TORCH: Tool-Oriented Rubric-based Co-evolving Helper for Reinforcement Learning in Agentic Systems

Adrian-Octavian Pătrașcu **Miriam Modiga**
University Politehnica of Bucharest
adrian.patrascu2205@stud.acs.upb.ro miriam.modiga@stud.fils.upb.ro

Abstract

Extending Reinforcement Learning to tool-using agents presents unique challenges beyond pure reasoning tasks. Current approaches to training such agents either rely on sparse outcome-based rewards that fail to address credit assignment in multi-step tool interactions, or employ static reward models that may be vulnerable to reward hacking as policies improve. We introduce **TORCH** (**T**ool-**O**riented **R**ubric-based **C**o-evolving **H**elper), a conceptual framework that explores three interconnected ideas: (1) rubric-based process rewards that could provide structured feedback on tool selection, input formation, and output interpretation; (2) adaptive co-evolution mechanisms where policies and Generative Reward Models (GenRMs) might be jointly trained using verifiable tool execution outcomes as meta-rewards; and (3) uncertainty-guided trajectory sampling strategies that prioritize informative training examples. This paper presents the theoretical foundation and motivation for TORCH, examining how structured process rewards combined with co-evolutionary dynamics might address fundamental challenges in training tool-using agents, including credit assignment, reward hacking, and generalization to novel tools.

1 Introduction

Reinforcement Learning has repeatedly demonstrated the ability to drive artificial agents to superhuman performance on complex reasoning tasks. Landmark systems such as AlphaGo [7] and AlphaZero [8] established RL as a practical technology for high-level problem solving. Recent breakthroughs like OpenAI's o1 [5] and DeepSeek-R1 [1] have shown that RL with verifiable rewards can enable sophisticated reasoning capabilities in Large Language Models (LLMs), achieving remarkable results on mathematics and coding benchmarks. As LLMs evolve into autonomous agents that must select and use external tools to solve real-world tasks, the limitations of current reward modeling approaches become increasingly apparent.

Tool-using agents face unique challenges that distinguish them from pure reasoning systems. First, **multi-step credit assignment** is more complex: determining which specific tool call in a long interaction sequence led to success or failure is non-trivial. Second, **partial observability** introduces ambiguity: tool outputs may be incomplete or require interpretation in context. Third, **compositional complexity** emerges: tools must be chained correctly, with the output of one tool informing the input to another. These challenges are compounded when agents must generalize to novel tools or tool combinations not seen during training.

Current approaches to training tool-using agents fall short in addressing these challenges. Outcome-based RL methods [9] provide only sparse signals at task completion, offering no intermediate guidance on which tool decisions were appropriate. Static Generative Reward Models (GenRMs), while capable of providing process-level feedback, become vulnerable to reward hacking as policies improve [3]—the policy learns to exploit fixed patterns in the reward model rather than developing

Preprint. Under review.

robust tool-use strategies. Existing process reward models [11], though effective for mathematical reasoning, treat tool use as atomic actions and fail to decompose the structure of tool interaction.

To address these limitations, we introduce **TORCH** (**T**ool-**O**riented **R**ubric-based **C**o-evolving **H**elper), a conceptual framework exploring three interconnected ideas:

- **Rubric-based Process Rewards for Tool Use:** Decomposing tool interactions into interpretable criteria—tool selection appropriateness, input quality, output interpretation correctness, and efficiency—could provide structured, fine-grained feedback at each decision point, potentially addressing the credit assignment problem through actionable and interpretable reward signals.
- **Adaptive Co-Evolution:** Rather than training a GenRM once and freezing it, continuously updating both the policy and GenRM might prevent reward hacking. Verifiable tool execution outcomes (e.g., successful API calls, correct file operations) could serve as meta-rewards to train the GenRM, creating a dynamic target that adapts as the policy improves.
- **Uncertainty-Guided Trajectory Sampling:** Prioritizing training on trajectories where the GenRM has high uncertainty or disagrees with final outcomes might maximize learning signal and enable structured exploration through branching at identified failure points.

Our main contributions are:

1. We identify key limitations of current GenRMs when applied to tool-using agents and propose a conceptual framework for rubric-based process rewards tailored to the structure of tool interaction.
2. We present an adaptive co-evolution mechanism that could jointly train the policy and GenRM, using verifiable tool outcomes as meta-rewards to continuously refine the reward model and potentially mitigate reward hacking.
3. We introduce the concept of uncertainty-guided trajectory sampling that might improve sample efficiency and enable structured exploration through GenRM-identified failure points.
4. We discuss how combining these three components could address fundamental challenges in training tool-using agents, including credit assignment, reward hacking, and generalization to novel tools.

The remainder of this paper is organized as follows. Section 2 reviews related work on generative reward models, co-evolution, and tool use in RL, providing context for the TORCH framework.

2 Related Work

2.1 Generative Reward Models

Generative Reward Models (GenRMs) extend RL beyond verifiable domains by providing nuanced, text-based feedback rather than scalar rewards [13]. We categorize recent work into three main approaches.

Process vs. Outcome Rewards. Outcome rewards evaluate only final results [1, 5], providing simple but sparse signals. Process rewards, such as those in RL Tango [11], evaluate each reasoning step, offering dense signals and better credit assignment. Recent surveys [13] show that process rewards consistently outperform outcome rewards on complex tasks, with improvements exceeding 20% on challenging reasoning benchmarks.

Reasoning Reward Models. A major advancement is training reward models to explicitly reason before rendering judgment, as demonstrated in LLM-as-a-Judge evaluations [14]. Recent work has explored generating natural language critiques before predicting scalar rewards, and training reasoning reward models with reinforcement learning using meta-rewards based on verdict correctness. Different reward formats have been explored across the field.

Specification-Based GenRMs. To address brittleness in rule-based verifiers, model-based semantic equivalence checking and multi-domain verifiers have been developed to handle diverse data types and reasoning tasks. While effective for reasoning, these approaches treat tool use as atomic actions, failing to decompose the internal structure of tool interactions.

2.2 Co-Evolution and Reward Hacking Prevention

A critical challenge in RL is reward hacking, where policies exploit shortcuts in static reward models rather than learning desired behaviors.

Self-Rewarding Systems. Self-Rewarding Language Models [10] enable a single model to alternate between policy and verifier roles, with iterative DPO enabling simultaneous improvement. Extensions to this approach include self-correction mechanisms and post-completion learning strategies. However, self-rewarding can amplify biases without external grounding.

Co-Optimization Frameworks. RL Tango [11] jointly trains a generator and process-level GenRM using only outcome-level rewards as meta-signals, achieving state-of-the-art results on 7B/8B models. Cooper [3] uses rule-based rewards to find trustworthy positive samples and an assistant LLM to generate challenging negatives dynamically. Critically, Cooper demonstrates that static reward models collapse (38.91% accuracy) while co-evolved models maintain performance (58.02%). Other work has explored unified player-referee models and hybrid reward schemes combining rule-based and generative approaches.

While these approaches demonstrate effective co-evolution for reasoning tasks, they do not address the unique challenges of tool use. TORCH adapts these principles by using verifiable tool execution outcomes as natural meta-rewards.

2.3 Tool Use and Agentic RL

Extending RL to tool-using agents introduces challenges beyond pure reasoning: agents must decide when to use tools, formulate appropriate inputs, and interpret outputs correctly.

Outcome-Based RL for Tool Use. ARTIST [9] applies outcome-based RL for autonomous tool selection and invocation, achieving up to 22% improvement on mathematical reasoning and strong gains on function calling benchmarks. VerlTool [6] provides a modular framework separating tool execution from RL workflow, achieving $2\times$ speedup through asynchronous rollouts. While effective, these approaches rely on sparse outcome rewards without intermediate guidance.

Trajectory Sampling Strategies. RLEP [12] introduces a two-phase framework that collects verified trajectories and replays them during training, achieving +1.7-5.2 pp improvements on mathematical benchmarks. Other approaches have explored retrospective replay mechanisms that dynamically revisit promising intermediate states from earlier training, and hindsight experience replay adaptations for language agents. These methods improve sample efficiency but do not address adaptive reward model refinement.

Rubric-Based Rewards. Rubrics as Rewards (RaR) [2] extends RLVR beyond verifiable domains using structured checklists for multi-criteria judgment, achieving +31% on HealthBench. Reinforcement Learning with Rubric Anchors [4] builds the largest rubric system to date (10,000+ rubrics), demonstrating that rubric diversity, granularity, and quantity are critical for performance. While these works focus on subjective tasks like creative writing, we explore how rubric-based rewards could be adapted specifically for tool use, where criteria can be grounded in verifiable execution outcomes.

How TORCH Differs. Existing tool-use RL systems either rely on sparse outcome rewards or use trajectory replay without adaptive reward model improvement. TORCH proposes to combine tool-specific rubric-based process rewards with co-evolution and uncertainty-guided sampling, potentially addressing both credit assignment and reward hacking challenges simultaneously.

References

References

[1] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 300+ authors; Submitted January 22, 2025; DeepSeek-AI.

[2] Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025. Submitted July 23, 2025; v2 October 3, 2025.

[3] Haitao Hong, Yuchen Yan, Xingyu Wu, Guiyang Hou, Wenqi Zhang, Weiming Lu, Yongliang Shen, and Jun Xiao. Cooper: Co-optimizing policy and reward models in reinforcement learning for large language models. *arXiv preprint arXiv:2508.05613*, 2025. Zhejiang University; Code: https://github.com/zju-real/cooper.

[4] Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, Jianguo Li, and Junbo Zhao. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*, 2025. 21 authors; 10,000+ rubrics dataset; Submitted August 18, 2025.

[5] Aaron Jaech et al. Learning to reason with llms. OpenAI Blog, 2024. o1 system card.

[6] Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, Tianyu Pang, and Wenhu Chen. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025. v1: Sept 1, 2025; v3: Oct 17, 2025; Code open-source.

[7] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[8] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[9] Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441*, 2025. Technical Report MSR-TR-042025-V1; Microsoft Research.

[10] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *International Conference on Machine Learning (ICML)*. Meta AI and New York University, 2024.

[11] Kaiwen Zha, Zhengqi Gao, Maohao Shen, Zhang-Wei Hong, Duane S. Boning, and Dina Katabi. Rl tango: Reinforcing generator and verifier together for language reasoning. In *Proceedings of the 38th Conference on Neural Information Processing Systems*, 2025.

[12] Hongzhi Zhang, Jia Fu, Jingyuan Zhang, Kai Fu, Qi Wang, Fuzheng Zhang, and Guorui Zhou. Rlep: Reinforcement learning with experience replay for llm reasoning. *arXiv preprint arXiv:2507.07451*, 2025. Code and datasets available; Submitted July 10, 2025.

[13] Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, Fangfu Liu, Xiang Xu, Jiaze Ma, Xuekai Zhu, Ermo Hua, Yihao Liu, Zonglin Li, Huayu Chen, Xiaoye Qu, Yafu Li, Weize Chen, Zhenzhao Yuan, Junqi Gao, Dong Li, Zhiyuan Ma, Ganqu Cui, Zhiyuan Liu, Biqing Qi, Ning Ding, and Bowen Zhou. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025. 120 pages, comprehensive survey; Tsinghua University; Submitted Sep 10, 2025, revised Oct 9, 2025.

[14] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.