal{K}_{std}, \end{aligned} \tag{7}$$

where the $\mathcal{K}_{avg}$ and $\mathcal{K}_{std}$ are recorded in Eq 6 when training the DetBlock. The kurtosis interval represents the range that the kurtosis of the model output distribution can commonly reach. According to the kurtosis interval, we use the predicted determinacy score $\hat{\tau}$ from DetBlock to calculate a target kurtosis $\mathcal{K}_{target}$ proportionately from the interval as follows:

$$\mathcal{K}_{target} = \hat{\tau} \cdot (\mathcal{K}_{upper} - \mathcal{K}_{lower}) + \mathcal{K}_{lower}, \tag{8}$$

The target kurtosis $\mathcal{K}_{target}$ lies in the kurtosis interval with $0 \leq \hat{\tau} \leq 1$. It constrains the range of the kurtosis of adjusted output distributions, which avoids overcorrection that the adjusted distributions are too steep or too flat.

Current Average Kurtosis Our next goal is to calculate the current average kurtosis of the answer so that we can know the starting point to be projected to target kurtosis. We use the mean of the kurtosises of the decoded token distributions to represent this kurtosis:

$$\kappa_{avg,t} = \frac{1}{t} \sum_{i=1}^{t} \kappa_i, \tag{9}$$

The $\kappa_{avg,t}$ is a running mean which is updated as the number of decoded tokens increases. We use the running mean to approximate it because we can not know the kurtosis of the whole output distribution before generation ends. Therefore, as the step t increases, the running mean $\kappa_{avg,t}$ will be approximate to the true average kurtosis of the whole answer distribution.

Estimated Temperature By changing the temperature of the Softmax function, we can adjust the distribution to project the average kurtosis $\kappa_{avg,t}$ of the answer to the target kurtosis $\mathcal{K}_{target}$. For the generation step t, we estimate the temperature as follows:

$$\hat{\mathcal{T}}_t = 1 + \eta \cdot (\kappa_{avg,t} - \mathcal{K}_{target})t, \tag{10}$$

$$\hat{\mathcal{T}}_t = \mathrm{Clamp}(\hat{\mathcal{T}}_t, \mathcal{T}_{min}, \mathcal{T}_{max}). \tag{11}$$

In Eq 10, the temperature in QuATS is decided by three factors: (1) the difference between $\kappa_{avg,t}$ and $\mathcal{K}_{target}$, (2) the generation step t, (3) a coefficient η to control the adjustment speed. For the first factor, if $\kappa_{avg,t} > \mathcal{K}_{target}$, it means the current average kurtosis is higher than the target kurtosis, thus we need to increase the temperature to flatten them, and vice versa. For the second factor, as the generation step t increases, the $\kappa_{avg,t}$ tends to approach the true average kurtosis of the whole answer. Thus the $(\kappa_{avg,t} - \mathcal{K}_{target})$ should exert a greater impact on the temperature adjustment. We need to clamp the temperature between an interval to avoid being too high or too low in Eq 11.

4 Experiment

In this section, we conduct experiments to showcase that QuATS can adaptively adjust the temperature according to various questions and greatly improve the model performance.

4.1 Training Setup of DetBlock

To train DetBlock, we collect 2.5k high-quality dialogues of different question types from the ShareGPT dataset(Dom Eccleston, 2023). We label the questions with the determinacy scores according to how deterministic the answers should be. We rate the questions for 4 levels from most creative (level 1) to most deterministic (level 4). We rescale the level score to (0, 1) as the final determinacy score.

We train DetBlock based on LLaMA 2-Chat 7b/13b/70b (Touvron et al., 2023) and Falcon-instruct 7b/40b (Penedo et al., 2023). We train for 2 epochs on the training dataset with a batch size of 32 on the 7b models with a learning rate of 2e-5, the 13b model with 1e-5, and the 40b/70b models with 5e-6.

4.2 Evaluation Setup

To verify the effectiveness of the QuATS, we use our question awareness evaluation dataset in Section 2 to evaluate if the LLMs with QuATS have a better awareness of different question types and better performance than the previous evaluation. Besides that, we choose two LLM benchmarks, namely AlpacaEval (Li et al., 2023) and MT-Bench (Zheng et al., 2023). These two benchmarks test if the models with QuATS can handle conversations of different scenarios. We set the model with a sampling temperature of 1 as the baseline.

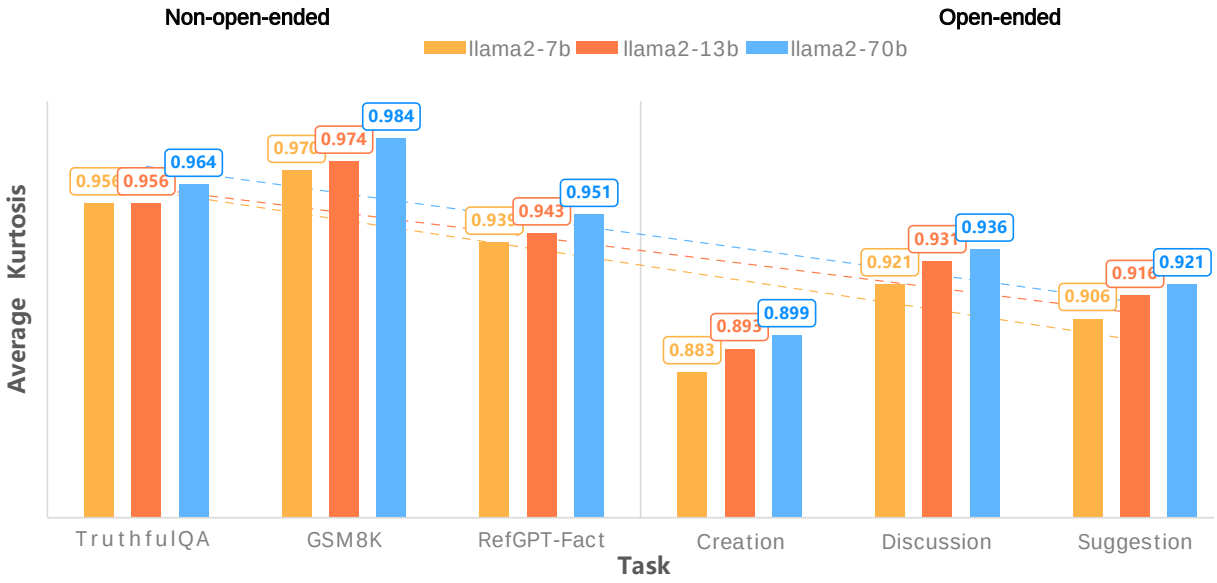

Figure 4: The result of question awareness evaluation of different LLMs using the QuATS.

Table 1: Evaluating the performance of LLMs using QuATS on various benchmarks. Acc represents the accuracy and Sco represents the score (1 to 10).

Model	Non-open-ended			Open-ended			Conversation	
	TruthfulQA Acc	**GSM8K** Acc	**RefGPT-Fact** Acc	**Creation** Sco	**Discussion** Sco	**Suggestion** Sco	**AlpacaEval** Sco	**MT-Bench** Sco
LLaMA 2 7b	50.0	21.0	51.0	**9.19**	9.35	9.40	8.51	6.88
+ QuATS	**55.0**	**29.0**	**56.0**	9.07	**9.35**	**9.43**	**8.71**	**7.19**
LLaMA 2 13b	62.0	43.0	58.0	9.22	9.26	9.50	8.81	7.43
+ QuATS	**63.0**	**46.0**	**60.0**	**9.25**	**9.30**	**9.52**	**8.96**	**7.56**
LLaMA 2 70b	59.0	**62.0**	66.0	**9.33**	9.48	9.49	9.20	7.78
+ QuATS	**61.0**	61.0	**68.0**	9.29	**9.50**	**9.52**	**9.24**	**7.83**
Falcon 7b	26.0	2.0	28.0	6.21	6.28	6.61	5.45	4.50
+ QuATS	**32.0**	**2.0**	**33.0**	**6.41**	**6.58**	**6.72**	**5.82**	**5.11**
Falcon 40b	50.0	13.0	46.0	7.33	7.91	**8.21**	7.26	6.30
+ QuATS	**53.0**	**15.0**	**50.0**	**7.57**	**8.01**	8.16	**7.42**	**6.59**

4.3 Results and Analysis

From Figure 4, we evaluate the question awareness of the LLaMA 2 models using QuATS. The descending trend lines have shown a distinction in the awareness between the non-open-ended questions and open-ended questions. The models with QuATS choose to be more deterministic in answering the non-open-ended questions, thus we can observe higher kurtosises in non-open-ended tasks. Similar findings can be observed in open-ended questions.

In table 1, we can see that QuATS largely improves the LLM performance in the various tasks, especially in the non-open-ended questions. It means that a better awareness of non-open-ended questions can alleviate the hallucination.

For the results of two comprehensive LLM benchmarks, both LLaMA 2 and Falcon have significant improvements over the baselines, which shows the QuATS is useful for different models with different sizes. We observe that smaller models like LLaMA 7b and Falcon 7b have more performance gains than larger models. It can be inferred that the distribution of larger models originally has more appropriate tokens with high probabilities thus the effectiveness of additional adjustment on the steepness of the distribution tends to be smaller.

4.4 Ablation Study

We conduct the ablation study to compare QuATS with baselines with different sampling temperatures on the LLaMA 2-Chat 13B. As shown in Figure 5, QuATS consistently outperformed the naive

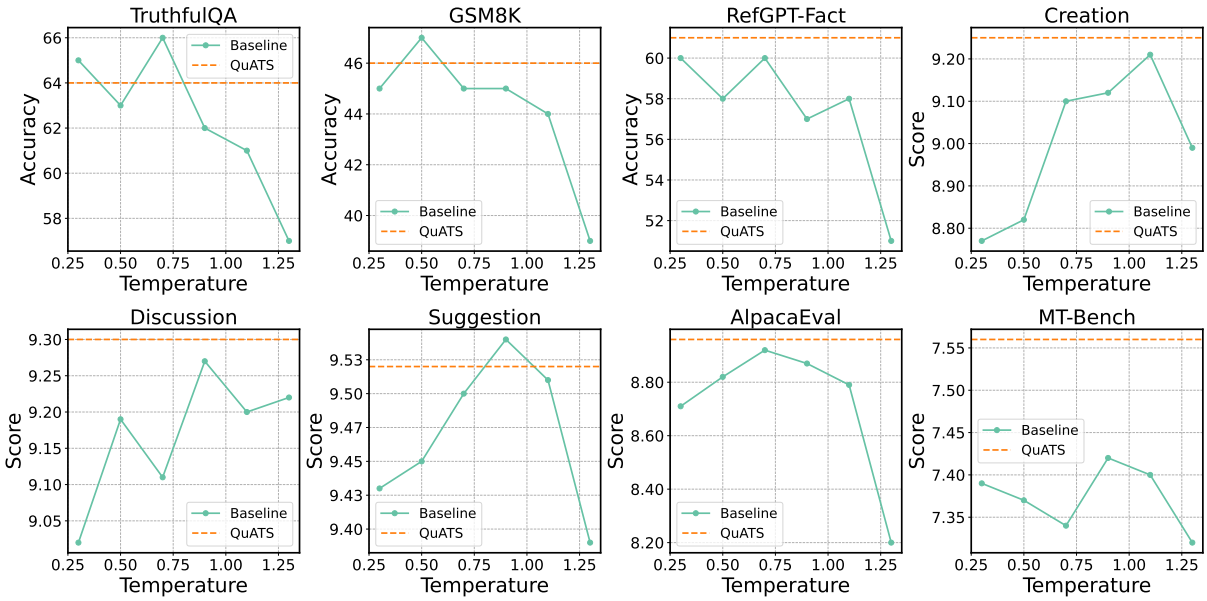

Figure 5: Comparison between QuATS and baselines with different sampling temperatures.

temperature sampling with different temperatures on these benchmarks.

5 Related Work

Controlling text generation in LLMs has seen significant advancements in recent years. Sampling methods play a crucial role in controlling the output quality and diversity of generated text. We introduce temperature sampling and corresponding advanced techniques in text generation.

Temperature Sampling Greedy sampling selects the token with the highest predicted probability, resulting in deterministic and often repetitive text. Random sampling selects tokens based on the probabilities, introducing randomness to alleviate the repetition. We can further adjust the temperature in the Softmax function to control the token probabilities. Temperature sampling can be seen as the trade-off between creativity and determinacy in the generated text. Our QuATS adaptively controls the steepness of output distributions by adjusting the temperature.

Post-processing Techniques Because the tokens with higher probabilities are probably appropriate choices, we can choose only to select these tokens, avoiding sampling nonsensical tokens. Top-k sampling (Fan et al., 2018) narrows down the token selection to the top-k most probable tokens, increasing the likelihood of coherent text and balancing diversity and quality. Similar to the motivation of top-k sampling, nucleus sampling (Holtzman et al., 2020), also known as top-p sampling, dynamically selects the top-p fraction of tokens with the highest probabilities. Locally typical sampling (Meister et al., 2023) posits the abstraction of natural language generation as a discrete stochastic process and samples tokens according to conditional entropy. Entmax sampling (Martins et al., 2020) leverages entmax transformation to train and sample from a natively sparse language model. Keyword-based sampling (au2 and Akhtar, 2023) uses knowledge distillation techniques to extract keywords and samples using these extracted keywords. It is noted that these post-processing techniques are compatible with QuATS because QuATS only adjusts the output distribution itself.

6 Conclusion

In this paper, we highlight the question awareness of LLMs, which receives little attention from previous studies. While LLMs exhibit a fundamental awareness of open-ended and non-open-ended questions, they do falter in certain domains, often leading to casual or inaccurate responses. To bridge the gap, we introduce Question Awareness Temperature Sampling (QuATS), enabling LLMs to autonomously adapt their response determinacy based on question type. Our experiments showcased the efficacy of QuATS, significantly enhancing LLM performance across various benchmarks.

Limitations

In this paper, we explore the question awareness of LLMs from the perspective of output distributions and enhance this ability by adjusting the sampling temperature. However, the question awareness should be the intrinsic ability that the model should have. QuATS improves this ability only by extrinsic force but does not influence the model itself.

We believe the question awareness of LLMs is a valuable subject. How to improve the intrinsic question awareness of LLMs is worthy of exploration for future work.

References

Anthropic. 2023. Introducing claude. https://www.anthropic.com/index/introducing-claude.

Jyothir S V au2 and Zuhaib Akhtar. 2023. Keyword based sampling (keys) for large language models.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems.

Steven Tey Dom Eccleston. 2023. Share your wildest chatgpt conversations with one click. https://github.com/domeccleston/sharegpt.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration.

Alex Lamb, Anirudh Goyal, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2020. Sparse text generation.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions.

Dongjie Yang, Ruifeng Yuan, YuanTao Fan, YiFei Yang, Zili Wang, Shusen Wang, and Hai Zhao. 2023. Refgpt: Reference -> truthful customized dialogues generation by gpts and for gpts.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.