Towards Intrinsic Self-Correction Enhancement in Monte Carlo Tree Search Boosted Reasoning via Iterative Preference Learning

Huchen Jiang[*1], **Yangyang Ma**[*1], **Chaofan Ding**[1], **Kexin Luan**[1], **Xinhan Di**[1]

[1]AI Lab, Giant Network

jianghuchen@ztgame.com, mayangyang@ztgame.com, dingchaofan@ztgame.com, luankexin@ztgame.com, dixinhan@ztgame.com

Abstract

With current state-of-the-art approaches aimed at enhancing the reasoning capabilities of Large Language Models(LLMs) through iterative preference learning inspired by AlphaZero, we propose to further enhance the step-wise reasoning capabilities through intrinsic self-correction to some extent. Our work leverages step-wise preference learning to enhance self-verification via reinforcement learning. We initially conduct our work through a two-stage training procedure. At the first stage, the self-correction reasoning ability of an LLM is enhanced through its own predictions, relying entirely on self-generated data within the intrinsic self-correction to some extent. At the second stage, the baseline step-wise preference learning is leveraged via the application of the enhanced self-correct policy achieved at the first stage. In the evaluation of arithmetic reasoning tasks, our approach outperforms OpenMath2-Llama3.1-8B, dart-math-mistral-7b-uniform on MATH with increases in accuracy to 71.34%(+4.18%) and 48.06%(+4.94%) and LLama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.1 on GSM8K with increases in accuracy to 86.76%(+2.00%) and 38.06%(+2.28%).

Introduction

The integration of MCTS (Coulom 2006; Kocsis and Szepesvári 2006), neural networks and RL techniques (Silver et al. 2017) has been successfully developed since AlphaZero (Silver et al. 2017) contributing to its superhuman performance across various domains. MCTS has subsequently been integrated as a policy improvement operator, transforming the current policy into an enhanced one (Grill et al. 2020). Moreover, integrating MCTS into the iterative process of policy development could lead to significant advancements in LLMs, especially in areas such as reasoning and decision-making that align with human-like preferences (Zhu et al. 2022; Hao et al. 2023). The instance-level approach utilizes sparse supervision, which may overlook important information and fail to fully exploit the potential of MCTS in enhancing LLMs (Wu et al. 2023). Another challenge is MCTS's dependence on a critic or a learned reward function, which is essential for providing meaningful feedback on the various rollouts generated by MCTS (Liu et al. 2023). To address this granularity issue, research in LLMs suggests that process-level or stepwise evaluations are superior to instance-level evaluations (Lightman et al. 2023; Li et al. 2023; Xie et al. 2023; Yao et al. 2024; Rafailov et al. 2024).

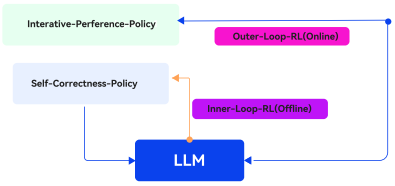

Figure 1: Overview of Towards Intrinsic Self-Correction Enhancement via Iterative Preference Learning. It's consisted of training two policies, self-correctness-policy in the inner-loop reinforcement learning and outer-loop-policy in the outer-loop reinforcement learning. Here, the purple box denotes the learned policy for the first stage. The pink box denotes the learned policy for the second stage.

However, the current iterations of step-level MCTS, LLMs, and RL techniques lack robust self-correction verification. In order to enhance the ability of self-correction of large langugae models(LLM), we therefore propose an improvement to the current state-of-the-art step-level MCTS-DPO (Xie et al. 2024) method. Specifically, we enhance the self-correction ability of LLMs via reinforcement learning (Figure 1) to play as the reward model of themselves in the baseline framework (Xie et al. 2024). Besides, our approach outperforms the OpenMath-Llama-3.1-8B(Toshniwal et al. 2024a), dart-math-mistral-7b-uniform (Tong et al. 2024) on MATH (Hendrycks et al. 2021) with increases in accuracy to 71.34%(+4.18%) and 48.06%(+4.94%) and LLama-3.1-8B-Instruct (Llama Team 2024), Mistral-7B-Instruct-v0.1 (Yu et al. 2023) on GSM8k (Cobbe et al. 2021) with increase in accuracy to 86.76%(+2.00%) and 38.06%(+2.28%)

Related Works

Among the current works aimed at enhancing reasoning abilities in LLMs, the following are the most closely related.

Self-correcting LLMs

Previous studies examine self-correction in LLMs across various assumptions and problem settings. Ground-truth answers are applied (Kim, Baldi, and McAleer 2024; Shinn, Labash, and Gopinath 2023) during self-correction. Weak prompts (Madaan et al. 2024) are applied for initial responses overestimating the total improvement possible. Then, access to a reward function for evaluating model are used to generate outputs (Akyürek et al. 2023; Welleck et al. 2022; Zhang et al. 2024; Qu et al. 2024). Separate models are trained to perform correction (Havrilla et al. 2024; Welleck et al. 2022; Akyürek et al. 2023; Paul et al. 2023). Finally, the intersection of LLMs and multi-turn RL builds machinery for optimizing rewards with value-based (Farebrother et al. 2024; Shani et al. 2024; Snell et al. 2022; Zhou et al. 2024), policy-based (Shao et al. 2024; Xiong et al. 2024), and model-based (Hong, Lee, and Thorne 2024)(self-correct) approaches. However, there has been little exploration of step-level integration between self-correcting LLMs and MCTS, for the enhancement of reasoning in intermediate steps through reinforcement learning.

Fusion of MCTS and LLMs

Current researches on combining MCTS and LLMs offer benefits for enhancing reasoning capabilities. The integration of MCTS as a policy improvement operator is built to transform the current policy into an improved policy (Grill et al. 2020). Integrating MCTS into the iterative policy development process could lead to substantial progress in LLMs, especially in areas such as reasoning and decision-making that align with human-like preferences(Zhou et al. 2024; Hao et al. 2023). The instance-level approach relies on sparse supervision, which may overlook crucial information and fail to fully capitalize on MCTS's potential to enhance LLMs (Wu et al. 2023). And the reliance of MCTS on a critic or a learned reward function may lead to possible incorrect information (Liu et al. 2023). Many researches have suggested that stepwise evaluations are superior to instance-level evaluations (Lightman et al. 2023; Li et al. 2023; Xie et al. 2023; Yao et al. 2024; Rafailov et al. 2024). However, the self-evaluation applied in the above integration is not strong. We therefore enhance the ability of self-evaluation in the iteration via reinforcement learning.

Methodology

We therefore propose a two-stage framework for the self-correction enhancement on the current state-of-the-art step-level MCTS-LLM-DPO (Xie et al. 2024). At the first stage, we enhance the ability of self-correction of LLMs via self-generated data without any external feedback. At the second stage, we employ the enhanced LLM for the verification enhancement in the step-level preference learning.

Stage-I: Intrinsic Data Generation Based Self-Correct LLM

In the first stage, our objective is to train LLMs to refine their predictions using solely self-generated data within the towards intrinsic self-correction framework, where models aim to improve their initial responses. (Kumar et al. 2024).

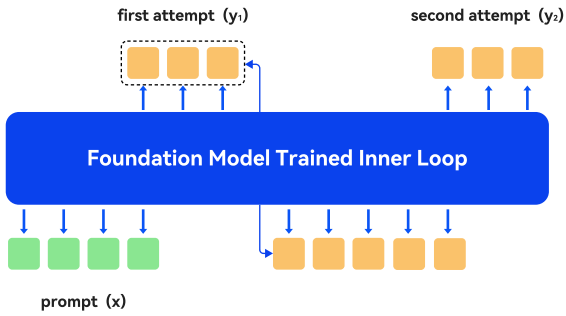

Figure 2: Towards intrinsic Self-Correct LLM in the Inner Loop (Stage I). Here, the green box denotes the input prompt for the LLM at the first stage. The orange box denotes the respondence of the first attempt given the prompt as the input. Then, for the second attempt, the large language model receives the respondence of the first attempt together with the green prompt as input and produces the response of the second attempt(orange box).

Concretely, given a dataset $D = \{(x_i, y_i^*)\}_{i=1}^N$ of problems x_i and response y_i^*, an LLM policy $\pi_\theta(\cdot|[x, \hat{y}_{1:l}, p_{1:l}])$ that, given the problem x, previous l model attempts $\hat{y}_{1:l}$ at the problem, and auxiliary instructions $p_{1:l}$(eg, instruction to find a mistake and improve the response), solves the problem x as correctly as possible. This formalism is akin to the multi-turn MDP in (Qu et al. 2024). We also assume access to an oracle reward $\hat{r}(y, y^*)$, such as an answer checker (Uesato et al. 2022), that evaluates the correctness of response y by comparing it with the oracle response y^*.

At this stage, we aim to find an LLM policy $\pi(\square \mid \circ)$ mapping input tokens $\circ$ to output tokens $\square$ that maximizes the correctness reward obtained from the vertifier at the end of $l+1$ turns. Formally:

$$\max_{\pi_{\theta_1}} \xi_{x,y^*\sim\mathcal{D},\hat{y}_{l+1}\sim\pi_{\theta_1}(\cdot|[x,\hat{y}_{1:l},p_{1:l}])}[\sum_{i=1}^{l+1} \hat{r}(\hat{y}_i, y^*)] \quad (1)$$

π_{θ_1} is trained over multiple attempts simultaneously, where intermediate turns are supervised indirectly to maximize the sum. We apply a REINFORCE policy gradient training approach (Ahmadian et al. 2024) with a KL-divergence penalty against a fix model.

Stage-II:Step-level Iterative Preference Learning

At the second stage, we employ step-level iterative preference learning with enhanced self-correction LLM (achieved at the first stage) to enhance the step-wise verification via self-correction.

We define the state at step t, s_t as the prefix of the reasoning chain, a as the corresponding action, with the addi-

Table 1: Based on the Llama-3.1-8B-Instruct and Mistral-7B-Instruct models, the performance of our method and other methods on GSM8K.

Base Model	Approach	Acc(%)
Llama-3.1-8B-Instruct	Baseline	84.76
	Intrinsic Self-Correct	86.05
	Base Model + MCTS-DPO	85.67
	Ours	**86.76**
Mistral-7B-Instruct	Baseline	35.78
	Intrinsic Self-Correct	37.45
	Base Model + MCTS-DPO	35.71
	Ours	**38.06**

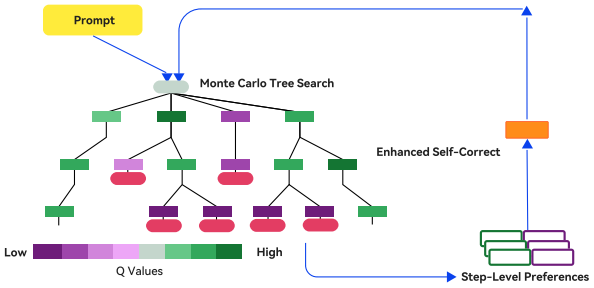

Figure 3: Step-wise Iterative Preference Learning in the Outer Loop. The red circle denotes the termination, the square denotes the intermediate node (Stage II). The orange box denotes the policy learned at the first stage. The Monte Carlo Tree and the step-level preference learning both represent two parts of the iterative preference learning via boosted MCTS.

.

tion of a new reasoning step transitioning the state to s_{t+1}, where s_{t+1} is the concatenation of s_t and a. Utilizing the model's current policy π_{θ_2}, we sample candidate steps from its probability distribution $\pi_{\theta_2}(a \mid x, s_t)$, with x being the task's input prompt.

Then, the MCTS process begins from a root node, s_0, as the sentence start or incomplete response, and unfolds in four iterative stages: selection, expansion, enhanced-self-verify, and backup. The four details are represented as the following:

Select The objective of this phase (Xie et al. 2024) is to identify nodes that balance search quality and computational efficiency. The selection is guided by two key variables: $Q(s_t, a)$, the value of taking action a in state s_t , and $N(s_t)$, the visitation frequency of state s_t. These variables are crucial for updating the search strategy, as explained in the backup section. To navigate the trade-off between exploring new nodes and exploiting visited ones, we employ the Predictor + Upper Confidence bounds applied to Trees (PUCT) (Rosin 2011). At node s_t, the choice of the subsequent node follows the formula:

$$s_{t+1}^{*} = \arg\max_{s_t}[\mathcal{Q}(s_t, a) + c_{puct} \cdot p(a \mid s_t)]\frac{\sqrt{N(s_t)}}{1 + N(s_{t+1})} \tag{2}$$

where $p(a \mid s_t) = \pi_{\theta_2}(a \mid x, s_t)/|a|^{\lambda}$ denotes the policy π_{θ_2}'s probability distribution for generating a step a, adjusted by a λ-weighted length penalty to prevent overly long reasoning chains.

Expand Expansion occurs at a leaf node during the selection process to integrate new nodes and access rewards (Xie et al. 2024). The reward $r(s_t, a)$ for executing step a in state s_t is quantified by the reward difference between $R(s_t)$ and $R(s_{t+1})$, highlighting the advantage of action a at s_t. As defined below, reward computation merges outcome correctness $\mathcal{O}$ with self-evaluation $\mathcal{C}$. We assign the outcome correctness to be 1, -1 and 0 for correct terminal, incorrect terminal, and unfinished intermediate states, respectively. Following (Xie et al. 2023), we define self-evaluation as the following equation, where A denotes the confidence score in token-level probability for the option indicating correctness. $prompt_{eval}$ denotes the prompt applied by the verified LLM. Future rewards are anticipated by simulating upcoming scenarios through roll-outs, following the selection and expansion process until reaching a terminal state.

$$\mathcal{R}(s_t) = \mathcal{O}(s_t) + \mathcal{C}(s_t) \tag{3}$$

$$\mathcal{C}(s_t) = \pi_{\theta_2}(A \mid prompt_{eval}, x, s_t) \tag{4}$$

Enhanced-Self-Verify After each expanding step, we conduct a self-correct to further correct possible misleading or even wrong reasoning generated intermediate outcome, the self-correction is provided from the same LLM achieving the policy π_{θ_1} at the first stage.

$$\mathcal{R}(s_t^{correct}) = \mathcal{R}(s_t) + \hat{\mathcal{C}}(s_t) \tag{5}$$

$$\hat{\mathcal{C}}(s_t)^{stage_1} = \pi_{\theta_1}(A|prompt_{eval}, x, s_t) \tag{6}$$

Backup Once a terminal state is reached, we carry out a bottom-up update from the terminal node back to the root. We update the visit count N, the state value V and the transition value Q which remains the same.

Table 2: Based on the OpenMath2-Llama3.1-8B and dart-math-mistral-7b-uniform models, the performance of our method and other methods on MATH

Base Model	Approach	Acc(%)
OpenMath2-Llama3.1-8B	Baseline	67.16
	Intrinsic Self-Correct	69.78
	Base Model + MCTS-DPO	68.06
	Ours	**71.34**
dart-math-mistral-7b-uniform	Baseline	43.42
	Intrinsic Self-Correct	44.50
	Base Model + MCTS-DPO	45.56
	Ours	**48.06**

Table 3: Ablation experiments: the performance of different models to initialize the policy model and reward model on MATH.

Policy model	Reference model	Reward model	Acc(%)
OpenMath2-Llama3.1-8B	OpenMath2-Llama3.1-8B	OpenMath2-Llama3.1-8B	68.06
OpenMath2-Llama3.1-8B	OpenMath2-Llama3.1-8B	SPL-Model	67.06
SPL-Model	SPL-Model	SPL-Model	**71.34**
ISC-Model	ISC-Model	SPL-Model	65.34
dart-math-mistral-7b-uniform	dart-math-mistral-7b-uniform	dart-math-mistral-7b-uniform	45.56
dart-math-mistral-7b-uniform	dart-math-mistral-7b-uniform	SPL-Model	46.84
SPL-Model	SPL-Model	SPL-Model	**48.06**
ISC-Model	ISC-Model	SPL-Model	47.22

Experiments

We evaluate the effectiveness of DPO by MCTS and iterative preference learning on GSM8K and MATH reasoning tasks. We use the Llama-3.1-8B-Instruct(Dubey et al. 2024), Mistral-7B-Instruct-v0.1 (Jiang et al. 2023) as the base model on GSM8K (Cobbe et al. 2021) and OpenMath2-Llama3.1-8B (Toshniwal et al. 2024b), dart-math-mistral-7b-uniform(Tong et al. 2024) as the base model on MATH (Hendrycks et al. 2021). Iterative Preference Learning(Kumar et al. 2024) combined with online MCTS-DPO(Xie et al. 2024) work well.

Datasets

We aim to demonstrate the effectiveness and versatility of our approach by focusing on arithmetic reasoning. We utilized two datasets: GSM8K (Cobbe et al. 2021), which consists of grade school math word problems, and MATH (Hendrycks et al. 2021), featuring challenging competition math problems.

Main Results

Our results on GSM8K and MATH are shown in Table 1 and Table 2. Our method achieved increases on accuracy across all four LLMs and two datasets. With towards intrinsic self-correct training, we obtained a model capable of self-correction. We further boosted LLMs performance on these datasets through iterative preference learning. Specifically, on GSM8K, we achieved 2.00% and 2.28% increases in accuracy for Llama-3.1-8B-Instruct and Mistral-7B-Instruct respectively. On MATH, we observed improvements of 4.18% and 4.64% for OpenMath2-Llama3.1-8B and dart-math-mistral-7b. All of these results demonstrate the effectiveness of our approach in enhancing LLM performance across datasets with various difficulties.

Ablation Studies

To demonstrate the effectiveness of Towards Intrinsic Self-Correct and Step-level Iterative Preference Learning, we need to verify that neither method alone can achieve the same performance as their combination. Since GSM8K is relatively simple and easier for models to learn, its persuasiveness as a test case is limited. Therefore, we conducted our verification on the more challenging MATH dataset. For clarity, we denote the models generated by Intrinsic Self-Correct and Step-level Iterative Preference Learning as ISC-Model and SPL-Model, respectively. As shown in Table3, the results indicate that the combination where both the policy model and the reward model are SPL-Model consistently outperforms other configurations. For OpenMath2-Llama3.1-8B, all SPL-Model setting achieves 71.34% in accuracy surpasses other configuration by 3.28% to 6.00%. Similarly, for dart-math-mistral-7b-uniform, all SPL-Model configurations consistently outperform other settings by 0.84% to 2.50%. All of these results demonstrate the superior synergy of the two methods.

Discussion

We start our step-level and self-correct level learning for enhancing the ability of reasoning for LLMs. At the first stage, we integrate an enhanced self-correct model via self-supervised learning with step-level preference learning. Then, we are going to begin our second stage which improves the step-level preference and self-correction via online reinforcement learning. Finally, we are planning to conduct experiments on other reasoning dataset.

References

Ahmadian, A.; Cremer, C.; Gallé, M.; Fadaee, M.; Kreutzer, J.; Pietquin, O.; Üstün, A.; and Hooker, S. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*.

Akyürek, A. F.; Akyürek, E.; Madaan, A.; Kalyan, A.; Clark, P.; Wijaya, D.; and Tandon, N. 2023. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs. *arXiv preprint arXiv:2305.08844*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Coulom, R. 2006. Efficient selectivity and backup operators in Monte-Carlo tree search. In *International conference on computers and games*, 72–83. Springer.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Farebrother, J.; Orbay, J.; Vuong, Q.; Taïga, A. A.; Chebotar, Y.; Xiao, T.; Irpan, A.; Levine, S.; Castro, P. S.; Faust, A.; et al. 2024. Stop regressing: Training value functions via classification for scalable deep rl. *arXiv preprint arXiv:2403.03950*.

Grill, J.-B.; Altché, F.; Tang, Y.; Hubert, T.; Valko, M.; Antonoglou, I.; and Munos, R. 2020. Monte-Carlo tree search as regularized policy optimization. In *International Conference on Machine Learning*, 3769–3778. PMLR.

Hao, S.; Gu, Y.; Ma, H.; Hong, J. J.; Wang, Z.; Wang, D. Z.; and Hu, Z. 2023. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*.

Havrilla, A.; Raparthy, S.; Nalmpantis, C.; Dwivedi-Yu, J.; Zhuravinskyi, M.; Hambro, E.; and Raileanu, R. 2024. Glore: When, where, and how to improve llm reasoning via global and local refinements. *arXiv preprint arXiv:2402.10963*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hong, J.; Lee, N.; and Thorne, J. 2024. Reference-free monolithic preference optimization with odds ratio. *arXiv e-prints*, arXiv–2403.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Kim, G.; Baldi, P.; and McAleer, S. 2024. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36.

Kocsis, L.; and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, 282–293. Springer.

Kumar, A.; Zhuang, V.; Agarwal, R.; Su, Y.; Co-Reyes, J. D.; Singh, A.; Baumli, K.; Iqbal, S.; Bishop, C.; Roelofs, R.; et al. 2024. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.

Li, Y.; Lin, Z.; Zhang, S.; Fu, Q.; Chen, B.; Lou, J.-G.; and Chen, W. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5315–5333.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Liu, J.; Cohen, A.; Pasunuru, R.; Choi, Y.; Hajishirzi, H.; and Celikyilmaz, A. 2023. Making ppo even better: Value-guided monte-carlo tree search decoding. *arXiv preprint arXiv:2309.15028*.

Llama Team, A. . M. 2024. The Llama 3 Herd of Models. *Journal of Mathematical Physics*, 36(92).

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Paul, D.; Ismayilzada, M.; Peyrard, M.; Borges, B.; Bosselut, A.; West, R.; and Faltings, B. 2023. Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904*.

Qu, Y.; Zhang, T.; Garg, N.; and Kumar, A. 2024. Recursive introspection: Teaching language model agents how to self-improve. *arXiv preprint arXiv:2407.18219*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Rosin, C. D. 2011. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3): 203–230.

Shani, L.; Rosenberg, A.; Cassel, A.; Lang, O.; Calandriello, D.; Zipori, A.; Noga, H.; Keller, O.; Piot, B.; Szpektor, I.; et al. 2024. Multi-turn Reinforcement Learning from Preference Human Feedback. *arXiv preprint arXiv:2405.14655*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shinn, N.; Labash, B.; and Gopinath, A. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366*, 2(5): 9.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Snell, C.; Kostrikov, I.; Su, Y.; Yang, M.; and Levine, S. 2022. Offline rl for natural language generation with implicit language q learning. *arXiv preprint arXiv:2206.11871*.

Tong, Y.; Zhang, X.; Wang, R.; Wu, R.; and He, J. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *arXiv preprint arXiv:2407.13690*.

Toshniwal, S.; Du, W.; Moshkov, I.; Kisacanin, B.; Ayrapetyan, A.; and Gitman, I. 2024a. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560*.

Toshniwal, S.; Du, W.; Moshkov, I.; Kisacanin, B.; Ayrapetyan, A.; and Gitman, I. 2024b. OpenMathInstruct-2: Accelerating AI for Math with Massive Open-Source Instruction Data. *arXiv preprint arXiv:2410.01560*.

Uesato, J.; Kushman, N.; Kumar, R.; Song, F.; Siegel, N.; Wang, L.; Creswell, A.; Irving, G.; and Higgins, I. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Welleck, S.; Lu, X.; West, P.; Brahman, F.; Shen, T.; Khashabi, D.; and Choi, Y. 2022. Generating sequences by learning to self-correct. *arXiv preprint arXiv:2211.00053*.

Wu, Z.; Hu, Y.; Shi, W.; Dziri, N.; Suhr, A.; Ammanabrolu, P.; Smith, N. A.; Ostendorf, M.; and Hajishirzi, H. 2023. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36: 59008–59033.

Xie, Y.; Goyal, A.; Zheng, W.; Kan, M.-Y.; Lillicrap, T. P.; Kawaguchi, K.; and Shieh, M. 2024. Monte Carlo Tree Search Boosts Reasoning via Iterative Preference Learning. *arXiv preprint arXiv:2405.00451*.

Xie, Y.; Kawaguchi, K.; Zhao, Y.; Zhao, X.; Kan, M.-Y.; He, J.; and Xie, Q. 2023. Decomposition enhances reasoning via self-evaluation guided decoding. *arXiv preprint arXiv:2305.00633*, 2.

Xiong, W.; Shi, C.; Shen, J.; Rosenberg, A.; Qin, Z.; Calandriello, D.; Khalman, M.; Joshi, R.; Piot, B.; Saleh, M.; et al. 2024. Building math agents with multi-turn iterative preference learning. *arXiv preprint arXiv:2409.02392*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Yu, L.; Jiang, W.; Shi, H.; Yu, J.; Liu, Z.; Zhang, Y.; Kwok, J. T.; Li, Z.; Weller, A.; and Liu, W. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Zhang, Y.; Khalifa, M.; Logeswaran, L.; Kim, J.; Lee, M.; Lee, H.; and Wang, L. 2024. Small Language Models Need Strong Verifiers to Self-Correct Reasoning. *arXiv preprint arXiv:2404.17140*.

Zhou, Y.; Zanette, A.; Pan, J.; Levine, S.; and Kumar, A. 2024. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*.

Zhu, X.; Wang, J.; Zhang, L.; Zhang, Y.; Gan, R.; Zhang, J.; and Yang, Y. 2022. Solving math word problems via cooperative reasoning induced language models. *arXiv preprint arXiv:2210.16257*.