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*—Reward function design is critical for reinforcement learning (RL) agents, yet traditional approaches rely heavily on manual expertise, limiting adaptability to complex environments. This paper proposes a Chain-of-Thought (CoT)–based reward generation framework leveraging large language models (LLMs) for structured reasoning, 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 design remains one of the most critical and challenging aspects of reinforcement learning (RL). 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 [1]. As RL tasks become increasingly complex and deployed in real-world scenarios with dynamic objectives, there is an urgent need for automated, interpretable, and adaptive reward design frameworks.

Recent advances in large language models (LLMs) have demonstrated remarkable reasoning and generalization capabilities across domains [2], [3]. In particular, Chain-of-Thought (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 [4]. 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 [5] and model selection strategies [6], 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

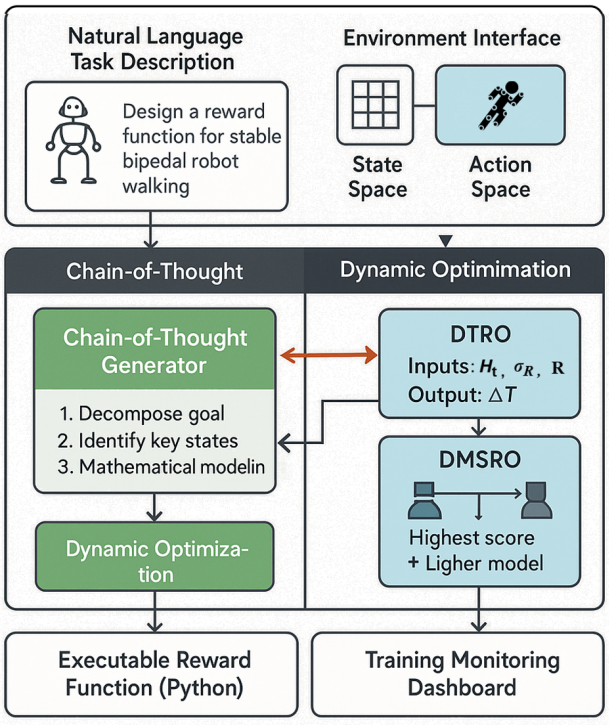

Fig. 1. Reward function design framework. Top: Natural language task specification and environment interface. Middle: Chain-of-Thought reasoning (left) and dynamic optimization modules (right). Bottom: Executable output and monitoring system.

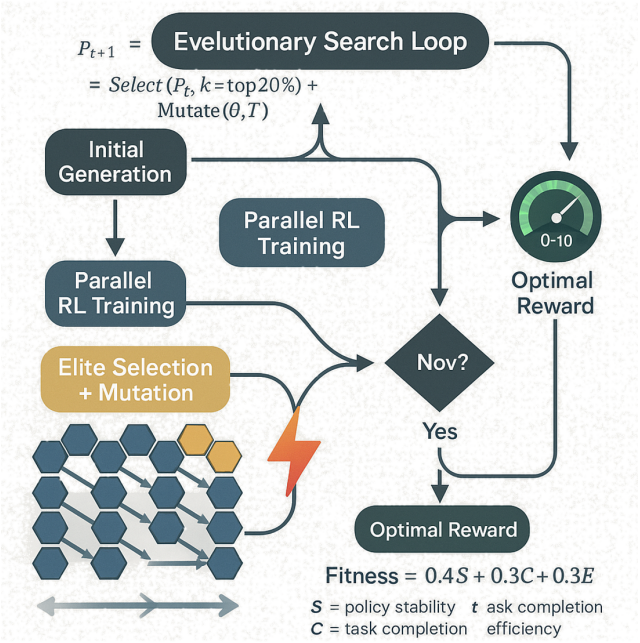

Fig. 2. Evolutionary search loop for reward optimization. The process iteratively refines reward functions through parallel RL training, with fitness evaluation driving selection and mutation.

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

Traditional reward engineering methods, including reward shaping [1], [7] and inverse reinforcement learning [8], offer theoretical foundations but lack scalability. Recent LLM-based systems such as EUREKA [9] and Text2Reward [10] leverage natural language but typically operate under static configurations. Chain-of-Thought prompting enhances reasoning capabilities [11], [12], while temperature control [5], [13], [14] and model adaptation [6], [15] remain underexplored in reward generation.

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:

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

C. Dynamic Optimization

The evolutionary loop in Fig. 2 operates through:

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

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:

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

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 Φ 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:

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

An ϵ-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

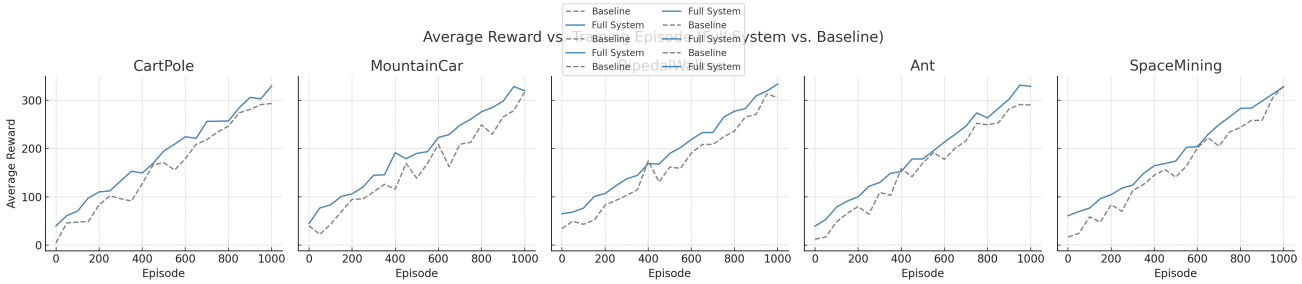

Fig. 3. Average reward over training episodes in five environments. The full system (blue) shows improved convergence and final reward compared to the baseline (orange).

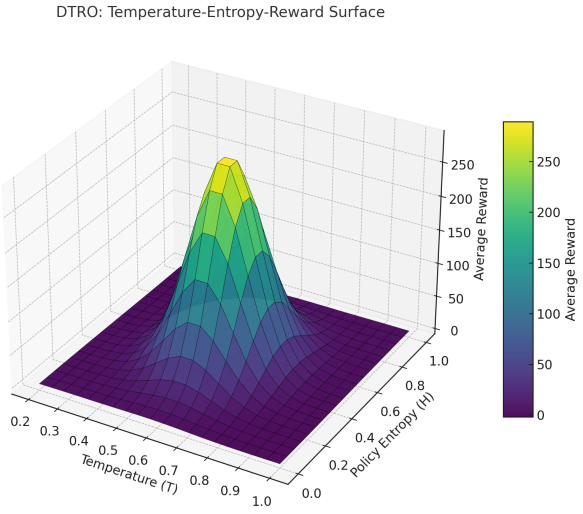

Fig. 4. Temperature-entropy-reward correlation heatmap under DTRO. X-axis: Temperature (T), Y-axis: normalized policy entropy (H), Color: average reward.

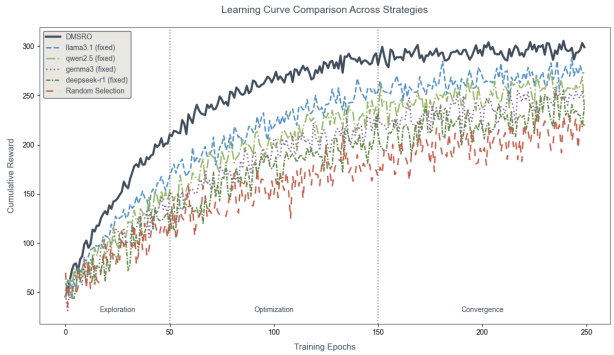

Fig. 5. DMSRO model switching log. X-axis: timestep, color: selected model (LLaMA-3, Qwen-2.5, DeepSeek-R1), line: reward progression. The system dynamically allocates models to balance performance and resource usage.

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 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.

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.

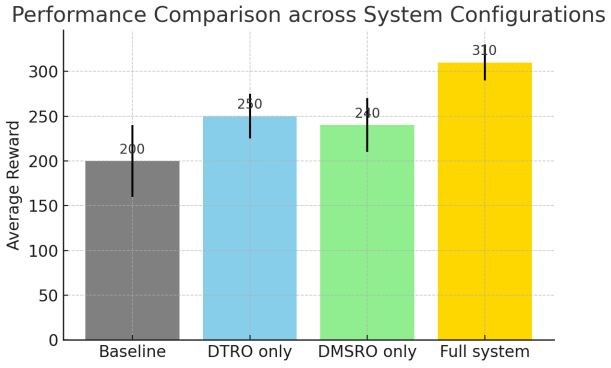

Fig. 6. Performance comparison across system configurations. Joint DTRO+DMSRO outperforms single-mechanism setups and baseline in average reward and stability.

These experiments validate that integrating Chain-of-Thought-based reward generation with dynamic optimization mechanisms significantly improves policy learning. DTRO 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 [16], [17], hybrid reward learning [18], 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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] C.-Y. Hsieh, C.-L. Li, C.-K. Yeh, H. Nakhost, Y. Fujii, A. Ratner, R. Krishna, C.-Y. Lee, and T. Pfister, "Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes," *arXiv preprint arXiv:2305.02301*, 2023.

[7] Y. Hu, W. Wang, H. Jia, Y. Wang, Y. Chen, J. Hao, F. Wu, and C. Fan, "Learning to utilize shaping rewards: A new approach of reward shaping," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 931–15 941, 2020.

[8] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] K. Vardhni, G. Devaraja, R. Dharhshita, R. K. Chowdary, and A. Mahadevan, "Performance evaluation and comparative ranking of llm variants in entity relationship prediction," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2024, pp. 1–7.

[16] 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.

[17] 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.

[18] 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.