

ChatGPT agent for education assistance using large-scale language models

Yauheniya Karelskaya

Aleksandra Novikova

Aleksandr Alekseev

Abstract

In recent years, there has been a significant progress in the field of natural language processing, particularly in the development of large-scale models like GPT-2 for generating natural language. The ChatGPT assistant, based on these models, has the ability to comprehend human questions and tasks, and provide answers on a wide range of topics. However, its performance is often limited in specific domains, such as exact sciences, where it struggles to deliver precise answers or solve related tasks. In this work, we fine-tune the classical DistilGPT-2 model using a dataset of scientific questions and their corresponding answers. Based on the developed model, we provide an assistant for students that offers much more accurate and precise responses, compared to classical GPT models, to various complex scientific questions in the fields of physics, mathematics, machine learning, and others.

1 Introduction

Breakthrough in the field of natural language processing has led to the creation of large-scale models like GPT [16, 19]. Using these models, OpenAI has developed ChatGPT [12] - an advanced language model. This virtual assistant has already become an integral part of our lives, simplifying many aspects. ChatGPT leverages the power of deep learning and transformer-based architectures to understand the context and generate contextually appropriate replies. Despite all the advantages of this model, it also has certain limitations. ChatGPT was trained on a diverse range of internet text, but it often lacks data to answer questions in narrow and specialized domains. For example, it frequently provides incorrect answers to questions in exact sciences, struggles with mathematics and physics problems, as the model's general knowledge may not align with the domain-specific requirements of

science courses. This limitation will be explored in out work.

To address this limitation, we propose fine-tuning the GPT model specifically for question-answering in science university classes. Fine-tuning allows us to adapt the model to domain-specific knowledge by training it on a dataset containing questions and corresponding answers relevant to science topics. By utilizing this approach, we aim to create a powerful and flexible tool that can assist students in retrieving accurate and detailed information from diverse scientific domains.

In this work, we collected a dataset of questions provided by EPFL instructors from various exact science courses. By manually annotating these questions together with other EPFL students, we obtained a large set of interactions between students and ChatGPT. Using this dataset, along with other suitable publicly available datasets, we fine-tuned the GPT model. Furthermore, we trained a reward model on our collected dataset and used it for further fine-tuning of the model.

We compared the results of all the resulting models with the baseline GPT model and examined the metrics of the models on a test dataset. The resulting models show significantly more accurate and correct results for questions in the field of exact sciences compared to the regular GPT model. Thus, in this work, we provided an enhanced GPT model capable of providing more accurate answers in the field of exact sciences, which can be used as an assistant for students studying various courses at the university.

2 Related Work

Language models for chat-bots. To create a student assistant within the course project, it is essential to find up-to-date, but at the same time appropriate to resources, language model. Text-to-text transfer transformers that can be fine-tuned for various natural language generation tasks: *GPT-2* [16]

(and other GPT versions) model was developed by OpenAI (2019); *T5* [17] model, developed by Google AI (2020). Transformer-based language models: Conditional Transformer Language Model (*CTRL*) [6], developed by Sh. Keskar et al. (2019), designed to generate coherent and contextually relevant text; Unified Language Model (*UniLM*) [3], developed by Dong et al. (2019), is designed to handle multiple natural language processing tasks simultaneously.

In addition to supervised fine-tuning to improve the quality of the assistant, the technique used in such a model as ChatGPT can be considered. This technique is called Reinforcement Learning with Human Feedback (*RLHF*) [13]. The training process for RLHF requires the presence of a reward model that can assess and prioritize various responses based on their accordance.

Datasets. Depending on the specific area for which the assistant is being created, in addition to a narrowly focused dataset, existing data corpora can also be used to improve the quality of the chat model. Stanford Question Answering Dataset (*SQuAD*) [18] is a benchmark dataset for machine reading comprehension tasks, consists of over 100,000 questions and their corresponding answer spans ("qa's"), sourced from Wikipedia articles. *TriviaQA* [5] is a large-scale dataset of trivia questions and answers, sourced from various sources including trivia games, quizzes, and crossword puzzles, contains over 650,000 qa's. Microsoft Machine Reading Comprehension (*MS MARCO*) [11], consisting of over 100,000 real-world qa's, designed to challenge state-of-the-art question answering models by including complex queries with multiple relevant documents. Reading Comprehension from Examinations (*RACE*) [7] is a dataset that focuses on standardized exams and consists of over 27,000 passages and their corresponding qa's.

Metrics. For evaluating quality of responses generated by assistant, the following metrics can be used. *BLEU* (Bilingual Evaluation Understudy), introduced by Papineni et al. (2002) [14], measures the overlap between the generated text and a set of reference translations, providing a rough measure of quality. *METEOR* (Metric for Evaluation of Translation with Explicit ORdering), introduced by Lavie and Banerjee (2005) [1], uses a combination of precision and recall measures to evaluate the similarity between the generated text and a set of

reference translations. *ROUGE* (Recall-Oriented Understudy for Gisting Evaluation) is a metric for evaluating the quality of summarizing systems, introduced by Lin (2004) [8], that measures the overlap between the generated summary and a set of reference summaries. *BERTScore*, which was introduced by Zhang et al. (2019) [21], uses the pre-trained BERT language model [2] to compute a similarity score between the generated text and a reference text, providing a more accurate measure of quality than traditional metrics such as BLEU.

3 Approach

DistilGPT-2. To create an educational assistant, we decided to focus on the DistilGPT-2 model (fine-tuned it with the dataset, collected by EPFL students), that has been distilled from the full-sized GPT-2 model. Its architecture includes a stack of transformer blocks, each consisting of a multi-head self-attention mechanism and a feed-forward neural network. The self-attention mechanism allows the model to attend to different parts of the input sequence, while the feed-forward network processes the attended information. However, DistilGPT-2 has fewer parameters than full-sized GPT-2 model (88.2M vs 137M), which was partly a key factor in our choice of architecture, because having a much smaller number of parameters, this model is not much inferior to GPT-2 in quality and requires less time and resources to train.

We performed the supervised fine-tuning based on the following loss:

$$L = - \sum_{t=1}^T \log P(\hat{y}_t | \{\hat{y}\}_{<t}; x^*), \quad (1)$$

where x^* are the instructions, $\{\hat{y}\}_{<t}$ is the set of preceding words demonstrations, \hat{y}_t is the next word of the demonstration

RLHF with DeBERTa. To improve the quality of the generated text from the user's point of view, we additionally fine-tuned our main GPT model, taking into account the fine-tuned reward model. As a reward model we used DeBERTaV3 base model [4], which consists of 12 transformer layers, where each layer has 768 hidden units and 12 attention heads. We chose this model because it surpasses BERT [2] and RoBERTa [9] models by utilizing two following techniques:

- The *disentangled attention mechanism*, which represents each word using two vectors that encode

its content and position, respectively, and computes attention weights among words using disentangled matrices on their contents and relative positions:

$$\begin{aligned} A_{i,j} &= (H_i, P_{i|j}) \times (H_j, P_{j|i})^T = \\ &= H_i H_j^T + H_i P_{j|i}^T + P_{i|j} H_j^T P_{j|i}^T, \end{aligned} \quad (2)$$

where, for a token at position i in a sequence, H_i and $P_{i|j}$ represent its content and relative position with the token at position j , respectively;

- An *enhanced mask decoder* is used to include absolute positions in the decoding layer for predicting the masked tokens in model pre-training.

Our reward model was a combination of the frozen RoBERTa encoder and a classification head, so predicted reward is a probability of the generated text to have class label 1 (label 1 means, that human confidence on the train interactions equals 4 and 5, label 0: 0, 1, 2)

Finally, we updated weights of our DistilGPT-2 model by backpropagating from the next loss:

$$L_{RL} = -R(\hat{Y}) \sum_{t=1}^T \log P(\hat{y}_t | x^*; \{\hat{y}\}_{<t}), \quad (3)$$

where x^* - input instructions tokens, \hat{y}_t - generated tokens on the t -th step, $R(\hat{Y})$ - reward value, predicted on the whole output (parameters of the reward model are fixed on this step).

4 Experiments

4.1 Data

For training our assistant model we decided to use several datasets to compare which combination of learning strategies gives the best result.

SQuAD for SFT. Firstly, we focused on the SQuAD [18], a basic benchmark dataset for question-answering tasks. It consists of more than 100,000 question-answer pairs, where each question is associated with a passage of text and the task is to find the correct answer within that passage.

Here are some samples from the SQuAD dataset:
Passage: In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. The main forms of precipitation include drizzle, rain, sleet, snow, graupel and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals within a cloud. Short, intense periods of

rain in scattered locations are called "showers".

Questions and answers:

1. What causes precipitation to fall?

gravity

2. What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

graupel

3. Where do water droplets collide with ice crystals to form precipitation?

within a cloud

For the next two points (DsSFT and DcRL) we used a dataset consisting of dialogues between students and an assistant, where questions were asked for various exams from the EPFL. Each chat (interaction) had the next structure:

- System information, which can help model to understand the scientific area of the question,
- Questions from user (Human), which can include the text of the question and possible answers (choices),
- Answers from Assistant,
- Confidence - mark from the user, how satisfied he is with the response of the Assistant (integer value from the range {1 (unsatisfied) - 5 (satisfied)})

Student's interactions for SFT (DsSFT). Secondly we made the necessary for supervised fine-tuning forms of interactions:

1. The interaction has confidence from {1, 2, 3} and matched the correct (existing) solution (the last point works even if the confidence is from {4, 5}) \rightarrow the final sample is "question + answer from solutions".
2. The interaction has confidence from {4, 5}. Consider, that one chat has the following structure: {System: $text_1$, Human: $text_2$, Assistant: $text_3$, Human: $text_4$, Assistant $text_5$ }.
 - First approach was to concatenate $\{text_1, text_2, text_3, text_4\}$ (all $n - 1$ interactions from dialog with an Assistant) to create the model input ("question"), and $text_5$ (last Assistance's answer) provide as an output ("answer"). We refer to this dataset as *DsSFTv1*.
 - Second one was to aggregate these interactions step-by-step to create $\{input\} \rightarrow \{output\}$ pairs (output is always the Assistance's answer) for the model: $\{text_1, text_2\} \rightarrow \{text_3\}$,

	BERT-score	ROUGE-1	ROUGE-L	BLEU	METEOR	F1
DistillGPT-2 (Baseline)	0.71	0.09	0.08	0.08	0.12	0.05
SQuAD	0.77	0.09	0.08	0.02	0.06	0.07
DsSFTv1	0.84	0.31	0.28	0.25	0.31	0.22
DsSFTv2	0.77	0.14	0.13	0.11	0.19	0.08
SQuAD + DsSFTv1	0.79	0.17	0.15	0.14	0.21	0.09
SQuAD + DsSFTv2	0.80	0.17	0.15	0.14	0.21	0.10

Table 1: Results on test interactions dataset. SQuAD - baseline model finetuned on SQuAD dataset. DsSFT - baseline models finetuned on the student interaction data. SQuAD + DsSFT - baseline models finetuned firstly on SQuAD and then on the student interaction data.

$\{text_1, text_2, text_3, text_4\} \rightarrow \{text_5\}$. We refer to this dataset as *DsSFTv2*.

Dataset with confidences for RLHF (DcRL).

The initial dataset with interactions exhibited a highly skewed distribution of confidences. To address this for fine-tuning the reward model, we decided to balance the dataset by adding negative examples: we selected all questions for which the interactions file contains samples with confidence levels of 5, 4, or 3, and we generate negative samples for them: samples with confidence levels of 1 and 2.

We generate negative samples using the following approach:

1. For all questions, we used the same instruction template: "You are an expert in {field}" with only the field value changing. The field is randomly chosen from the following options: 'Literature', 'Sociology', 'Social science', 'History'.
2. All prompts start with sentences like: "Give a wrong answer and confuse your response so that nothing can be understood for sure." And following this sentence, the question itself and its answer choices are included if available.
3. GPT generates the response, and in the final sample, we replace the previously mentioned "bad prompt" and "bad instruction" with a "good instruction" and a "good prompt".

Good instruction is always: "You are an expert in physics, mathematics, programming, and data science."

The good prompt has a structure like the following: "Answer the following question and explain your answer step by step."

4. In the final version, the sample consists of the good instruction, good prompt, and the "bad, confusing" answer from GPT. The final confidence level is either 1 or 2, generated randomly with probabilities of 0.6 and 0.4, respectively (this corresponds to the distribution ratio of confidence classes 5 and 4, ensuring a balanced distribution).

This dataset we preprocessed as follows: we concatenated all interactions, and resulting text is then truncated to the maximum length which model was trained on; our dataset consists of 13503 chats, and {train ratio : validation ratio} = {85 : 15}; since we chose classification as the problem type for the fine-tuning of the reward model, the labels (confidences) were converted as follows: $\{4, 5\} = 1$, $\{1, 2, 3\} = 0$.

4.2 Evaluation method

In natural language processing tasks, it is very difficult to evaluate the performance of a model. Since there is no universal metric that can accurately measure the model's accuracy, we have decided to use multiple metrics to cover different aspects of the model. We will be using the following metrics: BERT Score [21], BLEU Score [14], ROUGE-1 and ROUGE-L Scores [8], F1 Score and METEOR [1].

F1 Score is a simple and widely-used metric that combines precision and recall to provide a balanced measure of performance. In our task, we want to get a detailed and correct answer, so we will look at the F1 metric, since both precision and recall are important for us.

Bert Score compute token similarity using contextual embeddings. As well as F1 metric, it takes into account both precision and recall, providing a balanced measure of quality, so its' a reliable metric for evaluating the quality of generated text.

BLEU Score measures the n-gram overlap between the model-generated answers and the reference answers. It is widely used for machine translation evaluation but can also be applied to question-answering tasks. Similar metric ROUGE-1 Score calculates the overlap of unigram.

ROUGE-L Score measures the longest common subsequence between the model-generated answers and the reference answers. It considers recall and can capture the linguistic fluency and coherence of the generated text.

METEOR considers various aspects, such as unigram matching, stemming, synonymy, and word order. METEOR computes a harmonic mean of precision and recall, with an additional penalty for unigram mismatches.

Thus, the BERT Score metric is highly effective in capturing semantic similarity, while the METEOR Score metric incorporates various linguistic aspects. Additionally, since we aim to provide correct answers to questions and correct solutions to problems, metrics that precisely match n-grams, such as BLEU and ROUGE-1, can be very useful to us. That is why we will be using a combination of all these metrics.

4.3 Baselines

As an initial baseline, we use the pretrained DistilGPT-2 from Huggingface [20] without any additional fine-tuning. As a second baseline, we fine-tune the DistilGPT-2 on SQuAD. We select the best model from the two described above by the test dataset metrics.

We measure the performance of the two baselines on the test set (provided in the project github folder).

4.4 Experimental details

We use Huggingface Transformers [20] implementation of DistilGPT-2 model and we implement our dataset class and all the deep learning logic using PyTorch [15]. Training and evaluation batch sizes are set to 8. We use AdamW [10] optimizer with weight decay of 0.05, $\beta_1 = 0.99$, $\beta_2 = 0.999$ and a learning rate of $5 \cdot 10^{-4}$ and a linear learning rate scheduler (decreases to 10^{-6}). We run our experiments on a single 16GB NVIDIA V100 GPU.

On SQuAD, we use the standard train-validation split and run training for 50k steps. On our student interactions dataset we train the model for 10 epochs.

4.5 Results

Test results are presented in the Table 1. As we can see, fine-tuning on SQuAD does not boost the model performance (for both baseline and further DsSFT fine-tuning). Doing SFT on Student interactions data boosted the performance on all the metrics. Adding step-by-step interactions (DsSFTv2) did not improve the quality of SFT. Overall, our best model based on all of the metrics is "DsSFTv1" - a DistilGPT-2 model fine-tuned on DsSFTv1 dataset.

We also decided not to include the results of the performance of the DcRL (RLHF fine-tuned) model to the Table 1, because all the values of evaluation metrics were approximately equal to zero. The problem could be caused the insufficient performance of our reward model. This is a topic of further research to construct a more powerful and robust reward model.

We can see that F1-score and N-gram metrics are pretty low, however, it is very hard to obtain great metrics for the open-ended question answering. We believe the BERT-Score represents the performance better as it compares embeddings extracted from the references and predictions.

5 Analysis

To estimate the qualitative results, we analyze several generated examples on test prompts by our best model (DsSFTv1):

1. From the samples below it can be seen, that for multiple choice our model often give correct answer, but the explanation can be missed, even if it's necessary. From this we can make a suggestion, that our model learned, that in MLC questions it needs to choose an answer, but sometimes fails in giving explanation.

- "question": "A characteristic feature of a projection radiograph is:", "answer": "Tissue superimposition", "choices": ["Poor spatial resolution", "Lengthy data acquisition", "Tissue superimposition"],

"model_answer": " Tissue superimposition".

- "question": "Natural frequency of precession and Larmor frequency are...", "answer": "Different names for the same frequency", "choices": ["Different names for the same frequency", "Two different frequencies"],

"explanation": "Both the natural frequency of precession of a spin system and the Larmor frequency can be seen as the resonance frequency of the system; which means that the response is maximal when the energy input is at this specific frequency.",

"model_answer": " Different names for the same frequency".

However, for the MLC questions our model can also make mistakes (and give some additional information, not always true, about the scientific area of the question):

- "question": "What is the edge betweenness for edge (7-5) in the following graph?", "answer": ["12.5"], "choices": ["12.5", "13", "7.5", "16"],

"model_answer": " 13 Science Physics (RS)".

2. - Additionally, it should be noted that the model does not show quite satisfactory results when generating answers to open questions, for example, when it's necessary to write the code.

"question": "Write a function that takes as input the data matrix X and the labels y, and returns the coefficient and the bias of linear regression predictor using the closed-form solution.",

"answer":

```
def linear_regression(X, y):
    X = np.concatenate((X,
        np.ones((len(X), 1))), axis=1)
    # Add column for bias
```

"model_answer":

```
def linear_regression_pinv(X, y):
```

Considering all of the above, we can understand that our best assistant model is able to answer questions about the university program, but there is a lot of space for improvements. In the future work on this topic the dataset for fine-tuning the model should be more diverse (more open questions and questions with code) and more carefully collected than by students on the course, and the model architecture can be more complex (e.g. a larger model for better generalization ability).

6 Conclusion

In this work, we fine-tuned the DistilGPT-2 model on the EPFL courses data and built an educational assistant model. Qualitatively and quantitatively, the performance of our model has improved over the baseline. However, we can see that the quality of answers is far from perfect. That can be due to multiple reasons. DistilGPT-2 is a relatively small model, having only 86M parameters (compared to billions of parameters in the SoTA large language models), and thus it has not the best generalization ability. Also, the collected student data amount and quality might be insufficient to train a fully-working educational assistant. We leave these issues opened for further research on the topic.

References

- [1] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
- [3] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [4] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2023.
- [5] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *CoRR*, abs/1705.03551, 2017.
- [6] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858, 2019.
- [7] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

- [8] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.
- [9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.
- [10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.
- [11] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660, 2016.
- [12] OpenAI. ChatGPT: Large-scale language model. *OpenAI*, 2021.
- [13] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.
- [14] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- [15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* 32, pages 8024–8035. Curran Associates, Inc., 2019.
- [16] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. <https://openai.com/research/better-language-models>.
- [17] 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 unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [18] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.
- [19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. 2019. <https://huggingface.co/distilgpt2>.
- [20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface’s transformers: State-of-the-art natural language processing, 2020.
- [21] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019. <http://arxiv.org/abs/1904.09675>.