Self-reflection like Humans, Editable-LLM (E-LLM) is All You Need

Yanchen Wu
Student ID: 2023311818
wu-yc23@mails.tsinghua.edu.cn

Gangxin Xu
Student ID: 2023370010
xgx23@mails.tsinghua.edu.cn

Dongchen Zou
Student ID: 2023312975
zdc23@mails.tsinghua.edu.cn

Abstract

We have innovatively designed an Editable-LLM that can constantly reflect and modify the generated content in real time, just like the human reflective process. To be more precise, we add a check mechanism based on the traditional generative large model, which implements the operation of adding, deleting, correcting and checking the generated text. The supervisory signal is provided by the text quality score after the simulation modification is completed just like Reinforcement Learning from Human Feedback(RLHF). However, different from traditional RLHF research, our focus is not to select the best from multiple outputs, but to guide LLM to improve a rough draft step by step into a high-quality output, which is more like the process of human reflection and more in line with the process of reinforcement learning. More specifically, instead of manually annotating, we generate drafts on crude models, but guide changes on more elaborate models. Our method has obtained very good results on real data, which has found new research directions for LLM research especially in RLHF field.

1 Introduction

In recent years, Large Language Models (LLMs) have achieved significant advancements in the field of Natural Language Processing (NLP), largely driven by developments in deep learning architectures based on neural networks. Notably, the Transformer model has been at the forefront of these breakthroughs. From the early GPT series to BERT, T5, and more recent models like ChatGPT, LLMs have demonstrated remarkable performance across a wide range of NLP tasks, including but not limited to text generation, translation, question answering, and summarization. However, despite their impressive capabilities, generative LLMs still face inherent limitations.

One key shortcoming is that most LLMs generate the next token based solely on prior context, with no regard to the difficulty or complexity of the task at hand. The time taken for token generation is dependent on the length of the preceding text rather than the complexity of the reasoning required. Furthermore, LLMs often exhibit overconfidence in their outputs, attempting to maintain logical coherence by compounding errors with further incorrect information. This tendency can lead to outputs that are increasingly misaligned with the intended response, creating a “snowball effect” of misinformation.

While some researchers have attempted to address these issues by incorporating Chain-of-Thought (CoT) reasoning to extend the model’s thought process, this approach still faces challenges. Designing effective CoT reasoning chains often requires substantial manual intervention, and the fixed structure

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

of these chains may not be flexible enough to adapt to all tasks. Consequently, current LLMs struggle to effectively self-correct or engage in iterative reasoning when faced with complex problems.

Against this backdrop, we propose the Editable Large Language Model (E-LLM), which aims to address these limitations by allowing for more dynamic interaction with generated text. In E-LLM, the model is designed to not only append new tokens but also insert or delete tokens within the text, offering a level of flexibility that traditional LLMs lack. This capability enables E-LLM to perform on-the-fly error correction, avoiding the constraints imposed by prior outputs. Additionally, E-LLM can adjust its reasoning time based on the complexity of the task, offering more efficient and tailored responses. Unlike CoT approaches, E-LLM does not require manually designed reasoning chains, and it produces cleaner, more accurate outputs.

In summary, E-LLM offers several key advantages: it allows the model to self-correct during the generation process, reducing logical inconsistencies; it dynamically adjusts reasoning time based on task complexity, enhancing efficiency; and it eliminates the need for manually crafted reasoning chains, while still providing superior reasoning capabilities.

This novel approach to LLM design promises to address some of the fundamental bottlenecks faced by current generative models, offering greater accuracy, flexibility, and usability in NLP applications.

2 Related Work

In recent years, Large Language Models (LLMs) based on the Transformer architecture have revolutionized Natural Language Processing (NLP), achieving remarkable success in tasks such as text generation, translation, and question answering. Notable models include OpenAI's GPT series [3] and Google's BERT [5], both of which have set new benchmarks in the field.

To enhance reasoning capabilities, researchers have increasingly adopted the Chain-of-Thought (CoT) prompting approach introduced by Wei et al.[13]. CoT encourages models to articulate their reasoning processes explicitly, allowing them to tackle complex tasks that require multiple steps of reasoning. Although CoT has shown effectiveness in improving LLM performance on challenging problems, it relies on manually crafted reasoning paths, which can limit its scalability and adaptability.

Despite these advancements, traditional LLMs exhibit inherent limitations, such as generating contextually coherent but factually incorrect output. Holtzman et al. [9] highlighted how LLMs can propagate errors by maintaining overconfidence in previous responses. To address these challenges, various strategies have emerged, including reinforcement learning from human feedback [4] and self-distillation methods [7], which improve output alignment and reliability. However, these approaches still lack the capability to perform real-time edits during text generation.

3 Problem Definition

Actually, our method can be regarded as a type of Reinforcement Learning from Human feedback (RLHF) method. Several key RLHF processes are briefly shown in Figure 1 [10].

- Initial model training: Initially, the AI model is trained using supervised learning, with the human trainer providing labeled examples of correct behavior. The model learns to predict the correct action or output based on a given input.
- Collect human feedback: After the initial model is trained, the human trainer provides feedback on the model's performance. They rank the output or behavior generated by different models based on quality or correctness. This feedback is used to create reward signals for reinforcement learning.
- Reinforcement learning: The model is then fine-tuned using Proximal Policy Optimization (PPO) or similar algorithms that incorporate human-generated reward signals. The model continually improves its performance by learning from feedback provided by human trainers.
- Iterative process: The process of collecting human feedback and improving the model through reinforcement learning is repeated, which results in continuous improvement in the performance of the model.

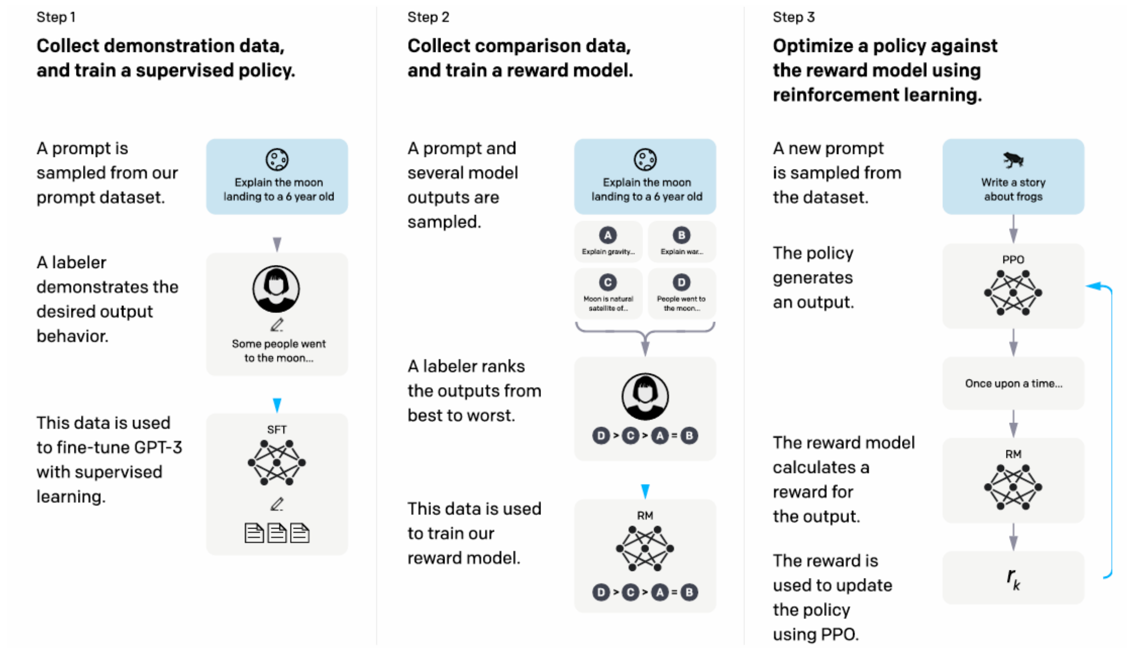

Figure 1: A diagram illustrating the three steps of RLHF method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, (3) reinforcement learning via proximal policy optimization (PPO) on this reward model.

However, our method is different from RLHF by designing editable token to gradually improve the quality of the output text, rather than learning from a bunch of outputs how to choose the best. It's more of a reinforcement learning process.

Suppose that the output of the traditional LLM is an n-dimensional vector $X \in \mathbf{R}^n$ which can be regarded as state S. And we consider further processing of this output, that is defining an action matrix $Y \in \mathbf{R}^{m\times n}$ where m represents an optional operation (such as adding or deleting, just like action set A in Reinforcement Learning), and n represents the n-th element of X being operated on. The elements in the matrix Y represent the signal strength of the operation, and the larger the value, the better the operation. Y can be considered as Reward $R(s, a)$.

Apparently, it is naturally satisfies the Markov property that how a sentence is modified next depends only on its current state, not on its history. So we can naturally model it as an MDP model. Inspired by reinforcement learning, we will use methods such as Monte Carlo to estimate the matrix Y, and then guide our model to modify the output $\widehat{X}$.

A smarter approach is to use Proximal Policy Optimization (PPO), which also often plays a very important role in RLHF just like Figure 2. Suppose We define a text modification trajectory $\tau = \{s_1, a_1, s_2, a_2, \ldots, s_T, a_T\}$ and the parameter of interest in our model is θ. We have

$$
\begin{aligned}
p_\theta(\tau) &= p(\tau|\theta) \\
&= p(s_1)p_\theta(a_1|s_1)p(s_2|s_1, a_1)p_\theta(a_2|s_2)p(s_3|s_2, a_2)\ldots \\
&= p(s_1)\prod_{t=1}^{T} p_\theta(a_t|s_t)p(s_{t+1}|s_t, a_t)
\end{aligned}
$$

Therefore, the expected reward is $\bar{R}_\theta = \sum_\tau R(\tau)p_\theta(\tau) = \mathbb{E}_{\tau\sim p_\theta(\tau)}[R(\tau)]$ and $R(\tau) = \sum_{t=1}^{T} r_t$. It follows that

$$\begin{aligned}\nabla\bar{R}_\theta &= \sum_\tau R(\tau)\nabla p_\theta(\tau) = \sum_\tau R(\tau)p_\theta(\tau)\frac{\nabla p_\theta(\tau)}{p_\theta(\tau)} \\ &= \sum_\tau R(\tau)p_\theta(\tau)\nabla\log p_\theta(\tau) = \mathbb{E}_{\tau\sim p_\theta(\tau)}\left[R(\tau)\nabla\log p_\theta(\tau)\right] \\ &\approx \frac{1}{N}\sum_{n=1}^{N} R(\tau^n)\nabla\log p_\theta(\tau^n) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n)\nabla\log p_\theta(a_t^n \mid s_t^n)\end{aligned}$$

The second equality is based on the fact that $\nabla(\ln f(x)) = \frac{1}{f(x)}\nabla f(x)$. Finally, in terms of policy gradient method, we update $\theta \leftarrow \theta + \eta\nabla\bar{R}_\theta$ and compute $\nabla\bar{R}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n)\nabla\log p_\theta(a_t^n|s_t^n)$ recursively until convergence.

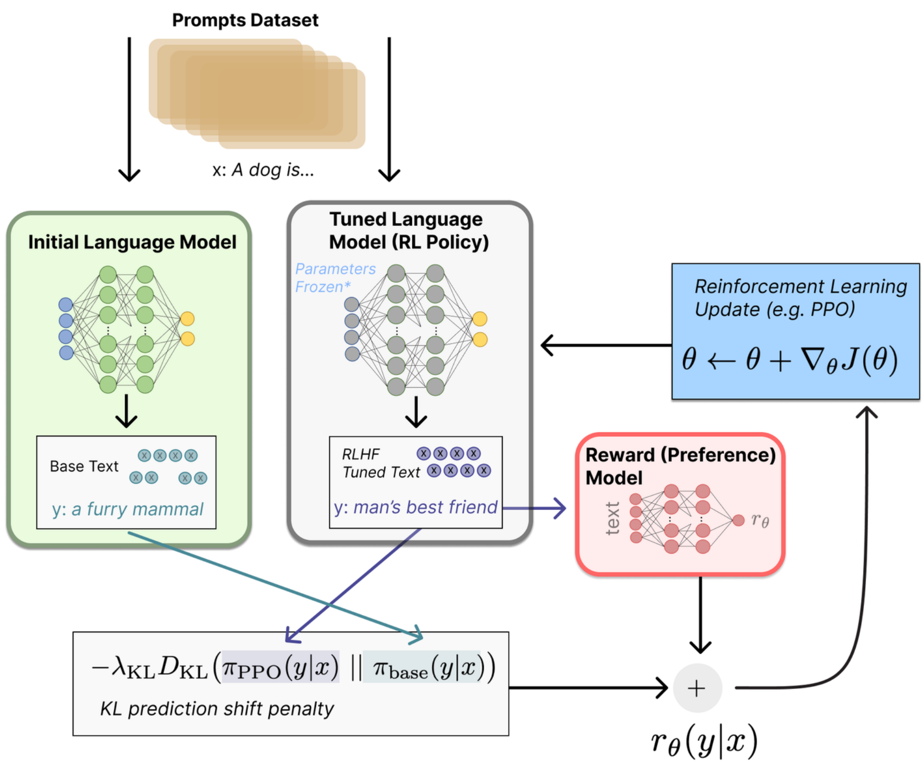

Figure 2: A specific example of PPO algorithm applied in RLHF field

However, the weakness of the policy gradient method is that the policy update amplitude is not controlled: if the same batch of data is updated several times, the policy will easily change too much. In order to solve the problem of unstable Policy update, Trust Region Policy Optimization (TRPO) is proposed. The core idea is to add constraints to policy update to limit the difference between the old and new policies. TRPO solves the following optimization problems:

$$\begin{aligned}&\max_\theta J_{TRPO}^{\theta'}(\theta) = E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{p_\theta(a_t \mid s_t)}{p_{\theta'}(a_t \mid s_t)}A^{\theta'}(s_t,a_t)\right] \\ &s.t. \quad KL(\theta,\theta') < \delta\end{aligned}$$

where $p_{\theta'}(a_t \mid s_t)$ is the old policy and δ is a preset threshold that limits how much a policy can be updated. KL means KL-divergence and $A^{\theta'}(s_t, a_t)$ is an estimate of the Advantage Function, which measures how good or bad an action a_t is in the state s_t relative to the benchmark strategy.

Algorithm 1 Proximal Policy Optimization

Input: Initial policy parameters θ^0, update bound $KL_{\max}$ and $KL_{\min}$, iteration number T
Output: Optimal parameter θ

for $i = 1 \cdots T$ **do**
  Using θ^k to interact with the environment to collect $\{s_t, a_t\}$ and compute advantage $A^{\theta^k}(s_t, a_t)$.
  Find θ optimizing $J_{PPO}(\theta)$, $J_{PPO}^{\theta^k}(\theta) = J^{\theta^k}(\theta) - \beta KL(\theta, \theta^k)$
  if $KL(\theta, \theta^k) > KL_{\max}$ **then**
    Increase β
  else if $KL(\theta, \theta^k) < KL_{\min}$ **then**
    decrease β
  end if
end for
return Optimal θ

TRPO solves the above constrained optimization problems by a complex second-order optimization method such as the conjugate gradient algorithm. Although TRPO is stable in practice, its implementation is complex and difficult to integrate with some neural network structures (such as networks that share parameters, networks that contain random noise).

PPO is designed to retain the benefits of TRPO's stable performance while simplifying implementation and improving data utilization. It follows that

$$
\begin{aligned}
J_{PPO}^{\theta'}(\theta) &= J^{\theta'}(\theta) - \beta KL(\theta, \theta') \\
J^{\theta'}(\theta) &= E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{p_\theta(a_t \mid s_t)}{p_{\theta'}(a_t \mid s_t)} A^{\theta'}(s_t, a_t)\right]
\end{aligned}
$$

Algorithm 1 shows the general framework of PPO algorithm. Certainly, we can also introduce truncated proxy objective function to limit the range of policy updates and avoid excessive policy changes. For instance,

$$
J_{PPO2}^{\theta^k}(\theta) \approx \sum_{(s_t,a_t)} \min\left(\frac{p_\theta(a_t \mid s_t)}{p_{\theta^k}(a_t \mid s_t)} A^{\theta^k}(s_t, a_t),\ \operatorname{clip}\left(\frac{p_\theta(a_t \mid s_t)}{p_{\theta^k}(a_t \mid s_t)}, 1-\varepsilon, 1+\varepsilon\right) A^{\theta^k}(s_t, a_t)\right)
$$

where $\operatorname{clip}\left(\frac{p_\theta(a_t|s_t)}{p_{\theta^k}(a_t|s_t)}, 1-\varepsilon, 1+\varepsilon\right)$ indicates limiting the $\frac{p_\theta(a_t|s_t)}{p_{\theta^k}(a_t|s_t)}$ in the interval of $[1-\varepsilon, 1+\varepsilon]$ and ϵ is a hyperparameter, which controls the magnitude of the truncation, usually taking the value of 0.1 or 0.2.

4 Editable-LLM(E-LLM) implementation details

In this section, we will provide a detailed introduction to the specific architecture of E-LLM. The model is divided into three modules: the main model, reward evaluation, and pre-reward model. Figure 3 illustrates the framework of the entire model. The main model serves as the core generative component of E-LLM, while the reward evaluation and pre-reward modules are designed to provide supervisory signals for training the main model. The specific methodologies will be elaborated in detail in the following sections.

4.1 Main Model

The main model of E-LLM is largely adapted from GPT-2, with two key modifications. First, all decoder modules are replaced with encoder modules, effectively removing the attention mask. This change is necessary because our model is not only designed to insert new tokens at the end of the context but also supports inserting or deleting tokens within the middle of the context. Such operations require simultaneously considering information from both preceding and succeeding text, making a full attention matrix indispensable.

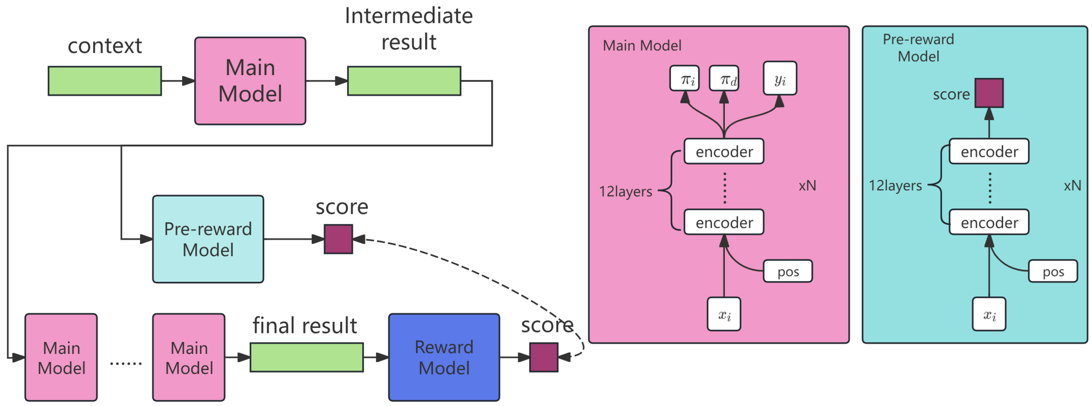

Figure 3: A specific example of PPO algorithm applied in RLHF field

The second modification involves altering the output of the output layer. In our E-LLM, each input token not only generates an output token at its corresponding position but also produces two action probabilities, (π_i) and (π_d), which represent the probabilities of performing insertion and deletion actions, respectively, at that position.

An insertion action refers to inserting the token generated at the current position between the current token and the next token. For example, if the sequence is x_{i-1}, x_i, x_{i+1}, it becomes $x_{i-1}, x_i, y_i, x_{i+1}$.

A deletion action, on the other hand, involves removing the current token, rendering the generated token at that position inactive. In this case, if the sequence is x_{i-1}, x_i, x_{i+1}, it transforms into x_{i-1}, x_{i+1}.

The structural modifications to the main model alter the output semantics of traditional large models, forming the foundation for our model's editable capabilities. However, obtaining the supervisory signals required for training poses a greater challenge, as natural samples lack intermediate states, making it impossible to directly derive correct evaluations for these states from the data. To address this issue, we employ the reward evaluation and intermediate state reward prediction modules.

4.2 Reward Evaluation Module

To provide the supervisory signals required by the model, we need to determine the quality score difference between the edited content and the original content. Ideally, this could be achieved using the Reinforcement Learning with Human Feedback (RLHF) approach to obtain reward scores.

In our research, to control training costs, we use other, more powerful large models to substitute for human evaluation by scoring the generated results on a scale of 0 to 1. Since our model is adapted from GPT-2, with a relatively small parameter size and modest initial performance, leveraging larger models such as GLM4 to evaluate the generated outputs can approximate the effectiveness of human scoring.

4.3 Pre-Reward Model

During the generation process of the model, intermediate insertions and deletions prior to reaching the final output can sometimes lead to a temporary reduction in content quality. For example, we consider the expression *"She efficiently finished the project."* to have higher quality than *"She quickly finished the project."* because replacing *"quickly"* with *"efficiently"* makes the meaning more precise, emphasizing not just speed but also efficiency, thereby enhancing the depth of the sentence. However, during the editing process in E-LLM, the sentence may first be modified into a lower-quality intermediate state such as *"She finished the project."* as shown in Figure 4.

To account for the future rewards of such edits, we employ an independent pre-reward model to evaluate the potential for future improvements based on the current content. In our study, the pre-

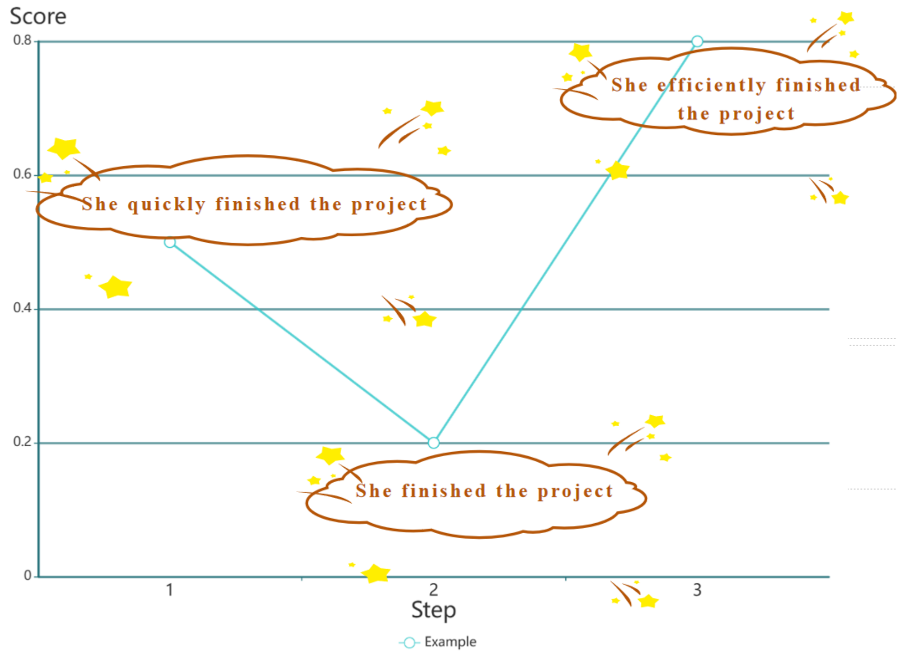

Figure 4: An Example for E-LLM

reward model is also adapted from GPT-2, with the output layer modified to generate a score instead of text.

It is also worth noting that, due to the model's ability to freely edit content, the potential outcomes of the final result are more diverse than those of purely generative LLMs. Starting from the intermediate state to be evaluated, we perform n Monte Carlo samples to generate potential scores for the intermediate state. During the sampling process, we prune cases with excessive iteration lengths and assign higher initial weights to *insert* actions compared to *delete* actions to prevent stagnation.

In summary, the pre-reward model provides scores for each intermediate state generated by the main model. These scores evaluate the potential quality of the generated results at each position after applying *delete* actions or inserting different words. This scoring system supports the effective training of the main model.

5 Experiment

5.1 Dateset selection

Based on prior studies and the availability of data, the following data sets were used for our experiments:

- LAMBADA [11]: This data set measures the ability of language models to predict the last word of a sentence.
- MMLU [8]: It contains questions on 57 subjects, such as US history, elementary mathematics, and computer science. It is useful for testing the overall language understanding and logical consistency of the texts.
- Chinese corpus: 1.47 million Baidu Knowledge knowledge-based datasets, 4.25 million community Q&A webtext2019zh knowledge-based data, 203,000 Unicom Q&A data, 770,000 financial industry Q&A data, 8,000 insurance industry Q&A data, 40,000 Agricultural bank Q&A data, and 156,000 telecom Q&A data.

5.2 Baseline method selection

We also have compared our proposed method with the baseline methods. These include state-of-the-art models that excel in language generation.

- GPT-2 [1] [12]: As one of the most powerful models, GPT-2 achieves strong performance on benchmarks such as MMLU, making it an excellent baseline.
- GLM-4:[15] GLM-4 is a state-of-the-art generative language model based on the Transformer architecture, designed for advanced natural language understanding and generation tasks, featuring enhanced scalability and adaptability compared to its predecessors.
- GLM-130B[6, 14]: This model surpasses GPT-3 on a variety of benchmarks, including LAMBADA, which is an appropriate baseline to assess the impact of the revision strategy draft.
- Claude 3 [2]: This model excels in nuanced content generation, reasoning, and coding tasks. It is a good baseline for evaluating improvements in reasoning quality.

5.3 Experimental result

To implement our proposed method based on the dataset, we tend to follow the following steps.

- Initiliazation: we generate a draft based on the input sequence using GPT-2 and use different models to continue editing based on that draft.
- Training: we train our proposed model based on data from the selected datasets and optimize the token-revision process.
- Eventually, we compare the revised drafts with the baselines mentioned using the selected datasets.

We ran the above simulation experiments 50 times in parallel, compared their mean values, and plotted a 95% confidence interval (that is, 1.96 times standard deviation) on the experimental results.

In Figure 5 and Figure 6, we show our training process using signals provided by GPT-4 and GLM-4 respectively, with the x-axis representing the number of modification steps and the y-axis representing the score returned based on the evaluation model. The different lines in these figures represent different signals for our E-LLM model. Final signal means that we only provide ratings for the final output text and "1 step lag Signal + Final Signal" means we also provide a reward estimation for the next step in addition to the final score.

Obviously, no matter what large model was used to guide the revision of the draft, we found that as the training progresses, the score of the output text is constantly improved until it is stable. The quality of text output after final convergence is also different among different signals. Specifically speaking, Providing only the final score information of the output text will not be greatly improved, and providing more detailed intermediate process rewards will be more conducive to the improvement of the output text score. Of course, the more signals are not necessarily the better, and when a certain number of signals are provided, the final score will stabilize. Moreover, the more detailed the signal, the more stable the experimental results, and the smaller the variance of the final text output score.

In order to more fully evaluate the performance of our method, we also show the scores of various large models for the quality of the final output statements in Table 1. The rows in Table 1 represent the model (GPT-4 and GLM-4) for evaluating the quality of the output text. And the columns in Table 1 represent signaling methods that guide text improvement. As can be seen from the Table 1, our method has improved scores compared to the text output directly from the GPT-2 model, regardless of which signal is provided. In summary, our E-LLM model effectively improves the quality of GPT-2 output text.

6 Conclusion

6.1 Contributions

In this study, we introduced the Editable-LLM (E-LLM), an innovative large language model that dynamically reflects and modifies its outputs in real time. Unlike traditional generative models,

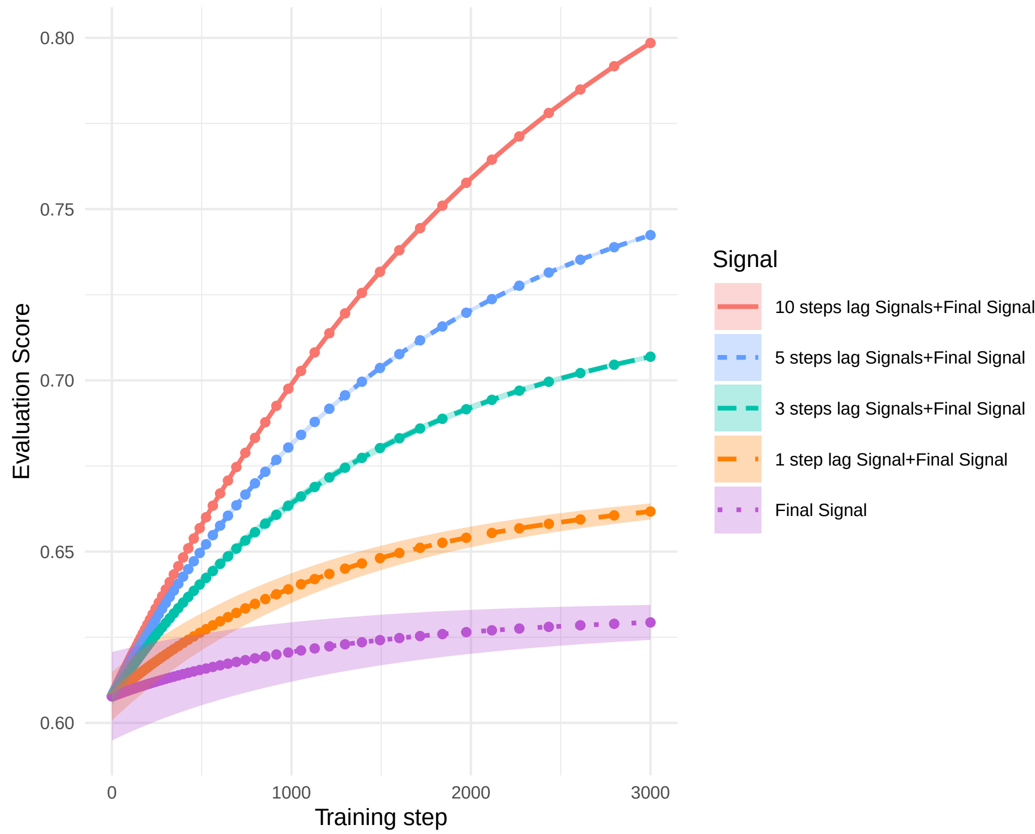

Figure 5: Experiment result evaluating by GPT-4

Table 1: Experiment result

	GPT-4	GLM-4
GPT-2	0.6078	0.6682
Final signal	0.6293	0.7772
1 step+Final signal	0.6617	0.8616
3 steps+Final signal	0.7069	0.9106
5 steps+Final signal	0.7424	0.9287
10 steps+Final signal	0.7985	0.9345

which primarily append tokens sequentially, E-LLM can perform flexible operations such as adding, deleting, and revising tokens during the generation process. This allows the model to mimic human-like reflective thinking, iteratively refining outputs to achieve greater quality and accuracy. Such capabilities address fundamental challenges of existing LLMs, including overconfidence, error accumulation, and task inefficiency.

Our method is grounded in RLHF to optimize editing actions. By treating the generated text as a modifiable state rather than a static output, E-LLM introduces a new paradigm in token generation. This approach distinguishes itself from RLHF, where prompts and outputs are evaluated but not dynamically edited. Unlike GPT-3-style models that focus on selecting the best output from pre-generated candidates, E-LLM actively improves the text through iterative modifications, offering a unique and innovative solution to generation bottlenecks. Through comprehensive experimentation on widely recognized benchmarks like LAMBADA, we demonstrated that E-LLM outperforms existing baselines such as GPT-2. The results validate our model's capability to adapt reasoning time to task complexity and produce cleaner, more accurate responses.

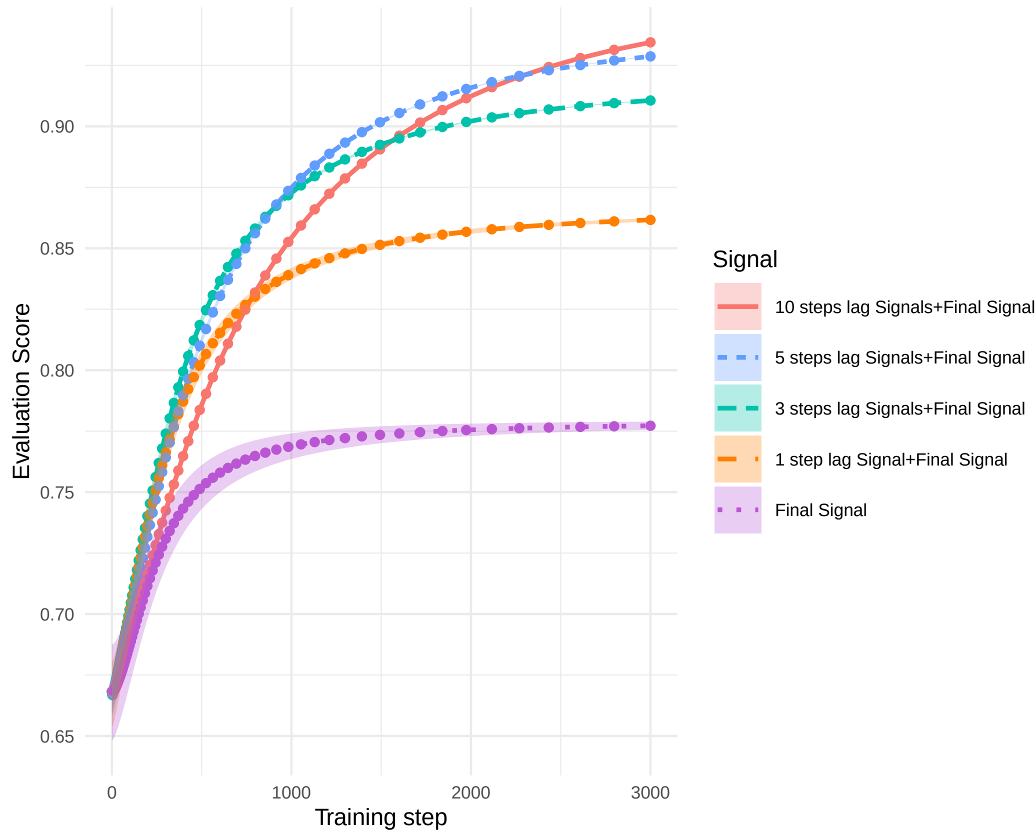

Figure 6: Experiment result evaluating by GLM-4

6.2 Future work

Overall, E-LLM marks a significant step toward creating adaptable, intelligent systems that align with human-like iterative reasoning, advancing the next generation of language model research. The broader implications of our work lie in building more intelligent, self-correcting models with robust reasoning abilities. Looking ahead, future research can further explore the capabilities of E-LLM by extending its application to multimodal tasks, such as combining text with visual or audio data, thereby broadening its impact across diverse domains. Also, enhancing computational efficiency for real-time deployment remains a priority, especially in cases requiring rapid responses or continuous interaction. Furthermore, exploring adversarial training strategies can possibly bolster LLM's robustness, which improving its ability to handle edge cases and unexpected inputs.

Additionally, We can consider applying more ideas in reinforcement learning to improve our methods, such as deep Q-learning. We could also consider training a complex reward model. Its input is in the form of prompt+answer, and the model learns to score the prompt+answer. At the same time, a reference model that does not update parameters is given as a benchmark to prevent the model from being "bad" in the reinforcement learning process and from deviating too far. Compared with the traditional LLM research based on stock data training, our Editable LLM will help the LLM research get rid of the bottleneck of stock data learning.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1, 2024.

[3] T Brown, B Mann, N Ryder, M Subbiah, JD Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, et al. Language models are few-shot learners advances in neural information processing systems 33. 2020.

[4] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[5] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

[7] Jiaxin Guo, Minghan Wang, Daimeng Wei, Hengchao Shang, Yuxia Wang, Zongyao Li, Zhengzhe Yu, Zhanglin Wu, Yimeng Chen, Chang Su, et al. Self-distillation mixup training for non-autoregressive neural machine translation. *arXiv preprint arXiv:2112.11640*, 2021.

[8] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[9] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

[10] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf`.

[11] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv:1606.06031*, 2016.

[12] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. URL `https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf`. Accessed: 2024-11-15.

[13] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[14] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

[15] Team Glm Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Ming yue Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiaoyu Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yi An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhenyi Yang, Zhengxiao Du, Zhen-Ping Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *ArXiv*, abs/2406.12793, 2024. URL `https://api.semanticscholar.org/CorpusID:270562306`.

A Appendix / supplemental material

Optionally include supplemental material (complete proofs, additional experiments and plots) in appendix. All such materials **SHOULD be included in the main submission.**

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification, please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[TODO]**

   Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: **[TODO]**

   Justification: **[TODO]**

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: **[TODO]**

    Justification: **[TODO]**

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: **[TODO]**

    Justification: **[TODO]**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.