The Dawn of Thinking Machines: An Overview on Large Reasoning Models

Jaward Sesay

Beijing Institute of Technology

Abstract

In his mind-bending paper "Computing Machinery and Intelligence", Alan Turing famously posed the seminal question *"Can machines think?"* (Turing A. M, 1950). This query has since spurred decades of research and development on the reasoning capabilities of artificial intelligence (AI) systems. Traditional Large Language Models (LLMs), primarily based on auto-regressive next-token prediction, were once deemed incapable of human-level reasoning. However, recent breakthroughs in Reinforcement Learning (RL) and inference-time scaling have proved otherwise, endowing LLMs with advance emergent reasoning capabilities. This paradigm shift in learning has led to the development of Large Reasoning Models (LRMs), models that can think or reason through a given prompt before answering. Synonymous to how we humans solve complex problems, LRMs can "reason" systematically through intricate tasks, leveraging a reinforced chain-of-thought with "think" tokens to generate coherent and accurate responses. We first got to see this level of reasoning in proprietary models like OpenAI's o-series, Google's Gemini Flash Thinking and most recently Grok 3 and Claude 3.7 Sonnet, with their underlying research remaining largely undisclosed. In contrast, open-weight models like Magistral, DeepSeek-R1/R1-Zero and QwQ-32B have provided unprecedented insights into the techniques that empower LLMs to reason. This paper examines current state-of-the-art research behind reasoning models. First, we cover the fundamental building blocks of LLMs, their initial shortcomings in reasoning, then we explore underlying algorithms, architectures and training methods used in developing LLMs into LRMs. Finally, we analyze noteworthy opensource projects, discuss limitations and future potential implications of reasoning models as we journey towards Artificial General Intelligence (AGI).

1 Introduction

The enduring quest to create machines that can think has long captured the minds of research scientists. Recent advancements in artificial intelligence (AI), mainly driven by breakthroughs in deep learning (LeCun et al., 2015) have brought us closer than ever to realizing "thinking machines". The field is witnessing groundbreaking progress in foundational research covering architectures, algorithms, data, compute and training techniques. A pivotal moment in the field was the introduction of the transformer architecture (Vaswani et al., 2017), unlike their predecessors (Hochreiter et al., 1997a; Cho et al., 2014), transformers leverage a self-attention mechanism to process entire sequences in

Preprint under review

parallel, enabling today's widespread use of Large Language Models (LLMs). LLMs have demonstrated remarkable proficiency across a wide range of intellectual tasks, including generating coherent and contextually relevant texts, translating languages, summarizing lengthy documents, and answering complex questions with impressive accuracy when given access to sources like the web (Lewis et al., 2020; Touvron et al., 2023). Over time, they evolved into multimodal models (GPT4o, OpenAI, 2024; Gemini, Gemini Team Google, 2023), capable of processing and generating not just text, but also images, audio, and video, further expanding use-cases across diverse domains.

Despite these remarkable capabilities, LLMs particularly lacked a significant indicator of human-level intelligence—the ability to reason and plan systematically through complex problems. Early auto-regressive LLMs, struggled with tasks requiring multi-step reasoning and causal inference since their training focused primarily on next-token prediction with barely any reasoning incentive. To address this, initial research explored several innovative methods that combined supervised and unsupervised learning. This includes generative pre-training on diverse corpora, followed by discriminative Supervised Fine-tuning (SFT) to enhance reasoning (Radford et al., 2018; Devlin et al., 2018). Subsequent post-training methods introduced Reinforcement Learning (RL) after SFT, for which Reinforcement Learning from Human Feedback (RLHF) has become the standard de-facto method widely adopted today (Ouyang et al., 2022). However, RLHF optimizes model performance on a lossy simulation of human preferences, contributing to limitations in scaling dense models (Chan et al., 2024). This led to experiments with scaling at Test-Time via Reinforcement Learning from AI Feedback RLAIF (Lee et al., 2023). In RLAIF, response evaluation is done by another model or the model being fine-tuned (DeepSeek AI, 2025a). Other inference-time scaling methods include prompt-based (Wei et al., 2022) and search-based (Yao et al., 2023; Zhao et al., 2025). While effective, these methods remain insufficient for tasks demanding advance reasoning, underscoring the need for architectural modifications and alternative learning algorithms.

To solve this inherent reasoning limitation, researchers embarked on developing a new generation of language models explicitly trained for robust reasoning, called Large Reasoning Models (LRMs). These models are trained to "think" or "reason" through complex problems before generating a response. This paradigm-shift in learning marked a pivotal move towards more reasoning-oriented reinforcement learning (RL) methods. Initially, RL in language modeling, was primarily used to enhance generalization and alignment in model performance. RLHF emerged as the dominant approach, leading to the introduction of a variety of policy optimization algorithms. Notable among these include Proximal Policy Optimization (PPO, Schulman et al., 2017a), Direct Preference Optimization (DPO, Rafailov et al., 2023), Group Relative Policy Optimization (GRPO, DeepSeek AI et al., 2024). More recent contributions include Debiased Group Relative Policy Optimization (DR. GRPO, Sea AI Lab, 2025) and Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO, ByteDance, 2025). Today, both proprietary and open-source models utilize these optimization methods, with open-source playing a key role in shaping their development.

In this paper, we explore recent research progress in the development of reasoning models, examining their evolution from traditional LLMs into LRMs. We cover architectural modifications, training methods and test-time scaling techniques that collectively enhance reasoning capabilities, offering insights into the broader goal of achieving AGI.

2 Background

2.1 “Can Machines Think?”

The question *"Can machines think?"* has been a subject of scientific inquiry long before the advent of modern computers. First echoed in Alan Turing’s 1950 seminal paper, "Computing Machinery and Intelligence", Turing recognized the inherent difficulties in defining "thinking" through subjective or metaphysical view points, He instead proposed an empirical alternative: the Imitation Game, fameouly known today as the Turing Test. This test posits a scenario where a human evaluator engages in text-based conversations with both a human and a machine, unaware of their identities. If the evaluator cannot reliably distinguish the machine from the human based solely on the conversational responses, Turing argued that we should, for all practical purposes, consider the machine to be "thinking." The brilliance of Turing’s approach lies in its operational definition of intelligence, shifting the focus from the internal workings of a system to its observable performance in a specifically designed task. He challenged the prevailing anthropocentric view of intelligence, suggesting that if a machine can convincingly mimic human-level conversation, the underlying mechanism, whether biological or silicon-based, becomes less relevant.

The introduction of large language models has reignited a new look into this age-old question, triggering diverse perspectives from prominent AI scholars. Leading AI research scientists like Geoffrey Hinton have argued that current LLMs, despite their limitations, exhibit some form of reasoning, pointing to the methods through which they predict and generate coherent text (Hinton, 2024. Romanes Lectures). Conversely, Yann LeCun argues that the auto-regressive nature of LLMs precludes genuine intelligence, emphasizing their lack of hierarchical planning and reasoning—key attributes he deems essential towards achieving AGI (LeCun, 2024 UW ECE Lecture). On the other hand Ilya Sutskever, a pioneer of LLMs, argues that with enough compute, data and increase in model size, LLMs can be scaled to reason in ways not explicitly trained to—emphasizing on their emergent reasoning capabilities (Sutskever, 2024 NeurIPS Talk). These varying perspectives underscore the ongoing debate and lack of consensus surrounding "thinking machines" in the era of LLMs. While current trending large reasoning models demonstrate impressive reasoning prowess, the question of whether they truly “reason” or "think" remains open and heavily debated.

2.2 Understanding Large Language Models

At their core, large language models are advanced neural networks trained to predict the next word, or more precisely, the next token in a sequence of texts (Radford et al., 2018b; Brown et al., 2020). This seemingly simple objective, when scaled up to models with billions or trillions of parameters and trained on massive, diverse datasets, can unlock remarkable emergent intellectual capabilities. In this section we will briefly explore the inner workings of LLMs covering architectures, data, training, compute and their scaling laws.

2.2.1 Architectures

LLMs are primarily based on the transformer architecture, which is a paradigm-shift from traditional language modeling architectures like Recurrent Neural Networks (RNNs) and LSTMs (Hochreiter et al., 1997b). Transformers, unlike their predecessors, enabled parallel processing of input tokens, effectively capturing long-range dependencies and handling

variable-length sequences. This allows them to discern relationships between words or sub-words regardless of distance, crucial for natural language understanding. This mechanism, termed Self-Attention, dynamically weighs the importance of each word or token in the input context when processing every other word or token. Stacking multiple self-attention layers results in Multi-Head Attention (MHA) – which allows for more robust representation learning by enabling the model to reach diverse parts of the input simultaneously.

However, as models increase in size, conventional Multi-Head Attention can become computationally intensive, largely due to the growing Key-Value (KV) cache from computing attention scores between tokens. This cache which stores KV vectors for each token, leads to increased memory usage and slower inference. To solve these challenges, several innovative iterations of the attention block have been introduced. Leading modifications include Multi-Query Attention (MQA; Shazeer et al., 2019) which minimizes the KV cache by employing a single KV head for all query heads, while Group-Query Attention (GQA; Ainslie et al., 2023) strikes a balance by grouping query heads to share one KV head per group. Additionally, Flash Attention (Dao et al., 2022) further accelerates computation by optimizing memory access for longer sequences, and more recently Multi-Latent Attention (MLA; Deepseek AI et al., 2024) which leverages learnable latent representations to reduce the computational load. Beyond attention, the adoption of deep Mixture of Experts (MoEs; Eigen et al., 2014) in frontier models has also proved tranformative. MoEs enhance compute efficiency by routing input tokens to specialized sub-networks or "experts", activating only a subset of parameters per task rather than the entire model. These architectural advancements have been instrumental in the progress of larger and more capable LLMs, including recent large reasoning models.

2.2.2 Data

The remarkable capabilities of LLMs are intrinsically linked to the vast quantity and quality of datasets they are trained on, underscoring the critical importance of data curation. This process involves not only amassing large quantities of text, images and videos from diverse sources (mainly from the internet) but also meticulously preprocessing this data for efficient use. Depending on the source, the data undergoes different preprocessing stages. Three major stages include:

1. **Data Filtering and Cleansing:** This initial stage focuses on removing unsafe, low-quality content from the dataset. Filters are designed to eliminate irrelevant or harmful information, ensuring that the data used for training is both safe and pertinent. Following this, cleansing processes further refine the data by correcting errors, standardizing formats, and handling missing values, thereby enhancing the overall quality and consistency of the dataset.
2. **De-duplication**: Deduplication involves identifying and removing duplicate texts from the dataset. Duplicates can skew the model's training by causing it to overfit or be biased towards certain patterns. By removing both exact duplicates and near-identical content, the training data becomes more diverse and representative, which helps the model generalize better when applied to unseen data
3. **Tokenization**: This process involves splitting raw text into smaller units called tokens, which can be words, subwords, or characters. Tokenization is essential for converting unstructured text into a structured format that models can process.

However, there is growing concern over the potential exhaustion of publicly available data, with estimates predicting a saturation point for training data derived from human-generated corpus (Villalobos et al., 2022). This has led research into alternative data sourcing approaches, like synthetic data generation (Amin et al., 2024) and data augmentation (Ding et al., 2024). On the other hand, there are ongoing efforts to create data with reasoning traces (OpenThoughts, Bespoke Labs, 2024) specifically for training LRMs.

2.2.3 Training

LLMs are generally trained using a mix of supervised and unsupervised learning methods, typically comprising of two main stages: Pre-Training and Post-Training. During pre-training, the model ingests vast corpora of text data to learn statistical language patterns through next-token prediction via unsupervised learning (Radford et al., 2018c). This phase requires optimizing billions of parameters on large-scale datasets, often requiring massive computational resources. The goal of pre-training is to develop a general understanding of language, spanning grammar, word associations, and contextual relationships. At this stage model learns to predict the next word or token via the self-attention mechanism, thereby capturing a broad spectrum of linguistic information without specific task-oriented fine-tuning. This stage mainly involves meticulously preprocessing data in a way that facilitates the learning process for succeeding stages, such as tokenization, normalization, and sequence padding. To cap compute costs and improve efficiency, architectural modifications have been made to the transformer, such as Sparse Transformers (Child et al., 2019) and Mixture-of-Experts (Shazeer et al., 2017). Additionally, numerous variants of attention mechanisms (e.g. MLA, MQA and Flash attention) along with distillation techniques (Hinton et al., 2015) have been employed to achieve high performance with fewer parameters.

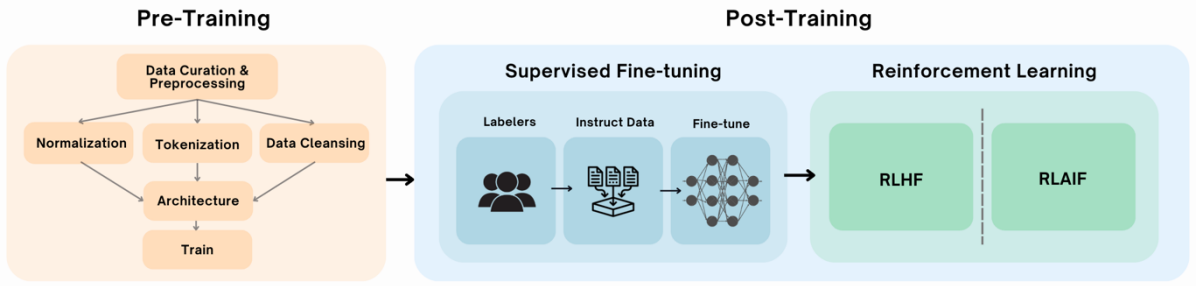

Figure 1 Overview of LLM Training pipeline, showing major stages in both pre- and post-training.

After pre-training, the pre-trained model enters a post-training stage where performance is enhanced through Supervised Fine-tuning (SFT) and Reinforcement Learning (RL). During the SFT phase, the model is trained on a range of task-specific datasets, typically composed of question-answer pairs tailored to desired preferences. Following SFT, reinforcement learning is employed to enhance alignment and specialization. During this phase, the model learns from feedback signals (often reward-based), adjusting its responses to maximize performance on predefined objectives (Schulman et al., 2017b). This involves utilizing gradient-based policy optimization algorithms (e.g. PPO, DPO, GRPO etc) to refine the model's behavior by iteratively updating its parameters in the direction that improves expected rewards. A widely adopted RL-based method is RLHF, which relies on preference-based feedback from human annotators to steer model behavior (Stiennon et al., 2020). Recently, RLAIF has emerged as a promising alternative, leveraging AI as critiques to enhance response quality with minimal or no human supervision (DeepSeek AI et al., 2025b)

2.2.4 Scaling Laws

The scaling laws are empirical observations used to quantify how model performance scales with its parameters, training data size, and compute (Kaplan et al., 2020; Hoffmann et al., 2022; Bahri et al., 2021). Initial work by Kaplan et al established that performance on language modeling tasks adheres to a power-law relationship across these dimensions, providing a roadmap for scaling model capabilities predictably. Hoffmann et al. built on this by emphasizing the need for a balanced ratio of model parameters to training data, demonstrating that optimal configurations can enhance performance while reducing compute costs. These findings have fueled the rise of increasingly large models, driving significant improvements in language modeling, machine translation, and complex reasoning (Wei et al., 2022b). However, scaling dense models with billions or trillions of parameters incurs substantial computational overhead and faces diminishing returns at extreme sizes (Strubell et al., 2019). This has spurred research into efficient alternatives, such as sparse architectures, mixture-of-experts models, and improved data curation techniques Additionally, breakthroughs with test-time scaling have enabled models to tackle intricate tasks through inference-time reasoning, bypassing the need for resource-intensive retraining. These advances enhance efficiency and broaden access to high-performing models, addressing practical concerns in large-scale AI development.

3 Reinforced-Reasoning LLMs: Large Reasoning Models

Fundamentally, auto-regressive LLMs are trained to predict the next token in a sequence using the attention mechanism. This mechanism computes the attention scores between tokens, enabling the model to capture contextual dependencies effectively. However, this strength does not inherently translate into deep reasoning or alignment with human intent. Early efforts explored Test-Time scaling methods. For instance, Instruction Tuning was utilized to better align models with human intent, while Chain-of-Thought prompting facilitated multi-step rudimentary reasoning, and Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) helped mitigate hallucinations. Despite their benefits, these techniques often fell short of instilling advanced reasoning capabilities. Recent breakthroughs with reinforcement learning (RL) methods have begun to bridge this gap, offering promising avenues to enhance deep reasoning in LLMs. Techniques like RLHF were developed to adjust outputs according to human preferences, primarily optimizing for perceived helpfulness rather than complex reasoning. To address these limitations, several optimization techniques have been introduced PPO (Schulman et al., 2017c), DPO (Rafailov et al., 2023b) and recently GRPO (DeepSeek AI, 2025) introduce reward-based objectives that explicitly encourage structured reasoning. Generally, these optimization methods share the same objective which is to improve model perfomance through a policy π_θ that maximizes the expected reward $J(\pi_\theta)$:

$$J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}\gamma^t R_t\right]$$

Where τ is a trajectory of model outputs, R_t is the reward at time step t, and γ is the discount factor. These methods guide Large Reasoning Models (LRMs) toward systematic problem decomposition, structured inference, and multi-step decision-making, transitioning from mere pattern recognition to models that demonstrate emergent reasoning capabilities.

3.1 Pioneering Reinforcement Learning Methods for Language Modeling

The inception of RL-based methods in language modeling traces back to foundational contributions addressing inherent limitations in Maximum Likelihood Estimation (MLE), particularly exposure bias—the discrepancy between training on ground-truth tokens and inference on the model's own predictions (Ranzato et al., 2015). This led researchers to explore reinforcement learning as an alternative, where sequence-level rewards guided model optimization rather than token-level likelihoods. The first major contribution introduced REINFORCE (Williams, 1992), a fundamental gradient-based policy method that optimizes the parameters θ of a stochastic policy $\pi_\theta(a|s)$ in the direction of the gradient of expected reward $J(\theta)$, using Monte Carlo sampling:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a|s) \cdot R_t\right]$$

Where $R_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$ is the discounted reward return at time step t. REINFORCE introduced a framework for training policies with sequence-level rewards but suffered from high variance, hindering optimization efficiency. To mitigate this, Ranzato et al., proposed MIXER, blending REINFORCE with cross-entropy loss to optimize both token-level accuracy and sequence-level metrics like BLEU scores. Rennie et al. (2017) advanced this with Self-Critical Sequence Training (SCST), reducing variance by using greedy-decoded outputs as baselines, enhancing stability for tasks like image captioning. Concurrently, Mnih et al. (2016) introduced A2C/A3C, employing a critic network to compute advantages, stabilizing learning and enabling parallel training, while Shen et al. (2016) developed Minimum Risk Training (MRT), optimizing sequence-level loss via risk minimization without policy gradients. Together, these methods addressed variance, exposure bias, and stability, laying the groundwork for modern RL approaches like RLHF, RLAIF, and techniques such as PPO, DPO, and GRPO, now integral to Large Reasoning Models (LRMs).

3.2 LRM Reasoning as a Classical Reinforcement Learning Problem

At a classical level in reinforcement learning, an agent learns to take actions within an environment to maximize a cumulative reward signal through the process of trial and error. Rooted in behavioral psychology and optimal control theory, RL is considered the closest artificial form of learning that mimics how humans and animals learn (Sutton and Barto, 1998). Given the auto-regressive nature of LLMs, their next-token prediction task can be modeled as a sequential decision-making problem in a Markov Decision Process (MDP) (Bellman et al., 1957). In this framework, the LLM serves as the agent, tasked with generating a sequence of tokens in response to an input prompt, which constitutes the environment. States are represented by the current sequence of tokens generated thus far, capturing the evolving context of the output. Actions correspond to the selection of the next token from the model's vocabulary, each choice incrementally shaping the response.

Under common decoding schemes such as greedy decoding or sampling with temperature, the transition dynamics may be treated as deterministic or stochastic, enabling application of standard reward algorithms. Rewards are defined by task-specific criteria, such as the accuracy of a response, response format, the logical coherence of a reasoning chain, often derived from external feedback or predefined metrics.

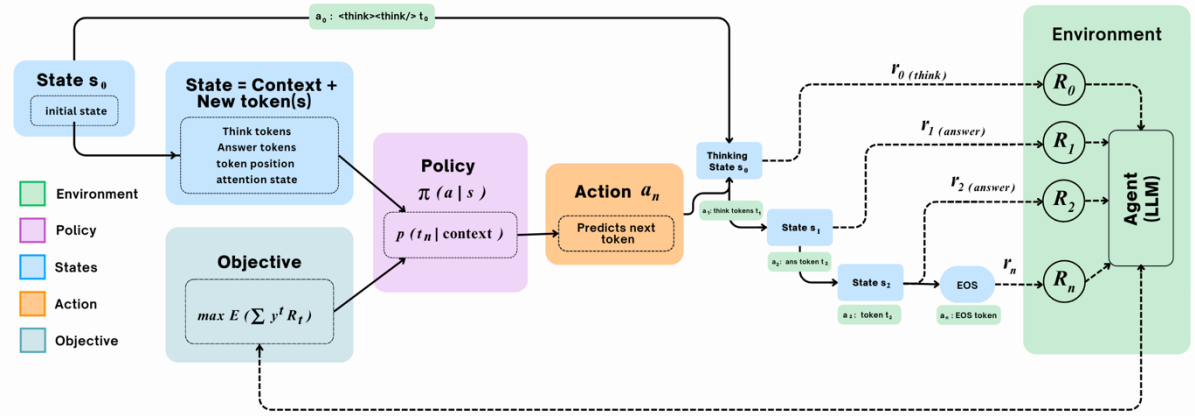

Figure 2 Modeling LRM reasoning as a classical RL problem in a Markov Decision Process. (MDP).

Thus the LLM's objective would be to derive an optimal policy that specifies the the probability distribution over next tokens given the current state, maximizing the expected cumulative reward.

3.3 Reasoning-Oriented Supervised Fine-tuning

Supervised fine-tuning (SFT) plays a pivotal role in enhancing the reasoning capabilities of large language models (LLMs), adapting their next-token prediction proficiency into structured inference and multi-step problem-solving skills essential for Large Reasoning Models (LRMs). Early SFT efforts leaned on datasets like SQUAD (Rajpurkar et al., 2016), fine-tuning models to extract answers from context via cross-entropy loss, yielding strong comprehension but weak multi-step reasoning due to a focus on patterns over logic. This evolved into fine-tuning with explanation-rich datasets such as e-SNLI (Camburu et al., 2018) and CommonsenseQA (Talmor et al., 2019), where natural language rationales accompanied answers, encouraging models to associate outcomes with justificatory reasoning, though effectiveness depended on explanation quality and risked memorization rather than generalization. Self-rationalization techniques (Narang et al., 2020) further advanced this by training models to generate their own reasoning steps, improving transparency but requiring robust datasets to avoid overfitting. Parameter-efficient fine-tuning (PEFT) marked a leap forward, with LoRA (Hu et al., 2021b) adapting LLMs for reasoning by updating low-rank weight matrices, and QLoRA (Dettmers et al., 2023b) quantizing models to 4-bit precision for memory-efficient tuning, both preserving reasoning gains with reduced resource demands.

Recently, more reasoning-oriented SFT methods have emerged, for instance, ReFT (Reasoning with REinforced Fine-Tuning) initializes models with SFT on chain-of-thought (CoT) data and then applies online RL via PPO to sample multiple reasoning paths per math problem, notably improving generalization over standard SFT (Luong et al., 2024). Similarly, Reinforced Functional Token Tuning (RFTT) which incorporates learnable functional tokens into the model's vocabulary, facilitating diverse reasoning behaviors (Zhang et al., 2025). Moreover, innovative reasoning-based distillation techniques like Fault-Aware DistIllation via Peer-Review (FAIR) (Li et al., 2024) push further, distilling CoT reasoning into compact models with remarkable precision, and Socratic CoT (Shridhar et al., 2023), seen in DeepSeek R1 distilled models (e.g DeepSeek-R1-Distill-Qwen-32B), optimizes reasoning tasks with minimal parameter tweaks, slashing resource demands. These recent SFT strides complements RL techniques that address past pitfalls, leading LRMs toward systematic, human-aligned reasoning with unparalleled depth.

3.4 Scaling Reasoning with Reinforcement Learning

Reinforcement Learning (RL) has proven to be the driving force behind major breakthroughs in AI, from powering the brain of AlphaGO (Silver et al., 2016) to enabling advance emergent reasoning capabilities in language models (OpenAI, 2024; Gemini Team Google, 2024; Wang et al., 2022). Attempts to boost reasoning in LLMs have proven successful with RL-based methods, leading to researchers exploring reward-based modeling with various policy optimization techniques. In this section we will discuss briefly the various types of reward modeling used in the two leading RL-based methods: Reinforcement Learning from Human Feedback (RLHF) and Reinforcement Learning from AI Feedback (RLAIF), then we will explore their underlying policy optimization algorithms (e.g. PPO, DPO, GRPO etc).

3.4.1 Reward Modeling

Reward modeling forms the foundation of modern RL methods used in language modeling. These methods (e.g RLHF and RLAIF) leverage feedback signals that help guide model behavior through preference ranking or rating of human-annotated or LLM-generated responses. Two commonly used forms of preference are: ***Pairwise Preference*** and ***Response Ranking Preference***. In Pairwise Preference, human annotators compare two responses y^+ and y^-, then select the preferred one. While in Response Ranking, the responses are arrange in order of preference. The reward model is then trained to predict these preferences by optimizing the probability of preferred responses using the Bradley-Terry model of preferences:

$$P(y^+ \succ y^- \mid x;\theta) = \frac{e^{R_\theta(y^+)}}{e^{R_\theta(y^+)} + e^{R_\theta(y^-)}}$$

where $R_\theta(y^+)$ and $R_\theta(y^-)$ are the respective reward scores assigned to each response. The model is updated using a cross-entropy loss function to maximize alignment with human judgments:

$$\mathcal{L}_{BT}(\theta) = -\sum_{(y^+, y^-)} \log P(y^+ \succ y^- \mid x;\theta)$$

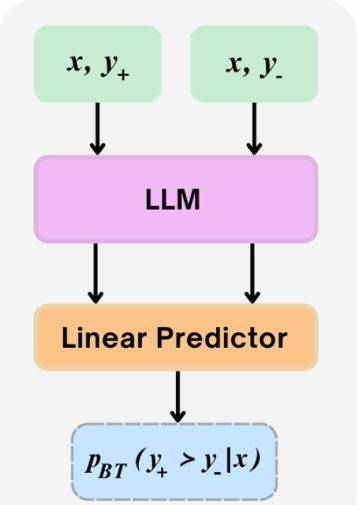

Figure 3 Illustration of a Bradley-Terry model

Several variations of reward modeling exist, the above example is called ***Explicit Reward Modeling*** – wherein human evaluators assign direct, predefined scores to response samples based on specific preferences (Lewkowycz et al., 2022). In contrast, ***Implicit Reward Modeling*** infers quality indirectly from contextual signals, such as user engagement metrics or system-level interactions. This approach reduces the burden of extensive human annotation by allowing AI-generated feedback to influence the reward signal. For example, RLAIF leverages implicit feedback by assigning higher rewards to responses that are contextually accurate, without needing explicit numerical scores. This method has proven effective in scaling the reward modeling process, maintaining alignment while reducing annotation overhead (Bai et al., 2022; DeepSeek AI, 2025c***).*** Recent developments further introduced ***Outcome Reward Modeling***—which focuses solely on the final result of a

reasoning process, rewarding the correctness of the answer regardless of the steps taken, and ***Process Reward Modeling*** emphasizes the quality of the reasoning steps themselves, rewarding coherent and logical intermediate outputs, for example Chain-of-Thought (CoT) prompting (Wei et al., 2022b).

Additionally, several hybrid approaches have also emerged, combining explicit, implicit, outcome, and process reward modeling. For instance, ***Multi-objective Reward Modeling*** optimizes both correctness and reasoning by dynamically weighting reward signals, as seen in frameworks like ArmoRM which uses a mixture-of-experts to adaptively combine objectives based on context (Wang et al., 2024). This dynamic weighting allows for real-time prioritization of competing objectives, enhancing overall performance and adaptability in complex environments.

3.4.2 Policy Optimization

The primary objective of policy optimization algorithms is to find the optimal policy that maximizes the expected cumulative reward in a given environment. In the context of LLMs, this involves tuning the model's parameters to generate responses that align with desired outcomes, like advance reasoning or other alignment objectives. Leading policy optimization methods include:

Proximal Policy Optimization (PPO). Introduced by Schulman et al. (2017), PPO is a widely adopted policy optimization algorithm used in RL-based post-training, valued for its simplicity and compute efficiency over then exisiting methods like Trust Region Policy Optimization (TRPO; Schulman et al., 2015). In RLHF, PPO optimizes policies against reward signals from human feedback while maintaining stable updates. It achieves this through a clipped surrogate objective that prevents excessive deviations from the old policy for every update step. Formally, PPO maximizes the following objective:

$$L^{CLIP}(\theta) = \widehat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\, \hat{A}_t \right) \right],$$

where $r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$ is the probability ratio, $\hat{A}_t$ is the advantage estimate, ϵ (usually 0.2) sets the clip range, thus $clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\, \hat{A}_t$ modifies the surrogate objective by clipping the probability ratio.

Direct Preference Optimization (DPO). DPO offers an efficient yet much simpler alternative to PPO, by directly optimizing the policy from pairwise preferences, eliminating the need for explicit reward sampling (Rafailov et al. 2023b). While prior RLHF methods learn a reward then optimize it via RL, DPO "secretely" uses the model being trained as the reward model to learn the policy with a single maximum likelihood objective:

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l)} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right],$$

where x is the input, y_w and y_l are a pair of responses, β is the scaling factor and π_{ref} as reference policy.

Group Relative Policy Optimization (GRPO). GRPO (Deepseek AI, 2024b) is a variant of PPO that eliminates the need for an explicit value model. As an online learning method, its objective is to maximize the advantage $\hat{A}_{i,t} = \frac{r_i - \text{mean}(r)}{\text{std}(r)}$ of generated completions by computing relative group rewards while preserving policy updates to achieve the following surrogate objective:

$$\mathcal{L}_{GRPO}(\theta) = -\frac{1}{\mathrm{G}}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{i=0}^{|o_i|}\left[\min\left(\frac{\pi_\theta\left(o_{i,t} \mid q, o_{i,<t}\right)}{\left[\pi_{\theta_{old}}\left(o_{i,t} \mid q, o_{i,<t}\right)\right]_{\text{no grad}}}\hat{A}_{i,t}\right) - \beta\mathbb{D}_{KL}[\pi_\theta||\,\pi_{\text{ref}}]\right]$$

Thus for multiple updates the loss becomes:

$$\mathcal{L}_{GRPO}(\theta) = -\frac{1}{\mathrm{G}}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{i=0}^{|o_i|}\left[\min\left(\frac{\pi_\theta\left(o_{i,t} \mid q, o_{i,<t}\right)}{\pi_{\theta_{old}}\left(o_{i,t} \mid q, o_{i,<t}\right)}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta\left(o_{i,t} \mid q, o_{i,<t}\right)}{\pi_{\theta_{old}}\left(o_{i,t} \mid q, o_{i,<t}\right)}, 1 - \epsilon, 1 + \epsilon\right)\hat{A}_{i,t}\right) - \beta\mathbb{D}_{KL}[\pi_\theta||\,\pi_{\text{ref}}]\right]$$

GRPO, recently established itself as a state-of-the-art optimization algorithm by eliminating the need for a critic network and instead computing advantages through group-based relative comparisons of multiple sampled responses. This significantly reduced compute overhead, improved training stability, and enabled emergent multi-step reasoning capabilities we see in frontier reasoning models.

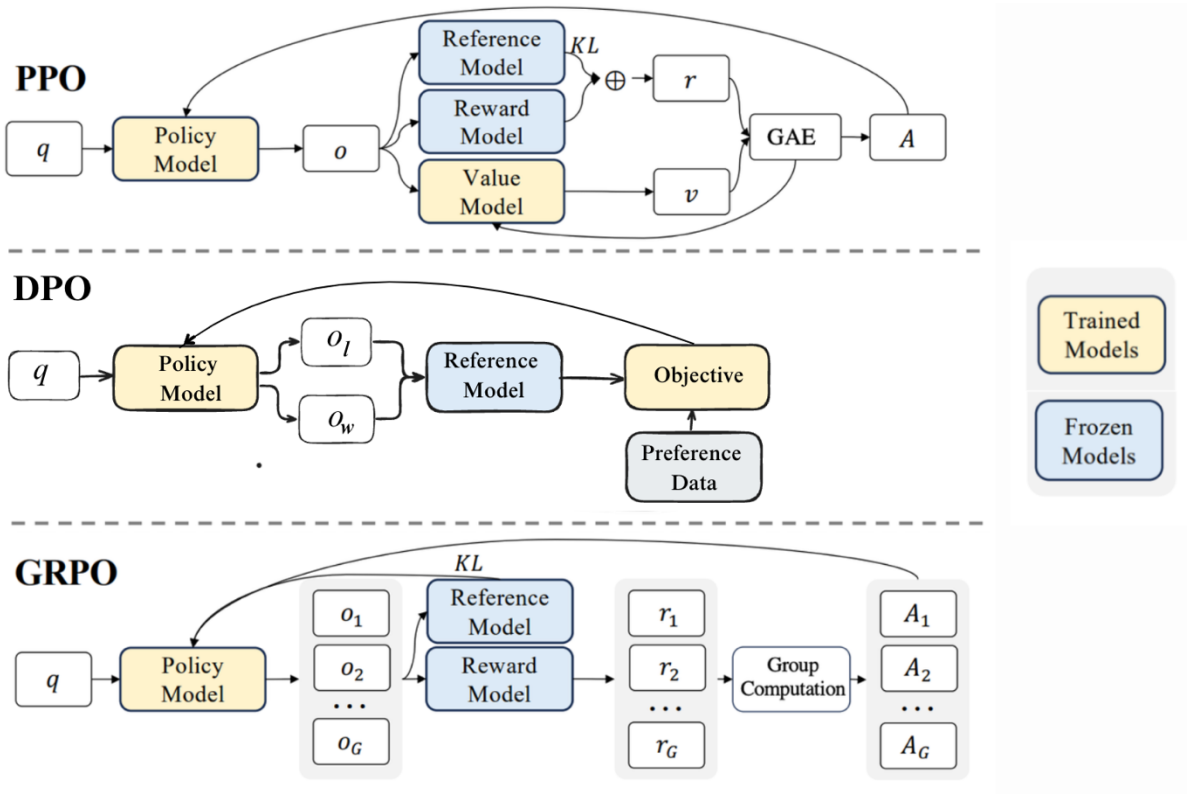

Figure 4 Comparing leading reward-based optimization methods for language modeling. PPO and GRPO flowcharts are from DeepSeekMath (DeepSeek AI, 2024)

Debiased Group Relative Policy Optimization (DR. GRPO). DR. GRPO addresses a key limitation of GRPO, which is an optimization bias towards longer response length. Specifically the authors claimed GRPO can inadvertently favor longer outputs (especially when they are incorrect), inflating verbosity without improving reasoning accuracy (Sea AI Lab, 2025b). DR. GRPO mitigates this this by debiasing the reward signal to remove length-related advantages, using a length-conditioned normalization and gradient correction that ensures the policy values accuracy over verbosity. This results in generation of responses that are concise yet accurate, improving both factual precision and reasoning efficiency across varying task lengths. This was eveident in a DR. GRPO trained 7B model, showing significant improvements in math reasoning tasks, achieving 43.3% accuracy on AIME.

Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO). DAPO builds on earlier methods by incorporating four innovative techniques: Clip-Higher (decoupling clipping ranges to prevent entropy collapse), Dynamic Sampling (filtering redundant questions for diverse batches), Token-Level Loss (averaging loss by response length for stability), and Overlong Responses Punishment (penalizing excessive verbosity). These features collectively enhance training efficiency and reasoning performance. DAPO's open-source implementation, including code, datasets (e.g., DAPO-Math-17K), and the verl framework, sets it apart by enabling reproducibility—a rare feature in RL research. When applied to the Qwen2.5-32B model, DAPO achieved 50 points on AIME 2024, surpassing DeepSeek-R1-Zero-Qwen-32B with 50% fewer training steps. Its Clip-Higher technique, for example, adjusts the upper clipping range to 0.28 (from PPO's 0.2), preserving token diversity, while Dynamic Sampling reduces training time by focusing on challenging inputs. These advancements make DAPO a state-of-the-art choice for large-scale LLM training (ByteDance, 2025).

3.4.3 SFT-Free Post-Training via Direct RL

DeepSeek-R1-Zero recently pioneered an SFT-free post-training method—one in which a pre-trained model scales in reasoning through direct reinforcement learning, without undergoing any prior supervised fine-tuning stage. In contrast to conventional post-training methods, that often require SFT, R1-Zero training applies RL (GRPO) directly to the base model, sampling multiple chain-of-thought trajectories per prompt and optimizing a clipped RL objective constrained by KL divergence to ensure stability (Shao et al., 2024). In this framework, rewards are deterministically assigned based on rule-based accuracy scoring, with explicit <think></think> tokens enforcing a structured reasoning format. This design draws inspiration from Satori's Chain of Action Thought tuning and self-improvement stages (Shen et al., 2025). Notably, this SFT-free RL approach yields emergent reasoning behaviors, demonstrating that appropriately incentivized policy gradients can foster robust multi-step reasoning without prior supervised fine-tuning (Tang et al., 2025). While initial iterations faced challenges such as reduced readability and occasional token mixing, subsequent refinements incorporating minimal cold-start data have effectively mitigated these issues. One of earliest SFT-free direct RL post-training attempt, was Reinforced Self-Training (ReST) which utilized offline RL to improve model performance without human feedback (Gulcehre et al., 2023). More recent developments include ReST-MCTS, which integrates process reward guidance with tree search MCTS for collecting higher-quality reasoning traces (Zhang et al., 2025). These developments collectively highlight the growing interest in SFT-free RL strategies for advancing large language model capabilities.

3.4.4 RLHF vs RLAIF

RLHF and RLAIF are two distinct post-training RL-based methods used in enhancing LLM reasoning and alignment, each with unique strengths. For instance, RLHF relies on human feedback to evaluate outputs, often using discriminative pairwise preference data in training the reward model. This method excels in aligning LLMs with subjective goals, like ethical reasoning or conversational nuance by emphasizing process-based rewards, such as scoring logical coherence in Chain-of-Thought (CoT). InstructGPT was the first known RLHF'ed LLM, leveraging policy gradient-based rewards to refine accuracy and instill human preferences via PPO (Ouyang et al., 2022b; Schulman et al., 2017c). However, its dependence on human effort limits scalability, introducing costs and potential lossy

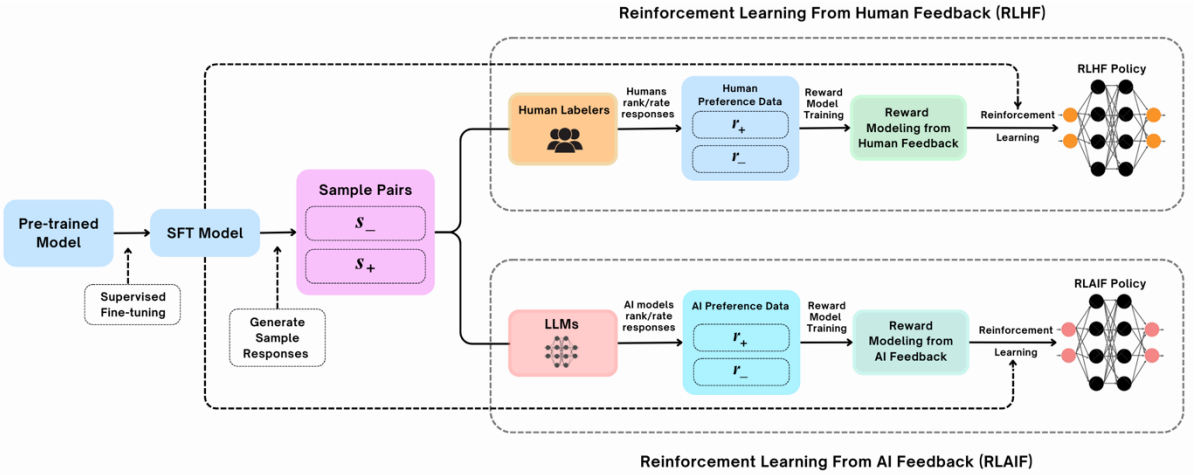

Figure 5 RLHF vs RLAIF

inconsistencies across desired preferences. In contrast, RLAIF substitutes human feedback with AI-generated feedback, wherein is a pretrained "critic" LLM assesses responses, generating implicit rewards (e.g., sentiment, fluency, engagement) or explicit scores (e.g., factual correctness) (Bai et al., 2022b). RLAIF often adopts outcome-based rewards, optimizing end-results over intermediate steps, and pairs with efficient methods like DPO or GRPO (Rafailov et al., 2023; DeepSeek AI et al., 2024). For example, DeepSeek-R1-Zero was trained from scratch with RLAIF, matching. Yet, RLAIF risks inheriting biases from its AI critic or overlooking human subtleties, making it less suited for tasks requiring deep contextual judgment compared to RLHF.

The trade-offs between RLHF and RLAIF hinge on the source, quality and scalability of the feedback used to train the reward model that subsequently helps align the language model. RLHF relies on direct human preference judgments, which are considered the gold standard for capturing nuanced human values but are expensive, time-consuming, potentially inconsistent, and difficult to scale. Conversely, RLAIF substitutes AI-generated feedback, often leveraging a powerful pre-trained model to act as the judge, offering significant advantages in terms of cost, speed, and scalability, allowing for potentially vast datasets. However, the primary drawback of RLAIF, lies in its reliance on a proxy for human judgment that is contingent upon the quality, performance and alignment of the AI judge model. This introduces the risk of optimizing towards flawed or biased preferences or amplifying the weaknesses of the supervising AI. Unlike RLHF which, despite its limitations, draws directly from the target human evaluators.

4 Scaling at Test-time

Test-time scaling enhances the reasoning capabilities of pre-trained models through inference-time optimization techniques, forgoing the need for additional training. These methods, spanning prompting and decoding strategies, hinge on the idea that even modestly proficient models can achieve superior performance when given extra computational "thinking time" or guided via structured prompting. By leveraging a model's existing knowledge more effectively, test-time scaling offers a flexible, resource-efficient way to boost accuracy, coherence, and problem-solving ability at deployment.

4.1 Search

4.1.1 Monte Carlo Tree Search (MCTS)

Monte Carlo Tree Search (MCTS, Coulom et al., 2006) is a heuristic search algorithm traditionally employed in game-playing AI (e.g. AlphaGo) to navigate decision trees by balancing exploration and exploitation. Adapted for LLMs, MCTS treats solution generation as a step-by-step decision process, represented as a tree of possible "thoughts." At each step, the model generates multiple potential continuations (branches), and MCTS explores these by simulating outcomes and using a scoring function (often a heuristic or value model) to determine which branches are most promising. The search then prioritizes these branches, with the flexibility to backtrack and explore alternatives later. In recent applications, MCTS leverages the LLM as both a world model and a policy provider, enhancing its ability to plan for complex tasks. For example, research on LLM-MCTS demonstrates significant improvements over standalone MCTS or LLM-induced policies (e.g., GPT-2, GPT-3.5) in tasks like multiplication, multi-hop travel planning, and object rearrangement, using minimum description length (MDL) as a guiding principle.

4.1.2 Beam Search

Beam Search is a decoding algorithm that maintains tractability in models by tracking the top K most likely partial sequences (beams) at each generation step (Meister et al., 2015). Unlike greedy decoding, which commits to a single token at a time, Beam Search explores multiple sequences in parallel, expanding each beam by one token and pruning less promising options based on probability. In reasoning contexts, this parallelism allows the model to avoid early mistakes; if one beam leads to a contradiction or low-probability outcome, another may represent a more coherent reasoning path. The highest-probability completed sequence is typically selected as the final answer, increasing the likelihood of logical consistency over single-path generation. However, Beam Search's focus on likelihood does not always guarantee factual or logical correctness. Recent advancements integrate stepwise self-evaluation into stochastic Beam Search, guiding exploration toward more accurate reasoning. For instance, a study demonstrated improvements in few-shot accuracy on GSM8K, AQuA, and StrategyQA benchmarks, respectively, compared to baseline methods.

4.1.3 Best-of-N (BoN) Search

Best-of-N (BoN) search involves generating N distinct sequences and selecting the best one based on an evaluation criterion, such as a reward model or verifier. This method enhances

output quality by sampling multiple possibilities and choosing the most promising, making it particularly useful for tasks where high accuracy or alignment with specific objectives is critical. However, BoN can be vulnerable to reward hacking if the evaluation model is imperfect, potentially skewing results. To address this, Regularized BoN (RBoN) introduces proximity regularization, improving performance when reward models are weakly correlated with the desired outcome. Inference-aware fine-tuning for BoN, using imitation learning and reinforcement learning, further optimizes LLMs by teaching a meta-strategy that balances best responses with diverse outputs, enhancing reasoning across tasks (Chan et al., 2024).

4.1.4 Search With Verifiers

Search with verifiers integrates language models with verification mechanisms to enhance the accuracy. In this framework, an LLM produces multiple candidate solutions or intermediate steps, which are subsequently evaluated by a verifier—this could be a separate model, a fine-tuned version of the LLM, or a rule-based system—to assess their correctness or quality. The search process then prioritizes paths validated by the verifier, iteratively refining the output. This approach is particularly effective for tasks requiring high factual accuracy or logical consistency, such as mathematical reasoning or code generation. For instance, Cobbe et al. (2021) demonstrated that training verifiers to solve math word problems significantly improved the problem-solving accuracy of LLMs. They introduced the GSM8K dataset, comprising 8.5K high-quality, linguistically diverse grade-school math word problems, to train verifiers capable of filtering out incorrect solutions early in the reasoning process. Building upon this, Qi et al. (2024) introduced VerifierQ, a novel approach that integrates Offline Q-learning into LLM verifier models. VerifierQ addresses challenges such as handling utterance-level Markov Decision Processes, managing large action spaces, and mitigating overestimation bias. By incorporating reinforcement learning techniques, VerifierQ enhances the efficiency, accuracy, and robustness of verifiers, leading to improved performance in mathematical reasoning tasks.

4.2 Test-time Compute Optimal Scaling

Test-time compute-optimal scaling focuses on efficiently allocating computational resources during inference to maximize LLM performance, adapting to the difficulty of individual prompts. Rather than scaling model parameters, this approach optimizes test-time compute, enabling smaller models to rival larger ones in a cost-effective and sustainable manner. It builds on the search strategies in 4.1 by introducing adaptive mechanisms to ensure compute is used judiciously. A key study by (Snell et al., 2024) highlights two primary mechanisms:

1. **PRM-based Search:** Using process-based reward models (PRMs) to evaluate multiple reasoning paths, prioritizing those with higher logical consistency rather than just final answers. This extends BoN Search by focusing on the reasoning process.
2. **Adaptive Distribution Updates:** Dynamically adjusting the model's generation process, such as through beam search or ToT, based on prompt complexity, allowing for flexible exploration of reasoning paths.

The effectiveness of these methods depends on prompt difficulty. For easy prompts, simple methods like BoN with a small N (e.g., N=5) suffice, while harder prompts benefit from PRM-based search or beam search with larger beam widths. Adaptive allocation involves

estimating prompt difficulty (e.g., via heuristics or lightweight models) and selecting the appropriate strategy, leading to efficiency gains of over 4x compared to fixed BoN baselines.

5 What Reasoning Models Mean for Embodied AI?

The advent of reasoning models marks a transformative leap for Embodied AI, enabling robotic systems to transcend traditional limitations by utilizing advanced reasoning capabilities to compete complex tasks in real-world environments. Embodied AI systems primarily consist of three main modular components: perception, reasoning and motion planning. These modules depend on each other to effectively complete complex tasks in the physical world. For instance, the perception module processes multi-modal sensory inputs into structured representations that can be used by the reasoning module to make informed decisions, which are then executed by the motion planning module. Despite their success, traditional embodied AI systems often necessitate extensive domain-specific knowledge and engineering for effective component design and integration.

Earlier research on integrating LLMs into embodied AI systems, introduced frameworks that leveraged LLMs to enhance scene understanding and task execution. One such framework is LLM-Planner, which investigates few-shot grounded planning for embodied agents, demonstrating how LLMs can enhance planning efficiency in real-world scenarios (Song et al., 2023). Similarly, Huang et al., showed that LLMs through environmental feedback, form inner monologues that facilitate embodied reasoning and planning (Huang et al., 2023). A notable advancement was PaLM-E by Google, which integrates sensor data with a language model, enhancing robot learning and visual-language modeling capabilities (Driess, et al., 2023). More recently, ELLMER (Embodied LLM-enabled Robot) framework, utilizing GPT-4 and retrieval-augmented generation, enables robots to perform long-horizon tasks in unpredictable environments (Mon-Williams, et al., 2024).

Current research progress involves enhancing reasoning capabilities in machines, enabling them to process natural language and "think" through complex, multi-step tasks in real time. For instance, Cosmos-Reason1, a family of multimodal reasoning models by NVIDIA, introduces physical common sense and embodied reasoning to enable robots to make decisions in the physical world through long chain-of-thought reasoning processes (Azzolini, et al., 2025). Additionally, Gemini Robotics, an advanced Vision-Language-Action (VLA) generalist model that uses Gemini 2.0 to enable robots perform complex tasks (Gemini Robotics Team, 2025).

6 Noteworthy Open-source LRM Projects

6.1 TinyZero

The TinyZero project became the first attempt to reproduce the DeepSeek-R1-Zero model and this was done at a remarkably low cost (<$30), with open-source code made available on GitHub (Pan et al., 2025). The project involved utilizing the core algorithm (GRPO) and training insights from the DeepSeek paper (DeepSeek AI, 2025) to train a 3B parameter base model to reason from scratch, by playing CountDown—a game in which players use basic arithmetic on selected numbers to reach a target number within a given time limit. Over time, the model evolved from generating random outputs to exhibiting emergent reasoning behaviors, including self-verification and iterative refinement strategies. The authors

conducted ablations on CountDown for a family of distilled models (Qwen-2.5-Base 0.5B, 1.5B, 3B to 7B. 0.5B), which showed an exponential increment in reasoning as model size increases. The ablations also revealed reasoning scales for both instruct and base models, with the choice of RL algorithm having minimal impact on scaling.

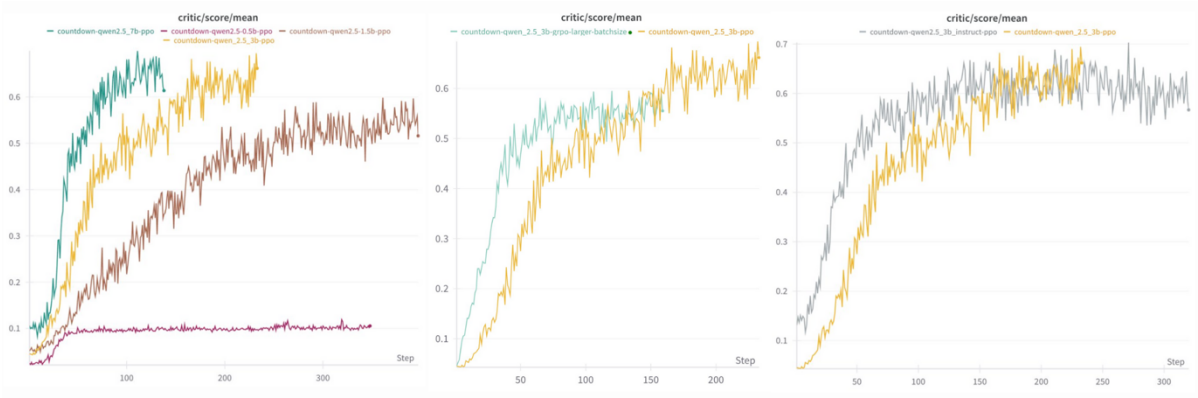

Figure 6 TinyZero Ablations on CountDown.

6.2 Open-R1

Spearheaded by Hugging Face, Open-R1 is a community-driven project aimed at reproducing DeepSeek-R1 and DeepSeek-R1-Zero with a focus on transparency and reproducibility. Open-R1 aimed to replicate and enhance DeepSeek-R1's results by providing open-source code, datasets, and training pipelines. Open-R1 offers scripts for fine-tuning models, generating synthetic datasets, and evaluating benchmarks. Its modular design allows researchers to focus on specific components such as data curation or RL pipelines. The project also extends reasoning applications beyond mathematics to fields like code, science, and medicine, marking a significant step toward democratizing advanced AI reasoning techniques. (Hugging face, 2025).

6.3 Open-Thought

Open-Thought is a collaborative effort (between Bespoke Labs and DataComp) focused on curating high-quality reasoning datasets to train small yet capable reasoning models. The initiative has released OpenThoughts-114k, a dataset comprising 114,000 examples across reasoning domains like math, science, code, and puzzles, designed to facilitate the training of models that surpass existing benchmarks. Inspired by the DeepSeek reasoning models, the authors also trained two models (OpenThinker-7B and OpenThinker-32B) on their curated datasets, both of which demonstrated competitive performance on benchmarks like AIME24 and MATH500, showcasing the advantage of well-curated data in enhancing reasoning capabilities. Find dataset and models here: https://www.open-thoughts.ai/.

6.4 VLM-R1: A Stable and Generalizable R1-Style Large Vision-Language Model

VLM-R1 (Shen, et al., 2025) extends the R1 paradigm to the visual domain by integrating reinforcement learning with rule-based rewards into vision-language models. By leveraging tasks with deterministic ground-truth annotations, VLM-R1 enables precise and stable

reward computation, facilitating the application of RL to visual understanding tasks. Experimental results indicate that the RL-based model not only delivers competitive performance on visual tasks but also surpasses supervised fine-tuning in generalization ability. Comprehensive ablation studies reveal insights such as the emergence of the "OD aha moment," the impact of training data quality, and the scaling behavior of RL across different model sizes.

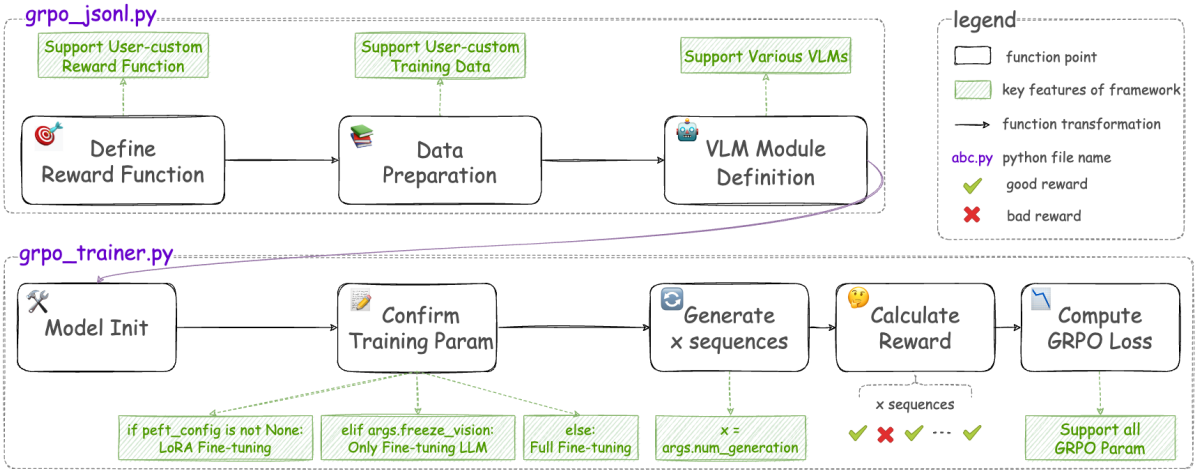

Figure 7 Illustration of the VLM-R1 framework from (Shen, et al., 2025).

7 Conclusion

The introduction of Large Reasoning Models (LRMs) marks a significant stride towards realizing *"Thinking Machines"* and a huge leap forward towards achieving AGI. LRMs demonstrate an unprecedented capability for advance problem-solving through structured analysis and verification of their own reasoning processes. This fundamental shift directs future research towards actively enhancing these emergent reasoning capabilities. Thus, as inference-time scaling techniques continue to improve, training methodologies evolve, and architectural innovations emerge, we can anticipate LRMs tackling increasingly more complex reasoning tasks across diverse domains. Despite significant progress, achieving robust, superhuman-like reasoning remains a challenge, underscoring the need for continued open research, dedicated benchmarks, and ethical diligence as we progress towards Artificial General Intelligence (AGI).

Acknowledgements

First, I would like to express my deepest gratitude to my mother for her unwavering support and encouragement, which has been the cornerstone of my journey throughout this research. I would also like to extend my sincere thanks to ***Prof. Yu Yue, Dr. Yemin Shi, Dr. Börje Karlsson, Dr. Siwei Dong, Dr. Guangyao Chen and Yu Shu*** for their invaluable guidance and insightful feedback during the research and writing process, as well as to the broader academic community for continually inspiring innovation in the field.

References

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, *59*(236), 433-460.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature*, *521*(7553), 436-444.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural computation, 9(8), 1735-1780.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., ... Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems, 33, 9459–9474

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971.*

OpenAI. (2024). *GPT-4o System Card*. arXiv preprint arXiv:2410.21276.

Gemini Team Google, 2023. Gemini: A Family of Highly Capable Multimodal Models. *arXiv: arXiv:2312.11805.*

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). *Improving language understanding by generative pre-training*. OpenAI.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners*. OpenAI.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155.*

DeepSeek AI (2025). Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

Chan, A. J., Sun, H., Holt, S., & van der Schaar, M. (2024). Dense Reward for Free in Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2402.00782*.

Lee, H., Phatale, S., Mansoor, H., Mesnard, T., Ferret, J., Lu, K., Bishop, C., Hall, E., Carbune, V., Rastogi, A., & Prakash, S. (2023). RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. *arXiv preprint arXiv:2309.00267*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint arXiv:2201.11903*.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *Advances in Neural Information Processing Systems, 36.*

Zhao, E., Awasthi, P., & Gollapudi, S. (2025). Sample, Scrutinize and Scale: Effective Inference-Time Search by Scaling Verification. *arXiv preprint arXiv:2502.01839*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., & Finn, C. (2023). Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv preprint arXiv:2305.18290*.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., & Guo, D. (2024). DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., & Lin, M. (2025). Understanding R1-Zero-Like Training: A Critical Perspective. *arXiv preprint arXiv:2503.20783*.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Fan, T., Liu, G., Liu, L., Liu, X., Lin, H., Lin, Z., Ma, B., Sheng, G., Tong, Y., Zhang, C., Zhang, M., Zhang, W., Zhu, H., Zhu, J., Chen, J., Chen, J., Wang, C., Yu, H., Dai, W., Song, Y., Wei, X., Zhou, H., Liu, J., Ma, W., Zhang, Y., Yan, L., Qiao, M., Wu, Y., & Wang, M. (2025). DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint arXiv:2503.14476*.

Geoffery Hinton, "Will digital intelligence replace biological intelligence?" Romanes Lectures 2024. *Video: https://www.youtube.com/watch?v=N1TEjTeQeg0*

Yann LeCun, “UW ECE 2023-2024 Dean W. Lytle Electrical & Computer Engineering Endowed Lecture Series”, 2024. *Video: https://www.youtube.com/watch?v=d_bdU3LsLzE*

Ilya Sutskever, “NeurIPS 2024 Test of Time Award” Talk. *Video:*

https://www.youtube.com/watch?v=YD-9NG1Ke5Y

Shazeer, N. (2019). Fast Transformer Decoding: One Write-Head is All You Need. *arXiv preprint arXiv:1911.02150.*

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., & Ré, C. (2022). FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *arXiv preprint arXiv:2205.14135*.

DeepSeek-AI, Liu, A., Feng, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Ruan, C., Dai, D., Guo, D., & others. (2024). DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model. *arXiv preprint arXiv:2405.04434.*

Eigen, D., Ranzato, M., & Sutskever, I. (2014). Learning Factored Representations in a Deep Mixture of Experts. *arXiv preprint arXiv:1312.4314*.

Villalobos, P., Ho, A., Sevilla, J., Besiroglu, T., Heim, L., & Hobbhahn, M. (2022). Will we run out of data? Limits of LLM scaling based on human-generated data. *arXiv preprint arXiv:2211.04325*.

Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F., & Sanghai, S. (2023). GQA: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Amin, K., Bie, A., Kong, W., Kurakin, A., Ponomareva, N., Syed, U., Terzis, A., & Vassilvitskii, S. (2024). Private prediction for large-scale synthetic text generation. *arXiv preprint arXiv:2407.12108*.

OpenThoughts Project, Bespoke Labs, 2025. URL: https://www.open-thoughts.ai/blog/launch.

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Sparse Transformers. *arXiv preprint arXiv:1904.10509*.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., ... & Christiano, P. (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems, 33*, 3008–3021.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... & Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... & Sifre, L. (2022). Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3645–3650). Association for Computational Linguistics.

Bahri, Y., Dyer, E., Kaplan, J., Lee, J., & Sharma, U. (2021). Explaining Neural Scaling

Laws. *arXiv preprint arXiv:2102.06701*.

Ranzato, M. A., Chopra, S., Auli, M., & Zaremba, W. (2016). Sequence level Training With Recurrent Neural Networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.

Williams, R. J. (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8(3–4), 229–256.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2017). Self-critical Sequence Training for Image Captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 7008–7024).

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., ... & Kavukcuoglu, K. (2016). Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)* (pp. 1928–1937).

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Bellman, R. (1957). A Markovian Decision Process. *Journal of Mathematics and Mechanics*, 6(5), 679–684.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., & Chen, W. (2021). LoRA: Low-rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*.

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., & Liu, Y. (2016). Minimum Risk Training for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1683–1692). Association for Computational Linguistics.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... & Hassabis, D. (2016). Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature*, 529(7587), 484–489.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., ... & McCandlish, S. (2022). Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073*.

Sun, J., Wang, Y., Li, X., Zhang, L., Chen, H., Liu, M., ... & Zhou, J. (2025). ARMOR v0.1: Empowering Autoregressive Multimodal Understanding Model with Interleaved Multimodal Generation via Asymmetric Synergy. *arXiv preprint arXiv:2503.06542*.

Zhou, Y., Li, J., Wang, S., Chen, R., Xu, F., & Yang, M. (2024). Automatic Curriculum Expert Iteration for Reliable LLM Reasoning. *arXiv preprint arXiv:2410.07627*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning* (Vol. 37, pp. 1889–1897). PMLR.

Lee, H., Phatale, S., Mansoor, H., Mesnard, T., Ferret, J., Lu, K., ... & Prakash, S. (2023). RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. *arXiv preprint arXiv:2309.00267*.

Coulom, R. (2006). Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In *Proceedings of the 5th International Conference on Computers and Games* (pp. 72–83). Springer.

Wang, L., Chen, Y., Liu, Z., & Sun, M. (2025). Thinking Preference Optimization. *arXiv preprint arXiv:2502.13173*.

Gemini Team, Gemini 2.0 Flash Thinking, 2024. *URL: https://deepmind.google/technologies/gemini/flash-thinking/*

Rajpurkar P, Zhang J, Lopyrev K, Liang P. SQUAD: 100,000+ Questions for Machine Comprehension of Text. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, TX: Association for Computational Linguistics; 2016:2383-2392.

Camburu O-M, Rocktäschel T, Lukasiewicz T, Blunsom P. e-SNLI: Natural Language Inference with Natural Language Explanations. In: *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*. Montréal, Canada: Curran Associates, Inc.; 2018:9539-9549.

Talmor A, Herzig J, Lourie N, Berant J. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Minneapolis, MN: Association for Computational Linguistics; 2019:4149-4158.

Narang S, Raffel C, Lee K, et al. Learning to Explain: A Generative Framework for Natural Language Explanations. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics; 2020:4930-4942.

Luong TQ, Xinbo Z, Zhanming J, Peng S, Xiaoran J, Hang L (2024). ReFT: Reasoning with Reinforced Fine-Tuning. *arXiv preprint arXiv:2401.08967*

Zhang, K., Yao, Q., Lai, B., Huang, J., Fang, W., Tao, D., Song, M., & Liu, S. (2025). Reasoning with Reinforced Functional Token Tuning. *arXiv preprint arXiv:2502.13389*.

Shridhar, K., Stolfo, A., & Sachan, M. (2022). Distilling Reasoning Capabilities into Smaller Language Models. *arXiv preprint arXiv:2212.00193*.

Li, Z., Ji, Y., Meng, R., & He, D. (2024). Learning From Committee: Reasoning Distillation From a Mixture of Teachers With Peer-Review. *arXiv preprint arXiv:2410.03663*.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

OpenAI. (2024). OpenAI o1 System Card. OpenAI. URL: https://arxiv.org/abs/2412.16720

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., & Zhou, D. (2022). Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv preprint arXiv:2203.11171.

Wang, H., Xiong, W., Xie, T., Zhao, H., & Zhang, T. (2024). Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. In Findings of the Association for Computational Linguistics: EMNLP 2024 (pp. 10582–10592).

Shen, M., Zeng, G., Qi, Z., Hong, Z.-W., Chen, Z., Lu, W., Wornell, G., Das, S., Cox, D., & Gan, C. (2025). Satori: Reinforcement Learning with Chain-of-Action-Thought Enhances LLM Reasoning via Autoregressive Search. *arXiv preprint arXiv:2502.02508*.

Tang, Y., Wang, S., & Munos, R. (2025). Learning to Chain-of-Thought with Jensen's Evidence Lower Bound. *arXiv preprint arXiv:2503.19618*

Ding, B., Qin, C., Zhao, R., Luo, T., Li, X., Chen, G., Xia, W., Hu, J., Luu, A. T., & Joty, S. (2024). Data Augmentation Using Large Language Models: Data Perspectives, Learning Paradigms, and Challenges. *Findings of the Association for Computational Linguistics: ACL 2024*, 97–108.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., & Misra, V. (2022). Solving Quantitative Reasoning Problems with Language Models. *Advances in Neural Information Processing Systems, 35*, 12087–12099.

Shen, H., Liu, P., Li, J., Fang, C., Ma, Y., Liao, J., Shen, Q., Zhang, Z., Zhao, K., Zhang, Q., Xu, R., & Zhao, T. (2025). VLM-R1: A Stable and Generalizable R1-style Large Vision-Language Model. arXiv preprint arXiv:2504.07615.

Song, C. H., Wu, J., Washington, C., Sadler, B. M., Chao, W.-L., & Su, Y. (2023). LLM-Planner: LLM-Planner: Few-Shot Grounded Planning for Embodied Agents with Large Language Models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., ... & Ichter, B. (2023). Inner Monologue: Embodied Reasoning Through Planning With Language Models. In Proceedings of the 6th Conference on Robot Learning (CoRL) (pp. 1769–1782). PMLR.

Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., ... & Florence, P. (2023). *PaLM-E: An Embodied Multimodal Language Model*. In *Proceedings of the 40th International Conference on Machine Learning (ICML)* (pp. 8469–8488). PMLR.

Mon-Williams, R., Li, G., Long, R., Du, W., & Lucas, C. G. (2025). Embodied Large Language Models Enable Robots to Complete Complex Tasks in Unpredictable *Environments*. Nature Machine Intelligence. Nature Machine Intelligence.

Gemini Robotics Team, Abeyruwan, S., Ainslie, J., Alayrac, J.-B., Gonzalez Arenas, M., Armstrong, T., ... & Zeng, A. (2025). Gemini Robotics: Bringing AI into the physical world. *arXiv preprint arXiv:2503.20020*.

Azzolini, A., Brandon, H., Chattopadhyay, P., Chen, H., Chu, J., Cui, Y., ... & Zhang, Z. (2025). Cosmos-Reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*.

Meister, C., Vieira, T., & Cotterell, R. (2020). Best-first beam search. *Transactions of the Association for Computational Linguistics*, 8, 795–809.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A Neural Image Caption Generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3156–3164).

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., & Schulman, J. (2021). Training Verifiers to Solve Math Word Problems. *arXiv. https://arxiv.org/abs/2110.14168.*

Qi, J., Lu, Y., Zeng, Y., Guo, J., et al. (2024). VerifierQ: Enhancing LLM Verification with Offline Q-learning. *arXiv preprint arXiv:2410.16033*.

Snell, C., Lee, J., Xu, K., & Kumar, A. (2024). Scaling LLM Test-Time Compute Optimally Can Be More Effective Than Scaling Model Parameters. *arXiv. https://arxiv.org/abs/2408.03314.*

Gulcehre C, Wang S, Gehrmann S, Huang S, Chang M, Brockman G, Zoph B, Borgeaud S, Brock A, Lee J, Bousquet O, McGrew B, Irving G, Zoph B, Novikova J, (2023). Reinforced self-training (ReST) for Language Modeling. *arXiv preprint arXiv:2308.08998.*

Zhang D, Zhoubian S, Hu Z, Yue Y, Dong Y, Tang J. ReST-MCTS: LLM Self-Training via Process Reward Guided Tree Search. arXiv preprint arXiv:2406.03816.

Jiayi Pan, et al., 2025, TinyZero GitHub. https://github.com/Jiayi-Pan/TinyZero.

Open-R1 Huggingface, et al., 2025. https://huggingface.co/blog/open-r1.