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

Dongjie Yang, Hai Zhao*
Department of Computer Science and Engineering, Shanghai Jiao Tong University,
Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University,
Shanghai Key Laboratory of Trusted Data Circulation and Governance in Web3
djyang.tony@sjtu.edu.cn,zhaohai@cs.sjtu.edu.cn

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 need to respond more deterministically but some do not. We refer to this as *question awareness* of LLMs. The lack of question awareness in LLMs leads to two phenomena that LLMs are: (1) too casual to answer non-open-ended questions or (2) too boring to answer open-ended questions. In this paper, we first evaluate the question awareness in LLMs. The experimental results show that LLMs have the issues of lacking awareness of questions in certain domains, e.g. factual knowledge, resulting in hallucinations during the generation. To mitigate these, we propose a method called Question Awareness Temperature Sampling (QuATS). This method enhances the question awareness of LLMs by adaptively adjusting the output distributions based on question features. The automatic adjustment in QuATS eliminates the need for manual temperature tuning in text generation and consistently improves model performance in various benchmarks.

1 Introduction

Large language models (LLMs) (OpenAI, 2022, 2023; Anthropic, 2023; Jiang et al., 2023; Bai et al., 2023; Team et al., 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 bases, it is hard to tell if LLMs are aware of the difference between the 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 deterministic answers? We refer to this as *question awareness* of LLMs that one knows which type of questions requires deterministic answers and which does not. It is significant to explore the question awareness of LLMs because it has a deep relationship to the model hallucinations that LLMs are prone to generate inaccurate content when they are not sure.

In this paper, we explore whether LLMs have question awareness on different types of questions. Because LLMs sample next tokens from output distributions, as shown in Figure 1, we 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 selecting which token in the vocabulary to be the next token and a flat one means the model does not have a clear preference for the next token. Therefore, the steepness of the output distributions reflects the question awareness by indicating determinacy about the generated answers. We utilize the kurtosis to measure the steepness of the distribution and investigate the question awareness by checking kurtosises of output distributions when LLMs are asked different types of questions. We evaluate LLaMA 2 (Touvron et al., 2023) and Falcon (Penedo et al., 2023) on different types of non-open-ended/open-ended questions for question awareness evaluation. Experimental results show that LLMs have a certain degree of question awareness but lack the awareness in some scenarios, e.g., factual knowledge, thus easily giving more casual and hallucinated answers.

As the steepness of output distributions reflects the question awareness, we utilize the temperature of the Softmax function (Bridle, 1989) to adjust

* Corresponding author; This paper was partially supported by Joint Research Project of Yangtze River Delta Science and Technology Innovation Community (No. 2022CSJGG1400).

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 2142–2152
November 12-16, 2024

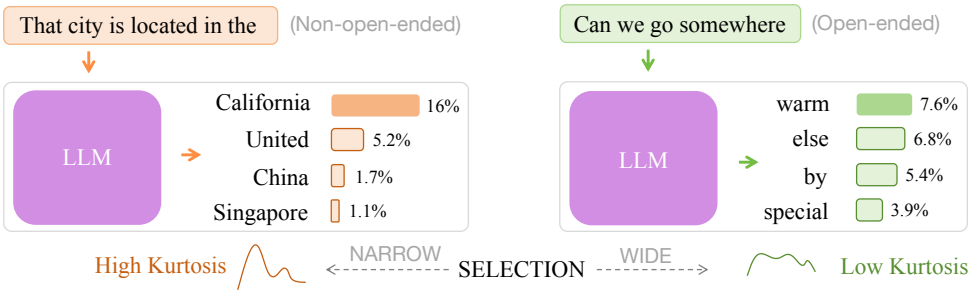

Figure 1: LLMs should choose to be deterministic to answer the question on the left but can have more choices to answer the one on the right.

the steepness to externally change the question awareness. We evaluate the model with different temperatures to explore the influence of question awareness on model performance. The results show a relatively lower temperature (steeper distribution) makes the model more deterministic and have better performance on non-open-ended questions.

Inspired by the adjustment of temperature on the question awareness, we propose Question Awareness Temperature Sampling (QuATS), a method that enhances question awareness of LLMs by adjusting the output distributions through the temperature. When facing different questions, LLMs choose to be more deterministic or not using an adaptive temperature strategy of QuATS, avoiding the tedious process of temperature tuning in the text generation. To sum up, our contributions are stated as follows:

- We evaluate the question awareness in 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 temperature without manual tuning.
- Our experimental results show that the QuATS enhances the question awareness of LLMs and consistently improves the model performance on various benchmarks.

2 Question Awareness Evaluation

In this section, we evaluate the question awareness of LLMs and how it influences the model performance on downstream tasks.

2.1 Formulation of the Next Token Prediction

To better clarify the question awareness, we first give a formulation of the next token prediction in the text generation. For an auto-regressive language model, denoted as ϕ, given a question x, we can calculate the output distribution of the next token $\hat{y}_t$ as follows:

$$p_\phi(\hat{y}_t|x, y_{<t}) = \text{Softmax}\left(\frac{l_{\phi,t}(x, y_{<t})}{\mathcal{T}}\right), \quad (1)$$

where $l_{\phi,t}(x, y_{<t})$ is the output logit of the token at the step t and $\mathcal{T}$ is the temperature of the Softmax function (Bridle, 1989). We sample from the output distribution $p_\phi(\hat{y}_t|x, y_{<t})$ to generate the next token. For the temperature $\mathcal{T}$, we can consider the original Softmax function without $\mathcal{T}$ as the Softmax function with a temperature of 1. As shown in Figure 1, if we sample the next token with a lower temperature, the output distribution will get steeper thus more likely sampling the token with a large probability. Therefore, the temperature influences the kurtosis of the output distributions and externally changes the question awareness of LLMs. In common practice, we tune the temperature, which is a hyperparameter, to decide how deterministic LLMs should be to answer the question. We usually select a fixed temperature and will not frequently change the value because it is tedious to tune for an optimal temperature for every question.

2.2 Metric

The steepness of the next-token distribution, $p_\phi(\hat{y}_t|x, y_{<t}) = (p_1, p_2, \ldots, p_n)$, where n stands for vocabulary size, indicates how deterministic the LLMs are, reflecting the question awareness. To measure the steepness, we introduce kurtosis as the metric. If the distribution is steeper, the model is more deterministic on this generated token

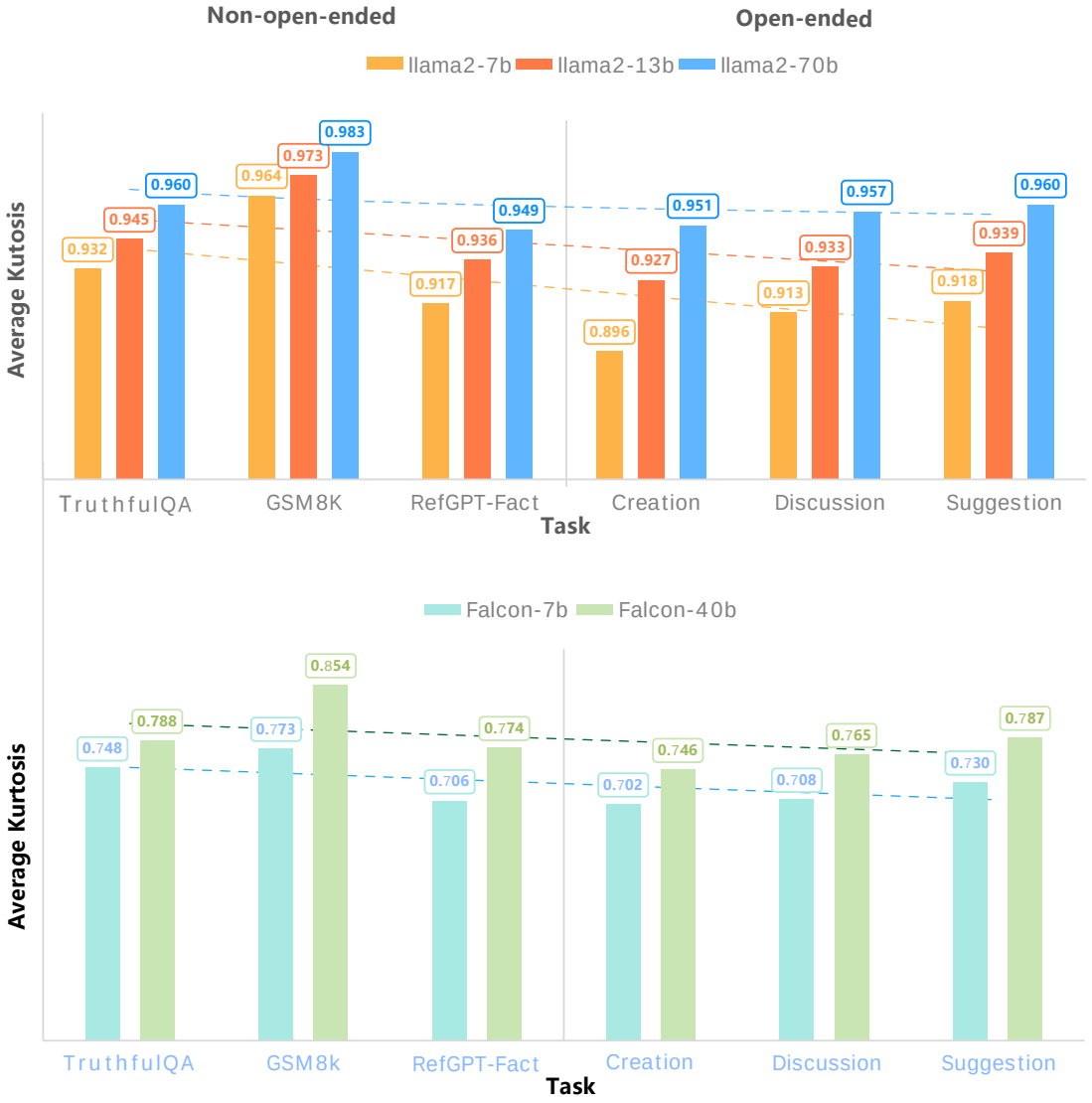

Figure 2: The result of question awareness evaluation. The dotted lines are the trend lines of the kurtosises, which are linearly fitted.

and the kurtosis gets larger. We use the average kurtosis of the distributions of all answer tokens to reflect the general determinacy of the answer. **To simplify the illustration, in this paper, we consider the average kurtosis and question awareness to be the same thing.** We calculate the average kurtosis $\mathcal{K}$ as follows:

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

where κ_t is the kurtosis of the distribution of the token at step t. For the discrete distribution, the value of kurtosis is related to the value n. As the vocabulary sizes of LLMs are different, we have to normalize the kurtosis for fair comparison. Because the one-hot distribution is the steepest and has the largest kurtosis, we divide the kurtosis by $\kappa_{\text{one-hot}}$ to normalize the kurtosis to $(0, 1)$.

2.3 Evaluation Process

We first evaluate the essential question awareness in LLMs by calculating the average kurtosises with the default temperature of 1. We then explore the influence of question awareness on the performance by externally adjusting average kurtosises (question awareness) using different temperatures.

To evaluate question awareness, we 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 ques-

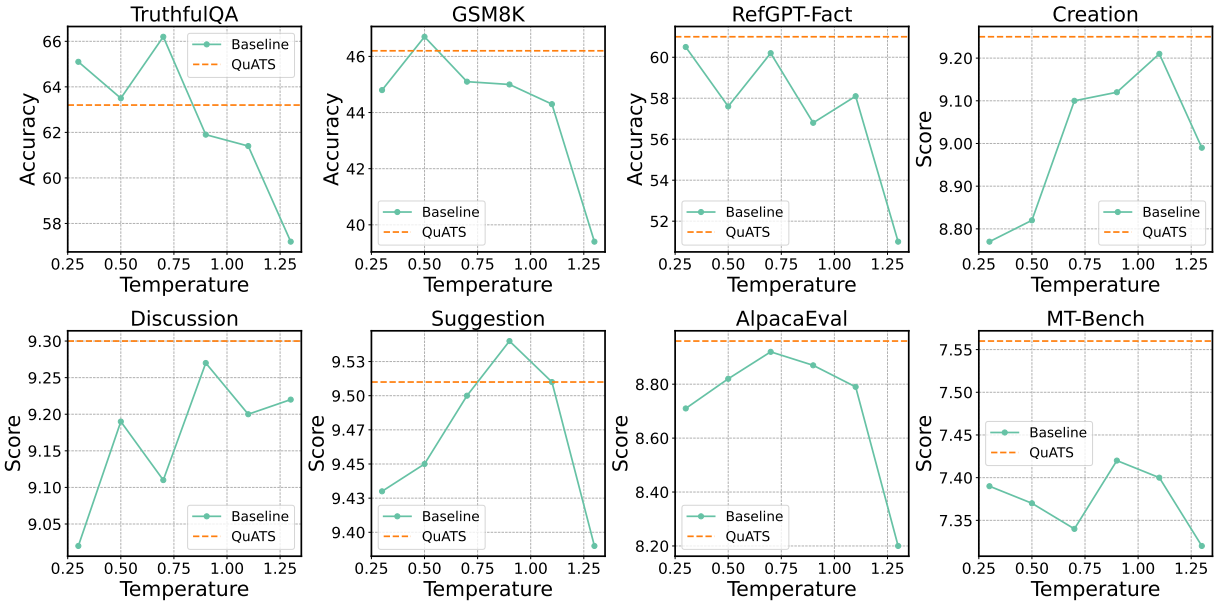

Figure 3: Comparison between QuATS and baselines with different fixed temperatures using LLaMA 2-Chat 13B (Touvron et al., 2023) on downstream tasks. The temperatures adjust the kurtosises, which influence the performance in open-ended and non-open-ended questions differently. In contrast, the adaptive temperature strategy of QuATS consistently outperforms temperature sampling with fixed temperatures.

tions. We collect three types of non-open-ended questions that have only fixed/limited answers: (1) TruthfulQA (Lin et al., 2022): questions about commonsense knowledge. (2) GSM8K (Cobbe et al., 2021): school math word problems. (3) RefGPT-Fact (Yang et al., 2023): questions about world knowledge, including 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 writing 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 are carefully selected by humans from ShareGPT dataset (Dom Eccleston, 2023). We evaluate chat models with different sizes, including LLaMA 2-Chat 7B/13B/70B (Touvron et al., 2023), Falcon-instruct 7B/40B (Penedo et al., 2023). It is noted that we do not evaluate closed-source models like GPT-4 (OpenAI, 2023) because we can not obtain the output distributions from the APIs.

To investigate the influence of question awareness on the performance, we evaluate the performance of LLaMA 2-Chat 13B (Touvron et al., 2023) on the evaluation dataset using different temperatures for generation. Details about the metric of open-ended questions in Figure 3 are introduced in Sec 4.1.

2.4 Results and Analysis

LLMs lack a strong sense of question awareness. In Figure 2, according to the trend lines, the kurtosises of non-open-ended questions are not significantly higher than the ones of open-ended questions in most models. For non-open-ended questions, LLMs have fundamental question awareness, e.g., 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 sometimes struggle to recognize questions about world knowledge that require careful answers, thus easily 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.

Question awareness greatly affects model performance. In Figure 3, for the non-open-ended questions, the results (green lines) show that the model has better performance with relatively low temperatures (steeper distributions) and the performance decreases as the temperatures get higher. It indicates LLM is not determinant enough

(with a default temperature of 1) and lacks question awareness essentially. Therefore, if we increase the steepness with a lower temperature, it improves the performance for non-open-ended questions. Opposite conclusions for open-ended questions can be also observed.

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 answers.

3 Question Awareness Temperature Sampling

Based on the findings above, we propose the Question Awareness Temperature Sampling (QuATS) to enhance the question awareness of LLMs by an adaptive temperature strategy, which greatly improves the model performance.

3.1 Training A DetBlock to Predict Determinacy

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. 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 LLMs should be based on the given questions. After DetBlock is ready, we convert the predicted determinacy score to the 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 determinacy scores based on the artificial criteria. To be specific, we rate open-ended questions requiring less determinacy with lower determinacy scores and vice versa. We use the questions as the input and determinacy scores as training labels.

DetBlock Structure As shown in Figure 4, 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:

$$\hat{\tau} = \text{DetBlock}(h_\phi(x)), \tag{3}$$

$$\mathcal{L}_{QuATS}(\phi) = \frac{1}{2}(\tau - \hat{\tau})^2. \tag{4}$$

During the training of DetBlock, we **freeze** the weights of the LLM so that the performance of the original model will not be 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:

$$\begin{aligned} \mathcal{K}_{avg,s} &= \beta \cdot \mathcal{K}_{avg,s-1} + (1-\beta) \cdot \hat{\mathcal{K}}_{avg,s}, \\ \mathcal{K}_{std,s} &= \beta \cdot \mathcal{K}_{std,s-1} + (1-\beta) \cdot \hat{\mathcal{K}}_{std,s}, \end{aligned} \tag{5}$$

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.

3.2 Inference with QuATS

Before sampling the next token in the inference, we use DetBlock to predict the determinacy score $\hat{\tau}$ in Eq 3 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 temperature, it may lead to overcorrection. Therefore, to avoid overcorrection, we propose QuATS to dynamically adjust the temperature of every decoded token based on both the determinacy score and original 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

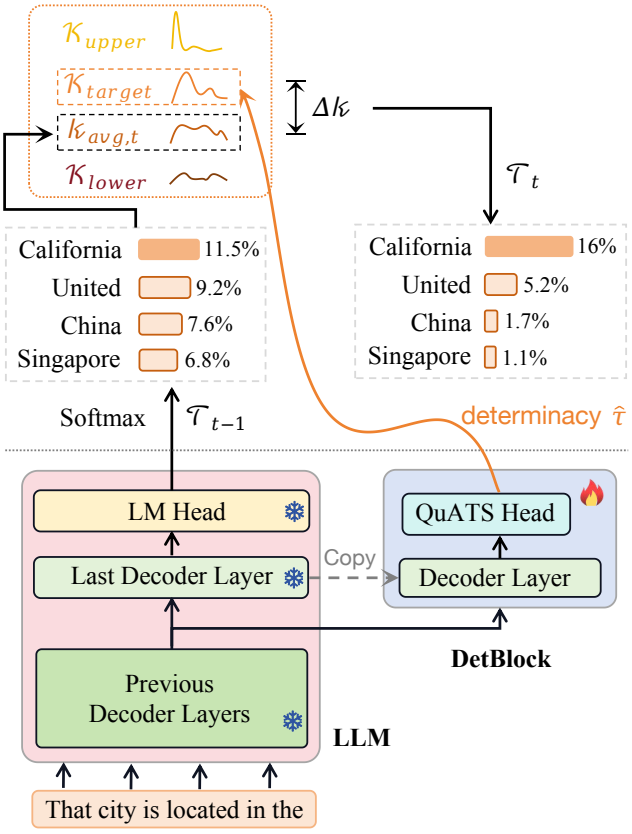

Figure 4: The overview of the QuATS.

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

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:

$$\begin{aligned} \mathcal{K}_{upper} &= \mathcal{K}_{avg} + \lambda \cdot \mathcal{K}_{std}, \\ \mathcal{K}_{lower} &= \mathcal{K}_{avg} - \lambda \cdot \mathcal{K}_{std}, \end{aligned} \tag{6}$$

where the $\mathcal{K}_{avg}$ and $\mathcal{K}_{std}$ are recorded in Eq 5 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:

$$\mathcal{K}_{target} = \hat{\tau} \cdot (\mathcal{K}_{upper} - \mathcal{K}_{lower}) + \mathcal{K}_{lower}, \tag{7}$$

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:

$$\kappa_{avg,t} = \frac{1}{t}\sum_{i=1}^{t}\kappa_i, \tag{8}$$

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

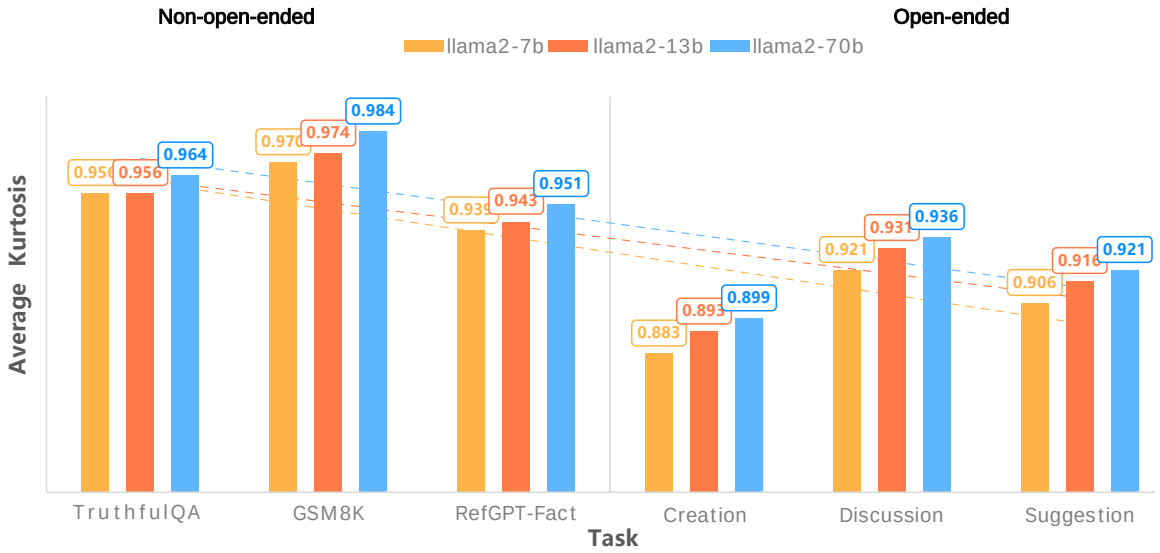

Figure 5: The result of question awareness evaluation of LLaMA 2-Chat models using the QuATS.

Table 1: Evaluating the performance of LLMs using QuATS on various benchmarks. Acc represents the accuracy and Sco represents the score, which is rated according to LLM-as-a-judge in MT-Bench (Zheng et al., 2023).

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	50.1	21.0	51.1	**9.19**	9.35	9.40	8.57	6.88
+ QuATS	**55.3**	**29.8**	**56.3**	9.07	**9.35**	**9.43**	**8.71**	**7.19**
LLaMA 2 13B	62.4	43.0	58.4	9.22	9.26	**9.55**	8.81	7.43
+ QuATS	**63.2**	**46.2**	**61.0**	**9.25**	**9.30**	9.51	**8.96**	**7.56**
LLaMA 2 70B	59.2	**62.7**	66.1	**9.33**	9.48	9.49	9.20	7.78
+ QuATS	**61.9**	61.5	**68.5**	9.29	**9.51**	**9.52**	**9.24**	**7.83**
Falcon 7B	26.1	2.1	28.8	6.21	6.28	6.61	5.45	4.50
+ QuATS	**32.7**	**2.9**	**33.4**	**6.41**	**6.54**	**6.72**	**5.82**	**5.11**
Falcon 40B	50.2	13.6	46.2	7.33	7.91	**8.21**	7.26	6.30
+ QuATS	**53.0**	**15.3**	**50.3**	**7.57**	**8.06**	8.16	**7.42**	**6.59**

generation step t, we estimate the temperature as follows:

$$\hat{\mathcal{T}}_t = 1 + \eta \cdot (\kappa_{avg,t} - \mathcal{K}_{target})t, \quad (9)$$

$$\hat{\mathcal{T}}_t = \text{Clamp}(\hat{\mathcal{T}}_t, \mathcal{T}_{min}, \mathcal{T}_{max}). \quad (10)$$

In Eq 9, 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 10.

4 Experiment

In this section, we conduct experiments to showcase that QuATS can enhance question awareness using the adaptive temperature strategy and consistently improve the model performance.

4.1 Evaluation Setup

To verify the effectiveness of the QuATS, we evaluate if LLMs with QuATS have a better awareness of different question types and better performance on our question awareness evaluation dataset in Sec 2. Besides that, we choose two LLM benchmarks, namely AlpacaEval (Li et al., 2023) and MT-Bench (Zheng et al., 2023), which test if the models with QuATS can handle conversations

of different scenarios. We set models with a temperature of 1 as the baselines. To check the answers to open-ended questions, we follow the official implementation of LLM-as-a-judge from the MT-Bench[1] and use GPT-4 turbo as a judge to score 1 to 10 for the answers, as shown in Table 1.

4.2 Results and Analysis

From Figure 5, we evaluate the question awareness of LLaMA 2-Chat models using QuATS. Compared to the ones in Figure 2, the descending trend lines have shown a distinction in the awareness between non-open-ended and open-ended questions. The models with QuATS choose to be more deterministic with higher kurtosises in answering non-open-ended questions. Similar findings can be observed in open-ended questions.

For model performance, 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, MT-Bench and AlpacaEval, both LLaMA 2 and Falcon have significant improvements over the baselines, which show the QuATS is useful for different models with different sizes on open-domain conversations. We observe that smaller models like LLaMA 2 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.

In Figure 3, we also compare the performance of the model using QuATS with baselines using different fixed temperatures. QuATS consistently outperforms the naive temperature sampling with different fixed temperatures on these tasks.

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 (Bridle, 1989) 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, which can be further post-processed.

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.

[1] https://github.com/lm-sys/FastChat/tree/main/fastchat

Limitations

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

We believe the question awareness of LLMs is a valuable subject, providing a new perspective of hallucinations in LLMs. How to improve the intrinsic question awareness to reduce the hallucinations 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.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report.

John Bridle. 1989. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2.

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.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, LÃl'lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, TimothÃl'e Lacroix, and William El Sayed. 2023. Mistral 7b.

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 AndrÃl' 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.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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: Dialogue generation of gpt, by gpt, and for gpt.

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.

A DetBlock Training Details

To train a DetBlock, we collect a training dataset consisting of 3.5K high-quality questions, which are carefully selected by humans from the ShareGPT dataset (Dom Eccleston, 2023). Using the criteria shown in Table 2, we manually label the questions with the determinacy scores according to how deterministic the answers should be. The questions are rated by two persons separately from not very deterministic (1 point) to highly deterministic (4 points). We average the two scores and rescale the averaged score to (0, 1) as the final determinacy score.

We train the 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 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.

Table 2: The criteria for rating the determinacy score in the training dataset.

Highly Deterministic (4 points)	Questions/instructions that have a unique answer, including mathematical calculations and factual knowledge.
Fairly Deterministic (3 points)	Questions/instructions related to logical reasoning, code modification and creation, text rewriting and summarization, text translation, reading comprehension.
Moderately Deterministic (2 points)	Questions/instructions related to code discussions and creative inquiries that require a certain level of expertise.
Not Very Deterministic (1 point)	Creative and open-ended questions/instructions (e.g., "What do you think about...?" "How do you see...?").