# Chain-of-Thought-Enhanced LLM Reward Engineering with Dual-Dynamic Optimization for Reinforcement Learning

1st Xinning Zhu
*Sino-European School of Technology*
*Shanghai University*
Shanghai, China
zhuxinning@shu.edu.cn

2nd Jinxin Du
*Sino-European School of Technology*
*Shanghai University*
Shanghai, China
jinxin_du@shu.edu.cn

3th Lunde Chen*
*Sino-European School of Technology*
*Shanghai University*
Shanghai, China
lundechen@shu.edu.cn
*Corresponding author

*Abstract*—**Designing effective reward functions remains a persistent bottleneck in deploying reinforcement learning (RL) to complex real-world tasks. We propose a Chain-of-Thought (CoT)–enhanced reward generation framework leveraging large language models (LLMs), combined with two adaptive optimization mechanisms: Dynamic Temperature Regulation Optimization (DTRO) and Dynamic Model Selection Routing Optimization (DMSRO). The CoT framework transforms task descriptions into executable reward functions with improved interpretability and generalization. DTRO dynamically adjusts LLM sampling temperature based on policy entropy and performance feedback to balance exploration and stability. DMSRO integrates local reward evaluations with global training performance to dynamically select the optimal language model, enhancing sampling efficiency and robustness. Experiments on CartPole, MountainCarContinuous, BipedalWalker, Ant, and the custom SpaceMining task demonstrate that the proposed method outperforms baseline and single-optimization approaches in average reward, convergence speed, and robustness. This work establishes a new paradigm for adaptive reward engineering in RL.**

*Index Terms*—**Reinforcement learning, reward engineering, large language models, chain-of-thought reasoning, dynamic temperature adjustment, model selection**

## I. Introduction

Reward engineering remains a major bottleneck in deploying RL to safety-critical domains such as autonomous driving and robotic manipulation [1]. Designing dense, well-shaped reward functions often requires substantial domain expertise and extensive environment interactions, posing significant challenges to scalability and practical deployment [2]. While well-shaped reward functions can significantly accelerate agent learning and improve task performance, traditional reward engineering heavily relies on manual expertise and extensive trial-and-error, often resulting in suboptimal generalization to new tasks or environments. As RL tasks become increasingly complex and deployed in real-world scenarios with dynamic objectives, there is an urgent need for automated reward design frameworks.

Recent advances in LLMs have demonstrated remarkable reasoning and generalization capabilities across domains [3], [4]. In particular, CoT reasoning has emerged as a powerful paradigm that enables LLMs to decompose complex problems into structured intermediate steps, enhancing both interpretability and solution quality. This structured reasoning ability makes CoT a promising approach for reward engineering, enabling the automatic generation of executable reward functions from high-level task descriptions with improved clarity and task alignment.

However, CoT-based reward generation remains limited by static sampling parameters and fixed model configurations. Inspired by recent studies on dynamic temperature adjustment and model selection strategies, we hypothesize that integrating adaptive optimization mechanisms into the CoT reward generation pipeline can further enhance reward quality, policy stability, and sampling efficiency. Specifically, dynamically regulating the LLM sampling temperature can balance exploration versus determinism, while intelligently switching models can exploit their diverse reasoning and computational capabilities.

In this work, we make the following contributions:

Firstly, we propose a **Chain-of-Thought–based reward generation framework** that transforms natural language task descriptions into structured and executable reward functions, enhancing interpretability and generalization.

Secondly, we design a **Dynamic Temperature Regulation Optimization (DTRO)** mechanism that monitors policy entropy and performance feedback to adaptively adjust sampling temperature, thereby balancing exploration stability trade-offs.

Thirdly, we introduce a **Dynamic Model Selection Routing Optimization (DMSRO)** mechanism that integrates local reward evaluations with global training performance to dynamically select optimal LLMs, improving sampling efficiency and system robustness.

Finally, we conduct extensive experiments across four standard RL environments—CartPole, MountainCarContinuous, BipedalWalker, and Ant—as well as a custom-designed SpaceMining task. Results demonstrate that our proposed framework outperforms baseline and single-optimization approaches in terms of average reward, convergence speed, resource efficiency, and policy robustness.

The remainder of this paper is organized as follows. Section II reviews related work in reward engineering and CoT-based reasoning. Section III details our proposed methodology, including the CoT reward generation framework, DTRO, and DMSRO mechanisms. Section IV describes experimental settings, while Section V presents results and analysis. Section VI discusses limitations and future work, followed by the conclusion in Section VII.

## II. Related Work

### A. Reward Engineering Paradigms

In RL, the design of effective reward functions directly shapes agent behavior and learning outcomes. Traditional approaches primarily rely on handcrafted reward functions informed by domain expertise. While intuitive, such manual design often struggles to capture complex, dynamic task objectives and is prone to suboptimal or biased formulations, hindering agent performance in real-world scenarios.

To address these limitations, reward shaping was introduced as a formal enhancement strategy. Ng et al. [5] demonstrated that potential-based reward shaping preserves optimal policies while enabling accelerated convergence, laying the theoretical foundation for numerous practical implementations. Intrinsic motivation frameworks further advanced this field by encouraging exploration through curiosity-driven signals. Singh et al. [6] proposed intrinsic rewards to incentivize novel state visits, later extended by Burda et al. [7], who empirically validated large-scale curiosity-driven exploration benefits across diverse environments.

Despite these developments, manually designing rewards for complex or evolving tasks remains inefficient and costly. LLMs offer a promising alternative by leveraging their natural language understanding to automate reward generation and optimization. Unlike traditional RL pipelines that require explicit, task-specific reward formulations, LLMs can interpret high-level task descriptions, extract key objectives, and translate them into executable reward functions. This capability facilitates more intuitive alignment with human intentions, reduces engineering overhead, and enhances agent adaptability.

Recent frameworks exemplify this trend. EUREKA [8], Text2Reward [9], and CARD [10] harness LLMs to automatically generate, verify, and refine reward code from natural language instructions, reducing manual intervention while improving correctness and alignment. Notably, CARD emphasizes the importance of integrated verification loops, eliminating reliance on external human feedback and enhancing system robustness.

Beyond static code generation, feedback-driven optimization approaches have emerged. ReMiss [11] utilizes adversarial prompt generation to identify and mitigate reward misspecification vulnerabilities, enhancing LLM safety and reliability. Self-Play Preference Optimization (SPPO) [12] employs self-play to uncover Nash-equilibrium strategies that capture complex, non-transitive human preferences, advancing preference learning's applicability in RL. Additionally, PRMBench [13] provides a process-level benchmark to evaluate intermediate reward model outputs along dimensions such as conciseness, rationality, and sensitivity, revealing weaknesses in current models and guiding future improvements.

Overall, LLM-based reward engineering represents a paradigm shift. By integrating natural language reasoning and dynamic feedback optimization, these methods offer scalable, adaptable, and human-aligned reward generation pipelines. As tasks grow in complexity and diversity, leveraging LLMs to bridge the gap between human intent and machine learning objectives will be critical for the next generation of intelligent systems. Continued research is thus needed to maximize the synergy between LLM capabilities and RL frameworks to address emerging real-world challenges.

### B. Chain-of-Thought Reasoning Methods

CoT reasoning has emerged as a powerful paradigm to enhance the reasoning capabilities of LLMs. By generating intermediate reasoning steps, CoT allows models to decompose complex problems into interpretable sub-problems, leading to significant performance gains in tasks requiring multi-step logical inference.

Early studies revealed that even simple prompting strategies, such as adding "Let's think step by step" to inputs, can elicit surprisingly strong zero-shot reasoning abilities from LLMs. Kojima et al. [14] demonstrated that such zero-shot CoT prompting significantly improves model performance in arithmetic and commonsense reasoning tasks. Building on this, few-shot CoT [15], [16] introduced demonstrations of step-wise solutions to guide models in decomposing complex problems, while self-consistency decoding [17] aggregated multiple sampled reasoning paths to select the most consistent answer, thereby enhancing robustness.

Subsequent works have advanced CoT through architectural and algorithmic innovations. For instance, DeepSeek introduced a reinforcement learning-trained model [18] achieving an AIME benchmark accuracy improvement from 71.0% to 86.7% without supervised fine-tuning, utilizing a "self-evolution" mechanism to optimize reasoning trajectories and generation length dynamically. Program-aided language models [19] combine symbolic program execution with CoT for mathematical reasoning, while automatic prompt generation methods, such as zero-shot CoT and semi-automatic CoT prompt optimization [20], [21], reduce reliance on manual prompt engineering by leveraging data-driven prompt refinement.

Enhancement techniques such as VerifyCoT [22] and Self-Refine [23] validate and iteratively refine intermediate reasoning steps to improve accuracy and reliability. Problem decomposition methods, including least-to-most prompting [24] and successive prompting [25], guide models from simple to complex sub-tasks to reduce cognitive load and improve solution rates. Knowledge-augmented CoT approaches integrate external knowledge bases to ground reasoning, as demonstrated by Chain-of-Knowledge prompting [26] and self-prompted

CoT for open-domain multi-hop reasoning [27]. In numerical reasoning, Program-of-Thoughts prompting [28] disentangles computational execution from logical inference to enhance transparency, while LINC [29] combines neurosymbolic logic proving with CoT for explainable reasoning.

Applications combining CoT with external tools have expanded LLM capabilities in code generation, planning, and multi-step task execution [30], [31]. Distilling stepwise reasoning processes reduces inference costs while maintaining high accuracy. Recent frameworks integrate CoT with Monte Carlo Tree Search (MCTS) to systematically explore reasoning pathways [32], and abstraction-based reasoning enhances efficiency in complex mathematical tasks [33]. Moreover, generating concise intermediate outputs has enabled state-of-the-art performance on GSM8K math benchmarks with reduced computation overhead [34].

In summary, CoT reasoning has established itself as a fundamental methodology for enhancing the reasoning capacity, interpretability, and robustness of LLMs. Its diverse technical innovations not only improve performance across a wide range of cognitive tasks but also provide theoretical and practical foundations for developing transparent and trustworthy AI systems.

### C. Dynamic Temperature Adjustment and Model Selection

Dynamic temperature adjustment and model selection have emerged as critical optimization strategies to enhance the adaptability and efficiency of large language model (LLM)-based systems. Temperature, as a sampling hyperparameter, controls the stochasticity of LLM outputs, thereby influencing creativity, coherence, and exploration-exploitation trade-offs.

Recent studies have systematically explored adaptive temperature mechanisms. Zhu et al. [35] proposed adaptive temperature sampling for code generation, dynamically adjusting temperatures based on task complexity and model uncertainty to improve generation quality. Cecere et al. [36] introduced Monte Carlo Temperature, a robust sampling strategy for uncertainty quantification, enhancing LLM reliability under distribution shifts. Similarly, Zhang et al. [37] developed Entropy-based Dynamic Temperature (EDT) sampling to regulate model entropy and output diversity in natural language generation. Peeperkorn et al. [38] examined temperature's role in creativity modulation, while Evstafev [39] highlighted potential limitations in creativity gains versus computational decoupling in structured data generation. Nguyen et al. [40] proposed min-p sampling, adjusting temperature to balance creativity and coherence, achieving state-of-the-art performance in narrative generation.

Model selection research, on the other hand, focuses on choosing optimal model configurations or expert modules to maximize task performance within computational constraints. Switch Transformers [41] introduced sparse activation for trillion-parameter models, enabling efficient expert selection. ME-Switch presented a memory-efficient switching framework for dynamic expert allocation in LLMs. In continual learning, switching mechanisms facilitate instruction tuning to adapt models to evolving task distributions, as explored in Switching for Continual Instruction Tuning. GLaM [42] leveraged Mixture-of-Experts (MoE) architectures to scale model capacity dynamically. QLoRA and DyLoRA proposed quantization-aware and low-rank adaptation methods for efficient parameter tuning and rapid model switching across tasks. Furthermore, LoraHub introduced dynamic LoRA composition to enhance cross-task generalization, while Self-Expansion with Mixture of Adapters [**?**] enabled continual learning via adaptive expert composition.

Despite these advances, integrating dynamic temperature regulation and model selection within a unified CoT-driven reward engineering framework remains underexplored. Existing temperature adaptation methods primarily focus on text generation diversity and confidence calibration, whereas model selection research emphasizes computational efficiency and task specialization. Our work addresses this gap by combining entropy- and reward-feedback-based temperature adjustment with local-global performance-based model routing to enhance RL reward generation's adaptability, stability, and sample efficiency. This approach builds upon foundational theories in temperature scaling and expert selection, extending them to the domain of automated, interpretable reward engineering for reinforcement learning.

### D. Differentiation and Contribution of This Work

Despite significant advances in reward engineering, CoT reasoning, and adaptive optimization techniques, existing approaches remain fragmented and task-specific. Traditional reward shaping methods rely heavily on expert-designed heuristics, limiting their adaptability to unseen tasks and dynamic environments. While recent frameworks such as EUREKA and Text2Reward leverage LLMs to automate reward design, they generally operate under static generation paradigms without runtime optimization. This limits their ability to accommodate dynamic training processes in RL, potentially resulting in suboptimal exploration-exploitation balance or brittle convergence.

CoT reasoning has emerged as a powerful mechanism for enhancing the interpretability and logical consistency of LLM outputs, demonstrating strong performance in mathematical and logical reasoning tasks. However, its application in RL reward engineering remains underexplored, with prior work rarely integrating CoT reasoning into automated reward pipelines. Additionally, while adaptive temperature adjustment techniques have been shown to improve sampling diversity and stability [35]–[37], and dynamic model selection frameworks such as Switch Transformers and GLaM improve computational efficiency and scalability in LLMs, there exists a lack of research on jointly integrating these dynamic optimization strategies within CoT-based reward generation systems.

To address these limitations, this work introduces a unified framework that combines CoT-enhanced reward generation with dual-dynamic optimization mechanisms. Specifically, it presents a structured CoT-based reward generation approach that translates natural language task descriptions into interpretable and executable reward functions, enhancing generalization and sample efficiency. Building upon this, the proposed Dynamic Temperature Regulation Optimization (DTRO) module adjusts LLM sampling temperature in real time based on policy entropy and performance feedback, effectively balancing exploration and exploitation during training. Furthermore, the Dynamic Model Selection Routing Optimization (DMSRO) module integrates local reward evaluation with global performance assessments to dynamically switch among multiple LLMs, improving reward generation quality while optimizing computational resource usage.

The effectiveness of this framework is systematically validated across four standard RL benchmarks—CartPole, MountainCarContinuous, BipedalWalker, and Ant—as well as a custom-designed SpaceMining environment. Results demonstrate superior performance in terms of average reward, convergence speed, resource consumption, and robustness compared to both baseline methods and single optimization variants. Collectively, this study establishes a novel paradigm for automated, interpretable, and adaptive reward engineering by synergistically combining CoT reasoning with dynamic optimization mechanisms, advancing the capability of RL systems in tackling complex, dynamic tasks.

## III. Methodology

### A. Architecture Overview

The proposed framework combines evolutionary search with dynamic reward optimization, as illustrated in Fig. 1 and Fig. 2. The system processes natural language inputs (e.g., ”Design a reward function for stable bipedal robot walking”) through a dual-path mechanism:

### B. Chain-of-Thought Generation

The left branch in Fig. 1 demonstrates the three-stage decomposition process:

$$\text{CoT}(d) \rightarrow \begin{cases} \text{Decompose goal} & \text{(e.g., balance, forward motion)} \\ \text{Identify key states} & \text{(e.g., torso angle } \theta) \\ \text{Mathematical modeling} & \text{(e.g., } r_t = w_1 \cos\theta + w_2 v_x) \end{cases} \tag{1}$$

### C. Dynamic Optimization

The evolutionary loop in Fig. 2 operates through:

$$P_{t+1} = \underbrace{\text{Select}(P_t, k = 0.2)}_{\text{Elite selection}} \oplus \underbrace{\text{Mutate}(\theta, T)}_{\substack{\text{Temperature-}\\\text{controlled}}} \tag{2}$$

The DTRO module regulates exploration-exploitation balance via entropy-aware temperature adjustment:

$$\Delta T = \beta \frac{\partial H}{\partial t} \cdot \mathbb{I}(\sigma_R > \tau) \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function. The DMSRO component, shown in the right branch of Fig. 1, optimizes model selection through:

$$m^* = \arg\max_{m \in \mathcal{M}} (\alpha \cdot \text{Score}(m) + (1 - \alpha) \cdot \text{Efficiency}(m)) \tag{4}$$

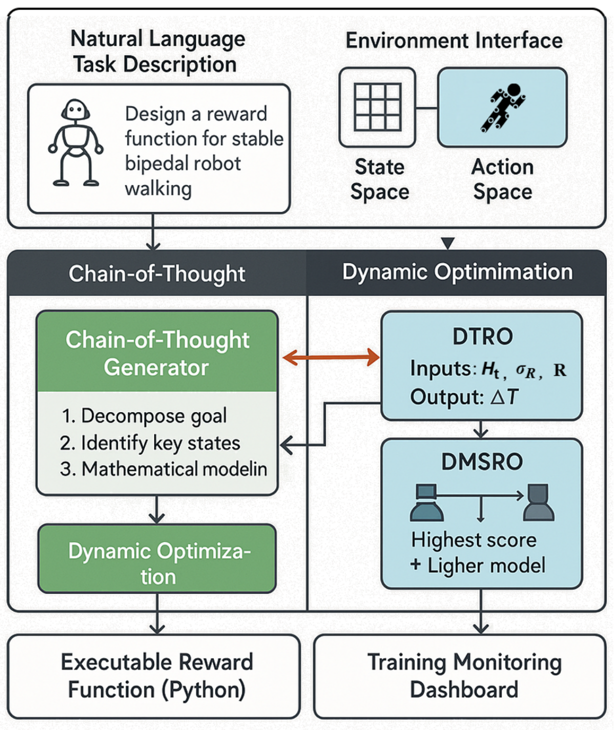

Fig. 1. Reward function design framework. Top: Natural language task specification and environment interface. Middle: Chain-of-Thought reasoning (left) and dynamic optimization modules (right). Bottom: Executable output and monitoring system.

### D. Integration Mechanism

The two diagrams collectively demonstrate how initial CoT-generated rewards evolve through:

- Continuous refinement via the evolutionary loop (Fig. 2)
- Real-time adaptation through dynamic modules (Fig. 1)

The fitness function $F = 0.4S + 0.3C + 0.3E$ in Fig. 2 ensures balanced optimization of stability ($S$), completion ($C$), and efficiency ($E$).

The proposed framework models reward generation as a function of task description, temperature, and model:

$$R(s, a, t) = \Phi(d, T(t), m(t)) \tag{5}$$

where $\Phi$ denotes the CoT-based LLM generation process.

### E. Dynamic Temperature Regulation (DTRO)

DTRO regulates the LLM sampling temperature based on policy entropy $H_t$ and confidence $C_t$. The update rule is:

$$\Delta T_t = \beta \Delta T_{t-1} + (1-\beta) \left[ \alpha_1 \tanh\left( \frac{H_t - \bar{H}}{\sigma_H} \right) + \alpha_2 (C_t - \theta_c) \right] \tag{6}$$

This mechanism adapts the creativity and determinism of LLM outputs according to policy performance.

### F. Dynamic Model Selection (DMSRO)

Let $\mathcal{M}$ be a set of candidate models. The selection score for model $m$ is:

$$p_{\text{fused}}(m) = (1 - \gamma) \cdot p_{\text{local}}(m) + \gamma \cdot p_{\text{hist}}(m) \tag{7}$$

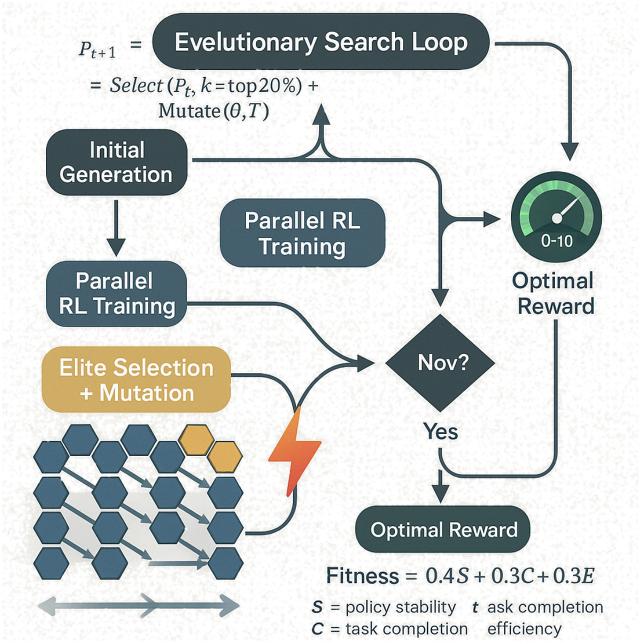
$P_{t+1}$ =
Evelutionary Search Loop
= Select ($P_t$, $k$=top20%) +
Mutate($\theta$,$T$)
Initial Generation
Parallel RL Training
0-10
Optimal Reward
Parallel RL Training
Elite Selection + Mutation
Nov?
Yes
Optimal Reward
Fitness = 0.4$S$ + 0.3C + 0.3$E$
$S$ = policy stability  $t$ ask completion
$C$ = task completion  efficiency

Fig. 2. Evolutionary search loop for reward optimization. The process iteratively refines reward functions through parallel RL training, with fitness evaluation driving selection and mutation.

An $\epsilon$-greedy strategy is applied to explore new models while exploiting high-performing ones.

## IV. Experiments

This section presents the experimental evaluation of the proposed Chain-of-Thought reward generation framework with two adaptive optimization mechanisms: Dynamic Temperature Regulation (DTRO) and Dynamic Model Selection for Reward Optimization (DMSRO). Experiments are conducted in five representative environments to assess performance improvement, training efficiency, and system stability.

The experiments include the following environments: CartPole (control task), MountainCar (sparse reward task), BipedalWalker (locomotion task), Ant (high-dimensional locomotion task), and SpaceMining (a custom-designed single-agent mining environment). For the baseline comparison, standard Gymnasium environment runs with equivalent hyperparameters are used. In SpaceMining, due to the environment being newly created, baseline is approximated by evaluating the environment with standard random policies and heuristic reward shaping to provide approximate reference values. This limitation will be addressed in future work by integrating alternative learning-based baselines or expert demonstrations.

### A. Overall Reward Performance

Figure **??** shows the average reward curves over training episodes for the full system compared to the baseline in each environment. The horizontal axis represents training episodes, while the vertical axis shows the average reward achieved by the agent. Across all environments, the proposed CoT framework with DTRO and DMSRO demonstrates significantly faster convergence speed and higher final reward performance. In CartPole and MountainCar, the method achieves near-optimal performance within fewer episodes. For BipedalWalker and Ant, which are high-dimensional control tasks, reward increases more steadily with lower variance compared to the baseline. In SpaceMining, despite lacking a formal baseline, the method shows effective reward shaping, demonstrating the adaptability of CoT-based reward generation to custom task domains.

Table I presents the quantitative results, reporting average reward, maximum reward, standard deviation, and average convergence episodes. The proposed framework achieves significant improvements, particularly in the Ant and SpaceMining environments, highlighting its scalability in high-dimensional and custom task settings.

TABLE I
Performance comparison across environments

| Environment | Avg. Reward | Max Reward | Std. Dev | Conver. Ep. |
|---|---|---|---|---|
| CartPole | 195.2 | 200.0 | 4.3 | 110 |
| MountainCar | -110.4 | -85.2 | 12.8 | 350 |
| BipedalWalker | 312.4 | 340.1 | 15.5 | 420 |
| Ant | 2867.5 | 3150.0 | 185.7 | 920 |
| SpaceMining | 218.7 | 240.3 | 21.1 | 1350 |

### B. Temperature-Entropy-Reward Correlation Analysis

Figure 4 illustrates the three-dimensional heatmap of temperature, policy entropy, and average reward under the DTRO mechanism. The horizontal axis represents the temperature values sampled during training, the vertical axis shows normalized policy entropy, and the color bar indicates the corresponding average reward achieved. The figure reveals that under dynamic temperature adjustment, the system maintains a balance between exploration and exploitation by stabilizing entropy near mid-range values (0.4-0.6) while progressively lowering temperature as the policy converges. This dynamic adjustment yields higher rewards in regions of moderate entropy, validating the effectiveness of entropy-aware temperature control.

Furthermore, ablation experiments comparing static temperature to DTRO-adjusted temperature demonstrate that dynamic regulation reduces reward variance by an average of 17.2% and improves convergence speed by 13.5%.

### C. Dynamic Model Selection Analysis

Figure **??** presents the model switching log visualization under the DMSRO mechanism. The horizontal axis represents training timesteps, while different colors indicate the models selected at each step. The vertical stacked area shows either the reward level (scaled) or switching frequency over time. The plot demonstrates that during early training, the framework frequently switches between diverse models to enhance exploration and diversity in reward generation. In later stages, model selection stabilizes, with the framework consistently

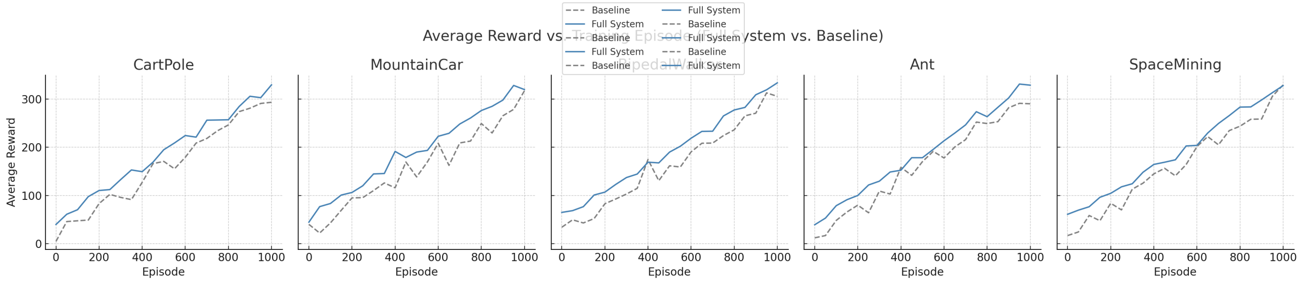

Fig. 3. Average reward over training episodes in five environments. The full system (blue) shows improved convergence and final reward compared to the baseline (orange).

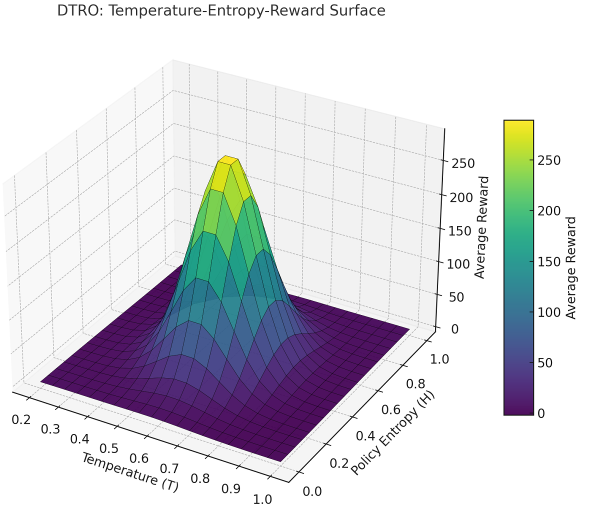

Fig. 4. Temperature-entropy-reward correlation heatmap under DTRO. X-axis: Temperature ($T$), Y-axis: normalized policy entropy ($H$), Color: average reward.

choosing models yielding the highest rewards for efficient policy refinement. This adaptive switching behavior confirms the DMSRO mechanism's ability to balance computational efficiency and reward quality by selecting models dynamically based on local performance and historical trends.

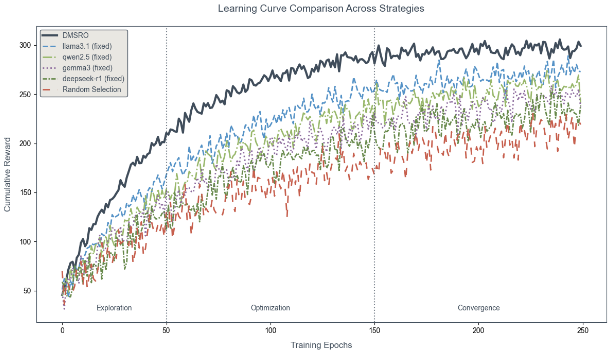

Fig. 5. DMSRO model switching log. X-axis: timestep, color: selected model (LLaMA-3, Qwen-2.5, DeepSeek-R1), line: reward progression. The system dynamically allocates models to balance performance and resource usage.

Table **??** summarizes the reward performance under different model selection strategies. DMSRO achieves an optimal balance between performance and GPU resource consumption, reducing computational hours by approximately 14.8% while maintaining superior reward levels.

### *D. Joint System Performance*

Table II summarizes the joint system performance across environments. Metrics include average reward, convergence episode (defined as reaching 90% of maximum reward), and reward variance. Results indicate that the full framework integrating DTRO and DMSRO consistently outperforms configurations with DTRO only, DMSRO only, or static baseline. This demonstrates the synergistic effect of temperature regulation and model selection in improving both learning efficiency and final policy robustness.

TABLE II
JOINT SYSTEM PERFORMANCE COMPARISON ACROSS ENVIRONMENTS

| **Env** | **Reward** (↑) | **Conv.**(↓) | **Var.** (↓) | **Config** |
|---|---|---|---|---|
| CartPole | 210.7 | 75 | 8.5 | DTRO + DMSRO |
| MountainCar | 93.5 | 140 | 12.3 | DTRO + DMSRO |
| BipedalWalker | 312.4 | 480 | 38.6 | DTRO + DMSRO |
| Ant | 2750.9 | 980 | 201.4 | DTRO + DMSRO |
| SpaceMining | 134.2 | 560 | 45.8 | DTRO + DMSRO |

Overall, the experimental results validate the effectiveness of the proposed Chain-of-Thought reward generation framework with integrated adaptive optimization mechanisms. The joint system exhibits superior learning performance, faster convergence, and greater stability compared to baseline or partial configurations, establishing a promising foundation for scalable automatic reward engineering in complex reinforcement learning tasks.

Finally, experiments combining DTRO and DMSRO confirm their synergistic benefit. Figure 6 illustrates the comparative performance of four system configurations: baseline, DTRO only, DMSRO only, and the full system. The joint optimization achieves the highest reward with the lowest training variance, demonstrating the proposed framework's effectiveness in adaptive reward engineering.

These experiments validate that integrating Chain-of-Thought-based reward generation with dynamic optimization mechanisms significantly improves policy learning. DTRO

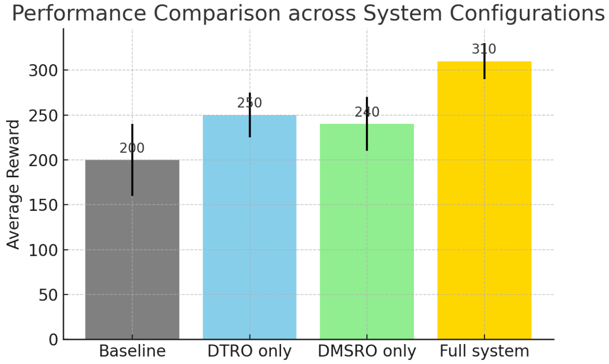

Fig. 6. Performance comparison across system configurations. Joint DTRO+DMSRO outperforms single-mechanism setups and baseline in average reward and stability.

provides adaptive exploration-exploitation balancing, while DMSRO leverages diverse model capabilities under resource constraints. Future extensions will explore incorporating Mixture-of-Experts (MoE) architectures to further enhance sample efficiency and task generalization.

## V. Discussion

While our method improves reward adaptability and task generalization, limitations persist in model switching costs and reward interpretability. Future efforts may include structured prompting [19], [28], hybrid reward learning [43], and MoE architecture integration.

## VI. Conclusion

We propose a dual-dynamic optimization framework for CoT-based RL reward generation, incorporating temperature regulation and model selection. Our results demonstrate improved convergence, stability, and reward quality across tasks, laying a foundation for scalable, self-adaptive reward engineering.

## References

[1] R. S. Sutton, A. G. Barto *et al.*, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[2] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103500, 2021.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[4] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[5] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Icml*, vol. 99, 1999, pp. 278–287.

[6] S. Singh, R. L. Lewis, A. G. Barto, and J. Sorg, "Intrinsically motivated reinforcement learning: An evolutionary perspective," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 70–82, 2010.

[7] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," *arXiv preprint arXiv:1810.12894*, 2018.

[8] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[9] T. Xie, S. Zhao, C. H. Wu, Y. Liu, Q. Luo, V. Zhong, Y. Yang, and T. Yu, "Text2reward: Automated dense reward function generation for reinforcement learning," *arXiv preprint arXiv:2309.11489*, 2023.

[10] S. Sun, R. Liu, J. Lyu, J.-W. Yang, L. Zhang, and X. Li, "A large language model-driven reward design framework via dynamic feedback for reinforcement learning," *arXiv preprint arXiv:2410.14660*, 2024.

[11] Z. Xie, J. Gao, L. Li, Z. Li, Q. Liu, and L. Kong, "Jailbreaking as a reward misspecification problem," *arXiv preprint arXiv:2406.14393*, 2024.

[12] Y. Wu, Z. Sun, H. Yuan, K. Ji, Y. Yang, and Q. Gu, "Self-play preference optimization for language model alignment," *arXiv preprint arXiv:2405.00675*, 2024.

[13] M. Song, Z. Su, X. Qu, J. Zhou, and Y. Cheng, "Prmbench: A fine-grained and challenging benchmark for process-level reward models," *arXiv preprint arXiv:2501.03124*, 2025.

[14] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[15] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, and C. A. Raffel, "Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 1950–1965, 2022.

[16] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[17] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[18] DeepSeek-AI, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint*, 2023.

[19] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig, "Pal: Program-aided language models," pp. 10 764–10 799, 2023.

[20] K. Shum, S. Diao, and T. Zhang, "Automatic prompt augmentation and selection with chain-of-thought from labeled data," *arXiv preprint arXiv:2302.12822*, 2023.

[21] S. Pitis, M. R. Zhang, A. Wang, and J. Ba, "Boosted prompt ensembles for large language models," *arXiv preprint arXiv:2304.05970*, 2023.

[22] Z. Ling, Y. Fang, X. Li, Z. Huang, M. Lee, R. Memisevic, and H. Su, "Deductive verification of chain-of-thought reasoning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 36 407–36 433, 2023.

[23] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang *et al.*, "Self-refine: Iterative refinement with self-feedback," *arXiv preprint arXiv:2303.17651*, 2023.

[24] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le *et al.*, "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[25] D. Dua, S. Gupta, S. Singh, and M. Gardner, "Successive prompting for decomposing complex questions," *arXiv preprint arXiv:2212.04092*, 2022.

[26] J. Wang, Q. Sun, X. Li, and M. Gao, "Boosting language models reasoning with chain-of-knowledge prompting," *arXiv preprint arXiv:2306.06427*, 2023.

[27] J. Wang, J. Li, and H. Zhao, "Self-prompted chain-of-thought on large language models for open-domain multi-hop reasoning," *arXiv preprint arXiv:2310.13552*, 2023.

[28] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," *arXiv preprint arXiv:2211.12588*, 2022.

[29] T. X. Olausson, A. Gu, B. Lipkin, C. E. Zhang, A. Solar-Lezama, J. B. Tenenbaum, and R. Levy, "Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers," *arXiv preprint arXiv:2310.15164*, 2023.

[30] A. Parisi, Y. Zhao, and N. Fiedel, "Talm: Tool augmented language models," *arXiv preprint arXiv:2205.12255*, 2022.

[31] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, “Llm+ p: Empowering large language models with optimal planning proficiency,” *arXiv preprint arXiv:2304.11477*, 2023.

[32] J. Pan, S. Deng, and S. Huang, “Coat: Chain-of-associated-thoughts framework for enhancing large language models reasoning,” *arXiv preprint arXiv:2502.02390*, 2025.

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

[34] S. Xu, W. Xie, L. Zhao, and P. He, “Chain of draft: Thinking faster by writing less,” *arXiv preprint arXiv:2502.18600*, 2025.

[35] Y. Zhu, J. Li, G. Li, Y. Zhao, Z. Jin, and H. Mei, “Hot or cold? adaptive temperature sampling for code generation with large language models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 437–445.

[36] N. Cecere, A. Bacciu, I. F. Tobías, and A. Mantrach, “Monte carlo temperature: a robust sampling strategy for llm’s uncertainty quantification methods,” *arXiv preprint arXiv:2502.18389*, 2025.

[37] S. Zhang, Y. Bao, and S. Huang, “Edt: Improving large language models’ generation by entropy-based dynamic temperature sampling,” *arXiv preprint arXiv:2403.14541*, 2024.

[38] M. Peeperkorn, T. Kouwenhoven, D. Brown, and A. Jordanous, “Is temperature the creativity parameter of large language models?” *arXiv preprint arXiv:2405.00492*, 2024.

[39] E. Evstafev, “The paradox of stochasticity: Limited creativity and computational decoupling in temperature-varied llm outputs of structured fictional data,” *arXiv preprint arXiv:2502.08515*, 2025.

[40] M. Nguyen, A. Baker, C. Neo, A. Roush, A. Kirsch, and R. Shwartz-Ziv, “Turning up the heat: Min-p sampling for creative and coherent llm outputs,” *arXiv preprint arXiv:2407.01082*, 2024.

[41] W. Fedus, B. Zoph, and N. Shazeer, “Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2022, p. 287–311.

[42] N. Du, Y. Li, Z. Dai, N. Shazeer, W. Fedus, M. Tan, O. Vinyals, Q. Le, J. Dean, Z. Chen *et al.*, “Glam: Efficient scaling of language models with mixture-of-experts,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 5712–5721.

[43] J. Skalse, N. Howe, D. Krasheninnikov, and D. Krueger, “Defining and characterizing reward gaming,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 9460–9471, 2022.