

---

# EMBEDDING DISTANCE AS A REWARD SIGNAL CAN REPLACE VERIFIERS FOR LLM REASONING

Abdelhakim Benechehab<sup>\*,1,2</sup>, Youssef Attia El Hili<sup>1</sup>, Albert Thomas<sup>1</sup>, Giuseppe Paolo<sup>3</sup>, Maurizio Filippone<sup>4</sup>

<sup>1</sup> Huawei Noah’s Ark Lab, Paris, France

<sup>2</sup> Department of Data Science, EURECOM

<sup>3</sup> Cognizant AI, Paris

<sup>4</sup> Statistics Program, KAUST

\* Correspondence: [abdelhakim.benechehab@gmail.com](mailto:abdelhakim.benechehab@gmail.com)

## ABSTRACT

Reinforcement Learning (RL) has emerged as a powerful paradigm for adapting Large language Models (LLM), offering advantages over Supervised Fine-tuning (SFT) including reduced catastrophic forgetting and improved generalization. However, these benefits require explicit reward signals, often obtained from human preferences or verifiable outcomes, which are unavailable in many cases. We address this gap by introducing a framework that derives reward functions directly from supervised data, enabling RL-based training without additional annotation. Our approach formulates reward functions as a weighted distance between embeddings of labels and generated answers. Experiments with LLM fine-tuning for a reasoning task demonstrate that our learned rewards match the performance of oracle RL that has access to ground-truth rewards.

## 1 INTRODUCTION

Generative models have become central to modern machine learning research, driving advances in text (Brown et al., 2020; DeepSeek-AI et al., 2025), image (Rombach et al., 2021; Ramesh et al., 2021), and multimodality (Zhang et al., 2024; Bai et al., 2025; Fu et al., 2025; Łajszczak et al., 2024; Yin et al., 2024) under the umbrella of “Generative AI” (*GenAI*). Their ability to synthesize realistic content has made them foundational in applications ranging from decision making (Shi et al., 2025; Kim et al., 2024; Intelligence et al., 2025) to scientific discovery (Manica et al., 2023; Lu et al., 2024).

Traditionally, such models are trained via Maximum Likelihood Estimation (MLE), where the parameters of the generative model are optimized to maximize the probability of observed data. This approach provides a principled framework for fitting models to large datasets and remains the backbone of many learning pipelines for generative models. Notably, this approach is omnipresent in today’s *Large Language Models* (LLMs) through the *next token prediction* paradigm (Vaswani et al., 2023; Brown et al., 2020; DeepSeek-AI et al., 2025).

However, recent breakthroughs in LLMs research, demonstrate the limitations of MLE alone. Techniques based on RL (Bellman, 1958), such as *Reinforcement Learning from Human Feedback* (RLHF) (Christiano et al., 2017; Stiennon et al., 2020) and more recently *Reinforcement Learning from Verifiable Rewards* (Shao et al., 2024; DeepSeek-AI et al., 2025), have proven more effective than SFT at aligning models with human preferences and improving generation quality (Shenfeld et al., 2025; Lai et al., 2025; Swamy et al., 2025). These methods leverage explicit or implicit reward signals to guide training beyond likelihood objectives.

In many real-world scenarios, explicit reward functions for the tasks we aim to solve are not readily available. Instead, we often have access to high-quality datasets that we wish to use for aligning our models. Depending on the structure of these datasets, several techniques have been proposed to derive reward functions, such as from preference data (Rafailov et al., 2023) or from demonstrations (Finn et al., 2016a;b) when framed within a Markov Decision Process (MDP) formalism. Despite these advances, the fundamental question surrounding this problem remains unresolved:

*How can we learn implicit reward functions from supervised data?*

---

In this paper, we propose the following contributions toward addressing this question:

- **Embedding distance Reward formulation:** We adopt a reward parametrization in the form of a negative scaled distance between embeddings of generative samples and targets. This formulation leads to dense reward functions available directly from supervised data.
- **Experimental results:** We evaluate our approach in a LLM finetuning reasoning task, where we find that we can reach equivalent performance to the oracle method that has access to the true reward function.

## 2 REWARD MODELING FRAMEWORK

Given the definition of a reward function as evaluating the quality of generative samples  $\hat{\mathbf{y}}$  against true labels  $\mathbf{y}$ , we assume that a valid reward has to be maximal in  $\mathbf{y}$ :  $\mathbf{y} \in \max_{\hat{\mathbf{y}}} r(\hat{\mathbf{y}}, \mathbf{y})$ . Therefore, we naturally define the reward function as a negative quadratic form of the difference between generative samples and true labels:

**Definition 2.1** (Reward model). *Let  $n$  be an integer, and  $\mathbf{U} \in S_n^{++}(\mathbb{R})$ , we define the reward model as the following quadratic form:*

$$\forall(\hat{\mathbf{y}}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n, \quad r_{\mathbf{U}}(\hat{\mathbf{y}}, \mathbf{y}) = -(\hat{\mathbf{y}} - \mathbf{y})^T \mathbf{U}(\hat{\mathbf{y}} - \mathbf{y}).$$

The matrix  $\mathbf{U}$  represents reward parameters, that can potentially be optimized to find optimal reward functions. In the case of complex data modalities such as images or text, we define a modified version where the reward measures distances in a precomputed embedding space:

**Definition 2.2** (Embedding reward model). *Let  $\mathbf{U} \in S_d^{++}(\mathbb{R})$  and an embedding model  $e: \mathbb{R}^n \rightarrow \mathbb{R}^d$ , we define the embedding reward model as the following quadratic form in the  $\mathbb{R}^d$  space:*

$$\forall(\hat{\mathbf{y}}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^n, r_{\mathbf{U}}(\hat{\mathbf{y}}, \mathbf{y}) = -(e(\hat{\mathbf{y}}) - e(\mathbf{y}))^T \mathbf{U}(e(\hat{\mathbf{y}}) - e(\mathbf{y})).$$

In the experimental section, we will assume that  $\mathbf{U} = \mathbf{I}_n$  as a simplified proof-of-concept, and will consider the optimization of the reward parameters in future follow-up works.

## 3 EXPERIMENTAL RESULTS

As discussed in the introduction, fine-tuning large models is the setting in which our method is expected to have the greatest impact. In the case of LLMs, we use the Group Relative Policy Optimization (GRPO) algorithm (Shao et al., 2024), which is specifically designed for LLMs and has demonstrated strong performance across a range of environments. The reward model is constructed using the embedding reward formulation (definition 2.2), where the embedding function  $e: \mathcal{Y} \rightarrow \mathbb{R}^d$  can be instantiated with any text embedding model, such as RoBERTa (Liu et al., 2020).

We focus on the Countdown task, in which an LLM must identify the correct algebraic operations applied to a given set of numbers (e.g.,  $nums = [93, 78, 46]$ ,  $target = 61$ , with the expected solution being  $93 - (78 - 46) = 61$ ). For this purpose, we rely on the HuggingFaceTB/Countdown-Task-GOLD dataset, which comprises 30k samples of annotated questions and answers, along with their corresponding reasoning traces: e.g. “<think> To solve this, I need to use the numbers 78, 46, and 93 to create an equation that equals 61. I’ll start by [...] </think><answer>93 - (78 - 46) = 61</answer>”. The oracle reward function for the Countdown task is defined as a binary signal, assigning a reward of 1 to correct equations that use all the required numbers, and 0 otherwise.

Fig. 1 reports the performance of several checkpoints extracted from training runs of 400 steps, evaluated on a held-out test set of 10k questions. We find that experiments using GRPO with our embedding-based reward function achieve performance comparable to that of the GRPO oracle baseline, which has access to the true reward. These results suggest that the proposed reward parametrization provides sufficient learning signal to fine-tune LLMs solely from annotated data, thereby extending the applicability of RL fine-tuning beyond strictly verifiable tasks.

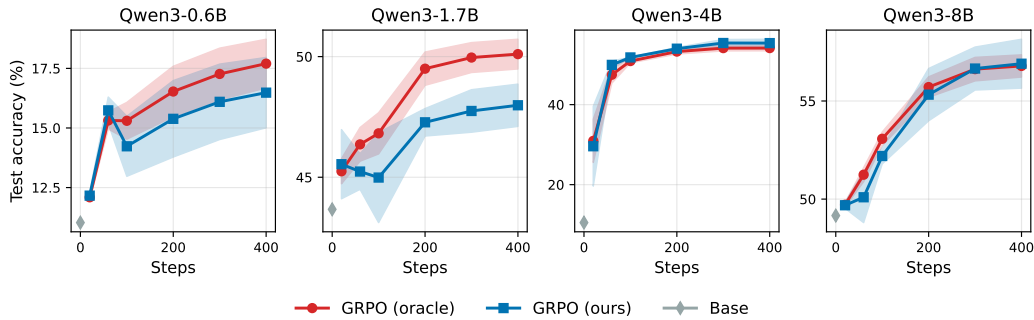


Figure 1: **Qwen3 fine-tuning using GRPO on the Countdown task.** Using our embedding reward models with  $I_n$  as parameters, and `EmbeddingGemma` as embedding model, we are able to achieve the same levels of accuracy as the oracle method that uses the true reward.

**Ablation study.** First, we note from Fig. 1 that the performance of our approach remains stable across different model sizes, indicating robustness with respect to the initial capability of the base model. In addition, we perform an ablation study examining the impact of the embedding model, as illustrated in the left panel of Fig. 2. We observe that the choice of the embedding model is indeed important as `EmbeddingGemma` and `RoBERTa` show the best performance compared to the less capable `MiniLM-v6` model. Finally, we investigate whether the embedding reward function should be computed using the `Full output` or only the `Final answer` in the right panel of Fig. 2. We conclude that our reward modeling framework strongly benefits from computing the reward only on the final answer, highlighting the importance of a structured output when computing distances in the embedding space.

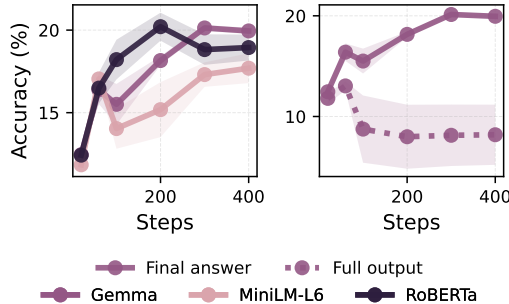


Figure 2: **Ablation.** For `Qwen3-0.6B`, the optimal performance is observed using the `EmbeddingGemma` embedding model, and only the final answer when computing rewards.

## 4 CONCLUSION

This paper studied how to find reward functions that, through distances in an embedding space can replace verifiers for LLM reasoning tasks. Our approach enables the definition of reward functions solely from supervised data, thus enabling RL fine-tuning when reward functions are not necessarily available. We then demonstrated the practical relevance of the approach in an experiment involving fine-tuning an LLM for the Countdown reasoning task. Finally, we showed that GRPO combined with rewards learned through our framework achieves test performance comparable to GRPO using the true task reward.

**Limitations & Future Work.** The reward parametrization considered in this study is intentionally simple, which may limit expressiveness in more complex tasks. An interesting direction for future work is to extend the framework to richer classes of parametric reward functions, and explore the optimization of these parameters to find optimal rewards. In addition, although we evaluated the method on a reasoning task, larger-scale experiments would help further characterize its behavior. Notably, our LLM experiments focus on a verifiable task where an oracle reward is available for comparison. A natural extension is to investigate non-verifiable tasks, where such rewards are unavailable, and to compare SFT with RL driven by rewards learned within our framework. We view these directions as promising avenues for future exploration.

---

## REPRODUCIBILITY STATEMENT

In order to ensure reproducibility we will release the code at [https://github.com/abenechehab/nll\\_to\\_po](https://github.com/abenechehab/nll_to_po), once the paper has been accepted. Implementation details and relevant hyperparameters are provided in Section 3 and Appendix C.

## REFERENCES

- Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning, ICML '04*, pp. 1, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015430. URL <https://doi.org/10.1145/1015330.1015430>.
- H Akaike. Information Theory and an Extension of the Maximum Likelihood Principle. In *2nd International Symposium on Information Theory, 1973*, pp. 268–281. Publishing House of the Hungarian Academy of Sciences, 1973.
- Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction, 2017. URL <https://arxiv.org/abs/1607.07086>.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL <https://arxiv.org/abs/2502.13923>.
- Richard Bellman. Dynamic programming and stochastic control processes. *Information and Control*, 1(3):228–239, 1958. ISSN 0019-9958. doi: [https://doi.org/10.1016/S0019-9958\(58\)80003-0](https://doi.org/10.1016/S0019-9958(58)80003-0). URL <https://www.sciencedirect.com/science/article/pii/S0019995858800030>.
- Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks, 2015. URL <https://arxiv.org/abs/1506.03099>.
- Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444, 14643510. URL <http://www.jstor.org/stable/2334029>.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL <https://arxiv.org/abs/2005.14165>.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf).
- Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=TyFrPOKYXw>.

---

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandolekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL [https://openreview.net/forum?id=rc8o\\_j8I8PX](https://openreview.net/forum?id=rc8o_j8I8PX).

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models, 2016a. URL <https://arxiv.org/abs/1611.03852>.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 49–58, New York, New York, USA, 20–22 Jun 2016b. PMLR. URL <https://proceedings.mlr.press/v48/finn16.html>.

Chaoyou Fu, Haojia Lin, Xiong Wang, Yi-Fan Zhang, Yunhang Shen, Xiaoyu Liu, Haoyu Cao, Zuwei Long, Heting Gao, Ke Li, Long Ma, Xiawu Zheng, Rongrong Ji, Xing Sun, Caifeng Shan, and Ran He. Vita-1.5: Towards gpt-4o level real-time vision and speech interaction, 2025. URL <https://arxiv.org/abs/2501.01957>.

Evan Greensmith, Peter Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. In T. Dietterich, S. Becker, and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL [https://proceedings.neurips.cc/paper\\_files/paper/2001/file/584b98aac2dddf59ee2cf19ca4ccb75e-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2001/file/584b98aac2dddf59ee2cf19ca4ccb75e-Paper.pdf).

Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical distance learning for semi-supervised and unsupervised skill discovery. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=H1l1mhaVtvr>.

- 
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=nZeVKeeFYf9>.
- Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky.  $\pi_{0.5}$ : a vision-language-action model with open-world generalization, 2025. URL <https://arxiv.org/abs/2504.16054>.
- Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model, 2024. URL <https://arxiv.org/abs/2406.09246>.
- Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Min Xie, Qingfu Zhang, Hongbin Liu, Gaofeng Meng, and Fei Zhu. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training, 2025. URL <https://arxiv.org/abs/2507.05386>.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}: A robustly optimized {bert} pretraining approach, 2020. URL <https://openreview.net/forum?id=SyxS0T4tvS>.
- Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.
- Matteo Manica, Jannis Born, Joris Cadow, Dimitrios Christofidellis, Ashish Dave, Dean Clarke, Yves Gaetan Nana Teukam, Giorgio Giannone, Samuel C. Hoffman, Matthew Buchan, Vijil Chenthamarakshan, Timothy Donovan, Hsiang Han Hsu, Federico Zipoli, Oliver Schilter, Akihiro Kishimoto, Lisa Hamada, Inkit Padhi, Karl Wehden, Lauren McHugh, Alexy Khrabrov, Payel Das, Seiji Takeda, and John R. Smith. Accelerating material design with the generative toolkit for scientific discovery. *npj Computational Materials*, 9(1), 2023. ISSN 2057-3960. doi: 10.1038/s41524-023-01028-1. URL <http://dx.doi.org/10.1038/s41524-023-01028-1>.
- Pietro Mazzaglia, Tim Verbelen, Bart Dhoedt, Aaron Courville, and Sai Rajeswar. Genrl: Multimodal-foundation world models for generalization in embodied agents, 2024. URL <https://arxiv.org/abs/2406.18043>.
- Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning, 2018. URL <https://arxiv.org/abs/1805.08296>.
- Mohammad Norouzi, Samy Bengio, zhifeng Chen, Navdeep Jaitly, Mike Schuster, Yonghui Wu, and Dale Schuurmans. Reward augmented maximum likelihood for neural structured prediction. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL [https://proceedings.neurips.cc/paper\\_files/paper/2016/file/2f885d0f8e2e131bfc9d98363e55d1d4-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2016/file/2f885d0f8e2e131bfc9d98363e55d1d4-Paper.pdf).
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155>.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In

- 
- Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=HPuSIXJaa9>.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/ramesh21a.html>.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks, 2016. URL <https://arxiv.org/abs/1511.06732>.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL’s razor: Why online reinforcement learning forgets less, 2025. URL <https://arxiv.org/abs/2509.04259>.
- Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models, 2025. URL <https://arxiv.org/abs/2502.19417>.
- Sumedh Anand Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Biyik, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. RoboCLIP: One demonstration is enough to learn robot policies. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=DVlawv2rSI>.
- Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS ’20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- Gokul Swamy, Sanjiban Choudhury, Wen Sun, Zhiwei Steven Wu, and J. Andrew Bagnell. All roads lead to likelihood: The value of reinforcement learning in fine-tuning, 2025. URL <https://arxiv.org/abs/2503.01067>.
- Bowen Tan, Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric Xing. Connecting the dots between mle and rl for sequence prediction, 2019. URL <https://arxiv.org/abs/1811.09740>.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL <https://arxiv.org/abs/2307.09288>.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL <https://arxiv.org/abs/1706.03762>.

- 
- Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 3024–3030. AAAI Press, 2015. ISBN 0262511290.
- Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL <https://github.com/huggingface/trl>.
- Yucen Wang, Rui Yu, Shenghua Wan, Le Gan, and De-Chuan Zhan. FOUNDER: Grounding foundation models in world models for open-ended embodied decision making. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=UTT50TyIWm>.
- R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12), November 2024. ISSN 2053-714X. doi: 10.1093/nsr/nwae403. URL <http://dx.doi.org/10.1093/nsr/nwae403>.
- Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey, 2024. URL <https://arxiv.org/abs/2304.00685>.
- Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=lq62uWRJjiY>.
- Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, David Z. Pan, Zhangyang Wang, Yuandong Tian, and Kai Sheng Tai. The path not taken: RLvr provably learns off the principals, 2025. URL <https://arxiv.org/abs/2511.08567>.
- Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pp. 1433–1438. AAAI Press, 2008. ISBN 9781577353683.
- Mateusz Łajszczak, Guillermo Cámara, Yang Li, Fatih Beyhan, Arent van Korlaar, Fan Yang, Arnaud Joly, Álvaro Martín-Cortinas, Ammar Abbas, Adam Michalski, Alexis Moinet, Sri Karlapati, Ewa Muszyńska, Haohan Guo, Bartosz Putrycz, Soledad López Gambino, Kayeon Yoo, Elena Sokolova, and Thomas Drugman. Base tts: Lessons from building a billion-parameter text-to-speech model on 100k hours of data, 2024. URL <https://arxiv.org/abs/2402.08093>.

---

# Appendix

## TABLE OF CONTENTS

<b>A Motivation</b>	<b>10</b>
<b>B Related Work</b>	<b>11</b>
<b>C Implementation Details</b>	<b>11</b>
C.1 Training Framework . . . . .	11
C.2 Model Architecture and Fine-tuning . . . . .	12
C.3 Dataset . . . . .	12
C.4 Training Hyperparameters . . . . .	14
C.5 GRPO-Specific Configuration . . . . .	14

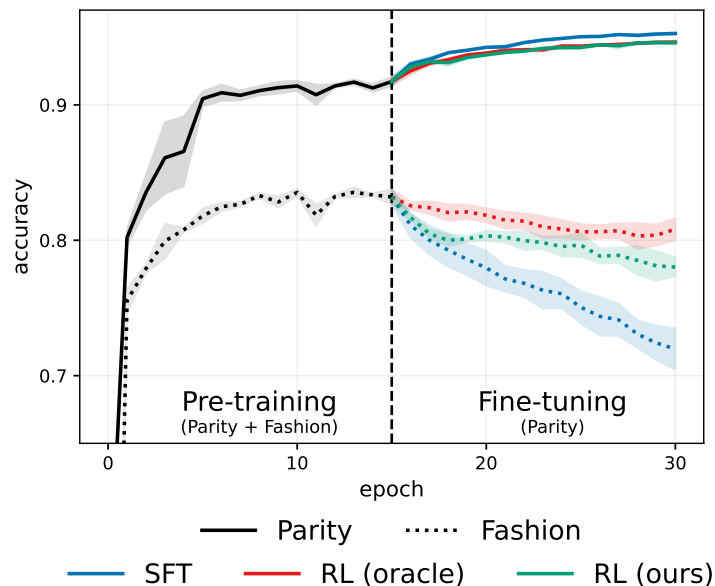


Figure 3: **RL using rewards learned from SFT data.** Our framework enables the learning of reward functions solely from the data presented to SFT, achieving a level of catastrophic forgetting mitigation comparable to RL with access to the oracle reward function.

## A MOTIVATION

In RL, Policy Gradient (PG) methods are traditionally viewed as producing unbiased yet high-variance gradient estimates, especially in long-horizon or high-dimensional tasks (Greensmith et al., 2001). In contrast, MLE has historically served as the dominant paradigm in supervised learning and probabilistic modeling (Akaike, 1973). However, in the current era of large pretrained models and advanced RL algorithms, these limitations have become less restrictive, giving rise to many cases where PG methods are more advantageous than MLE.

*Catastrophic forgetting* is an area where these advantages are the most apparent. When adapting LLM to downstream tasks through post-training, it is often desirable to preserve prior knowledge while specializing to new distributions. Recent studies (Shenfeld et al., 2025; Lai et al., 2025; Swamy et al., 2025; Zhu et al., 2025) suggest that on-policy RL fine-tuning achieves this balance more effectively than SFT, since its updates converge to solutions closest in KL divergence to the original policy.

To elucidate the motivation behind our approach, we conduct an experiment to demonstrate how our method can mitigate the issue of catastrophic forgetting. Specifically, we replicate the experimental setup from section 4 of Shenfeld et al. (2025), where the *ParityMNIST* task is introduced. This task involves predicting the parity of the number in the input image, where the predicted parity corresponds to the parity of the predicted class ( $0 \rightarrow 9$ ). The experiment begins with a pre-training phase, during which an MLP model is trained on data from both *ParityMNIST* and *FashionMNIST*, where the latter consists in classifying 10 different cloth types. Subsequently, we run a fine-tuning phase focused solely on the *ParityMNIST* task. During this phase, we compare SFT (cross-entropy minimization) and RL (based on the REINFORCE algorithm) with either of the oracle parity reward or the reward function derived from our framework. Given that the model outputs logits for 10 classes, multiple optimal policies exist for *ParityMNIST*, e.g. mapping even  $\rightarrow 0$  and odd  $\rightarrow 1$ , or even  $\rightarrow 0, 2, 4, 6, 8$  and odd  $\rightarrow 1, 3, 5, 7, 9$ . As shown in Fig. 3, by using the same data as SFT, and assuming that the true reward function is unknown, we are able to learn a reward function through our framework, achieving a similar reduction in catastrophic forgetting as the RL oracle variant, which has access to the optimal parity reward. These observations motivate our approach: we propose a general framework that interprets data signals as reward functions, thereby also enabling RL methods.

---

## B RELATED WORK

**RL vs SFT.** The tension between maximum likelihood estimation and reinforcement learning-based training has a long history in sequence modeling. Early work identified fundamental limitations of MLE in autoregressive models, including compounding errors and exposure bias (Bengio et al., 2015; Venkatraman et al., 2015; Ranzato et al., 2016; Bahdanau et al., 2017; Tan et al., 2019). These observations motivated policy gradient alternatives such as REINFORCE-based training (Williams, 1992), Reward-Augmented Maximum Likelihood (Norouzi et al., 2016), and scheduled transitions from MLE to RL (Ranzato et al., 2016). More recently, this debate has intensified in the context of large language model fine-tuning. Shenfeld et al. (2025), Lai et al. (2025), and Swamy et al. (2025) provide theoretical and empirical evidence that on-policy RL fine-tuning mitigates catastrophic forgetting more effectively than supervised fine-tuning, as RL updates converge to solutions closest in KL divergence to the original policy. Complementing these findings, Zhu et al. (2025) demonstrate that RL with verifiable rewards provably explores reasoning paths that supervised fine-tuning cannot reach. Our work addresses a key bottleneck in leveraging these RL advantages: the need for explicit reward functions.

**Reward modeling.** Existing approaches to reward learning typically require structured supervision beyond standard labeled data. Learning from preferences (Bradley & Terry, 1952; Ouyang et al., 2022; Rafailov et al., 2023) transforms pairwise comparisons into reward models and has become standard for LLM post-training (Touvron et al., 2023; Dai et al., 2024; Shao et al., 2024; DeepSeek-AI et al., 2025). However, this requires costly preference annotations. Inverse RL methods (Abbeel & Ng, 2004; Ziebart et al., 2008; Finn et al., 2016a;b) learn rewards under which expert demonstrations are optimal, assuming access to state-action trajectories within an MDP formalism. Goal-conditioned approaches define rewards via spatial (Nachum et al., 2018; Mazzaglia et al., 2024), temporal (Hartikainen et al., 2020; Wang et al., 2025), or semantic (Sontakke et al., 2023; Fan et al., 2022) distances to target states. Our framework complements these paradigms by deriving rewards from supervised data without requiring preferences or an MDP structure.

## C IMPLEMENTATION DETAILS

### C.1 TRAINING FRAMEWORK

We implement our reinforcement learning framework using the Transformer Reinforcement Learning (TRL) library von Werra et al. (2020), which provides efficient implementations of modern RLHF algorithms. Specifically, we use the Group Relative Policy Optimization (GRPO) algorithm Shao et al. (2024) through the GRPOTrainer class, which enables stable policy optimization through group-based advantage normalization.

GRPO is a policy gradient algorithm designed to improve the stability and sample efficiency of RL finetuning of LLMs. Unlike traditional policy optimization methods that compute advantages against a global baseline, GRPO employs a group-based normalization scheme that compares multiple generations from the same prompt.

For each training prompt  $x$ , GRPO generates  $K$  independent completions  $\{y_1, y_2, \dots, y_K\}$  using the current policy  $\pi_\theta$ . Each completion is evaluated using a reward function  $R(x, y_i)$ , and the advantages are computed relative to the group mean:

$$A_i = \frac{R(x, y_i) - \mu_{\text{group}}}{\sigma_{\text{group}} + \epsilon}$$

where

$$\mu_{\text{group}} = \frac{1}{K} \sum_{j=1}^K R(x, y_j), \quad \sigma_{\text{group}} = \sqrt{\frac{1}{K} \sum_{j=1}^K (R(x, y_j) - \mu_{\text{group}})^2}$$

This group-based normalization provides several benefits: (1) it reduces variance by comparing samples from the same distribution, (2) it automatically adapts to the difficulty of each prompt, and (3) it mitigates reward hacking by preventing the policy from exploiting absolute reward values.

The policy is updated by maximizing the following objective function (omitting technical details such as likelihood ratios and clipping):

$$\mathcal{L}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)} [A(x, y) \log \pi_\theta(y|x) - \beta \cdot d_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})]$$

where  $\pi_{\text{ref}}$  is the reference policy (typically the supervised fine-tuned model before RL training),  $\beta$  is the KL penalty coefficient that controls the deviation from the reference policy, and  $d_{\text{KL}}$  is the Kullback-Leibler divergence. The KL penalty serves as a regularization term that prevents catastrophic forgetting of the pretrained knowledge and ensures stable training dynamics.

## C.2 MODEL ARCHITECTURE AND FINE-TUNING

We evaluate multiple model sizes from the Qwen3 family [Yang et al. \(2025\)](#), ranging from 0.6B to 8B parameters, to analyze the impact of model scale on reinforcement learning performance. For parameter-efficient fine-tuning, we employ Low-Rank Adaptation (LoRA) [Hu et al. \(2022\)](#) with the configuration detailed in [Table 1](#).

Table 1: LoRA Configuration for Parameter-Efficient Fine-tuning

Parameter	Value
LoRA Rank ( $r$ )	8
LoRA Alpha ( $\alpha$ )	16
LoRA Dropout	0.05
Target Modules	q_proj, v_proj
Trainable Parameters	~0.1% of base model

The LoRA modules are applied exclusively to the query and value projection matrices in the attention mechanism, following recent findings that selective module targeting can achieve comparable performance to full fine-tuning while significantly reducing memory overhead [Zhang et al. \(2023\)](#).

## C.3 DATASET

We conduct our experiments on the HuggingFaceTB/Countdown-Task-GOLD dataset, a mathematical reasoning benchmark that requires models to construct arithmetic equations to reach target values using a given set of numbers. We specifically use the subset verified by Qwen2.5-7B-Instruct, which ensures groundtruth annotations. The dataset is split into 90% training and 10% validation sets.

Each training example follows a structured chat format with a system prompt that establishes the reasoning paradigm, followed by a user query specifying the task. An annotated example prompt is shown below:

```
[
  {
    "content": "You are a helpful assistant. You first think about the reasoning process in the mind and then provide the user with the answer.",
    "role": "system"
  },
  {
    "content": "Using the numbers [78, 46, 93], create an equation that equals 61. You can use basic arithmetic operations (+, -, *, /) and each number can only be used once. Show your work"
```

---

```

        in <think> </think> tags. And return the
        final equation and answer in <answer> </answer>
        tags, for example <answer> (1 + 2) / 3 = 1
        </answer>.",
    "role": "user"
  },
  {
    "content": "<think>
To solve this, I need to use the numbers 78,
46, and 93 to create an equation that equals
61. I'll start by considering the basic
arithmetic operations and how they can be used
to get close to 61.

First, I'll try subtraction, as it's the most
straightforward operation to get close to 61:
- 78 - 46 = 32 (too low)
- 93 - 46 = 47 (closer but still too low)
- 93 - 78 = 15 (too low)

Next, I'll try addition and subtraction in
combination:
- 78 - 93 = -15 (too low, and I need to add
something positive)
- 46 + 93 = 139 (too high, and I need to
subtract something)

Since 93 is the highest number, I'll try to
use it in a subtraction:
- 93 - 46 = 47 (still too low, I need to
add 14)
- 78 - 46 = 32 (still too low, I need to
add 29)

I can use 78 and 46 to get 32, and then add
29 to get 61. 29 can be derived from 93 - 64,
but since I can only use each number once, I
need to find a way to use 93 directly.

The correct combination is:
93 - (78 - 46) = 93 - 32 = 61

This uses each number exactly once and the
result is 61.
</think>
<answer>
93 - (78 - 46) = 61
</answer>",
    "role": "assistant"
  }
]

```

The structured output format with `<think>` and `<answer>` tags enables clear separation between the reasoning trace and the final solution, facilitating both automated evaluation and reward computation during reinforcement learning. This format allows our reward functions to selectively evaluate either the complete response or only the final answer, depending on the experimental configuration (see Fig. 2 for the comparison).

---

#### C.4 TRAINING HYPERPARAMETERS

Table 2 presents the core training hyperparameters used across all experiments. We use a cosine learning rate schedule with warmup to ensure stable optimization dynamics.

Table 2: Core Training Hyperparameters

Parameter	Value
Learning Rate	$5 \times 10^{-5}$
LR Schedule	Cosine
Warmup Ratio	0.05
Training Steps	400
Per-device Train Batch Size	4
Gradient Accumulation Steps	2
Effective Batch Size	8 per device
Mixed Precision	bfloat16

All experiments are conducted using mixed-precision training (bfloat16) with Flash Attention 2 ? for computational efficiency.

#### C.5 GRPO-SPECIFIC CONFIGURATION

The GRPO algorithm requires several algorithm-specific hyperparameters that control the policy optimization process. These are detailed in Table 3.

Table 3: GRPO Algorithm Hyperparameters

Parameter	Value
Number of Generations	8
KL Penalty Coefficient ( $\beta$ )	0.001
Max Prompt Length	256 tokens
Max Completion Length	1024 tokens
Temperature	1.0 (default)
Top-p Sampling	1.0 (default)

During training, the model generates 8 completions per prompt for group-based advantage estimation. The KL penalty coefficient  $\beta$  controls the divergence between the learned policy and the reference policy, preventing catastrophic forgetting of the pretrained knowledge.