Self-Evaluation Improves Selective Generation in Large Language Models

Jie Ren*, **Yao Zhao***, **Tu Vu**†, **Peter J. Liu***, **Balaji Lakshminarayanan***
{jjren,yaozhaoyz,ttvu,peterjliu,balajiln}@google.com
Google DeepMind*, Google Research†

Abstract

Safe deployment of large language models (LLMs) may benefit from a reliable method for assessing their generated content to determine when to abstain or to *selectively generate*. While likelihood-based metrics such as perplexity are widely employed, recent research has demonstrated the limitations of using sequence-level probability estimates given by LLMs as reliable indicators of generation quality. Conversely, LLMs have demonstrated strong calibration at the token level, particularly when it comes to choosing correct answers in multiple-choice questions or evaluating true/false statements. In this work, we reformulate open-ended generation tasks into token-level prediction tasks, and leverage LLMs' superior calibration at the token level. We instruct an LLM to *self-evaluate* its answers, employing either a multi-way comparison or a point-wise evaluation approach, with the option to include a *"None of the above"* option to express the model's uncertainty explicitly. We benchmark a range of scoring methods based on self-evaluation and evaluate their performance in selective generation using TRUTHFULQA and TL;DR. Through experiments with PALM-2 and GPT-3, we demonstrate that self-evaluation based scores not only improve accuracy, but also correlate better with the overall quality of generated content.

1 Introduction

Large language models (LLMs) are often pre-trained on a vast corpus of text and then fine-tuned on supervised data to follow instructions [Devlin et al., 2018, Radford et al., 2018, Raffel et al., 2020, Adiwardana et al., 2020, Wei et al., 2021, Ouyang et al., 2022, Chung et al., 2022]. Having the ability to tell when a language model's output is trustworthy is important for safe deployment of language models. For example, the model's trustworthiness can be used as signal to *selectively generate* answers based on how confident the LLM is in the quality of its output.

Prior research has demonstrated that the distance to the training distribution in the embedding space predicts output quality for conditional generative models [Ren et al., 2023b]. Extending this work to large language models is challenging because their training distribution is too large to estimate and extracting embeddings from well-integrated LLM systems requires significant engineering effort.

Alternatively, a straightforward approach to estimating a language model's confidence in its output is to calculate the sequence probability or the length-normalized sequence probabilities [Adiwardana et al., 2020]. However, studies have shown that language models' sequence probabilities on open-ended generations do not reliably rank-order their outputs by quality [Liu et al., 2022, Ren et al., 2023b]. Human feedback can be used to fine-tune language models to better align with human-judged quality, such as with Reinforcement Learning from Human Feedback (RLHF) [Stiennon et al., 2020], SLiC-HF [Zhao et al., 2023] and DPO [Rafailov et al., 2023], resulting in better *quality-calibrated* models.

Since human feedback data is expensive to obtain, we explore leveraging the self-evaluation ability of LLMs to improve quality-calibration. Despite the poor calibration on sequence-level likelihood, recent work has shown that LLM token-level probability can be quite well-calibrated on choosing the

Figure 1: Demonstration of our approach.

correct option of multi-choice question answering and true/false questions [Kadavath et al., 2022, OpenAI, 2023, Robinson et al., 2022]. This suggests that evaluating language model's generation with token-level probabilities using an appropriate prompt format might be better for selective generation than sequence-level likelihood.

In this study, we focus on obtaining a con dence score that is quality-calibrated on free-form generation tasks. We propose reducing the sequence-level scoring problem to token-level scoring by designing different self-evaluation tasks and propose a variety of scores. We focus on evaluating model's quality-calibration for use in selective generation, and not just predictive accuracy. We show that our proposed con dence estimation signi cantly improves the quality calibration, and can be used to abstain poor quality outputs using the TRUTHFULQA and TL;DR benchmarks.

2 Methods

Background: sequence likelihood. Given a question x and an answer $y; y = y^1 y^2 : : : y^l$, we have sequence-level likelihood score,

$$\log p(y|x) = \sum_{t=1}^{l} \log p(y^t | y^1 : : : y^{t-1}; x): \qquad \text{(Sequence likelihood)}$$

Though $\log p(y|x)$ is statistically meaningful, it has been shown that it is biased towards sequence length, i.e. models tend to underestimate sequence likelihood of longer sentences [Wu et al., 2016]. The length normalized likelihood is an alternative score to use,

$$\log p(y|x) = \frac{1}{l} \sum_{t=1}^{l} \log p(y^t | y^1 : : : y^{t-1}; x): \qquad \text{(Length normalized sequence likelihood)}$$

Although sequence-level scores have weak predictive power, the previous results show that LLMs are well-calibrated on multiple choice question answer tasks and true/false evaluation tasks [Kadavath et al., 2022, OpenAI, 2023], suggesting the model has better calibration on token-level scores. Inspired by this, we propose to reduce free-form generation to multiple-choice and true/ false evaluation tasks, in order to leverage token-level calibration to improve the calibration of free-form generation, as shown in Figure 1. Ren et al. [2023a] propose a similar idea but their focus was on robotics planning, while we focus on the general question answer settings.

To convert free-form generation to multi-choice question answer task, we rst sample multiple candidate answers. For a given question x, we sample n answers $\{y_i\}; i = 1; : : : ; n$ from an LLM. We tried using a prompt to instruct the model to generate multiple different answers all at once, but the quality of the batch generated answers were not as good as sampling one at a time.

2.1 Sample and Select: reduce free-form generation to multi-choice question answer task

Given a question and a set of candidate answers $\{y_i\}_m$, we append alphabet characters $c_i = A; B; C; \ldots$, to the answers and form it into a multiple choice format. A straightforward score could be the softmax probability for the character $p(c_i|x; \{c_i\}_m)$, which was used in Ren et al. [2023a]. The selected answer would be the one with the highest softmax probability $y_r; r = \arg\max_i p(c_i|x; \{c_i\}_m)$. However, there are a few issues with that score:

Position bias The score could change as the position of the candidate answers change. See Figure 3 (left). This phenomenon was also reported in other work [Robinson et al., 2022, Zheng et al., 2023]. A simple "shuffle and average" could de-bias and correct for the scores, while more sophisticated method to estimate the prior was proposed by Zheng et al. [2023]. In our work, we use the simple shuffle and average de-bias method. The ablation study of the effect of position bias is in Table 4.

Probability dispersion among multiple true answers. Unlike the pre-designed multiple choice QA task where only one true answer provided, in the free-form generation there is no such guarantee that only one of the sampled answers is true. When more than one true answers are in the candidate list, the probability of the true is dispersed among the true answers, see Figure 3 (middle). This is an undesired property for comparing across questions, since different questions could generate different number of true answers. Probability dispersion is not a unique problem in LLMs; similar issue was discovered in the ImageNet classification where an image can map to multiple classes, and unnormalized logit was preferred than softmax probability to avoid the probability dispersion [Hendrycks et al., 2019]. Therefore we propose,

$$\log p(c_i|x; \{c_i\}_m); \; c = \{A; B; \ldots\}: \qquad \text{(Sample and Select)}$$

No answer is true It is possible that when the model does not know the answer, none of the sampled answers is true. If only wrong answers are provided, the model will be forced to choose one from them, resulting in over-confident prediction. See Figure 3 (right). To mitigate that, we add "NONE OF THE ABOVE" as an additional candidate answer to give model a chance to reject the sampled answers, $\{y\}_{+nota} = \{y\} \cup \{\text{nota}\}$. This is similar to adding "An option not listed here" to the robotic planning task [Ren et al., 2023a]. We obtain the score corresponding to the "NONE OF THE ABOVE" answer,

$$p(c_{nota}|x; \{c_i\}_{+nota}) \qquad \text{(Sample and Select w/ NONE OF THE ABOVE)}$$

A higher nota score indicates that the selected answer is less likely to be correct. So we use $p(c_{nota}|x; \{c_i\}_{+nota})$ as the confidence score of the selected answer $y_r = y_r; r = \arg\max_i p(c_i|x; \{c_i\}_m)$. Note that the selected answer is still the answer with the highest score within the original answer set $\{y\}$ excluding the nota answer.

2.2 Sample and Eval: reduce free-form generation to true/false evaluation task

We can also evaluate a question and an answer pair using pointwise evaluation format. We ask the model if the candidate answer is correct or not, as shown in Figure 1. Since the task is a binary classification task, we can normalize the output score using softmax function to a probability,

$$p(\text{Yes}|x; y_i): \qquad \text{(Sample and Eval)}$$

This is similar the P(True) proposed in [Kadavath et al., 2022]. They also propose to include candidate answers in the prompt,

$$p(\text{Yes}|x; y_i; \{y\}): \qquad \text{(Sample and Eval w/ other candidates)}$$

But that work focuses on the scaling law of the score's calibration, and did not compare it with sequence-level score and Sample and Select score.

2.3 Combining the best of both worlds: select the answer via multi-choice evaluation and score the selected answer via pointwise evaluation

Sample and Select and Sample and Eval have their own pros and cons. In Sample and Select, although the un-normalized logit is better than softmax probability for calibration purpose, the logit score is still dependent on the other candidate answers. For fairly comparing across $(x; y)$ pairs, a good score should measure the confidence to $(x; y)$ itself, not dependent on other candidate answers. Sample and Eval score $p(\text{Yes}|y_i; x)$ is indeed independent of other answers. On the other

hand, Sample and Select provides the opportunity for comparing different answers and select the best. Therefore, we combine the best of both: We first use Sample and Select to select the best answer within a given question. The answer with the highest softmax probability score is selected, $\hat{y} = y_r; r = \arg\max_i p(c_i|x; f_{cy}g)$. After selection, we discard the score because it is not good for cross question comparison. We score the selected answer via Sample and Eval, $p(\text{Yes}|x; \hat{y})$.

$$p(\text{Yes}|x; \hat{y}); \text{where} \hat{y} = y_r; r = \arg\max_i p(c_i|x; f_{cy}g): \quad \text{(Hybrid)}$$

In the case where NONE OF THE ABOVE answer is added, we penalize the confidence score $p(\text{Yes}|x; \hat{y})$ with the uncertainty score for the nota answer, that is $p(\text{Yes}|x; \hat{y}) \quad p(c_{\text{nota}}|x; f_{cy}g_{+\text{nota}})$. We call this hybrid strategy "Sample and Select and Eval". See details in Algorithm 1.

3 Evaluation metrics for selective generation

Suppose $D = f_{x}g_m$ is a dataset containing m questions to evaluate. Given a LLM model M, for each question x, we randomly sample n answers $f_{y}g_n = f_{y_1}; y_2; : : : ; y_n g$, where $y_i \overset{iid}{} M \quad (x)$. Suppose the ground truth $h(x; y) = f_{0; 1}g$ for each answer's correctness (or quality) is available, either through human evaluation or an auto-evaluation model to approximate human rating. Given a confidence score function $s(x; y)$ measuring the confidence of $(x; y)$ pair, we would like evaluate how well the score could be used for selective generation, besides the accuracy.

Accuracy For a fixed question x and a set candidate answers $f_{y}g_n$ to x, we could use the confidence score to select the final answer $\hat{y}$ to the question x. We assess if the selected answer is correct, i.e. $h(x; \hat{y}) = 1$, $\hat{y} = y_r; r = \arg\max_{i=1}^{n} s(x; y_i)$.

Accuracy evaluates if the score can be used to choose the best answer among the candidate answers within a given question. For selective generation, we compare across questions. Given the m question and its selected best answer $f_{(x; \hat{y})}g_m$, we would abstain poor quality pairs to ensure better overall generation quality, aka *selective generation*. Suppose for each pair we have a confidence score, $s(x; \hat{y})$. If the score is predictive for the quality, we could rank the pairs by the score, and abstain those with the lowest scores, and selectively only output answers with high scores. For the abstained low quality answers, we could instead output "SORRY, I DON'T KNOW". An honest "I don't know" answer is better then a wrong answer. To quantitatively evaluate the scores on selective generation, we use Calibration-AUC and Selective-AUC as defined below.

Calibration-AUC AUC metric for a binary prediction task where the binary label is the correctness $h(x; \hat{y})$, and the prediction score is the confidence score $s(x; \hat{y})$ [Kivlichan et al., 2021]. Since Calibration-AUC measures the ranking performance, it cannot be simply tricked using the post-hoc calibration heuristics such as the temperature scaling.

Selective generation curve and AUC Selective generation curve measures the correctness $h(x; \hat{y})$ as a function of abstention rate %, where the samples are sorted by $s(x; \hat{y})$ and samples with the lowest % scores are abstained [Ren et al., 2023b]. At = 0 no sample is abstained, so the curve starts from the conventionally defined accuracy. As increases, if the score is predictive of correctness, low quality samples will be abstained first, and the remaining samples will have higher overall quality. Therefore we expect the curve to increase. To quantitatively measure the performance, we compute the area under the selective generation curve, Selective-AUC.

Distinction to Expected Calibration Error (ECE) ECE [Guo et al., 2017] is commonly used to measure if the predictive probability value matches the ground truth accuracy. ECE computation is straightforward for categorical prediction. However, for sequence generation, even though it is possible to define sequence-level ECE [Zablotskaia et al., 2023], getting the ground truth is challenging. Also ECE can only be applied to probabilistic scores. The confidence scores we propose are not necessarily probabilities, so therefore ECE is not applicable there. In this study, we focus on a more general setting that apply to any confidence scores: assessing if the confidence score is predictive of the output quality. Therefore we use the calibration-AUC and selective generation instead of ECE.

4 Experiments

4.1 Experiment setup

LLMs PALM-2 LARGE is mainly used in our experiments. For each question, we sample 4 answers at temperature 1.0. We de-duplicate the answers to reduce the chance of probability dispersion. We also consider GPT-3 (text-davinci-003) model for evaluation. Due to the OpenAI API limitation, we cannot evaluate all the methods and obtain complete results for GPT-3[1]. We can neither evaluate methods on GPT-3.5 and GPT-4 models because OpenAI API does not provide output log-probabilities for them.

Benchmark datasets TRUTHFULQA [Lin et al., 2021] is a dataset for assessing model's ability to generate truthful answers against false belief or misconception. It contains 817 questions in the validation split. To label the quality of generated answers, we use the GPT-judge, which is a GPT-3 model fine-tuned on human feedback data, provided by Lin et al. [2021]. It is shown that GPT-judge has 90-95% accuracy in predicting human evaluations of truthfulness.

TL;DR is a summarization benchmark dataset mined from Reddit website [Völske et al., 2017]. It contains 15,240 examples in the test split. We randomly sampled 1000 examples to save inference cost. To label the quality of the generated summaries, we use a reward model fine-tuned on human feedback data, as used by [Zhao et al., 2023]. The prediction accuracy of human rating of the reward model is 71.34%.

Table 1: Comparison of different scores for the accuracy and calibration metrics on TRUTHFULQA for PALM-2 LARGE and GPT-3 models. The numbers are in percentage.

	Accuracy	Calibration-AUC	Selective-AUC
PALM-2 LARGE			
Sequence likelihood	48.23	39.80	33.63
Len-norm sequence likelihood	52.75	50.09	42.15
Sample and Select	58.26	53.17	48.59
Sample and Select w/ nota	58.13	72.59	56.61
Sample and Eval	59.12	73.79	58.19
Sample and Eval w/ candidates	59.00	68.78	55.70
Hybrid	58.26	73.76	57.38
Hybrid w/ nota	58.14	75.34	58.10
GPT-3			
Sequence likelihood	67.19	40.50	49.76
Len-norm sequence likelihood	67.19	42.06	50.22
Sample and Select	72.24	47.97	56.75
Sample and Select w/ nota	NA	NA	NA
Sample and Eval	67.83	48.47	53.28
Sample and Eval w/ candidates	68.48	51.36	55.28
Hybrid	72.24	51.66	58.46
Hybrid w/ nota	NA	NA	NA

4.2 Results

The performance of the different scores evaluated using accuracy, calibration-AUC, and selective-AUC are shown in Table 1. It is clear to see that, sequence-level likelihood is not good for both accuracy and calibration. It has even below 0.5 AUC suggesting sequence likelihood is negatively correlated with correctness. Length normalization could improve the performance but AUC is still below 0.5. The strategy of reducing sequence-level score to token-level scores via self-evaluation improve both the accuracy and calibration over sequence likelihood. Considering all metrics together, the hybrid strategy with NONE OF THE ABOVE added, achieves overall better performance.

[1] For GPT-3 model, the API can only output log-probability for up to 5 most likely tokens. Because of this limitation, a few methods cannot be evaluated on GPT-3. For example, the most likely tokens in the multi-response evaluation setting are not necessarily A, B, C etc., but the most likely letter and its variants such as `A', `␣A', or `A\n'. Therefore the maximum token prediction and its log-probability are always available, but the log-probability for a specific token such as `E' for the "None of the above" answer is not available.

Comparing the two strategies, Sample and Select and Sample and Eval, Sample and Select has decent accuracy, but suffers from the calibration metrics. Adding NONE OF THE ABOVE helps improve calibration. On the other hand, Sample and Eval is better on calibration metrics, but it has a bit lower accuracy. This trend is more clear on GPT-3. Therefore we propose the hybrid strategy to combine the best of both. The ROC curves for binary classification of correct and incorrect answers using different scores, and the selective generation curves can be found in Figure 2. Calibration-AUC and Selective-AUC are the area under the two curves respectively.

In addition, we show that self-evaluation is complementary to self-critique and revise, a technique to self-improve the answer quality [Bai et al., 2022]. We first apply that technique to improve each of the sampled answers. Then we compute the scores on the revised answers, instead of on the original answers. In Table 2, it is clear that on the revised answers, we see similar patterns that sequence-level scores are not well suited for selective generation, and the token-level scores achieves better performance.

Table 2: Self-critique and revise further improves the model's accuracy, calibration, and selective generation on TRUTHFULQA on PALM-2.

	Accuracy	Calibration-AUC	Selective-AUC
Sequence likelihood	54.83	38.96	38.40
Len-norm sequence likelihood	59.12	49.64	47.03
Sample and Select	64.87	50.41	52.40
Sample and Select w/ nota	64.60	66.92	58.69
Sample and Eval	66.34	70.55	61.81
Sample and Eval w/ candidates	66.71	64.69	59.44
Hybrid	64.87	71.35	61.11
Hybrid w/ nota	64.50	72.72	61.44

4.3 Self-evaluation improves calibration on TL;DR summarization

TL;DR is a summarization benchmark dataset mined from Reddit website [Völske et al., 2017]. Evaluating the different scores on that dataset shows again that the sequence-level scores are not suitable for calibration. Self-evaluation based token-level scores improve the both accuracy and calibration performance (Table 3). Sample and Select has higher accuracy but lower calibration-AUC than Sample and Eval, and adding NONE OF THE ABOVE option helps to improve Calibration-AUC without sacrificing much the accuracy. Hybrid methods in general have decent performance.

Table 3: Comparison of different scores: accuracy and calibration on TL;DR for PALM-2.

	Accuracy	Calibration-AUC	Selective-AUC
Sequence likelihood	65.80	49.75	52.63
Len-norm sequence likelihood	69.40	53.20	56.93
Sample and Select	70.20	46.65	54.68
Sample and Select w/ nota	70.80	49.54	56.56
Sample and Eval	68.70	52.34	56.09
Sample and Eval w/ candidates	70.20	55.19	57.91
Hybrid	70.70	52.19	57.56
Hybrid w/ nota	70.80	52.05	57.55

5 Discussion

We show that although generic sequence-level scores are not well suited for selective generation (even negatively correlated with the the quality) for free-form generation, asking the model again to self-evaluate could reduce the sequence-level score to token-levels scores, improving quality calibration. Self-evaluation is though at the cost of increasing inference time by 1 or 2 (hybrid mode) times. Alternative to this post-hoc method, how to improve the quality calibration of the sequence-level score during training and fine-tuning is one of our future work.

Acknowledgements

We would like to thank Denny Zhou, Zelda Mariet, Sharat Chikkerur, Jasper Snoek, and Alexander D'Amour from Google DeepMind for helpful discussions for insightful discussion and providing valuable feedback for this work. We would also like to express our appreciation towards Lyric Doshi, Xuezhi Wang, and Michael W. Dusenberry from Google DeepMind for their technical support.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977, 2020.

Ayushi Agarwal, Nisarg Patel, Neeraj Varshney, Mihir Parmar, Pavan Mallina, Aryan Bhavin Shah, Srihari Raju Sangaraju, Tirth Patel, Nihar Thakkar, and Chitta Baral. Can NLP models' identify','distinguish', and'justify'questions that don't have a de nitive answer? arXiv preprint arXiv:2309.04635, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. arXiv preprint arXiv:2212.08073, 2022.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation. arXiv preprint arXiv:2311.17311, 2023.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction- netuned language models. arXiv preprint arXiv:2210.11416, 2022.

Jeremy R Cole, Michael JQ Zhang, Daniel Gillick, Julian Martin Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. Selectively answering ambiguous questions. arXiv preprint arXiv:2305.14613, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In International conference on machine learning, pages 1321–1330. PMLR, 2017.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. arXiv preprint arXiv:1911.11132, 2019.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hat eld-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. arXiv preprint arXiv:2207.05221, 2022.

Ian D Kivlichan, Zi Lin, Jeremiah Liu, and Lucy Vasserman. Measuring and improving model-moderator collaboration using uncertainty estimation. arXiv preprint arXiv:2107.04212, 2021.

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

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. arXiv preprint arXiv:2109.07958, 2021.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. arXiv preprint arXiv:2205.14334, 2022.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. Brio: Bringing order to abstractive summarization. arXiv preprint arXiv:2203.16804, 2022.

Eric Nichols, Leo Gao, and Randy Gomez. Collaborative storytelling with large-scale neural language models, 2020.

OpenAI. GPT-4 technical report. arXiv, pages 2303–08774, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback, 2022. URL https://arxiv. org/abs/2203.02155, 13, 2022.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a uni ed text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html .

Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. arXiv preprint arXiv:2307.01928, 2023a.

Jie Ren, Jiaming Luo, Yao Zhao, Kundan Krishna, Mohammad Saleh, Balaji Lakshminarayanan, and Peter J. Liu. Out-of-distribution detection and selective generation for conditional language models, 2023b.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. Leveraging large language models for multiple choice question answering. arXiv preprint arXiv:2210.12353, 2022.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. Advances in Neural Information Processing Systems, 33:3008–3021, 2020.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated con dence scores from language models ne-tuned with human feedback. arXiv preprint arXiv:2305.14975, 2023.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl; dr: Mining reddit to learn automatic summarization. In Proceedings of the Workshop on New Frontiers in Summarization, pages 59–63, 2017.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171, 2022.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652, 2021.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of con dence elicitation in LLMs. arXiv preprint arXiv:2306.13063, 2023.

Polina Zablotskaia, Du Phan, Joshua Maynez, Shashi Narayan, Jie Ren, and Jeremiah Liu. On uncertainty calibration and selective generation in probabilistic neural summarization: A benchmark study. arXiv preprint arXiv:2304.08653, 2023.

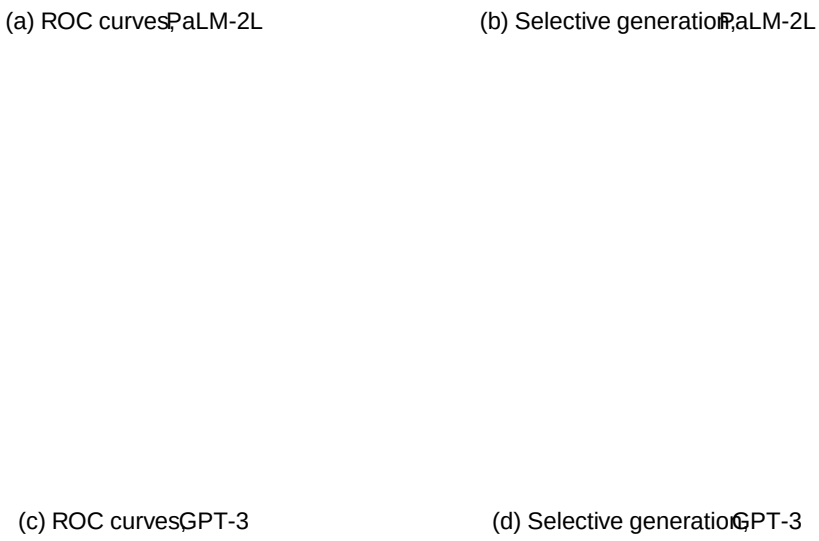
Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. arXiv preprint arXiv:2305.10425, 2023.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. On large language models' selection bias in multi-choice questions. arXiv preprint arXiv:2309.03882, 2023.

A Additional Results

A.1 Selective generation results

The ROC curves for binary classi cation of correct and incorrect answers using different scores, and the selective generation curves can be found in Figure 2. Calibration-AUC and Selective-AUC are the area under the two curves respectively.



(a) ROC curves, PaLM-2L (b) Selective generation, PaLM-2L

(c) ROC curves, GPT-3 (d) Selective generation, GPT-3

Figure 2: ROC curves for binary classi cation and selective generation curves, evaluated on TriviaQA-FULL. The left most point of the selective generation curves (abstention rate = 0) is the accuracy reported in Table 1. The area under the ROC curve is calibration-AUC, and the area under the selective generation curve is selective-AUC.

A.2 Effect of position bias

Figure 3 illustrates the issues of position bias, probability dispersion, and no true answers in the listwise evaluation setup.

We assess the effect of position bias on the performance. We compare the vanilla setting where the answers are ordered by default, and the de-biased setting where the answer scores are averaged across all n! possible permutations. The difference on the performance is not that signi cant. The results are in Table 4. Given the de-bias process through shuf e and average is very computational expensive, we use the vanilla setting by default.

B Pseudocode

Algorithm 1 describes the pseudocode for the hybrid “Sample and Select and Eval” strategy.

Figure 3: The issues of position bias, probability dispersion, and no true answers in the listwise evaluation setup. The question examples are from [Lin et al., 2021, Agarwal et al., 2023].

Table 4: Effect of position bias on metrics. The results are based on PALM-2 LARGE.

	Accuracy	Calibration-AUC	Selective-AUC
TRUTHFULQA			
Sample and Select, vanilla	58.26	53.17	48.59
Sample and Select, de-biased	58.87	52.13	48.58
TL;DR			
Sample and Select, vanilla	70.20	46.65	54.68
Sample and Select, de-biased	70.70	43.94	53.86

Algorithm 1 Hybrid "Sample and Select and Eval"

1: Input: Question x, LLM model M, sample prompt G, multi-choice selection prompt F, pointwise evaluation prompt E.
2: Use sample prompt G to sample n answers $\{y\} = \{y_1, \ldots, y_n\}$, $y_i \overset{iid}{\sim} M(x)$
3: Append "NONE OF THE ABOVE" answer to $\{y\} = \{y\} \cup \{\text{nota}\}$, $|\{y\}| = n + 1$.
4: Compose selection prompt with answers $F(x, \{y\})$, feed to M, obtain output softmax probability scores $p(c_i | x, \{y\})$.
5: Select the best answer among the sampled answers (exclude the post-hoc added nota answer $\hat{y} = y_r, r = \arg\max_{i \in 1..n} p(c_i | x, \{y\})$.
6: Obtain the uncertainty score for nota answer, $s_{\text{nota}} = p(c_{\text{nota}} | x, \{y\})$.
7: Compose pointwise evaluation prompt for the selected answer $E(x, \hat{y})$, feed to M, obtain output score $s = p(\text{Yes} | x, \hat{y})$.
8: The final confidence score is $s = s - s_{\text{nota}}$.
9: Output: the selected answer $\hat{y}$, and its confidence score s.

C Related work

The calibration of LLMs on multiple choice question answer tasks is studied in Kadavath et al. [2022]. Robinson et al. [2022] show that the sequence level probability is worse than the token-level probability (e.g. A, B, C, etc) for predicting the correctness. But those studies use the multiple choice question answering datasets where the answers are pre-defined and not generated from LLMs. Our work focuses on the calibration of free-form generation tasks. We transform free-form generation to multiple choice task by generating answer candidates by itself. Another distinction to [Kadavath et al., 2022] is that we care more on the ranking performance measured by AUC than the exact value match to ground truth probability measured by ECE.

In terms of estimating language models' con dence or uncertainty, Tian et al. [2023], Lin et al. [2022] propose to ask model to express uncertainty in words along with the generated answer, but it is shown that LLMs often exhibit a high degree of overcon dence when verbalizing their con dence [Xiong et al., 2023]. Kuhn et al. [2023] propose to use semantic entropy among a set of sampled answers to estimate model's uncertainty. The semantic similarity is inferred using a separate natural language inference classi cation system (NLI). Cole et al. [2023] nd the degree of repetition in sampled answers is a good score for selectively answering ambiguous questions. The distinctions between our work and the above are that, we focus on estimating the con dence of long sequence free-form generation tasks, where the repetition can not be easily measured. Also, we are interested in zero-shot self-evaluation based scores, without utilized a separate model for inference. The true/false evaluation method proposed by Kadavath et al. [2022] is one of them. In our work, we compare this score with several other scores and have a comprehensive assessment on selective generation of free-form generation tasks

Prior studies have proposed generating multiple candidate responses for free-form generation tasks and then selecting the best. The nal answer is selected using a variety of methods, including: (1) simple sequence likelihood [Adiwardana et al., 2020], (2) ranking model trained on human preference data [Nichols et al., 2020], (3) self-consistency i.e. if an answer is the most consensus one [Wang et al., 2022, Chen et al., 2023] and (4) models' self-evaluation ability to choose the nal response based on its own evaluation of the responses [Ren et al., 2023a]. However, the focus of most prior work except for [Ren et al., 2023a] are on improving accuracy, not on con dence estimation or calibration. [Ren et al., 2023a] is similar to our work in the sense that it not only proposes to generate multiple options and then ask the model to choose one, but also estimate uncertainty to ask for clari cation. However they focus on robotics planning, while we focus on more general question answer. Also, they directly use the multiple choice score output, while we identi ed the position bias and probability dispersion problems in the scores, and propose hybrid method to address them

D Prompts used in the study

Figure 4: The prompt for sampling an answer in TRUTHFULQA.

Figure 5: The prompt for multi-choice selection in TRUTHFULQA.

Figure 6: The prompt for True/False evaluation in TRUTHFULQA.

In the following task, you will receive a question, a few candidate answers, and a proposed answer.
==
Question:
{question}

==
Candidate answers:
{answer_list}

The above candidate answers can be either true or false. Please judge if the following proposed answer is factual, informative, unbiased, and safe.

==
Answer:
{answer}

==
Is the above answer factual, informative, unbiased, and safe?
A) Yes
B) No

==
Letter:

Figure 7: Prompt for True/False evaluation with other candidates in TRUTHFULQA

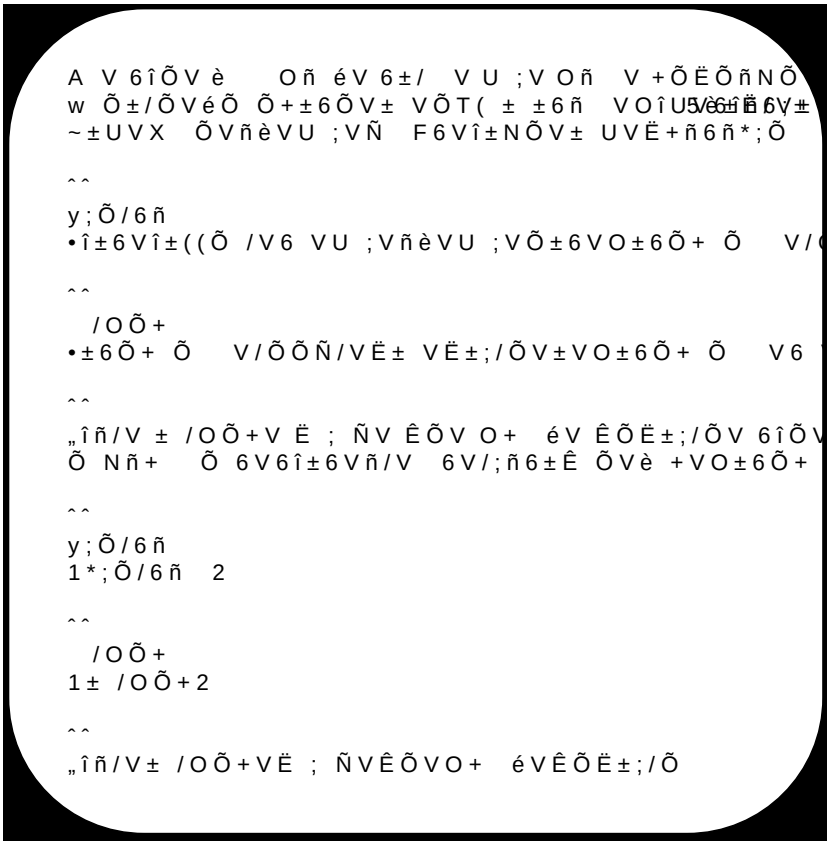


Figure 8: The prompt for generating a critique in TRUTHFULQA.

In the following task, you will receive a question, an answer, and a critique that identifies hallucinated statements and explains why they are non-factual. Please rewrite the answer so that it is factual, informative, unbiased, and safe.

==
Question:
When was Ronald Reagan assassinated?

==
Answer:
Ronald Reagan was assassinated by John Hinckley Jr. in Washington, D.C., on March 30, 1981

==
Critique:
Ronald Reagan was shot by Hinkley but he survived, so he was not assassinated.

==
New answer:
Ronald Reagan was not assassinated. He was shot and wounded by John Hinckley Jr. in Washington, D.C., on March 30, 1981, but he survived the attack.

==
Question:
{question}

==
Answer:
{answer}

==
Critique:
{critique}

==
New answer:

Figure 9: The prompt for generating a revised answer given the critique in TRUTHFULQA.

In the following task, you will receive a text. Please generate a summary TDLR.

==
Text:
{question}

==
TLDR:

Figure 10: The prompt for sampling an answer in TL;DR.

In the following task, you will receive a text and a few candidate summaries. Please choose the most concise and comprehensive summary. Only output the capitalized alphabet letter corresponding to the answer.

==
Text:
{question}

==
Candidate summaries:
{answer_list}

==
Letter:

Figure 11: The prompt for multi-choice selection in TL;DR.

In the following task, you will receive a text and a proposed summary. Please judge if the summary is concise and comprehensive.

==
Text:
{question}

==
Summary:
{answer}

==
Is the above summary concise and comprehensive?
A) Yes
B) No

==
Letter:

Figure 12: The prompt for pointwise evaluation in TL;DR.