LLM-Evolve: Evaluation for LLM's Evolving Capability on Benchmarks

Jiaxuan You[1], Mingjie Liu[1], Shrimai Prabhumoye[1],
Mostofa Patwary[1], Mohammad Shoeybi[1], Bryan Catanzaro[1],
[1]NVIDIA,
{jiaxuany, mingjiel, sprabhumoye, mpatwary, mshoeybi, bcatanzaro}@nvidia.com

Abstract

The advancement of large language models (LLMs) has extended their use to dynamic and interactive real-world applications, where models engage continuously with their environment and potentially enhance their performance over time. Most existing LLM benchmarks evaluate LLMs on i.i.d. tasks, overlooking their ability to learn iteratively from past experiences. Our paper bridges this evaluation gap by proposing a novel framework, LLM-Evolve, which extends established benchmarks to sequential problem-solving settings. LLM-Evolve evaluates LLMs over multiple rounds, providing feedback after each round to build a demonstration memory that the models can query in future tasks. We applied LLM-Evolve to the MMLU, GSM8K, and AgentBench benchmarks, testing 8 state-of-the-art open-source and closed-source models. Results show that LLMs can achieve performance improvements of up to 17% by learning from past interactions, with the quality of retrieval algorithms and feedback significantly influencing this capability. These insights advocate for more understanding and benchmarks for LLMs' performance in evolving interactive scenarios.

1 Introduction

The rapid development of LLMs has expanded their application to dynamic and interactive real-world scenarios, as known as LLM-based agents. In these contexts, LLMs interact continuously with their environment, and their performance can evolve based on these interactions. Despite these advancements, standard benchmarks for evaluating LLMs, such as the MMLU benchmark (Hendrycks et al., 2020), treat each problem as an i.i.d. sample. This conventional approach fails to assess the ability of LLMs to learn from past experiences and enhance their performance over time.

Our study addresses this gap by exploring whether existing benchmarks can be adapted to evaluate LLMs' capabilities in iterative problem-solving scenarios. While developing entirely new benchmarks could provide insights into LLM self-evolution, the process is often resource-intensive and costly. Alternatively, combining or sampling from existing benchmarks, as seen in recent works (Sakaguchi et al., 2021; Wang et al., 2023; Gema et al., 2024), provides a more feasible solution but still requires extensive computational resources to rerun evaluations for all LLMs involved. Instead, we chose a different approach: modifying the settings of established benchmarks, such as MMLU, without changing their test sets and metrics. This method offers flexibility, convenience, and direct comparability with existing benchmark results, enabling a deeper understanding of how LLMs improve by leveraging previous interactions.

In this paper, we introduce LLM-Evolve, a novel evaluation framework that transforms popular LLM benchmarks into a sequential problem-solving setting. Under LLM-Evolve, LLMs are assessed across multiple rounds, where the environment will provide feedback on each round's outcomes to inform subsequent LLM evaluations. Specifically, the framework saves a given LLM's inputs, outputs, and feedback to a demonstration memory, which the model can then query in future rounds to retrieve relevant experiences for few-shot learning on new tasks. This iterative process allows us to measure how effectively an LLM can use its past experiences to evolve its capabilities over time.

We apply the LLM-Evolve framework to three prominent benchmarks: MMLU for general language tasks, GSM8K for math problem solving (Cobbe et al., 2021), and AgentBench (Liu et al., 2023) which evaluates multi-round interaction capabilities of LLMs. Our experiments include closed-source models including the GPT family and open-source models including the Llama family, Mistral, and Qwen2, with a fixed generation temperature of 0 to eliminate the impact of random

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 16937–16942
November 12-16, 2024

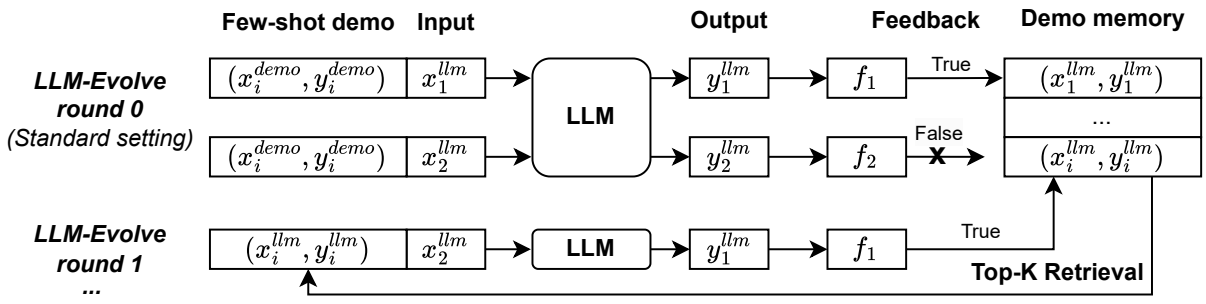

Figure 1: **Overview of LLM-Evolve pipeline**. In each round of evaluation, input-output pairs $(x_i^{\text{llm}}, y_i^{\text{llm}})$ of an LLM with positive feedback $f_i = \text{True}$ will be saved into a demonstration memory $\mathcal{D}$, which can be retrieved with a dense retriever $r_\phi(\cdot)$ as the few-shot demonstration in the next round of LLM-Evolve evaluation.

guessing. The results show consistent predictive performance gains of 1-17% across all benchmarks, demonstrating the efficacy of LLMs in learning from their past experiences.

Further analysis reveals that the quality of the retrieval algorithm and feedback signals significantly influences an LLM's evolving capabilities, and that more capable LLMs tend to benefit less from multi-round interactions. These interesting findings underscore the potential for adapting existing benchmarks to provide a more systematic evaluation of LLMs in dynamic and interactive environments.

2 Related Works

Retrieval-based in-context learning Most LLM benchmarks evaluate a model's capability in a few-shot manner where fixed demonstrations are provided as context to the given query (Hendrycks et al., 2020; Cobbe et al., 2021; Liu et al., 2023). (Xu et al., 2024) provides a comprehensive survey on algorithms for retrieving relevant demonstrations tailored to each input query. We extend this idea to adapt prominent LLM benchmarks to an interactive setting and gain insights into the performance of state-of-the-art LLMs.

LLM benchmarks derived from existing benchmarks Researchers have extended existing LLM benchmarks to study different properties of LLMs. For example, Winogrande (Sakaguchi et al., 2021) was introduced to reduce bias in Winograd (Levesque et al., 2012). MMLU-Redux (Gema et al., 2024), is a subset of 3,000 manually re-annotated questions of MMLU, used to study the discrepancies with the model performance metrics that were originally reported. MMLU-Pro (Wang et al., 2024) makes the original MMLU more robust, especially on reasoning-focused questions. Our work extends the settings of existing LLM benchmarks to study the interactive evolving capabilities of LLMs. It further reveals the correlations between benchmark problems, which could help create more diverse and robust LLM benchmarks in the future.

Self-evolving LLMs. Due to the rising size and capabilities of LLMs, recent efforts have improved the model accuracy on downstream tasks by using inference-based techniques such as refining the generated output through feedback (Madaan et al., 2024; Shinn et al., 2023), and learning through mistakes in ICL by understanding core principles (Zhang et al., 2024). We differ from these approaches by focusing on understanding LLM's evolving capabilities in standard benchmarks.

3 LLM-Evolve Framework

Preliminaries for LLM benchmarks. We use p_θ to denote a pre-trained LLM with parameter θ, x as the problem input, and y as the output. In a standard LLM benchmark setting, fixed few-shot input-output (IO) pairs are provided to facilitate LLM inference, which can be represented as

$$y^{\text{lm}} = p_\theta(x, \{x_i^{\text{demo}}, y_i^{\text{demo}}\}) \quad (1)$$

where y^{lm} is the LLM output and $\{x_i^{\text{demo}}, y_i^{\text{demo}}\}$ are few-shot demonstrations that are *fixed* for all input x, provided by the benchmark.

LLM-Evolve settings. The key assumption of LLM-Evolve is that LMs can gradually achieve better benchmark performance by using better few-shot IO pairs based on their evolving interaction histories. Specifically, we extend the existing LLM benchmark settings by constructing a demonstra-

Table 1: LLMs evaluated under LLM-Evolve settings on MMLU demonstrate consistent accuracy gain, where larger models tend to benefit less from multi-round LLM-Evolve settings. Numbers indicate accuracy in percentage.

MMLU (all 57 subjects)	Llama2-7B	Mistral-7B-v0.2	Llama3-8B	Llama2-70B	Llama3-70B	Qwen2-72B
Standard	47.19	58.90	65.35	63.08	78.97	84.00
LLM-Evolve round1	52.04	63.05	70.36	67.48	82.42	84.79
LLM-Evolve round2	**53.17**	**63.99**	71.02	68.00	**82.82**	**85.33**
LLM-Evolve round3	52.93	63.75	**71.08**	**68.06**	82.82	84.92
LLM-Evolve Gain	***5.74***	4.85	*5.73*	4.98	3.85	1.33

tion memory $\mathcal{D}$ in LLM-Evolve.

$$\mathcal{D} = \{(x_i^{\text{lm}}, y_i^{\text{lm}}, f_i)\} \tag{2}$$

where each tuple includes the input x_i^{lm} and output y_i^{lm} for an LLM, and binary feedback f_i from the environment, indicating whether the experience is desirable. Given a benchmark, the feedback f_i can obtained from the ground truth label in the benchmark. Alternatively, the feedback could be provided by an LLM, in this case, the LLM-Evolve setting reduced to the self-reflecting framework (Shinn et al., 2023) suppose the same LLM is used to provide feedback, or a multi-agent LLM framework suppose a different LLM is used to provide feedback (Wu et al., 2023). In our current implementation, only experiences with positive feedback are saved into the demonstration memory $\mathcal{D}$, and the negative experiences are discarded.

Based on the demonstration memory $\mathcal{D}$, an LLM being evaluated with LLM-Evolve can leverage a retriever r_ϕ, e.g., BERT (Devlin et al., 2018) or Contriever (Izacard et al., 2021), to fetch the relevant experiences from the demonstration memory and replace the originally fixed prompts $\{x^{\text{demo}}, y^{\text{demo}}\}$ in the standard benchmark setting. Specifically, given a new problem input x, we retrieve top-k most relevant positive experiences from the memory $\mathcal{D}$ to obtain the output $y^{\text{LLM-Evolve}}$

$$y^{\text{LLM-Evolve}} = p_\theta(x, \{x_i^{\text{lm}}, y_i^{\text{lm}}\}) \tag{3}$$
$$i \in \underset{x_j^{\text{lm}} \in \mathcal{D}, f_j^{\text{lm}}=\text{True}}{\text{topk-min}} \{||r_\phi(x) - r_\phi(x_j^{\text{lm}})||_2\}$$

Extend LLM-Evolve to multi-turn settings. The discussions above assume a single-turn LLM evaluation with the benchmark. LLM-Evolve can extend to multi-turn settings, such as MT-bench (Zheng et al., 2024) and AgentBench, by saving the full multi-turn interactions in the memory

$$\mathcal{D} = \{(x_{i1}^{\text{lm}}, y_{i1}^{\text{lm}}, ..., x_{it}^{\text{lm}}, y_{it}^{\text{lm}}, f_i)\} \tag{4}$$

and the retrieval module will retrieve the multi-turn demonstrations based on the first input x_{i1}^{lm}.

Multi-round LLM-Evolve. We outlined the steps of applying 1 round of LLM-Evolve evaluation in the discussions above. After obtaining new LLM output $y^{\text{LLM-Evolve}}$ based on Equation 3, we can refresh the demonstration memory $\mathcal{D}$ based on the new experience. The updated $\mathcal{D}$ can then be used to initiate a new round of LLM-Evolve evaluation. Figure 1 provides an overview of LLM-Evolve in multi-round LLM evaluation settings.

4 Experiments

Benchmarks. We apply the LLM-Evolve framework to three prominent benchmarks: (1) MMLU for general language tasks, including 14K multi-choice problems across 57 subject domains, commonly regarded as the gold standard in evaluating LLMs; (2) GSM8K, containing 8.5K high-quality linguistically diverse grade school math word problems, requiring an LLM to accurately generate the exact numerical answer; (3) AgentBench, which evaluates LLMs' multi-turn interaction capabilities when solving real-world problems, such as solving real-world problems on an operating system. We extend these benchmarks with LLM-Evolve with 4 rounds of experiments: all the LLMs start with the standard benchmark evaluation setting, then, 3 additional rounds of LLM-Evolve experiments are applied. We report the accuracy in all three benchmarks. Due to limited resources, we focus on the os-std dataset in AgentBench, where an LLM needs on average 8 turns to solve each problem.

Models. We consider state-of-the-art open-source LLM of different scales including Llama model family (Touvron et al., 2023a,b), Llama2-7B, Llama2-70B, Llama3-8B, and Llama3-70B, Mistral-7B-v0.2 (Jiang et al., 2023), and Qwen2-72B (Bai et al., 2023). We also experimented with closed-source LMs, including GPT-4 (Achiam et al., 2023) and GPT-3.5-turbo provided by Azure

Table 2: LLMs evaluated under LLM-Evolve settings on AgentBench os-std demonstrate consistent accuracy gain. Numbers indicate accuracy in percentage.

AgentBench (os-std)	Llama3-8B	Llama3-70B	GPT-3.5	GPT-4
Standard	18.8	32.6	32.7	43.8
LLM-Evolve round1	21.5	38.9	41.7	47.2
LLM-Evolve round2	24.3	42.4	43.8	47.2
LLM-Evolve round3	**27.1**	**45.1**	**43.8**	**50.7**
LLM-Evolve round4	25.7	45.1	43.8	50.0
LLM-Evolve Gain	8.3	***12.5***	*11.1*	6.9

Table 3: LLMs evaluated under LLM-Evolve settings on GSM-8K demonstrate consistent accuracy gain. Numbers indicate accuracy in percentage.

GSM-8K	Llama2-7B	Llama2-70B	Llama3-8B
Standard	23.28	46.78	52.99
LLM-Evolve round1	29.80	55.19	64.97
LLM-Evolve round2	33.43	57.70	68.91
LLM-Evolve round3	**36.39**	**58.38**	**70.28**
LLM-Evolve Gain	*13.11*	11.60	**17.29**

OpenAI Service. For all the models, we set the generation temperature to 0 to avoid any undesirable behaviors due to randomness. We run 7B/8B models on 1 NVIDIA A100 GPU, and 70B models on 8 NVIDIA A100 GPUs; getting each round of LLM-Evolve for 1 LLM approximately takes 1 hour. We use Contriever (Izacard et al., 2021) as the retriever in LLM-Evolve due to its popularity.

Results on MMLU. The results on MMLU are shown in Table 1. Overall, LLM-Evolve offers an impressive 1-6% accuracy gains on MMLU, indicating that an LLM can significantly benefit from feedback on its generated results. Notably, the major accuracy gain is obtained in the first round of LLM-Evolve; the subsequent gains from additional rounds of LLM-Evolve are within 1%. Interestingly, we found that larger models tend to benefit less from their past experience; our explanation is that larger models are capable of storing the necessary problem-solving knowledge in their weights, therefore rely less on the few-shot demonstrations in the input. Another interesting finding is that the state-of-art Qwen2-72B model benefits the least from LLM-Evolve; such observations were not reported in standard LLM benchmark settings and are worth further investigation.

Results on AgentBench os-std. AgentBench is more challenging than MMLU, both in terms of the task (writing multi-turn Linux scripts to solve real-world OS problems, and in terms of evaluation format (instead of doing multi-choice selection, an LLM needs to obtain precisely correct console output). As is shown in Table 2, LLM-Evolve offers a significant 7-13% accuracy on AgentBench. Our explanation is that the challenging nature of AgentBench tasks makes it very beneficial for the LLMs to leverage their successful past experiences when solving problems. Notably, Llama3-70B, while being 11% less accurate compared to GPT-4 under the standard setting, can surpass a standard GPT-4 after 3 rounds of LLM-Evolve augmentation. We found additional rounds of LLM-Evolve do not help with further improving LLM's performance after 3 rounds.

Results on GSM8K. As is shown in Table 3, LLM-Evolve offers a significant 12-17% accuracy gain on GSM8K. Similar to AgentBench, GSM8K requires an LLM to conduct math reasoning and generate precisely correct numerical outputs. The results further confirm our finding that the more challenging the benchmark problems are, the more beneficial for LLMs to leverage successful past experiences to solve previously failed problems.

Ablation study on LLM-Evolve design. We conduct a comprehensive ablation study for LLM-Evolve with Llama3-8B on the full MMLU dataset, shown in Table 4. (1) Regarding the choice of a specific dense retriever, we found that switching the Contriever to BERT offers a very minor accuracy drop, indicating that any reasonable dense retriever could serve the purpose of LLM-Evolve well. (2) We further found that high-quality feedback data, which is used to filter and select the relevant experiences from the demonstration memory is crucial; in particular, we switch the source of feedback from using the ground-truth label provided in the MMLU benchmark to using Llama3-70B model, and observed around 2% accuracy drop when evaluating Llama-8B model on MMLU. (3) Finally, we demonstrate the effectiveness of the retrieval module; here, we randomly shuffle the order of the original fixed few-shot prompt in the MMLU datasets, and report a problem to be successfully solved if *any* of the answers is correct in the previous rounds; this is a strong baseline - suppose an LLM always generates 4 different answers in 4 rounds, the model would have 100% accuracy; this setting indeed boosts LLM's accuracy but are still 2% less accurate compared to the LLM-Evolve setting.

Table 4: Ablation study shows that effective retriever and high-quality feedback are essential in LLM-Evolve.

MMLU (all 57 subjects)	Llama3-8B			
	round0	round1	round2	round3
Standard	65.35	65.35	65.35	65.35
LLM-Evolve	65.35	**70.36**	**71.02**	**71.08**
(-) From Contriver to BERT	65.35	70.34	71.00	71.04
(-) Feedback from Llama-70B	65.35	68.27	68.53	68.57
(-) No dense retrieval	65.35	67.77	68.90	69.61

5 Conclusion

We presented LLM-Evolve, a framework that transforms standard LLM benchmarks into a sequential problem-solving format, enabling the evaluation of LLMs' abilities to learn and improve through iterative interactions, and paving the way for future research to develop more sophisticated evaluation methods and enhance the evolving learning capabilities of LLMs in real-world applications.

Limitations

While our study provides a relatively comprehensive analysis across three benchmarks and eight LLMs, there are notable areas where further exploration could enhance the generalizability and depth of our findings. Expanding LLM-Evolve to encompass additional benchmarks and a broader spectrum of LLMs would likely offer richer insights, at the cost of additional computational resources.

Our research includes an ablation study that examines the impact of different sources of feedback in constructing the demonstration memory. Specifically, we compare feedback derived from ground-truth benchmark labels against feedback generated by the LLMs themselves. Despite this exploration, our primary results focus on benchmark-derived feedback. This choice is practical and beneficial in controlled experimental settings, yet it does not reflect the complexities of real-world LLM deployment, where ground-truth labels are often unavailable.

We investigated various methods of retrieving past experiences from the demonstration memory, including using experiences with both positive and negative feedback as few-shot demonstrations, and masking experiences directly relevant to the current problem. Our initial findings suggest that these alternative settings are not as ideal as the current LLM-Evolve framework. Nevertheless, these areas warrant further investigation in future research to refine and enhance the LLM-Evolve framework.

Given the constraints of a short paper, we have prioritized the simplicity of our approach which underscores the importance of understanding the evolving capabilities of LLMs in interactive environments. Future research could focus on more sophisticated strategies to improve in-context learning and adaptive response generation.

In summary, while LLM-Evolve provides novel understandings of the evolving capabilities of LLMs on the standard LLM benchmarks, additional opportunities exist for future research to expand upon these findings in more diverse and realistic scenarios.

Ethical Considerations

The development and evaluation of LLM-Evolve in this study are entirely based on benchmarks, which inherently minimizes direct ethical concerns. However, as LLMs with self-evolving capabilities are increasingly deployed in real-world applications, it becomes crucial to anticipate and address potential ethical implications proactively.

While our current implementation of LLM-Evolve relies solely on feedback from benchmarks and other LLMs, future iterations of this framework may incorporate human feedback. In such scenarios, aligning the evolving capabilities of LLMs with human values and societal norms becomes crucial. This alignment is essential to ensure that LLMs operate in ways that are beneficial, fair, and non-harmful to users and society at large.

References

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. 2024. Are we done with mmlu? *arXiv preprint arXiv:2406.04127*.

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. 2023. Mint: Evaluating llms in multi-turn interaction with tools and language feedback. *arXiv preprint arXiv:2309.10691*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Xin Xu, Yue Liu, Panupong Pasupat, Mehran Kazemi, et al. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.

Tianjun Zhang, Aman Madaan, Luyu Gao, Steven Zheng, Swaroop Mishra, Yiming Yang, Niket Tandon, and Uri Alon. 2024. In-context principle learning from mistakes. *arXiv preprint arXiv:2402.05403*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.