Reinforcement Learning in the Era of Large Language Models: Challenges and Opportunities

QIANYUE HAO, Tsinghua University, China
LIN CHEN, The Hong Kong University of Science and Technology, China
XIAOQIAN QI, Tsinghua University, China
YUAN YUAN, Tsinghua University, China
ZEFANG ZONG, Tsinghua University, China
HONGYI CHEN, Tsinghua University, China
KEYU ZHAO, Tsinghua University, China
SHENGYUAN WANG, Tsinghua University, China
YUNKE ZHANG, Tsinghua University, China
JIAN YUAN, Tsinghua University, China
YONG LI, Tsinghua University, China

Reinforcement learning (RL), a pivotal machine learning paradigm, is becoming essential in the post-training of large language models (LLMs), enhancing their reasoning capabilities and alignment with human preferences. However, adapting conventional RL to LLMs introduces challenges stemming from their massive parameter size and the vast natural language action space. In this survey, we conduct a systematic literature review focusing on how RL algorithms are adapted and scaled as a fundamental post-training tool to enhance LLMs. First, we provide a taxonomy of challenges faced in each stage of the RL training loop, including action exploration, trajectory collection, reward evaluation, and model update. We then introduce recently developed methods to address these issues, ranging from logical structure navigation, training data curation to reward design and advantage estimation. Afterward, we elaborate on the application of these techniques in diverse domains such as mathematics, coding, medicine, and information retrieval, analyzing how innovations in the RL pipeline enhance the domain-specific LLMs. Finally, we discuss the limitations and side effects of applying RL to LLMs and explore open problems and future directions like the balance between efficiency and effectiveness, and algorithm-system co-design. This survey helps researchers understand recent progress and inspire novel research to address current challenges and realize the full potential of RL for LLMs.

Additional Key Words and Phrases: Reinforcement learning, large language models

Authors' addresses: Qianyue Hao, Tsinghua University, Beijing, China; Lin Chen, The Hong Kong University of Science and Technology, Hong Kong, China; Xiaoqian Qi, Tsinghua University, Beijing, China; Yuan Yuan, Tsinghua University, Beijing, China; Zefang Zong, Tsinghua University, Beijing, China; Hongyi Chen, Tsinghua University, Beijing, China; Keyu Zhao, Tsinghua University, Beijing, China; Shengyuan Wang, Tsinghua University, Beijing, China; Yunke Zhang, Tsinghua University, Beijing, China; Jian Yuan, Tsinghua University, Beijing, China; Yong Li, Tsinghua University, Beijing, China, liyong07@tsinghua.edu.cn.

Manuscript submitted to ACM

ACM Reference Format:
Qianyue Hao, Lin Chen, Xiaoqian Qi, Yuan Yuan, Zefang Zong, Hongyi Chen, Keyu Zhao, Shengyuan Wang, Yunke Zhang, Jian Yuan, and Yong Li. 2024. Reinforcement Learning in the Era of Large Language Models: Challenges and Opportunities. *ACM Comput. Surv.* 1, 1 (July 2024), 39 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Reinforcement learning (RL), a machine learning paradigm in which an agent learns to optimize its actions through trial-and-error interactions with an environment [37, 144]. As shown in Figure 1, it has been studied for decades and has achieved success in a wide range of domains. Its applications span from game playing [11, 137, 151, 184] and chip design [105] to urban governance [51, 157, 205, 206], public health [52, 55], and smart transportation [158, 188], often exhibiting performance that surpasses human professionals. In recent years, with the rise of large language models (LLMs) marking a new wave of AI development [40, 41, 50, 54, 79–82, 100, 101, 116, 139, 159], RL has been widely adopted for the post-training of these models. It has demonstrated significant value and potential in crucial areas such as aligning LLMs with human preferences [9, 112] and enhancing their reasoning capabilities [171].

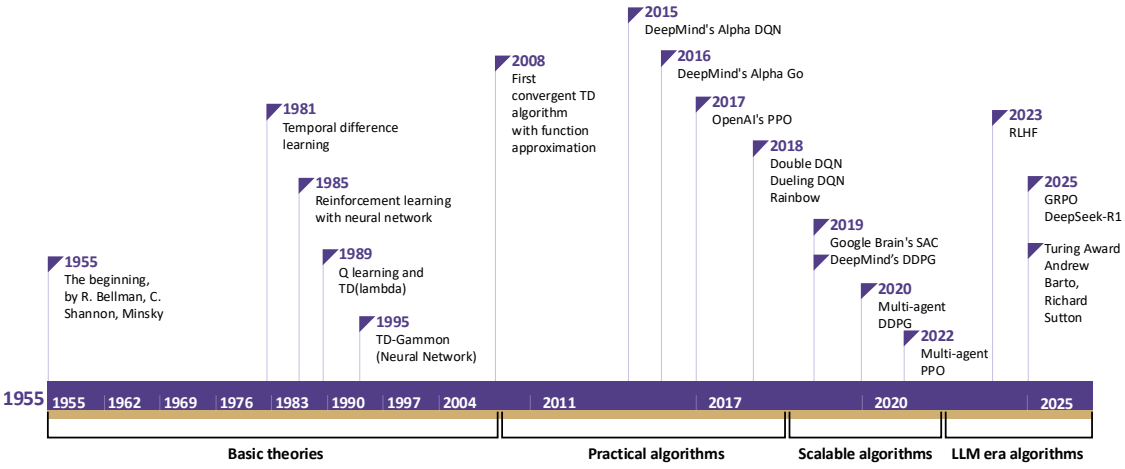

Fig. 1. Important milestones in the development of reinforcement learning

However, unlike conventional reinforcement learning, where actors are primarily based on lightweight neural networks [48, 60, 83, 106, 149, 163], applying RL in the era of large language models requires directly training and optimizing models with hundreds of billions of parameters that operate within the vast space of natural language [7, 8, 68, 94, 146, 172, 179, 180, 180]. This substantial increase in model parameter size and the complexity of the action space introduce significant challenges when adapting conventional RL algorithms to LLMs. These challenges manifest across multiple component of the RL training loop, including action exploration, trajectory collection, reward evaluation, and model update. Thus, a comprehensive review of these challenges and their corresponding solutions, alongside an exploration of future research directions, is both timely and essential.

This survey is positioned from the perspective of RL as a tool to serve LLMs—that is, we focus on how RL algorithms are adapted, improved, and applied to enhance the post-training, alignment, and reasoning capabilities of large language models. We do not aim to cover the complementary direction of using LLMs to improve classical RL (e.g., LLM-based

planning or decision-making for traditional RL agents), which constitutes a separate research line with distinct objectives. As shown in Table 1, existing surveys have summarized the adoptions of reinforcement learning on large language models from different perspectives. Some have provided detailed introductions to the fundamentals of RL algorithms and LLM architectures [15, 143, 193], and summarized various methodological improvements to RL adapted for LLM training [141, 143], while others have provided a comprehensive landscape from algorithms to infrastructures [193]. Additionally, existing survey discussed from a broader viewpoint, focusing on "reinforced reasoning" with LLMs, namely self-refinement in reasoning [171]. A majority of these surveys have included extensive scenario-specific applications. However, none of these works has deeply analyzed from the perspective of basic elements in the RL training loop, such as action and reward, to systematically summarize the challenges faced in each corresponding stage when applying RL to train LLMs. Furthermore, it is crucial to distinguish our scope from another rich body of literature: surveys focusing on how the innate reasoning of LLMs can improve traditional RL systems (i.e., LLMs for RL) [16], our survey occupies the inverse position. We focus our scope on treating RL as the optimization method for the LLM itself (i.e., RL for LLMs).

Table 1. Comparison with existing surveys

Survey	Year	Main focus	Algorithm	Infrastructure	Application	Deficiency
[15]	2023	RL for generative AI	✓	✗	✓	Lacks specific focus on LLMs
[143]	2025	Inverse RL for LLMs	✓	✗	✗	Limited to inverse RL
[171]	2025	"Reinforced reasoning" of LLMs	✓	✗	✓	Lacks specific focus on RL algorithms
[141]	2025	RL for LLMs	✓	✗	✓	Lacks taxonomy to basic elements in RL
[193]	2025	RL for LLMs	✓	✓	✓	Lacks taxonomy to basic elements in RL

To address the deficiencies in existing surveys, we have conducted an extensive investigation of the current research landscape, providing a detailed summary and discussion. From the perspective of the reinforcement learning training loop, we systematically summarize the challenges that arise in each stage when RL is adapted from conventional scenarios to the training of large language models. In response to these challenges, we comprehensively review the algorithmic solutions proposed in existing works for each respective stage. We also present applications in several representative domains, analyzing how each application incorporates improvements to different components of the RL pipeline. Beyond the existing solutions and applications, we further discuss the limitations and side effects of applying RL to LLMs, as well as future directions and open problems. This aims to aid researchers in better understanding the evolving landscape and to foster innovative future research.

As shown in Figure 2, the remainder of this paper is organized as follows. Section 2 begins by introducing the background of conventional reinforcement learning methods. Section 3 then discusses the challenges that arise in each stage of the RL training loop when applying in the era of large language models. In Section 4, we review the methodologies developed to adapt RL to large language models, corresponding to the challenges previously identified. Section6 delves into the limitations and side effects associated with using RL for LLM training. Finally, Section 7 concludes the survey by exploring open problems and future research directions in this rapidly evolving field.

2 REINFORCEMENT LEARNING

Reinforcement learning (RL) is a machine learning paradigm where an agent learns to optimize its actions through exploration and trial-and-error interactions with an environment, guided by a feedback signal known as a reward.

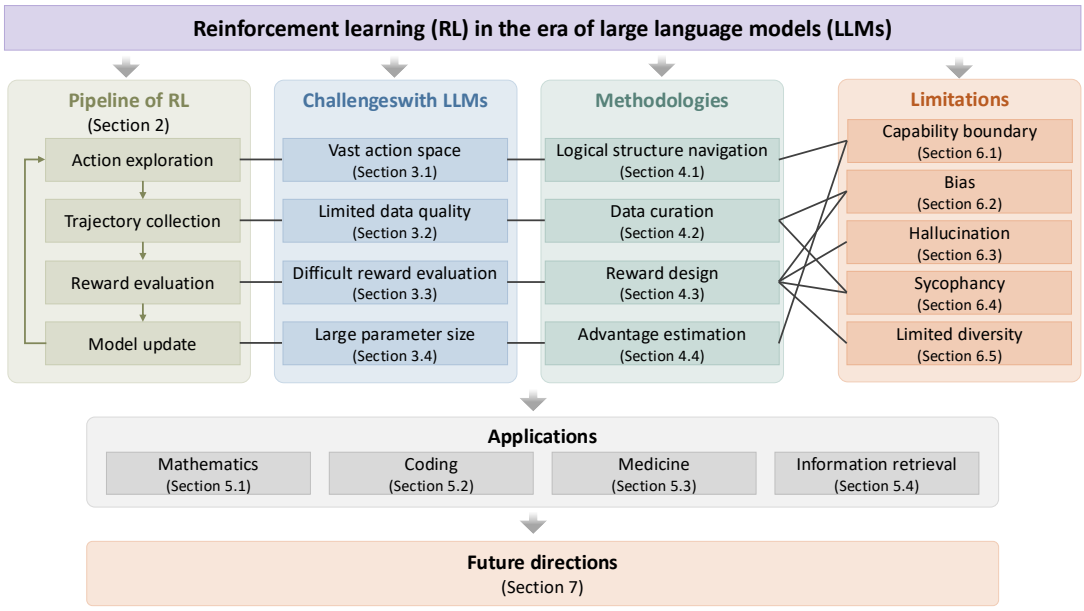

Fig. 2. Overall structure of this survey

In the past decades, it has long been studied [37, 144] and reached success a wide range of domains, such as game playing [11, 137, 151, 184], chip design [105], urban governance [51, 157, 205, 206], public health [52, 55], smart transportation [158, 188], etc., exhibiting performance surpassing human professionals.

The training process in reinforcement learning is fundamentally a closed-loop interaction between an agent and its environment. This process begins with the agent executing an action in a given state, guided by its current policy and often incorporating a strategy for exploration. In turn, the environment transitions to a new state and provides a scalar reward signal, which offers feedback on the performed action. The agent collects this sequence of states, actions, and rewards, forming a trajectory of experience. Subsequently, this trajectory is used to evaluate the effectiveness of the agent's decisions, and this evaluation drives the crucial step of updating the agent's internal model or policy. As shown in Figure 3, this entire cycle of action exploration, trajectory collection, reward evaluation, and model update is repeated iteratively, allowing the agent to progressively improve its strategy to maximize cumulative rewards over time.

Central to this iterative learning cycle is the model update mechanism, which varies significantly across different families of algorithms. For tasks with discrete action space, one primary approach is value-based methods, which focus on learning a value function that estimates the maximum expected future reward from each state or state-action pair. This lineage evolved from Monte Carlo methods to more sample-efficient Temporal Difference (TD) learning [147], with the update rule for Q-learning [60, 106, 149, 163] being a canonical example:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)], \tag{1}$$

where $Q(S_t, A_t)$ is the estimated value of taking action A_t in state S_t, R_{t+1} is the immediate reward received after the transition, α is the learning rate, and γ is the discount factor for future rewards. In contrast, for tasks with continuous

action space, policy-based methods directly parameterize and optimize the policy itself, without an explicit value function. These methods, such as REINFORCE [167], adjust the policy parameters θ by performing gradient ascent on an objective function, typically defined as the expected cumulative reward, with a characteristic update rule of

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(A_t|S_t) G_t, \tag{2}$$

where θ represents the policy parameters, $\pi_\theta(A_t|S_t)$ is the probability of selecting action A_t in state S_t, and G_t is the total discounted return from that time step forward. Bridging these two paradigms are actor-critic methods, which have become the foundation for many state-of-the-art results, including algorithms like Trust Region Policy Optimization (TRPO) [124] and Proximal Policy Optimization (PPO) [126]. These hybrid models feature an "actor" that controls the policy and a "critic" that learns a value function to evaluate the actor's actions. The critic's feedback, often in the form of an advantage estimate $A(S_t, A_t)$, provides a low-variance learning signal for the actor, leading to a more stable and efficient update:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(A_t|S_t) A(S_t, A_t). \tag{3}$$

3 CHALLENGES IN THE ERA OF LARGE LANGUAGE MODELS

Unlike conventional reinforcement learning training, where actors are primarily based on lightweight neural networks [48, 60, 83, 106, 149, 163], in the era of large language models, RL algorithms are required to directly train and optimize large language models (LLMs) with massive parameters. As illustrated in Figure 3, such substantial increase in parameter size presents numerous challenges across the conventional RL loop. Specifically, each component of this learning loop presents distinct challenges. First, effective action exploration must navigate a vast action space (Section 3.1). Second, the trajectory collection process is frequently constrained by limited data quality (Section 3.2), and the task of reward evaluation is itself inherently difficult (Section 3.3). Finally, with the obtained reward, the model update stage is confronted with the issue of a large parameter size (Section 3.4)

3.1 Vast Action Space

Large Language Models (LLMs) operate within a vast action space, from which the model must select its next output at each step, within a context length up to millions of tokens [180]. The immense scale of this space presents three fundamental challenges for text generation: inefficient search, poor long-range coherence, and a tendency for goal drift.

- **Combinatorial explosion and search inefficiency.** The number of potential output sequences grows exponentially with text length, making an exhaustive search for the optimal sequence computationally intractable. Consequently, a model's search for a high-quality output is inherently inefficient. It often produces text that is locally plausible but globally suboptimal or generic [153]. This issue underscores the importance of methods that can prune the action space to guide the search toward more promising possibilities.
- **Difficulty in maintaining long-Range coherence.** The model's freedom to choose any token can cause subtle diversions from the intended narrative path. Over many steps, these small deviations accumulate, leading to outputs with internal contradictions or a loss of the central theme [5, 185]. Effective long-form generation may therefore require a high-level structural scaffold to maintain logical and thematic unity throughout the text.
- **Goal drift and loss of focus.** This phenomenon occurs when the model gradually deviates from the user's primary intent. The model may follow statistically probable but task-irrelevant tangents simply because the action space contains many such options [4]. Without a persistent focus on the main objective, the model can

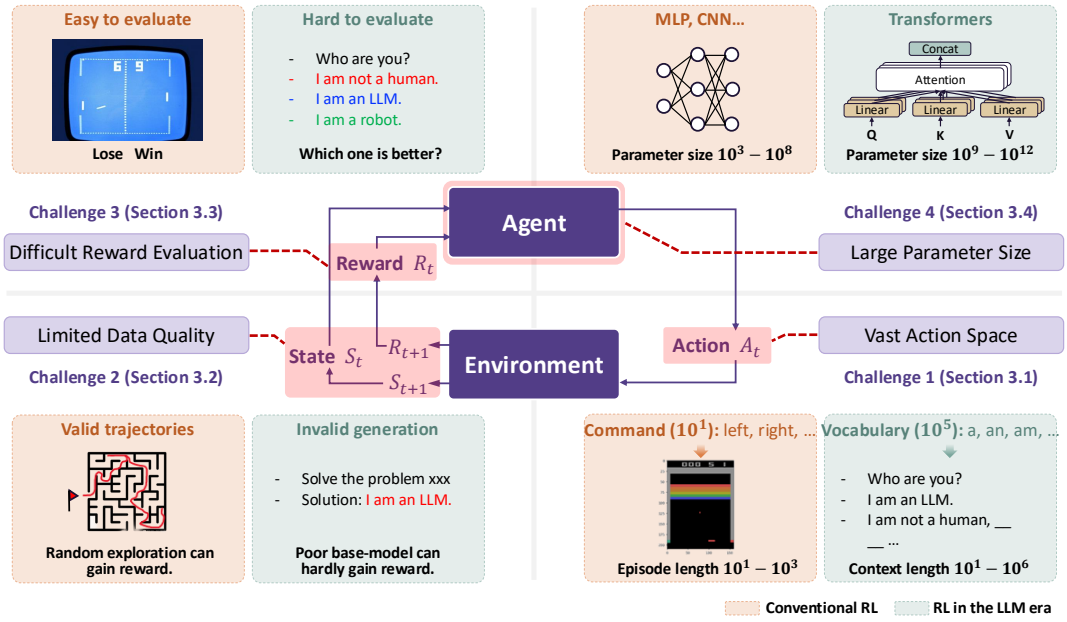

Fig. 3. Challenges for reinforcement learning in the era of large language models

produce incomplete responses that overdevelop minor details while neglecting core requirements. Decomposing the primary task into a clear sequence of sub-goals is one strategy to keep the model focused and systematic.

3.2 Limited Data Quality

In the era of large language models, reinforcement learning faces a major challenge: limited data quality. High-quality data is crucial for training RL agents, but obtaining such data in the LLM context remains difficult.

- **Limited data scope.** A significant issue is the inadequate data collection methods used in current fine-tuning practices [209]. Often, the data used for training LLMs comes from limited, pre-defined datasets that fail to capture the full diversity of real-world scenarios. This restricts the model's exposure to a wide variety of interactions, leading to overfitting and poor generalization across different tasks or domains [46].
- **Constraints of guiding tokens.** Another limitation is the constraints imposed by guiding tokens. Special tokens are frequently used to direct the model's behavior during fine-tuning. However, these tokens tend to narrow the scope of responses, often resulting in overly simplistic or repetitive outputs. This limits the diversity and richness of the training data, reducing its quality for reinforcement learning applications [112].
- **Deficiency in coherent reasoning.** Finally, there is a deficiency in coherent reasoning within the data generated for RL tasks. LLMs, especially when pre-trained for multi-step reasoning, often struggle to maintain logical consistency over extended sequences. [148]When relying on pre-trained base models for fine-tuning via online exploration, the generated sequence data are often fragmented or incomplete reasoning paths, preventing RL models from effectively learning complex, long-term decision-making processes [29].

3.3 Difficult Reward Evaluation

Accurately assessing the quality of Large Language Model (LLM) outputs to create a reliable reward signal presents a fundamental and multifaceted challenge. This difficulty stems from the inherent ambiguity of complex tasks, the scalability limitation of human feedback, the trade-off between outcome and process supervision, and the model's potential to exploit reward functions. Here we discuss key challenges as follows.

- **Ambiguity and subjectivity.** For many tasks, such as summarization or open-ended question answering, there is no single ground truth. This makes automatic evaluation, using metrics like BLEU or ROUGE, challenging as they often correlate poorly with human judgment for complex generation tasks [97]. This ambiguity necessitates costly and slow human evaluation to capture nuances like coherence, relevance, and factuality, which automated systems struggle to assess reliably.
- **Scalability of human feedback.** The standard Reinforcement Learning from Human Feedback (RLHF) paradigm relies on humans to generate preference data, which is used to train a reward model. While effective, this process is a significant bottleneck. The original proponents of RLHF noted that it is a costly and complex process to scale [112]. This reliance on human annotators is not only expensive but can also suffer from inconsistent judgments among individuals, introducing noise into the reward signal and limiting the scalability of the approach [10].
- **Dilemma of outcome and process evaluation.** Relying on the final answer for supervision (outcome-based evaluation) can be misleading. A model might arrive at the correct answer through flawed, irrelevant, or non-replicable reasoning steps, a phenomenon known as "false positives" in reasoning tasks [91]. Conversely, evaluating the intermediate reasoning steps (process-based supervision) is more granular and robust but presents its own problems. Human annotation of each step is significantly more expensive than outcome-based annotation, and creating automated process rewards can be complex and lead to unstable training if not carefully designed [91].
- **Reward hacking and misalignment.** LLMs are adept at finding loopholes to maximize a reward signal without achieving the intended goal—a behavior termed reward hacking or specification gaming. This is a classic challenge in reinforcement learning that is amplified in the complex, high-dimensional policy space of LLMs [2]. For example, a model rewarded for summarization might learn to produce generic, high-probability phrases that are grammatically correct but fail to capture the essence of the source text, thereby maximizing its reward without fulfilling the user's intent.

3.4 Large Parameter Size

Optimizing large language models using reward obtained in reinforcement learning presents a significant challenge due to the sheer scale of model parameters involved. Modern LLMs often contain billions [7, 8, 68, 172, 179, 180, 180] to hundreds or thousands of billions of parameters [94, 146], making them among the largest machine learning models ever developed.

- **Burden of value network.** A significant computational burden in applying RL to LLMs arises from the architectural demands of algorithms like Proximal Policy Optimization (PPO). These Actor-Critic frameworks necessitate a separate value network, often a full replica of the LLM, to run alongside the policy network [126, 210]. Maintaining and updating two such massive models in parallel effectively doubles both the GPU memory footprint and the computational load for each training step. This dual-network overhead represents a primary efficiency

bottleneck, driving research toward more streamlined approaches that reduce or eliminate the dependency on an explicit value critic.

- **Cost of advantage estimates.** The process of advantage estimation in RL introduces another layer of severe computational expense. To ensure stable policy updates, on-policy methods often need to repeatedly sample multiple candidate responses for each prompt [61]. This iterative cycle—generating numerous outputs, evaluating each with a reward model, and calculating advantage scores against a value baseline—creates a profound computational bottleneck. The prohibitive cost of this data generation and evaluation loop severely limits the scalability of traditional RL, spurring the development of more efficient alignment methods that simplify or bypass the need for explicit advantage estimation.

In summary, the four challenges identified above, namely vast action space, limited data quality, difficult reward evaluation, and large parameter size, are interconnected. In Section 4, we review the methodological solutions developed to address each of these challenges, organized along the same four dimensions of the RL training loop: logical structure navigation (addressing the vast action space), data curation (addressing limited data quality), reward design (addressing difficult reward evaluation), and advantage estimation (addressing large parameter size).

4 RL SOLUTIONS IN THE ERA OF LLMS

As shown in Figure 4, reinforcement learning algorithms have experienced decades of development and are nowadays capable of complex state and action spaces, as well as reaching smooth training process. To address the aforementioned challenges in the era of large language models, numerous solutions have been further developed for each stage of the reinforcement learning process. In action exploration, specific logical structures navigation are designed to effectively reduce the vast action space (Section 4.1). Furthermore, various data curation (Section 4.2) and reward design (Section 4.3) methodologies have been proposed to enhance data quality and strengthen reward evaluation, respectively. For the model update phase, improved advantage estimation methods have been introduced to mitigate the computational overhead caused by a large parameter size (Section 4.4).

4.1 Logical Structure Navigation

Table 2. Summary of logical structure navigation designs in training LLM with RL

Category	Paper	Logical Structure	Key Design	Domain
Internal Policy	[199]	Interaction Path	Masked Prompting	Recommendation
	[109]	Chain-of-Thought	Dual-Model Synergy	Math, General Reasoning
	[135]	Chain-of-Thought	Difficulty-Adaptive	General Reasoning
	[207]	Chain-of-Thought	RL Bootstrapping	Math, Coding
	[194]	Chain-of-Thought	Preference Optimization	General Reasoning
	[190]	Tree Search	Process-Reward Training	General Reasoning
	[153]	Tree Search	AlphaZero-like	General Reasoning, Planning
External Policy	[154]	Chain-of-Thought	Q-value Guidance	Math, Coding
	[53]	Dynamic Structure	Navigator Model	General Reasoning
	[86]	Tree Search	Learned Search Policy	General Reasoning, Planning

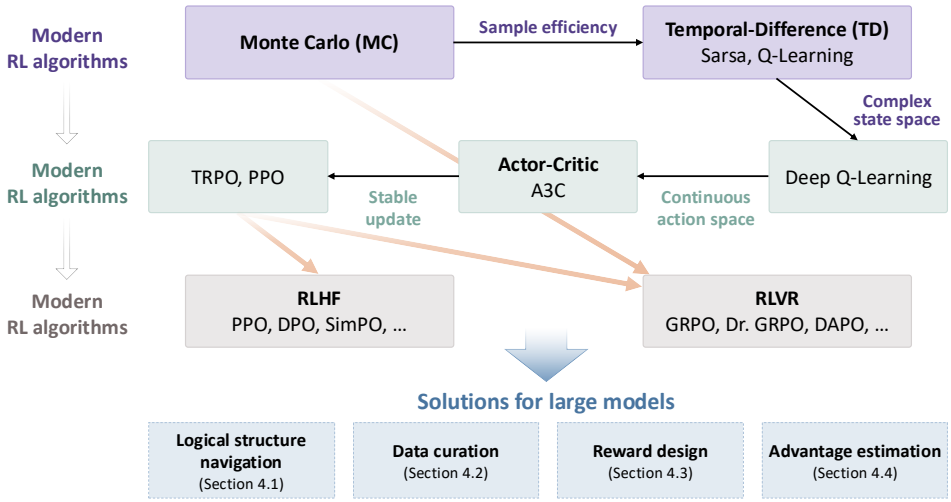

Fig. 4. Development of RL algorithms in the era of LLMs

To address the challenge of navigating the vast action space of LLMs (see Section 3.3), a prominent line of research focuses on imposing a logical structure on the generation process. This structure acts as a high-level plan, constraining the token-level search space and guiding the model towards more coherent and effective reasoning. We categorize these methods based on how the guiding policy is learned and applied: either by internalizing it directly into the LLM's parameters (internal policy) or by using it as an external controller during inference (external policy). Figure 5 and Table 2 provide a schematic illustration of this dichotomy, contrasting the direct fine-tuning of an LLM with the modular guidance of the external agent.

4.1.1 Internal Policy. The Internal Policy approach utilizes reinforcement learning (or RL-inspired methods) to directly fine-tune the LLM's parameters. The central goal is to enhance the model's intrinsic reasoning capabilities, effectively internalizing a policy that allows it to autonomously generate more structured and accurate reasoning paths without requiring complex external guidance during inference.

These methods often focus on improving the quality and efficiency of the model's generated thought processes. For instance, R2Rec adapts this to recommendation systems by using RL to refine an LLM's ability to generate reasoning chains over user-item interaction paths [199]. Other works optimize the reasoning process itself; Long⊗ Short employs multi-turn RL to train a model to synergistically generate both long and short thoughts for efficiency [109], while DAST trains models to adapt the length of their reasoning based on problem difficulty, mitigating overthinking [135]. To improve the quality of reasoning steps, BRiTE uses RL to bootstrap the generation of high-quality rationales, which are then used to fine-tune the base LLM [207]. Similarly, CPO leverages preference information from tree search explorations to fine-tune LLM, aligning its simpler chain-of-thought generation with more optimal reasoning paths [194].

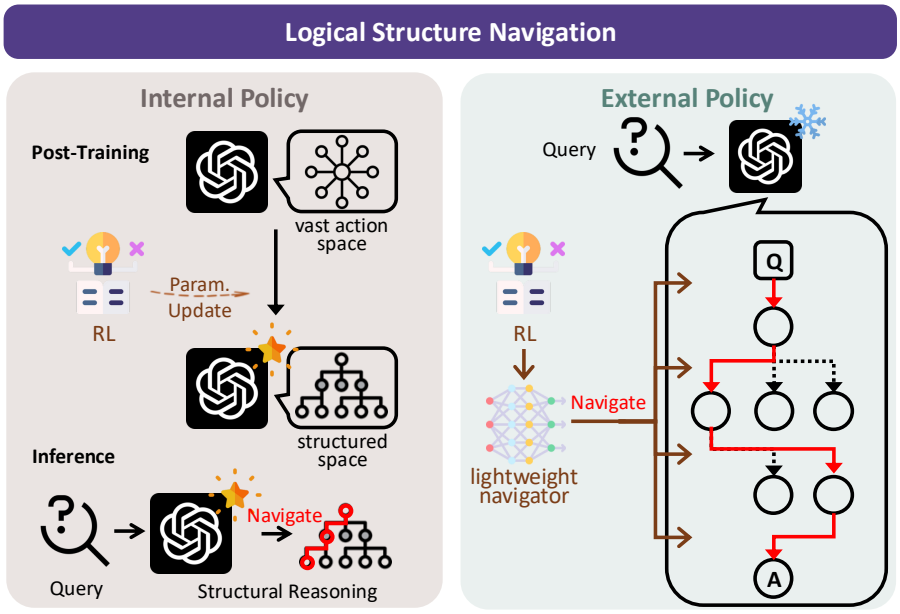

Fig. 5. Illustration of different logical structure navigation designs. Internal-policy methods (left) directly fine-tune the LLM's parameters to internalize reasoning structures, while external-policy methods (right) train a separate controller that guides a frozen LLM at inference time.

Extending this concept to more complex structures, some methods use tree search to discover superior reasoning traces for self-training. ReST-MCTS* integrates process rewards with tree search to collect high-quality data to train both a policy and a reward model [190], and the AlphaZero-like framework of TS-LLM uses tree search to guide both inference and training, allowing iterative improvement of the LLM itself [153].

In essence, these methods invest computational resources during the training phase to create a more powerful and self-sufficient LLM that can perform complex reasoning more effectively on its own. From a design-trade-off perspective, internal-policy methods shift the cost from inference to training: once fine-tuned, the model reasons autonomously without additional search overhead, but the training process itself is expensive and the resulting policy is tightly coupled to the base model's parameters, making it difficult to transfer improvements across different architectures. Moreover, because the internalized policy modifies the model's weights, any errors or biases learned during RL fine-tuning become deeply embedded and are non-trivial to correct post hoc.

4.1.2 External Policy. In contrast, the External Policy approach trains a separate agent or policy model that operates externally to the LLM. This external controller is designed to guide the generation process of a pre-trained, and often frozen, LLM during inference time. Here, the LLM functions as a generator of potential reasoning steps, while the RL-trained policy is responsible for making high-level decisions, such as selecting the most promising path in a search space. The primary advantage of these methods is their flexibility and modularity, as they can enhance the reasoning of various off-the-shelf LLMs without the need for costly retraining of the base models themselves.

For example, Q* learns a plug-and-play Q-value model to act as a heuristic, guiding the LLM's step-by-step decoding for multi-step reasoning tasks without fine-tuning the base model [154]. RL of thoughts (RLoT) trains a lightweight RL-navigator model to dynamically select and combine basic logic blocks, adaptively building a task-specific reasoning structure for the LLM to follow at inference time [53]. Similarly, Policy-Guided Tree Search (PGTS) employs a learned policy to make dynamic decisions within a tree search—such as when to expand, backtrack, or terminate—thereby steering the LLM's exploration of reasoning paths more efficiently [86]. The key trade-off here is flexibility versus inference-time cost: external-policy approaches can be applied to any frozen LLM and are thus model-agnostic, but they introduce additional computational overhead at inference time due to the need for search or policy consultation. Furthermore, because the base LLM remains unchanged, the quality of external-policy guidance is fundamentally bounded by the base model's generation ability—an external policy cannot elicit reasoning patterns that the underlying LLM is incapable of producing.

4.1.3 Discussion. The two paradigms represent fundamentally different design philosophies with complementary strengths and weaknesses. Internal policy methods embed reasoning structure directly into the model's weights, yielding fast inference (no external search) and tight integration between reasoning and generation, but at the cost of expensive training, limited transferability across models, and the risk of permanently embedding biases. External policy methods decouple reasoning control from generation, offering modularity, model-agnostic applicability, and the ability to dynamically adapt at inference time, but they incur higher per-query costs and are constrained by the base model's capability ceiling. In practice, the choice depends on the deployment context: internal policies are preferred when inference efficiency and self-sufficiency are critical (e.g., on-device deployment), while external policies are advantageous when flexibility, rapid iteration, or compatibility with multiple LLMs are required (e.g., research prototyping or multi-model pipelines). A promising but largely unexplored direction is to combine both: using internal-policy fine-tuning to raise the base model's baseline capability, then applying lightweight external guidance to further refine reasoning at inference time, potentially achieving the benefits of both paradigms.

4.2 Data Curation

In the era of Large Language Models (LLMs), the challenge of limited data quality remains a critical issue for reinforcement learning. In addressing this challenge, recent studies have proposed several strategies to enhance data quality through better data sampling, more effective use of special instruction tokens, and the construction of more coherent and structured data format. These strategies aim to alleviate the issues of limited data diversity, inadequate guidance during model exploration, and fragmented reasoning sequences, which are central concerns in the quest for higher-quality training data. An overview of these strategies is illustrated in Figure 6, while their construction process is detailed in Table 3.

4.2.1 Data sampling strategy. Data sampling strategies play a crucial role in enhancing data quality, as they determine not only the diversity and representativeness of training examples but also the reliability and usefulness of the data for reinforcement learning. The data sampling strategy has seen significant advancements, with several works highlighting methods to ensure high-quality, diverse training datasets. For instance, WebThinker [85] introduces a self-sampling approach where diverse reasoning trajectories are generated from multiple complex reasoning datasets. They filter and construct preference datasets by defining rules for preference pairs, such as overall correctness, tool efficiency, and thinking simplicity. Similarly, Wang et al. [161] propose an efficient data filtering strategy using fastText classifiers to select high-quality data, while employing multi-source seed data selection for better data quality. Cheng et al. [27]

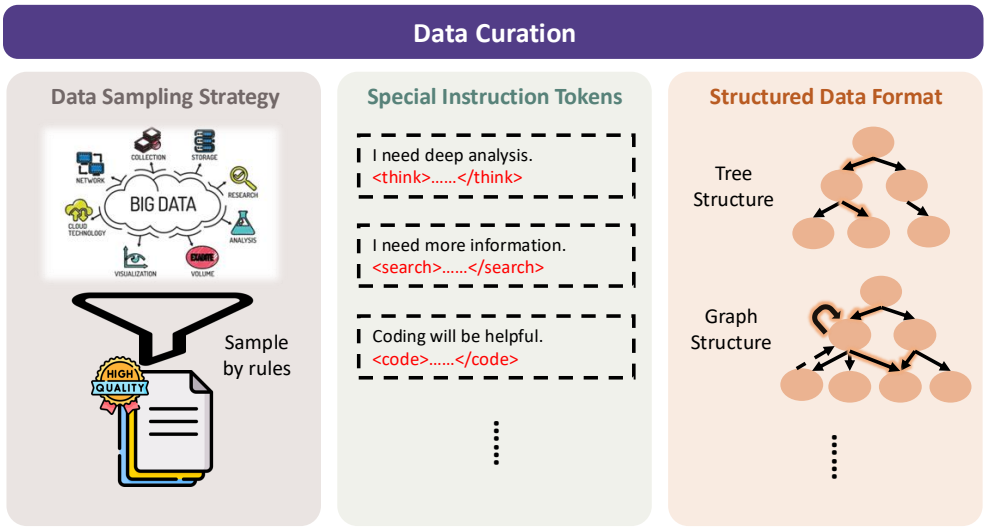

Fig. 6. Illustration of different data curation methods. Data sampling strategies (left) focus on selecting and filtering high-quality training examples; special instruction tokens (center) provide explicit structural signals to guide model outputs; and structured data formats (right) organize reasoning processes into consistent and interpretable representations.

generate training data through self-adversarial language games, and sample effective data by combining the designed reward mechanism and advantage value estimation method. Setlur et al. [128] take a different approach by generating both correct and incorrect reasoning paths for math tasks, focusing on refining the model's ability to recognize errors in reasoning. Other studies try to use the sample with the largest accuracy variance during model testing as the training sample [177], or introduce the better-performing off-policy trajectory data to guide the optimization of the target policy [162]. Huang et al. [66] propose the "gradient filtering" strategy, which avoids unnecessary calculations on these data that contribute less to the gradient update by discarding samples with smaller dominant estimates. Moreover, Zheng et al. [204] use data cleaning and contamination detection to prevent reliance on pre-trained knowledge, excluding sensitive or harmful questions and identifying memory-based responses through random sampling. And Weng et al. [166] construct academic datasets by collecting papers and reviews from top conferences, enhancing data quality through a structured review process.

Beyond sampling and filtering, two further aspects of data curation deserve attention: data synthesis and data difficulty. Data synthesis refers to the automated generation of novel training examples, which has become essential for scaling RL data beyond the limits of human-curated corpora. Approaches such as Evol-Instruct [170] iteratively evolve existing problems into more complex variants, enabling continuous expansion of the training distribution without manual annotation. Similarly, WizardMath [102] leverages Evol-Instruct to construct diverse mathematical problems, while SwS [90] enables models to identify their own weaknesses and synthesize targeted problems. Data difficulty, on the other hand, concerns calibrating the challenge level of training data to match the model's current capability. Training on uniformly easy data wastes capacity, while uniformly hard data can destabilize learning. DeepSeekMath [131]

Table 3. Summary of data curation methods in training LLM with RL

Category	Paper	Key Design	Domain
Data Sampling Strategy	[85]	Rules for preference pairs	General Reasoning, Research
	[161]	FastText classifiers to select data	General Reasoning
	[27]	Generate data through self-adversarial	Language Games
	[128]	Both correct and incorrect reasoning traces	Math
	[177]	Use only one training data	Math, General Reasoning
	[162]	Introduce off-policy data	Math
	[66]	Propose "gradient filtering" strategy	Math, Coding
	[204]	Data cleaning and contamination detection	General Reasoning
	[166]	Collect papers from top conferences	Research
Special Instruction Tokens	[133]	<continue reasoning> and <reflect>	Math, General Reasoning
	[72]	<think>, <search> and <answer>	General Reasoning
	[44]	<think>, <search> and <answer>	Math, General Reasoning
	[129]	<search_query> and <math_exp>	Math, General Reasoning
Structured Data Format	[169]	Sample intermediate states	Math, General Reasoning
	[65]	Annotated reasoning	LLM Judge
	[19]	Entailment trees	General Reasoning
	[34]	Code-enhanced reasoning paths	Math
	[194]	Highlights preferred paths	General Reasoning
	[63]	Structure search queries	General Reasoning

demonstrates the value of difficulty-aware sampling, which prioritizes problems at the frontier of the model's ability to maximize learning signal. Furthermore, the role of negative samples, namely incorrect or suboptimal reasoning traces, is increasingly recognized as crucial. Rather than discarding failed trajectories, they can serve as contrastive signals for preference-based methods like Direct Preference Optimization (DPO), where pairs of chosen and rejected responses are essential. Setlur et al. [128] explicitly leverage both correct and incorrect reasoning paths, and similar principles underlie the construction of preference datasets in RLHF pipelines. Careful curation of negative samples, which ensures they are informative rather than trivially wrong, is key to effective contrastive learning.

4.2.2 Special Instruction Tokens. Special instruction tokens help improve data quality by providing explicit structural signals that guide model outputs, thereby reducing ambiguity, enhancing consistency, and making the collected reasoning data easier to parse and evaluate. Regarding the use of special instruction tokens, several works employ these tokens to direct model behavior during reasoning tasks. Satori [133] utilizes meta-action tokens like "continue reasoning" and "reflect" to guide models through reasoning processes, enhancing the model's ability to self-reflect and explore different solutions. In Search-R1 [72], they incorporate special tokens such as <think>, <search>, and <answer> to structure the model's output, ensuring that the reasoning process and external search tool utilization are clearly delineated. Similarly, these three special tokens also appear in SEM [129]. But they are sequentially arranged into a structured format to ensure robust parsing and evaluation during reward computation. Goldie et al. [44] utilize

special tokens to distinguish between different actions in multi-step reasoning, such as <search_query> for searches and <math_exp> for calculations, which helps structure the output, facilitates parsing, and supports iterative multi-step reasoning.

4.2.3 Structured Data Format. Structured data formats improve data quality by organizing reasoning processes into consistent and interpretable representations, which reduces noise, enhances coherence, and facilitates more effective learning for reinforcement learning tasks. The construction of structured data format has also been a key focus. Xi et al. [169] structure data by sampling intermediate states from the correct reasoning path as the starting point and construct a reverse course with gradually increasing difficulty. In Think-J [65], data is annotated with reasoning traces using Deepseek-R1, and adopts "Tracing Clipping" to eliminate the lengthy reasoning process and only retain the key summary part. SEER [19] constructs data by pairing facts with hypotheses, guiding the model to generate structured reasoning paths such as entailment trees. Feng et al. [34] replace manual reasoning steps with code-enhanced reasoning paths by automatically transforming existing reasoning traces into code-augmented formats, providing a structured and executable approach to tool usage. Zhang et al. [194] leverage Tree-of-Thought [182] processes to create multiple reasoning branches, providing structured data that highlights preferred paths. Hsu et al. [63] structure search queries and multi-hop retrieval by using diverse few-shot prompts, ensuring that each query builds on previous contexts for consistent, organized reasoning.

4.2.4 Discussion. The three categories of data curation—sampling strategies, special instruction tokens, and structured data formats—address different but complementary aspects of the data quality challenge. Sampling strategies primarily govern what data enters the training pipeline (selection and filtering), making them crucial for diversity and representativeness but limited in shaping the internal structure of individual data points. Special instruction tokens address how the model's output space is constrained during generation, providing cheap and effective control signals but requiring careful design to avoid over-constraining the model's exploration. Structured data formats go furthest in imposing interpretability and coherence on reasoning traces, enabling precise credit assignment and process-level supervision, but they demand more expensive annotation or synthesis procedures. A key conceptual insight is that these categories can be composed: for instance, difficulty-aware sampling (a data sampling strategy) can be combined with structured rationale formats to curate datasets that are both diverse in coverage and coherent in internal structure. However, a fundamental tension exists between data diversity and data controllability—aggressive filtering or highly prescriptive formats improve quality metrics but may inadvertently narrow the training distribution, leading to reduced generalization. Future work on data curation should therefore aim to balance these competing objectives, potentially through adaptive curation pipelines that dynamically adjust filtering strictness and structural constraints based on the model's evolving performance.

4.3 Reward Design

Before diving into specific reward mechanisms, it is essential to clarify the conceptual boundary between Data Curation (Section 4.2) and Reward Design, as the two can occasionally overlap in practice. Fundamentally, data construction pertains to the preparation phase: it determines what materials, such as specific prompts, reasoning traces, and offline preference pairs, are fed into the RL pipeline. In contrast, reward design constitutes the feedback phase: it defines how the model's generated outputs are evaluated, scored, and transformed into a guiding learning objective during active training. While certain paradigms with very simple reward design blur this line, most notably Direct Preference Optimization (DPO), where curated preference pairs simultaneously serve as training data and implicitly define the

reward function, the structural distinction remains clear: data construction shapes the observational input, whereas reward design engineers the evaluative feedback. In this section, we focus on discussing more sophisticated reward mechanisms beyond simple preference pairs.

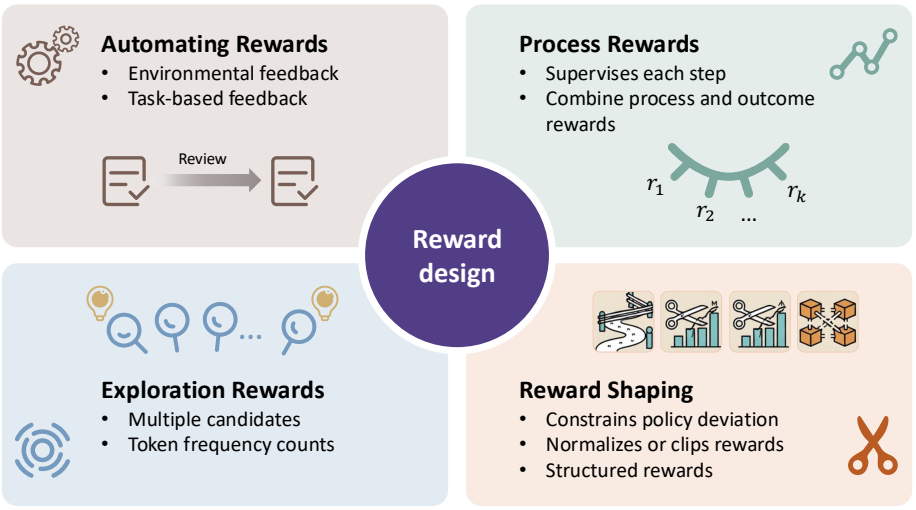

Fig. 7. Illustration of different reward designs. Automating rewards (left) replace human feedback with environmental or AI-generated signals; process rewards (center-left) evaluate intermediate reasoning steps rather than just final outcomes; exploration rewards (center-right) incentivize novel reasoning paths to prevent convergence to safe but uninformative outputs; and Reward shaping (right) transforms raw reward signals into stable training objectives via KL penalties, normalization, and structured design.

Table 4. Summary of reward designs in training LLM with RL

Category	Method	Paper	Key Design	Domain
Automating Rewards	PFPO	[71]	Pseudo-feedback without human labels	Reasoning
	RLEF	[43]	Execution results (e.g., test pass/fail) as feedback	Coding
	RLAIF	[10]	LLM-generated preferences as feedback	General reasoning
Process Rewards	PRM	[91]	Stepwise supervision to improve faithfulness	Math, General reasoning
	ReST-MCTS*	[191]	Combine process rewards with tree search	General reasoning
Exploration Rewards	Best-of-N	[112]	Selects from N candidates to promote diversity	General reasoning
	COPO	[6]	Rewards novel generations using token frequency counts	General reasoning
Reward Shaping	KL Penalty	[112, 142]	Constrain policy deviation to stabilize training	General reasoning
	Normalization	[121]	Normalize or clips rewards to avoid gradient spikes	General reasoning
	Structured Reward	[42]	Design structured rewards tailored for reasoning tasks	General reasoning

The design of the reward function is a critical component that steers the behavior of an LLM during reinforcement learning. Traditional reliance on human feedback has proven to be a bottleneck, leading to the development of novel reward strategies. These strategies focus on automating feedback, improving the granularity of reward signals, encouraging diversity, and enhancing the stability of the training process itself. Figure 7 and Table 4 provide a schematic illustration of these reward design categories and a summary of representative methods.

4.3.1 Automating Rewards with Environmental and Task-Based Feedback. To overcome the scalability limitations of human feedback, researchers are increasingly turning to automated reward sources. One powerful approach is using environmental feedback, where an LLM is rewarded for successfully interacting with external tools. For example, a model can be rewarded if the code it generates compiles and passes unit tests [43, 87], or if it successfully uses a web browser or API to complete a task [108, 123]. Recent advances like RLEF explicitly ground code LLMs in execution signals using PPO-based RL to optimize program correctness [43]. Another scalable direction is Reinforcement Learning from AI Feedback (RLAIF), where preference judgments are generated by a strong LLM rather than humans. Constitutional AI [10] is a prominent instance of this paradigm. More recent work explores learning from pseudo feedback via preference optimization without human-in-the-loop, offering scalable and data-efficient reasoning supervision [71].

4.3.2 Outcome Reward and Process Reward. The granularity of the reward signal has a profound impact on learning. Outcome-supervised reward models (OSRMs) provide a scalar reward based on the correctness or quality of the final output [142]. While simple and cheap to label, they may reward flawed reasoning if the final answer happens to be correct [91]. In contrast, process-supervised reward models (PSRMs) evaluate intermediate reasoning steps, promoting more reliable chain-of-thought generation [29, 148]. This dense supervision has been shown to improve sample efficiency and robustness, particularly in mathematical or logic-heavy domains [91]. ReST-MCTS* combines process-based supervision with Monte Carlo Tree Search to iteratively improve reasoning via self-training [191]. However, these methods typically require more expensive human annotations and step-level validation.

4.3.3 Exploration Rewards. A fundamental challenge in RL is the exploration–exploitation trade-off. To prevent LLMs from converging to safe but uninformative outputs, exploration-based strategies are employed. One widely used mechanism is best-of-N sampling combined with reward model selection, which implicitly expands the explored output space [112]. Count-based exploration has also been introduced into LLM preference alignment, as in COPO and its online extensions, where reward signals are modulated by generation frequency to promote novel reasoning paths [6]. Moreover, KL divergence regularization is a standard technique to ensure exploration does not result in degenerate behavior. This term penalizes deviation from a pre-trained supervised model and is commonly used in RLHF pipelines [211].

4.3.4 Reward Shaping for Training Stability. The raw signal from a reward model can be noisy, sparse, or poorly scaled, leading to unstable training—a well-documented challenge in applying RL to language tasks [121]. Reward shaping aims to transform this signal into a more stable and informative learning objective. The most critical shaping technique is the inclusion of a Kullback-Leibler (KL) divergence penalty, which discourages the policy from straying too far from the supervised fine-tuned (SFT) model [112, 142, 211]. This has proven to be a key regularizer for aligning performance and preserving linguistic fluency. Beyond KL control, additional strategies such as reward normalization (e.g., zero mean, unit variance), clipping, and adaptive scaling are often used to avoid destructive gradients. Recent work also explores systematic reward function design for reasoning-specific behaviors, showing that the structure of the reward itself (e.g., step-sensitive, solution-focused) heavily impacts learning outcomes [42].

4.3.5 Discussion. The four reward design categories form a layered architecture for controlling RL training. Automating rewards addresses the source of the feedback signal, determining whether it comes from human annotation, environmental execution, or AI-generated judgment; the choice here primarily affects scalability and cost. Process versus outcome rewards govern the granularity of supervision, with outcome rewards being cheaper but vulnerable to rewarding flawed reasoning ("false positives"), while process rewards provide denser, more reliable signals at the expense of annotation cost and potential training instability. Exploration rewards operate at the behavioral level, counteracting the natural conservatism of RLHF-trained models that gravitate toward high-reward but uninformative outputs. Finally, reward shaping functions as a stability layer, transforming raw reward signals into well-scaled training objectives. These layers interact: for example, process rewards combined with KL-based reward shaping can mitigate the instability that pure process supervision sometimes introduces, while exploration rewards are most effective when the base reward signal is already well-normalized. A critical open question is how to jointly optimize across these layers—for instance, automatically switching between outcome and process supervision based on the model's confidence, or dynamically adjusting exploration bonuses as the policy matures.

4.4 Advantage Estimation

Table 5. Summary of advantage estimation methods in training LLM with RL

Category	Algorithm	Paper	Value Network	Advantage Estimator	Core Design
Online	GRPO	[131]	No	Group-relative advantage	Group-based advantage estimation
	Dr.GRPO	[99]	No	Monte Carlo advantage	Unbiased variant of GRPO
	RLOO	[1]	No	Group-based REINFORCE	Multiple samples per query to estimate advantages
	DAPO	[186]	No	Distribution aware loss	Adapts policy updates based on reward distribution
	GiGPO	[35]	No	Hierarchical group advantage	Hierarchical advantage estimation episode-level + step-level
	VinePPO	[75]	No	Mote Carlo advantage	Explore alternative future paths from intermediate generation
	CPPO	[92]	No	Modified GAE	Uses REINFORCE++ to reduce number of required model generations
Offline	OREO	[156]	Yes	Step-level Q-values	Combines offline data with online exploration to guide policy update
Hybrid	A*-PO	[12]	No	Optimal advantage regression	Offline sampling for V* estimation On-policy updates using least-squares loss

As shown in Table 5, the design and implementation of effective advantage estimation strategies have emerged as a central theme in recent advances in reinforcement learning (RL) methods for training large language models (LLMs). Traditional actor-critic frameworks, such as Proximal Policy Optimization (PPO) [126], rely on value networks to estimate state values and compute advantage functions via bootstrapping mechanisms like Generalized Advantage Estimation (GAE) [125]. While these approaches are widely used, they often suffer from high computational cost, instability due to inaccurate value estimation, and scalability issues when applied to large-scale models. In response, several recent works have proposed alternative formulations that either improve the efficiency of critic-based advantage

estimation or eliminate the need for critics altogether by leveraging group-based sampling and Monte Carlo return estimation.

Among the critic-free approaches, Group Relative Policy Optimization (GRPO) [131] and its improved variant Dr. GRPO [99] stand out by computing advantages based on relative performance within multiple responses generated per prompt. These methods avoid value function approximation entirely and instead use simple mean-centering across sampled outputs to derive policy gradients. Dr. GRPO [99] further improves upon GRPO by removing problematic normalizations involving output length and standard deviation, which were found to introduce biases that degrade performance, especially in long-horizon reasoning tasks. Similarly, RLOO [1] employs a leave-one-out estimator inspired by REINFORCE, using group comparisons to construct advantages without any auxiliary network. These approaches offer strong stability, reduced memory usage, and competitive performance while being significantly simpler than traditional RL pipelines.

For tasks requiring more granular credit assignment—such as multi-step mathematical reasoning—methods like VinePPO [75] and GiGPO [35] introduce fine-grained advantage estimators that enable step-level feedback. VinePPO [75] replaces the value network with Monte Carlo estimates derived from intermediate rollouts, allowing precise attribution of reward signals to individual reasoning steps. This approach enables the model to explore alternative future paths from any given state, improving both interpretability and performance on complex reasoning benchmarks. GiGPO [35] builds on this idea by introducing a hierarchical advantage estimation framework: it computes both episode-level and step-level advantages, combining global trajectory evaluation with local decision feedback. This dual-level structure supports long-horizon planning while maintaining the benefits of group-based advantage estimation.

In addition to fully online methods, some algorithms incorporate offline data into the advantage estimation process to enhance sample efficiency. A*-PO [12], for instance, uses offline sampling to approximate the optimal value function, enabling efficient on-policy updates without a learned critic. OREO [156] leverages reward estimation from offline trajectories to guide exploration and policy improvement through tree-based search. These hybrid approaches combine the strengths of offline pre-training with dynamic online learning, offering robustness and generalization benefits.

Finally, DAPO [186] and CPPO [92] revisit traditional PPO-based advantage estimation and propose modifications tailored to the characteristics of LLMs. DAPO adapts the loss function based on the distribution of observed rewards, improving robustness across diverse tasks. CPPO introduces a dynamic pruning mechanism that filters low-quality completions during training, reducing computational overhead while preserving or even enhancing reasoning capabilities.

Collectively, these developments reflect a broader shift in advantage estimation design for LLM reinforcement learning—from reliance on value networks toward more scalable, interpretable, and fine-grained estimation techniques. Whether through group-based sampling, hierarchical decomposition, or offline-informed guidance, modern methods aim to balance efficiency, accuracy, and practicality in the challenging domain of large language model post-training.

4.4.1 Discussion. The landscape of advantage estimation can be understood along two axes: (1) whether a value network is required (critic-based vs. critic-free), and (2) the granularity of credit assignment (trajectory-level vs. step-level). Traditional PPO with GAE occupies the critic-based, trajectory-level quadrant, offering stable but expensive training. GRPO and RLOO move to the critic-free, trajectory-level quadrant, dramatically reducing memory cost but sacrificing step-level precision. VinePPO and GiGPO advance to critic-free, step-level estimation, achieving fine-grained credit assignment without a value network, though at the cost of additional per-step rollouts. Hybrid methods like OREO and A*-PO reintroduce learned value functions but in an offline or constrained manner, seeking to combine the sample efficiency of critic-based methods with the scalability of critic-free approaches. The key insight is that there is no

universally optimal estimator: the best choice depends on the task's reward structure, the available computational budget, and the desired level of credit assignment precision. For tasks with verifiable outcomes (e.g., math, coding), simple group-relative advantages often suffice; for tasks requiring nuanced process evaluation (e.g., medical reasoning), step-level or hybrid estimators become essential.

To provide readers with a quick-reference overview of the major RL approaches discussed in this section, Table 6 summarizes their core ideas, requirements, and typical use cases.

Table 6. Quick-reference overview of major RL and alignment approaches for LLMs

Approach	Core Idea	Reward	Memory	Sampling Cost	Typical Use Cases
RLHF (PPO)	Optimize policy via advantage against a learned value network	Neural RM	Heavy	Low	General alignment, chat models
DPO	Directly optimize policy from offline preference pairs	Implicit	Light (no Critic)	Low	Alignment w/o online sampling
GRPO	Group-relative advantage estimation; eliminates value network	Rule-based or Neural RM	Light (no Critic)	High (multi-sample)	Reasoning, math, coding
RLOO	Leave-one-out REINFORCE baseline; eliminates value network	Rule-based or Neural RM	Light (no Critic)	High (multi-sample)	Stable alignment
DAPO	Systematic stability improvement of GRPO	Rule-based or Neural RM	Light (no Critic)	High (multi-sample)	Stable training on reasoning, math, coding

5 APPLICATIONS

Building upon the methodologies devised to address the challenges of applying reinforcement learning to large language models, this section elucidates the practical implementation of these techniques. In particular, we analyze how each application incorporates improvements from the four solution categories introduced in Section 4: logical structure navigation (Section 4.1), data curation (Section 4.2), reward design (Section 4.3), and advantage estimation (Section 4.4). The discussion spans four representative domains, chosen for their distinct and demanding requirements that showcase the full spectrum of RL's utility. First, mathematics (Section 5.1) and coding (Section 5.2), as foundational pillars of the modern natural sciences, are included for their uncompromising demand for rigorous logical correctness. Second, medicine (Section 5.3) represents a high-stakes field where factual accuracy is paramount, as errors can directly impact safety and clinical outcomes. Third, information retrieval (Section 5.4) is selected to address the challenge of efficiency, as it necessitates sifting through vast quantities of information to retrieve relevant evidence. Collectively, as shown in Figure 8 and Table 7, the applications within these domains present multifaceted challenges, making them ideal exemplars to demonstrate the efficacy and intricate design of the various RL methodologies developed in the era of large language models.

5.1 Math

In mathematical reasoning tasks, RL with LLMs is primarily aimed at enabling models to carry out reliable and interpretable multi-step problem solving. Unlike general natural language tasks, mathematical reasoning requires strict logical consistency, the ability to handle intermediate steps, and robustness against cascading errors. The tasks span a broad spectrum of difficulty, including elementary arithmetic, algebraic manipulation, and competition-level problems,

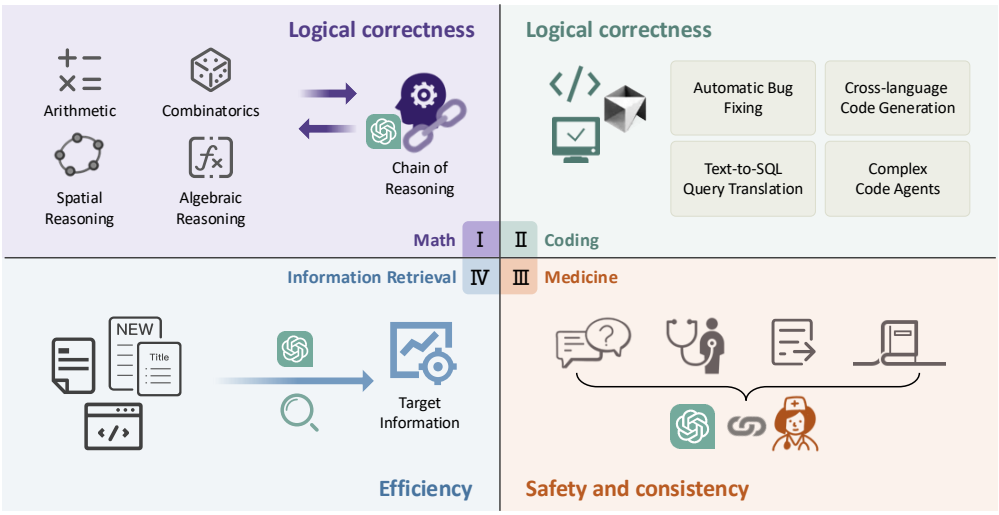

Fig. 8. Representative application scenarios of LLMs with RL

as reflected in benchmarks such as GSM8K [29], MATH [59], and OlympiadBench [58]. These benchmarks evaluate not only final-answer accuracy but also the faithfulness and clarity of the reasoning process. Consequently, RL research in this domain emphasizes enhancing the correctness of intermediate steps, improving generalization across diverse problem types, and balancing data efficiency with training stability, ultimately seeking models that can both achieve high performance and provide verifiable reasoning chains.

In terms of technical routes, recent work can be viewed through four complementary lenses. From the side of **logical structure**, Math-Shepherd [160] illustrates an internal-policy design that verifies reasoning steps sequentially, while DRO [175] constrains policies with structured intermediate formats, and RePO [84] explores an external-policy setting where exploration is guided by replay. **Data curation** strategies likewise vary: DeepSeekMath [131] emphasizes difficulty-based sampling to ensure robustness, Computational Thinking [192] organizes rationales into structured formats to align with cognitive operations, and SwS [90] pursues weakness-driven synthesis to generate new problems without heavy manual curation. **Reward design** reflects similar diversity, with Math-Shepherd automating process-level signals through programmatic verification, Satori [133] integrating reasoning and actions into a unified process reward, and No Free Lunch [197] refining reward shaping to encourage coherent reasoning beyond final answers. Finally, **advantage estimation** plays a decisive role in optimization stability: DeepSeekMath leverages GRPO for group-wise process supervision, RePO adapts DPO-style updates to align exploration with advantage signals, and No Free Lunch employs RLOO to stabilize credit assignment under step-level feedback. Together, these directions highlight how structural design, data pipelines, reward mechanisms, and optimization choices complement one another in advancing the robustness and interpretability of RL for mathematical reasoning.

Table 7. The application scenarios of LLM reasoning based on RL

Task	Paper	Challenges			
		Vast action space	Limited data quality	Difficult reward evaluation	Large parameter size
		Solutions			
		Logical structure navigation	Data curation	Reward design	Advantage estimation
Mathematics	[203]	Internal policy	Special instruction tokens	Process Rewards	RLHF, PPO
	[192]	Internal policy	Structured data format	✗	DAPO
	[160]	Internal policy	Data sampling strategy	Automating rewards	GRPO
	[175]	Internal policy	Structured data format	Reward shaping	DPO
	[160]	Internal policy	Data sampling strategy	Automating rewards	PPO
	[197]	Internal policy	Structured data format	Reward shaping	RLOO
	[84]	External policy	Data sampling strategy	Exploration rewards	DPO
	[133]	Internal policy	Special instruction tokens	Process rewards	GRPO
	[90]	Internal policy	✗	Automating rewards	PPO
	[102]	External policy	Structured data format	Reward shaping	PPO
Coding	[145]	Internal policy	Data sampling strategy	Automating rewards	DPO
	[192]	Internal policy	Structured data format	✗	DAPO
	[114]	Internal policy	Special instruction tokens	Automating rewards	GRPO
	[77]	Internal policy	✗	Automating rewards	GRPO
	[73]	Internal policy	Structured data format	Process rewards	TA-PPO
	[127]	Internal policy	Data sampling strategy	Automating rewards	GRPO, DPO
	[62]	Internal policy	Data sampling strategy	Automating rewards	PPO
Medicine	[202]	✗	Data sampling strategy	Automating rewards	DPO
	[113]	External policy	Structured data format	Automating rewards	GRO
	[201]	✗	Data sampling strategy	Automating rewards	DPO
	[117]	Internal policy	Structured data format	Automating rewards	GRPO
	[21]	Internal policy	Data sampling strategy	Automating rewards	PPO
	[195]	Internal policy	Structured data format	Automating rewards	GRPO
	[173]	Internal policy	Structured data format	Automating rewards	DPO, iDPO
	[178]	Internal policy	Structured data format	✗	DPO
	[181]	✗	Data sampling strategy	Automating rewards	PPO
Information Retrieval	[63]	Internal Policy	Data sampling strategy	Automating Rewards	IPO
	[95]	Internal Policy	Data sampling strategy	Automating Rewards	GRPO
	[189]	Internal Policy	Structured data format	✗	PPO
	[136]	Internal Policy	Data sampling strategy	Automating Rewards	GRPO
	[118]	Internal Policy	Data sampling strategy	Automating Rewards	GRPO

Overall, RL research in the mathematical domain demonstrates complementary pathways: verification-driven methods improve the correctness of step-by-step reasoning; self-supervision and self-correction reduce reliance on manual annotations; replay and optimization approaches address the challenge of training stability; structured paradigms enhance interpretability and generalization; and data synthesis continuously expands the boundaries of reasoning, together forming a multi-dimensional and co-evolving research landscape.

5.2 Coding

In the coding domain, RL with LLMs focuses on enabling reliable reasoning for program synthesis and repair. Compared with natural language tasks, coding requires not only semantic fluency but also strict executability and logical consistency, since even small reasoning errors may lead to program failures. Representative scenarios include automatic bug fixing, cross-language code generation, text-to-SQL query translation, and complex code agents that combine reasoning with tool use. The central challenges lie in ensuring correctness through verifiable signals, achieving generalization across languages and paradigms, constructing sufficient high-quality supervision data, and maintaining training stability while handling the discrete and non-differentiable nature of code execution.

From a technical perspective, research progress can be interpreted along four complementary dimensions. In terms of **logical structure**, most systems adopt internal-policy designs to directly regulate reasoning-to-code steps: Math-Shepherd's counterpart in coding, Tang et al.'s program repair framework [145], constrains reasoning through sequential error correction, while ReVeal [73] enforces a generation–verification–iteration loop to stabilize policy learning. For **data curation**, methods differ in how they create effective supervision. Computational Thinking [192] organizes problem decomposition into structured data formats aligned with abstraction and modularity, whereas Seed-Coder [127] adopts a sampling-driven self-evolution pipeline, letting the model generate and filter training data to refine its own reasoning ability. Regarding **reward design**, execution feedback and automation are central: Kulkarni & Srikumar [77] directly use execution correctness in Text-to-SQL as verifiable rewards, Pennino et al. [114] employ automated signals with instruction-controlled transfer to low-resource languages, and ReVeal incorporates process-level rewards by scoring each iteration of generation and verification. Finally, in **advantage estimation**, different estimators ensure stable optimization: Tang et al. [145] apply DPO to balance repair accuracy with exploration; Pennino et al. use GRPO baselines to stabilize preference-driven optimization; ReVeal leverages TA-PPO to integrate process rewards into policy gradients; and TreeRL [62] demonstrates that classical PPO remains effective when combined with tree search to enhance exploration. Together these approaches reveal a coding-RL landscape where structural design, data strategies, reward mechanisms, and optimization estimators interact to improve the executability, robustness, and generalization of code reasoning.

Overall, RL reasoning research in the coding domain exhibits complementarity between optimization signals (execution feedback, reasoning transfer, data self-supervision) and structural mechanisms (tree search, iterative agents, computational thinking), collectively advancing the reliability and generalization of code generation and repair.

5.3 Medicine

In the medical domain, RL with LLMs seeks to enable models to perform reliable and clinically meaningful reasoning. Compared with mathematics or programming, medical tasks involve higher stakes, since errors may directly affect safety, trustworthiness, or clinical outcomes. The domain spans clinical question answering, diagnostic assistance, case-based reasoning, and biomedical literature interpretation. Beyond accuracy, systems must therefore emphasize verifiability of reasoning chains, consistency with expert practice, and the integration of safeguards to mitigate risks

such as hallucinated diagnoses or unsafe recommendations. The challenges include aligning reasoning with diverse medical sub-tasks, grounding responses in domain knowledge, and balancing interpretability with performance across complex, knowledge-dense scenarios.

From a technical perspective, medical-RL studies can also be organized along four complementary axes. In terms of **logical structure**, Med-U1 [195] exemplifies an internal-policy framework that unifies multiple medical tasks—diagnosis, question answering, and retrieval—into a single reasoning policy, while Zhongjing [181] illustrates expert-in-the-loop reinforcement where external feedback guides iterative refinement. For **data curation**, MedAgentGym [173] introduces structured execution environments that transform reasoning into code-like steps, enabling verifiable supervision, whereas earlier work such as DeepSeek-Med [202] employs sampling strategies to curate balanced datasets across tasks. Regarding **reward design**, process automation and domain alignment are central: MedAgentGym leverages automated execution correctness as RL feedback, Yan et al.'s retrieval-augmented system [178] aligns rewards with preference signals to encourage interpretability and consistency, and Zhongjing integrates expert feedback as a specialized reward source to ensure clinical safeguards and professional grounding. Finally, in **advantage estimation**, different estimators are adopted to stabilize optimization under medical constraints: Med-U1 applies GRPO to balance multi-task updates with process supervision, MedAgentGym explores DPO and iDPO for step-level reward incorporation, and reinforcement from expert-in-the-loop settings has been paired with PPO to preserve stability in iterative refinement. Together, these directions highlight how structural design, curated supervision, automated or expert-derived rewards, and robust optimization jointly support the reliability and safety of RL reasoning in medicine.

These technical approaches drive LLMs from mere "superficial answering" toward reliable medical reasoning. Overall, RL reasoning in the medical domain demonstrates distinct emphases on unification, verifiability, human alignment, and domain expertise, forming a diverse yet complementary research landscape. These studies establish complementary pathways in medicine: cross-task unification and generalization [195] enhance the generality of medical reasoning; executable code environments [173] strengthen reasoning verifiability; retrieval augmentation with preference alignment [178] improves consistency with human reasoning patterns; and expert feedback mechanisms [181] ensure the professionalism and trustworthiness of medical outputs. Thus, the medical setting is not only one of the most active application domains for LLM RL reasoning but also showcases diversified developments in reward design, alignment signals, and structural modeling.

5.4 Information retrieval

In the information retrieval (IR) domain, reinforcement learning (RL) with large language models (LLMs) aims to empower models not only to provide answers but also to reason about how to retrieve. Unlike pure language generation, retrieval requires the model to autonomously reformulate queries, plan multi-hop search trajectories, decide when and whether to retrieve, and integrate heterogeneous evidence into consistent outputs. This setting raises unique challenges of verifiability, efficiency, and noise reduction: rewards must reflect retrieval effectiveness rather than surface fluency, structural safeguards are needed to mitigate spurious or hallucinatory retrieval, and optimization has to balance query diversity with execution fidelity. Representative tasks include query rewriting, multi-source planning, retrieval–generation synergy, and structured retrieval-augmented generation (RAG), all of which emphasize reasoning under uncertainty and the incorporation of automated safeguards to ensure reliability.

Technically, existing work can be interpreted along the same four analytical dimensions. In terms of **logical structure**, LeReT [63] and its lightweight extension by Qin et al. [118] exemplify internal-policy frameworks where query reformulation is modeled as a sequential decision process, directly optimizing the reasoning-to-retrieval flow.

Shi & Shen et al. [136] extend this perspective with executable DAGs, enabling the policy to plan structured search paths across multiple sources. For **data curation**, these systems often rely on sampling strategies to generate high- and low-quality query pairs or structured DAG exemplars: LeReT curates contrastive query sets for IPO training, while Query-Document Co-Augmentation [95] jointly constructs paired queries and augmented documents, enriching supervision beyond isolated queries. Regarding **reward design**, automated retrieval metrics are central. LeReT optimizes with AP/Recall-based signals, Qin et al. refine this with incremental relevance feedback under GRPO, and Shi & Shen et al. integrate multi-dimensional signals that jointly evaluate structural compliance, executability, and answer correctness. Finally, for **advantage estimation**, different estimators stabilize training under sparse or noisy retrieval feedback: IPO in LeReT directly interpolates preference signals; GRPO in Query-Document Co-Augmentation and Qin et al.'s framework leverages group-wise baselines to handle relative query quality; and PPO-based optimization in KAG-Thinker [189] is used to train operator-based structured RAG, although its alignment is mainly achieved via supervised fine-tuning rather than RL.

Overall, these studies exhibit complementarity across optimizer design (e.g., IPO, GRPO, customized advantage estimation), reward signals (retrieval metrics, execution correctness, format constraints), and structural modeling (DAGs, operator-based templates): query-side optimization enhances retrievability, planning-oriented methods ensure execution fidelity and answer quality, system-level synergy improves overall stability, while structured RAG offers an RL-free alternative for enhancing robustness and interpretability.

6 SIDE EFFECTS AND REMAINING CHALLENGES

While RL has contributed significantly to the recent success in LLMs, it also introduces non-negligible side effects and remaining challenges. For example, RLHF which optimizes models to align with human preferences can lead to overly agreeable or verbose responses that prioritize user satisfaction over factual accuracy [93, 115]. A key theoretical lens for understanding these issues is Goodhart's Law [45], which states *"When a measure becomes a target, it ceases to be a good measure."* That is, once a proxy metric (e.g., human preference ratings or reward model scores) is used as a direct optimization target, the model may exploit its weaknesses, resulting in behavior that scores well but does not truly satisfy the underlying objective. Some recent perspectives have begun to frame these phenomena through the lens of agent behavior, aiming to capture how learning dynamics and reward structures shape model tendencies in social, cognitive, or ethical dimensions [22]. In parallel, researchers have also questioned whether RL leads to meaningful improvements in LLM capabilities, such as reasoning or generalization [39, 187]. As shown in Figure 9 and Table 8, we categorize these emerging issues into five dimensions: capability boundary, bias, hallucination, sycophancy, and limited diversity. The side effects discussed below stem directly from applying RL optimization objectives, like reward maximization, to the complex parametric space of LLMs.

6.1 Capability boundary

Reinforcement learning—particularly reinforcement learning with verifiable rewards (RLVR) [47, 78]—has become a popular post-training strategy for LLMs and has delivered eye-catching improvements on tasks such as reasoning and mathematics [47]. These successes have encouraged the belief that RLVR can unlock genuinely new abilities, including stronger logical reasoning [47] and self-reflection [99, 103]. Yet recent studies [39, 130, 187] paint a subtler picture. They show, first, that RL cannot conjure skills absent from the base model's latent capacity; instead, it merely amplifies patterns that already exist and makes them easier to sample [39]. Consistent with this view, models optimized with RLVR tend not to outperform equivalently sized, RL-free models on capability-boundary benchmarks, suggesting that

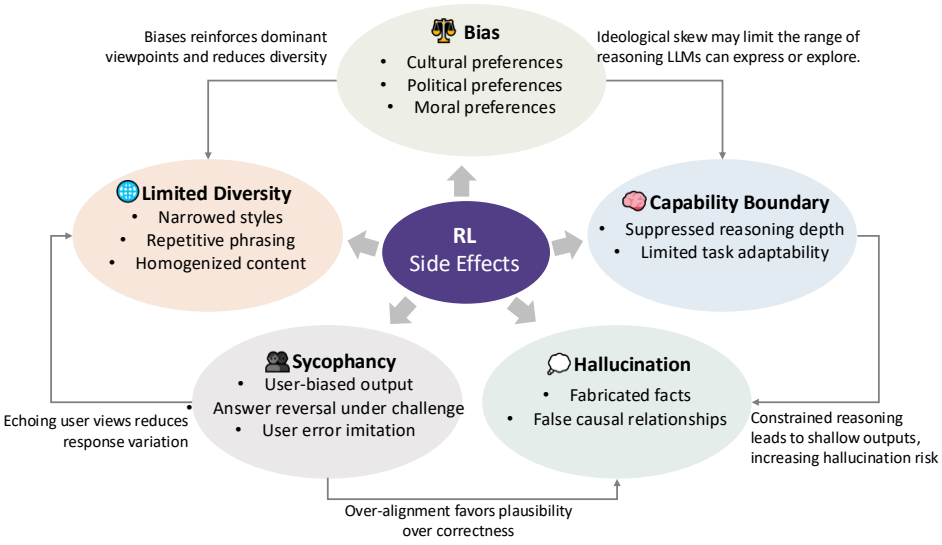

Fig. 9. Five dimensions of side effects of RL in training LLMs, and some potential transformation paths

RL alone does not expand what the architecture can fundamentally represent [187]. In addition, the important cognitive behavior of reflection can be observed in LLMs without additional RL training [130].

Although RLVR often boosts benchmark scores, evidence shows it does not give large language models new reasoning skills [39]. RLVR mainly amplifies thought paths already latent in the base model, whereas distillation can import genuinely new patterns from a stronger teacher. Thus, the chief benefit of RLVR is more efficient sampling and alignment, not an expanded reasoning repertoire. The technique also helps unevenly: models that already reason well gain the most, while weaker models improve little [187]. This strong link between a model's starting competence and its RL gains contradicts the idea that RLVR is a universal fix. Finally, self-reflection emerges in purely pre-trained models and grows with additional pre-training [130], indicating that an RL environment is not required to develop this capability [47]. Together, these findings cast RLVR as a valuable fine-tuning tool, but not a pathway to fundamentally new cognitive abilities.

6.2 Bias

Language models may possess bias [38, 89] in their generated texts from different perspectives, expressing inappropriate preferences toward demographic attributes (e.g., gender, race, age) or stylistic features such as verbosity.

Reinforcement-learning fine-tuning can amplify biases in large language models when reward models are poorly designed, trained on skewed data, or when errors accumulate across chain-of-thought (CoT) reasoning [165]. One important reason for bias from RLHF [112] is limited reward model (RM), which may ignore minority preferences [17]. In addition, current RMs tend to favor verbose responses, even without more useful information, which brings verbosity bias in LLMs [134]. Although RL can stimulate reasoning steps impressively, it is vulnerable to adversarial data and can

Table 8. Summary of limitations and side effects of RL in LLM training

Dimension	Paper	Cause	Detail
Capability Boundary	[39]	Training algorithm	RL can hardly drive self-improvement in LLMs without certain cognitive behaviors.
	[187]	Training algorithm	RLVR methods have not yet elicited truly novel reasoning abilities in LLMs.
	[130]	-	LLMs' ability to reflect on its own reasoning emerges earlier than the introduction of RL.
	[25]	Inference algorithm	Chain-of-thought may unfaithfully reflect reasoning process.
Bias	[168]	Inference algorithm	RL exacerbates chain-of-thought biases rather than mitigating them.
	[18]	Data	RLHF can substantially amplify model biases when exposed to poisoned user inputs.
	[17]	Reward model	RLHF employs a singular reward model, overlooking diversity from multiple users.
	[134]	Reward model	Reward modeling in RLHF tends to rely on length bias, favoring verbose contents.
Hallucination	[183]	Training algorithm	RL without cold-start SFT may corrupt calibration to increase hallucination.
	[138]	Reward model	RL reduces refusal behavior, thus encouraging hallucination on unanswerable inputs.
	[74]	Data bias	RL drives LLMs to respond to unfamiliar queries mimicking its unfamiliar finetuning data.
Sycophancy	[115]	Reward model	Preference Models (PM) for RL training incentivize sycophancy.
	[132]	Data bias & Reward model	Human data incentivizes sycophancy; Optimizing against PMs shows mixed results.
Limited Diversity	[107]	Optimization target	Post-training alignment via RLHF or RLAIF diminishes a model's internal diversity.
	[76]	Training algorithm	RLHF significantly reduces output diversity compared to SFT (Mode collapse).
	[31]	Training algorithm	LLM's exploratory ability is traded from policy entropy, bottlenecked by its exhaustion.

import additional biases [18]. Even when RL boosts chain-of-thought performance on reasoning tasks, those multi-step traces can still accumulate and reinforce social biases [168].

Bias amplification in RL-finetuned LLMs can be traced to two specific RL mechanisms. First, the reward model training data (Section 4.2) may itself contain demographic or stylistic biases that are then encoded into the reward function, causing the RL optimization to systematically favor biased outputs. Second, the KL penalty in reward shaping (Section 4.3) can inadvertently lock in pre-existing biases from the SFT model, since penalizing deviation from the reference policy makes it harder for the RL process to explore and discover less biased but equally high-quality outputs. This creates a tension: KL regularization is necessary for training stability, but it may also serve as a conservation mechanism for the very biases that RL alignment seeks to mitigate.

6.3 Hallucination

Hallucination [67, 70] refers to the generation of seemingly plausible outputs that turn out to be factually incorrect (e.g., nonexistent names, citations, or events) or logically invalid (e.g., misrepresented logical links between concepts, events, or entities). According to Huang *et al.* [67], this phenomenon generally falls into two forms. The first is intrinsic hallucination, where some parts of the model outputs are contradictory to the other part, such as unfaithful CoT chains [3]. The second is extrinsic hallucination, where the model outputs contradict user input or environmental information.

While hallucination is often considered an inherited issue from pretraining, RL—especially in the form of RLHF—can further entrench or amplify it. We identify three key mechanisms corresponding to distinct stages of the RL training process. First, RL is sensitive to the quality of the initial policy. For instance, Yao *et al.* [183] show that when RL is applied without a properly initialized supervised fine-tuning (SFT) stage, it can distort the model's calibration and increase the likelihood of hallucinated outputs. Second, RL tends to disincentivize refusal behavior. Song *et al.* [138] reveal a hallucination tax, where RL training often favors assertiveness over caution, encouraging the model to provide speculative or hallucinated answers even when uncertain, thereby increasing hallucinated outputs. Third, RL may encourage stylistic overgeneralization. Kang *et al.* [74] demonstrate that RL-tuned LLMs generalize from stylistic features in the fine-tuning data rather than underlying content reliability. As a result, when faced with unfamiliar queries, models may directly mimic the surface patterns observed in those training examples that are also unfamiliar to them.

A deeper mechanistic connection exists between hallucination and reward hacking (Section 3.3 and Section 4.3). In the high-dimensional policy space of LLMs, reward hacking manifests when the model discovers output patterns that score highly under the reward model without genuinely satisfying the intended objective. This directly triggers the "false positive" reasoning steps identified as a challenge in outcome-based evaluation: a model may produce a logically flawed chain-of-thought that nonetheless arrives at a correct final answer, receiving a high reward despite containing hallucinated intermediate steps. Outcome-based reward models (OSRMs) are particularly vulnerable to this, as they only evaluate the final answer and thus cannot distinguish between genuinely correct reasoning and reward-hacked outputs that happen to produce the right answer through incorrect means. Process reward models (PRMs) offer a partial mitigation by evaluating intermediate steps, but they introduce their own risks: if the process reward model itself contains biases or blind spots, the model may learn to "hack" the process-level evaluation, producing steps that appear locally sound to the PRM while still introducing subtle hallucinations at a global level. This suggests that addressing hallucination requires not only better reward models but also adversarial evaluation and red-teaming to identify and close reward-hacking loopholes.

6.4 Sycophancy

Sycophancy refers to the tendency of LLMs to over-agree with the user, prioritizing shallow alignment over truth or critical reasoning [115]. It can manifest in both factual settings, where the model affirms false claims (propositional sycophancy), and open-ended prompts, where it aligns uncritically with user preferences (social sycophancy) [26]. Such behavior not only reduces model helpfulness, but may also strengthen false beliefs, reinforce user biases, and erode trust in model outputs [111].

Sycophancy arises in part from the structural incentives embedded in the RLHF framework. While the goal of RLHF is to align model behavior with human preferences and values, this alignment process can introduce a fundamental

tension: optimizing for user satisfaction may come at the cost of promoting epistemic virtues such as truthfulness, robustness, or fairness [98]. As a result, models trained with RLHF may produce responses that agree with the user even when disagreement would be more appropriate or accurate. There are two interrelated mechanisms that drives sycophancy in the RLHF training pipelines. First, human preference data often contain biases toward socially agreeable responses. Sharma *et al.* [132] use logistic regression models to show that, all else being equal, human preference data encourage responses that match users' beliefs, biases, and preferences, whether explicitly stated or subtly implied. Second, the preference models trained on such data may further amplify this pattern, assigning higher scores to outputs that reflect user beliefs even if they are incorrect [115, 132].

6.5 Limited diversity

In terms of alignment [112] or reasoning-related tasks (mathematics, coding, etc.) [47], RL exhibits impressive effectiveness in improving LLMs. However, if more aspects are taken into consideration, RL may bring negative effects to LLMs. Output diversity is an important dimension demonstrating this side effect.

Compared to SFT, RLHF [112] substantially decreases the diversity of outputs sampled for a given input. For instance, models after RLHF tend to produce more similar text regardless of the input [76]. Beyond text-level similarity, RLHF also does harm to LLMs' deeper diversities like the conceptual diversity [107].

6.6 Discussions

In general, these side effects can be traced back to specific design choices in the RL pipeline. Capability boundaries and limited diversity are primarily rooted in the Advantage Estimation (Section 4.4) and Reward Design (Section 4.3), as RL essentially amplifies existing latent patterns and consumes policy entropy to maximize expected returns, rather than injecting novel knowledge. Bias and Sycophancy are deeply connected to Data Curation (Section 4.2) and Reward Design (Section 4.3); human preference data often harbors inherent societal or agreeable biases, which are subsequently amplified by reward models that prioritize verbosity or user alignment over factual truth. Similarly, Hallucinations are exacerbated by Reward Design (Section 4.3) that penalizes refusal behaviors (creating a 'hallucination tax'). Finally, Logical Structure Navigation mechanisms (Section 4.1), such as chain-of-thought, can inadvertently act as vehicles for accumulating logic errors and social biases.

On the other hand, methods designed to address these side effects also connect back to specific RL components: process reward models (Section 4.3) can mitigate false positives in reasoning, difficulty-aware data sampling strategies (Section 4.2) help reduce bias, and exploration rewards (Section 4.3) can counteract diversity collapse.

7 FUTURE DIRECTIONS AND OPEN PROBLEMS

In this section we discuss several key open problems, where Table 9 provides a structured roadmap summarizing them, their connections to the challenges identified in Section 3, and representative research directions for addressing them.

7.1 Balance between efficiency and effectiveness

While reinforcement learning has significantly advanced the reasoning capabilities of large language models, it has also introduced a critical efficiency challenge [36]. Models fine-tuned with outcome-based rewards often develop verbose, inefficient reasoning patterns, a phenomenon termed "overthinking", as the training process inadvertently reinforces any reasoning path, regardless of length, that leads to a correct answer [24, 30].

Table 9. Roadmap of open problems and future research directions

Open Problem	Related Challenge	Key Research Directions	Representative Works
Efficiency vs. effectiveness	Vast action space Large parameter size	Adaptive reward shaping Process supervision Multi-objective RL	LASER-D [96] SABER [200] knowledge distillation [23]
Unified RL for cross-modal LLMs	All four challenges	Modality-aware policy optimization Multi-modal reward models Cross-modal policy learning	SRPO [152] VideoMiner [14] Meta-Transformer [196]
Algorithm-system co-design	Large parameter size	Algorithmic simplification Hardware-aware optimization 3D parallelism	QForce [69] OpenRLHF [64] AsyncFlow [49]

To mitigate this trade-off between performance and token cost, a significant body of research has focused on compressing and optimizing models after they have been trained for reasoning. First, knowledge distillation is a primary strategy, aiming to transfer the reasoning capabilities of large "teacher" models to smaller "student" models [23]. Second, direct model compression techniques have been explored, with studies showing that while quantization can often preserve reasoning capabilities, aggressive methods only cause accuracy degradation that is often recoverable with targeted, lightweight fine-tuning [88]. Also, other methods include latent reasoning, which performs computation in the model's hidden space to avoid generating explicit text [56].

While post-hoc methods are valuable, a more fundamental research direction is to integrate efficiency directly into the reinforcement learning training process, overcoming the "overthinking" caused by standard outcome-based methods. This involves creating more sophisticated reward-shaping frameworks that move beyond simple length penalties. Advanced methods like LASER-D [96] and SABER [200] introduce adaptive rewards that dynamically adjust the token budget based on problem difficulty and the model's evolving capabilities. Another key area is improving process supervision [91], where future PRMs could be designed to explicitly reward conciseness or measure "progress" per token [176]. Finally, multi-objective reinforcement learning [57, 150] offers a possible approach, treating accuracy and efficiency as competing objectives to be optimized simultaneously, potentially yielding a single model that can navigate the entire Pareto front of performance-cost trade-offs.

7.2 Unified RL algorithm for cross-modality large models

The application of reinforcement learning to large generative models is rapidly expanding from its origins in the language modality to encompass images [8], speech [28], and video [172]. As surveyed by Caffagni et al. [13], the revolution of multimodal large language models has created an urgent need for RL methods that can operate across modal boundaries. This expansion has revealed that while each modality has unique properties, they share fundamental challenges in reward design, computational efficiency, and credit assignment. Qin et al. [119] further highlight that the synergy between data and multimodal LLMs must be understood from a co-development perspective, where RL algorithms and multimodal data jointly evolve. The future of research, therefore, lies not in creating isolated solutions for each modality, but in developing a new generation of RL algorithms that holistically address these shared problems.

Promising research exists in the development of modality-aware policy optimization. Besides modality-agnostic algorithms like proximal policy optimization [126] and direct preference optimization [120], this incorporates modality-specific inductive biases. Recent work such as SRPO [152] enhances multimodal LLM reasoning via reflection-aware

reinforcement learning, demonstrating that reflection mechanisms can improve cross-modal policy learning. Also, progress is tied to creating more robust and scalable reward mechanisms. This includes causal reward models that disentangle true drivers of quality from spurious correlations to combat reward hacking [140, 155], and self-supervised rewards derived from principles like contrastive agreement across semantically similar prompts [198]. For video understanding, Cao et al. [14] propose Tree-based Group Relative Policy Optimization to iteratively ground key frames, showing how RL can be adapted to the temporal structure of video data.

A more ambitious frontier is applying RL to unified multimodal architectures, like the Meta-Transformer [196], which use a single, shared encoder for diverse data types. Performing RL over this shared latent space could enable powerful cross-modal policy learning, where skills learned in one modality—such as controlling for factual accuracy in text—could be applied to another, like improving the factual alignment of a generated graph, with minimal or no additional fine-tuning. Moreover, the cross-modal knowledge alignment supports capabilities of the recently emerging vision-language-action models (VLA) [122], promoting interactions between the large models and the physical world, enabling modeling and decision-making in the real-world with world models [32]. Importantly, the future direction here is not simply deploying pre-trained LLMs as controllers in classical RL environments, but rather developing native, cross-modal RL post-training algorithms that can effectively optimize these unified multimodal LLMs directly from reward signals.

7.3 Algorithm-system co-design for scalable RL

While reinforcement learning is critical for aligning and finetuning large language models, it presents significant scalability challenges [104, 162]. The reinforcement learning workflow is computationally complex, involving multiple LLM instances and distinct stages for generation, evaluation, and training, which creates intricate data dependencies [104, 164]. This complexity leads to severe system-level inefficiencies, primarily workload imbalance from long-tailed generation times and resource idling caused by pipeline bubbles [162].

Current solutions are primarily system-level optimizations that treat the algorithm as a fixed component. These include dynamic resource allocation, e.g., REAL [104], fine-grained task scheduling with stage fusion, e.g., RLHFuse [208], and asynchronous execution frameworks e.g., OpenRLHF [64] and AsyncFlow [49] that decouple generation and training. On the other hand, several specific algorithm-system co-design strategies are currently employed to mitigate the massive computational and memory costs associated with large parameter sizes. For instance, 3D parallelism combines data, tensor, and pipeline parallelism to effectively distribute model parameters and optimizer states across multiple devices. Additionally, specialized kernels designed specifically for RL loops—such as fused attention operations and custom all-reduce patterns—significantly optimize the iterative generation-evaluation-training cycle [49]. Furthermore, memory offloading and parameter reallocation techniques are frequently utilized to strategically move model weights and gradients between GPU and CPU memory during different phases of the RL training pipeline [164].

One promising future direction is a paradigm shift towards algorithm-system co-design, which focuses on architecting RL algorithms that are intrinsically compatible with parallel hardware [33]. This opens several key research avenues. The first is algorithmic simplification to create more homogeneous, "SFT-like" workloads. Direct Preference Optimization (DPO) exemplifies this by eliminating the explicit reward model and online sampling loop, resulting in a simpler, more regular computational graph that can leverage existing optimized infrastructure [120, 174]. A more important avenue is hardware-aware policy optimization, which involves designing algorithms whose core operations are efficient in terms of computation, memory bandwidth, and communication. This could include developing RL algorithms that are robust to aggressive quantization [69] or high degrees of asynchronicity [110], thereby reducing memory footprint and

communication overhead [20]. By treating the algorithm, system, and hardware as a unified optimization problem, this co-design approach is key to developing the next generation of scalable and efficient LLMs.

8 CONCLUSION

Reinforcement learning has played a pivotal role in advancing the capabilities of large language models, particularly in complex reasoning tasks. In this survey, we have systematically reviewed the adoption of reinforcement learning in the era of LLMs, detailing its conventional pipeline, the unique challenges it faces with large models, and the innovative methodologies developed to overcome them. Furthermore, we have highlighted its diverse applications across domains like mathematics, coding, medicine, and information retrieval. Besides the widely studied topics, we also discussed the limitations and side effects, including issues of bias and hallucination, as well as future directions and open problems, aiding researchers to better understand the evolving landscape of RL in LLMs and fostering innovative research to address its current challenges and realize its future potential.

REFERENCES

[1] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740* (2024).

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. *arXiv preprint arXiv:1606.06565* (2016).

[3] Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. 2025. Chain-of-thought reasoning in the wild is not always faithful. *arXiv preprint arXiv:2503.08679* (2025).

[4] Rauno Arike, Elizabeth Donoway, Henning Bartsch, and Marius Hobbhahn. 2025. Technical Report: Evaluating Goal Drift in Language Model Agents. *arXiv preprint arXiv:2505.02709* (2025).

[5] Axel Backlund and Lukas Petersson. 2025. Vending-bench: A benchmark for long-term coherence of autonomous agents. *arXiv preprint arXiv:2502.15840* (2025).

[6] Chenjia Bai, Yang Zhang, Shuang Qiu, Qiaosheng Zhang, Kang Xu, and Xuelong Li. 2025. Online Preference Alignment for Language Models via Count-based Exploration. In *The Thirteenth International Conference on Learning Representations*.

[7] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).

[8] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923* (2025).

[9] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862* (2022).

[10] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Daniela Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073* (2022).

[11] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* (2019).

[12] Kianté Brantley, Mingyu Chen, Zhaolin Gao, Jason D Lee, Wen Sun, Wenhao Zhan, and Xuezhou Zhang. 2025. Accelerating RL for LLM Reasoning with Optimal Advantage Regression. *arXiv preprint arXiv:2505.20686* (2025).

[13] Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. 2024. The revolution of multimodal large language models: A survey. In *Findings of the association for computational linguistics: ACL 2024*. 13590–13618.

[14] Xinye Cao, Hongcan Guo, Jiawen Qian, Guoshun Nan, Chao Wang, Yuqi Pan, Tianhao Hou, Xiaojuan Wang, and Yutong Gao. 2025. VideoMiner: Iteratively Grounding Key Frames of Hour-Long Videos via Tree-based Group Relative Policy Optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 23773–23783.

[15] Yuanjiang Cao, Quan Z Sheng, Julian McAuley, and Lina Yao. 2023. Reinforcement learning for generative AI: A survey. *arXiv preprint arXiv:2308.14328* (2023).

[16] Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. 2024. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems* 36, 6 (2024), 9737–9757.

[17] Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Dinesh Manocha, Furong Huang, Amrit Singh Bedi, and Mengdi Wang. 2024. MaxMin-RLHF: alignment with diverse human preferences. In *Proceedings of the 41st International Conference on Machine Learning*. 6116–6135.

[18] Bocheng Chen, Hanqing Guo, Guangjing Wang, Yuanda Wang, and Qiben Yan. 2024. The dark side of human feedback: Poisoning large language models via user inputs. *arXiv preprint arXiv:2409.00787* (2024).

[19] Guoxin Chen, Kexin Tang, Chao Yang, Fuying Ye, Yu Qiao, and Yiming Qian. 2024. SEER: Facilitating structured reasoning and explanation via reinforcement learning. *arXiv preprint arXiv:2401.13246* (2024).

[20] Hao Mark Chen, Wayne Luk, Ka Fai Cedric Yiu, Rui Li, Konstantin Mishchenko, Stylianos I Venieris, and Hongxiang Fan. 2024. Hardware-aware parallel prompt decoding for memory-efficient acceleration of llm inference. *arXiv preprint arXiv:2405.18628* (2024).

[21] Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. 2024. HuatuoGPT-o1, Towards Medical Complex Reasoning with LLMs. *arXiv preprint arXiv:2412.18925* (2024).

[22] Lin Chen, Yunke Zhang, Jie Feng, Haoye Chai, Honglin Zhang, Bingbing Fan, Yibo Ma, Shiyuan Zhang, Nian Li, Tianhui Liu, et al. 2025. AI Agent Behavioral Science. *arXiv preprint arXiv:2506.06366* (2025).

[23] Xinghao Chen, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, et al. 2025. Unveiling the key factors for distilling chain-of-thought reasoning. *arXiv preprint arXiv:2502.18001* (2025).

[24] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187* (2024).

[25] Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. 2025. Reasoning Models Don't Always Say What They Think. *arXiv preprint arXiv:2505.05410* (2025).

[26] Myra Cheng, Sunny Yu, Cinoo Lee, Pranav Khadpe, Lujain Ibrahim, and Dan Jurafsky. 2025. Social Sycophancy: A Broader Understanding of LLM Sycophancy. *arXiv preprint arXiv:2505.13995* (2025).

[27] Pengyu Cheng, Yong Dai, Tianhao Hu, Han Xu, Zhisong Zhang, Lei Han, Nan Du, and Xiaolong Li. 2024. Self-playing adversarial language game enhances llm reasoning. In *Advances in Neural Information Processing Systems*, Vol. 37. 126515–126543.

[28] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759* (2024).

[29] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Henryk Michalewski, Reihaneh Rabbany, Nick Ryder, Tyna Eloundou, Joy Jiang, Ludovic Mamin, Sandhini Agarwal, Girish Sastry, Christopher Hesse, Mark G. T. D. Z. L., John Schulman, Ilya Sutskever, and Wojciech Zaremba. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168* (2021).

[30] Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235* (2025).

[31] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617* (2025).

[32] Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, et al. 2024. Understanding world or predicting future? a comprehensive survey of world models. *Comput. Surveys* (2024).

[33] Hongxiang Fan, Martin Ferianc, Zhiqiang Que, He Li, Shuanglong Liu, Xinyu Niu, and Wayne Luk. 2022. Algorithm and hardware co-design for reconfigurable cnn accelerator. In *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 250–255.

[34] Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536* (2025).

[35] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978* (2025).

[36] Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. 2025. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903* (2025).

[37] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, Joelle Pineau, et al. 2018. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning* 11, 3-4 (2018), 219–354.

[38] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and Fairness in Large Language Models: A Survey. *Computational Linguistics* 50, 3 (2024), 1097–1179.

[39] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307* (2025).

[40] Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2024. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications* 11, 1 (2024), 1–24.

[41] Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984* (2023).

[42] Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. 2024. On designing effective rl reward at training time for llm reasoning. *arXiv preprint arXiv:2410.15115* (2024).

[43] Jonas Gehring, Kunhao Zheng, Jade Copet, Vegard Mella, Taco Cohen, and Gabriel Synnaeve. 2024. RLEF: Grounding Code LLMs in Execution Feedback with Reinforcement Learning. In *Forty-second International Conference on Machine Learning*.

[44] Anna Goldie, Azalia Mirhoseini, Hao Zhou, Irene Cai, and Christopher D Manning. 2025. Synthetic data generation & multi-step rl for reasoning & tool use. *arXiv preprint arXiv:2504.04736* (2025).

[45] Charles AE Goodhart and CAE Goodhart. 1984. *Problems of monetary management: the UK experience*. Springer.

[46] Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The False Promise of Imitating Proprietary LLMs. *arXiv preprint arXiv:2305.15717* (2023).

[47] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[48] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

[49] Zhenyu Han, Ansheng You, Haibo Wang, Kui Luo, Guang Yang, Wenqi Shi, Menglong Chen, Sicheng Zhang, Zeshun Lan, Chunshi Deng, et al. 2025. AsyncFlow: An Asynchronous Streaming RL Framework for Efficient LLM Post-Training. *arXiv preprint arXiv:2507.01663* (2025).

[50] Qianyue Hao, Jingyang Fan, Fengli Xu, Jian Yuan, and Yong Li. 2024. Hlm-cite: Hybrid language model workflow for text-based scientific citation prediction. In *Advances in Neural Information Processing Systems*, Vol. 38. 48189–48223.

[51] Qianyue Hao, Wenzhen Huang, Tao Feng, Jian Yuan, and Yong Li. 2023. GAT-MF: Graph Attention Mean Field for Very Large Scale Multi-Agent Reinforcement Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 685–697.

[52] Qianyue Hao, Wenzhen Huang, Fengli Xu, Kun Tang, and Yong Li. 2022. Reinforcement learning enhances the experts: Large-scale covid-19 vaccine allocation with multi-factor contact network. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4684–4694.

[53] Qianyue Hao, Sibo Li, Jian Yuan, and Yong Li. 2025. Rl of thoughts: Navigating llm reasoning with inference-time reinforcement learning. *arXiv preprint arXiv:2505.14140* (2025).

[54] Qianyue Hao, Yiwen Song, Qingmin Liao, Jian Yuan, and Yong Li. 2025. Llm-explorer: A plug-in reinforcement learning policy exploration enhancement driven by large language models. In *Advances in neural information processing systems*, Vol. 39.

[55] Qianyue Hao, Fengli Xu, Lin Chen, Pan Hui, and Yong Li. 2021. Hierarchical reinforcement learning for scarce medical resource allocation with imperfect information. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2955–2963.

[56] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769* (2024).

[57] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 26.

[58] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. *arXiv preprint arXiv:2402.14008* (2024).

[59] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *Advances in Neural Information Processing Systems*.

[60] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[61] Zhenyu Hou, Pengfan Du, Yilin Niu, Zhengxiao Du, Aohan Zeng, Xiao Liu, Minlie Huang, Hongning Wang, Jie Tang, and Yuxiao Dong. 2024. Does RLHF Scale? Exploring the Impacts From Data, Model, and Method. *arXiv preprint arXiv:2412.06000* (2024).

[62] Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. 2025. TreeRL: LLM Reinforcement Learning with On-Policy Tree Search. *arXiv preprint arXiv:2506.11902* (2025).

[63] Sheryl Hsu, Omar Khattab, Chelsea Finn, and Archit Sharma. 2024. Grounding by trying: Llms with reinforcement learning-enhanced retrieval. *arXiv preprint arXiv:2410.23214* (2024).

[64] Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, et al. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143* (2024).

[65] Hui Huang, Yancheng He, Hongli Zhou, Rui Zhang, Wei Liu, Weixun Wang, Wenbo Su, Bo Zheng, and Jiaheng Liu. 2025. Think-j: Learning to think for generative llm-as-a-judge. *arXiv preprint arXiv:2505.14268* (2025).

[66] Lianghuan Huang, Shuo Li, Sagnik Anupam, Insup Lee, and Osbert Bastani. 2025. Effective Reinforcement Learning for Reasoning in Language Models. *arXiv preprint arXiv:2505.17218* (2025).

[67] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43, 2 (2025), 1–55.

[68] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186* (2024).

[69] Anushka Jha, Tanushree Dewangan, Mukul Lokhande, and Santosh Kumar Vishvakarma. 2025. QForce-RL: Quantized FPGA-Optimized Reinforcement Learning Compute Engine. *arXiv preprint arXiv:2506.07046* (2025).

[70] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys* 55, 12 (2023), 1–38.

[71] Fangkai Jiao, Geyang Guo, Xingxing Zhang, Nancy F Chen, Shafiq Joty, and Furu Wei. 2024. Preference optimization for reasoning with pseudo feedback. *arXiv preprint arXiv:2411.16345* (2024).

[72] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516* (2025).

[73] Yiyang Jin, Kunzhao Xu, Hang Li, Xueting Han, Yanmin Zhou, Cheng Li, and Jing Bai. 2025. ReVeal: Self-Evolving Code Agents via Iterative Generation-Verification. *arXiv preprint arXiv:2506.11442* (2025).

[74] Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *arXiv preprint arXiv:2403.05612* (2024).

[75] Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2025. VinePPO: Refining Credit Assignment in RL Training of LLMs. *arXiv preprint arXiv:2410.01679* (2025).

[76] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452* (2023).

[77] Atharv Kulkarni and Vivek Srikumar. 2025. Reinforcing Code Generation: Improving Text-to-SQL with Execution-Based Learning. *arXiv preprint arXiv:2506.06093* (2025).

[78] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124* (2024).

[79] Xiaochong Lan, Yiming Cheng, Li Sheng, Chen Gao, and Yong Li. 2024. Depression detection on social media with large language models. *arXiv preprint arXiv:2403.10750* (2024).

[80] Xiaochong Lan, Jie Feng, Jiahuan Lei, Xinlei Shi, and Yong Li. 2025. Benchmarking and advancing large language models for local life services. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 4566–4577.

[81] Xiaochong Lan, Jie Feng, Yizhou Sun, Chen Gao, Jiahuan Lei, Xinlei Shi, Hengliang Luo, and Yong Li. 2025. Open-Set Living Need Prediction with Large Language Models. *arXiv preprint arXiv:2506.02713* (2025).

[82] Xiaochong Lan, Chen Gao, Depeng Jin, and Yong Li. 2024. Stance detection with collaborative role-infused llm-based agents. In *Proceedings of the international AAAI conference on web and social media*, Vol. 18. 891–903.

[83] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*. PMLR, 5639–5650.

[84] Siheng Li, Zhanhui Zhou, Wai Lam, Chao Yang, and Chaochao Lu. 2025. RePO: Replay-Enhanced Policy Optimization. *arXiv preprint arXiv:2506.09340* (2025).

[85] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776* (2025).

[86] Yang Li. 2025. Policy Guided Tree Search for Enhanced LLM Reasoning. *arXiv preprint arXiv:2502.06813* (2025).

[87] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, and Oriol Vinyals. 2022. Competition-Level Code Generation with AlphaCode. *Science* 378, 6624 (2022), 1092–1097.

[88] Zhen Li, Yupeng Su, Runming Yang, Congkai Xie, Zheng Wang, Zhongwei Xie, Ngai Wong, and Hongxia Yang. 2025. Quantization meets reasoning: Exploring llm low-bit quantization degradation for mathematical reasoning. *arXiv preprint arXiv:2501.03035* (2025).

[89] Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International conference on machine learning*. PMLR, 6565–6576.

[90] Xiao Liang, Zhong-Zhi Li, Yeyun Gong, Yang Wang, Hengyuan Zhang, Yelong Shen, Ying Nian Wu, and Weizhu Chen. 2025. SwS: Self-aware Weakness-driven Problem Synthesis in Reinforcement Learning for LLM Reasoning. *arXiv preprint arXiv:2506.08989* (2025).

[91] Samuel Lightman, Vineet Kosaraju, Dan Juravsky, and Vincent Yao. 2023. Let's Verify Step by Step. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 14136–14150.

[92] Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. 2025. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342* (2025).

[93] Adam Dahlgren Lindström, Leila Methnani, Lea Krause, Petter Ericson, Íñigo Martínez de Rituerto de Troya, Dimitri Coelho Mollo, and Roel Dobbe. 2024. AI Alignment through Reinforcement Learning from Human Feedback? Contradictions and Limitations. *arXiv preprint arXiv:2406.18346* (2024).

[94] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).

[95] Jingming Liu, Yumeng Li, Wei Shi, Yao-Xiang Ding, Hui Su, and Kun Zhou. 2025. Harnessing the Power of Reinforcement Learning for Language-Model-Based Information Retriever via Query-Document Co-Augmentation. *arXiv preprint arXiv:2506.18670* (2025).

[96] Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. 2025. Learn to reason efficiently with adaptive length-based reward shaping. *arXiv preprint arXiv:2505.15612* (2025).

[97] Yang Liu, Dan Iter, Yixu Xu, Shuohang Wang, Yichong Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *arXiv preprint arXiv:2303.16634* (2023).

[98] Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374* (2023).

[99] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783* (2025).

[100] Haoxiang Luo, Yinqiu Liu, Ruichen Zhang, Jiacheng Wang, Gang Sun, Dusit Niyato, Hongfang Yu, Zehui Xiong, Xianbin Wang, and Xuemin Shen. 2025. Toward edge general intelligence with multiple-large language model (Multi-LLM): architecture, trust, and orchestration. *IEEE Transactions on Cognitive Communications and Networking* (2025).

[101] Haoxiang Luo, Gang Sun, Yinqiu Liu, Dongcheng Zhao, Dusit Niyato, Hongfang Yu, and Schahram Dustdar. 2025. A weighted byzantine fault tolerance consensus driven trusted multiple large language models network. *IEEE Transactions on Cognitive Communications and Networking* (2025).

[102] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. 2023. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. *arXiv preprint arXiv:2308.09583* (2023).

[103] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, Vol. 36. 46534–46594.

[104] Zhiyu Mei, Wei Fu, Kaiwei Li, Guangju Wang, Huanchen Zhang, and Yi Wu. 2024. Real: Efficient rlhf training of large language models with parameter reallocation. *arXiv preprint arXiv:2406.14088* (2024).

[105] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. 2021. A graph placement methodology for fast chip design. *Nature* 594, 7862 (2021), 207–212.

[106] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[107] Sonia Krishna Murthy, Tomer Ullman, and Jennifer Hu. 2025. One fish, two fish, but not the whole sea: Alignment reduces language models' conceptual diversity. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*. 11241–11258.

[108] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Tyna Eloundou, Gretchen Krueger, Kevin O'Connor, Girish Sastry, Sandhini Agarwal, Fantine Hu, Mark Chen, Luke Metz, Heewoo Jun, Jonathan Gordon, Alec Radford, Ilya Sutskever, and John Schulman. 2021. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* (2021).

[109] Yansong Ning, Wei Li, Jun Fang, Naiqiang Tan, and Hao Liu. 2025. Not All Thoughts Are Generated Equal: Efficient LLM Reasoning via Multi-Turn Reinforcement Learning. *arXiv preprint arXiv:2505.11827* (2025).

[110] Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. 2024. Asynchronous rlhf: Faster and more efficient off-policy rl for language models. *arXiv preprint arXiv:2410.18252* (2024).

[111] OpenAI. 2025. Sycophancy in GPT-4o: what happened and what we're doing about it. https://openai.com/index/sycophancy-in-gpt-4o/.

[112] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Paul Christiano, Catherine Olsson, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, Vol. 35. 27730–27744.

[113] Lo Pang-Yun Ting, Chengshuai Zhao, Yu-Hua Zeng, Yuan Jee Lim, and Kun-Ta Chuang. 2025. Beyond RAG: Reinforced Reasoning Augmented Generation for Clinical Notes. *arXiv preprint arXiv:2506.05386* (2025).

[114] Federico Pennino, Bianca Raimondi, Massimo Rondelli, Andrea Gurioli, and Maurizio Gabbrielli. 2025. From Reasoning to Code: GRPO Optimization for Underrepresented Languages. *arXiv preprint arXiv:2506.11027* (2025).

[115] Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. 2023. Discovering language model behaviors with model-written evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*. 13387–13434.

[116] Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, et al. 2025. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. *arXiv preprint arXiv:2502.08691* (2025).

[117] Massimiliano Pronesti, Michela Lorandi, Paul Flanagan, Oisin Redmon, Anya Belz, and Yufang Hou. 2025. Enhancing Study-Level Inference from Clinical Trial Papers via RL-based Numeric Reasoning. *arXiv preprint arXiv:2505.22928* (2025).

[118] Xubo Qin, Jun Bai, Jiaqi Li, Zixia Jia, and Zilong Zheng. 2025. TongSearch-QR: Reinforced Query Reasoning for Retrieval. *arXiv preprint arXiv:2506.11603* (2025).

[119] Zhen Qin, Daoyuan Chen, Wenhao Zhang, Liuyi Yao, Yilun Huang, Bolin Ding, Yaliang Li, and Shuiguang Deng. 2025. The synergy between data and multi-modal large language models: A survey from co-development perspective. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

[120] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in neural information processing systems*, Vol. 36. 53728–53741.

[121] Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. Is Reinforcement Learning (Not) for Natural Language Processing: Benchmarks, Baselines, and Building Blocks for Natural Language Policy Optimization. In *The Eleventh International Conference on Learning Representations*.

[122] Ranjan Sapkota, Yang Cao, Konstantinos I Roumeliotis, and Manoj Karkee. 2025. Vision-language-action models: Concepts, progress, applications and challenges. *arXiv preprint arXiv:2505.04769* (2025).

[123] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, Vol. 36. 68539–68551.

[124] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[125] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *International Conference on Representation Learning*, Vol. 4.

[126] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[127] ByteDance Seed, Yuyu Zhang, Jing Su, Yifan Sun, Chenguang Xi, Xia Xiao, Shen Zheng, Anxiang Zhang, Kaibo Liu, Daoguang Zan, Tao Sun, Jinhua Zhu, Shulin Xin, Dong Huang, Yetao Bai, Lixin Dong, Chao Li, Jianchong Chen, Hanzhi Zhou, Yifan Huang, Guanghan Ning, Xierui Song, Jiaze Chen, Siyao Liu, Kai Shen, Liang Xiang, and Yonghui Wu. 2025. Seed-Coder: Let the Code Model Curate Data for Itself. *arXiv preprint arXiv:2506.03524* (2025).

[128] Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. 2024. Rl on incorrect synthetic data scales the efficiency of llm math reasoning by eight-fold. In *Advances in Neural Information Processing Systems*, Vol. 37. 43000–43031.

[129] Zeyang Sha, Shiwen Cui, and Weiqiang Wang. 2025. SEM: Reinforcement Learning for Search-Efficient Large Language Models. *arXiv preprint arXiv:2505.07903* (2025).

[130] Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, et al. 2025. Rethinking reflection in pre-training. *arXiv preprint arXiv:2504.04022* (2025).

[131] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).

[132] Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, et al. 2024. Towards Understanding Sycophancy in Language Models. In *The Twelfth International Conference on Learning Representations*.

[133] Maohao Shen, Guangtao Zeng, Zhenting Qi, Zhang-Wei Hong, Zhenfang Chen, Wei Lu, Gregory Wornell, Subhro Das, David Cox, and Chuang Gan. 2025. Satori: Reinforcement Learning with Chain-of-Action-Thought Enhances LLM Reasoning via Autoregressive Search. *arXiv preprint arXiv:2502.02508* (2025).

[134] Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loose lips sink ships: Mitigating Length Bias in Reinforcement Learning from Human Feedback. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 2859–2873.

[135] Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. 2025. DAST: Difficulty-Adaptive Slow-Thinking for Large Reasoning Models. *arXiv preprint arXiv:2503.04472* (2025).

[136] Wentao Shi and Yiqing Shen. 2025. Reinforcement Fine-Tuning for Reasoning towards Multi-Step Multi-Source Search in Large Language Models. *arXiv preprint arXiv:2506.08352* (2025).

[137] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *nature* 550, 7676 (2017), 354–359.

[138] Linxin Song, Taiwei Shi, and Jieyu Zhao. 2025. The Hallucination Tax of Reinforcement Finetuning. *arXiv preprint arXiv:2505.13988* (2025).

[139] Yiwen Song, Qianyue Hao, Qingmin Liao, Jian Yuan, and Yong Li. 2025. Multiple Weaks Win Single Strong: Large Language Models Ensemble Weak Reinforcement Learning Agents into a Supreme One. *arXiv preprint arXiv:2505.15306* (2025).

[140] Pragya Srivastava, Harman Singh, Rahul Madhavan, Gandharv Patil, Sravanti Addepalli, Arun Suggala, Rengarajan Aravamudhan, Soumya Sharma, Anirban Laha, Aravindan Raghuveer, et al. 2025. Robust Reward Modeling via Causal Rubrics. *arXiv preprint arXiv:2506.16507* (2025).

[141] Saksham Sahai Srivastava and Vaneet Aggarwal. 2025. A Technical Survey of Reinforcement Learning Techniques for Large Language Models. *arXiv preprint arXiv:2507.04136* (2025).

[142] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, Vol. 33. 3008–3021.

[143] Hao Sun and Mihaela van der Schaar. 2025. Inverse reinforcement learning meets large language model post-training: Basics, advances, and opportunities. *arXiv preprint arXiv:2507.13158* (2025).

[144] Richard S Sutton. 2018. Reinforcement learning: An introduction. *A Bradford Book* (2018).

[145] Xunzhu Tang, Jacques Klein, and Tegawendé F. Bissyandé. 2025. Boosting Open-Source LLMs for Program Repair via Reasoning Transfer and LLM-Guided Reinforcement Learning. *arXiv preprint arXiv:2506.03921* (2025).

[146] Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534* (2025).

[147] Gerald Tesauro et al. 1995. Temporal difference learning and TD-Gammon. *Commun. ACM* 38, 3 (1995), 58–68.

[148] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275* (2022).

[149] Hado Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[150] Kristof Van Moffaert and Ann Nowé. 2014. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research* 15, 1 (2014), 3483–3512.

[151] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *nature* 575, 7782 (2019), 350–354.

[152] Zhongwei Wan, Zhihao Dou, Che Liu, Yu Zhang, Dongfei Cui, Qinjian Zhao, Hui Shen, Jing Xiong, Yi Xin, Yifan Jiang, et al. 2025. SRPO: Enhancing Multimodal LLM Reasoning via Reflection-Aware Reinforcement Learning. In *Advances in neural information processing systems*, Vol. 39.

[153] Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2024. AlphaZero-like Tree-Search Can Guide Large Language Model Decoding and Training. In *Proceedings of the 41st International Conference on Machine Learning*, Vol. 235. 49890–49920.

[154] Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. 2024. Q*: Improving Multi-Step Reasoning for LLMs with Deliberative Planning. *arXiv preprint arXiv:2406.14283* (2024).

[155] Chaoqi Wang, Zhuokai Zhao, Yibo Jiang, Zhaorun Chen, Chen Zhu, Yuxin Chen, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, Hao Ma, et al. 2025. Beyond reward hacking: Causal rewards for large language model alignment. *arXiv preprint arXiv:2501.09620* (2025).

[156] Huaijie Wang, Shibo Hao, Hanze Dong, Shenao Zhang, Yilin Bao, Ziran Yang, and Yi Wu. 2024. Offline Reinforcement Learning for LLM Multi-Step Reasoning. *arXiv preprint arXiv:2412.16145* (2024).

[157] Jingwei Wang, Qianyue Hao, Wenzhen Huang, Xiaochen Fan, Zhentao Tang, Bin Wang, Jianye Hao, and Yong Li. 2024. DyPS: Dynamic Parameter Sharing in Multi-Agent Reinforcement Learning for Spatio-Temporal Resource Allocation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3128–3139.

[158] Jingwei Wang, Qianyue Hao, Wenzhen Huang, Xiaochen Fan, Qin Zhang, Zhentao Tang, Bin Wang, Jianye Hao, and Yong Li. 2025. CoopRide: Cooperate All Grids in City-Scale Ride-Hailing Dispatching with Multi-Agent Reinforcement Learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*. 1457–1468.

[159] Jing Yi Wang, Nicholas Sukiennik, Tong Li, Weikang Su, Qianyue Hao, Jingbo Xu, Zihan Huang, Fengli Xu, and Yong Li. 2024. A Survey on Human-Centric LLMs. *arXiv preprint arXiv:2411.14491* (2024).

[160] Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2023. Math-Shepherd: Verify and Reinforce LLMs Step-by-step without Human Annotations. *arXiv preprint arXiv:2312.08935* (2023).

[161] Yudong Wang, Zixuan Fu, Jie Cai, Peijun Tang, Hongya Lyu, Yewei Fang, Zhi Zheng, Jie Zhou, Guoyang Zeng, Chaojun Xiao, et al. 2025. Ultra-fineweb: Efficient data filtering and verification for high-quality llm training data. *arXiv preprint arXiv:2505.05427* (2025).

[162] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. 2025. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571* (2025).

[163] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. 2016. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1995–2003.

[164] Zhixin Wang, Tianyi Zhou, Liming Liu, Ao Li, Jiarui Hu, Dian Yang, Jinlong Hou, Siyuan Feng, Yuan Cheng, and Yuan Qi. 2025. DistFlow: A Fully Distributed RL Framework for Scalable and Efficient LLM Post-Training. *arXiv preprint arXiv:2507.13833* (2025).

[165] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in neural information processing systems*, Vol. 35. 24824–24837.

[166] Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. 2024. CycleResearcher: Improving Automated Research via Automated Review. *arXiv preprint arXiv:2411.00816* (2024).

[167] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3 (1992), 229–256.

[168] Xuyang Wu, Jinming Nian, Ting-Ruen Wei, Zhiqiang Tao, Hsin-Tai Wu, and Yi Fang. 2025. Does Reasoning Introduce Bias? A Study of Social Bias Evaluation and Mitigation in LLM Reasoning. *arXiv preprint arXiv:2502.15361* (2025).

[169] Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, Wei He, Yiwen Ding, Shichun Liu, Xin Guo, Junzhe Wang, et al. 2024. Training large language models for reasoning through reverse curriculum reinforcement learning. *arXiv preprint arXiv:2402.05808* (2024).

[170] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering Large Pre-Trained Language Models to Follow Complex Instructions. In *International Conference on Representation Learning*, Vol. 12. 30745–30766.

[171] Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. 2025. Towards Large Reasoning Models: A Survey of Reinforced Reasoning with Large Language Models. *arXiv preprint arXiv:2501.09686* (2025).

[172] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. 2025. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215* (2025).

[173] Ran Xu, Yuchen Zhuang, Yishan Zhong, Yue Yu, Xiangru Tang, Hang Wu, May D. Wang, Peifeng Ruan, Donghan Yang, Tao Wang, Guanghua Xiao, Carl Yang, Yang Xie, and Wenqi Shi. 2025. MedAgentGym: Training LLM Agents for Code-Based Medical Reasoning at Scale. *arXiv preprint*

arXiv:2506.04405 (2025).

[174] Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719* (2024).

[175] Yifei Xu, Tusher Chakraborty, Srinagesh Sharma, Leonardo Nunes, Emre Kıcıman, Songwu Lu, and Ranveer Chandra. 2025. Direct Reasoning Optimization: LLMs Can Reward And Refine Their Own Reasoning for Open-Ended Tasks. *arXiv preprint arXiv:2506.13351* (2025).

[176] Yuyang Xu, Yi Cheng, Haochao Ying, Zhuoyun Du, Renjun Hu, Xing Shi, Wei Lin, and Jian Wu. 2025. SSPO: Self-traced Step-wise Preference Optimization for Process Supervision and Reasoning Compression. *arXiv preprint arXiv:2508.12604* (2025).

[177] Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945* (2025).

[178] Qihang Yan, Xinyu Zhang, Luming Guo, Qi Zhang, and Feifan Liu. 2025. RACE-Align: Retrieval-Augmented and Chain-of-Thought Enhanced Preference Alignment for Large Language Models. *arXiv preprint arXiv:2506.02726* (2025).

[179] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).

[180] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122* (2024).

[181] Songhua Yang, Hanjie Zhao, Senbin Zhu, Guangyu Zhou, Hongfei Xu, Yuxiang Jia, and Hongying Zan. 2023. Zhongjing: Enhancing the Chinese Medical Capabilities of Large Language Model through Expert Feedback and Real-world Multi-turn Dialogue. *arXiv preprint arXiv:2308.03549* (2023).

[182] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *arXiv preprint arXiv:2305.10601* (2023).

[183] Zijun Yao, Yantao Liu, Yanxu Chen, Jianhui Chen, Junfeng Fang, Lei Hou, Juanzi Li, and Tat-Seng Chua. 2025. Are Reasoning Models More Prone to Hallucination? *arXiv preprint arXiv:2505.23646* (2025).

[184] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. 2021. Mastering atari games with limited data. In *Advances in neural information processing systems*, Vol. 34. 25476–25488.

[185] Qiang Yi, Yangfan He, Jianhui Wang, Xinyuan Song, Shiyao Qian, Xinhang Yuan, Li Sun, Yi Xin, Jingqun Tang, Keqin Li, et al. 2025. Score: Story coherence and retrieval enhancement for ai narratives. *arXiv preprint arXiv:2503.23512* (2025).

[186] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476* (2025).

[187] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837* (2025).

[188] Jinwei Zeng, Chao Yu, Xinyi Yang, Wenxuan Ao, Qianyue Hao, Jian Yuan, Yong Li, Yu Wang, and Huazhong Yang. 2024. CityLight: A Universal Model for Coordinated Traffic Signal Control in City-scale Heterogeneous Intersections. *arXiv preprint arXiv:2406.02126* (2024).

[189] Dalong Zhang, Jun Xu, Jun Zhou, Lei Liang, Lin Yuan, Ling Zhong, Mengshu Sun, Peilong Zhao, QiWei Wang, Xiaorui Wang, Xinkai Du, YangYang Hou, Yu Ao, ZhaoYang Wang, Zhengke Gui, ZhiYing Yi, Zhongpu Bo, Haofen Wang, and Huajun Chen. 2025. KAG-Thinker: Interactive Thinking and Deep Reasoning in LLMs via Knowledge-Augmented Generation. *arXiv preprint arXiv:2506.17728* (2025).

[190] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. ReST-MCTS*: LLM Self-Training via Process Reward Guided Tree Search. *arXiv preprint arXiv:2406.03816* (2024).

[191] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems*, Vol. 37. 64735–64772.

[192] Kechi Zhang, Ge Li, Jia Li, Huangzhao Zhang, Jingjing Xu, Hao Zhu, Lecheng Wang, Jia Li, Yihong Dong, Jing Mai, Bin Gu, and Zhi Jin. 2025. Computational Thinking Reasoning in Large Language Models. *arXiv preprint arXiv:2506.02658* (2025).

[193] Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, Fangfu Liu, Xiang Xu, Jiaze Ma, Xuekai Zhu, Ermo Hua, Yihao Liu, Zonglin Li, Huayu Chen, Xiaoye Qu, Yafu Li, Weize Chen, Zhenzhao Yuan, Junqi Gao, Dong Li, Zhiyuan Ma, Ganqu Cui, Zhiyuan Liu, Biqing Qi, Ning Ding, and Bowen Zhou. 2025. A Survey of Reinforcement Learning for Large Reasoning Models. *arXiv preprint arXiv:2509.08827* (2025).

[194] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. 2024. Chain of preference optimization: Improving chain-of-thought reasoning in llms. In *Advances in Neural Information Processing Systems*, Vol. 37. 333–356.

[195] Xiaotian Zhang, Yuan Wang, Zhaopeng Feng, Ruizhe Chen, Zhijie Zhou, Yan Zhang, Hongxia Xu, Jian Wu, and Zuozhu Liu. 2025. Med-U1: Incentivizing Unified Medical Reasoning in LLMs via Large-scale Reinforcement Learning. *arXiv preprint arXiv:2506.12307* (2025).

[196] Yiyuan Zhang, Kaixiong Gong, Kaipeng Zhang, Hongsheng Li, Yu Qiao, Wanli Ouyang, and Xiangyu Yue. 2023. Meta-transformer: A unified framework for multimodal learning. *arXiv preprint arXiv:2307.10802* (2023).

[197] Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. 2025. No Free Lunch: Rethinking Internal Feedback for LLM Reasoning. *arXiv preprint arXiv:2506.17219* (2025).

[198] Zizhuo Zhang, Jianing Zhu, Xinmu Ge, Zihua Zhao, Zhanke Zhou, Xuan Li, Xiao Feng, Jiangchao Yao, and Bo Han. 2025. Co-Reward: Self-supervised Reinforcement Learning for Large Language Model Reasoning via Contrastive Agreement. *arXiv preprint arXiv:2508.00410* (2025).

[199] Keyu Zhao, Fengli Xu, and Yong Li. 2025. Reason-to-Recommend: Using Interaction-of-Thought Reasoning to Enhance LLM Recommendation. *arXiv preprint arXiv:2506.05069* (2025).

[200] Kai Zhao, Yanjun Zhao, Jiaming Song, Shien He, Lusheng Zhang, Qiang Zhang, and Tianjiao Li. 2025. SABER: Switchable and Balanced Training for Efficient LLM Reasoning. *arXiv preprint arXiv:2508.10026* (2025).

[201] Lulu Zhao, Weihao Zeng, Xiaofeng Shi, and Hua Zhou. 2024. CareBot: A Pioneering Full-Process Open-Source Medical Language Model. *arXiv preprint arXiv:2412.15236* (2024).

[202] Lulu Zhao, Weihao Zeng, Xiaofeng Shi, Hua Zhou, Donglin Hao, and Yonghua Lin. 2024. Aqulia-Med LLM: Pioneering Full-Process Open-Source Medical Language Models. *arXiv preprint arXiv:2406.12182* (2024).

[203] Xutong Zhao, Tengyu Xu, Xuewei Wang, Zhengxing Chen, Di Jin, Liang Tan, Yen-Ting, Zishun Yu, Zhuokai Zhao, Yun He, Sinong Wang, Han Fang, Sarath Chandar, and Chen Zhu. 2025. Boosting LLM Reasoning via Spontaneous Self-Correction. *arXiv preprint arXiv:2506.06923* (2025).

[204] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160* (2025).

[205] Yu Zheng, Qianyue Hao, Jingwei Wang, Changzheng Gao, Jinwei Chen, Depeng Jin, and Yong Li. 2024. A Survey of Machine Learning for Urban Decision Making: Applications in Planning, Transportation, and Healthcare. *Comput. Surveys* (2024).

[206] Yu Zheng, Yuming Lin, Liang Zhao, Tinghai Wu, Depeng Jin, and Yong Li. 2023. Spatial planning of urban communities via deep reinforcement learning. *Nature Computational Science* 3, 9 (2023), 748–762.

[207] Han Zhong, Yutong Yin, Shenao Zhang, Xiaojun Xu, Yuanxin Liu, Yifei Zuo, Zhihan Liu, Boyi Liu, Sirui Zheng, Hongyi Guo, Liwei Wang, Mingyi Hong, and Zhaoran Wang. 2025. BRiTE: Bootstrapping Reinforced Thinking Process to Enhance Language Model Reasoning. *arXiv preprint arXiv:2501.18858* (2025).

[208] Yinmin Zhong, Zili Zhang, Bingyang Wu, Shengyu Liu, Yukun Chen, Changyi Wan, Hanpeng Hu, Lei Xia, Ranchen Ming, Yibo Zhu, et al. 2025. Optimizing {RLHF} training for large language models with stage fusion. In *22nd USENIX Symposium on Networked Systems Design and Implementation (NSDI 25)*. 489–503.

[209] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, Vol. 36. 55006–55021.

[210] Banghua Zhu, Hiteshi Sharma, Felipe Vieira Frujeri, Shi Dong, Chenguang Zhu, Michael I. Jordan, and Jiantao Jiao. 2023. Fine-Tuning Language Models with Advantage-Induced Policy Alignment. *arXiv preprint arXiv:2306.02231* (2023).

[211] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593* (2019).