Test-time Compute: from System-1 Thinking to System-2 Thinking

Anonymous ACL submission

Abstract

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 a few possible future directions.

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 (Radford et al., 2018, 2019; 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, the o1 model (OpenAI, 2024), equipped with System-2 thinking, has gained attention for its 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 works have explored test-time adaptation (TTA) to improve model robustness by updating parameters (Wang et al., 2021; Ye et al., 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, unable to exhibit explicit, logical thinking process 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 language models to perform explicit System-2 thinking (Hagendorff et al., 2023). However, vanilla CoT is limited by error accumulation and linear thinking pattern (Stechly et al., 2024; Sprague et al., 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 employ 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

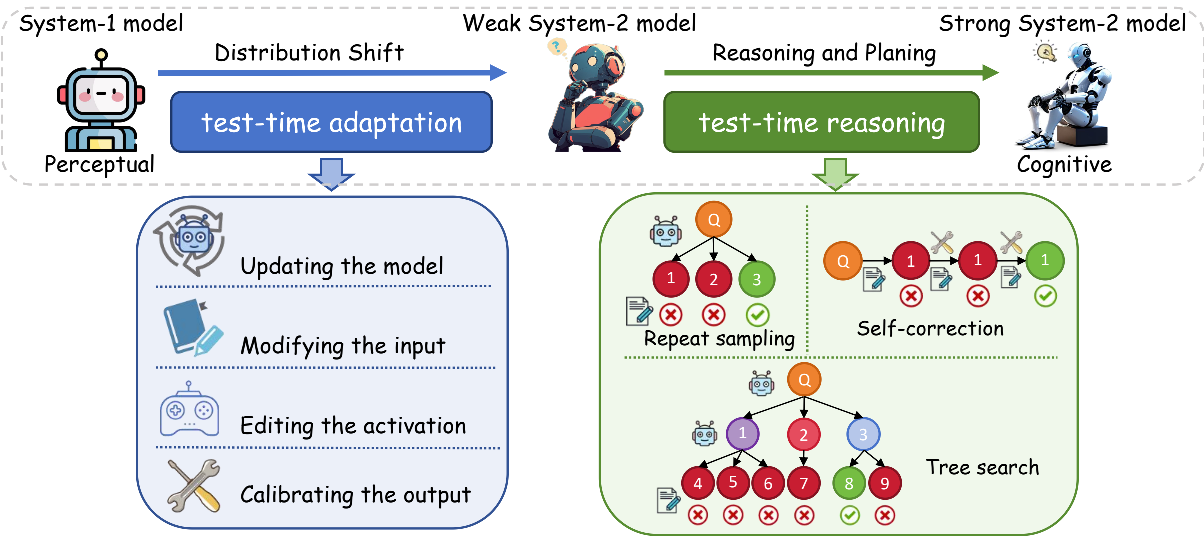

Figure 1: Illustration of test-time compute in the System-1 and System-2 model.

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 the best of our knowledge, this paper 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. In Section 2, we present the background of System-1 and System-2 thinking. Section 3 and Section 4 detail the test-time compute methods for the System-1 and System-2 models. Then, we discuss future directions in Section 5 and Appendix C. Additionally, we review benchmarks and open-source frameworks in Appendix D.

2 Background

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 AI, 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 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. Nowadays, the strong generation and reasoning capabilities of LLMs make it possible to build System-2 models. Wei et al. (2022) propose the 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; Tang et al., 2024b; 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. As a result, CoT-enabled LLMs are still at the weak System-2 thinking stage.

3 Test-time Adaptation for System-1 Thinking

3.1 Updating the Parameters

Model updating utilizes test sample information to further finetune model parameters during the inference stage, enabling the model to adapt to the test distribution. In the inference stage, the ground-truth of test samples is unavailable. Thus many works attempt to design unsupervised or self-

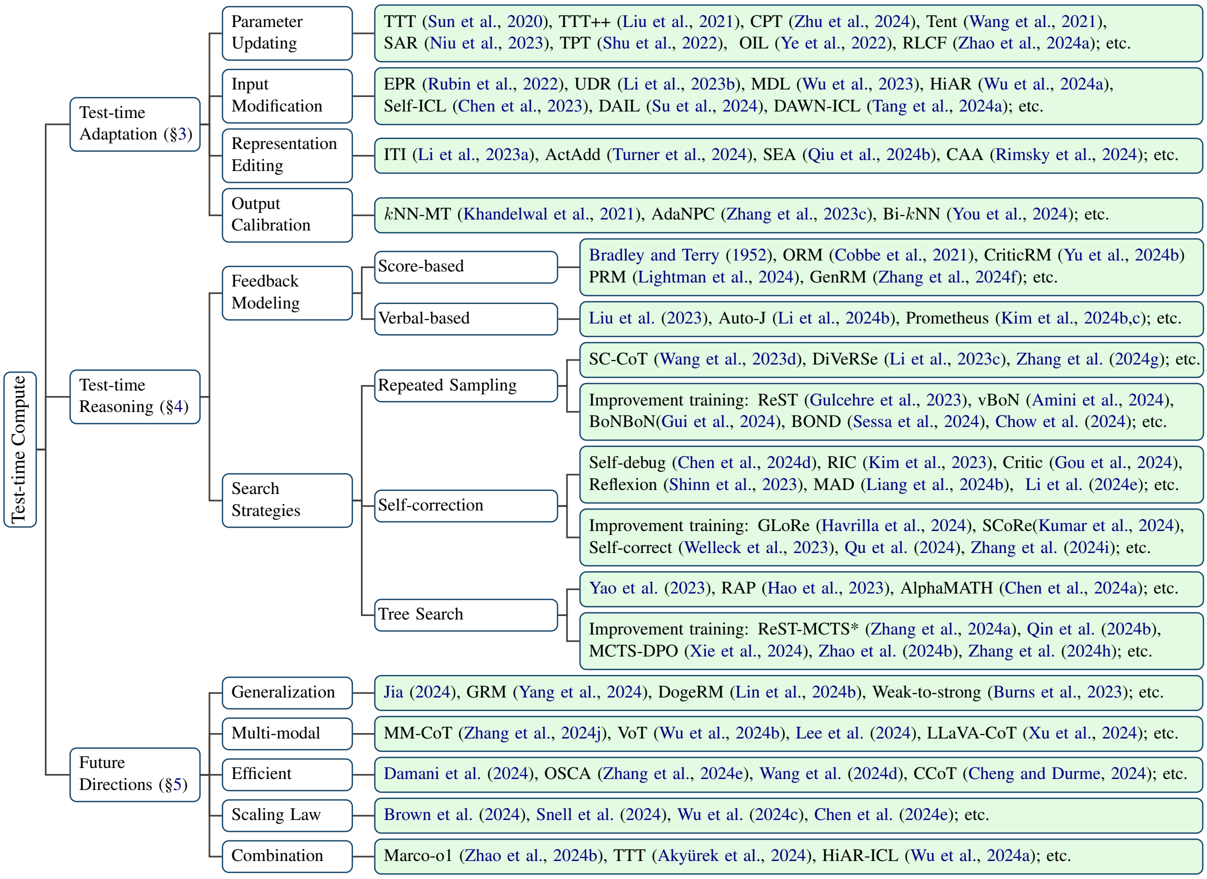

Figure 2: Taxonomy of test-time compute methods and future directions.

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 autoencoding (Gandelsman et al., 2022) and contrastive learning (Liu et al., 2021; Chen et al., 2022; Zhu et al., 2024).

In contrast, FTTA 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. MEMO (Zhang et al., 2022a) augments the data for a single test sample and then minimizes its marginal entropy, which is more stable 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). Some works propose new regularization terms for minimizing entropy to avoid model collapse, including Kullback-Leibler divergence (Su et al., 2023a), moment matching (Hassan et al., 2023) and entropy matching (Bar et al., 2024). For specific tasks, human feedback (Gao et al., 2022; Li et al., 2022b) or external model rewards (Zhan et al., 2023) can also serve as high-quality learning signals. In cross-modal tasks, RLCF (Zhao et al., 2024a) demonstrates CLIP scores are effective TTA signals. In language modeling, training with semantically relevant contexts at test time can reduce perplexity (Hardt and Sun, 2024; Wang et al., 2024h). Hübotter et al. (2025) theoretically shows that it reduces the uncertainty of test samples and proposes a more effective active learning selection strategy. In practical applications, the efficiency and stability of parameter updates are also important research directions, which we review in detail in the Appendix A.

3.2 Modifying the Input

Input-modification methods, which leverage in-context learning (ICL) without updating parameters, have become mainstream for LLMs because of their efficiency and stability. ICL improves performance by adding demonstrations before the test sample but is highly sensitive to demonstration selection and order. Therefore, the key to input-modification TTA is choosing the appropriate demonstrations and arranging them optimally.

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 are used to retrieve demonstrations semantically closest to the test sample (Qin et al., 2024a; Luo et al., 2023a; Rubin et al., 2022; Li et al., 2023b). Then, 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). 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 et al. (2022b) consider the sequential dependency among demonstrations, and model it as a sequential decision problem and optimize demonstration selection and ordering through reinforcement learning.

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., 2024a) further models the traversal order of test samples as a planning task and optimizes it by the Monte Carlo tree search (MCTS).

3.3 Editing the Representation

For generative LLMs, some works 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; Scalena 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.

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). 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 kNN 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 kNN machine translation (kNN-MT). kNN-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. kNN-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., 2023b; You et al., 2024) or applying its methods to other NLP tasks (Wang et al., 2022b; Bhardwaj et al., 2023).

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 manual prior knowledge.*

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.

4.1 Feedback Modeling

Score-based Feedback Score-based feedback, also known as the verifier, aims to score generated results, evaluating their alignment with ground truth or human cognitive processes. Its training process is typically similar to the reward model in RLHF (Gao et al., 2023a), 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 mainly divided into two categories: outcome-based (ORMs) and process-based verifiers (PRMs). ORMs (Cobbe et al., 2021) use the correctness of the final CoT result as training feedback, while PRMs (Uesato et al., 2022; Lightman et al., 2024; Zhang et al., 2024d) are trained based on feedback from each reasoning step. PRM not only evaluates intermediate reasoning steps but also evaluates the entire reasoning process more accurately than ORM. However, PRM requires more human effort to annotate feedback for the intermediate steps. Math-Shepherd (Wang et al., 2024g) and OmegaPRM (Luo et al., 2024) utilize MCTS algorithm to collect high-quality process supervision data automatically. Setlur et al. (2024) argue that PRM 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 Monte Carlo simulations. 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 ORM or directly train PRM on outcome labels, respectively. Score-based feedback modeling overlooks the generative capabilities of LLMs, making it difficult to detect fine-grained errors. Thus, recent works propose generative score-based verifiers (Ankner et al., 2024; Ye et al., 2024). GenRM (Zhang et al., 2024f) 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'. Critic-RM (Yu et al., 2024b) jointly trains the critique model and the verifier. During inference, the verifier scores according to answers and verbal-based feedback generated by the critique model.

Verbal-based Feedback Although the verifier can accurately 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. Verbal-based feedback, also referred to critic, fully leverages the LLM's instruction-following ability. By designing specific instructions, it can perform pairwise comparisons, evaluate answers from multiple dimensions, and even provide suggestions for revision in natural language. Powerful closed-source LLMs 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 et al., 2023b; 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., 2024f; Stureborg et al., 2024). LLM-as-a-Judge (Zheng et al., 2023) carefully designs system instructions to mitigate the interference of biases.

To obtain cheaper verbal-based feedback, open-source LLMs can also serve as competitive alternatives through supervised fine-tuning (SFT) (Wang et al., 2024i; Zhu et al., 2023a; Liang et al., 2024c; 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,c) 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.

4.2 Search Strategies

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. 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* (Wang et al., 2023d; 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. However, the majority does not always hold the truth, as they may make similar mistakes. Therefore, some studies perform validation and filtering before voting. For example, the PROVE framework (Toh et al., 2024) converts CoT into executable programs, filtering out samples if the program's results are inconsistent with the reasoning chain's outcomes.

Best-of-N sampling uses a verifier to score each generated result and selects the one with the highest score as the final answer (Stiennon et al., 2020; Cobbe et al., 2021; Nakano et al., 2022). 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 weight. In addition, some works aim to improve the efficiency of BoN. Inspired by speculative decoding, Zhang et al. (2024g); Qiu et al. (2024a); Sun et al. (2024) and Manvi et al. (2024) evaluate each reasoning step by an efficient verifier. They 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.

Improvement Training Repeated sampling, especially the BoN strategy, has proven to be a simple yet effective method in many studies, even can surpassing models fine-tuned with RLHF (Gao et al., 2023a). However, it comes at the cost of inference times that are 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-tune the policy model by offline reinforcement learning. In each iteration, ReST samples new training data. vBoN (Amini et al., 2024), BoNBoN (Gui et al., 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.

4.2.2 Self-correction

Self-correction is a sequential test-time compute method that enables LLMs to iteratively revise and refine generated results 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, *tool checking* provides accurate feedback (Gou et al., 2024; Chen et al., 2024d; Gao et al., 2023b). For example, Yasunaga and Liang (2020) propose to obtain feedback from compilers in code repair and generation tasks. In embodied tasks, the environment can provide precise feedback on the action trajectories of LLM-based agents (Wang et al., 2024a).

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., 2023; Xiong et al., 2023; Liang et al., 2024b; Chen et al., 2024b; Wang et al., 2024e) 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., 2023; Cheng et al., 2024). 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 et al., 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., 2024c). 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.

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 et al., 2023) iteratively prompt LLMs to self-correct their responses in tasks such as arithmetic reasoning. IoE (Li et al., 2024e) observes that LLMs may over-criticize themselves during self-critique, leading to performance degradation, and designs prompt to guide LLMs in assessing confidence. However, the effectiveness of self-correction has remained controversial, and we discuss it in Appendix B.

Improvement Training Most of the aforementioned self-correction methods demonstrate significant performance improvements on advanced LLMs. However, for medium-scale models with weaker reasoning capabilities, we need to further fine-tune them to unlock their self-correction capabilities. SFT optimizes the model using high-quality multi-turn correction data, either manually annotated (Saunders et al., 2022) or sampled from stronger LLMs (An et al., 2023; Paul et al., 2024; Qu et al., 2024; Gao et al., 2024c; Zhang et al., 2024i; 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. 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. SCoRe (Kumar et al., 2024) proposes using the multi-turn RL method to improve self-critique and self-correction capability.

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. 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 does not rely on specific heuristic information but 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 represented by MCTS is widely used in reasoning tasks (Hao et al., 2023; Zhang et al., 2024b; Bi et al., 2024). MCTS gradually optimizes search results through four steps: selection, expansion, simulation, and backpropagation, thereby approaching the optimal solution. Long (2023) trains an LLM controller by reinforcement learning to guide the LLM reasoner's search path.

Value function The value function evaluates the value of each action and guides the tree to expand towards branches with higher values in MCTS. 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 handcrafted value function with a learned LLM value function, automatically generating reasoning process and step-level evaluation signals in MCTS. 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 inspired by contrast decoding, which utilizes multiple external reward models as value functions.

Improvement Training Tree search can guide LLMs to generate long reasoning processes, and these data help train LLMs with stronger reasoning abilities. 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. Recently, many o1-like models (Qin et al., 2024b; Zhao et al., 2024b; Zhang et al., 2024h) also confirm the necessity of using tree search to construct high-quality long reasoning chain data for training.

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, but may not perform well with poor feedback or weak reasoning capability. Tree search optimizes complex planning tasks globally but involves complex implementation.*

5 Future Directions

Test-time compute is a promising path toward System-2 models, with several directions for future exploration. First, the **generalization** capabilities of System-2 models remain challenging, particularly in cross-domain and weak-to-strong generalization. Second, human cognition relies not only on language, thus System-2 models should integrate **multimodal** collaboration. Another critical issue is the high computational cost of test-time compute presents a significant challenge, requiring **a balance between efficiency and performance**. Additionally, while we have a qualitative understanding of the scaling effects of test-time compute, the quantitative **scaling law** is still lacking. Finally, **combining multiple strategies** is worth further investigation, as it has the potential to lead to better performance. We provide a more detailed explanation in Appendix C.

6 Conclusion

In this paper, 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 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 in this area.

Limitations

Test-time compute, especially the strategies in System-2, is evolving rapidly. While we have made efforts to provide a comprehensive survey of existing research, it is challenging to cover all the latest developments. This review includes papers up to December 2024, with more recent advancements to be updated in future versions. TTA has seen many successful applications and task-specific strategies in CV tasks. Since the primary audience of our paper is researchers in NLP, we do not systematically present these works, and interested readers can refer to Liang et al. (2024a) for details.

References

Ekin Akyürek, Mehul Damani, Linlu Qiu, Han Guo, Yoon Kim, and Jacob Andreas. 2024. The surprising effectiveness of test-time training for abstract reasoning. *Preprint*, arXiv:2411.07279.

Alfonso Amayuelas, Xianjun Yang, Antonis Antoniades, Wenyue Hua, Liangming Pan, and William Wang. 2024. Multiagent collaboration attack: Investigating adversarial attacks in large language model collaborations via debate. *Preprint*, arXiv:2406.14711.

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Variational best-of-n alignment. *Preprint*, arXiv:2407.06057.

Shengnan An, 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*.

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D. Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *Preprint*, arXiv:2408.11791.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Preprint*, arXiv:2406.11717.

Daman Arora, 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, Singapore. Association for Computational Linguistics.

Mikel Artetxe, 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, Online. Association for Computational Linguistics.

Jacob Austin, 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*.

Zhangir Azerbayev, 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*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Yarin Bar, Shalev Shaer, and Yaniv Romano. 2024. Protected test-time adaptation via online entropy matching: A betting approach. *Preprint*, arXiv:2408.07511.

Alexander Bartler, Andre Bühler, 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. PMLR.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, André F. T. Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2024. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *Preprint*, arXiv:2406.18403.

Rishabh Bhardwaj, 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, Singapore. Association for Computational Linguistics.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2024. Forest-of-thought: Scaling test-time compute for enhancing llm reasoning. *Preprint*, arXiv:2412.09078.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Preprint*, arXiv:2407.21787.

Tom Brown, 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.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *Preprint*, arXiv:2312.09390.

Yuanpu Cao, 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. *Preprint*, arXiv:2406.00045.

Federico Cassano, 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*.

Chi-Min Chan, 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*.

Dian Chen, 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.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024a. Alphamath almost zero: Process supervision without process. *Preprint*, arXiv:2405.03553.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024b. 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, Bangkok, Thailand. Association for Computational Linguistics.

Lichang Chen, Chen Zhu, Jiuhai Chen, Davit Soselia, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024c. ODIN: Disentangled reward mitigates hacking in RLHF. In *Forty-first International Conference on Machine Learning*.

Mark Chen, 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*.

Wei-Lin Chen, 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, Singapore. Association for Computational Linguistics.

Xingyu Chen, 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. 2025. Do not think that much for 2+3=? on the overthinking of o1-like llms. *Preprint*, arXiv:2412.21187.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024d. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*.

Yanxi Chen, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. 2024e. A simple and provable scaling law for the test-time compute of large language models. *Preprint*, arXiv:2411.19477.

Zhaorun Chen, 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. 2024f. Mj-bench: Is your multimodal reward model really a good judge for text-to-image generation? *Preprint*, arXiv:2407.04842.

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *Preprint*, arXiv:2412.13171.

Xiaoxue Cheng, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2024. ChainLM: Empowering large language models with improved chain-of-thought prompting. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2969–2983, Torino, Italia. ELRA and ICCL.

Cheng-Han Chiang 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, Toronto, Canada. Association for Computational Linguistics.

Yinlam Chow, 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. *Preprint*, arXiv:2412.15287.

Karl Cobbe, 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. *Preprint*, arXiv:2110.14168.

Damai Dai, 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, Toronto, Canada. Association for Computational Linguistics.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. 2024. Learning how hard to think: Input-adaptive allocation of lm computation. *Preprint*, arXiv:2410.04707.

Sumanth Dathathri, 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*.

Yangruibo Ding, 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*.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024a. RLHF workflow: From reward modeling to online RLHF. *Transactions on Machine Learning Research*.

Qingxiu Dong, 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 *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *Preprint*, arXiv:2305.14325.

Ahmed Elgohary, Christopher Meek, Matthew Richardson, Adam Fourney, Gonzalo Ramos, and Ahmed Hassan Awadallah. 2021. NL-EDIT: Correcting semantic parse errors through natural language interaction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5599–5610, Online. Association for Computational Linguistics.

Chun-Mei Feng, 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.

Guhao Feng, 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.

Xidong Feng, 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. *Preprint*, arXiv:2309.17179.

Adam Fisch, 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, Hong Kong, China. Association for Computational Linguistics.

Qichen Fu, 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*.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A Efros. 2022. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024a. Omni-math: A universal olympiad level mathematic benchmark for large language models. *Preprint*, arXiv:2410.07985.

Ge Gao, 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, Dublin, Ireland. Association for Computational Linguistics.

Jun Gao, Yongqi Li, Ziqiang Cao, and Wenjie Li. 2024b. Interleaved-modal chain-of-thought. *Preprint*, arXiv:2411.19488.

Kuofeng Gao, 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. *Preprint*, arXiv:2410.10735.

Leo Gao, John Schulman, and Jacob Hilton. 2023a. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023b. 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, Toronto, Canada. Association for Computational Linguistics.

Zitian Gao, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. 2024d. Interpretable contrastive monte carlo tree search reasoning. *Preprint*, arXiv:2410.01707.

Mor Geva, 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.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *Preprint*, arXiv:2411.04872.

Linyuan Gong, 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*.

Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. 2024b. Sotta: Robust test-time adaptation on noisy data streams. *Advances in Neural Information Processing Systems*, 36.

Zhibin Gou, 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*.

Alex Gu, 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*.

Lin Gui, 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*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. Reinforced self-training (rest) for language modeling. *Preprint*, arXiv:2308.08998.

Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. 2025. Can we generate images with cot? let's verify and reinforce image generation step by step. *Preprint*, arXiv:2501.13926.

Thilo Hagendorff, 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.

Shibo Hao, 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, Singapore. Association for Computational Linguistics.

Moritz Hardt and Yu Sun. 2024. Test-time training on nearest neighbors for large language models. In *The Twelfth International Conference on Learning Representations*.

Jameel Hassan, 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.

Alex Havrilla, Sharath Raparthy, 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. *Preprint*, arXiv:2402.10963.

Chaoqun He, 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, Bangkok, Thailand. Association for Computational Linguistics.

Dan Hendrycks, 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.

Dan Hendrycks, 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)*.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-STar: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*.

Neil Houlsby, 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. PMLR.

Edward J Hu, 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*.

Jian Hu, Xibin Wu, Zilin Zhu, Xianyu, Weixun Wang, Dehao Zhang, and Yu Cao. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *Preprint*, arXiv:2405.11143.

Jie Huang, 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*.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, Yiwei Qin, Yan Ma, Jiadi Su, Yixiu Liu, Yuxiang Zheng, Shaoting Zhang, Dahua Lin, Yu Qiao, and Pengfei Liu. 2024b. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai. *Preprint*, arXiv:2406.12753.

Zhen Huang, 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*.

Jonas Hübotter, 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*.

Raza Imam, Hanan Gani, Muhammad Huzaifa, and Karthik Nandakumar. 2024. Test-time low rank adaptation via confidence maximization for zero-shot generalization of vision-language models. *Preprint*, arXiv:2407.15913.

Chen Jia. 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. Springer.

Daniel Kahneman. 2011. Thinking, fast and slow. *Farrar, Straus and Giroux*.

Ryo Kamoi, 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. *Preprint*, arXiv:2406.01297.

Akbir Khan, 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*.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *Preprint*, arXiv:2303.17491.

Hyuhng Joon Kim, 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. *Preprint*, arXiv:2206.08082.

Kyungha Kim, 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. *Preprint*, arXiv:2402.07401.

Seungone Kim, 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*.

Seungone Kim, 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, Miami, Florida, USA. Association for Computational Linguistics.

Sunghwan Kim, 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*.

Takeshi Kojima, 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.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2024. Training language models to self-correct via reinforcement learning. *Preprint*, arXiv:2409.12917.

Philippe Laban, Lidiya Murakhovs'ka, Caiming Xiong, and Chien-Sheng Wu. 2024. Are you sure? challenging llms leads to performance drops in the flipflop experiment. *Preprint*, arXiv:2311.08596.

Yuhang Lai, 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. PMLR.

Nathan Lambert, 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. *Preprint*, arXiv:2403.13787.

Tian Lan, 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*.

Yann LeCun. 2022. A path towards autonomous machine intelligence.

Junlin Lee, 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, Bangkok, Thailand. Association for Computational Linguistics.

Brian Lester, 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Noam Levi. 2024. A simple model of inference scaling laws. *arXiv preprint arXiv:2410.16377*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Will LeVine, 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*.

Patrick Lewis, 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, Online. Association for Computational Linguistics.

Jia Li, 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*.

Junlong Li, 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*.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024c. More agents is all you need. *Preprint*, arXiv:2402.05120.

Kenneth Li, 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*.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, Lingpeng Kong, and Qi Liu. 2024d. Vlrewardbench: A challenging benchmark for vision-language generative reward models. *Preprint*, arXiv:2411.17451.

Loka Li, Zhenhao Chen, Guangyi Chen, Yixuan Zhang, Yusheng Su, Eric Xing, and Kun Zhang. 2024e. Confidence matters: Revisiting intrinsic self-correction capabilities of large language models. *Preprint*, arXiv:2402.12563.

Xiang Li, 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, Torino, Italia. ELRA and ICCL.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang 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, Toronto, Canada. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

Yifei Li, 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, Toronto, Canada. Association for Computational Linguistics.

Yujia Li, 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.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024g. Improving multi-agent debate with sparse communication topology. *Preprint*, arXiv:2406.11776.

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

Zhiyuan Li, 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*.

Zichao Li, 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, Dublin, Ireland. Association for Computational Linguistics.

Jian Liang, Ran He, and Tieniu Tan. 2024a. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, pages 1–34.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024b. Encouraging divergent thinking in large language models through multi-agent debate. *Preprint*, arXiv:2305.19118.

Xiaobo Liang, Haoke Zhang, Helan hu, Juntao Li, Jun Xu, and Min Zhang. 2024c. Fennec: Fine-grained language model evaluation and correction extended through branching and bridging. *Preprint*, arXiv:2405.12163.

Hunter Lightman, 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*.

Lei Lin, 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, Bangkok, Thailand. Association for Computational Linguistics.

Tzu-Han Lin, 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, Miami, Florida, USA. Association for Computational Linguistics.

Zicheng Lin, 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, Bangkok, Thailand. Association for Computational Linguistics.

Jiachang Liu, 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, Dublin, Ireland and Online. Association for Computational Linguistics.

Jiawei Liu, 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.

Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024b. In-context vectors: Making in context learning more effective and controllable through latent space steering. *Preprint*, arXiv:2311.06668.

Tongxuan Liu, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. 2024c. Groupdebate: Enhancing the efficiency of multi-agent debate using group discussion. *Preprint*, arXiv:2409.14051.

Yang Liu, 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, Singapore. Association for Computational Linguistics.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2024d. Rm-bench: Benchmarking reward models of language models with subtlety and style. *arXiv preprint arXiv:2410.16184*.

Yuejiang Liu, 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.

Jieyi Long. 2023. Large language model guided tree-of-thought. *Preprint*, arXiv:2305.08291.

Dawn Lu and Nina Rimsky. 2024. Investigating bias representations in llama 2 chat via activation steering. *Preprint*, arXiv:2402.00402.

Jianqiao Lu, 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*.

Yao Lu, 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, Dublin, Ireland. Association for Computational Linguistics.

Liangchen Luo, 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. *Preprint*, arXiv:2406.06592.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023a. Dr.icl: Demonstration-retrieved in-context learning. *Preprint*, arXiv:2305.14128.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023b. Chatgpt as a factual inconsistency evaluator for text summarization. *Preprint*, arXiv:2303.15621.

Xinxi Lyu, 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, Toronto, Canada. Association for Computational Linguistics.

Xiaosong Ma, Jie Zhang, Song Guo, and Wenchao Xu. 2023. Swapprompt: Test-time prompt adaptation for vision-language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Rohin Manvi, Anikait Singh, and Stefano Ermon. 2024. Adaptive inference-time compute: Llms can predict if they can do better, even mid-generation. *Preprint*, arXiv:2410.02725.

Debjyoti Mondal, 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.

Dilxat Muhtar, Yelong Shen, Yaming Yang, Xiaodong Liu, Yadong Lu, Jianfeng Liu, Yuefeng Zhan, Hao Sun, Weiwei Deng, Feng Sun, Xueliang Zhang, Jianfeng Gao, Weizhu Chen, and Qi Zhang. 2024. Streamadapter: Efficient test time adaptation from contextual streams. *Preprint*, arXiv:2411.09289.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

Deepak Nathani, 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, Singapore. Association for Computational Linguistics.

Shuaicheng Niu, 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. PMLR.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. 2023. Towards stable test-time adaptation in dynamic wild world. In *The Eleventh International Conference on Learning Representations*.

Theo X. Olausson, 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*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Debjit Paul, 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, St. Julian's, Malta. Association for Computational Linguistics.

Ori Press, Ravid Shwartz-Ziv, Yann LeCun, and Matthias Bethge. 2024. The entropy enigma: Success and failure of entropy minimization. *Preprint*, arXiv:2405.05012.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *Preprint*, arXiv:2408.06195.

Chengwei Qin, 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, Miami, Florida, USA. Association for Computational Linguistics.

Yiwei Qin, 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. *Preprint*, arXiv:2410.18982.

Jiahao Qiu, 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. *Preprint*, arXiv:2410.16033.

Yifu Qiu, 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*.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. *Preprint*, arXiv:2407.18219.

Qwen. 2024. Qvq: To see the world with wisdom. *Qwen Team, blog*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *Open AI, blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *Open AI, blog*.

Nina Rimsky, 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, Bangkok, Thailand. Association for Computational Linguistics.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code llama: Open foundation models for code. *Preprint*, arXiv:2308.12950.

Ohad Rubin, 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, Seattle, United States. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *Preprint*, arXiv:2206.05802.

Daniel Scalena, 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, Miami, Florida, US. Association for Computational Linguistics.

Alexander Scarlatos and Andrew Lan. 2024. Reticl: Sequential retrieval of in-context examples with reinforcement learning. *Preprint*, arXiv:2305.14502.

Steffen Schneider, 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.

Holger Schwenk, 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, Online. Association for Computational Linguistics.

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shariari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos, Amélie Héliou, Aliaksei Severyn, Matt Hoffman, Nikola Momchev, and Olivier Bachem. 2024. Bond: Aligning llms with best-of-n distillation. *Preprint*, arXiv:2407.14622.

Amrith Setlur, 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. *Preprint*, arXiv:2410.08146.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Manli Shu, 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.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025. Prmbench: A fine-grained and challenging benchmark for process-level reward models. *arXiv preprint arXiv:2501.03124*.

Zayne Sprague, 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. *Preprint*, arXiv:2409.12183.

Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems. *Preprint*, arXiv:2310.12397.

Kaya Stechly, 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*.

Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. 2024. Teaching models to balance resisting and accepting persuasion. *Preprint*, arXiv:2410.14596.

Nisan Stiennon, 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. 2024. Improving instruction-following in language models through activation steering. *Preprint*, arXiv:2410.12877.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *Preprint*, arXiv:2405.01724.

Hongjin Su, 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. *Preprint*, arXiv:2209.01975.

Yi Su, Yixin Ji, Juntao Li, Hai Ye, and Min Zhang. 2023a. 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, Singapore. Association for Computational Linguistics.

Yi Su, Yixin Ji, Juntao Li, Hai Ye, and Min Zhang. 2023b. Test-time adaptation with perturbation consistency learning. *Preprint*, arXiv:2304.12764.

Yi Su, 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, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Hanshi Sun, 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*.

Yu Sun, 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. PMLR.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Niket Tandon, Aman Madaan, Peter Clark, Keisuke Sakaguchi, and Yiming Yang. 2021. Interscript: A dataset for interactive learning of scripts through error feedback. *Preprint*, arXiv:2112.07867.

Niket Tandon, 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, Seattle, United States. Association for Computational Linguistics.

Xinyu Tang, Xiaolei Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2024a. Dawn-icl: Strategic planning of problem-solving trajectories for zero-shot in-context learning. *arXiv preprint arXiv:2410.20215*.

Zecheng Tang, Keyan Zhou, Juntao Li, Yuyang Ding, Pinzheng Wang, Yan Bowen, Renjie Hua, and Min Zhang. 2024b. CMD: a framework for context-aware model self-detoxification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1930–1949, Miami, Florida, USA. Association for Computational Linguistics.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Vernon Y. H. Toh, Deepanway Ghosal, and Soujanya Poria. 2024. Not all votes count! programs as verifiers improve self-consistency of language models for math reasoning. *Preprint*, arXiv:2410.12608.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *Preprint*, arXiv:2308.10248.

Gladys Tyen, 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, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023a. Can large language models really improve by self-critiquing their own plans? *Preprint*, arXiv:2310.08118.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023b. 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*.

Pablo Villalobos, 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. *Preprint*, arXiv:2211.04325.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2021. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*.

Dexin Wang, 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, Dublin, Ireland. Association for Computational Linguistics.

Hanlin Wang, Chak Tou Leong, Jian Wang, and Wenjie Li. 2024a. E^{2}CL: Exploration-based error correction learning for embodied agents. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7626–7639, Miami, Florida, USA. Association for Computational Linguistics.

Haohan Wang, 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.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024b. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*.

Jiaan Wang, 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, Singapore. Association for Computational Linguistics.

Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M Ni, et al. 2024c. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671*.

Junlin Wang, 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, Miami, Florida, USA. Association for Computational Linguistics.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024e. Mixture-of-agents enhances large language model capabilities. *Preprint*, arXiv:2406.04692.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024f. 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, Bangkok, Thailand. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024g. 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, Bangkok, Thailand. Association for Computational Linguistics.

Shuhe Wang, Xiaoya Li, Yuxian Meng, Tianwei Zhang, Rongbin Ouyang, Jiwei Li, and Guoyin Wang. 2022b. $k$nn-ner: Named entity recognition with nearest neighbor search. *Preprint*, arXiv:2203.17103.

Tianlu Wang, 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. *Preprint*, arXiv:2308.04592.

Xinyi Wang, 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*.

Xuezhi Wang, 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*.

Yan Wang, Dongyang Ma, and Deng Cai. 2024h. With greater text comes greater necessity: Inference-time training helps long text generation. In *First Conference on Language Modeling*.

Yidong Wang, 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. 2024i. PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization. In *The Twelfth International Conference on Learning Representations*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024j. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jason Wei, 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*.

Sean Welleck, 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*.

Jason Weston and Sainbayar Sukhbaatar. 2023. System 2 attention (is something you might need too). *Preprint*, arXiv:2311.11829.

Mitchell Wortsman, 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. PMLR.

Jinyang Wu, 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. *Preprint*, arXiv:2411.18478.

Wenshan Wu, 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*.

Yangzhen Wu, 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. *Preprint*, arXiv:2408.00724.

Zhiyong Wu, 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, Toronto, Canada. Association for Computational Linguistics.

Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Do, Wenyu Zhan, Xiao Wang, Rui Zheng, Tao Ji, Xiaowei Shi, Yitao Zhai, Rongxiang Weng, Jingang Wang, Xunliang Cai, Tao Gui, Zuxuan Wu, Qi Zhang, Xipeng Qiu, Xuanjing Huang, and Yu-Gang Jiang. 2024. Enhancing llm reasoning via critique models with test-time and training-time supervision. *Preprint*, arXiv:2411.16579.

Heming Xia, 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*.

Enze Xie, 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. 2025. Sana 1.5: Efficient scaling of training-time and inference-time compute in linear diffusion transformer. *Preprint*, arXiv:2501.18427.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. *Preprint*, arXiv:2405.00451.

Yuxi Xie, 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*.

Kai Xiong, 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, Singapore. Association for Computational Linguistics.

Tianyi Xiong, 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*.

Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2024. Llava-cot: Let vision language models reason step-by-step. *Preprint*, arXiv:2411.10440.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. Regularizing hidden states enables learning generalizable reward model for LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Michihiro Yasunaga and Percy Liang. 2020. Graph-based, self-supervised program repair from diagnostic feedback. In *International Conference on Machine Learning (ICML)*.

Hai Ye, 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hai Ye 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, Miami, Florida, USA. Association for Computational Linguistics.

Hai Ye, 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, Toronto, Canada. Association for Computational Linguistics.

Zihuiwen Ye, Fraser Greenlee-Scott, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gallé. 2024. Improving reward models with synthetic critiques. *Preprint*, arXiv:2405.20850.

WangJie You, 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, Bangkok, Thailand. Association for Computational Linguistics.

Hao Yu, 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.

Yongcan Yu, Lijun Sheng, Ran He, and Jian Liang. 2023. Benchmarking test-time adaptation against distribution shifts in image classification. *arXiv preprint arXiv:2307.03133*.

Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, Melanie Kambadur, Dhruv Mahajan, and Rui Hou. 2024b. Self-generated critiques boost reward modeling for language models. *Preprint*, arXiv:2411.16646.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. 2024a. Advancing llm reasoning generalists with preference trees. *Preprint*, arXiv:2404.02078.

Lifan Yuan, 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*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024c. Self-rewarding language models. *Preprint*, arXiv:2401.10020.

Parvez Zamil and Gollam Rabby. 2024. Aime problems 1983 to 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Zhongshen Zeng, 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*.

Runzhe Zhan, 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, Toronto, Canada. Association for Computational Linguistics.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts*: Llm self-training via process reward guided tree search. *Preprint*, arXiv:2406.03816.

Di Zhang, 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. *Preprint*, arXiv:2406.07394.

Fengji Zhang, 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, Singapore. Association for Computational Linguistics.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024c. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. *arXiv preprint arXiv:2410.02506*.

Hanning Zhang, Pengcheng Wang, Shizhe Diao, Yong Lin, Rui Pan, Hanze Dong, Dylan Zhang, Pavlo Molchanov, and Tong Zhang. 2024d. Entropy-regularized process reward model. *arXiv preprint arXiv:2412.11006*.

Kexun Zhang, Shang Zhou, Danqing Wang, William Yang Wang, and Lei Li. 2024e. Scaling llm inference with optimized sample compute allocation. *Preprint*, arXiv:2410.22480.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024f. Generative verifiers: Reward modeling as next-token prediction. *Preprint*, arXiv:2408.15240.

Marvin Zhang, Sergey Levine, and Chelsea Finn. 2022a. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642.

Ruiqi Zhang, Momin Haider, Ming Yin, Jiahao Qiu, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024g. Accelerating best-of-n via speculative rejection. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ICML 2024)*.

Xiaotian Zhang, 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*.

Yifan Zhang, 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. PMLR.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022b. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuxiang Zhang, Shangxi Wu, Yuqi Yang, Jiangming Shu, Jinlin Xiao, Chao Kong, and Jitao Sang. 2024h. o1-coder: an o1 replication for coding. *Preprint*, arXiv:2412.00154.

Zhihan Zhang, Tao Ge, Zhenwen Liang, Wenhao Yu, Dian Yu, Mengzhao Jia, Dong Yu, and Meng Jiang. 2024i. Learn beyond the answer: Training language models with reflection for mathematical reasoning. *Preprint*, arXiv:2406.12050.

Zhuosheng Zhang, Aston Zhang, Mu Li, hai zhao, George Karypis, and Alex Smola. 2024j. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*.

Zhuosheng Zhang, 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*.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. 2023. On pitfalls of test-time adaptation. In *International Conference on Machine Learning*, pages 42058–42080. PMLR.

Shuai Zhao, 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*.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024b. Marco-o1: Towards open reasoning models for open-ended solutions. *Preprint*, arXiv:2411.14405.

Chujie Zheng, 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*.

Lianmin Zheng, 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*.

Denny Zhou, 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*.

Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Rmb: Comprehensively benchmarking reward models in llm alignment. *Preprint*, arXiv:2410.09893.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023a. Judgelm: Fine-tuned large language models are scalable judges. *Preprint*, arXiv:2310.17631.

Wenhao Zhu, Shujian Huang, Yunzhe Lv, Xin Zheng, and Jiajun Chen. 2023b. 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, Toronto, Canada. Association for Computational Linguistics.

Yuhan Zhu, Guozhen Zhang, Chen Xu, Haocheng Shen, Xiaoxin Chen, Gangshan Wu, and Limin Wang. 2024. Efficient test-time prompt tuning for vision-language models. *Preprint*, arXiv:2408.05775.

Fuzhen Zhuang, 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.

Terry Yue Zhuo, Vu Minh Chien, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen GONG, James Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, David Lo, Binyuan Hui, Niklas Muennighoff, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro Von Werra. 2025. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *The Thirteenth International Conference on Learning Representations*.

A Parameter updating-based TTA in Real-world Scenarios

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; Su et al., 2023b), soft prompt (Lester et al., 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), adapter module (Houlsby et al., 2019; Muhtar et al., 2024; Su et al., 2023a) 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 (Wang et al., 2021; 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.

B Arguments about Self-correction

The effectiveness of self-correction 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., 2024d), planning (Valmeekam et al., 2023a), and graph coloring (Stechly et al., 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. 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.

C Future Directions

C.1 Generalizable System-2 Model

Currently, most o1-like models exhibit strong reasoning abilities only in specific domains such as math and code and struggle to adapt to cross-domain or general tasks. The key to addressing this issue lies in enhancing the generalization ability of verifiers or critics (LeVine et al., 2023; Kim et al., 2024d; Chen et al., 2024c). Currently, some works utilize multi-objective training (Wang et al., 2024b), model ensemble (Lin et al., 2024b) or regularization constraints (Yang et al., 2024; Jia, 2024) to make verifiers more generalizable. Nevertheless, there is still significant room for improvement in the generalization of the verifier. Additionally, weak-to-strong generalization (Burns et al., 2023) 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. We think that more generalized System-2 models may not rely on a single feedback source but instead obtain multi-source feedback through interactions between LLM-based agents and tools, experts, or other agents (Nathani et al., 2023; Lan et al., 2024).

C.2 Multimodal Reasoning

In System-1 thinking, TTA has been successfully applied to multimodal LLMs, improving perfor-

Category	sub-category	Representative Methods	Tasks	Verifier/Critic	Train-free
Repeat Sampling	Majority voting	CoT-SC (2023d)	Math, QA	self-consistency	✓
		PROVE (2024)	Math	compiler	✓
	Best-of-N	Cobbe et al. (2021)	Math	ORM	✗
		DiVeRSe (2023c)	Math	PRM	✗
Self-correction	Human feedback	NL-EDIT (2021)	Semantic parsing	Human	✗
		FBNET (2022)	Code	Human	✗
	External tools	DrRepair (2020)	Code	compiler	✗
		Self-debug (2024d)	Code	compiler	✓
		CRITIC (2024)	Math, QA, Detoxifying	text-to-text APIs	✓
	External models	REFINER (2024)	Math, Reason	critic model	✗
		Shepherd (2023b)	QA	critic model	✗
		Multiagent Debate (2023)	Math, Reason	multi-agent debate	✓
		MAD (2024b)	Translation, Math	multi-agent debate	✓
	Intrinsic feedback	Self-Refine (2023)	Math, Code, Controlled generation	self-critique	✓
		Reflexion (2023)	QA	self-critique	✓
		RCI (2023)	Code, QA	self-critique	✓
Tree Search	Uninformed search	ToT (2023)	Planing, Creative writing	self-critique	✓
		Xie et al. (2023)	Math	self-critique	✓
	Heuristic search	RAP (2023)	Planing, Math, Logical	self-critique	✓
		TS-LLM (2024b)	Planing, Math, Logical	ORM	✗
		rStar (2024)	Math, QA	multi-agent consistency	✓
		ReST-MCTS* (2024a)	Math, QA	PRM	✗

Table 1: Overview of search strategies.

mance in tasks such as zero-shot image classification, image-text retrieval, and image captioning (Zhao et al., 2024a). However, test-time compute methods in System-2 thinking remain limited to text modalities. Visual, speech, and other modalities are crucial for model understanding and interaction with the world. To achieve cognitive intelligence, System-2 models must be able to fully integrate multimodal information for reasoning. The exploration of multimodal CoT (Zhang et al., 2024j; Wu et al., 2024b; Mondal et al., 2024; Lee et al., 2024; Gao et al., 2024b) and multimodal critics or verifiers (Xiong et al., 2024) open up the possibility of building multimodal System-2 models. Xu et al. (2024) are the first to apply test-time compute to visual reasoning tasks. They divide the visual reasoning process into four stages: task summary, caption, reasoning, and answer conclusion. They propose a stage-level beam search method, which repeatedly samples at each stage and selects the best result for the next stage. Nowadays, Qwen team has released the open-weight multimodal reasoning model QVQ (Qwen, 2024), OpenAI and Kimi (Team et al., 2025) have released their multimodal reasoning products. We believe test-time compute still holds significant potential for development in multimodal reasoning. For example, incorporating more modalities like speech and video into reasoning tasks, applying successful methods such as reflection mechanisms and tree search to multimodal reasoning, or aligning the multimodal reasoning process with human cognitive processes. Besides understanding and reasoning tasks, Xie et al. (2025) and Guo et al. (2025) show test-time compute can improve image generation performance, with great potential for multimodal generation in the future.

C.3 Efficiency and Performance Trade-off

The successful application of test-time compute shows that sacrificing reasoning efficiency can lead to better reasoning performance. However, researchers continue to seek a balance between performance and efficiency, aiming to achieve optimal performance under a fixed reasoning latency budget. This requires adaptively allocating computational resources for each sample. Damani et al. (2024) train a lightweight module to predict the difficulty of a question, and allocate computational resources according to its difficulty. Zhang et al. (2024e) further extend the allocation targets to more hyperparameters. Chen et al. (2025) systematically evaluate the overthinking problem in o1-like models and mitigate it by length preference optimizing. 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., 2025), token pruning (Fu et al., 2024; Zhang et al., 2024c), and speculative decoding (Leviathan et al., 2023;

Xia et al., 2024) with test-time compute, and how to predict problem difficulty more accurately.

C.4 Scaling Law

Unlike training-time computation scaling, test-time compute still lacks a universal scaling law. Some works 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. (2024e) models 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.

C.5 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. 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. By effectively combining these strategies, we can develop System-2 models that achieve or surpass o1-level performance.

D Benchmarks and Open-source Frameworks

D.1 Benchmarks

Test-time Adaptation In System-1 models, distribution shifts include adversarial robustness, cross-domain and cross-lingual scenarios. In the field of CV, ImageNet-C (Hendrycks and Dietterich, 2019), ImageNet-R (Hendrycks et al., 2021a), 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 et al., 2020), MRQA (Fisch et al., 2019), CCMatrix (Schwenk et al., 2021) and Ted Talks (Qi et al., 2018).

Feedback Modeling 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., 2024d) further evaluates the impact of response style on reward models. RMB (Zhou et al., 2024) 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. (2025) propose PRMBench, which evaluates PRMs whether 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., 2024d) and MJ-Bench (Chen et al., 2024f).

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 et al., 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) employs 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.

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), 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) and BigCodeBench (Zhuo et al., 2025).

Commonsense reasoning requires LLMs to possess both commonsense knowledge and reasoning abilities. Early benchmarks (Zellers et al., 2019; Talmor et al., 2019; Sakaguchi et al., 2021; Bisk et al., 2020) focus on evaluating LLMs' commonsense ability. StrategyQA (Geva et al., 2021) collects more complex and subtle multi-hop reasoning questions. MMLU (Hendrycks et al., 2021b) and MMLU-Pro (Wang et al., 2024j) cover commonsense reasoning questions across various domains, including STEM, the humanities, the social sciences, etc.

Planning aims to enable LLMs to take optimal actions based on the current state and environment to successfully complete tasks. Current planning benchmarks primarily focus on small-scale synthetic tasks, such as Blocksworld (Valmeekam et al., 2023b), Crosswords, and Game-of-24 (Yao et al., 2023).

D.2 Projects

OpenR (Wang et al., 2024c)[1] is an open-source test-time reasoning framework that integrates various test-time compute strategies, PRM training, and improvement training. It currently supports beam search, BoN, MCTS, and rStar, and implements popular online reinforcement learning algorithms like APPO, GRPO, and TPPO.

RLHFlow (Dong et al., 2024a) offers a comprehensive framework for reward modeling[2] and online RLHF training[3]. Its standout feature is the integration of various reward model training methods, including the vanilla preference reward model, multi-objective reward models, PRM, etc.

OpenRLHF (Hu et al., 2024)[4] also integrates reward modeling and RLHF training but focuses more on the efficient implementation of reinforcement learning algorithms and training tricks. Its strength lies in the integration of distributed training and efficient fine-tuning, enabling users to easily train large language models with more than 70B parameters.

[1] https://github.com/openreasoner/openr
[2] https://github.com/RLHFlow/RLHF-Reward-Modeling
[3] https://github.com/RLHFlow/Online-RLHF
[4] https://github.com/OpenRLHF/OpenRLHF