

# Survey

## A Survey of Test-time Compute: From Intuitive Inference to Deliberate Reasoning

Yixin Ji<sup>1</sup>, Juntao Li<sup>1\*</sup>, Yang Xiang<sup>1</sup>, Hai Ye<sup>2</sup>, Kaixin Wu<sup>3</sup>, Kai Yao<sup>3</sup>, Jia Xu<sup>3</sup>, Linjian Mo<sup>3</sup>, and Min Zhang<sup>1</sup>

<sup>1</sup>Soochow University

jyixin169@gmail.com, ljt@suda.edu.cn, minzhang@suda.edu.cn

<sup>2</sup>National University of Singapore

<sup>3</sup>Ant Group

*The remarkable performance of the o1 model in complex reasoning demonstrates that test-time compute scaling can further unlock the model's potential, enabling powerful System-2 thinking. However, there is still a lack of comprehensive surveys for test-time compute scaling. We trace the concept of test-time compute back to System-1 models. In System-1 models, test-time compute addresses distribution shifts and improves robustness and generalization through parameter updating, input modification, representation editing, and output calibration. In System-2 models, it enhances the model's reasoning ability to solve complex problems through repeated sampling, self-correction, and tree search. We organize this survey according to the trend of System-1 to System-2 thinking, highlighting the key role of test-time compute in the transition from System-1 models to weak System-2 models, and then to strong System-2 models. We also point out advanced topics and future directions.<sup>1</sup>*

### 1. Introduction

Over the past decades, deep learning with its scaling effects has been the driving engine behind the artificial intelligence revolution. Particularly in the text modality, large language models (LLMs) represented by the GPT series (Brown et al. 2020; Ouyang et al. 2022; OpenAI 2023) have demonstrated that larger models and more training data lead to better performance on downstream tasks. However, on the one hand, further scaling in the training phase becomes difficult due to the scarcity of data and computational resources (Villalobos et al. 2024); on the other hand, existing models still perform far below expectations in terms of robustness and handling complex tasks. These shortcomings are attributed to the model's reliance on fast, intuitive System-1 thinking, rather than slow, deep System-2 thinking (Weston and Sukhbaatar 2023). Recently, large

---

\* Corresponding author.

1 [https://github.com/Dereck0602/Awesome\\_Test\\_Time\\_LLMs](https://github.com/Dereck0602/Awesome_Test_Time_LLMs).

Action Editor: Anh Tuan Luu. Submission received: 29 August 2025; revised version received: 13 April 2026; accepted for publication: 1 May 2026.

<https://doi.org/10.1162/COLLa.629>

reasoning models (LRMs), represented by OpenAI-o1/o3 (OpenAI 2024), DeepSeek-R1 (Guo et al. 2025a), and Gemini 2.5 (DeepMind 2025), equipped with System-2 thinking, have gained attention for their outstanding performance in complex reasoning tasks. It demonstrates a test-time compute scaling effect: The greater the computational effort in the inference, the better the model’s performance.

The concept of test-time compute emerged before the rise of LLMs and was initially applied to System-1 models (illustrated in Figure 1). These System-1 models can only perform limited perceptual tasks, relying on patterns learned during training for predictions. As a result, they are constrained by the assumption that training and testing are identically distributed and lack robustness and generalization to distribution shifts (Zhuang et al. 2020). Many studies have explored test-time adaptation (TTA) to improve model robustness by updating parameters (Wang et al. 2021; Ye, Xie, and Ng 2023), modifying the input (Dong et al. 2024b), editing representations (Rimsky et al. 2024), and calibrating the output (Zhang et al. 2023c). With TTA, the System-1 model slows down its thinking process and has better generalization. However, TTA is an implicit slow-thinking process, unable to exhibit explicit, logical thinking processes like humans, and struggles to handle complex reasoning tasks. Thus, TTA-enabled models perform weak System-2 thinking.

Currently, advanced LLMs with chain-of-thought (CoT) prompting (Wei et al. 2022) have enabled LLMs to perform explicit System-2 thinking (Hagendorff, Fabi, and Kosinski 2023). However, vanilla CoT is limited by error accumulation and linear thinking (Stechly, Valmeekam, and Kambhampati 2024), making it difficult to fully simulate non-linear human cognitive processes such as brainstorming, reflection, and backtracking. To achieve stronger System-2 models, researchers use test-time compute strategies to extend model reasoning’s breadth, depth, and accuracy, such as repeated sampling (Cobbe et al. 2021), self-correction (Shinn et al. 2023), and tree search (Yao et al. 2023). Repeated sampling simulates the diversity of human thinking, self-correction enables LLMs to reflect, and tree search enhances reasoning depth and backtracking.

To provide a comprehensive overview of the current landscape of test-time compute (Figure 2), this survey covers test-time compute strategies ranging from System-1 to

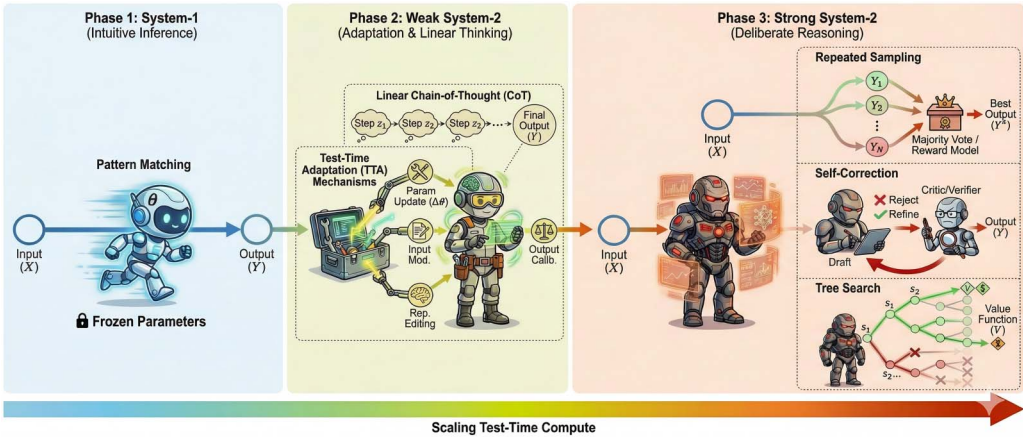


**Figure 1** The spectrum of test-time compute. Transitioning from intuitive System-1 models to complex and deliberate System-2 reasoning models.

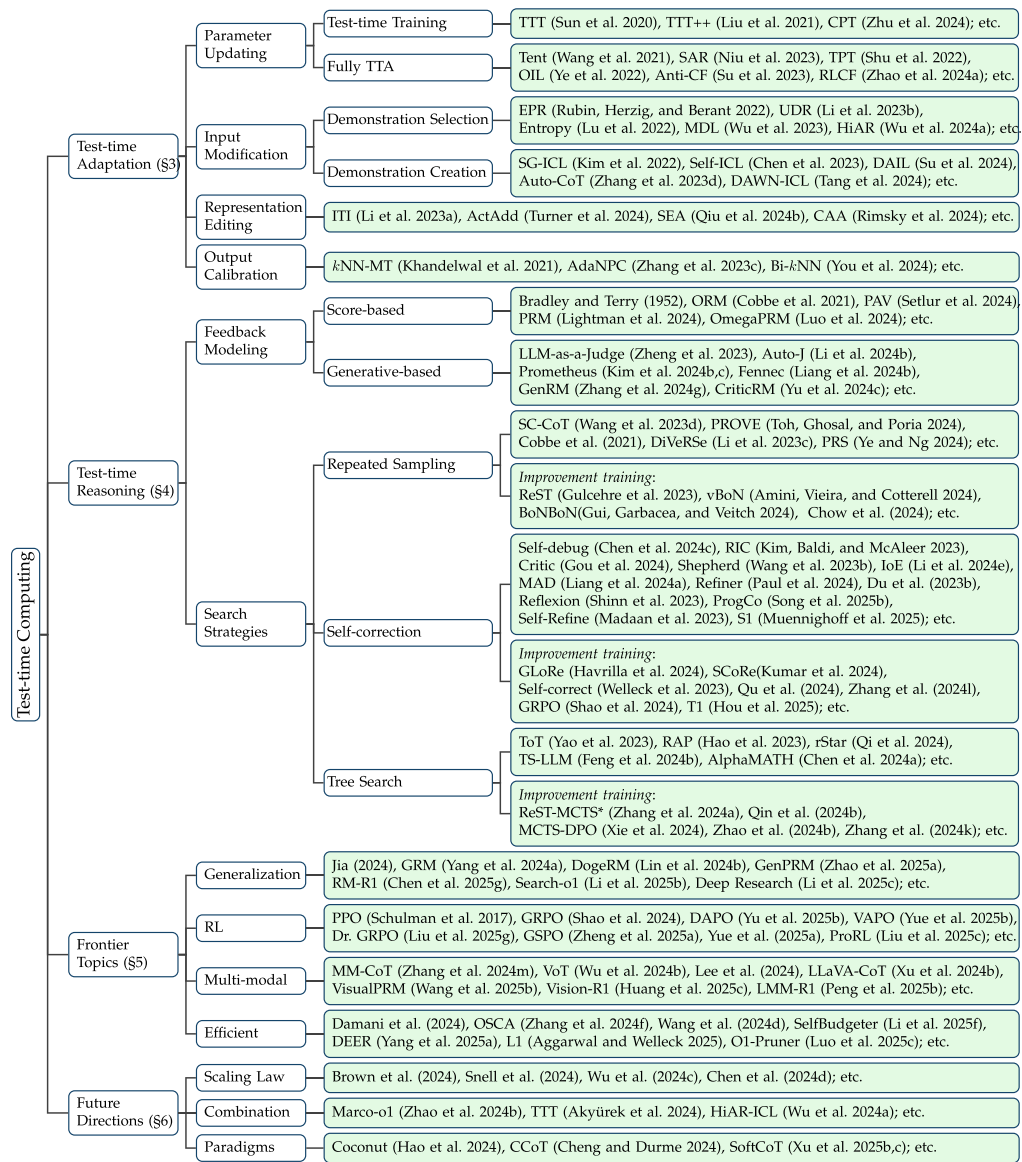


**Figure 2**  
Taxonomy of test-time computing methods, frontier topics, and future directions.

System-2 models, as well as training methods that enable or enhance a model’s test-time compute capabilities. Although these training methods shift part of the computational cost to the training phase, their primary goal remains improving the model’s test-time compute capacity, making them an integral component of test-time compute research. In Section 2, we present the background of System-1 and System-2 thinking, and a unified framework for test-time compute. Section 3 and Section 4 detail the test-time compute methods for the System-1 and System-2 models. Then, we discuss frontier topics toward real-world applications and future directions in Sections 5 and 6. Additionally, we review benchmarks in Section 7. To the best of our knowledge, this article is the first to

systematically review test-time compute methods and thoroughly explore their critical role in advancing models from System-1 to weak System-2, and ultimately to strong System-2 thinking. Compared with contemporaneous surveys (Xiang et al. 2025; Wu 2025; Li et al. 2025g; Chen et al. 2025d), this article is not limited to CoT-based test-time compute techniques; instead, it covers a broader range of test-time compute methods within System-1 and discusses the technical evolution and integration between the two.

## 2. Preliminary

### 2.1 System-1 and System-2 Thinking

System-1 and System-2 thinking are psychological concepts (Kahneman 2011). When recognizing familiar patterns or handling simple problems, humans often respond intuitively. This automatic, fast thinking is called System-1 thinking. In contrast, when dealing with complex problems like mathematical proofs or logical reasoning, deep and deliberate thought is required, referred as System-2 thinking—slow and reflective. In the field of artificial intelligence, researchers also use these terms to describe different types of models (LeCun 2022). System-1 models respond directly based on internally encoded perceptual information and world knowledge without showing any intermediate decision-making process. In contrast, System-2 models explicitly generate reasoning processes and solve tasks incrementally. Before the rise of LLMs, System-1 models were the mainstream in AI. Although many deep learning models, such as ResNet, Transformer, and BERT, achieve excellent performance in various tasks in computer vision and natural language processing, these System-1 models, similar to human intuition, lack sufficient robustness and are prone to errors (Geirhos et al. 2020; Wang, Wang, and Yang 2022; Du et al. 2023a). Nowadays, the strong generation and reasoning capabilities of LLMs make it possible to build System-2 models. Wei et al. (2022) propose CoT, which allows LLMs to generate intermediate reasoning steps progressively during inference. Empirical and theoretical results show that this approach significantly outperforms methods that generate answers directly (Kojima et al. 2022; Zhou et al. 2023; Feng et al. 2024a; Li et al. 2024h). However, current System-2 models represented by CoT prompting still have shortcomings. The intermediate processes generated by LLMs may contain errors, leading to cumulative mistakes and ultimately resulting in incorrect answers. Although retrieval-augmented generation (RAG) helps mitigate factual errors (Trivedi et al. 2023; Guan et al. 2024; Wang et al. 2024m), its impact on improving reasoning abilities remains limited. As a result, CoT-enabled LLMs are still at the weak System-2 thinking stage.

### 2.2 The Unified Framework of Test-time Compute

To systematically analyze the diverse methods emerging in this field, we propose a unified framework. While traditional inference performs a single forward pass  $y \sim P_\theta(y|x)$  to generate the output, test-time compute invests additional computational resources to search for an optimal state that maximizes a specific objective. Formally, we define a test-time compute strategy as a tuple  $\mathcal{T} = (S, \mathcal{J}, \mathcal{A})$ , consisting of three core components:

- Search Space ( $S$ ): The set of variables or states that are optimized during inference.

- **Test-time Objective ( $\mathcal{J}$ ):** The function used to evaluate the quality of a state  $s \in \mathcal{S}$ , serving as the optimization signal.
- **Optimization Operation ( $\mathcal{A}$ ):** The method employed to navigate the search space  $\mathcal{S}$  to find the optimal state  $s^*$ .

Under this framework, the final output  $\hat{y}$  is derived from the optimization process:

$$\hat{y} = \mathcal{M}_{dec} \left( \arg \max_{s \in \mathcal{S}} \mathcal{J}(s; x, \theta) \right)$$

This framework provides a clear boundary and connection between the two main paradigms discussed in this survey. For test-time adaptation for System-1 models (§3), the search space lies in the model state and contextual inputs, addressing distribution shifts by intervening in or optimizing the parameter space, input space, representation space, and output probability space. For LLMs, the emergence of CoT leads to an exponential expansion of the generation decoding space. As a result, test-time reasoning (§4) methods focus their search on the decoding space, seeking optimal decoding trajectories through optimization strategies such as repeated sampling, self-correction, and tree search, with the objective of maximizing reasoning performance. Table 1 systematically maps representative methods from both test-time adaptation and test-time reasoning to the  $(\mathcal{S}, \mathcal{J}, \mathcal{A})$  tuple.

3. Test-time Adaptation for System-1 Thinking

3.1 Updating the Model

Model updating utilizes test sample information to fine-tune model parameters during the inference stage, enabling the model to adapt to the test distribution (Figure 3). The key lies in obtaining test distribution information to provide learning signals and using appropriate parameters and optimization algorithms to achieve efficient and stable updates. Under the unified framework, parameter-updating TTA searches over the model parameter space, with objectives  $\mathcal{J}$  such as entropy minimization or self-supervised loss, and uses gradient-based optimization  $\mathcal{A}$ .

**Table 1**  
Mapping representative methods to the unified framework  $\mathcal{T} = (\mathcal{S}, \mathcal{J}, \mathcal{A})$ .

| Category               | Method                               | Search Space ( $\mathcal{S}$ ) | Objective ( $\mathcal{J}$ )    | Optimization ( $\mathcal{A}$ ) |
|------------------------|--------------------------------------|--------------------------------|--------------------------------|--------------------------------|
| Parameter Updating     | Tent (Wang et al. 2021)              | Model parameters               | Entropy minimization           | Gradient descent               |
| Parameter Updating     | TTT (Sun et al. 2020)                | Model parameters               | Self-supervised auxiliary loss | Gradient descent               |
| Input Modification     | EPR (Rubin, Herzig, and Berant 2022) | Demonstration space            | Retrieval similarity           | Contrastive retrieval          |
| Input Modification     | Auto-CoT (Zhang et al. 2023d)        | CoT demonstration space        | Diversity + representativeness | Clustering-based selection     |
| Representation Editing | ActAdd (Turner et al. 2024)          | Representation space           | Contrastive steering           | Vector addition                |
| Output Calibration     | kNN-MT (Khandelwal et al. 2021)      | Output probability space       | Distributional alignment       | Weighted interpolation         |
| Repeated Sampling      | CoT-SC (Wang et al. 2023d)           | Decoding trajectory space      | Self-consistency               | Majority voting                |
| Repeated Sampling      | Best-of-N (Cobbe et al. 2021)        | Decoding trajectory space      | Verifier reward                | Top-1 selection                |
| Self-Correction        | Self-Refine (Madaan et al. 2023)     | Decoding trajectory space      | Critique reward                | Iterative refinement           |
| Self-Correction        | Multi-agent Debate (Du et al. 2023b) | Decoding trajectory space      | Debate feedback                | Iterative refinement           |
| Tree Search            | ToT (Yao et al. 2023)                | Decoding trajectory space      | Local action value             | BFS / DFS                      |
| Tree Search            | RAP (Hao et al. 2023)                | Decoding trajectory space      | Q-value                        | MCTS                           |

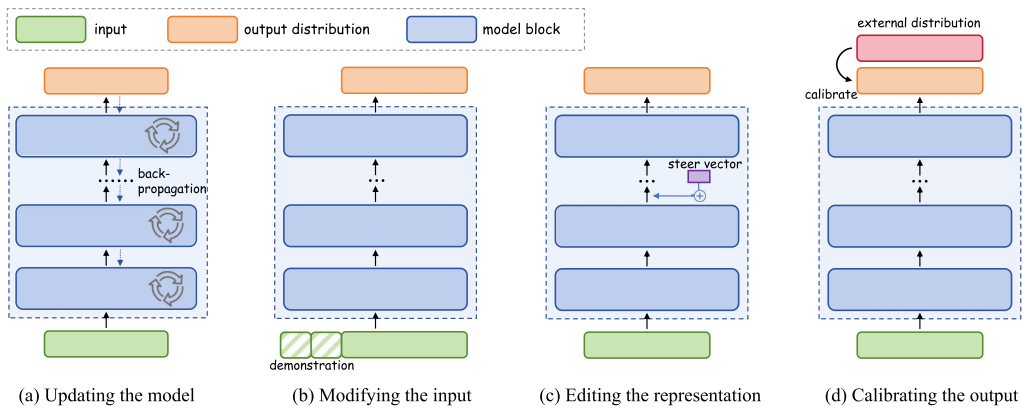


**Figure 3**  
Illustration of various kinds of test-time adaptation methods.

*Learning Signal.* In the inference stage, the ground-truth of test samples is unavailable. Thus, many works attempt to design unsupervised or self-supervised objectives as learning signals. Existing learning signals can be classified into two categories based on whether the training process can be modified: **test-time training** (TTT) and **fully test-time adaptation** (FTTA). TTT assumes users can modify the training process by incorporating distribution-shift-aware auxiliary tasks. During test-time adaptation, the auxiliary task loss serves as the learning signal for optimization. Many self-supervised tasks have been shown to be effective as auxiliary tasks in image modality, such as rotation prediction (Sun et al. 2020), meta learning (Bartler et al. 2022), masked auto-encoding (Gandelsman et al. 2022), and contrastive learning (Liu et al. 2021; Chen et al. 2022). Among them, contrastive learning has been successfully applied to test-time adaptation for visual-language tasks due to its generalization of self-supervised learning within and across modalities (Zhu et al. 2024).

In contrast, FTFA is free from accessing the training process and instead uses internal or external feedback on test samples as learning signals. Uncertainty is the most commonly learned signal, driven by the motivation that when test samples shift from the training distribution, the model’s confidence in its predictions is lower, resulting in higher uncertainty. Tent (Wang et al. 2021) uses the entropy of model predictions as a measure of uncertainty and updates the model by minimizing the entropy. MEMO (Zhang, Levine, and Finn 2022) augments the data for a single test sample and then minimizes its marginal entropy, which is more stable compared to Tent in the single-sample TTA setting. However, minimizing entropy also has pitfalls, as blindly reducing prediction uncertainty may cause the model to collapse and make trivial predictions (Press et al. 2024; Zhao et al. 2023; Su et al. 2023). Some studies propose new regularization terms for minimizing entropy to avoid model collapse, including Kullback-Leibler divergence (Su et al. 2023), moment matching (Hassan et al. 2023), and entropy matching (Bar, Shaer, and Romano 2024). For specific tasks, a small amount of human feedback or external model rewards can also serve as high-quality learning signals. Gao, Choi, and Artzi (2022) and Li et al. (2022b) utilize user feedback to adapt the QA model. Zhan et al. (2023) apply test-time adaptation to multilingual machine translation tasks by using COMET (Rei et al. 2020) for evaluating translation quality. In cross-modal tasks such as image-text retrieval and image captioning, RLCF (Zhao et al.

2024a) demonstrates its effectiveness by using CLIP scores (Radford et al. 2021) as TTA signals. In language modeling, training with relevant contextual text at test time can reduce perplexity (Hardt and Sun 2024; Wang, Ma, and Cai 2024). Hübötter et al. (2025) theoretically shows that it reduces the uncertainty of test samples and proposes a more effective active learning selection strategy.

*Updating Parameters.* To advance the application of TTA in real-world scenarios, researchers must address challenges of efficiency and stability. To improve efficiency, many methods only fine-tune a small subset of parameters, such as normalization layers (Schneider et al. 2020), soft prompt (Lester, Al-Rfou, and Constant 2021; Shu et al. 2022; Hassan et al. 2023; Ma et al. 2023; Feng et al. 2023; Niu et al. 2024), low-rank module (Hu et al. 2022; Imam et al. 2024; Hu et al. 2025b), adapter module (Houlsby et al. 2019; Muhtar et al. 2024; Su et al. 2023), and cross-modality projector (Zhao et al. 2024a). Although the number of parameters to fine-tune is reduced, TTA still requires an additional backward propagation. Typically, the time cost of a backward propagation is approximately twice that of a forward propagation. Thus, Niu et al. (2024) propose FOA, which is free from backward propagation by adapting soft prompt through covariance matrix adaptation evolution strategy.

The stability of TTA is primarily shown in two aspects. On the one hand, unsupervised or self-supervised learning signals inevitably introduce noise into the optimization process, resulting in TTA optimizing the model in the incorrect gradient direction. To address this, Niu et al. (2023) and Gong et al. (2024b) propose noise data filtering strategies and the robust sharpness-aware optimizer. On the other hand, in real-world scenarios, the distribution of test samples may continually shift, but continual TTA optimization may lead to catastrophic forgetting of the model’s original knowledge. Episodic TTA (Shu et al. 2022; Zhao et al. 2024a) is a setting to avoid forgetting, which resets the model parameters to their original state after TTA on a single test sample. However, episodic TTA frequently loads the original model, leading to higher inference latency and also limiting the model’s incremental learning capability. To overcome the dilemma, a common trick is the exponential moving average (Wortsman et al. 2022; Ye et al. 2022), which incorporates information from previous model states.

Takeaways: Parameter-updating TTA

- **Goal:** Parameter-updating TTA aims to adapt models to test-time distribution shifts using unsupervised, self-supervised, or external feedback signals.
- **Methods:** Main approaches include fine-tuning lightweight components with learning signals such as entropy minimization, self-supervised objectives, or external rewards, alongside efficient optimizers and backward-free strategies.
- **Challenges & limitations:** Noisy learning signals cause instability or collapse, and continual adaptation risks catastrophic forgetting; existing tricks struggle to involve trade-offs among stability, efficiency, and incremental learning.

3.2 Modifying the Input

When it comes to LLM, the large number of parameters makes model update-based TTA methods face a tougher dilemma of efficiency and stability. As a result, input-modification-based methods, which do not rely on parameter updates, have become the

mainstream method for TTA in LLMs. The effectiveness of input-modified TTA stems from the in-context learning (ICL) capability of LLMs, which can significantly improve the performance by adding some demonstrations before the test sample. ICL is highly sensitive to the selection and order of demonstrations. Therefore, the core objective of input-modification TTA is to select appropriate demonstrations for the test samples and arrange them in the optimal order to maximize the effectiveness of ICL.

First, empirical studies (Liu et al. 2022) show that the more similar the demonstrations are to the test sample, the better the ICL performance. Therefore, retrieval models like BM25 and SentenceBERT are used to retrieve demonstrations semantically closest to the test sample and rank them in descending order of similarity (Qin et al. 2024a; Luo et al. 2023). To improve the accuracy of demonstration retrieval, Rubin, Herzig, and Berant (2022) and Li et al. (2023b) specifically train the demonstration retriever by contrastive learning. Then, as researchers delve deeper into the mechanisms of ICL, ICL is considered to conduct implicit gradient descent on the demonstrations (Dai et al. 2023). Therefore, from the perspective of training data, demonstrations also need to be informative and diverse (Su et al. 2022; Li and Qiu 2023). Wang et al. (2023c) view language models as topic models and formulate the demonstration selection problem as solving a Bayesian optimal classifier. Additionally, the ordering of examples is another important area for improvement. Lu et al. (2022) and Wu et al. (2023) use information theory as a guide to select the examples with maximum local entropy and minimum description length for ranking, respectively. Scarlatos and Lan (2024) and Zhang, Feng, and Tan (2022) model it as a sequential decision problem and optimize demonstration selection and ordering through reinforcement learning (RL).

Another line of work (Chen et al. 2023; Lyu et al. 2023; Kim et al. 2022; Zhang et al. 2023d) argues that in practice, combining a limited set of externally provided examples may not always be the optimal choice. LLMs can leverage their generative and annotation capabilities to create better demonstrations. DAIL (Su et al. 2024) constructs a demonstration memory, storing previous test samples and their predictions as candidate demonstrations for subsequent samples. DAWN-ICL (Tang et al. 2024) further models the traversal order of test samples as a planning task and optimizes it by the Monte Carlo tree search (MCTS).

#### Takeaways: Input-modification TTA

- **Goal:** Input-modification TTA leverages in-context learning to adapt LLMs to test-time distribution shifts by constructing proper demonstrations.
- **Methods:** Main approaches include retrieving semantically similar, informative, and diverse demonstrations, optimizing demonstration ordering, and generating or reusing demonstrations through self-generated memory.
- **Challenges & limitations:** Performance is sensitive to demonstration quality, diversity, and order; retrieval and generation may introduce noise, and suboptimal demonstrations may cause instability or degraded generalization.

### 3.3 Editing the Representation

For generative LLMs, some studies have found that the performance bottleneck is not in encoding world knowledge, but in the large gap between the information in intermediate layers and the output. During the inference phase, editing the representation can help externalize the intermediate knowledge into the output. PPLM (Dathathri et al.

2020) performs gradient-based representation editing under the guidance of a small language model to control the style of outputs. ActAdd (Turner et al. 2024) selects two semantically contrastive prompts and calculates the difference between their representations as a steering vector, which is then added to the residual stream. Representation editing based on contrastive prompts has demonstrated its effectiveness in broader scenarios, including instruction following (Stolfo et al. 2024), alleviating hallucinations (Li et al. 2023a; Arditi et al. 2024), reducing toxicity (Liu et al. 2024b; Lu and Rimsky 2024), and personality (Cao et al. 2024). SEA (Qiu et al. 2024b) projects representations onto directions with maximum covariance with positive prompts and minimum covariance with negative prompts. They also introduce nonlinear feature transformations, allowing representation editing to go beyond linearly separable representations. Scalena, Sarti, and Nissim (2024) conduct an in-depth study on the selection of steering intensity. They find that applying a gradually decreasing steering intensity to each output token can improve control over the generation without compromising quality.

Takeaways: Representation-editing TTA

- **Goal:** Representation-editing TTA improves controllable generation by directly intervening in hidden-layer representations at inference time.
- **Methods:** Typical methods edit activations using steering vectors derived from gradients, contrastive prompts, or spectral objectives.
- **Challenges & limitations:** Key issues include selecting appropriate steering directions and intensities, and most methods rely on heuristic or linear assumptions that may not generalize across tasks.

3.4 Calibrating the Output

Using external information to calibrate the model’s output distribution is also an efficient yet effective test-time adaptation method (Khandelwal et al. 2020). It operates in the output probability space, defining the objective to align the distributions of model predictions and references, and executing the optimization via weighted interpolation. AdaNPC (Zhang et al. 2023c) designs a memory pool to store training data. During inference, given a test sample, AdaNPC recalls  $k$  samples from the memory pool and uses a  $k$ NN classifier to predict the test sample. It then stores the test sample and its predicted label in the memory pool. Over time, the sample distribution in the memory pool gradually aligns with the test distribution. In NLP, the most representative application of such methods is  $k$ NN machine translation.  $k$ NN-MT (Khandelwal et al. 2021) constructs a datastore to store contextual representations and their corresponding target tokens. During translation inference, it retrieves the  $k$ -nearest candidate tokens from the datastore based on the decoded context and processes them into probabilities. Finally, it calibrates the translation model’s probability distribution by performing a weighted fusion of the model’s probabilities and the retrieved probabilities.  $k$ NN-MT has demonstrated superior transferability and generalization compared to traditional models in cross-domain and multilingual MT tasks. Subsequent studies have focused on improving its performance and efficiency (Wang et al. 2022a; Zhu et al. 2023; You et al. 2024) or applying its methods to other NLP tasks (Wang et al. 2022b; Bhardwaj et al. 2023).

Takeaways: Output-calibration TTA

- **Goal:** Output-calibration TTA improves robustness and generalization at test time by adjusting the output probability distribution using external information.
- **Methods:** Typical methods build external datastores to retrieve nearest neighbors and fuse retrieved distributions with the model’s original predictions.
- **Challenges & limitations:** Output-calibration TTA shares similar limitations with input-modification TTA, and its practical value is diminishing as models’ in-context learning capabilities improve.

*Summary 1: Parameter updating and output calibration are the most versatile TTA methods. However, parameter updating suffers from training instability and inefficiency in LLMs, while output calibration relies on target domain information and risks knowledge leakage. Input modification and representation editing are free from training but have limited applicability: Input modification is related to ICL capabilities, and representation editing demands prior knowledge. To complement the qualitative summary above, table 2 provides a systematic comparison of the four TTA paradigms across key practical dimensions.*

4. Test-time Reasoning for System-2 Thinking

Test-time reasoning aims to spend more inference time to search for the most human-like reasoning process within the vast decoding search space. In this section, we introduce the two core components of test-time reasoning: feedback modeling and search strategies (as shown in Figure 4). Under the unified framework, feedback modeling defines the objective of test-time reasoning, while search strategies carry out the optimization.

4.1 Feedback Modeling

*Discriminative-based Feedback.* Discriminative-based feedback, also known as the verifier, aims to predict scalar scores to evaluate the alignment of generated results with ground truth or human cognitive processes (Table 3). Its training process is typically similar to the reward model in reinforcement learning from human feedback (RLHF), using various forms of feedback signals and modeling it as a classification (Cobbe et al. 2021) or rank task (Bradley and Terry 1952; Yuan et al. 2024a; Hosseini et al. 2024). In reasoning tasks, verifiers are primarily categorized into two types: outcome-based

**Table 2**  
Comparative analysis of test-time adaptation paradigms across key practical dimensions.

| Dimension       | Parameter Updating | Input Modification       | Representation Editing | Output Calibration |
|-----------------|--------------------|--------------------------|------------------------|--------------------|
| Search Space    | Model parameters   | Demonstration space      | Representation space   | Output probability |
| Compute         | High               | Low–Medium               | Low                    | Low                |
| Latency         | High               | Medium                   | Low                    | Low                |
| Memory          | High               | Low–Medium               | Low                    | Medium–High        |
| Training Access | TTT: Yes; FTTA: No | No                       | No                     | No                 |
| Task Generality | Broad              | LLMs with ICL capability | Generative LLMs        | Broad              |

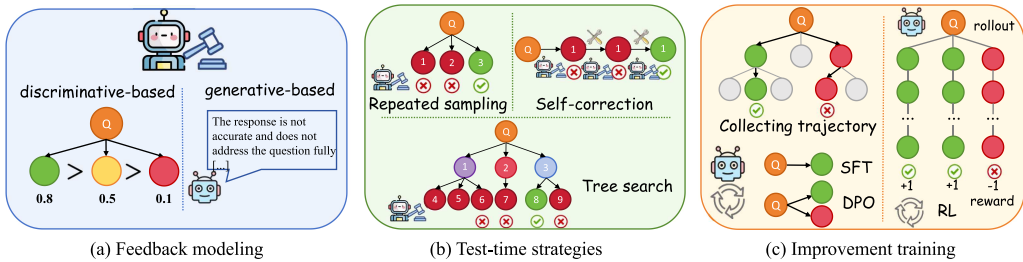


**Figure 4** Illustration of feedback modeling, search strategies, and improvement training in test-time reasoning.

**Table 3** Overview of feedback modeling methods.

| Category       | Sub-category   | Representative Methods             | Domain     | Objective                   | Description                                                |
|----------------|----------------|------------------------------------|------------|-----------------------------|------------------------------------------------------------|
| Discriminative | ORM            | Cobbe et al. (2021)                | Math       | classification              | Human annotated data                                       |
|                |                | Acemath (Liu et al. 2025h)         | Math       | list-wise Bradley-Terry     | Sampling training data from multiple LLMs                  |
|                | PRM            | Lightman et al. (2024)             | Math       | classification              | Human annotated data                                       |
|                |                | Math-shepherd (Wang et al. 2024h)  | Math       | classification              | Annotating processes via MC estimation                     |
|                |                | Zhang et al. (2025d)               | Math       | classification / regression | Annotating processes via MCTS and LLM-as-a-judge           |
| Generative     | Training-free  | Implicit PRM (Yuan et al. 2024b)   | Math       | implicit reward modeling    | Training PRMs with outcome labels                          |
|                |                | ASPRM (Liu et al. 2025f)           | Math, Code | classification              | Adaptive segment; annotating processes via MC estimation   |
|                | Training-based | LLM-as-a-Judge (Zheng et al. 2023) | General    | –                           | Designing system instructions to mitigate biases           |
|                |                | BSM (Saha et al. 2024)             | General    | –                           | Dividing into multiple criteria and then merging           |
|                |                | Shepherd (Wang et al. 2023b)       | General    | SFT                         | Collecting data from human annotation and the Internet     |
|                |                | Prometheus (Kim et al. 2024c)      | General    | SFT                         | Training single and pairwise models and then merging them  |
|                |                | EvalPlanner (Saha et al. 2025)     | General    | DPO                         | Planning evaluation processes and then evaluating          |
|                |                | GenRM (Zhang et al. 2024g)         | Math       | SFT                         | Synthesizing critique data from external LLMs              |
|                |                | R-PRM (She et al. 2025)            | Math       | SFT & DPO                   | Synthesizing critique data from external LLMs              |
|                |                | Critic-RM (Yu et al. 2024c)        | General    | SFT & Bradley-Terry         | Synthesizing and filtering critique data via self-critique |
|                |                | C.Cloud (Ankner et al. 2024)       | General    | SFT & Bradley-Terry         | Synthesizing data from external LLMs and self-critique     |

verifiers (ORMs) and process-based verifiers (PRMs). ORM (Cobbe et al. 2021) use the correctness of the final CoT result as training signals. Liu et al. (2025h) provide a detailed recipe for training a strong ORM. In contrast, PRMs (Uesato et al. 2022; Lightman et al. 2024; Zhang et al. 2024e) are trained based on the correctness of each reasoning step. Compared with ORMs, PRMs cannot only evaluate intermediate reasoning steps but also assess the entire reasoning process more precisely. However, PRMs require more human effort to annotate feedback for the intermediate steps. Math-Shepherd (Wang et al. 2024h), OmegaPRM (Luo et al. 2024), and EpicPRM (Sun et al. 2025) utilize MCTS algorithm to collect high-quality process supervision data automatically. Zhang et al. (2025d) utilize critic models to evaluate process annotations collected by MCTS, filtering out low-quality data to improve the training effectiveness of PRMs. Setlur et al. (2024) argue that PRMs should evaluate the advantage of each step for subsequent reasoning rather than focusing solely on its correctness. They propose process advantage verifiers (PAVs) and efficiently construct training data through MCTS. Furthermore, Lu et al. (2024) and Yuan et al. (2024b) notice that ORMs implicitly model the advantage of each step, leading them to automatically annotate process supervision data using ORMs or directly train PRMs on outcome labels. Liu et al. (2025f) adaptively segment reasoning steps based on tokens with higher uncertainty, which is used to train better PRMs.

### Takeaways: Discriminative-based Feedback

- **Goal:** Discriminative-based feedback assigns numerical scores to model outputs to evaluate their correctness, instruction following, and alignment with human preferences, thereby providing guidance signals for test-time reasoning.
- **Methods:** Main approaches include ORMs that score the overall answers and PRMs that assess intermediate steps, trained via classification or ranking with human or automatic annotation data.
- **Challenges & limitations:** Discriminative feedback, especially PRMs, faces severe scalability challenges. A key open problem is how to obtain large amounts of high-quality scoring data at low cost to train stronger and more generalizable feedback models.

*Generative-based Feedback.* Although the verifier can evaluate the correctness of generated answers or steps, it lacks interpretability, making it unable to locate the specific cause of errors or provide correction suggestions. Generative-based feedback, also referred to as critic, fully leverages the LLM's generative and instruction-following ability (Ankner et al. 2024; Ye et al. 2024; Chiang, Lee, and Lukasik 2025; She et al. 2025). By designing specific instructions, it can perform pointwise or pairwise evaluation from multiple dimensions, and even provide suggestions for revision in natural language. Powerful closed-source LLMs, such as GPT-4 and Claude, are effective critics. They can perform detailed and controlled assessments of generated texts, such as factuality, logical errors, coherence, and alignment, with high consistency with human evaluations (Wang et al. 2023a; Luo, Xie, and Ananiadou 2023; Liu et al. 2023; Chiang and Lee 2023). However, they still face biases such as length, position, and perplexity (Bavaresco et al. 2024; Wang et al. 2024g; Stureborg, Alikaniotis, and Suhara 2024). LLM-as-a-Judge (Zheng et al. 2023) carefully designs system instructions to mitigate the interference of biases. BSM (Saha et al. 2024) evaluates based on multiple criteria and then merges them. Peng et al. (2025a) employ multi-agents to jointly evaluate answers' factuality and instruction-following.

To obtain cheaper generative-based feedback, open-source LLMs can also serve as competitive alternatives through supervised fine-tuning (SFT) (Wang et al. 2024k; Zhu, Wang, and Wang 2023; Liang et al. 2024b; Paul et al. 2024). Shepherd (Wang et al. 2023b) collects high-quality training data from human annotation and online communities to fine-tune an evaluation model. Auto-J (Li et al. 2024b) collects queries and responses from various scenarios and designs evaluation criteria for each scenario. GPT-4 then generates critiques of the responses based on these criteria and distills its critique ability to open-source LLMs. Prometheus (Kim et al. 2024b, 2024c) designs more fine-grained evaluation dimensions. It trains a single evaluation model and a pairwise ranking model separately, then unifies them into one LLM by weight merging. To reduce reliance on human annotations and external LLMs, Wang et al. (2024i) propose a self-training method: The critique model generates positive and negative responses, then collects critique data via rejection sampling to perform iterative fine-tuning. Building on self-training, EvalPlanner enables (Saha et al. 2025) the critique model to plan evaluation processes and criteria, conduct critiques based on these, and then collect positive and negative samples to improve the critique model via direct preference optimization (DPO) (Rafailov et al. 2023). Yu et al. (2025a) carefully synthesize and

filter data, enabling the base model to achieve strong critique abilities with only 40K samples for SFT and DPO training. GenRM (Zhang et al. 2024g) leverages instruction tuning to enable the verifier to answer “Is the answer correct (Yes/No)?” and uses the probability of generated “Yes” token as the score. GenRM can also incorporate CoT, allowing the verifier to generate the corresponding rationale before answering “Yes” or “No”. ThinkPRM (Khalifa et al. 2025) evaluates each reasoning step with long CoT and requires only 1% of the process supervision data compared to discriminative PRMs. Critic-RM (Yu et al. 2024c) jointly trains the critic and the verifier. Compared with discriminative reward models, generative reward models have greater potential for generalization, which is crucial for building generalizable System-2 models. We discuss this in detail in Section 5.1.

Takeaways: Generative-based Feedback

- **Goal:** Generative-based feedback aims to provide interpretable and fine-grained supervision by generating natural-language critiques, explanations, and revision suggestions, offering richer guidance signals for test-time reasoning.
- **Methods:** Typical approaches utilize LLMs as critics to perform multi-criteria evaluation via prompting. Open-source large models can enhance their generative critique capabilities through SFT and preference learning.
- **Challenges & limitations:** Generative feedback is sensitive to prompt design, and its evaluation quality depends on prior criteria, making it prone to inconsistency, hallucinations, and implicit bias.

4.2 Search Strategies

Table 4 lists the different search strategies discussed in this section.

*4.2.1 Repeated Sampling.* Sampling strategies such as top- $p$  and top- $k$  are commonly used decoding algorithms in LLM inference. They introduce randomness during decoding to enhance text diversity, allowing for parallelly sampling multiple generated texts. Through repeated sampling, we have more opportunities to find the correct answer. From a unified perspective, repeated sampling operates in the decoding trajectory space, with objectives defined by self-consistency or external feedback, and the optimization is implemented via majority voting or selection. Repeated sampling is particularly suitable for tasks that can be automatically verified, such as code generation, where we can easily identify the correct solution from multiple samples using unit tests (Li et al. 2022a; Rozière et al. 2024). For tasks that are difficult to verify, like math word problems, the key to the effectiveness of repeated sampling is the verification strategy.

*Verification Strategy.* Verification strategies include two types: majority voting and best-of- $N$  (BoN) sampling. **Majority voting** (Li et al. 2024c; Lin et al. 2024a) selects the most frequently occurring answer in the samples as the final answer, which is motivated by ensemble learning. Majority voting is simple yet effective. For instance, self-consistency CoT (Wang et al. 2023d) can improve accuracy by 18% over vanilla CoT in math reasoning tasks. However, the majority does not always hold the truth, as they may make similar mistakes. Therefore, some studies perform filtering or weighting before

**Table 4**  
Overview of search strategies.

| Category        | Sub-category       | Representative Methods               | Tasks                             | Verifier/Critic           |
|-----------------|--------------------|--------------------------------------|-----------------------------------|---------------------------|
| Repeat Sampling | Majority voting    | CoT-SC (Wang et al. 2023d)           | Math, QA                          | self-consistency compiler |
|                 |                    | PROVE (Toh, Ghosal, and Poria 2024)  | Math                              |                           |
|                 | Best-of-N          | Cobbe et al. (2021)                  | Math                              | ORM                       |
|                 |                    | DiVeRSe (Li et al. 2023c)            | Math                              | PRM                       |
| Self-correction | Human feedback     | Knockout (Liu et al. 2025d)          | Math                              | critic                    |
|                 |                    | LOVER (Wang et al. 2025c)            | General                           | logical consistency       |
|                 | External tools     | NL-EDIT (Elgohary et al. 2021)       | Semantic parsing                  | Human                     |
|                 |                    | FBNET (Tandon et al. 2022)           | Code                              | Human                     |
|                 | External models    | DrRepair (Yasunaga and Liang 2020)   | Code                              | compiler                  |
|                 |                    | Self-debug (Chen et al. 2024c)       | Code                              | compiler                  |
|                 | Intrinsic feedback | CRITIC (Gou et al. 2024)             | Math, QA, Detoxifying             | text-to-text APIs         |
|                 |                    | REFINER (Paul et al. 2024)           | Math, Reason                      | critic model              |
|                 | Uninformed search  | Shepherd (Wang et al. 2023b)         | QA                                | critic model              |
|                 |                    | Multi-agent Debate (Du et al. 2023b) | Math, Reason                      | multi-agent debate        |
| Tree Search     | Heuristic search   | MAD (Liang et al. 2024a)             | Translation, Math                 | multi-agent debate        |
|                 |                    | Self-Refine (Madaan et al. 2023)     | Math, Code, Controlled generation | self-critique             |
|                 | Uninformed search  | Reflexion (Shinn et al. 2023)        | QA                                | self-critique             |
|                 |                    | RCI (Kim, Baldi, and McAleer 2023)   | Code, QA                          | self-critique             |
|                 | Heuristic search   | ToT (Yao et al. 2023)                | Planning, Creative writing        | self-critique             |
|                 |                    | Xie et al. (2023)                    | Math                              | self-critique             |

voting. For example, the PROVE framework (Toh, Ghosal, and Poria 2024) converts CoT into executable programs, filtering out samples if the program’s results are inconsistent with the reasoning chain’s outcomes. Kang, Zhao, and Song (2025) propose the self-certainty metric to weight votes based on their ranking. Huang et al. (2025a) use the calibrated confidence as vote weights, and adjust the number of samples based on response agreement to improve efficiency (Aggarwal et al. 2023). RPC (Zhou et al. 2025) demonstrates theoretically and experimentally that filtering low-probability samples improves the performance and efficiency of majority voting.

**Best-of-N sampling (BoN)** uses a verifier to score each response and selects the one with the highest score as the final answer (Stiennon et al. 2020; Cobbe et al. 2021; Nakano et al. 2022; Liu et al. 2025a). Li et al. (2023c) propose a voting-based BoN variant, which performs weighted voting on all answers based on the verifier’s scores and selects the answer with the highest score. Liu et al. (2025d) design BoN in a knockout tournament, using pairwise comparison verifiers to filter out the best response. In addition, some works aim to improve the efficiency of BoN. Inspired by speculative decoding, Zhang et al. (2024i), Qiu et al. (2024a), Sun et al. (2024), Yu, Li, and Wang (2025), and Manvi, Singh, and Ermon (2024) evaluate each reasoning step and prune low-scoring sampled results, halting further generation for those paths, thereby significantly reducing the overall time cost. PRS (Ye and Ng 2024) enables LLMs to self-critique and self-correct, guiding the model to generate expected responses with fewer sampling times. Li et al. (2025a) compare BoN and majority voting, demonstrating that BoN is suitable for harder questions with moderate diversity of response distribution. As discussed in Section 4.1, existing feedback modeling suffers from generalization bottlenecks on general tasks, making the guidance signals for BoN unreliable. Consequently, a line of work attempts to explore using intrinsic LLM states to select the best response. For

example, CoT-decoding (Wang and Zhou 2024) branches reasoning paths on top- $k$  early tokens and then selects the most reliable one via the probability confidence margin. LOVER (Wang et al. 2025c) constructs contrastive assertions over multiple reasoning paths and trains a discriminator from the model’s hidden states to predict assertion truth by the self-supervised logical consistency objective (Burns et al. 2023b).

*Improvement Training.* Repeated sampling has proven to be a simple yet effective method, even surpassing models fine-tuned with RLHF (Gao, Schulman, and Hilton 2023; Hou et al. 2024). However, it costs so much inference time that it is difficult to afford in practical applications. Therefore, many studies have attempted to train the model by BoN sampling to approximate the BoN distribution, thereby reducing the search space during inference. ReST (Gulcehre et al. 2023) samples responses with reward values above a threshold from the policy model as self-training data and fine-tunes the policy model by offline reinforcement learning. In each iteration, ReST samples new training data. vBoN (Amini, Vieira, and Cotterell 2024), BoNBoN (Gui, Garbacea, and Veitch 2024) and BOND (Sessa et al. 2024) derive the BoN distribution and minimize the difference between the policy model’s distribution and the BoN distribution. Chow et al. (2024) design a BoN-aware loss to make the policy model more exploratory during fine-tuning.

Takeaways: Repeated Sampling

- **Goal:** Repeated sampling aims to improve performance by generating diverse candidates and obtaining the best answer through verification strategies.
- **Methods:** Core strategies include majority voting and best-of-N sampling. Recent work improves efficiency via step-level pruning, adaptive sampling, or aligning the model with the best-of-N distribution.
- **Challenges & limitations:** Repeated sampling is inference-expensive and heavily depends on reliable verification. Majority voting can amplify shared errors, while BoN suffers from verifier generalization bottlenecks.

4.2.2 *Self-correction.* Repeated sampling in parallel expands the search space, while for sufficiently difficult problems, it may fail to find the correct answer due to insufficient search depth. In contrast, self-correction (Figure 5) is a sequential test-time compute method that enables LLMs to expand search depth by iteratively revising and refining based on external or internal feedback (Shinn et al. 2023).

*Feedback Sources.* The feedback used for self-correction is typically presented in natural language and comes from various sources, including human evaluation, tool checking, external model evaluation, and intrinsic feedback. *Human evaluation* is the gold standard for feedback, but due to its high cost and limited scalability, it is mainly used in early research to explore the upper limits of self-correction capabilities (Tandon et al. 2021; Elgohary et al. 2021; Tandon et al. 2022). For certain domain-specific tasks, *external tool checking* provides accurate and efficient feedback (Chen et al. 2024c; Gao et al. 2023). For example, Yasunaga and Liang (2020) propose to obtain feedback from compilers in code repair and generation tasks. CRITIC (Gou et al. 2024) enables LLMs to invoke external tools for verification, generate natural-language critiques based on tool feedback, and

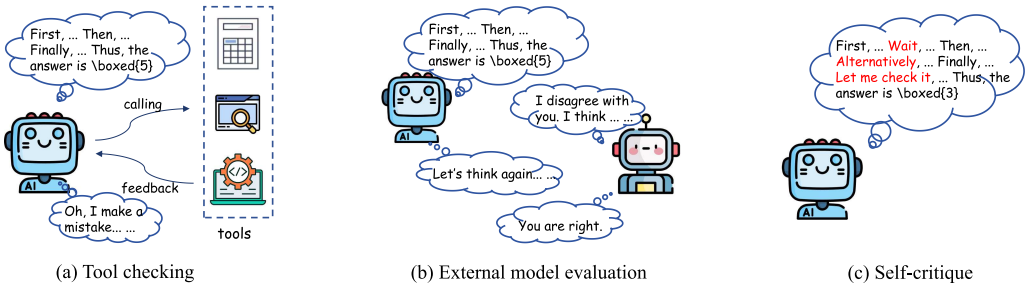


**Figure 5**  
Illustration of self-correction with tool checking, external model evaluation, and self-critique.

iteratively refine their answers. In embodied tasks, the environment can provide precise feedback on the action trajectories of LLM-based agents (Wang et al. 2024b).

*External model evaluation* is an effective feedback source for general tasks, such as various verbal-based critique models described in Section 4.1. For example, Paul et al. (2024) first define multiple error types for natural language reasoning tasks and then design the corresponding feedback templates. They train an evaluation model using synthetic feedback training data, and with the critic, the reasoning model achieves substantial performance improvement. Multi-agent debate (Du et al. 2023b; Xiong et al. 2023; Liang et al. 2024a; Chen, Saha, and Bansal 2024; Wang et al. 2024f) is another mechanism that leverages external feedback to enhance reasoning capabilities. In this approach, models do not have distinct roles as reasoners and critics. Instead, multiple models independently conduct reasoning, critique each other, and defend or refine their reasoning based on feedback. This process continues until agents reach a consensus or a judge model summarizes the final reasoning results. The multi-agent debate has shown its potential in fact-checking (Kim et al. 2024a; Khan et al. 2024), commonsense QA (Xiong et al. 2023), faithful evaluations (Chan et al. 2024), and complex reasoning (Du et al. 2023b). However, multi-agent debate may be unstable, as LLMs are susceptible to adversarial information and may revise correct answers to incorrect ones in response to misleading inputs (Laban et al. 2024; Amayuelas et al. 2024). Therefore, a successful multi-agent debate requires that LLMs maintain their stance when faced with incorrect answers from other models while remaining open to valid suggestions (Stengel-Eskin, Hase, and Bansal 2024). In general, the more LLMs involved in the debate, the stronger the overall reasoning performance. However, this significantly increases the number of LLM inferences required, and the length of input context, posing a major challenge to LLM inference costs (Liu et al. 2024d). To reduce debate inference costs, Li et al. (2024g) investigate the impact of topological connections among multiple agents and show that sparse connections, such as ring structures, are not inferior to the fully connected topology. GroupDebate (Liu et al. 2024d) divides LLMs into groups that conduct debates internally and only share the consensus results between groups.

*Self-critique* assumes that LLMs can self-evaluate their outputs and optimize them through intrinsic feedback (Yuan et al. 2024c). This idea stems from a fundamental principle in computational complexity theory: verifying whether a solution is correct is typically easier than solving the problem. Bai et al. (2022) propose self-correcting harmful responses from LLMs by prompting themselves. Self-Refine (Madaan et al. 2023) and RCI Prompting (Kim, Baldi, and McAleer 2023) iteratively prompt LLMs to self-correct their responses in tasks such as arithmetic reasoning. IoE (Li et al. 2024d)

observes that LLMs may over-criticize themselves during self-critique, leading to performance degradation, and designs prompt to guide LLMs in assessing confidence. ProgCo (Song et al. 2025b) leverage the advantages of code in expressing complex logic, enabling LLMs to generate responses in pseudo-code form, followed by self-critique and refinement. SETS (Chen et al. 2025b) combines the strengths of repeated sampling and self-critique, applying self-critique and correction to each sampled reasoning path and choosing the final solution via majority voting. S1 (Muennighoff et al. 2025) adds the “wait” to the reasoning process, prompting the LLM to critique its reasoning.

*Arguments.* The effectiveness of self-correction, especially self-critique, has remained controversial. Several empirical studies on code generation (Olausson et al. 2024), commonsense QA (Huang et al. 2024a), math problem-solving (Wang et al. 2024e), planning (Valmeekam, Marquez, and Kambhampati 2023), and graph coloring (Stechly, Marquez, and Kambhampati 2023) confirm that self-correction is not a guaranteed solution for improving performance.

Kamoi et al. (2024) think the effectiveness of self-correction has been overestimated. Previous successes either rely on oracle answers or weak initial answers. Only tasks that can be broken down into easily verifiable sub-tasks can truly benefit from self-correction. They suggest fine-tuning specific evaluation models to achieve better self-correction. Zhang et al. (2024h) try to interpret and alleviate the failure of self-critique via human-like cognitive bias. Tyen et al. (2024) decouple the abilities of LLMs to identify and correct errors and create the corresponding evaluation datasets. The evaluation results show that LLMs do not lack the ability to correct errors during self-correction, and their main performance bottleneck lies in locating the errors. Yang et al. (2024b) decompose self-critique into confidence and critique capabilities. Empirical studies show that fine-tuning is necessary to enhance both capabilities simultaneously, while prompt engineering can only achieve a trade-off.

*Improvement Training.* Most of the aforementioned self-correction methods demonstrate significant performance improvements on advanced closed-source large models or open-source LLMs with over 70B parameters. However, for medium-scale open-source models with weaker capabilities, we need to further fine-tune them to unlock their self-correction capabilities. Supervised fine-tuning optimizes the model using high-quality multi-turn correction data, either manually annotated (Saunders et al. 2022), self-rationalized (Zelikman et al. 2022; Yuan et al. 2025c), multi-agent debate (Subramaniam et al. 2025) or sampled from stronger LLMs (An et al. 2023; Paul et al. 2024; Qu et al. 2024; Gao et al. 2024c; Zhang et al. 2024l; Xi et al. 2024). GLoRe (Havrilla et al. 2024) considers that LLMs need global or local refinement for different types of errors. To address this, they construct training sets for global and local refinement, train verifiers to identify global and local errors, and develop LLMs for refinement based on different global or local feedback signals. Xi et al. (2024) design a scalable framework for synthesizing self-correction training data, enabling reasoning models to generate controlled errors and receive feedback from critics to self-correct. Although SFT is effective, training data from offline-generated self-correction trajectories can only simulate limited correction patterns. This leads to the distribution mismatch with the actual self-correction behavior during model inference. Self-correct (Welleck et al. 2023) adopts online imitation learning, re-sampling new self-correction trajectories for training after each training epoch. To further expand the exploration space of LLMs, many studies adopt flexible RL algorithms to surpass the performance limits of SFT. SCoRe (Kumar et al. 2024) proposes

using the multi-turn RL method to improve self-critique and self-correction capability. T1 (Hou et al. 2025) employs self-correction training data for SFT cold-start, followed by RL training using the RLOO algorithm (Ahmadian et al. 2024). During the RL phase, high-temperature sampling and entropy rewards encourage the LLM to explore more diverse reasoning paths.

#### Takeaways: Self-correction

- **Goal:** Self-correction aims to improve LLM reasoning at test time by iteratively revising responses using external or intrinsic feedback.
- **Methods:** Feedback can come from diverse sources, including human evaluation, external tool checking, external evaluators, multi-agent debate, and the model's own self-critique. SFT and reinforcement learning can effectively enhance the model's self-criticism and self-correction capabilities.
- **Challenges & limitations:** Self-correction is not always effective. LLMs still struggle to accurately localize errors, may over-correct or be misled by incorrect feedback, and depend heavily on feedback quality and task verifiability.

**4.2.3 Tree Searching.** Repeated sampling and self-correction scale test-time compute in parallel and sequentially, respectively. Human thinking is a tree search that combines brainstorming in parallel with backtracking to find other paths to solutions when it encounters a dead end. We model the search space  $\mathcal{S}$  as a decoding tree, and navigating this space requires an optimization strategy  $\mathcal{A}$  executed via breadth-first search (BFS), depth-first search (DFS), or MCTS, which is guided by evaluation objectives  $\mathcal{J}$  based on local or long-term action value. Search algorithms and value functions are two critical components in tree searching.

**Search Algorithm.** In LLM reasoning, current search algorithms include uninformed search and heuristic search. Uninformed search explores the search space according to a fixed rule. For example, tree-of-thought (ToT) (Yao et al. 2023) adopts the BFS or DFS to search, while Xie et al. (2023) use beam search. Uninformed search is usually less efficient for problems with large search spaces, so heuristic search strategies represented by  $A^*$  (Meng et al. 2024; Wang et al. 2024a) and MCTS (Hao et al. 2023; Bi et al. 2024; Park et al. 2024) are widely used in reasoning tasks. MCTS, which eliminates the need for explicit heuristics, leverages stochastic simulations and adaptive tree expansion under uncertain environments, making it well-suited for large state spaces. It optimizes search results gradually through four steps: selection, expansion, simulation, and backpropagation, approaching the optimal solution. In contrast,  $A^*$  uses a heuristic function-guided deterministic search to guarantee optimal paths, but its performance depends on the design of the heuristic function. As a result, MCTS has been successfully applied to tasks such as RAG (Hu et al. 2024a; Jiang et al. 2024; Li and Ng 2024; Feng et al. 2025), QA (Luo et al. 2025a; Gan et al. 2025), hallucination mitigation (Cheng et al. 2025), text-to-SQL (Yuan et al. 2025b), etc. Additionally, Long (2023) trains an LLM controller using reinforcement learning to guide the LLM reasoner's search path, and Chari et al. (2025) utilize an ant colony evolutionary algorithm to guide tree search.

**Value Function.** The value function evaluates the value of each state and guides the tree to expand towards branches with higher values in heuristic tree search. Xu (2023) trains an energy function by noise-contrastive estimation as the value function. RAP (Hao et al.

2023) designs a series of heuristic value functions, including the likelihood of the action, the confidence of the state, self-evaluation results, and task-specific reward, and combines them according to task requirements. Reliable and generalized value functions facilitate the application of MCTS to more complex problems with deeper search spaces. AlphaMath (Chen et al. 2024a) and TS-LLM (Feng et al. 2024b) replace the hand-crafted value function with a learned LLM value function, automatically generating reasoning process and step-level evaluation signals in MCTS. VerifierQ (Qi, Tang, and Zhu 2024) integrates implicit Q-learning and contrastive Q-learning to train the value function, effectively mitigating the overestimation issue at the step level. Traditional MCTS methods expand only one trajectory, while rStar (Qi et al. 2024) argues that the current value function struggles to guide the selection of the optimal path accurately. Therefore, rStar retains multiple candidate paths and performs reasoning with another LLM, ultimately selecting the path where both LLMs’ reasoning results are consistent. Gao et al. (2024d) propose SC-MCTS, which combines multiple reward models, including contrastive reward, likelihood, and self-evaluation as value functions. MCTSr (Zhang et al. 2024b) and SR-MCTS (Zhang et al. 2024c) take complete responses as nodes, expanding the search space through self-critique and correction. SR-MCTS utilizes pairwise preference rewards and global quantile score as the value function, offering a more robust value function estimation.

*Improvement Training.* Tree search can guide LLMs to generate long reasoning processes, and these data help train LLMs with stronger reasoning abilities (Zhai et al. 2024; Xu et al. 2024a; Guan et al. 2025). ReST-MCTS\* (Zhang et al. 2024a) uses process rewards as a value function to guide MCTS, collecting high-quality reasoning trajectories and the value of each step to improve the policy model and reward model. Due to the step-by-step exploration of tree search, it can obtain finer-grained step-level feedback signals. MCTS-DPO (Xie et al. 2024) collects step-level preference data through MCTS and uses DPO for preference learning. AlphaLLM-CPL (Wang et al. 2024j) ranks trajectories based on preference reward gaps and policy prediction gaps, utilizing curriculum learning to efficiently utilize MCTS-collected trajectories. Recently, many LRMs (Qin et al. 2024b; Zhao et al. 2024b; Zhang et al. 2024k) have also confirmed the necessity of using tree search to construct high-quality long reasoning chain data for training.

Takeaways: Tree search

- **Goal:** Tree search aims to scale test-time compute by exploring multiple reasoning paths, combining parallel exploration and backtracking to sample higher-quality long reasoning trajectories.
- **Methods:** The main approaches include unguided and heuristic tree search. Most current methods are based on MCTS, combined with learnable value functions to guide tree search. The high-quality reasoning trajectories sampled from tree search can be further used for post-training.
- **Challenges & limitations:** Tree search heavily depends on the accuracy and generalization of the value function. Inaccurate value estimating misguides the search and limits its scalability to deeper search spaces.

*Summary 2: Repeated sampling is easy to implement and improves answer diversity, making it suitable for open-ended or easily verifiable tasks, though computationally inefficient. Self-correction relies on precise, fine-grained feedback and works well for easily verifiable tasks,*

**Table 5**  
Comparative analysis of test-time reasoning strategies across key practical dimensions.  
*N*: number of samples, *K*: correction rounds, *b*: branching factor, *d*: search depth.

| Dimension              | Repeated Sampling                                                     | Self-Correction                                              | Tree Search                                                  |
|------------------------|-----------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| Compute Scaling        | $O(N)$ parallel                                                       | $O(K)$ sequential                                            | $O(b^d)$ or $O(N \cdot d)$                                   |
| Latency                | Low (parallelizable)                                                  | Medium (sequential rounds)                                   | High (sequential branching)                                  |
| Task Suitability       | Easily verifiable tasks                                               | Domain-specific with clear feedback                          | Complex multi-step reasoning                                 |
| External Resources     | Verifier or no required                                               | External feedback                                            | Step-level value model                                       |
| Ease of Implementation | Simple                                                                | Moderate                                                     | Complex                                                      |
| Key Limitation         | Verifier dependency for BoN;<br>Diminishing returns as <i>N</i> grows | Error localization bottleneck;<br>Over-correct valid answers | Value function accuracy and<br>generalization across domains |

*but may not perform well with poor feedback or weak reasoning capability. Tree search optimizes complex planning tasks globally but involves complex implementation. To help practitioners navigate the landscape of test-time reasoning strategies, table 5 provides a comparative analysis across key practical dimensions.*

5. Frontier Topics in Test-time Compute Toward Real-world Applications

Although test-time compute models demonstrate strong performance on specialized tasks such as mathematics and coding, deploying them in complex real-world scenarios remains challenging. This section focuses on four key directions: building System-2 models with cross-domain generalization capabilities, enhancing test-time scaling through reinforcement learning, integrating multimodal capabilities for comprehensive understanding, and achieving a balance between performance and cost.

5.1 Generalizable System-2 Model

Currently, most LRMs exhibit strong reasoning performance in specific domains such as math and code, but they struggle to generalize to cross-domain, cross-lingual, or general tasks (Son et al. 2025; Zhao et al. 2025b). On the one hand, as the foundation of current System-2 models, CoT shows little effectiveness in non-symbolic reasoning tasks (Sprague et al. 2024). On the other hand, verifiers and critics have limited generalization capabilities, making it difficult to provide effective guidance to the reasoning model (LeVine et al. 2023; Kim et al. 2024d; Chen et al. 2024b). For the former limitation, integrating external tools and multi-agent feedback is a promising direction (Nathani et al. 2023; Lan et al. 2024). Many works aim to enable reasoning models to learn to perform retrieval or tool use during the reasoning process. Search-o1 (Li et al. 2025b) triggers external search tools by generating special tokens that represent search actions, and Search-R1 (Jin et al. 2025) further enhances the interaction between the LLM and search tools through RL. Deep Research (OpenAI 2025; Google 2024; Zheng et al. 2025b; Li et al. 2025c) integrates various tools such as Web search and code execution, demonstrating distinguished performance on general tasks like information synthesis, report writing, and complex code generation.

For the latter challenge, some studies utilize multi-objective training (Wang et al. 2024c), model ensemble (Lin et al. 2024b), soft reward (Su et al. 2025), or regularization constraints (Yang et al. 2024a; Jia 2024) to make verifiers more generalizable. Compared with discriminative verifiers, generative critics inherit the generalization ability of LLMs and can leverage test-time compute to realize greater potential (Kim et al. 2025).

GenPRM (Zhao et al. 2025a) and GRM (Liu et al. 2025i) leverage repeated sampling for test-time compute, and GRM trains a Meta RM to evaluate the quality of the critic’s outputs. RM-R1 (Chen et al. 2025g) and JudgeLRM (Chen et al. 2025c) use RL training to enable the critic with self-correction ability.

Additionally, weak-to-strong generalization (Burns et al. 2023a) is a topic worth further exploration. People are no longer satisfied with solving mathematical problems with standard answers; they hope System-2 models can assist in scientific discovery and the proofs of mathematical conjectures. In such cases, even human experts struggle to provide accurate feedback, while weak-to-strong generalization offers a promising direction to address this issue (Tang et al. 2025).

## 5.2 Reinforcement Learning for Incentivizing Test-time Compute

With the success of OpenAI-o1 and DeepSeek-R1, researchers have increasingly recognized that RL-based improvement training demonstrates stronger test-time scalability in practical applications, offering substantial research and practical value. Compared with SFT, RL often yields a higher performance ceiling and stronger multi-task generalization (Shenfeld, Pari, and Agrawal 2025).

*Simplifying the Training Infrastructure.* A clear trend in RL for reasoning is the progressive removal of auxiliary components to improve scalability. PPO (Schulman et al. 2017) is a classical actor-critic algorithm that estimates advantages via a learned value function and applies a clipped trust-region policy gradient update. However, in long CoT settings, the value model often becomes a performance bottleneck when rewards are sparse, noisy, or length-heterogeneous. Several concurrent approaches have emerged to eliminate the critic network while adopting different baseline strategies for variance reduction. GRPO (Shao et al. 2024) removes the explicit critic network and instead computes relative advantages within a group of sampled responses under the same prompt, significantly simplifying large-scale RL for reasoning models. REINFORCE++ (Hu et al. 2025a) takes an alternative critic-free approach by combining PPO’s tricks, achieving comparable or superior stability to GRPO while maintaining the simplicity of the classical REINFORCE framework. Similarly, RLOO (Ahmadian et al. 2024) revisits REINFORCE-style optimization with a leave-one-out baseline estimator, providing an unbiased and low-variance advantage estimate without any learned value function. This family of algorithms illustrates a consistent trend toward simpler training pipelines: By removing the critic network and adopting lightweight baseline strategies, RL training can scale to larger models with substantially reduced memory overhead and engineering complexity.

*Stabilizing Long-horizon Optimization.* Long CoT sequences pose unique optimization challenges, such as sparse and delayed rewards, heterogeneous sequence lengths, and high variance in advantage estimates that require designing new RL techniques. VAPO (Yue et al. 2025b) augments PPO within a value-based framework tailored for reasoning models, explicitly targeting value-model bias and reward sparsity in long-CoT RL. VRPO (Zhu et al. 2025) improves the robustness of policy optimization under noisy supervision by enhancing the value model’s noise filtering and advantage estimation capabilities. DAPO (Yu et al. 2025b) proposes decoupled clipping and dynamic difficulty-aware sampling to stabilize and accelerate long-horizon reinforcement learning. Dr. GRPO (Liu et al. 2025g) identifies that GRPO’s per-response length normalization introduces an optimization bias that systematically favors longer responses,

and proposes an unbiased token-level objective that corrects this bias, improving token efficiency and training stability for long-CoT tasks. Despite these advances, optimization stability remains an active challenge (MiniMax et al. 2025; Zhao et al. 2025c; Su et al. 2026; Wang et al. 2025a). These works share a common finding: Long reasoning sequences amplify small biases in advantage estimation, such as value-model prediction errors, length normalization bias, and noisy group-relative baselines, leading to the unreliable learning signal.

*Architecture- and Granularity-aware Optimization.* As reasoning models increasingly adopt heterogeneous architectures and tackle diverse task formats, RL algorithms must be co-designed with model structure and adapted to the appropriate optimization granularity. GSPO (Zheng et al. 2025a) replaces token-level importance ratios with sequence-likelihood-based metrics and adopts sequence-level truncation and optimization strategies, enhancing training stability and efficiency, especially for mixture of experts (MoE) models. At a different point on the granularity spectrum, Sullivan and Koller (2026) show that GRPO with an outcome reward can be interpreted as implicitly performing process-reward-aware optimization, and based on this analysis propose  $\lambda$ -GRPO to mitigate a bias caused by imbalanced process steps.

*Does RL Truly Extend Reasoning Frontiers, or Merely Sharpen Existing Distributions?* Although RL training has shown significant effectiveness in improving test-time reasoning, its underlying mechanisms remain debated. Empirical studies (Yue et al. 2025a; Setlur et al. 2025) demonstrate that reinforcement learning primarily acts as a form of distribution sharpening, amplifying high-quality reasoning paths that already exist in the model but have low probability, thereby increasing the chance that test-time compute strategies will discover correct answers. In contrast, Liu et al. (2025c) and Yuan et al. (2025a) demonstrate that with sufficiently stable and prolonged RL training, the reasoning frontier of the model can be continually extended, enabling the discovery of novel reasoning patterns. Overall, RL training helps test-time compute more efficiently approach optimal solutions within a vast search space.

### 5.3 Multimodal Test-time Compute

Visual, audio, video, and other modalities are crucial for models' understanding and interaction with the world. Like LLMs, large multimodal models (LMMs) also face challenges in scaling training compute. Thus, enabling efficient test-time scaling for multimodal tasks has become an advanced research topic. In System-1 thinking, TTA has been successfully applied to LMMs, improving performance in tasks such as zero-shot image classification, image-text retrieval, and image captioning (Zhao et al. 2024a). However, extending test-time compute to multimodal reasoning and generation tasks requires rethinking all three components of the unified framework.

For multimodal reasoning tasks, the exploration of multimodal CoT (Zhang et al. 2024m; Wu et al. 2024b; Mondal et al. 2024; Lee et al. 2024; Gao et al. 2024b; Zhang et al. 2024j) opens up the possibility of building multimodal System-2 models. For example, IoT (Zhou et al. 2024b) and SketchPAD (Hu et al. 2024b) utilize visual tools to obtain critical visual information and integrate it into CoT to refine multimodal reasoning. Xu et al. (2024b) divide the visual reasoning process into four stages: task summary, caption, reasoning, and answer conclusion, explicitly defining a stage-level search space for multimodal reasoning. Audio-Reasoner (Xie et al. 2025c) transfers the four-stage CoT framework to the audio modality. For generative tasks, the search space takes

different forms. Mainstream multimodal generative models are divided into diffusion-based (Peebles and Xie 2023; Yang et al. 2025c) and autoregressive-based (Tian et al. 2024; Xie et al. 2025b) models. While diffusion models lack explicit CoTs, they can expand the search space at test time by sampling various Gaussian noises (Xie et al. 2025a). Autoregressive models treat intermediate generated images as reasoning steps (Guo et al. 2025b), naturally defining a sequential search space.

Regarding the objective, text-based objectives benefit from symbolic verifiability and effective reward models. However, for multimodal reasoning, recent benchmarks (Yasunaga, Zettlemoyer, and Ghazvininejad 2025; Hu et al. 2025c) reveal that even the strongest vision-language model judges achieve only 66–75% accuracy on multimodal reward evaluation, highlighting the immaturity of multimodal test-time compute objectives. Despite these difficulties, several studies have begun to build multimodal verifiers and reward models. Multimodal critics or verifiers (Xiong et al. 2024; Tu et al. 2025) provide the foundation for multimodal objectives. VisualPRM (Wang et al. 2025b) constructs process annotation data through Monte Carlo estimation and trains a lightweight PRM verifier, extending step-level objectives from text to vision. Guo et al. (2025b) propose a Potential Assessment Reward Model (PARM) to evaluate the quality potential of intermediate generated images. For video generation, visual quality assessment, audio fidelity, and video coherence require perceptual evaluation that is inherently subjective and harder to automate. Cong et al. (2025) design frame-level PRM objectives, while Liu et al. (2025b) propose stage-specific criteria that evaluate videos at early, middle, and late stages with different objectives.

Regarding the optimization strategy, many recent efforts have sought to bring established test-time compute techniques to multimodal settings. For multimodal reasoning, repeated sampling (Lin et al. 2025; Xu et al. 2024b), self-correction (Ma et al. 2025b; Chen et al. 2025a; Shen et al. 2025a; Deng et al. 2025; Huang et al. 2025c; Peng et al. 2025b), and tree search (Yao et al. 2024; Dong et al. 2024a) have been adapted to improve LMMs’ performance. For multimodal generation, Cong et al. (2025) and Liu et al. (2025b) apply frame-level beam search and frame tree search to video generation, while Guo et al. (2025b) introduce a reflection mechanism for image generation. Despite this progress, a fundamental concern remains: Current multimodal test-time compute may not effectively utilize visual information during the reasoning process. Liu et al. (2025e) systematically investigate the role of visual modality in multimodal mathematical reasoning and find that many models rely predominantly on textual cues while underutilizing visual inputs. This suggests that future work should explore how test-time compute genuinely engages with multimodal content rather than bypassing it through text-dominant reasoning shortcuts (Tong et al. 2025).

## 5.4 Efficient Test-time Compute

While test-time compute scaling has been shown to effectively improve reasoning performance, it introduces substantial computational overhead. As demonstrated in tables 2 and 5, different test-time strategies exhibit markedly different computational characteristics. Parallelizable methods such as repeated sampling typically incur lower latency, whereas sequential approaches, including tree search, require considerably more time and memory. Furthermore, studies on overthinking and underthinking phenomena (Chen et al. 2025f; Wang et al. 2025d) reveal that LRMs often overcomplicate simple problems or frequently switch reasoning paths on difficult ones, both of which waste compute without improving accuracy (Zhang et al. 2025c). These observations

motivate a growing body of work on efficient test-time compute, which seeks to achieve optimal performance under a fixed reasoning budget.

A natural starting point is adaptive compute allocation, which aims to match the amount of test-time compute to the difficulty of each problem. Snell et al. (2024) show that dynamically adjusting compute based on problem difficulty can achieve  $4\times$  efficiency gains over static Best-of-N. Damani et al. (2024) train a lightweight module to predict question difficulty and allocate resources accordingly, and Zhang et al. (2024f) further extend the allocation targets to more hyperparameters. While these methods improve efficiency by better distributing computation, an orthogonal line of work focuses on reducing the amount of reasoning required per instance. Early approaches primarily rely on SFT with variable-length CoT data (Yu et al. 2024b; Liu et al. 2024c; Kang et al. 2025; Xia et al. 2025; Ma et al. 2025a; Munkhbat et al. 2025; Cui et al. 2025; Yang et al. 2025b), whereas more recent methods incorporate length-based rewards into RL (Team et al. 2025; Luo et al. 2025c; Arora and Zanette 2025; Yeo et al. 2025; Yi, Wang, and Li 2025; Yu et al. 2026; Sang et al. 2026) to encourage concise responses. Xiao et al. (2025) design a complexity-aware curriculum learning and GRPO algorithm that dynamically adjusts the reward function based on task complexity. L1 (Aggarwal and Welleck 2025) and Elastic Reasoning (Xu et al. 2025d) enable precise control over response length based on a given token budget, while SelfBudgeter (Li et al. 2025e) can autonomously estimate the optimal token budget before generation. However, both SFT and RL approaches require substantial training resources, motivating training-free alternatives including prompting (Han et al. 2024; Xu et al. 2025a; Aytes, Baek, and Hwang 2025; Lee, Che, and Peng 2025), model merging (Wu et al. 2025; Team et al. 2025), and early exit (Fu et al. 2025; Yang et al. 2026; Xiang et al. 2026). Notably, DEER (Yang et al. 2026) makes early-exit decisions during CoT generation based on the confidence of intermediate answers, and Dai, Yang, and Si (2025) further introduce a new RL paradigm that encourages early thinking termination when appropriate.

Rather than simply shortening reasoning, a more flexible approach is hybrid reasoning, which dynamically switches between concise responses and long-chain reasoning based on task complexity. These methods (Lou et al. 2025; Luo et al. 2025b; Jiang et al. 2025; Fang, Ma, and Wang 2025; Zhang et al. 2025a) follow a similar pipeline: They start with SFT on a mixed dataset of long and short CoT samples as a cold start, and then refine the policy optimization algorithm to train the model to acquire hybrid reasoning capabilities. An even more radical direction is latent reasoning, which performs reasoning in compressed latent spaces rather than verbose token sequences, as explored by Coconut (Hao et al. 2024), CCoT (Cheng and Durme 2024), and LightThinker (Zhang et al. 2025b). There are still many open questions worth exploring, such as how to integrate inference acceleration strategies, e.g., model compression (Li et al. 2024f; Huang et al. 2024c; Li et al. 2025f), token pruning (Fu et al. 2024; Zhang et al. 2024d), and speculative decoding (Leviathan, Kalman, and Matias 2023; Xia et al. 2024) with test-time compute, and how to allocate optimal reasoning budget according to problem difficulty (Wang et al. 2024d; Han et al. 2024; Cheng and Durme 2024).

## 6. Future Directions

### 6.1 Test-time Scaling Law

Unlike training-time computation scaling, test-time compute still lacks a universal scaling law. Some studies have attempted to derive scaling laws for specific test-time compute strategies (Wu et al. 2024c; Levi 2024). Brown et al. (2024) demonstrate that

the performance has an approximately log-linear relationship with repeated sampling times. Chen et al. (2024d) model repeated sampling as a knockout tournament and league-style algorithm, proving theoretically that the failure probability of repeated sampling follows a power-law scaling. Snell et al. (2024) investigate the scaling laws of repeated sampling and self-correction, and propose the computing-optimal scaling strategy. There are two major challenges to achieving a universal scaling law: First, current test-time compute strategies are various, each with different mechanisms to steer the model; thus, it lacks a universal framework for describing them. Second, the performance of test-time compute is affected by a variety of factors, including the difficulty of samples, the accuracy of feedback signals, and decoding hyperparameters, and we need empirical studies to filter out the critical factors.

## 6.2 Strategy Combination

Different test-time compute strategies are suited to various tasks and scenarios, so combining multiple strategies is one way to achieve better System-2 thinking. For example, Marco-o1 (Zhao et al. 2024b) combines the MCTS and self-correction, using MCTS to plan reasoning processes, and self-correction to improve the accuracy of each step. TPO (Li et al. 2025d) combines BoN sampling and self-correction. Moreover, test-time adaptation strategies in System-1 models can also be combined with test-time reasoning strategies. Akyürek et al. (2024) combine test-time training with repeated sampling. They further optimize the language modeling loss on test samples, then generate multiple candidate answers through data augmentation, and finally determine the answer by majority voting. They demonstrate the potential of test-time training in reasoning tasks, surpassing the human average on the ARC challenge. Therefore, we think that for LLM reasoning, it is crucial to focus not only on emerging test-time strategies but also on test-time adaptation methods. For instance, representation editing techniques could be repurposed as reasoning steering vectors to control deliberation depth (Venhoff et al. 2025); and entropy-based adaptation criteria from TTA could provide training-free heuristics for adaptive compute allocation, triggering early stopping for confident predictions or additional compute for uncertain ones. These cross-paradigm transfers suggest that TTA and test-time reasoning offer complementary technical insights, and systematically exploring their integration is a promising direction for building more capable and efficient System-2 models.

## 6.3 New Test-time Compute Paradigms

Latent CoT reasoning has emerged as a promising paradigm for test-time compute. By performing internal reasoning in latent space, it overcomes the limitations of conventional explicit CoT in terms of computational efficiency and abstract reasoning capability (Goyal et al. 2024; Xu et al. 2025b; Shen et al. 2025b; Xu et al. 2025c). Coconut (Hao et al. 2024) performs reasoning in a continuous latent space by directly feeding the model’s final hidden state as the next-step input embedding, thereby giving rise to advanced reasoning patterns. CCoT (Cheng and Durme 2024) introduces a framework that generates continuous and variable-length contemplation tokens as compressed representations of reasoning chains, enabling efficient and accurate reasoning. Light-Thinker (Zhang et al. 2025b) improves LLM efficiency by compressing intermediate reasoning steps into compact gist tokens via specialized data construction and attention mask design, enabling the model to discard verbose chains while maintaining the ability to reason over condensed representations. Latent CoT currently faces several

challenges, including the difficulty of direct supervision, limited generalization, and concerns over interpretability (Chen et al. 2025e). This calls for further exploration of token-level strategies and internal mechanisms to enable more robust and transparent latent reasoning.

## 7. Benchmarks

*Test-time Adaptation.* In System-1 models, distribution shifts include adversarial robustness and cross-domain and cross-lingual scenarios. In the field of CV, ImageNet-C (Hendrycks and Dietterich 2019), ImageNet-R (Hendrycks et al. 2021a), and ImageNet-Sketch (Wang et al. 2019) are common datasets for TTA. Yu et al. (2023) propose a benchmark to conduct a unified evaluation of TTA methods across different TTA settings and backbones on 5 image classification datasets. For NLP tasks, TTA is primarily applied in QA and machine translation tasks, with commonly used datasets such as MLQA (Lewis et al. 2020), XQuAD (Artetxe, Ruder, and Yogatama 2020), MRQA (Fisch et al. 2019), and CCMatrix (Schwenk et al. 2021). These benchmarks remain relevant for evaluating classical TTA methods. However, for frontier LLMs, the above benchmarks are outdated. What is urgently needed are benchmarks that evaluate LLM robustness and generalization in more realistic and complex reasoning scenarios, such as robustness to adversarially perturbed math problems (Huang et al. 2025b; Sun, Dai, and Ye 2026).

*Feedback Modeling.* The quality of verifiers and reward models is a critical bottleneck for test-time compute (Snell et al. 2024). Benchmarking these feedback models is therefore essential for understanding the practical ceiling of test-time reasoning strategies. RewardBench (Lambert et al. 2024) collects 20.2k prompt-choice-rejection triplets covering tasks such as dialogue, reasoning, and safety. It evaluates the accuracy of reward models in distinguishing between chosen and rejected responses. RM-Bench (Liu et al. 2024e) further evaluates the impact of response style on reward models. RMB (Zhou et al. 2024a) extends the evaluation to the more practical BoN setting, where reward models are required to select the best response from multiple candidates. CriticBench (Lin et al. 2024c) is specifically designed to evaluate a critic model's generation, critique, and correction capabilities. For PRM, Song et al. (2025a) propose PRMBench, which evaluates whether PRMs can identify the earliest incorrect reasoning step in math tasks. ProcessBench (Zheng et al. 2024) provides a more fine-grained evaluation, including redundancy, soundness, and sensitivity. In addition, there are benchmarks for evaluating multimodal feedback modeling, such as VL-RewardBench (Li et al. 2024e), VCR-Bench (Qi et al. 2025), and MJ-Bench (Chen et al. 2024e).

*Test-time Reasoning.* Reasoning capability is the core of System-2 models, including mathematics, code, commonsense, planning, etc. (Zeng et al. 2024). *Math reasoning* is one of the most compelling reasoning tasks. With the advancements in LLM and test-time compute, the accuracy on some previously challenging benchmarks, like GSM8K (Cobbe et al. 2021) and MATH (Hendrycks et al. 2021b), have surpassed the 90% mark. Thus, more difficult college admissions exam (Zhang et al. 2023b; Arora, Singh, and Mausam 2023; Azerbayev et al. 2024) and competition-level (Gao et al. 2024a) math benchmarks have been proposed. Some competition-level benchmarks are not limited to textual modalities in algebra, logic reasoning, and word problems. For instance, OlympiadBench (He et al. 2024), OlympicArena (Huang et al. 2024b), and AIME (Zamil and Rabby 2024) provide images for geometry problems, incorporating visual

information to aid in problem-solving, while AlphaGeometry (Trinh et al. 2024) uses symbolic rules for geometric proofs. The most challenging benchmark currently is FrontierMath (Glazer et al. 2024), with problems crafted by mathematicians and covering major branches of modern mathematics. Even the most advanced o3 has not achieved 30% accuracy. The difficulty spectrum of math benchmarks has important implications for test-time compute strategy selection. On easier benchmarks like GSM8K, the cutting-edge base model already achieves near-ceiling performance, making additional compute yield diminishing returns. In contrast, competition-level benchmarks such as AIME and FrontierMath exhibit strong test-time scaling behavior. This suggests that future benchmarks should include problems spanning a wide difficulty range, so that the benefits of different test-time compute strategies can be clearly distinguished.

*Code ability* is a key aspect of LLM reasoning, with high practical value, covering code completion (Ding et al. 2023; Zhang et al. 2023a; Gong et al. 2024a), code reasoning (Gu et al. 2024; Wei et al. 2025), and code generation (Chen et al. 2021; Austin et al. 2021) tasks. Among these, code generation gains more attention. HumanEval (Chen et al. 2021) and MBPP (Austin et al. 2021) provide natural language descriptions of programming problems, requiring LLMs to generate corresponding Python code and use unit tests for evaluation. MultiPL-E (Cassano et al. 2022) extend them to 18 program languages. EvalPlus (Liu et al. 2024a) automatically augments test cases to assess the robustness of the generated code. Recently, some studies collect benchmarks from open-source projects, which are closed to realistic applications and more challenging due to complex function calls, such as DS-1000 (Lai et al. 2023), CoderEval (Yu et al. 2024a), EvoCodeBench (Li et al. 2024a), LiveCodeBench (Jain et al. 2025), and BigCodeBench (Zhuo et al. 2025). Beyond single-turn code generation, SWE-bench (Jimenez et al. 2024; Zan et al. 2025) evaluates whether models can resolve real GitHub issues within full code repositories. Solving such tasks typically requires multi-turn interaction with the codebase by iteratively navigating files, editing code, running tests, and refining their patches based on execution feedback.

*Commonsense reasoning* requires LLMs to possess both commonsense and reasoning abilities. StrategyQA (Geva et al. 2021) collects complex and subtle multi-hop reasoning questions. MMLU (Hendrycks et al. 2021b) and MMLU-Pro (Wang et al. 2024l) cover commonsense reasoning questions across various domains, including STEM, the humanities, the social sciences, etc. GPQA (Rein et al. 2024) targets graduate-level questions that are difficult even for domain experts, and SuperGPQA (Du et al. 2025) further scales this evaluation to 285 graduate disciplines, providing a comprehensive testbed for evaluating whether test-time compute can bridge the gap between model capability and expert-level reasoning across diverse domains. *Planning* aims to enable LLMs to take optimal actions based on the current state and environment to complete tasks. Current planning benchmarks primarily focus on synthetic tasks, such as Blocksworld (Valmeekam et al. 2023), Crosswords, and Game-of-24 (Yao et al. 2023). For commonsense and planning tasks, the lack of symbolic verifiability makes it difficult to apply BoN sampling or tree search with automated verifiers. These benchmarks are better suited for evaluating self-correction and multi-agent debate strategies, where natural-language critique can provide meaningful feedback.

Despite the breadth of existing benchmarks, several gaps remain for test-time compute evaluation. Most critically, no benchmark explicitly measures the *scaling behavior* of test-time compute, i.e., how performance improves as a function of additional inference-time computation. Such benchmarks would enable principled comparisons across strategies and inform practical deployment decisions. Additionally, most benchmarks use static problem sets that are vulnerable to data contamination; dynamic

benchmarks like LiveCodeBench (Jain et al. 2025) and NPHardEval (Fan et al. 2024) that continuously refresh problems offer promising solutions. Finally, current benchmarks evaluate only task accuracy without penalizing excessive computation, and *compute-aware* metrics that jointly consider accuracy and cost would better reflect real-world deployment trade-offs.

## 8. Conclusion

In this article, we conduct a comprehensive survey of existing works on test-time compute. We introduce various test-time compute methods in System-1 and System-2 models, and look forward to future directions for this field. We believe that test-time compute can help models handle complex real-world distributions and tasks better, making it a promising path for advancing LLMs toward cognitive intelligence. We hope this paper will promote further research.

## References

- Aggarwal, Pranjal, Aman Madaan, Yiming Yang, and Mausam. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396. <https://doi.org/10.18653/v1/2023.emnlp-main.761>
- Aggarwal, Pranjal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.
- Ahmadian, Arash, 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267. <https://doi.org/10.18653/v1/2024.acl-long.662>
- Akyürek, Ekin, Mehul Damani, Linlu Qiu, Han Guo, Yoon Kim, and Jacob Andreas. 2024. The surprising effectiveness of test-time training for abstract reasoning. *arXiv preprint arXiv:2411.07279*.
- Amayuelas, Alfonso, Xianjun Yang, Antonis Antoniadis, Wenyue Hua, Liangming Pan, and William Wang. 2024. Multiagent collaboration attack: Investigating adversarial attacks in large language model collaborations via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6929–6948. <https://doi.org/10.18653/v1/2024.findings-emnlp.407>
- Amini, Afra, Tim Vieira, and Ryan Cotterell. 2024. Variational Best-of-N alignment. *arXiv preprint arXiv:2407.06057*.
- An, Shengnan, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes LLM better reasoner. *arXiv preprint arXiv:2310.20689*.
- Ankner, Zachary, Mansheej Paul, Brandon Cui, Jonathan D. Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*.
- Arditi, Andy, Oscar Obeso, Aaqib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*. <https://doi.org/10.52202/079017-4322>
- Arora, Daman, Himanshu Singh, and Mausam. 2023. Have LLMs advanced enough? A challenging problem solving benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7527–7543. <https://doi.org/10.18653/v1/2023.emnlp-main.468>
- Arora, Daman and Andrea Zanette. 2025. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*. <https://doi.org/10.52202/085713-2032>
- Artetxe, Mikel, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637. <https://doi.org/10.18653/v1/2020.acl-main.421>

- Austin, Jacob, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.
- Aytes, Simon A., Jinheon Baek, and Sung Ju Hwang. 2025. Sketch-of-thought: Efficient LLM reasoning with adaptive cognitive-inspired sketching. *arXiv preprint arXiv:2503.05179*. <https://doi.org/10.18653/v1/2025.emnlp-main.1236>
- Azerbaiyev, Zhanger, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations*.
- Bai, Yuntao, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.
- Bar, Yarin, Shalev Shaer, and Yaniv Romano. 2024. Protected test-time adaptation via online entropy matching: A betting approach. *arXiv preprint arXiv:2408.07511*. <https://doi.org/10.52202/079017-2714>
- Bartler, Alexander, Andre Böhler, Felix Wiewel, Mario Döbler, and Bin Yang. 2022. MT3: Meta test-time training for self-supervised test-time adaption. In *International Conference on Artificial Intelligence and Statistics*, pages 3080–3090.
- Bavaresco, Anna, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, et al. 2024. LLMs instead of human judges? A large scale empirical study across 20 NLP evaluation tasks. *arXiv preprint arXiv:2406.18403*. <https://doi.org/10.18653/v1/2025.acl-short.20>
- Bhardwaj, Rishabh, Yingting Li, Navonil Majumder, Bo Cheng, and Soujanya Poria. 2023. kNN-CM: A non-parametric inference-phase adaptation of parametric text classifiers. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13546–13557. <https://doi.org/10.18653/v1/2023.findings-emnlp.903>
- Bi, Zhenni, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2024. Forest-of-thought: Scaling test-time compute for enhancing LLM reasoning. *arXiv preprint arXiv:2412.09078*.
- Bradley, Ralph Allan and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345. <https://doi.org/10.1093/biomet/39.3-4.324>
- Brown, Bradley, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.
- Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.
- Burns, Collin, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. 2023a. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.
- Burns, Collin, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023b. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations*.
- Cao, Yuanpu, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. 2024. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *arXiv preprint arXiv:2406.00045*. <https://doi.org/10.52202/079017-1567>
- Cassano, Federico, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q. Feldman, et al. 2022. MultiPL-E: A scalable and extensible approach to benchmarking neural code generation. *arXiv preprint arXiv:2208.08227*. <https://doi.org/10.1109/TSE.2023.3267446>
- Chan, Chi Min, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*.

- Chari, Anirudh, Aditya Tiwari, Richard Lian, Suraj Reddy, and Brian Zhou. 2025. Pheromone-based learning of optimal reasoning paths. *arXiv preprint arXiv:2501.19278*.
- Chen, Dian, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. 2022. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305. <https://doi.org/10.1109/CVPR52688.2022.00039>
- Chen, Guoxin, Minpeng Liao, Chengxi Li, and Kai Fan. 2024a. Alphamath almost zero: Process supervision without process. <https://neurips.cc/virtual/2024/poster/94884>. <https://doi.org/10.52202/079017-0870>
- Chen, Hardy, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025a. SFT or RL? An early investigation into training R1-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*.
- Chen, Jiefeng, Jie Ren, Xinyun Chen, Chengrun Yang, Ruoxi Sun, and Sercan Ö. Arik. 2025b. Sets: Leveraging self-verification and self-correction for improved test-time scaling. *arXiv preprint arXiv:2501.19306*.
- Chen, Justin, Swarnadeep Saha, and Mohit Bansal. 2024. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085. <https://doi.org/10.18653/v1/2024.ac1-long.381>
- Chen, Lichang, Chen Zhu, Jiuhai Chen, Davit Soselia, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Sholeybi, and Bryan Catanzaro. 2024b. ODIN: Disentangled reward mitigates hacking in RLHF. In *Forty-first International Conference on Machine Learning*.
- Chen, Mark, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Chen, Nuo, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. 2025c. JudgeLRM: Large reasoning models as a judge. *arXiv preprint arXiv:2504.00050*.
- Chen, Qiguang, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025d. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*. <https://doi.org/10.1007/s11432-025-4665-8>
- Chen, Wei Lin, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023. Self-ICL: Zero-shot in-context learning with self-generated demonstrations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15651–15662. <https://doi.org/10.18653/v1/2023.emnlp-main.968>
- Chen, Xinghao, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. 2025e. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. *arXiv preprint arXiv:2505.16782*.
- Chen, Xingyu, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025f. Do NOT think that much for 2+3=? on the overthinking of o1-like LLMs. *arXiv preprint arXiv:2412.21187*.
- Chen, Xinyun, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024c. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*.
- Chen, Xiusi, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. 2025g. RM-R1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*.
- Chen, Yanxi, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024d. A simple and provable scaling law for the test-time compute of large language models. *arXiv preprint arXiv:2411.19477*. <https://doi.org/10.52202/085713-5570>
- Chen, Zhaorun, Yichao Du, Zichen Wen, Yiyang Zhou, Chenhang Cui, Zhenzhen Weng, Haoqin Tu, Chaoqi Wang, Zhengwei Tong, Qinglan Huang, Canyu Chen, Qinghao Ye, Zhihong Zhu, Yuqing Zhang, Jiawei Zhou, Zhuokai Zhao, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. 2024e. MJ-Bench: Is your multimodal reward model really a good judge for text-to-image generation? *arXiv preprint*

- arXiv:2407.04842*. <https://doi.org/10.52202/085713-2081>
- Cheng, Jeffrey and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*.
- Cheng, Xiaoxue, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Think more, hallucinate less: Mitigating hallucinations via dual process of fast and slow thinking. *arXiv preprint arXiv:2501.01306*. <https://doi.org/10.18653/v1/2025.findings-acl.417>
- Chiang, Cheng Han and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631. <https://doi.org/10.18653/v1/2023.acl-long.870>
- Chiang, Cheng Han, Hung-yi Lee, and Michal Lukasik. 2025. TRACT: Regression-aware fine-tuning meets chain-of-thought reasoning for LLM-as-a-Judge. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2934–2952. <https://doi.org/10.18653/v1/2025.acl-long.147>
- Chow, Yinlam, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Sridhar Thiagarajan, Craig Boutilier, Rishabh Agarwal, Aviral Kumar, and Aleksandra Faust. 2024. Inference-aware fine-tuning for Best-of-N sampling in large language models. *arXiv preprint arXiv:2412.15287*.
- Cobbe, Karl, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Cong, Wenyan, Hanqing Zhu, Peihao Wang, Bangya Liu, Dejie Xu, Kevin Wang, David Z. Pan, Yan Wang, Zhiwen Fan, and Zhangyang Wang. 2025. Can test-time scaling improve world foundation model? *arXiv preprint arXiv:2503.24320*.
- Cui, Yingqian, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, Suhang Wang, Yue Xing, Jiliang Tang, and Qi He. 2025. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18581–18597. <https://doi.org/10.18653/v1/2025.findings-acl.956>
- Dai, Damai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? Language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019. <https://doi.org/10.18653/v1/2023.findings-acl.247>
- Dai, Mz, Chenxu Yang, and Qingyi Si. 2025. S-GRPO: Early exit via reinforcement learning in reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/085713-1608>
- Damani, Mehul, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. 2024. Learning how hard to think: Input-adaptive allocation of LM computation. *arXiv preprint arXiv:2410.04707*.
- Dathathri, Sumanth, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.
- DeepMind, Google. 2025. Gemini 2.5: Our most intelligent AI model.
- Deng, Yihe, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. 2025. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *arXiv preprint arXiv:2503.17352*. <https://doi.org/10.52202/085713-4126>
- Ding, Yangruibo, Zijian Wang, Wasi Uddin Ahmad, Hantian Ding, Ming Tan, Nihal Jain, Murali Krishna Ramanathan, Ramesh Nallapati, Parminder Bhatia, Dan Roth, and Bing Xiang. 2023. CrossCodeEval: A diverse and multilingual benchmark for cross-file code completion. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. <https://doi.org/10.52202/075280-2023>
- Dong, Guanting, Chenghao Zhang, Mengjie Deng, Yutao Zhu, Zhicheng Dou, and Ji-Rong Wen. 2024a. Progressive multimodal reasoning via active retrieval. *arXiv preprint arXiv:2412.14835*.
- Dong, Qingxiu, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun,

- Lei Li, and Zhifang Sui. 2024b. A survey on in-context learning. In *EMNLP 2024*, pages 1107–1128. <https://doi.org/10.18653/v1/2024.emnlp-main.64>
- Du, Mengnan, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023a. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67(1):110–120. <https://doi.org/10.1145/3596490>
- Du, Xeron, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, et al. 2025. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. <https://doi.org/10.52202/085713-3766>
- Du, Yilun, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023b. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.
- Elgohary, Ahmed, Christopher Meek, Matthew Richardson, Adam Fourney, Gonzalo Ramos, and Ahmed Hassan Awadallah. 2021. NL-EDIT: Correcting semantic parse errors through natural language interaction. In *NAACL 2021*, pages 5599–5610. <https://doi.org/10.18653/v1/2021.naacl-main.444>
- Fan, Lizhou, Wenyue Hua, Lingyao Li, Haoyang Ling, and Yongfeng Zhang. 2024. NPHardEval: Dynamic benchmark on reasoning ability of large language models via complexity classes. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4092–4114. <https://doi.org/10.18653/v1/2024.acl-long.225>
- Fang, Gongfan, Xinyin Ma, and Xinchao Wang. 2025. Thinkless: LLM learns when to think. *arXiv preprint arXiv:2505.13379*. <https://doi.org/10.52202/085713-5060>
- Feng, Chun Mei, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. 2023. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2704–2714. <https://doi.org/10.1109/ICCV51070.2023.00255>
- Feng, Guhao, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. 2024a. Towards revealing the mystery behind chain of thought: A theoretical perspective. *Advances in Neural Information Processing Systems*, 36. <https://doi.org/10.52202/075280-3100>
- Feng, Wenfeng, Chuzhan Hao, Yuewei Zhang, Jingyi Song, and Hao Wang. 2025. AirRAG: Activating intrinsic reasoning for retrieval augmented generation using tree-based search. *arXiv preprint arXiv:2501.10053*.
- Feng, Xidong, Ziyu Wan, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2024b. Alphazero-like tree-search can guide large language model decoding and training. <https://neurips.cc/virtual/2023/82861>
- Fisch, Adam, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13. <https://doi.org/10.18653/v1/D19-5801>
- Fu, Qichen, Minsik Cho, Thomas Merth, Sachin Mehta, Mohammad Rastegari, and Mahyar Najibi. 2024. LazyLLM: Dynamic token pruning for efficient long context LLM inference. *arXiv preprint arXiv:2407.14057*.
- Fu, Yichao, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. 2025. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*.
- Gan, Bingzheng, Yufan Zhao, Tianyi Zhang, Jing Huang, Yusu Li, Shu Xian Teo, Changwang Zhang, and Wei Shi. 2025. MASTER: A multi-agent system with LLM specialized MCTS. *arXiv preprint arXiv:2501.14304*. <https://doi.org/10.18653/v1/2025.naacl-long.476>
- Gandelsman, Yossi, Yu Sun, Xinlei Chen, and Alexei A. Efros. 2022. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*. <https://doi.org/10.52202/068431-2130>
- Gao, Bofei, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. 2024a. Omni-MATH: A universal Olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.
- Gao, Ge, Eunsol Choi, and Yoav Artzi. 2022. Simulating bandit learning from user feedback for extractive question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational*

- Linguistics (Volume 1: Long Papers)*, pages 5167–5179. <https://doi.org/10.18653/v1/2022.acl-long.355>
- Gao, Jun, Yongqi Li, Ziqiang Cao, and Wenjie Li. 2024b. Interleaved-modal chain-of-thought. *arXiv preprint arXiv:2411.19488*. <https://doi.org/10.1109/CVPR52734.2025.01818>
- Gao, Kuofeng, Huanqia Cai, Qingyao Shuai, Dihong Gong, and Zhifeng Li. 2024c. Embedding self-correction as an inherent ability in large language models for enhanced mathematical reasoning. *arXiv preprint arXiv:2410.10735*.
- Gao, Leo, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866.
- Gao, Luyu, Zhu Yun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508. <https://doi.org/10.18653/v1/2023.acl-long.910>
- Gao, Zitian, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. 2024d. Interpretable contrastive Monte Carlo Tree Search reasoning. *arXiv preprint arXiv:2410.01707*.
- Geirhos, Robert, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673. <https://doi.org/10.1038/s42256-020-00257-z>
- Geva, Mor, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361. [https://doi.org/10.1162/tac1\\_a.00370](https://doi.org/10.1162/tac1_a.00370)
- Glazer, Elliot, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, et al. 2024. FrontierMath: A benchmark for evaluating advanced mathematical reasoning in AI. *arXiv preprint arXiv:2411.04872*.
- Gong, Linyuan, Sida Wang, Mostafa Elhoushi, and Alvin Cheung. 2024a. Evaluation of LLMs on syntax-aware code fill-in-the-middle tasks. In *Forty-first International Conference on Machine Learning*.
- Gong, Taesik, Yewon Kim, Taeckyeung Lee, Sorn Chottananurak, and Sung-Ju Lee. 2024b. SoTTA: Robust test-time adaptation on noisy data streams. *Advances in Neural Information Processing Systems*, 36. <https://doi.org/10.52202/075280-0620>
- Google. 2024. Try Deep Research and our new experimental model in Gemini, your AI assistant. Google, blog.
- Gou, Zhibin, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.
- Goyal, Sachin, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations*.
- Gu, Alex, Baptiste Roziere, Hugh James Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida Wang. 2024. CRUXEval: A benchmark for code reasoning, understanding and execution. In *Forty-first International Conference on Machine Learning*.
- Guan, Xinyan, Yanjiang Liu, Hongyu Lin, Yaojie Lu, Ben He, Xianpei Han, and Le Sun. 2024. Mitigating large language model hallucinations via autonomous knowledge graph-based retrofitting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18126–18134. <https://doi.org/10.1609/aaai.v38i16.29770>
- Guan, Xinyu, Li Lina Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rStar-Math: Small LLMs can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.
- Gui, Lin, Cristina Garbacea, and Victor Veitch. 2024. BoNBon alignment for large language models and the sweetness of Best-of-N sampling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/079017-0094>
- Gulcehre, Caglar, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte

- Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.
- Guo, Daya, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025a. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Guo, Ziyu, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. 2025b. Can we generate images with CoT? Let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*.
- Hagendorff, Thilo, Sarah Fabi, and Michal Kosinski. 2023. Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in chatgpt. *Nature Computational Science*, 3(10):833–838. <https://doi.org/10.1038/s43588-023-00527-x>, PubMed: 38177754
- Han, Tingxu, Chunrong Fang, Shiyu Zhao, Shiqing Ma, Zhenyu Chen, and Zhenting Wang. 2024. Token-budget-aware LLM reasoning. *arXiv preprint arXiv:2412.18547*. <https://doi.org/10.18653/v1/2025.findings-acl.1274>
- Hao, Shibo, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173. <https://doi.org/10.18653/v1/2023.emnlp-main.507>
- Hao, Shibo, Sainbayar Sukhbaatar, Dilja 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*.
- Hardt, Moritz and Yu Sun. 2024. Test-time training on nearest neighbors for large language models. In *The Twelfth International Conference on Learning Representations*.
- Hassan, Jameel, Hanan Gani, Noor Hussein, Muhammad Uzair Khattak, Muzammal Naseer, Fahad Shahbaz Khan, and Salman Khan. 2023. Align your prompts: Test-time prompting with distribution alignment for zero-shot generalization. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 80396–80413. <https://doi.org/10.52202/075280-3525>
- Havrilla, Alex, Sharath Rapparthi, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. 2024. GLoRe: When, where, and how to improve LLM reasoning via global and local refinements. *arXiv preprint arXiv:2402.10963*.
- He, Chaoqun, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850. <https://doi.org/10.18653/v1/2024.acl-long.211>
- Hendrycks, Dan, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. 2021a. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349. <https://doi.org/10.1109/ICCV48922.2021.00823>
- Hendrycks, Dan, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Hendrycks, Dan and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.
- Hosseini, Arian, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordani, and Rishabh Agarwal. 2024. V-STaR: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*.
- Hou, Zhenyu, 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*.
- Hou, Zhenyu, Xin Lv, Rui Lu, Jiajie Zhang, Yuijiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. 2025. Advancing language model reasoning through reinforcement learning and inference scaling. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 23976–24003.
- Houlsby, Neil, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*, pages 2790–2799.
- Hu, Edward J, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Hu, Jian, Jason Klein Liu, Haotian Xu, and Wei Shen. 2025a. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*.
- Hu, Jinwu, Zhitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Minghui Tan. 2025b. Test-time learning for large language models. *arXiv preprint arXiv:2505.20633*.
- Hu, Minda, Licheng Zong, Hongru Wang, Jingyan Zhou, Jingjing Li, Yichen Gao, Kam-Fai Wong, Yu Li, and Irwin King. 2024a. SeRTS: Self-rewarding tree search for biomedical retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1321–1335. <https://doi.org/10.18653/v1/2024.findings-emnlp.71>
- Hu, Yushi, Reyhane Askari-Hemmat, Melissa Hall, Emily Dinan, Luke Zettlemoyer, and Marjan Ghazvininejad. 2025c. Multimodal RewardBench 2: Evaluating omni reward models for interleaved text and image. *arXiv preprint arXiv:2512.16899*.
- Hu, Yushi, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. 2024b. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/079017-4423>
- Huang, Chengsong, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. 2025a. Efficient test-time scaling via self-calibration. *arXiv preprint arXiv:2503.00031*.
- Huang, Jie, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024a. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.
- Huang, Kaixuan, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. 2025b. MATH-perturb: Benchmarking LLMs’ math reasoning abilities against hard perturbations. In *Forty-second International Conference on Machine Learning*.
- Huang, Wenxuan, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025c. Vision-R1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.
- Huang, Zhen, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, et al. 2024b. OlympicArena: Benchmarking multi-discipline cognitive reasoning for superintelligent AI. *arXiv preprint arXiv:2406.12753*. <https://doi.org/10.52202/079017-0607>
- Huang, Zhen, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. 2024c. O1 replication journey-Part 2: Surpassing O1-preview through simple distillation, big progress or bitter lesson? *arXiv preprint arXiv:2411.16489*.
- Hübotter, Jonas, Sascha Bongni, Ido Hakimi, and Andreas Krause. 2025. Efficiently learning at test-time: Active fine-tuning of LLMs. In *The Thirteenth International Conference on Learning Representations*.
- Imam, Raza, Hanan Gani, Muhammad Huzaifa, and Karthik Nandakumar. 2024. Test-time low rank adaptation via confidence maximization for zero-shot generalization of vision-language models. *arXiv preprint arXiv:2407.15913*. <https://doi.org/10.1109/WACV61041.2025.00532>
- Jain, Naman, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. LiveCodeBench: Holistic and contamination free evaluation of large

- language models for code. In *The Thirteenth International Conference on Learning Representations*.
- Jia, Chen. 2024. Generalizing reward modeling for out-of-distribution preference learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 107–124. [https://doi.org/10.1007/978-3-031-70362-1\\_7](https://doi.org/10.1007/978-3-031-70362-1_7)
- Jiang, Jinhao, Jiayi Chen, Junyi Li, Ruiyang Ren, Shijie Wang, Wayne Xin Zhao, Yang Song, and Tao Zhang. 2024. RAG-Star: Enhancing deliberative reasoning with retrieval augmented verification and refinement. *arXiv preprint arXiv:2412.12881*.
- Jiang, Lingjie, Xun Wu, Shaohan Huang, Qingxiu Dong, Zewen Chi, Li Dong, Xingxing Zhang, Tengchao Lv, Lei Cui, and Furu Wei. 2025. Think only when you need with large hybrid-reasoning models. *arXiv preprint arXiv:2505.14631*. <https://doi.org/10.52202/085713-5323>
- Jimenez, Carlos E., John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. 2024. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations*.
- Jin, Bowen, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Serkan 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*.
- Kahneman, Daniel. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux.
- Kamoi, Ryo, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can LLMs actually correct their own mistakes? A critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440. <https://doi.org/10.1162/tac1.a.00713>
- Kang, Yu, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3oT: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320. <https://doi.org/10.1609/aaai.v39i23.34608>
- Kang, Zhewei, Xuandong Zhao, and Dawn Song. 2025. Scalable Best-of-N selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*. <https://doi.org/10.52202/085713-0663>
- Khalifa, Muhammad, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *arXiv preprint arXiv:2504.16828*.
- Khan, Akbir, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive LLMs leads to more truthful answers. In *Forty-first International Conference on Machine Learning*.
- Khandelwal, Urvashi, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.
- Khandelwal, Urvashi, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.
- Kim, Geunwoo, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *arXiv preprint arXiv:2303.17491*. <https://doi.org/10.52202/075280-1723>
- Kim, Hyuhng Joon, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2022. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *arXiv preprint arXiv:2206.08082*.
- Kim, Kyungha, Sangyun Lee, Kung-Hsiang Huang, Hou Pong Chan, Manling Li, and Heng Ji. 2024a. Can LLMs produce faithful explanations for fact-checking? Towards faithful explainable fact-checking via multi-agent debate. *arXiv preprint arXiv:2402.07401*.
- Kim, Seungone, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024b. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.
- Kim, Seungone, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024c. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353.

- <https://doi.org/10.18653/v1/2024.emnlp-main.248>
- Kim, Seungone, Ian Wu, Jinu Lee, Xiang Yue, Seongyun Lee, Mingyeong Moon, Kiril Gashteovski, Carolin Lawrence, Julia Hockenmaier, Graham Neubig, and Sean Welleck. 2025. Scaling evaluation-time compute with reasoning models as process evaluators. *arXiv preprint arXiv:2503.19877*. <https://doi.org/10.18653/v1/2026.findings-acl.2102>
- Kim, Sunghwan, Dongjin Kang, Taeyoon Kwon, Hyungjoo Chae, Jungsoo Won, Dongha Lee, and Jinyoung Yeo. 2024d. Evaluating robustness of reward models for mathematical reasoning. *arXiv preprint arXiv:2410.01729*.
- Kojima, Takeshi, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213. <https://doi.org/10.52202/068431-1613>
- Kumar, Aviral, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D. Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. 2024. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.
- Laban, Philippe, Lidiya Murakhovska, Caiming Xiong, and Chien-Sheng Wu. 2024. Are you sure? Challenging LLMs leads to performance drops in the FlipFlop experiment. *arXiv preprint arXiv:2311.08596*.
- Lai, Yuhang, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2023. DS-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pages 18319–18345.
- Lambert, Nathan, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. RewardBench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*. <https://doi.org/10.18653/v1/2025.findings-naacl.96>
- Lan, Tian, Wenwei Zhang, Chengqi Lyu, Shuaibin Li, Chen Xu, Heyan Huang, Dahua Lin, Xian-Ling Mao, and Kai Chen. 2024. Training language models to critique with multi-agent feedback. *arXiv preprint arXiv:2410.15287*. <https://doi.org/10.18653/v1/2025.findings-emnlp.78>
- LeCun, Yann. 2022. A path towards autonomous machine intelligence. <https://openreview.net/pdf?id=BZ5a1r-kVsf>
- Lee, Ayeong, Ethan Che, and Tianyi Peng. 2025. How well do LLMs compress their own chain-of-thought? A token complexity approach. *arXiv preprint arXiv:2503.01141*.
- Lee, Junlin, Yequan Wang, Jing Li, and Min Zhang. 2024. Multimodal reasoning with multimodal knowledge graph. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10767–10782. <https://doi.org/10.18653/v1/2024.acl-long.579>
- Lester, Brian, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059. <https://doi.org/10.18653/v1/2021.emnlp-main.243>
- Levi, Noam. 2024. A simple model of inference scaling laws. *arXiv preprint arXiv:2410.16377*.
- Leviathan, Yaniv, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286.
- LeVine, Will, Benjamin Pikus, Anthony Chen, and Sean Hendryx. 2023. A baseline analysis of reward models’ ability to accurately analyze foundation models under distribution shift. *arXiv preprint arXiv:2311.14743*.
- Lewis, Patrick, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330. <https://doi.org/10.18653/v1/2020.acl-main.653>
- Li, Jia, Ge Li, Xuanming Zhang, Yihong Dong, and Zhi Jin. 2024a. EvoCodeBench: An evolving code generation benchmark aligned with real-world code repositories. *arXiv preprint arXiv:2404.00599*. <https://doi.org/10.52202/079017-1837>
- Li, Jiachun, Pengfei Cao, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2025a. Rewarding curse: Analyze and mitigate reward modeling issues for LLM reasoning. *arXiv preprint arXiv:2503.05188*.

- Li, Junlong, Shichao Sun, Weizhe Yuan, Run-Ze Fan, hai zhao, and Pengfei Liu. 2024b. Generative judge for evaluating alignment. In *The Twelfth International Conference on Learning Representations*.
- Li, Junyi and Hwee Tou Ng. 2024. Think&Cite: Improving attributed text generation with self-guided tree search and progress reward modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9928–9942. <https://doi.org/10.18653/v1/2025.acl-long.490>
- Li, Junyou, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024c. More agents is all you need. *arXiv preprint arXiv:2402.05120*.
- Li, Kenneth, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/075280-1797>
- Li, Loka, Zhenhao Chen, Guangyi Chen, Yixuan Zhang, Yusheng Su, Eric Xing, and Kun Zhang. 2024d. Confidence matters: Revisiting intrinsic self-correction capabilities of large language models. *arXiv preprint arXiv:2402.12563*.
- Li, Lei, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, Lingpeng Kong, and Qi Liu. 2024e. VL-RewardBench: A challenging benchmark for vision-language generative reward models. *arXiv preprint arXiv:2411.17451*. <https://doi.org/10.1109/CVPR52734.2025.02296>
- Li, Xiang, Shizhu He, Jiayu Wu, Zhao Yang, Yao Xu, Yang jun Jun, Haifeng Liu, Kang Liu, and Jun Zhao. 2024f. MoDE-CoTD: Chain-of-Thought Distillation for complex reasoning tasks with mixture of decoupled LoRA-experts. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11475–11485. <https://doi.org/10.63317/2rxtdcyfrcxn>
- Li, Xiaonan, Kai Lv, Hang Yan, Tianxian Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023b. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668. <https://doi.org/10.18653/v1/2023.acl-long.256>
- Li, Xiaonan and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235. <https://doi.org/10.18653/v1/2023.findings-emnlp.411>
- Li, Xiaoxi, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025b. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*. <https://doi.org/10.18653/v1/2025.emnlp-main.276>
- Li, Xiaoxi, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025c. WebThinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*. <https://doi.org/10.52202/085713-4011>
- Li, Yafu, Xuyang Hu, Xiaoye Qu, Linjie Li, and Yu Cheng. 2025d. Test-time preference optimization: On-the-fly alignment via iterative textual feedback. *arXiv preprint arXiv:2501.12895*.
- Li, Yifei, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023c. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333. <https://doi.org/10.18653/v1/2023.acl-long.291>
- Li, Yujia, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. 2022a. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097. <https://doi.org/10.1126/science.abq1158>, PubMed: 36480631
- Li, Yunxuan, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024g. Improving multi-agent debate with sparse communication topology. *arXiv preprint arXiv:2406.11776*. <https://doi.org/10.18653/v1/2024.findings-emnlp.427>
- Li, Zhen, Yupeng Su, Runming Yang, Zhongwei Xie, Ngai Wong, and Hongxia Yang. 2025e. Quantization meets reasoning: Exploring LLM low-bit quantization degradation for mathematical reasoning. *arXiv preprint arXiv:2501.03035*.
- Li, Zheng, Qingxiu Dong, Jingyuan Ma, Di Zhang, and Zhifang Sui. 2025f.

- SelfBudgeter: Adaptive token allocation for efficient LLM reasoning. *arXiv preprint arXiv:2505.11274*. <https://doi.org/10.18653/v1/2026.findings-acl.1063>
- Li, Zhiyuan, Hong Liu, Denny Zhou, and Tengyu Ma. 2024h. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*.
- Li, Zhong-Zhi, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. 2025g. From System 1 to System 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*. <https://doi.org/10.1109/TPAMI.2025.3637037>, PubMed: 41289126
- Li, Zichao, Prakhar Sharma, Xing Han Lu, Jackie Cheung, and Siva Reddy. 2022b. Using interactive feedback to improve the accuracy and explainability of question answering systems post-deployment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 926–937. <https://doi.org/10.18653/v1/2022.findings-acl.75>
- Liang, Tian, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024a. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*. <https://doi.org/10.18653/v1/2024.emnlp-main.992>
- Liang, Xiaobo, Haoke Zhang, Helan Hu, Juntao Li, Jun Xu, and Min Zhang. 2024b. Fennec: Fine-grained language model evaluation and correction extended through branching and bridging. *arXiv preprint arXiv:2405.12163*.
- Lightman, Hunter, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.
- Lin, Lei, Jiayi Fu, Pengli Liu, Qingyang Li, Yan Gong, Junchen Wan, Fuzheng Zhang, Zhongyuan Wang, Di Zhang, and Kun Gai. 2024a. Just ask one more time! Self-agreement improves reasoning of language models in (almost) all scenarios. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3829–3852. <https://doi.org/10.18653/v1/2024.findings-acl.230>
- Lin, Tzu Han, Chen-An Li, Hung-yi Lee, and Yun-Nung Chen. 2024b. DogeRM: Equipping reward models with domain knowledge through model merging. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15506–15524. <https://doi.org/10.18653/v1/2024.emnlp-main.868>
- Lin, Yujie, Ante Wang, Moye Chen, Jingyao Liu, Hao Liu, Jinsong Su, and Xinyan Xiao. 2025. Investigating inference-time scaling for chain of multi-modal thought: A preliminary study. *arXiv preprint arXiv:2502.11514*. <https://doi.org/10.18653/v1/2025.findings-acl.808>
- Lin, Zicheng, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. 2024c. CriticBench: Benchmarking LLMs for critique-correct reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1552–1587. <https://doi.org/10.18653/v1/2024.findings-acl.91>
- Liu, Fan, Wenshuo Chao, Naiqiang Tan, and Hao Liu. 2025a. Bag of tricks for inference-time computation of LLM reasoning. *arXiv preprint arXiv:2502.07191*. <https://doi.org/10.52202/085713-4815>
- Liu, Fangfu, Hanyang Wang, Yimo Cai, Kaiyan Zhang, Xiaohang Zhan, and Yueqi Duan. 2025b. Video-T1: Test-time scaling for video generation. *arXiv preprint arXiv:2503.18942*. <https://doi.org/10.1109/ICCV51701.2025.01735>
- Liu, Jiachang, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114. <https://doi.org/10.18653/v1/2022.deeLIO-1.10>
- Liu, Jiawei, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2024a. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36. <https://doi.org/10.52202/075280-0943>
- Liu, Mingjie, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025c. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

- <https://doi.org/10.52202/085713-0608>, PubMed: 40922072
- Liu, Sheng, Haotian Ye, Lei Xing, and James Zou. 2024b. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*.
- Liu, Tengxiao, Qipeng Guo, Xiangkun Hu, Cheng Jiayang, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2024c. Can language models learn to skip steps? *Advances in Neural Information Processing Systems*, 37:45359–45385. <https://doi.org/10.52202/079017-1441>
- Liu, Tongxuan, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. 2024d. GroupDebate: Enhancing the efficiency of multi-agent debate using group discussion. *arXiv preprint arXiv:2409.14051*.
- Liu, Yang, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522. <https://doi.org/10.18653/v1/2023.emnlp-main.153>
- Liu, Yantao, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2024e. RM-Bench: Benchmarking reward models of language models with subtlety and style. *arXiv preprint arXiv:2410.16184*.
- Liu, Yantao, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2025d. Pairwise RM: Perform Best-of-N sampling with knockout tournament. *arXiv preprint arXiv:2501.13007*.
- Liu, Yuejiang, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. 2021. TTT++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820.
- Liu, Yufang, Yao Du, Tao Ji, Jianing Wang, Yang Liu, Yuanbin Wu, Aimin Zhou, Mengdi Zhang, and Xunliang Cai. 2025e. The role of visual modality in multimodal mathematical reasoning: Challenges and insights. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22596–22611. <https://doi.org/10.18653/v1/2025.acl-long.1102>
- Liu, Yuliang, Junjie Lu, Zhaoling Chen, Chaofeng Qu, Jason Klein Liu, Chonghan Liu, Zefan Cai, Yunhui Xia, Li Zhao, Jiang Bian, Chuheng Zhang, Wei Shen, and Zhouhan Lin. 2025f. AdaptiveStep: Automatically dividing reasoning step through model confidence. *arXiv preprint arXiv:2502.13943*.
- Liu, Zichen, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025g. Understanding R1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.
- Liu, Zihan, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025h. AceMath: Advancing frontier math reasoning with post-training and reward modeling. *arXiv preprint arXiv:2412.15084*. <https://doi.org/10.18653/v1/2025.findings-acl.206>
- Liu, Zijun, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025i. Inference-time scaling for generalist reward modeling. *arXiv preprint arXiv:2504.02495*.
- Long, Jieyi. 2023. Large language model guided tree-of-thought. *arXiv preprint arXiv:2305.08291*.
- Lou, Chenwei, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. 2025. AdaCoT: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*.
- Lu, Dawn and Nina Rimskey. 2024. Investigating bias representations in Llama 2 Chat via activation steering. *arXiv preprint arXiv:2402.00402*.
- Lu, Jianqiao, Zhiyang Dou, Hongru WANG, Zeyu Cao, Jianbo Dai, Yunlong Feng, and Zhijiang Guo. 2024. AutoPSV: Automated process-supervised verifier. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/079017-2540>
- Lu, Yao, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098. <https://doi.org/10.18653/v1/2022.acl-long.556>
- Luo, Haoran, Haihong E., Yikai Guo, Qika Lin, Xiaobao Wu, Xinyu Mu, Wenhao Liu, Meina Song, Yifan Zhu, and Luu Anh Tuan. 2025a. KBQA-o1: Agentic Knowledge Base Question Answering with Monte Carlo Tree Search. *arXiv preprint arXiv:2501.18922*.

- Luo, Haotian, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. 2025b. Ada-RL: Hybrid-CoT via bi-level adaptive reasoning optimization. *arXiv preprint arXiv:2504.21659*. <https://doi.org/10.52202/085713-1986>
- Luo, Haotian, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025c. O1-Pruner: Length-harmonizing fine-tuning for O1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*. <https://doi.org/10.18653/v1/2026.findings-acl.697>
- Luo, Liangchen, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*.
- Luo, Man, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaitė, and Vincent Y. Zhao. 2023. Dr.ICL: Demonstration-retrieved in-context learning. *arXiv preprint arXiv:2305.14128*. <https://doi.org/10.3724/2096-7004.di.2024.0012>
- Luo, Zheheng, Qianqian Xie, and Sophia Ananiadou. 2023. ChatGPT as a factual inconsistency evaluator for text summarization. *arXiv preprint arXiv:2303.15621*.
- Lyu, Xinxu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Z-ICL: Zero-shot in-context learning with pseudo-demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2304–2317. <https://doi.org/10.18653/v1/2023.acl-long.129>
- Ma, Xiaosong, Jie Zhang, Song Guo, and Wenchao Xu. 2023. SwapPrompt: Test-time prompt adaptation for vision-language models. *Advances in Neural Information Processing Systems*, 36:65252–65264. <https://doi.org/10.52202/075280-2847>
- Ma, Xinyin, Guangnian Wan, Runkeng Yu, Gongfan Fang, and Xinchao Wang. 2025a. CoT-Valve: Length-compressible chain-of-thought tuning. *arXiv preprint arXiv:2502.09601*. <https://doi.org/10.18653/v1/2025.acl-long.300>
- Ma, Yan, Steffi Chern, Xuyang Shen, Yiran Zhong, and Pengfei Liu. 2025b. Rethinking RL scaling for vision language models: A transparent, from-scratch framework and comprehensive evaluation scheme. *arXiv preprint arXiv:2504.02587*.
- Madaan, Aman, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/075280-2019>
- Manvi, Rohin, Anikait Singh, and Stefano Ermon. 2024. Adaptive inference-time compute: LLMs can predict if they can do better, even mid-generation. *arXiv preprint arXiv:2410.02725*.
- Meng, Silin, Yiwei Wang, Cheng-Fu Yang, Nanyun Peng, and Kai-Wei Chang. 2024. LLM-A\*: Large language model enhanced incremental heuristic search on path planning. *arXiv preprint arXiv:2407.02511*. <https://doi.org/10.21203/rs.3.rs-4613568/v1>
- MiniMax: Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. 2025. MiniMax-M1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*.
- Mondal, Debijoti, Suraj Modi, Subhadarshi Panda, Rituraj Singh, and Godawari Sudhakar Rao. 2024. KAM-CoT: Knowledge augmented multimodal chain-of-thoughts reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18798–18806. <https://doi.org/10.1609/aaai.v38i17.29844>
- Muennighoff, Niklas, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. sl: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*. <https://doi.org/10.18653/v1/2025.emnlp-main.1025>
- Muhtar, Dilxat, Yelong Shen, Yaming Yang, Xiaodong Liu, Yadong Lu, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Weiwei Deng, Feng Sun, et al. 2024. StreamAdapter: Efficient test time adaptation from contextual streams. *arXiv preprint arXiv:2411.09289*. <https://doi.org/10.32388/G5H95J>
- Munkhbat, Tergel, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. Self-training elicits

- concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*. <https://doi.org/10.18653/v1/2025.findings-acl.1289>
- Nakano, Reiichiro, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2022. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.
- Nathani, Deepak, David Wang, Liangming Pan, and William Wang. 2023. MAF: Multi-aspect feedback for improving reasoning in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6591–6616. <https://doi.org/10.18653/v1/2023.emnlp-main.407>
- Niu, Shuaicheng, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. 2024. Test-time model adaptation with only forward passes. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 38298–38315.
- Niu, Shuaicheng, Jiayang Wu, Yifan Zhang, Zhiqian Wen, Yaofu Chen, Peilin Zhao, and Minghui Tan. 2023. Towards stable test-time adaptation in dynamic wild world. In *The Eleventh International Conference on Learning Representations*.
- Olausson, Theo X., Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. 2024. Is self-repair a silver bullet for code generation? In *The Twelfth International Conference on Learning Representations*.
- OpenAI. 2023. GPT-4 technical report. *ArXiv*, abs/2303.08774.
- OpenAI. 2024. Learning to reason with LLMs. Open AI, blog.
- OpenAI. 2025. Introducing deep research. Open AI, blog.
- Ouyang, Long, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*. <https://doi.org/10.52202/068431-2011>
- Park, Sungjin, Xiao Liu, Yeyun Gong, and Edward Choi. 2024. Ensembling large language models with process reward-guided tree search for better complex reasoning. *arXiv preprint arXiv:2412.15797*. <https://doi.org/10.18653/v1/2025.naacl-long.515>
- Paul, Debjit, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2024. REFINER: Reasoning feedback on intermediate representations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1100–1126. <https://doi.org/10.18653/v1/2024.eacl-long.67>
- Peebles, William and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205. <https://doi.org/10.1109/ICCV51070.2023.00387>
- Peng, Hao, Yunjia Qi, Xiaozhi Wang, Zijun Yao, Bin Xu, Lei Hou, and Juanzi Li. 2025a. Agentic reward modeling: Integrating human preferences with verifiable correctness signals for reliable reward systems. *arXiv preprint arXiv:2502.19328*. <https://doi.org/10.18653/v1/2025.acl-long.775>
- Peng, Yingzhe, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. 2025b. LMM-R1: Empowering 3b LMMs with strong reasoning abilities through two-stage rule-based RL. *arXiv preprint arXiv:2503.07536*.
- Press, Ori, Ravid Shwartz-Ziv, Yann LeCun, and Matthias Bethge. 2024. The entropy enigma: Success and failure of entropy minimization. *arXiv preprint arXiv:2405.05012*.
- Qi, Jianing, Hao Tang, and Zhigang Zhu. 2024. VerifierQ: Enhancing LLM test time compute with q-learning-based verifiers. *arXiv preprint arXiv:2410.08048*.
- Qi, Yukun, Yiming Zhao, Yu Zeng, Xikun Bao, Wenxuan Huang, Lin Chen, Zehui Chen, Jie Zhao, Zhongang Qi, and Feng Zhao. 2025. VCR-Bench: A comprehensive evaluation framework for video chain-of-thought reasoning. *arXiv preprint arXiv:2504.07956*.
- Qi, Zhenting, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller LLMs stronger problem-solvers. *arXiv preprint arXiv:2408.06195*.
- Qin, Chengwei, Aston Zhang, Chen Chen, Anirudh Dagar, and Wenming Ye. 2024a. In-context learning with iterative

- demonstration selection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7441–7455. <https://doi.org/10.18653/v1/2024.findings-emnlp.438>
- Qin, Yiwei, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, and Pengfei Liu. 2024b. O1 replication journey: A strategic progress report – Part 1. *arXiv preprint arXiv:2410.18982*.
- Qiu, Jiahao, Yifu Lu, Yifan Zeng, Jiacheng Guo, Jiayi Geng, Huazheng Wang, Kaixuan Huang, Yue Wu, and Mengdi Wang. 2024a. TreeBoN: Enhancing inference-time alignment with speculative tree-search and Best-of-N sampling. *arXiv preprint arXiv:2410.16033*. <https://doi.org/10.18653/v1/2025.findings-emnlp.1140>
- Qiu, Yifu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo Ponti, and Shay B. Cohen. 2024b. Spectral editing of activations for large language model alignment. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/079017-1815>
- Qu, Yuxiao, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. <https://doi.org/10.52202/079017-1754>
- Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763.
- Rafailov, Rafael, 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 *Thirty-seventh Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/075280-2338>
- Rei, Ricardo, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702. <https://doi.org/10.18653/v1/2020.emnlp-main.213>
- Rein, David, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level Google-proof Q&A benchmark. In *First Conference on Language Modeling*.
- Rimsky, Nina, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522. <https://doi.org/10.18653/v1/2024.acl-long.828>
- Rozière, Baptiste, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. 2024. Code Llama: Open foundation models for code.
- Rubin, Ohad, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671. <https://doi.org/10.18653/v1/2022.naacl-main.191>
- Saha, Swarnadeep, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2024. Branch-solve-merge improves large language model evaluation and generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8352–8370. <https://doi.org/10.18653/v1/2024.naacl-long.462>
- Saha, Swarnadeep, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. 2025. Learning to plan & reason for evaluation with Thinking-LLM-as-a-Judge. *arXiv preprint arXiv:2501.18099*.
- Sang, Hejian, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. 2026. CRISP: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*.
- Saunders, William, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*
- Scalena, Daniel, Gabriele Sarti, and Malvina Nissim. 2024. Multi-property steering of large language models with dynamic activation composition. In *Proceedings of the*

- 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 577–603. <https://doi.org/10.18653/v1/2024.blackboxnlp-1.34>
- Scarlatos, Alexander and Andrew Lan. 2024. RetlCL: Sequential retrieval of in-context examples with reinforcement learning. *arXiv preprint arXiv:2305.14502*.
- Schneider, Steffen, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. 2020. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33:11539–11551.
- Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Schwenk, Holger, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500. <https://doi.org/10.18653/v1/2021.acl-long.507>
- Sessa, Pier Giuseppe, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shariari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, et al. 2024. BOND: Aligning LLMs with Best-of-N Distillation. *arXiv preprint arXiv:2407.14622*.
- Setlur, Amrith, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for LLM reasoning. *arXiv preprint arXiv:2410.08146*.
- Setlur, Amrith, Matthew Y. R. Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. 2025. e3: Learning to explore enables extrapolation of test-time compute for LLMs. *arXiv preprint arXiv:2506.09026*.
- Shao, Zhihong, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- She, Shuaijie, Junxiao Liu, Yifeng Liu, Jiajun Chen, Xin Huang, and Shujian Huang. 2025. R-PRM: Reasoning-driven process reward modeling. *arXiv preprint arXiv:2503.21295*. <https://doi.org/10.18653/v1/2025.emnlp-main.679>
- Shen, Haozhan, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. 2025a. VLM-R1: A stable and generalizable R1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.
- Shen, Zhenyi, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025b. CODI: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*. <https://doi.org/10.18653/v1/2025.emnlp-main.36>
- Shenfeld, Idan, Jyothish Pari, and Pulkit Agrawal. 2025. RL’s razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*.
- Shinn, Noah, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*. <https://doi.org/10.52202/075280-0377>
- Shu, Manli, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. 2022. Test-time prompt tuning for zero-shot generalization in vision-language models. *Advances in Neural Information Processing Systems*, 35:14274–14289. <https://doi.org/10.52202/068431-1038>
- Snell, Charlie, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.
- Son, Guijin, Jiwoo Hong, Hyunwoo Ko, and James Thorne. 2025. Linguistic generalizability of test-time scaling in mathematical reasoning. *arXiv preprint arXiv:2502.17407*. <https://doi.org/10.18653/v1/2025.acl-long.699>
- Song, Mingyang, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025a. PRMBench: A fine-grained and challenging benchmark for process-level reward models. *arXiv preprint arXiv:2501.03124*. <https://doi.org/10.18653/v1/2025.acl-long.1230>
- Song, Xiaoshuai, Yanan Wu, Weixun Wang, Jiaheng Liu, Wenbo Su, and Bo Zheng. 2025b. ProgCo: Program helps self-correction of large language models.

- arXiv preprint arXiv:2501.01264*.  
<https://doi.org/10.18653/v1/2025.acl-short.73>
- Sprague, Zayne, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*.
- Stechly, Kaya, Matthew Marquez, and Subbarao Kambhampati. 2023. GPT-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems. *arXiv preprint arXiv:2310.12397*.
- Stechly, Kaya, Karthik Valmeekam, and Subbarao Kambhampati. 2024. Chain of thoughtlessness? An analysis of CoT in planning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/079017-0917>
- Stengel-Eskin, Elias, Peter Hase, and Mohit Bansal. 2024. Teaching models to balance resisting and accepting persuasion. *arXiv preprint arXiv:2410.14596*. <https://doi.org/10.18653/v1/2025.naacl-long.412>
- Stiennon, Nisan, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.
- Stolfo, Alessandro, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. 2024. Improving instruction-following in language models through activation steering. *arXiv preprint arXiv:2410.12877*. <https://doi.org/10.32388/BTHT4K>
- Stureborg, Rickard, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *arXiv preprint arXiv:2405.01724*.
- Su, Hongjin, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.
- Su, Yi, Yixin Ji, Juntao Li, Hai Ye, and Min Zhang. 2023. Beware of model collapse! Fast and stable test-time adaptation for robust question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12998–13011. <https://doi.org/10.18653/v1/2023.emnlp-main.803>
- Su, Yi, Yunpeng Tai, Yixin Ji, Juntao Li, Yan Bowen, and Min Zhang. 2024. Demonstration augmentation for zero-shot in-context learning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14232–14244. <https://doi.org/10.18653/v1/2024.findings-acl.846>
- Su, Yi, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. 2025. Crossing the reward bridge: Expanding RL with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*. <https://doi.org/10.18653/v1/2026.acl-long.178>
- Su, Zhenpeng, Lei Yu Pan, Xue Bai, Dening Liu, Guanting Dong, Jiaming Huang, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. 2026. Klear-Reasoner: Advancing reasoning capability via gradient-preserving clipping policy optimization. *arXiv preprint arXiv:2508.07629*.
- Subramaniam, Vighnesh, Yilun Du, Joshua B. Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. 2025. Multiagent finetuning of language models. In *The Thirteenth International Conference on Learning Representations*.
- Sullivan, Michael and Alexander Koller. 2026. GRPO is secretly a process reward model. *arXiv preprint arXiv:2509.21154*.
- Sun, Hanshi, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast Best-of-N decoding via speculative rejection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/079017-1025>
- Sun, Wei, Qianlong Du, Fuwei Cui, and Jiajun Zhang. 2025. An efficient and precise training data construction framework for process-supervised reward model in mathematical reasoning. *arXiv preprint arXiv:2503.02382*. <https://doi.org/10.18653/v1/2025.acl-long.216>, PubMed: 40131626
- Sun, Yu, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248.

- Sun, Zhishen, Guang Dai, and Haishan Ye. 2026. MSCR: Exploring the vulnerability of LLMs' mathematical reasoning abilities using multi-source candidate replacement. In *The Fourteenth International Conference on Learning Representations*.
- Tandon, Niket, Aman Madaan, Peter Clark, Keisuke Sakaguchi, and Yiming Yang. 2021. Interscript: A dataset for interactive learning of scripts through error feedback. *arXiv preprint arXiv:2112.07867*.
- Tandon, Niket, Aman Madaan, Peter Clark, and Yiming Yang. 2022. Learning to repair: Repairing model output errors after deployment using a dynamic memory of feedback. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 339–352. <https://doi.org/10.18653/v1/2022.findings-naacl.26>
- Tang, Xinyu, Xiaolei Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2024. Dawn-ICL: Strategic planning of problem-solving trajectories for zero-shot in-context learning. *arXiv preprint arXiv:2410.20215*. <https://doi.org/10.18653/v1/2025.naacl-long.96>
- Tang, Zhengyang, Ziniu Li, Zhenyang Xiao, Tian Ding, Ruoyu Sun, Benyou Wang, Dayiheng Liu, Fei Huang, Tianyu Liu, Bowen Yu, and Junyang Lin. 2025. Enabling scalable oversight via self-evolving critic. *arXiv preprint arXiv:2501.05727*.
- Team, Kimi, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. Kimi k1. 5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*.
- Tian, Keyu, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. 2024. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/079017-2694>
- Toh, Vernon Y. H., Deepanway Ghosal, and Soujanya Poria. 2024. Not all votes count! Programs as verifiers improve self-consistency of language models for math reasoning. *arXiv preprint arXiv:2410.12608*.
- Tong, Jingqi, Yurong Mou, Hangcheng Li, Mingzhe Li, Yongzhuo Yang, Ming Zhang, Qiguang Chen, Tianyi Liang, Xiaomeng Hu, Yining Zheng, et al. 2025. Thinking with video: Video generation as a promising multimodal reasoning paradigm. *arXiv preprint arXiv:2511.04570*.
- Trinh, Trieu H., Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. 2024. Solving Olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482. <https://doi.org/10.1038/s41586-023-06747-5>, PubMed: 38233616
- Trivedi, Harsh, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037. <https://doi.org/10.18653/v1/2023.acl-long.557>
- Tu, Haoqin, Weitao Feng, Hardy Chen, Hui Liu, Xianfeng Tang, and Cihang Xie. 2025. VILBENCH: A suite for vision-language process reward modeling. *arXiv preprint arXiv:2503.20271*. <https://doi.org/10.18653/v1/2025.emnlp-main.344>
- Turner, Alexander Matt, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.
- Tyen, Gladys, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13894–13908. <https://doi.org/10.18653/v1/2024.findings-acl.826>
- Uesato, Jonathan, 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*.
- Valmeekam, Karthik, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118*.
- Valmeekam, Karthik, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. <https://doi.org/10.52202/075280-1693>
- Venhoff, Constantin, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda.

2025. Understanding reasoning in thinking language models via steering vectors. In *Workshop on Reasoning and Planning for Large Language Models*.
- Villalobos, Pablo, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Will we run out of data? Limits of LLM scaling based on human-generated data. *arXiv preprint arXiv:2211.04325*.
- Wang, Chaojie, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. 2024a. Q\*: Improving multi-step reasoning for LLMs with deliberative planning. *arXiv preprint arXiv:2406.14283*.
- Wang, Dequan, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2021. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.
- Wang, Dexin, Kai Fan, Boxing Chen, and Deyi Xiong. 2022a. Efficient cluster-based k-nearest-neighbor machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2175–2187. <https://doi.org/10.18653/v1/2022.acl-long.154>
- Wang, Hanlin, Chak Tou Leong, Jian Wang, and Wenjie Li. 2024b. E2CL: Exploration-based error correction learning for embodied agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7626–7639. <https://doi.org/10.18653/v1/2024.findings-emnlp.448>
- Wang, Haohan, Songwei Ge, Zachary Lipton, and Eric P. Xing. 2019. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32.
- Wang, Haoxiang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024c. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*. <https://doi.org/10.18653/v1/2024.findings-emnlp.620>
- Wang, Jiaan, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is ChatGPT a good NLG evaluator? A preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 1–11. <https://doi.org/10.18653/v1/2023.newsum-1.1>
- Wang, Junlin, Siddhartha Jain, Dejiao Zhang, Baishakhi Ray, Varun Kumar, and Ben Athiwaratkun. 2024d. Reasoning in token economies: Budget-aware evaluation of LLM reasoning strategies. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19916–19939. <https://doi.org/10.18653/v1/2024.emnlp-main.1112>
- Wang, Junlin, Siddhartha Jain, Dejiao Zhang, Baishakhi Ray, Varun Kumar, and Ben Athiwaratkun. 2024e. Reasoning in token economies: Budget-aware evaluation of LLM reasoning strategies. *arXiv preprint arXiv:2406.06461*. <https://doi.org/10.18653/v1/2024.emnlp-main.1112>
- Wang, Junlin, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024f. Mixture-of-agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*.
- Wang, Peiyi, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024g. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450. <https://doi.org/10.18653/v1/2024.acl-long.511>
- Wang, Peiyi, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024h. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439. <https://doi.org/10.18653/v1/2024.acl-long.510>
- Wang, Shenzhi, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. 2025a. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. *arXiv preprint arXiv:2506.01939*. <https://doi.org/10.52202/085713-3850>
- Wang, Shuhe, Xiaoya Li, Yuxian Meng, Tianwei Zhang, Rongbin Ouyang, Jiwei Li, and Guoyin Wang. 2022b. kNN-ner: Named entity recognition with nearest neighbor search. *arXiv preprint arXiv:2203.17103*.
- Wang, Tianlu, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024i. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*.

- Wang, Tianlu, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023b. Shepherd: A critic for language model generation. *arXiv preprint arXiv:2308.04592*.
- Wang, Weiyun, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, et al. 2025b. VisualPRM: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*.
- Wang, Xinyi, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023c. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/075280-0687>
- Wang, Xinyu, Changzhi Sun, Lian Cheng, Yuanbin Wu, Dell Zhang, Xiaoling Wang, and Xuelong Li. 2025c. Logic-regularized verifier elicits reasoning from LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32617–32630. <https://doi.org/10.18653/v1/2025.acl-long.1567>
- Wang, Xiyao, Linfeng Song, Ye Tian, Dian Yu, Baolin Peng, Haitao Mi, Furong Huang, and Dong Yu. 2024j. Towards self-improvement of LLMs via MCTS: Leveraging stepwise knowledge with curriculum preference learning. *arXiv preprint arXiv:2410.06508*.
- Wang, Xuezhi, Haohan Wang, and Diyi Yang. 2022. Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586. <https://doi.org/10.18653/v1/2022.naacl-main.339>
- Wang, Xuezhi, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023d. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.
- Wang, Xuezhi and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *Advances in Neural Information Processing Systems*, 37:66383–66409. <https://doi.org/10.52202/079017-2123>
- Wang, Yan, Dongyang Ma, and Deng Cai. 2024. With greater text comes greater necessity: Inference-time training helps long text generation. In *First Conference on Language Modeling*.
- Wang, Yidong, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2024k. PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization. In *The Twelfth International Conference on Learning Representations*.
- Wang, Yubo, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhrranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. 2024l. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. <https://doi.org/10.52202/079017-3018>
- Wang, Yue, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025d. Thoughts are all over the place: On the underthinking of o1-like LLMs. *arXiv preprint arXiv:2501.18585*. <https://doi.org/10.52202/085713-1027>
- Wang, Zihao, Anji Liu, Haowei Lin, Jiaqi Li, Xiaojian Ma, and Yitao Liang. 2024m. RAT: Retrieval augmented thoughts elicit context-aware reasoning in long-horizon generation. *arXiv preprint arXiv:2403.05313*.
- Wei, Anjiang, Jiannan Cao, Ran Li, Hongyu Chen, Yuhui Zhang, Ziheng Wang, Yuan Liu, Thiago S. F. X. Teixeira, Diyi Yang, Ke Wang, and Alex Aiken. 2025. EquiBench: Benchmarking large language models' understanding of program semantics via equivalence checking. *arXiv preprint arXiv:2502.12466*. <https://doi.org/10.18653/v1/2025.emnlp-main.1718>
- Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*. <https://doi.org/10.52202/068431-1800>
- Welleck, Sean, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating

- sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.
- Weston, Jason and Sainbayar Sukhbaatar. 2023. System 2 Attention (is something you might need too). *arXiv preprint arXiv:2311.11829*.
- Wortsman, Mitchell, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998.
- Wu, Han, Yuxuan Yao, Shuqi Liu, Zehua Liu, Xiaojin Fu, Xiongwei Han, Xing Li, Hui-Ling Zhen, Tao Zhong, and Mingxuan Yuan. 2025. Unlocking efficient long-to-short LLM reasoning with model merging. *arXiv preprint arXiv:2503.20641*.
- Wu, Jinyang, Mingkuan Feng, Shuai Zhang, Feihu Che, Zengqi Wen, and Jianhua Tao. 2024a. Beyond examples: High-level automated reasoning paradigm in In-context Learning via MCTS. *arXiv preprint arXiv:2411.18478*.
- Wu, Wenshan, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024b. Mind’s eye of LLMs: Visualization-of-thought elicits spatial reasoning in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/079017-2866>
- Wu, Xiaobao. 2025. Sailing by the stars: A survey on reward models and learning strategies for learning from rewards. *arXiv preprint arXiv:2505.02686*.
- Wu, Yangzhen, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024c. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*.
- Wu, Zhiyong, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1423–1436. <https://doi.org/10.18653/v1/2023.acl-long.79>
- Xi, Zhiheng, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Do, Wenyu Zhan, et al. 2024. Enhancing LLM reasoning via critique models with test-time and training-time supervision. *arXiv preprint arXiv:2411.16579*.
- Xia, Heming, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. TokenSkip: Controllable chain-of-thought compression in LLMs. *arXiv preprint arXiv:2502.12067*. <https://doi.org/10.18653/v1/2025.emnlp-main.165>
- Xia, Heming, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *arXiv preprint arXiv:2401.07851*. <https://doi.org/10.18653/v1/2024.findings-acl.456>
- Xiang, Violet, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, et al. 2025. Towards System 2 reasoning in LLMs: Learning how to think with Meta Chain-of-Thought. *arXiv preprint arXiv:2501.04682*.
- Xiang, Yang, Yixin Ji, Ruotao Xu, Dan Qiao, Zheming Yang, Juntao Li, and Min Zhang. 2026. When is thinking enough? Early exit via sufficiency assessment for efficient reasoning. *arXiv preprint arXiv:2604.06787*. <https://doi.org/10.18653/v1/2026.acl-long.1080>
- Xiao, Wenyi, Leilei Gan, Weilong Dai, Wanggui He, Ziwei Huang, Haoyuan Li, Fangxun Shu, Zhelun Yu, Peng Zhang, Hao Jiang, and Fei Wu. 2025. Fast-slow thinking for large vision-language model reasoning. *arXiv preprint arXiv:2504.18458*. <https://doi.org/10.52202/085713-5712>
- Xie, Enze, Junsong Chen, Yuyang Zhao, Jincheng Yu, Ligeng Zhu, Yujun Lin, Zhekai Zhang, Muyang Li, Junyu Chen, Han Cai, Bingchen Liu, Daquan Zhou, and Song Han. 2025a. SANA 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *arXiv preprint arXiv:2501.18427*.
- Xie, Jinheng, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. 2025b. Show-o: One single transformer to unify multimodal understanding and generation. In *The Thirteenth International Conference on Learning Representations*.

- Xie, Yuxi, Anirudh Goyal, Wenye Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte Carlo Tree Search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*.
- Xie, Yuxi, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning. In *Thirty-seventh Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/075280-1802>
- Xie, Zhifei, Mingbao Lin, Zihang Liu, Pengcheng Wu, Shuicheng Yan, and Chunyan Miao. 2025c. Audio-Reasoner: Improving reasoning capability in large audio language models. *arXiv preprint arXiv:2503.02318*. <https://doi.org/10.18653/v1/2025.emnlp-main.1216>
- Xiong, Kai, Xiao Ding, Yixin Cao, Ting Liu, and Bing Qin. 2023. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7572–7590. <https://doi.org/10.18653/v1/2023.findings-emnlp.508>
- Xiong, Tianyi, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. 2024. LLaVA-Critic: Learning to evaluate multimodal models. *arXiv preprint arXiv:2410.02712*. <https://doi.org/10.1109/CVPR52734.2025.01271>
- Xu, Bin, Yiguan Lin, Yinghao Li, and Yang Gao. 2024a. SRA-MCTS: Self-driven reasoning augmentation with Monte Carlo Tree Search for code generation. *arXiv preprint arXiv:2411.11053*. <https://doi.org/10.24963/ijcai.2024/965>
- Xu, Guowei, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2024b. LLaVA-CoT: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*. <https://doi.org/10.1109/ICCV51701.2025.00202>
- Xu, Haotian. 2023. No train still gain. Unleash mathematical reasoning of large language models with Monte Carlo Tree Search guided by energy function. *arXiv preprint arXiv:2309.03224*.
- Xu, Silei, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025a. Chain of Draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.
- Xu, Yige, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025b. SoftCoT: Soft Chain-of-Thought for efficient reasoning with LLMs. *arXiv preprint arXiv:2502.12134*. <https://doi.org/10.18653/v1/2025.ac1-long.1137>
- Xu, Yige, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025c. SoftCoT++: Test-Time Scaling with Soft Chain-of-Thought reasoning. *arXiv preprint arXiv:2505.11484*.
- Xu, Yuhui, Hanze Dong, Lei Wang, Doyen Sahoo, Junnan Li, and Caiming Xiong. 2025d. Scalable chain of thoughts via elastic reasoning. *arXiv preprint arXiv:2505.05315*.
- Yang, Chenxu, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2026. Dynamic early exit in reasoning models. In *The Fourteenth International Conference on Learning Representations*.
- Yang, Chenxu, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Zheng Lin, Li Cao, and Weiping Wang. 2025a. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.
- Yang, Rui, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024a. Regularizing hidden states enables learning generalizable reward model for LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/079017-1990>
- Yang, Wenkai, Shuming Ma, Yankai Lin, and Furu Wei. 2025b. Towards thinking-optimal scaling of test-time compute for LLM reasoning. *arXiv preprint arXiv:2502.18080*. <https://doi.org/10.52202/085713-1452>
- Yang, Zhe, Yichang Zhang, Yudong Wang, Ziyao Xu, Junyang Lin, and Zhifang Sui. 2024b. Confidence v.s. critique: A decomposition of self-correction capability for LLMs. *arXiv preprint arXiv:2412.19513*.
- Yang, Zhuoyi, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. 2025c. CogVideoX: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*.
- Yao, Huanjin, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, and Dacheng Tao. 2024. Mulberry: Empowering MLLM with o1-like reasoning and reflection via collective Monte Carlo Tree Search. *arXiv*

- preprint *arXiv:2412.18319*. <https://doi.org/10.52202/085713-1004>
- Yao, Shunyu, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36:11809–11822. <https://doi.org/10.52202/075280-0517>
- Yasunaga, Michihiro and Percy Liang. 2020. Graph-based, self-supervised program repair from diagnostic feedback. In *International Conference on Machine Learning (ICML)*.
- Yasunaga, Michihiro, Luke Zettlemoyer, and Marjan Ghazvininejad. 2025. Multimodal RewardBench: Holistic evaluation of reward models for vision language models. *arXiv preprint arXiv:2502.14191*.
- Ye, Hai, Yuyang Ding, Juntao Li, and Hwee Tou Ng. 2022. Robust question answering against distribution shifts with test-time adaption: An empirical study. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6179–6192. <https://doi.org/10.18653/v1/2022.findings-emnlp.460>
- Ye, Hai and Hwee Tou Ng. 2024. Preference-guided reflective sampling for aligning language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21646–21668. <https://doi.org/10.18653/v1/2024.emnlp-main.1206>
- Ye, Hai, Qizhe Xie, and Hwee Tou Ng. 2023. Multi-source test-time adaptation as dueling bandits for extractive question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9647–9660. <https://doi.org/10.18653/v1/2023.acl-long.537>
- Ye, Zihuiwen, Fraser Greenlee-Scott, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gallé. 2024. Improving reward models with synthetic critiques. *arXiv preprint arXiv:2405.20850*. <https://doi.org/10.18653/v1/2025.findings-naacl.254>
- Yeo, Edward, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in LLMs. *arXiv preprint arXiv:2502.03373*.
- Yi, Jingyang, Jiazheng Wang, and Sida Li. 2025. ShorterBetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *arXiv preprint arXiv:2504.21370*. <https://doi.org/10.52202/085713-1302>
- You, WangJie, Pei Guo, Juntao Li, Kehai Chen, and Min Zhang. 2024. Efficient domain adaptation for non-autoregressive machine translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13657–13670. <https://doi.org/10.18653/v1/2024.findings-acl.810>
- Yu, Fei, Yingru Li, and Benyou Wang. 2025. Scaling flaws of verifier-guided search in mathematical reasoning. *arXiv preprint arXiv:2502.00271*.
- Yu, Hao, Bo Shen, Dezhi Ran, Jiaxin Zhang, Qi Zhang, Yuchi Ma, Guangtai Liang, Ying Li, Qianxiang Wang, and Tao Xie. 2024a. CoderEval: A benchmark of pragmatic code generation with generative pre-trained models. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*, pages 1–12. <https://doi.org/10.1145/3597503.3623316>
- Yu, Jiachen, Shaoning Sun, Xiaohui Hu, Jiaxu Yan, Kaidong Yu, and Xuelong Li. 2025a. Improve LLM-as-a-Judge ability as a general ability. *arXiv preprint arXiv:2502.00271*. <https://doi.org/10.18653/v1/2025.emnlp-main.712>
- Yu, Ping, Jing Xu, Jason Weston, and Ilia Kulikov. 2024b. Distilling System 2 into System 1. *arXiv preprint arXiv:2407.06023*.
- Yu, Qiyang, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, GaoHong Liu, Juncai Liu, et al. 2025b. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. <https://doi.org/10.52202/085713-3775>
- Yu, Yongcan, Lijun Sheng, Ran He, and Jian Liang. 2023. Benchmarking test-time adaptation against distribution shifts in image classification. *arXiv preprint arXiv:2307.03133*.
- Yu, Yue, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuwei Wang, Suchin Gururangan, Chao Zhang, Melanie Kambadur, Dhruv Mahajan, and Rui Hou. 2024c. Self-generated critiques boost reward modeling for language models. *arXiv preprint arXiv:2411.16646*. <https://doi.org/10.18653/v1/2025.naacl-long.573>
- Yu, Zewei, Lirong Gao, Sheng Guo, Yuke Zhu, Bo Zheng, Junbo Zhao, and Haobo

- Wang. 2026. Stop unnecessary reflection: Training LLMs for efficient reasoning with adaptive reflection and length coordinated penalty. In *The Fourteenth International Conference on Learning Representations*.
- Yuan, Lifan, Weize Chen, Yuchen Zhang, Ganqu Cui, Hanbin Wang, Ziming You, Ning Ding, Zhiyuan Liu, Maosong Sun, and Hao Peng. 2025a. From  $f(x)$  and  $g(x)$  to  $f(g(x))$ : LLMs learn new skills in RL by composing old ones. *arXiv preprint arXiv:2509.25123*.
- Yuan, Lifan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. 2024a. Advancing LLM reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*.
- Yuan, Lifan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. 2024b. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*.
- Yuan, Shuozhi, Liming Chen, Miaomiao Yuan, Jin Zhao, Haoran Peng, and Wenming Guo. 2025b. MCTS-SQL: An effective framework for text-to-SQL with Monte Carlo Tree Search. *arXiv preprint arXiv:2501.16607*.
- Yuan, Siyu, Zehui Chen, Zhiheng Xi, Junjie Ye, Zhengyin Du, and Jiecao Chen. 2025c. Agent-R: Training language model agents to reflect via iterative self-training. *arXiv preprint arXiv:2501.11425*.
- Yuan, Weizhe, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024c. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.
- Yue, Yang, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025a. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Yue, Yu, Yufeng Yuan, Qiyang Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. 2025b. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.
- Zamil, Parvez and Gollam Rabby. 2024. Aime problems 1983 to 2024. <https://doi.org/10.34740/KAGGLE/DSV/8834060>
- Zan, Daoguang, Zhirong Huang, Wei Liu, Hanwu Chen, Shulin Xin, Linhao Zhang, Qi Liu, Aoyan Li, Lu Chen, Xiaojian Zhong, et al. 2025. Multi-SWE-bench: A multilingual benchmark for issue resolving. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. <https://doi.org/10.52202/085713-2111>
- Zelikman, Eric, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*. <https://doi.org/10.52202/068431-1126>
- Zeng, Zhongshen, Yinhong Liu, Yingjia Wan, Jingyao Li, Pengguang Chen, Jianbo Dai, Yuxuan Yao, Rongwu Xu, Zehan Qi, Wanru Zhao, et al. 2024. MR-BEN: A comprehensive meta-reasoning benchmark for large language models. *arXiv preprint arXiv:2406.13975*.
- Zhai, Yuanzhao, Tingkai Yang, Kele Xu, Feng Dawei, Cheng Yang, Bo Ding, and Huaimin Wang. 2024. Enhancing decision-making for LLM agents via step-level Q-value models. *arXiv preprint arXiv:2409.09345*. <https://doi.org/10.1609/aaai.v39i25.34924>
- Zhan, Runzhe, Xuebo Liu, Derek F. Wong, Cuilian Zhang, Lidia S. Chao, and Min Zhang. 2023. Test-time adaptation for machine translation evaluation by uncertainty minimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 807–820. <https://doi.org/10.18653/v1/2023.acl-long.47>
- Zhang, Dan, Sining Zhou, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. ReST-MCTS\*: LLM self-training via process reward guided tree search. *arXiv preprint arXiv:2406.03816*. <https://doi.org/10.52202/079017-2066>
- Zhang, Di, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024b. Accessing GPT-4 level mathematical Olympiad solutions via Monte Carlo tree self-refine with LLaMa-3 8B. *arXiv preprint arXiv:2406.07394*.
- Zhang, Di, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, Wanli Ouyang, and Dongzhan Zhou. 2024c. LLaMa-Berry: Pairwise optimization for O1-like Olympiad-level mathematical reasoning. *arXiv preprint arXiv:2410.02884*. <https://doi.org/10.18653/v1/2025.naacl-long.375>
- Zhang, Fengji, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao,

- Jian-Guang Lou, and Weizhu Chen. 2023a. RepoCoder: Repository-level code completion through iterative retrieval and generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2471–2484. <https://doi.org/10.18653/v1/2023.emnlp-main.151>
- Zhang, Guibin, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024d. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. *arXiv preprint arXiv:2410.02506*.
- Zhang, Hanning, Pengcheng Wang, Shizhe Diao, Yong Lin, Rui Pan, Hanze Dong, Dylan Zhang, Pavlo Molchanov, and Tong Zhang. 2024e. Entropy-regularized process reward model. *arXiv preprint arXiv:2412.11006*.
- Zhang, Jiajie, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025a. AdaptThink: Reasoning models can learn when to think. *arXiv preprint arXiv:2505.13417*. <https://doi.org/10.18653/v1/2025.emnlp-main.184>
- Zhang, Jintian, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. 2025b. LightThinker: Thinking step-by-step compression. *arXiv preprint arXiv:2502.15589*. <https://doi.org/10.18653/v1/2025.emnlp-main.673>
- Zhang, Kexun, Shang Zhou, Danqing Wang, William Yang Wang, and Lei Li. 2024f. Scaling LLM inference with optimized sample compute allocation. *arXiv preprint arXiv:2410.22480*.
- Zhang, Lunjun, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024g. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*.
- Zhang, Marvin, Sergey Levine, and Chelsea Finn. 2022. MEMO: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, 35:38629–38642. <https://doi.org/10.52202/068431-2799>
- Zhang, Qingjie, Han Qiu, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, and Minlie Huang. 2024h. Understanding the dark side of LLMs’ intrinsic self-correction. *arXiv preprint arXiv:2412.14959*.
- Zhang, Ruiqi, Momin Haider, Ming Yin, Jiahao Qiu, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024i. Accelerating Best-of-N via speculative rejection. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ICML 2024)*.
- Zhang, Ruohong, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. 2024j. Improve vision language model chain-of-thought reasoning. *arXiv preprint arXiv:2410.16198*.
- Zhang, Wenyuan, Shuaiyi Nie, Xinghua Zhang, Zefeng Zhang, and Tingwen Liu. 2025c. S1-Bench: A simple benchmark for evaluating System 1 thinking capability of large reasoning models. *arXiv preprint arXiv:2504.10368*.
- Zhang, Xiaotian, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. 2023b. Evaluating the performance of large language models on GAOKAO benchmark. *arXiv preprint arXiv:2305.12474*.
- Zhang, Yifan, Xue Wang, Kexin Jin, Kun Yuan, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. 2023c. AdaNPC: Exploring non-parametric classifier for test-time adaptation. In *International Conference on Machine Learning*, pages 41647–41676.
- Zhang, Yiming, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148. <https://doi.org/10.18653/v1/2022.emnlp-main.622>
- Zhang, Yuxiang, Shangxi Wu, Yuqi Yang, Jiangming Shu, Jinlin Xiao, Chao Kong, and Jitao Sang. 2024k. o1-coder: An o1 replication for coding. *arXiv preprint arXiv:2412.00154*.
- Zhang, Zhenru, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025d. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*. <https://doi.org/10.18653/v1/2025.findings-acl.547>
- Zhang, Zhihan, Tao Ge, Zhenwen Liang, Wenhao Yu, Dian Yu, Mengzhao Jia, Dong Yu, and Meng Jiang. 2024l. Learn beyond the answer: Training language models with reflection for mathematical reasoning. *arXiv preprint arXiv:2406.12050*. <https://doi.org/10.18653/v1/2024.emnlp-main.817>
- Zhang, Zhuosheng, Aston Zhang, Mu Li, hai zhao, George Karypis, and Alex Smola.

- 2024m. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*.
- Zhang, Zhuosheng, Aston Zhang, Mu Li, and Alex Smola. 2023d. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.
- Zhao, Hao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. 2023. On pitfalls of test-time adaptation. In *International Conference on Machine Learning*, pages 42058–42080.
- Zhao, Jian, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. 2025a. GenPRM: Scaling test-time compute of process reward models via generative reasoning. *arXiv preprint arXiv:2504.00891*. <https://doi.org/10.1609/aaai.v40i41.40797>
- Zhao, Shuai, Xiaohan Wang, Linchao Zhu, and Yi Yang. 2024a. Test-time adaptation with CLIP reward for zero-shot generalization in vision-language models. In *The Twelfth International Conference on Learning Representations*.
- Zhao, Weixiang, Xingyu Sui, Jiahe Guo, Yulin Hu, Yang Deng, Yanyan Zhao, Bing Qin, Wanxiang Che, Tat-Seng Chua, and Ting Liu. 2025b. Trade-offs in large reasoning models: An empirical analysis of deliberative and adaptive reasoning over foundational capabilities. *arXiv preprint arXiv:2503.17979*. <https://doi.org/10.1609/aaai.v40i41.40802>
- Zhao, Yu, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024b. Marco-ol: Towards open reasoning models for open-ended solutions. *arXiv preprint arXiv:2411.14405*.
- Zhao, Yuzhong, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. 2025c. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*.
- Zheng, Chujie, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025a. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.
- Zheng, Chujie, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2024. ProcessBench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*.
- Zheng, Lianmin, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. <https://doi.org/10.52202/075280-2020>
- Zheng, Yuxiang, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. 2025b. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*. <https://doi.org/10.18653/v1/2025.emnlp-main.22>
- Zhou, Denny, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.
- Zhou, Enyu, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, et al. 2024a. RMB: Comprehensively benchmarking reward models in LLM alignment. *arXiv preprint arXiv:2410.09893*.
- Zhou, Qiji, Ruochen Zhou, Zike Hu, Panzhong Lu, Siyang Gao, and Yue Zhang. 2024b. Image-of-thought prompting for visual reasoning refinement in multimodal large language models. *arXiv preprint arXiv:2405.13872*.
- Zhou, Zhi, Tan Yuhao, Zenan Li, Yuan Yao, Lan-Zhe Guo, Xiaoxing Ma, and Yu-Feng Li. 2025. Bridging internal probability and self-consistency for effective and efficient LLM reasoning. *arXiv preprint arXiv:2502.00511*. <https://doi.org/10.52202/085713-2922>
- Zhu, Dingwei, Shihan Dou, Zhiheng Xi, Senjie Jin, Guoqiang Zhang, Jiazheng Zhang, Junjie Ye, Mingxu Chai, Enyu Zhou, Ming Zhang, et al. 2025. VRPO: Rethinking value modeling for robust RL training under noisy supervision. *arXiv preprint arXiv:2508.03058*. <https://doi.org/10.18653/v1/2026.ac1-long.1103>
- Zhu, Lianghui, Xinggang Wang, and Xinlong Wang. 2023. JudgeLM: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.
- Zhu, Wenhao, Shujian Huang, Yunzhe Lv, Xin Zheng, and Jiajun Chen. 2023. What

- knowledge is needed? Towards explainable memory for kNN-MT domain adaptation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2824–2836. <https://doi.org/10.18653/v1/2023.findings-acl.177>
- Zhu, Yuhao, Guozhen Zhang, Chen Xu, Haocheng Shen, Xiaoxin Chen, Gangshan Wu, and Limin Wang. 2024. Efficient test-time prompt tuning for vision-language models. *arXiv preprint arXiv:2408.05775*.
- Zhuang, Fuzhen, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76. <https://doi.org/10.1109/JPROC.2020.3004555>
- Zhuo, Terry Yue, Vu Minh Chien, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widayarsi, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. 2025. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *The Thirteenth International Conference on Learning Representations*.