PROCESSTHINKER: ENHANCING MULTI-MODAL LARGE LANGUAGE MODELS REASONING VIA ROLLOUT-BASED PROCESS REWARD

Jingpei Wu[1,5]* **Xiao Han**[1]* **Weixiang Shen**[1] **Boer Zhang**[2] **Zifeng Ding**[3,4] **Volker Tresp**[1,5]
[1]LMU Munich [2]Harvard University [3]University of Cambridge [4]Mina AI
[5]Konrad Zuse School of Excellence in Reliable AI (relAI)

ABSTRACT

Visual question answering increasingly requires multi-step reasoning. Recent post-training with reinforcement learning under verifiable rewards (RLVR) and Group Relative Policy Optimization (GRPO) can improve multimodal reasoning, but most approaches rely on sparse outcome-only rewards. As a result, they struggle to tell whether an incorrect answer comes from a small mistake late in the reasoning or from an unhelpful trajectory from the start. A common solution is to train a process reward model (PRM) for step-level supervision, but this typically requires large-scale high-quality chain-of-thought annotations and additional training cost. We propose **ProcessThinker**, a practical post-training pipeline that provides step-level *process rewards* without training an explicit PRM. ProcessThinker first rewrites reasoning traces into a step-tagged format for cold-start supervised fine-tuning, then applies GRPO with a standard format reward and our rollout-based process reward. Concretely, for each intermediate step, we sample multiple continuations from that step and use the empirical success rate (final-answer verification) as the step reward. This gives dense credit assignment and encourages reasoning steps that more reliably support a correct conclusion, helping reduce inconsistent or self-contradictory progress across steps – a key issue in logical reasoning. Across four challenging video benchmarks (Video-MMMU, MMVU, VideoMathQA, and LongVideoBench), ProcessThinker consistently improves over the baseline model QWEN3-VL-8B-INSTRUCT.

1 INTRODUCTION

Multimodal large language models (MLLMs) have made rapid progress in open-ended visual understanding and question answering. With chain-of-thought (CoT) prompting, MLLMs can produce multi-step reasoning traces and often improve task performance. This long-horizon reasoning ability is increasingly important for complex problems where the answer depends on a sequence of intermediate inferences. Recently, post-training with reinforcement learning under verifiable rewards (RLVR), often paired with group-based objectives such as Group Relative Policy Optimization (GRPO), has further strengthened multi-step reasoning in both text-only and multimodal settings (Shao et al., 2024; DeepSeek-AI, 2025; Su et al., 2025; Sim et al., 2025; Feng et al., 2025; Zhang et al., 2025b; Park et al., 2025; Wang et al., 2025b; Zhang et al., 2025d; Yang et al., 2025; Huang et al., 2025). However, the supervision signal in GRPO-style RLVR is typically *sparse*: the verifier only checks the final answer. For long reasoning traces, many samples within a group can receive identical outcome rewards, which weakens learning signals and motivates reward shaping and sampling strategies (Yao et al., 2025; Chen et al., 2025; Zhang et al., 2025c; Niu et al., 2026; Yari & Koto, 2026; Tao et al., 2025; Lyu et al., 2025). Even with these improvements, outcome-only supervision still provides little information about *which intermediate steps* were helpful when the final answer is wrong.

A natural way to densify supervision is to score intermediate steps. Process reward models (PRMs) provide step-level feedback and have been used for reranking, search, and test-time scaling by evaluating the quality of the reasoning process (Lightman et al., 2023; Setlur et al., 2024; Khalifa

*Equal contribution.

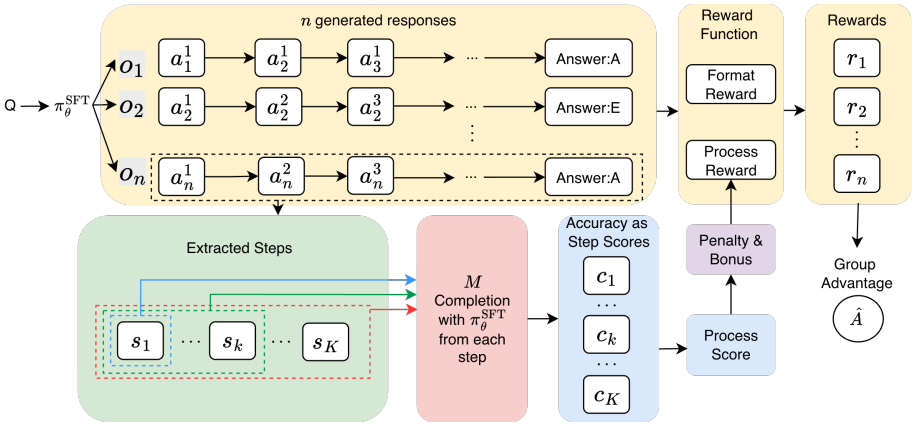

Figure 1: Rollout-based process reward inside one GRPO update. For a question Q, we sample a group of G candidate responses from the current policy. For each response, we extract step segments $\{s_t\}$ (incl. multiple a) and score each step k by the success rate of M continuation rollouts from previous steps $(s_1, \ldots, s_k)$ under policy model, producing step scores c_t and an averaged process score. The final reward for each response combines format reward, process reward, and step-count shaping (bonus + penalty gate), which is then used to compute group-relative advantages for GRPO.

et al., 2025; Zhao et al., 2025a; Wang et al., 2025a; Du et al., 2025). Recent work also explores using PRMs to supply process rewards during RL training (Luo et al., 2025). However, PRM-based supervision usually requires high-quality step annotations (or a complex pipeline to synthesize them), and automated approaches often rely on Monte-Carlo rollouts or MCTS-style search, which can be noisy and sensitive to how "steps" are defined (Zhang et al., 2024; 2025a;e; Tan et al., 2025; Ding et al., 2025). Moreover, training and maintaining a separate PRM adds engineering overhead and can introduce a mismatch between the PRM and the final policy.

This raises a question central to logical multi-step reasoning: *can we obtain step-level training signals that encourage more consistent reasoning traces* ***without*** *training a separate PRM?* Step-GRPO (Zhang et al., 2025b) is an important step in this direction, using rule-based step-wise rewards (e.g., rewarding the presence of key steps and enforcing a well-structured reasoning format). However, it still does not directly measure whether a specific intermediate step actually makes the problem easier to solve. VinePPO (Kazemnejad et al., 2025) shares a similar intuition, using Monte Carlo rollouts to estimate step-level values for credit assignment in PPO; however, it targets text-only LLMs and does not provide rollout-based process rewards within a GRPO framework. We propose **ProcessThinker**, which assigns a rollout-based *process reward* to each reasoning step via *continuation solvability* (Figure 1). The key idea is simple: an intermediate step is useful if, conditioned on the partial trace, the model is more likely to reach the correct final answer. We estimate this by sampling multiple continuations from the current policy starting from each step prefix and computing the empirical success rate under the same final-answer verifier used in RLVR. This provides a direct, model-free estimate of step utility and encourages reasoning traces whose steps more reliably support a correct conclusion, reducing inconsistent progress across steps—a core challenge in logical reasoning (Lightman et al., 2023). We instantiate ProcessThinker on QWEN3-VL-8B-INSTRUCT (Bai et al., 2025) and train in two stages: (i) a SFT warm-up on a step-tagged dataset obtained by rewriting VIDEO-R1-COT traces into explicit step decompositions using a stronger teacher model, and (ii) GRPO post-training with a weighted combination of sparse outcome reward and our rollout-based process reward, along with lightweight formatting incentives. We evaluate in the video domain, but the proposed reward construction is model- and modality-agnostic.

In summary, we make the following contributions: (1) We propose a simple GRPO-based post-training framework that incorporates step-level rewards *without* training an explicit process reward model (PRM). (2) We introduce a rollout-based process reward that scores each reasoning step by the empirical success rate of multiple continuations conditioned on the step prefix. (3) We demonstrate consistent improvements over QWEN3-VL-8B-INSTRUCT on four video reasoning benchmarks, and ablations show that increasing the weight of the process reward yields larger gains.

2 METHOD

Given a multimodal context x (video frames or an image plus text prompt), the model generates an output y. We enforce a step-tagged reasoning format

$$y = \langle\texttt{think}\rangle \ldots \langle\texttt{step}\rangle s_k \langle/\texttt{step}\rangle \ldots \langle/\texttt{think}\rangle \langle\texttt{answer}\rangle ans \langle/\texttt{answer}\rangle, \tag{1}$$

which makes intermediate steps explicit and enables step-wise scoring (Figure 1).

2.1 SFT DATA CONSTRUCTION (FORMAT + FILTERING)

Starting from VIDEO-R1-COT-165K (Feng et al., 2025), we rewrite each sample into the step-tagged format using a stronger teacher model QWEN3-VL-30B-A3B-INSTRUCT (Bai et al., 2025). The teacher is instructed to preserve the original solution while segmenting the reasoning into non-trivial, non-redundant steps. To reduce rewriting noise (missing/duplicated steps, semantic drift, step-answer mismatch), we apply a second-pass filter with the teacher that scores: (i) answer fidelity vs. the original solution, (ii) consistency between steps and the final answer, and (iii) step quality. We keep the top 19k samples for SFT and sample 1,250 prompts for RL. We fine-tune QWEN3-VL-8B-INSTRUCT on the 19k set to obtain ProcessThinker-SFT, which reliably emits parsable step-tagged traces.

2.2 GRPO WITH ROLLOUT-BASED PROCESS REWARDS

GPRO For each prompt x, we sample a group of G responses $\{y^{(g)}\}_{g=1}^{G}$ from the current policy $\pi_\theta(\cdot|x)$, compute a scalar reward $r^{(g)}$ for each response, and normalize rewards within the group to obtain relative advantages (mean/variance normalization). The policy is then updated using the standard GRPO recipe with KL regularization to a reference policy. ProcessThinker differs from prior RLVR work mainly in the reward design below.

Rollout-based process reward (continuation solvability). For a sampled response y with steps $s_{1:K}$ and ground-truth answer $ans^\star$, we score each prefix $p_i = (s_1, \ldots, s_i)$ by how often the model can successfully *finish* the problem when conditioned on that prefix. We sample M continuations $\hat{y}_i^{(m)} \sim \pi_\theta(\cdot|x, p_i)$ and define the step score as the empirical success rate:

$$c_i = \frac{1}{M} \sum_{m=1}^{M} \mathbf{I}\Big[\mathrm{Ans}(\hat{y}_i^{(m)}) = ans^\star\Big]. \tag{2}$$

The trajectory-level process reward averages prefix solvability:

$$R_{\text{proc}}(y) = \frac{1}{\min(K, K_{\max})} \sum_{i=1}^{\min(K, K_{\max})} c_i, \tag{3}$$

using M=4 and $K_{\max}$=6 unless noted. This gives dense credit assignment: early steps can receive partial credit even if the final answer in y is wrong.

Format reward, bounded step bonus, and penalty gate. We use a strict format reward r_{fmt} that is awarded only if tags are properly nested and $K \in [K_{\min}, K_{\max}]$ (and optionally a length bonus if within $[L_{\min}, L_{\max}]$), similar in spirit to step-structured RL recipes. To encourage using more than the minimum number of steps without step inflation, we add a bounded step bonus

$$B(K) = \alpha \sqrt{\mathrm{clip}\Big(\frac{K - K_{\min}}{K_{\max} - K_{\min}},\, 0,\, 1\Big)}. \tag{4}$$

To reduce reward hacking (too few steps or shallow steps that only satisfy format), we gate rewards with a simple penalty:

$$\bar{R}_{\text{acc}} = \begin{cases} 1, & R_{\text{acc}} = 1, \\ -B(K), & \text{otherwise}, \end{cases} \qquad \bar{R}_{\text{proc}} = \begin{cases} R_{\text{proc}}, & R_{\text{proc}} \geq \tau, \\ -B(K), & \text{otherwise}, \end{cases} \tag{5}$$

where $R_{\text{acc}} \in \{0, 1\}$ is final-answer accuracy and τ=0.5.

Final reward. For format-valid responses, the reward is

$$r^{(g)} = (r_{\text{fmt}} + \beta) \;+\; \lambda_{\text{acc}} \bar{R}_{\text{acc}} \;+\; \lambda_{\text{proc}} \bar{R}_{\text{proc}} \;+\; B(K), \qquad \lambda_{\text{acc}} + \lambda_{\text{proc}} = 1. \tag{6}$$

If formatting is invalid, we set $r^{(g)}$=0 and skip continuation rollouts for efficiency.

Model	Accuracy (%) ↑				
	Video-MMMU	MMVU (mc)	VideoMathQA	LongVideoBench	Avg.
QWEN3-VL-8B-INSTRUCT	62.89	65.60	25.20	71.50	56.30
VIDEO-R1-7B(Feng et al., 2025)	53.89	65.92	26.67	58.30	51.20
PROCESSTHINKER-SFT	58.78	64.48	23.57	68.50	53.83
ProcessThinker (outcome-only)	60.78	67.36	27.86	74.20	57.55
ProcessThinker (outcome + process)	61.67	67.52	27.86	74.60	57.91
ProcessThinker (process-only)	**63.33**	**68.48**	**31.67**	**75.40**	**59.72**

Table 1: Main results on four video reasoning benchmarks. All ProcessThinker variants share the same SFT warm-up and are trained with GRPO using the overall reward in equation 6. "process-only" uses our rollout-based step-wise process reward (Section 2.2), while "outcome-only" uses only final-answer correctness.

3 EXPERIMENTS

We evaluate on four video reasoning benchmarks: VIDEO-MMMU (Hu et al., 2025), MMVU (Zhao et al., 2025b), VIDEOMATHQA (Rasheed et al., 2025), and LONGVIDEOBENCH (Wu et al., 2024), and report accuracy following each benchmark's official protocol.

Training setup. All ProcessThinker variants share the same SFT warm-up and differ only in the reward mixture $(\lambda_{\text{acc}}, \lambda_{\text{proc}})$ in equation 6. Unless otherwise stated, we sample $G{=}4$ responses per prompt for GRPO and compute process rewards with $M{=}4$ continuation rollouts per step, capped at $K_{\max}{=}6$.

Main results. As shown in Table 1, ProcessThinker (process-only) improves over the QWEN3-VL-8B-INSTRUCT baseline on all four benchmarks, raising the average score from 56.30 to 59.72 (+3.42). VIDEO-R1-7B (Feng et al., 2025) is included for reference, though it uses the older QWEN2.5-VL backbone and is not directly comparable. The largest gain is on VIDEOMATHQA (+6.47), which requires multi-step reasoning while integrating sparse cues over time; we also observe a clear improvement on LONGVIDEOBENCH (+3.90), suggesting better reasoning under long-context inputs. Overall, the consistent gains across benchmarks indicate that step-level credit assignment from rollout-based process rewards transfers beyond a single dataset type.

SFT warm-up: necessary but insufficient. PROCESSTHINKER-SFT underperforms the instruction-tuned baseline despite improved format compliance, suggesting that step-tag supervision mainly teaches *how to write* structured traces, but does not directly optimize verified task success. GRPO post-training is therefore critical to turn well-formed reasoning traces into higher final-answer accuracy.

Process reward vs. outcome reward. Among reward mixtures, *process-only* performs best, while *outcome-only* and a balanced mixture lag behind. This supports our hypothesis that outcome-only supervision is too sparse for long-horizon reasoning, whereas rollout-based process rewards provide denser signals that better capture the usefulness of intermediate steps.

4 CONCLUSION

ProcessThinker shows that step-wise credit assignment can be obtained from sparse verifiable outcomes without training a separate PRM: we score each step by its *continuation solvability* via rollout success rates. This rollout-based process reward encourages intermediate steps that more reliably support a correct conclusion, addressing a key issue in logical multi-step reasoning: unproductive or inconsistent progress across steps. Combined with strict formatting constraints and GRPO-style RLVR, ProcessThinker consistently improves QWEN3-VL-8B-INSTRUCT on four video reasoning benchmarks, with the strongest gains when emphasizing process rewards over outcome-only supervision. The main limitation is efficiency: although we avoid PRM annotation/training, continuation-solvability requires multiple rollouts per step, and the added inference cost can offset the benefit of using fewer data. The resulting reward can also be noisy due to rollout stochasticity and sensitivity to step segmentation.

ACKNOWLEDGMENTS

This paper is supported by the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the Federal Ministry of Research, Technology and Space.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. 2025.

Weilong Chen, Xiaohang Li, Zhihao Yang, Jiajie Jin, and Yi Yang. Xrpo: Pushing the limits of grpo with targeted exploration and exploitation. 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. 2025.

Yuyang Ding, Xinyu Shi, Juntao Li, Xiaobo Liang, Zhaopeng Tu, and Min Zhang. Scan: Self-denoising monte carlo annotation for robust process reward learning. 2025.

Lingxiao Du, Fanqing Meng, Zongkai Liu, Zhixiang Zhou, Ping Luo, Qiaosheng Zhang, and Wenqi Shao. Mm-prm: Enhancing multimodal mathematical reasoning with scalable step-level supervision. 2025.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in multimodal large language models. 2025.

Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. 2025.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. 2025.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. In *International Conference on Machine Learning*, 2025.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. 2025.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. 2023.

Ruilin Luo, Zhuofan Zheng, Yifan Wang, Xinzhe Ni, Zicheng Lin, Songtao Jiang, Yiyao Yu, Chufan Shi, Lei Wang, Ruihang Chu, Yue Qian, and Dayiheng Liu. Unlocking multimodal mathematical reasoning via process reward model. 2025.

Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, Weihan Cao, Jiangning Liu, Hongwei Liu, Junnan Liu, Songyang Zhang, Dahua Lin, and Kai Chen. Exploring the limit of outcome reward for learning mathematical reasoning. 2025.

Rui Niu, Yaqi Wu, Bin Wang, and Xuefeng Xiang. From absolute to relative: Rethinking reward shaping in group-based reinforcement learning. 2026.

Jinyoung Park, Jeehye Na, Jinyoung Kim, and Hyunwoo J. Kim. Deepvideo-r1: Video reinforcement fine-tuning via difficulty-aware regressive grpo. 2025.

Hanoona Rasheed, Abdelrahman Shaker, Anqi Tang, Muhammad Maaz, Ming-Hsuan Yang, Salman Khan, and Fahad S. Khan. Videomathqa: Benchmarking mathematical reasoning via multimodal understanding in videos. 2025.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. 2024.

Shang Hong Sim, Tej Deep Pala, Vernon Toh, Hai Leong Chieu, Amir Zadeh, Chuan Li, Navonil Majumder, and Soujanya Poria. Lessons from training grounded llms with verifiable rewards. 2025.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. 2025.

Xiaoyu Tan, Tianchu Yao, Chao Qu, Bin Li, Minghao Yang, Dakuan Lu, Haozhe Wang, Xihe Qiu, Wei Chu, Yinghui Xu, and Yuan Qi. Aurora: Automated training framework of universal process reward models via ensemble prompting and reverse verification. 2025.

Leitian Tao, Ilia Kulikov, Swarnadeep Saha, Tianlu Wang, Jing Xu, Sharon Li, Jason E. Weston, and Ping Yu. Hybrid reinforcement: When reward is sparse, it's better to be dense. 2025.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Ziyi Wang, Ziyu Zhang, Kaiyan Zhang, Runze Liu, and Jian Zhao. Visualprm: An effective process reward model for multimodal reasoning. 2025a.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, Xiangnan Fang, Zewen He, Zhenbo Luo, Wenxuan Wang, Junqi Lin, Jian Luan, and Qin Jin. Time-r1: Post-training large vision language model for temporal video grounding. 2025b.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. 2024.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, Bo Zhang, and Wei Chen. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. pp. 2376–2385, October 2025.

Huanjin Yao, Qixiang Yin, Jingyi Zhang, Min Yang, Yibo Wang, Wenhao Wu, Fei Su, Li Shen, Minghui Qiu, Dacheng Tao, and Jiaxing Huang. R1-sharevl: Incentivizing reasoning capability of multimodal large language models via share-grpo. 2025.

Saeed Yari and Fajri Koto. Amir-grpo: Inducing implicit preference signals into grpo. 2026.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. 2024.

Dan Zhang, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-rl: Process reward guided reinforcement learning for large language model reasoning. 2025a.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. 2025b.

Xiaohang Zhang, Wenhao Wu, Yong Peng, Huanqin Zhou, Renzhi Wu, and Liang Pang. Grpo-lead: A difficulty-aware reinforcement learning approach for concise mathematical reasoning in language models. 2025c.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Tinyllava-video-r1: Towards smaller lmms for video reasoning. 2025d.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. 2025e.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. Genprm: Scaling test-time compute of process reward models via generative reasoning. 2025a.

Yilun Zhao, Yuxuan Li, Haoning Wu, Yuanhan Zhang, Ziwei Liu, and Yu Qiao. Mmvu: Measuring expert-level multi-discipline video understanding. 2025b.