Trustworthiness-as-Reward: Improving LLM Performance on Text Classification through Reinforcement Learning

Yiqing Zhao[1], Xiaohui Shen[2] and Lanfeng Pan[3,*]

[1] *Amazon.com, Inc., 4575 La Jolla Village Dr, San Diego, CA, USA*
[2] *Amazon.com, Inc., 500 Boren Ave N, Seattle, WA, USA*
[3] *Amazon.com, Inc., 4575 La Jolla Village Dr, San Diego, CA, USA*

Abstract

Text classification has become increasingly important with the exponential growth of digital text data, finding applications in sentiment analysis, spam detection, topic categorization, and content moderation across various domains. Our research introduced a novel approach that integrates reinforcement learning with a specialized reasoning path. This methodology enabled smaller 7B parameter language models to increase performance significantly to the level comparable to larger models e.g. Claude 3.7, on an open source Pubmed multilabel text classification task. We experimented with 1) Claude 3.7 and DeepSeek-R1-Distill-Qwen-7B (Qwen-7B) zero shot, 2) Supervised Fine-Tuned (SFT) Qwen-7B, 3) Reinforcement Learning (RL) Qwen-7B and 4) SFT + RL Qwen-7B. We also experimented with different reasoning paths: 1) no reasoning, and 2) Socratic reasoning, as well as different evaluation metrics as reward: 1) F1 score as reward, 2) Trustworthiness (or reasoning process accuracy) as reward. The training data are composed of ~11,000 pubmed publication abstracts. We evaluated the performance in another ~1,000 abstract. SFT + RL Qwen-7B with Socratic reasoning and F1 score as reward achieved the highest F1 score of 0.8348. In summary, we proposed an innovative post-training paradigm integrating SFT, RL, Socratic reasoning path, and Trustworthiness-as-Reward. With this paradigm, we were able to double the F1 score compared to the base 7B model and achieved a ~ 0.15 lift in F1 score compared to using SFT alone without reasoning. Our pipeline demonstrates that strategic optimization of smaller models can achieve superior results compared to simply scaling up the model size.

Keywords

Large Language Models, Reinforcement Learning, Reasoning, Text classification

1. Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks, including text classification. Their performance can often be further enhanced through supervised fine-tuning (SFT). However, in our previous experiments on various datasets, it was found that using SFT alone cannot increase the performance of 7B models to the level of Claude 3.7 on various tasks. This finding motivates us to explore opportunities to further improve the 7B models through reinforcement learning and reasoning path design. Moreover, we propose increasing the trustworthiness of LLM reasoning process to increase the LLM's performance on the final task.

This paper introduces a novel approach that integrates SFT, reinforcement learning (RL) with "Trustworthiness-as-Reward" and a specialized reasoning path to improve LLM performance specifically on text classification tasks. Using the evaluation metrics (F1-score or Trustworthiness) of classification tasks as reward signals, we create a feedback loop that allows the model to learn from its own predictions and gradually improve its performance. This method offers a promising direction for optimizing LLMs through RL in a task-specific, metric-specific manner. Our approach not only aims to boost classification performance but also explores the potential of reasoning path for self-improvement in language models, paving the way for more adaptive and efficient AI systems in the field of natural language processing.

The outline of the paper includes: Background and related work survey (Section 2), the Proposed method (Section 3), Experiments (Section 4), and Conclusions (Section 5).

TRUST-AI: The European Workshop on Trustworthy AI. Organized as part of the European Conference of Artificial Intelligence - ECAI 2025. October 2025, Bologna, Italy.

[*] Corresponding author.

zhaoyiqi@amazon.com (Y. Zhao); xiaohus@amazon.com (X. Shen); planfeng@amazon.com (L. Pan)

2. Background

2.1. LLM for text classification

Recent advances in text classification using LLMs have demonstrated diverse approaches to enhance classification performance while addressing computational and resource constraints. Several innovative methods have emerged, including self-training techniques where LLMs generate augmented training data and assist smaller models [1], ensemble approaches that combine multiple LLMs with traditional machine learning classifiers [2], and knowledge distillation frameworks where LLMs serve as teachers for smaller student models [3, 4].

Researchers have also explored adaptive boosting frameworks, such as RGPT, which creates specialized classification LLMs through recurrent ensemble of base learners [5]. To address the challenge of minimal supervision, some studies have developed methods that combine LLMs with taxonomy enrichment and corpus-specific features [6], while others have integrated LLMs within active learning frameworks to optimize human annotation efforts [7]. Cost-effective approaches have gained attention, with some researchers proposing multi-stage in-context learning methods [8] and others focusing on domain-specific fine-tuning strategies [9]. The integration of instruction fine-tuning has also shown promise in improving classification performance for specific domains [10]. A notable trend across these studies is the focus on reducing computational resources and annotation costs while maintaining or improving classification accuracy.

2.2. Reinforcement Learning for LLM

Reinforcement Learning (RL) is a learning approach where an AI agent learns through trial and error by interacting with its environment. Instead of being directly taught, it discovers optimal actions by receiving feedback (rewards or penalties) based on its choices. Natural Language Processing (NLP), particularly in modern LLMs, shares some fundamental connections with RL.

The transformer architecture used in modern LLMs can be viewed as implementing a sophisticated form of RL's state-action mapping, where the attention mechanism helps determine the most relevant context (state) for generating the next token (action). Recent trends to combine RL with LLMs target 1) model performance improvement through fine-tuning and 2) prompt optimization. Fine-tuning methods modify the LLM's parameters, while prompt optimization methods focus on improving how we interact with unchanged models.

2.2.1. RL-Fine tuning

Human input plays an important role in RL-Fine tuning. Human input can be incorporated into fine-tuning through two main channels: policy model training (where humans demonstrate desired LLM behavior) and reward model training (where humans rank LLM outputs). In one study [11], researchers utilized reinforcement learning to predict human-preferred Reddit post summaries, using a supervised learning model as a reward function. The approach, which used Proximal Policy (PPO) Optimization Algorithms [12] for fine-tuning, proved more effective than traditional NLP metrics like ROUGE in generating summaries aligned with human preferences. Instruct-GPT, developed by Ouyang et al. [13], demonstrated improved truthfulness and harmlessness through a three-step process: First, training a policy model using human-demonstrated behaviors; second, developing a reward model trained on human-ranked outputs; and third, fine-tuning LLM using RL with the reward model. The result showed enhanced performance while maintaining generalization capabilities.

2.2.2. RL-Prompt optimization

Prompt optimization can often align LLM behavior with human preferences without the computational burden of fine-tuning. Most studies focus on tuning soft prompts (e.g., embeddings), which are difficult to interpret and non-transferable across different LLMs [14]. On the other hand, discrete prompts, which consist of concrete tokens from vocabulary, are hard to optimize efficiently. Recent studies have explored using RL to optimize discrete prompts, aiming to enhance LLM performance across various tasks with minimal training data. RLPROMPT, developed by Deng et al. [14], takes a different approach by training a transferable policy network for prompt generation. Their research revealed that effective prompts don't necessarily need to follow human language patterns, often appearing as grammatical "gibberish." Unlike TEMPERA, which requires access to embedding vectors, RLPROMPT treats the LLM as a black box and considers the entire vocabulary as potential actions.

2.3. Reasoning for LLM

Improvements in LLM reasoning are closely tied to advancements in a variety of techniques in inference scaling at test time and learning-to-reason at training time. On the other hand, the release of Reasoning Language Models (RLMs) such as OpenAI's o1 and DeepSeek's R1, marked a significant increase in research dedicated to learning-to-reason approaches.

2.3.1. Inference Scaling

While Chain-of-Thought (CoT) laid the groundwork, researchers have developed more complex frameworks such as Tree-of-Thought (ToT) and Forest-of-Thought (FoT), with the latter introducing sparse activation and dynamic self-correction strategies for improved efficiency [15]. Some works have focused on verification-based approaches, combining multiple reasoning paths with specialized verifiers to assess and rank outputs [16]. The GLoRe framework introduced Stepwise Outcome Reward Models (SORMs) trained on synthetic data to detect incorrect reasoning steps and implement both global and local refinements [17]. Some researchers have explored bidirectional reasoning through reverse thinking strategies [18], while others have focused on inference-time computation scaling [19] and automated reasoning chain evaluation methods [20]. The field has also seen advances in controlling reasoning processes through strategic thinking intervention [21] and grounding explanations in explicit reasoning sequences [22].

2.3.2. Learning-to-reason

Recent advances in LLM reasoning studies have seen a surge in RLMs that simulate inference, generating trajectories that capture potential reasoning paths using supervised and/or reinforcement learning. Training innovations have included the development of preference trees for comprehensive reasoning alignment [23] and rule-based reinforcement learning approaches using synthetic logic puzzles [24]. Process reward models (PRMs) have emerged as a promising direction, with innovations like step-level advantages and process advantage verifiers (PAVs) showing improvements in both accuracy and compute efficiency [25]. While traditional RLHF methods remain influential, [26] reveals that different algorithms like Expert Iteration, PPO, and Return-Conditioned RL perform comparably well for improving reasoning capabilities. Novel approaches include offline RL methods, with [27] introducing OREO, which jointly optimizes a policy model and value function using the soft Bellman Equation, showing superior performance on mathematical reasoning tasks. Several papers explore domain-specific applications, such as [28]'s SWE-RL, which employs a lightweight rule-based reward system for software engineering tasks, and [29]'s Rank-R1, which enhances document reranking through RL-based reasoning. More recent developments include[30]'s ReSearch framework, which integrates search operations into the reasoning chain without supervised data on reasoning steps, and [31]'s DAPO algorithm, which introduces decoupled clip and dynamic sampling policy optimization for large-scale RL training. These advancements are characterized by diverse reward mechanisms, from simple rule-based approaches to more sophisticated joint optimization strategies, all contributing to enhanced reasoning capabilities in LLMs.

3. Proposed Method

The purpose of the paper is to compare and identify the best LLM setup for Pubmed publication category classification. Specifically, we introduced novel approaches leveraging 'Socratic reasoning' and 'Trustworthiness-as-Reward' to improve classification performance. 'Socratic reasoning' refers to the prompt instructions to guide LLM to examine several key aspects for the classification task through Question and Answer (QA). 'Trustworthiness' score was calculated by evaluating the answer accuracy of LLMs to the fifteen Socratic questions provided in the prompt. Our definition of 'Trustworthiness' is not focused on LLM's internal mechanism behind the generation process. It aims to measure how well LLM collects all the useful information from the input and how accurate LLM understand key aspects of the input. The Socratic QA reasoning path + 'Trustworthiness-as-Reward' helps LLM to examine all useful information from the input and corrects incorrect understanding of the input during RL. With our hypothesis, this setup could provide users with the most accurate and trustworthy final answer that is based a correct and thorough synthesis of the input.

We experimented with several post-training methods: 1) Claude 3.7/DeepSeek-R1-Distill-Qwen-7B (Qwen-7B) zero-shot, 2) Supervised Fine-Tuned (SFT) Only, 3) Reinforcement Learning (RL) Only and 4) SFT + RL for Qwen-7B. We also experimented with different reasoning path prompt for inference optimization: 1) no reasoning, and 2) Socratic reasoning. Finally, we compared using F1-score as reward vs using F1-score + Trustworthiness' as reward during RL training.

3.1. Data Summary

We leveraged an open source Kaggle dataset [32] to select training and test data for our experiments. Our training data includes ~11,000 randomly selected Pubmed publication abstracts from the original Kaggle dataset. Our test data is another randomly selected sample of ~1,100 Pubmed publication abstracts. The output used as training material was generated by Claude 3.7. Claude 3.7 was given the gold standard publication category and was instructed to provide reasoning (when applicable) on why the publication should belong to those categories. On average, each publication has 5.7 MeSH tags associated with it.

3.2. Models

We compared three different base LLMs: 1) Claude 3.7 from Anthropic, released on 02/19/2025 and 2) DeepSeek-R1-Distill Qwen-7B [33] to be the base model for the task.

3.3. LLM Post-training Setup

We experimented with four different post-training setups of LLM: 1) Zero-shot, 2) SFT-only, 3) RL-only, 4) SFT + RL. For SFT + RL, we performed SFT first, followed by RL as a second step. For RL, we adopted Group Relative Policy Optimization (GRPO) algorithm [34], a reinforcement learning algorithm that extends the concept of Proximal Policy Optimization (PPO) to handle group fairness constraints in decision-making systems. It was introduced as a method to address fairness concerns in reinforcement learning while maintaining good performance. For both SFT and RL, we employed QLoRA (Quantized Low-Rank Adaptation) [35] for both SFT and RL phases. QLoRA enables efficient model tuning by utilizing 4-bit quantization and low-rank adapters while maintaining model quality. This approach significantly reduces memory requirements compared to full-parameter fine-tuning, allowing the training of large language models on consumer-grade hardware. For RL, we tried three different reward functions, including: 1) F1-score as reward, 2) Trustworthiness (or reasoning process accuracy) as reward. 'Trustworthiness' score was calculated by evaluating the answer accuracy of LLMs to the fifteen Socratic questions provided in the prompt.

3.4. Reasoning Path

We experimented three different prompt instructions (see Figure 1): 1) No reasoning: provide LLM with direct instruction of the task, which is to assign fifteen categories to each publication, 2) Socratic reasoning: asking LLM to answer fifteen binary (Yes/No) questions, which were designed based on domain knowledge (also provided by Claude 3.7) of the definitions of the fifteen categories.

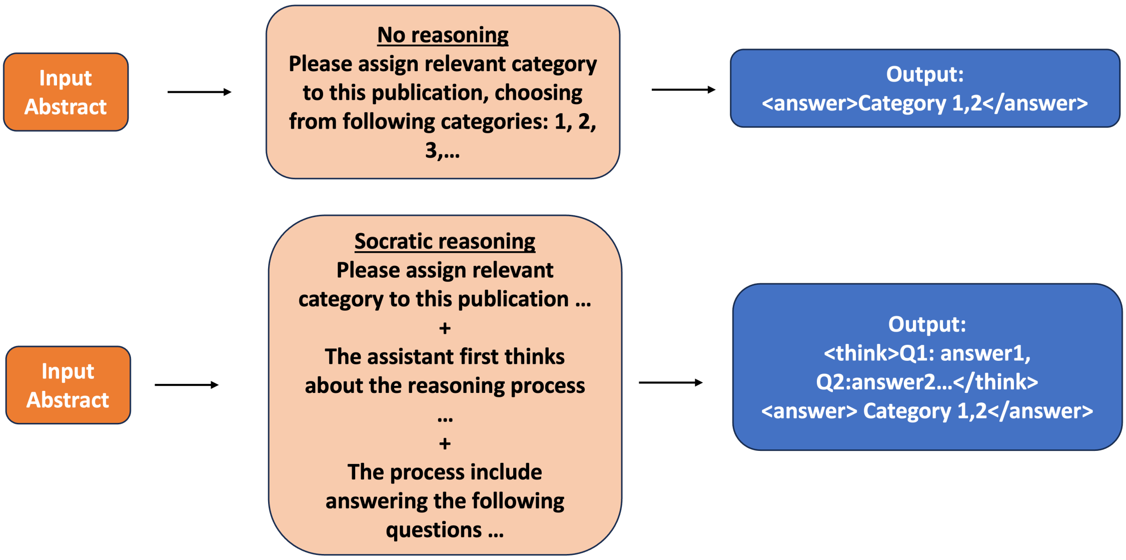
Input Abstract
No reasoning
Please assign relevant category to this publication, choosing from following categories: 1, 2, 3,…
Output:
<answer>Category 1,2</answer>
Input Abstract
Socratic reasoning
Please assign relevant category to this publication …
+
The assistant first thinks about the reasoning process
…
+
The process include answering the following questions …
Output:
<think>Q1: answer1, Q2:answer2…</think>
<answer> Category 1,2</answer>

Figure 1: Comparison between Different Reasoning Paths.

3.5. Evaluation Metric

We used two metrics for evaluation of model performance: F1-score and Trustworthiness. F1-score was calculated based on Precision and Recall calculated based on gold standard topic list vs model output topic list. Trustworthiness was calculated based on the accuracy of answer fifteen questions during reasoning process. During RL training, we used F1 or Trustworthiness as reward, and applied the same logic each generated output and feedback to the model.

4. Experiments

In this section, we will share and compare performance with different reasoning paths, different training methods, and different reward function for RL.

Table 1 shows the F1 score of Claude 3.7 performance on the task of Pubmed publication classification, using different reasoning paths. It shows with a Socratic reasoning path, Claude 3.7 can achieve a ~ 2% lift in F1 score compared not instructed to reason. F1 lift mostly comes from improving on Recall (~ 8% lift), which is aligned with our hypothesis that using Socratic reasoning can induce more thorough examination of details in the abstract and reduce information skipped or missed during summarization.

Table 2 shows F1 score of fine-tuned Qwen-7B's performance, using different reasoning paths. With only SFT, providing no reasoning instructions achieves a comparable F1 score compare to using Socratic reasoning. However, when combining with R1, training using Socratic reasoning path provide most notable improvement on model performance (~ 4% lift for SFT+RL vs SFT-only). Although guiding language models with Socratic reasoning path improved Recall, it does not lead to better precision than models without such reasoning instructions. The addition of complex reasoning prompts may actually increase the likelihood of errors and fabricated responses. This highlights the critical need to enhance the reliability and accuracy of model outputs to achieve better precision and overall performance (F1-score).

Table 1
Comparison of Pubmed publication Classification Performance of Claude 3.7 with or without Reasoning Path

Reasoning	Precision	Recall	F1
No	0.8611	0.5095	0.6402
Socratic	0.8476	0.5426	0.6616

Table 2
Comparison of Pubmed publication Classification Performance of Qwen-7B with or without Reasoning Path

Method	Reasoning	Precision	Recall	F1
SFT	No	0.8229	0.8163	0.8196
SFT	Socratic	0.7625	0.8118	0.7864
SFT+RL - F1	No	0.8654	0.7781	0.8194
SFT+RL - F1	Socratic	0.7743	0.8739	0.8211

Table 3
Comparison of Pubmed publication Classification Performance of Qwen-7B using Training Methods and Rewards with Socratic Reasoning Path

Method	Reasoning	Precision	Recall	F1	Trustworthiness
Zero-shot	Socratic	0.4416	0.2792	0.3421	0.0238
SFT	Socratic	0.7464	0.8668	0.8021	0.8017
RL - F1	Socratic	0.6201	0.6578	0.6384	0.0025
SFT+RL - F1	Socratic	0.7743	0.8739	0.8211	0.8425
SFT+RL - Trustworthiness+F1	Socratic	0.8246	0.8452	0.8348	0.8625

Table 3 provides an overview of performance for models trained with Socratic reasoning path but using different training paradigm and reward functions. Specifically, we introduced a new metric/reward function - 'Trustworthiness' that calculates how accurately language models answer a set of fifteen Socratic questions in the prompt. We can see the performance for zero-shot baseline is poor. However, Both SFT and RL individually led to substantial improvements. SFT proved more effective than RL alone because using RL alone does not induce LLM's ability to follow instructed reasoning path. We observed that the combination of SFT and RL techniques produced the best results with the Qwen-7B model. Moreover, we observed that using Trustworthiness + F1 as reward generates superior performance (~ 1%) than using F1 score as reward alone. The lift in F1 score is mostly contributed by lift in Precision by ~ 6% and lift in Trustworthiness by ~ 2% ('SFT+RL - F1' vs 'SFT+RL - Trustworthiness+F1'). This proves that 'Trustworthiness' is a very important metrics that will help to improve LLM's performance in text classification tasks when used as a reward function during RL.

5. Conclusions

In summary, we proposed an innovative LLM training paradigm combining SFT, RL, Socratic reasoning path and Trustworthiness reward. By leveraging RL with Socratic reasoning path and Trustworthiness as reward, our paradigm effectively enhances the LLM's capacity for learning and reasoning, while optimizing key performance metrics: we were able to double the F1 score compared to the base 7B model and achieved a ~ 0.15 lift in F1 score compared to using SFT alone without reasoning. This innovative combination proposed a new direction to maximize the potential of LLMs in various applications.

Declaration on Generative AI

The author(s) has not employed any Generative AI tools to write this manuscript.

References

[1] R. Zhang, Y.-S. Wang, Y. Yang, Generation-driven contrastive self-training for zero-shot text classification with instruction-following llm, arXiv preprint arXiv:2304.11872 (2023).

[2] H. Abburi, M. Suesserman, N. Pudota, B. Veeramani, E. Bowen, S. Bhattacharya, Generative ai text classification using ensemble llm approaches, arXiv preprint arXiv:2309.07755 (2023).

[3] N. Pangakis, S. Wolken, Knowledge distillation in automated annotation: Supervised text classification with llm-generated training labels, arXiv preprint arXiv:2406.17633 (2024).

[4] T. Kuzman, N. Ljubešić, Llm teacher-student framework for text classification with no manually annotated data: A case study in iptc news topic classification, IEEE Access (2025).

[5] Y. Zhang, M. Wang, C. Ren, Q. Li, P. Tiwari, B. Wang, J. Qin, Pushing the limit of llm capacity for text classification, arXiv preprint arXiv:2402.07470 (2024).

[6] Y. Zhang, R. Yang, X. Xu, R. Li, J. Xiao, J. Shen, J. Han, Teleclass: Taxonomy enrichment and llm-enhanced hierarchical text classification with minimal supervision, in: Proceedings of the ACM on Web Conference 2025, 2025, pp. 2032–2042.

[7] H. Rouzegar, M. Makrehchi, Enhancing text classification through llm-driven active learning and human annotation, arXiv preprint arXiv:2406.12114 (2024).

[8] M. Liu, G. Shi, Poliprompt: A high-performance cost-effective llm-based text classification framework for political science, Available at SSRN 4940136 (2024).

[9] F. Wei, R. Keeling, N. Huber-Fliflet, J. Zhang, A. Dabrowski, J. Yang, Q. Mao, H. Qin, Empirical study of llm fine-tuning for text classification in legal document review, in: 2023 IEEE International Conference on Big Data (BigData), IEEE, 2023, pp. 2786–2792.

[10] K. Yin, C. Liu, A. Mostafavi, X. Hu, Crisissense-llm: Instruction fine-tuned large language model for multi-label social media text classification in disaster informatics, arXiv preprint arXiv:2406.15477 (2024).

[11] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, P. F. Christiano, Learning to summarize with human feedback, Advances in neural information processing systems 33 (2020) 3008–3021.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[13] 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 35 (2022) 27730–27744.

[14] M. Deng, J. Wang, C.-P. Hsieh, Y. Wang, H. Guo, T. Shu, M. Song, E. P. Xing, Z. Hu, Rlprompt: Optimizing discrete text prompts with reinforcement learning, arXiv preprint arXiv:2205.12548 (2022).

[15] Z. Bi, K. Han, C. Liu, Y. Tang, Y. Wang, Forest-of-thought: Scaling test-time compute for enhancing llm reasoning, arXiv preprint arXiv:2412.09078 (2024).

[16] Z. Liang, Y. Liu, T. Niu, X. Zhang, Y. Zhou, S. Yavuz, Improving llm reasoning through scaling inference computation with collaborative verification, arXiv preprint arXiv:2410.05318 (2024).

[17] A. Havrilla, S. Raparthy, C. Nalmpantis, J. Dwivedi-Yu, M. Zhuravinskyi, E. Hambro, R. Raileanu, Glore: When, where, and how to improve llm reasoning via global and local refinements, arXiv preprint arXiv:2402.10963 (2024).

[18] J. C.-Y. Chen, Z. Wang, H. Palangi, R. Han, S. Ebrahimi, L. Le, V. Perot, S. Mishra, M. Bansal, C.-Y. Lee, et al., Reverse thinking makes llms stronger reasoners, arXiv preprint arXiv:2411.19865 (2024).

[19] S. Parashar, B. Olson, S. Khurana, E. Li, H. Ling, J. Caverlee, S. Ji, Inference-time computations for llm reasoning and planning: A benchmark and insights, arXiv preprint arXiv:2502.12521 (2025).

[20] S. Hao, Y. Gu, H. Luo, T. Liu, X. Shao, X. Wang, S. Xie, H. Ma, A. Samavedhi, Q. Gao, et al., Llm reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models, arXiv preprint arXiv:2404.05221 (2024).

[21] T. Wu, C. Xiang, J. T. Wang, P. Mittal, Effectively controlling reasoning models through thinking intervention, arXiv preprint arXiv:2503.24370 (2025).

[22] V. Cahlik, R. Alves, P. Kordik, Reasoning-grounded natural language explanations for language models, arXiv preprint arXiv:2503.11248 (2025).

[23] L. Yuan, G. Cui, H. Wang, N. Ding, X. Wang, J. Deng, B. Shan, H. Chen, R. Xie, Y. Lin, et al., Advancing llm reasoning generalists with preference trees, arXiv preprint arXiv:2404.02078 (2024).

[24] T. Xie, Z. Gao, Q. Ren, H. Luo, Y. Hong, B. Dai, J. Zhou, K. Qiu, Z. Wu, C. Luo, Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning, arXiv preprint arXiv:2502.14768 (2025).

[25] A. Setlur, C. Nagpal, A. Fisch, X. Geng, J. Eisenstein, R. Agarwal, A. Agarwal, J. Berant, A. Kumar, Rewarding progress: Scaling automated process verifiers for llm reasoning, arXiv preprint arXiv:2410.08146 (2024).

[26] A. Havrilla, Y. Du, S. C. Raparthy, C. Nalmpantis, J. Dwivedi-Yu, M. Zhuravinskyi, E. Hambro, S. Sukhbaatar, R. Raileanu, Teaching large language models to reason with reinforcement learning, arXiv preprint arXiv:2403.04642 (2024).

[27] H. Wang, S. Hao, H. Dong, S. Zhang, Y. Bao, Z. Yang, Y. Wu, Offline reinforcement learning for llm multi-step reasoning, arXiv preprint arXiv:2412.16145 (2024).

[28] Y. Wei, O. Duchenne, J. Copet, Q. Carbonneaux, L. Zhang, D. Fried, G. Synnaeve, R. Singh, S. I. Wang, Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution, arXiv preprint arXiv:2502.18449 (2025).

[29] S. Zhuang, X. Ma, B. Koopman, J. Lin, G. Zuccon, Rank-r1: Enhancing reasoning in llm-based document rerankers via reinforcement learning, arXiv preprint arXiv:2503.06034 (2025).

[30] M. Chen, T. Li, H. Sun, Y. Zhou, C. Zhu, F. Yang, Z. Zhou, W. Chen, H. Wang, J. Z. Pan, et al., Learning to reason with search for llms via reinforcement learning, arXiv preprint arXiv:2503.19470 (2025).

[31] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, T. Fan, G. Liu, L. Liu, X. Liu, et al., Dapo: An open-source llm reinforcement learning system at scale, arXiv preprint arXiv:2503.14476 (2025).

[32] O. Ahmad, Pubmed multilabel text classification dataset mesh, 2022. URL: https://www.kaggle.com/datasets/owaiskhan9654/pubmed-multilabel-text-classification.

[33] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al., Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, arXiv preprint arXiv:2501.12948 (2025).

[34] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al., Deepseek-math: Pushing the limits of mathematical reasoning in open language models, arXiv preprint arXiv:2402.03300 (2024).

[35] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, Advances in neural information processing systems 36 (2023) 10088–10115.