

# A Survey on Parallel Reasoning

Ziqi Wang<sup>1,2\*</sup>, Boye Niu<sup>2,3\*</sup>, Zipeng Gao<sup>1\*</sup>, Zhi Zheng<sup>1</sup>, Tong Xu<sup>1</sup>  
Linghui Meng<sup>2</sup>, Zhongli Li<sup>2</sup>, Jing Liu<sup>2</sup>, Yilong Chen<sup>2</sup>, Chen Zhu<sup>1</sup>  
Hua Wu<sup>2</sup>, Haifeng Wang<sup>2†</sup>, Enhong Chen<sup>1†</sup>

<sup>1</sup>USTC, <sup>2</sup>Baidu, <sup>3</sup>USYD

\*Equal Contributors, †Corresponding Authors

{wzq142857, gaozp619}@mail.ustc.edu.cn, bniu6645@uni.sydney.edu.au

## Abstract

With the increasing capabilities of Large Language Models (LLMs), parallel reasoning has emerged as a new inference paradigm that enhances reasoning robustness by concurrently exploring multiple lines of thought before converging on a final answer. It has become a significant trend to explore parallel reasoning to overcome the fragility of standard sequential methods and improve practical performance. In this paper, we aim to survey and summarize the progress and challenges of parallel reasoning. We first present a formal definition of parallel reasoning and clarify its distinction from related concepts like Chain-of-Thought. Then, we organize and discuss advanced techniques based on a novel taxonomy, including non-interactive reasoning, interactive reasoning, and efficiency-focused decoding strategies. Additionally, we explore various application scenarios, such as solving complex problems and enhancing the reliability of LLM outputs. Finally, we highlight the core challenges of parallel reasoning and suggest potential directions for future research. We hope that our work can provide a useful roadmap for beginners and encourage more research on improving parallel reasoning methods. Related source can be available in <https://github.com/PPPP-kaqiu/Awesome-Parallel-Reasoning>.

## 1 Introduction

Modern large language models (LLMs) have acquired powerful foundational capabilities by scaling parameters and training data during their development (Brown et al., 2020; Chowdhery et al., 2023; OpenAI, 2023; Touvron et al., 2023a,b; Team et al., 2025). Subsequent efforts explored inference-time scaling that extends the Chain-of-Thought (CoT), demonstrating significant improvements in reasoning performance (Wei et al., 2022; OpenAI, 2024; DeepSeek-AI et al., 2025). Building on these successes, we examine an orthogonal

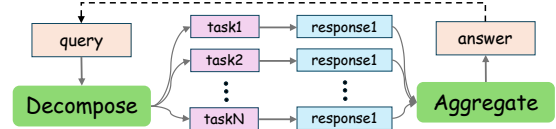


Figure 1: An overall framework for a recurrent parallel reasoning paradigm. The process begins with a decomposition stage, followed by parallel processing of sub-tasks, and concludes by aggregating the outputs into a single answer.

approach: concurrently exploring multiple reasoning paths to broaden inference and further enhance reasoning depth and breadth.

Standard sequential reasoning often proves fragile on complex tasks, as it can fall into the so-called “prefix trap” (Luo et al., 2025); once the model commits to an early reasoning path, it struggles to self-correct and may never reach the optimal solution. This weakness is starkly illustrated by the gap between single-pass performance (Pass@1) and the best outcomes achieved through multiple samples (Pass@k), showing that a sequential reasoning approach cannot fully leverage the model’s potential (Chen et al., 2021). In contrast, parallel reasoning mimics a breadth-first search by exploring multiple paths at once, boosting robustness and tapping into a wider range of the model’s problem-solving abilities.

The key idea of parallel reasoning is to make multiple attempts in parallel before answering a question and then aggregate the proposed solutions. Figure 1 provides an example of this paradigm. The process begins with a divergence step, wherein the language model concurrently explores multiple reasoning paths. This is achieved either by generating several complete solution candidates in parallel or by decomposing the problem into sub-tasks that are solved simultaneously. Following this parallel generation, a convergence step aggregates the various outputs to produce a single, final response.

In contrast to the serial execution of a single reasoning path typical of sequential reasoning, the parallel method is designed to enhance robustness by identifying the correct answer from a diverse set of potential solutions.

As a novel inference paradigm, parallel reasoning (PR) offers several compelling advantages. First, by exploring multiple reasoning paths before converging on a final answer, PR can produce higher-quality responses than single-path methods — thereby enhancing user experience in real-world applications. This represents a genuine improvement in the model’s practical performance, distinguishing it from metrics like pass@K, which reflect theoretical potential but, being ground-truth-based, do not measure the performance a user actually experiences. Second, PR is orthogonal to the chain-of-thought paradigm (Wei et al., 2022; Lightman et al., 2024; Fu et al., 2023): whereas CoT extends reasoning depth through step-by-step inference, PR expands reasoning breadth via concurrent exploration. This design enables PR to scale independently and continue benefiting from advances in foundation model capabilities. Third, PR can achieve comparable or even superior performance to sequential reasoning with improved computational efficiency, especially when paired with algorithmic-system co-design optimizations such as KV cache reuse (Yang et al., 2025a). Furthermore, the core philosophy of parallelism embodied by PR also informs broader strategies for accelerating inference, including techniques like parallel decoding and speculative execution.

To structure this landscape, Figure 2 presents a taxonomy of parallel reasoning along three key dimensions: (1). Non-interactive Parallel Reasoning: The model independently generates various attempts and then produces a final reply. (2). Interactive Parallel Reasoning: This involves active interaction within the reasoning paths during the inference process. (3). Efficiency: This covers both engineering designs aimed at accelerating parallel reasoning and novel decoding methods guided by the principles of parallelism. We highlight the challenges and potential directions, and hope our work provides a useful road map for beginners interested in this area and sheds light on future research.

## 2 Definition and Formulation

Here, we provide a formal definition of parallel reasoning, an inference-time paradigm that enhances

a language model’s reasoning capacity by generating and aggregating multiple concurrent reasoning paths from a single query, thereby producing a more robust final response.

Formally, given an input query  $Q$ , a language model  $M$  produces a final prediction  $\Pi(Q)$  through a three-stage pipeline consisting of decomposition, parallel processing, and aggregation:

$$\Pi(Q) = (A \circ P_M \circ D)(Q)$$

Here,  $D$  is a decomposition operator that maps the input query to a set of sub-inputs, which may represent distinct sub-inputs or identical queries intended to initiate diverse reasoning paths;  $P_M$  denotes the parallel application of model  $M$  to these inputs; and  $A$  is an aggregation operator that synthesizes the intermediate results into the final response.

This formulation captures the essence of parallel reasoning: the ability to process multiple reasoning paths simultaneously and synthesize them into a coherent output. It stands in contrast to traditional sequential reasoning, which processes a single path in a strictly serial manner.

**Decomposition ( $D$ ):** Given a query  $Q$ , the decomposition operator generates a set of  $n$  sub-inputs  $\mathbf{T} = \{T_1, T_2, \dots, T_n\}$ , where each  $T_i$  represents a distinct sub-task or problem component:

$$D(Q) \rightarrow \{T_1, \dots, T_n\}$$

In the simplest case,  $D$  can be an identity mapping that duplicates the original query, i.e.,  $T_i = Q$  for all  $i$ , allowing the model to explore diverse reasoning trajectories from the same prompt.

**Parallel Processing ( $P_M$ ):** Applies the language model  $M$  concurrently to each sub-input  $T_i$ , producing a set of intermediate results  $\mathbf{R} = \{R_1, \dots, R_n\}$ :

$$(R_1, \dots, R_n) = P_M(T_1, \dots, T_n)$$

**Aggregation Operator ( $A$ ):** Combines the set of intermediate results  $\mathbf{R} = \{R_1, \dots, R_n\}$  into a single, coherent prediction.

$$\Pi(Q) = A(R_1, \dots, R_n)$$

The behavior of  $A$  is characterized by two key properties: its *granularity*—whether aggregation occurs at the sequence level or token level—and the choice of *aggregation function*.

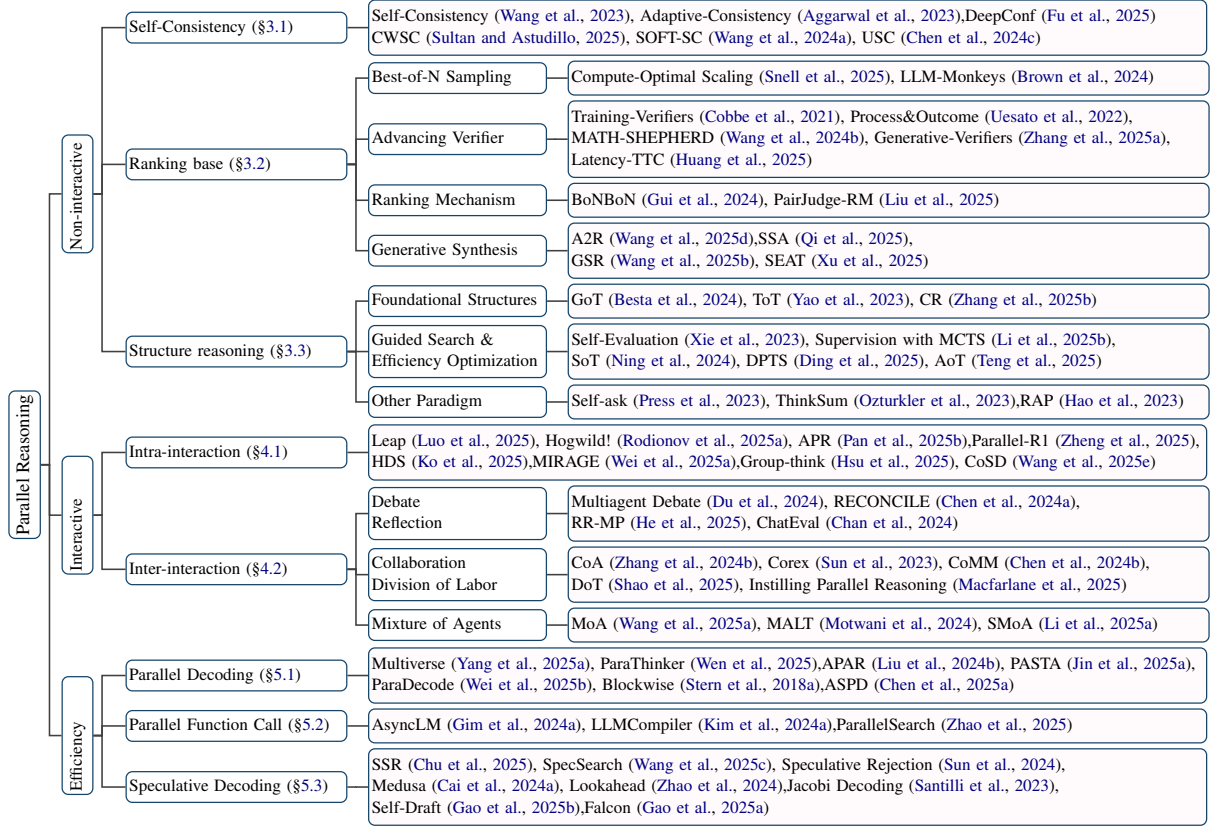


Figure 2: Taxonomy of Parallel Reasoning

Based on this definition, parallel reasoning is distinguished from other related inference-time strategies. Unlike Chain-of-Thought, which decomposes a problem into a strictly sequential series of steps, parallel reasoning processes multiple sub-inputs concurrently. A further distinction lies with long-thinking processes, which scale reasoning horizontally by iteratively extending a single reasoning chain through self-reflection and self-refinement. In contrast, parallel reasoning scales computation vertically by simultaneously generating multiple distinct reasoning paths.

### 3 Non-interactive

In this section, we introduce **Non-interactive Parallel Reasoning** as a foundational paradigm wherein multiple candidate reasoning paths are generated without overt communication, followed by a convergence step that aggregates these diverse outputs to produce a single answer.

$$\Pi(Q) = A(P_M(Q_1, \dots, Q_n))$$

where  $Q_1, \dots, Q_n$  are identical copies of the input query  $Q$ . Our examination of this paradigm is structured around three categories. We begin with the seminal approach of aggregation by self-consistency, which first established the power of

consensus in improving accuracy. The paradigm then evolved into the now-dominant method of aggregation by ranking—commonly known as Best-of-N sampling—where verifiers or reward models are used to select the optimal answer. Finally, we cover a parallel line of research that employs structured exploration to enable the parallel execution of distinct reasoning branches.

#### 3.1 Self-Consistency

The foundational concept of parallel reasoning stems from Self-Consistency (Wang et al., 2023). This approach prompts a model to generate multiple responses and then selects the most common one through a voting process.

$$A(R) = \operatorname{argmax}_a \sum_{i=1}^n \mathbf{I}(E(P_M(Q_i)) = a)$$

where  $E(\cdot)$  is an extraction function that isolates the final answer from an output sequence,  $a$  is a candidate answer from the potential answer space, and  $\mathbf{I}(\cdot)$  is the indicator function. The underlying principle is that as the number of generated answers increases, the model converges toward a high-confidence solution that more accurately represents its true problem-solving capabilities, whereas a single reasoning path is more susceptible to noise.

However, this method is computationally expensive and not always practical for general scenarios. To address these limitations, early work like Adaptive-Consistency (Aggarwal et al., 2023) introduced a dynamic stopping criterion to optimize the computational budget. More advanced methods such as DeepConf (Fu et al., 2025) further refine this by leveraging the model’s internal confidence scores. DeepConf can dynamically terminate low-confidence reasoning paths during generation or use these scores to weight the final votes, drastically reducing computation while boosting accuracy. Following a similar principle, the Confidence-Weighted Token Set Cover (Sultan and Astudillo, 2025) approach improves token efficiency by pruning intermediate hypotheses based on two indicators: the model’s confidence in each path and the lexical diversity across all paths. It uses a fast weighted set cover algorithm to select a high-quality and diverse set of reasoning paths, achieving token savings of 10-35% in many cases.

To generalize the approach to open-ended tasks, Soft Self-Consistency (Wang et al., 2024a) replaced majority voting with a "soft" vote based on the model’s aggregated generation probabilities, shifting the focus from the most common answer to the one with the highest model-assigned confidence. In parallel, Universal Self-Consistency (Chen et al., 2024c) uses the LLM itself to judge and select the most consistent response from a set of candidates. This evolution from simple consensus to model-based judgment foreshadowed the development of more sophisticated aggregation paradigms.

### 3.2 Ranking-Base

The Self-Consistency method determines the final answer based on a consensus among multiple generated outputs, typically through majority voting. However, its limitation lies in the possibility that the correct answer, while present within the solution space, may not be the most frequent one and is thus never selected. In contrast, ranking-based methods address this by utilizing an external scoring function—a verifier or reward model(RM)—to evaluate and rank all candidate answers.

$$A(R) = E \left( \operatorname{argmax}_{R_i \in \mathbf{R}} V(R_i) \right)$$

where  $V(\cdot)$  is a scoring or verification function that assigns a quality score to an output sequence. This is commonly implemented through a Best-of-N (BoN) sampling paradigm, which selects the

highest-scoring solution from the generated set. Consequently, this approach places immense importance on the quality and sophistication of the selection mechanism, which has become a major area of research in its own right.

The **BoN** sampling paradigm operates on a simple premise: a model’s ability to generate at least one correct solution across  $N$  attempts is strictly greater than in a single attempt. Cobbe et al. (2021) first demonstrated this approach’s effectiveness by training a verifier to select correct answers from numerous generated solutions to math problems, significantly boosting performance. Subsequent work revealed that this improvement follows predictable inference-time scaling laws. Notably, Large Language Monkeys (Brown et al., 2024) found that coverage—the fraction of problems with at least one correct solution—scales consistently with the number of samples  $N$ . Formalizing this further, Compute-Optimal Scaling (Snell et al., 2025) analyzed the trade-off between model size and sample count for a fixed inference budget. Their findings suggest that for large budgets, increasing the sample count  $N$  is often a more effective path to improvement than simply scaling up the model. These studies establish BoN as a fundamental axis for scaling model capabilities, but they also illuminate two parallel challenges: the efficiency of compute allocation and the quality of the verifier.

Addressing the first challenge, the scaling laws reveal a core inefficiency of fixed strategies, which tend to overspend compute on simple queries while under-provisioning for complex ones. To solve this, Latency-TTC (Huang et al., 2025) introduces a query-adaptive approach that dynamically selects the optimal inference method and budget for each query. Its key innovation lies in a utility function that uniquely incorporates wall-clock latency—a critical factor for user experience—in addition to token cost. Guided by lightweight predictors of accuracy and cost, this adaptive strategy achieves a superior accuracy-efficiency trade-off compared to static approaches.

Complementing these system-level optimizations for compute, a parallel line of research addresses the second challenge by improving the core ranking component itself: the **verifier**. These verifiers, which provide the critical signal for ranking candidates, have evolved into two main approaches: Outcome Reward Models(ORMs) and Process Reward Models(PRM) (Uesato et al., 2022). An ORM evaluates an entire reasoning chain and as-



signs a single, holistic score based solely on the final answer’s correctness. In contrast, a PRM provides more granular feedback by assigning a reward to each intermediate step, a form of supervision that ensures the logical coherence of the process and not just the accuracy of the final output. The Generative Verifier(GenRM) (Zhang et al., 2025a) reframes verification as a next-token prediction task. Instead of outputting a score, the verifier generates a "Yes" or "No" token to answer the question, "Is the following solution correct?", with the reward derived from the token’s probability. This approach also enables GenRM-CoT, where the verifier first generates a rationale explaining its assessment before delivering a verdict. Concurrently, the MATH-SHEPHERD (Wang et al., 2024b) automates process-wise supervision by labeling a reasoning step as "good" if it has a high potential to lead to the correct final answer. Such innovations in verifier training have been spurred to break the data bottleneck created by the expensive, step-by-step human annotations upon which the superiority of traditional PRMs depends.

Given the difficulty of assigning reliable absolute scores, advanced **ranking mechanisms** often shift to relative judgments. PairJudge RM (Liu et al., 2025) trains a model on pairwise comparisons, using a knockout tournament to iteratively filter candidates until a single champion emerges. To address the computational expense of such powerful verifiers, BoNBoN (Gui et al., 2024) employs a multi-stage, hierarchical strategy, in which an efficient ranker first filters a large set of candidates into a promising subset, which is then re-ranked by a more powerful and computationally expensive verifier. This approach significantly improves efficiency by focusing intensive computation only on the most viable solutions.

A more profound shift moves beyond mere selection to **generative synthesis**, where the goal is to construct a new, superior solution by integrating insights from all candidates. This is often realized through a two-stage "explorer-synthesizer" architecture. For instance, the Asymmetric Two-Stage Reasoning(A2R) identifies that the synthesizer’s capability is the critical driver of performance, which leads to an efficient “small-to-big” configuration—using a small model for parallel exploration and a larger model for synthesis—that can surpass the performance of a much larger monolithic model at a lower cost. Similarly, the Sample Set Aggregator(SSA) (Qi et al., 2025) and SEAT (Xu et al.,

2025) frameworks train dedicated aggregator models with reinforcement learning to review, reconcile, and synthesize a final answer from a set of candidates. AGGLM specifically highlights the importance of a balanced training curriculum of easy and hard problems to ensure the model can recover correct answers even when they are not the most frequent. In contrast to these decoupled approaches, Generative Self-Refinement(GSR) (Wang et al., 2025b) trains a single, unified model to perform both roles, first generating diverse candidates and then executing self-refinement on its own outputs. This dual capability is instilled through a hybrid training pipeline that jointly optimizes for both direct problem-solving and refinement tasks.

### 3.3 Structure Reasoning

A parallel line of research marks a pivotal shift from the linear progression of Chain-of-Thought to **structured reasoning**, a paradigm of dynamic problem-solving topologies that shares the conceptual goal of generative synthesis in constructing a final solution by exploring and integrating multiple lines of thought. This began with Tree-of-Thoughts (Yao et al., 2023), which framed reasoning as a tree search, enabling the parallel exploration, self-evaluation, and pruning of multiple solution paths. Graph-of-Thoughts (Besta et al., 2024) advanced this concept further by employing a graph structure, which unlocks more sophisticated operations like the aggregation of disparate reasoning lines and the refinement of ideas through feedback loops. Complementing these exploration-based methods, Cumulative Reasoning (Zhang et al., 2025b) offers a distinct, iterative approach that ensures logical robustness by sequentially solving sub-problems and accumulating each validated result into the context for the next step. Together, these methods define a new paradigm that moves beyond simple pathfinding to instead construct solutions by synthesizing insights from multiple, potentially incomplete paths, hinting at a future focused on developing a "thought algebra" for formally combining reasoning intermediates.

The vast search spaces and significant inference overhead introduced by these complex structures necessitate two parallel lines of optimization: **guided search strategies and efficiency enhancements**. To navigate the combinatorial explosion of possibilities, Self-Evaluation Guided Beam Search (Xie et al., 2023) uses the model’s intrinsic ability to critique candidate steps as a heuristic to

prune the search beam. In a different paradigm, Process Supervision with Monte Carlo Tree Search (Li et al., 2025b) uses the search algorithm not just for guidance but also to generate fine-grained data for progressively enhancing the model’s capabilities. Alongside these search improvements, new efficiency optimizations have emerged at distinct layers of abstraction. At the prompting layer, Skeleton-of-Thought (Ning et al., 2024) reduces latency by generating a concise outline before parallel expansion; at the system layer, Dynamic Parallel Tree Search (Ding et al., 2025) optimizes execution with advanced KV cache management; and at the theoretical layer, Atom-of-Thoughts (Teng et al., 2025) redefines the process to prevent computational load from accumulating. These developments highlight a crucial co-evolution: complex reasoning structures demand sophisticated search and optimization algorithms, which in turn enable and validate more elaborate structures.

A final class of frameworks adopts a different philosophical approach, fundamentally shifting the "locus of intelligence" by redefining the LLM’s role. Self-ask (Press et al., 2023) still treats the LLM as a complete reasoner but compels it to externalize its thought process by breaking down complex problems into explicit sub-questions. ThinkSum (Ozturkler et al., 2023) introduces a sharper division of labor, recasting the LLM as a specialized "System 1" for associative knowledge retrieval while offloading formal "System 2" logical inference to an external model. Taking this deconstruction even further, Reasoning via Planning (Hao et al., 2023) reframes the task as a classical AI planning problem, repurposing the LLM as a flexible semantic simulator while an external algorithm like MCTS provides the overarching strategic intelligence. This conceptual shift—from treating the LLM as a monolithic reasoner to a specialized component—signals the rise of hybrid cognitive architectures and the emerging discipline of "cognitive engineering."

## 4 Interactive

Interactive parallel reasoning is a finer-grained paradigm in which multiple reasoning paths or agents operate in parallel while dynamically exchanging information during inference, rather than generating independently and aggregating only at the end. Formally,

$$\begin{aligned}\Pi(Q) &= A\left(\{R_i^{(T)}\}_{i=1}^n\right), \\ R_i^{(t+1)} &= P\left(T_i; \{R_j^{(t)}\}_{j \neq i}\right).\end{aligned}$$

where  $R_i^{(t)}$  denotes the intermediate output of the  $i$ -th branch at interaction round  $t$ . Unlike the non-interactive case, each branch is iteratively updated based not only on its own input  $T_i$  but also on the intermediate outputs from other branches  $\{R_j^{(t)} : j \neq i\}$ . The process continues for  $T$  rounds of interaction before applying the aggregation operator to produce the final answer.

This interaction can take two main forms: **intra-interaction** (Section 4.1), where different reasoning threads within a single model share information and adjust their trajectories during the generation process, and **inter-interaction** (Section 4.2), where multiple autonomous models or agents collaborate by exchanging intermediate results or engaging in dialogue. By enabling real-time communication among concurrent processes, interactive parallel reasoning supports dynamic error correction, mitigates the influence of early mistakes, and yields more robust and efficient reasoning.

### 4.1 Intra-interaction

$$\begin{aligned}\Pi(Q) &= A\left(\{R_i^{(T)}\}_{i=1}^n\right), \\ R_i^{(t+1)} &= P\left(T_i, \mathcal{S}\left(\{R_j^{(t)} : j \neq i\}\right)\right).\end{aligned}$$

Intra-interaction parallel reasoning allows different reasoning paths within a single model to share intermediate information during generation. Formally, each branch is updated by referencing both its own input  $T_i$  and the shared information  $\mathcal{S}(\cdot)$  aggregated from other branches, and the final answer is obtained through aggregation.

Unlike independent sampling or post-hoc ensembling, intra-interaction parallel reasoning allows the reasoning paths to exchange information rather than remain isolated. Share information during the reasoning process itself so that each path can be updated in light of what others are producing. This makes the paradigm both parallel, as multiple trajectories are explored simultaneously, and interactive, since intermediate states are exchanged rather than aggregated only at the end. Within this general setting, prior works differ mainly in

when and how such exchanges occur. Some methods focus on scheduling decisions, controlling the number of parallel branches to create and when to terminate them. Others introduce staged or step-wise mechanisms that allow paths to periodically exchange summaries or revisions. Still others operate directly at the decoding level, enabling token-by-token visibility and continuous cross-attention across streams. We next review representative works under each of these categories. One group of methods focuses on scheduling-oriented interaction, where the model adaptively controls the number of branches to explore and decides when to terminate them. Entropy-based termination exploits the negative correlation between semantic diversity and correctness to halt exploration once consensus emerges, thereby reducing redundant computation (Xu et al., 2025). Thread-based frameworks capture the same adaptive principle by introducing explicit spawn and join operations: a spawn corresponds to opening additional reasoning branches when more exploration is judged useful, while a join corresponds to merging or terminating branches once verification is sufficient (Pan et al., 2025b). Extensions to retrieval-augmented reasoning push this idea further by enabling hybrid scheduling: some sub-queries are dispatched in parallel to maximize coverage, while others are routed sequentially to preserve dependency constraints, yielding a dynamic mix of parallel and serial processing within the same task (Ko et al., 2025). Finally, reinforcement learning has also been applied to directly instill adaptive branching policies into the model itself. Parallel-R1 adopts a progressive curriculum that first familiarises the model with the format of parallel reasoning on simpler problems, then transitions to RL with alternating structural and outcome-based rewards (Zheng et al., 2025).

A second line of work introduces staged or step-wise interaction, in which different reasoning paths periodically share partial results to incorporate peer feedback. In this setting, LeaP allows each path to generate concise summaries that are routed to other paths under diverse or clustered strategies, helping reduce error propagation and enabling peers to escape locally suboptimal trajectories (Luo et al., 2025). Collaborative speculative decoding (CoSD) extends speculative decoding by letting multiple drafts exchange probability and semantic information during generation rather than only at the end, thus turning speculative exploration into a collaborative process (Wang

et al., 2025e). MIRAGE follows a similar staged philosophy in retrieval-augmented reasoning: it decomposes a complex query into multiple sub-questions, launches parallel reasoning chains that iteratively retrieve and update knowledge from a shared workspace, and finally employs a synthesis stage that cross-validates chains and resolves conflicts across their outputs (Wei et al., 2025a). Together, these methods highlight the role of periodic summarisation, cross-sharing, and staged verification in enabling more robust reasoning than independent parallel sampling. At the decoding level, the most tightly coupled approaches give paths continuous access to each other’s outputs. Group Think realises this by instantiating multiple thinkers whose token streams are mutually visible at every step, producing emergent division of labour and reduced redundancy (Hsu et al., 2025), while Hogwild! Inference achieves a similar effect by enabling concurrent workers to share key-value caches, with RoPE-based adjustments ensuring positional consistency so that each branch can immediately draw on the evolving memory of others (Rodionov et al., 2025a). Together, these methods illustrate a spectrum of intra-interaction strategies, ranging from adaptive scheduling to periodic summarisation and revision to token-level collaboration, all adhering to the principle that each reasoning path is updated not only from its own trajectory but also from shared information aggregated from its peers.

## 4.2 Inter-interaction

$$\Pi(Q) = A\left(\{R_i^{(T)}\}_{i=1}^k\right),$$

$$R_i^{(t+1)} = M_i\left(Q, \mathcal{I}\left(\{R_j^{(t)} : j \neq i\}\right)\right).$$

Inter-interaction parallel reasoning involves multiple models or agents  $\{M_1, M_2, \dots, M_k\}$  that exchange intermediate results during inference. Each agent  $M_i$  updates its output  $R_i^{(t+1)}$  based on the original query  $Q$  and the interaction function  $\mathcal{I}(\cdot)$  aggregating the contributions of other agents. The final prediction  $\Pi(Q)$  is obtained by applying the aggregation operator  $A$  over all agents’ outputs.

Inter-interaction parallel reasoning refers to settings where collaboration occurs across multiple autonomous models or agents, rather than within a single model’s internal processes. In this paradigm, reasoning is distributed among different agents that explicitly exchange intermediate results or engage in conversation. One prominent style is **debate and**

**reflection**, where multiple agents engage in structured dialogue to reach stronger conclusions than any single model could achieve. Multi-agent debate frameworks show that when independent LLM instances propose answers and then iteratively critique or defend them, the resulting consensus significantly improves both factuality and reasoning quality, while reducing hallucinations (Du et al., 2024). Building on this “society of minds” perspective, round-table designs extend the debate to diverse models, incorporating confidence scores and human demonstrations to guide persuasion and convergence, thereby leveraging model heterogeneity as a key driver of performance gains (Chen et al., 2024a). Other work emphasises combining debate with reflection, pairing reactive agents with reflective counterparts in multi-path settings to prevent degeneration of thought and strengthen reasoning robustness, particularly in scientific domains (He et al., 2025). Beyond task-solving, debate has also been applied to evaluation itself: by treating assessment as a group deliberation problem, multi-agent referee teams collaboratively discuss system outputs, leading to judgments that better approximate human evaluation standards (Chan et al., 2024).

Beyond dialogue-based approaches, another major direction within inter-interaction parallel reasoning is **collaboration and division of labour**, which emphasises distributing reasoning workloads across agents or roles in a coordinated manner. A first line of work focuses on task decomposition and management, where complex problems are split into sub-tasks handled by different agents before being synthesized. For example, segmenting long-context inputs among worker agents with a manager responsible for aggregation alleviates context window limitations, while hybrid architectures allocate simpler tasks to lightweight local models and outsource more complex components to larger cloud-based LLMs, thereby reducing costs while maintaining accuracy (Zhang et al., 2024b; Shao et al., 2025). A second group emphasises role specialisation and heterogeneous collaboration, where agents are prompted to take on complementary roles or adopt distinct reasoning modes. By incorporating mechanisms such as discussion, review, and retrieval, or by assigning expert roles along different reasoning paths, these frameworks harness diversity in perspectives to improve performance on challenging scientific and reasoning tasks (Sun et al., 2023; Chen et al., 2024b). Finally, a third direction explores instilling parallel reasoning capa-

bilities within a single model, where parallel traces distilled from teacher models enable the student model to decompose problems, explore multiple strategies concurrently, and dynamically allocate reasoning depth based on complexity, effectively internalising the benefits of multi-agent division of labor (Macfarlane et al., 2025). Together, these approaches demonstrate how decomposition, specialisation, and knowledge distillation provide complementary pathways to scale reasoning through collaborative or distributed structures, improving robustness, efficiency, and adaptability.

Building on both debate-based and division-of-labour paradigms, a further strand of inter-interaction research develops **mixture-of-agent** methods, which focus on harnessing the complementary expertise of multiple agents through structured mixture mechanisms. One approach builds a layered architecture where each agent conditions on the outputs of others in previous layers, allowing collective strengths to be integrated across stages and yielding performance that surpasses even state-of-the-art single models (Wang et al., 2025a). A related line of work explores mixture-of-agents in the training paradigm: by dividing reasoning into generation, verification, and refinement, and constructing a multi-agent search tree, cooperative training frameworks enable agents to specialize in different roles and learn from both successful and failed trajectories, ultimately strengthening end-to-end reasoning quality (Motwani et al., 2024). While these dense multi-agent structures can be powerful, they often introduce efficiency and redundancy challenges; to address this, sparse mixture frameworks restrict interactions among agents through mechanisms such as selective response sharing and early stopping, while also encouraging diversity through distinct role descriptions, thereby achieving performance comparable to dense systems at a fraction of the computational cost (Li et al., 2025a). Taken together, these works highlight how mixture-of-agents, whether through layered inference architectures, cooperative training pipelines, or sparse interaction designs, provide a flexible and scalable means to integrate multiple reasoning agents into a unified system that balances performance, efficiency, and diversity.

## 5 Efficiency

The approaches discussed earlier primarily focus on the horizontal expansion of reasoning, enabling



large language models to explore multiple reasoning paths in parallel so as to improve the quality and robustness of their outputs. In this section, we shift our attention to strategies for enhancing the efficiency of parallel reasoning, which can be categorized by the granularity at which the optimization is applied, including parallel decoding, parallel function calling, and speculative decoding. We formalize these optimization methods as follows:

$$\Pi(Q) = (A \circ P_M \circ D^{(V,\ell)})(Q),$$

where  $D^{(V,\ell)}$  operates along the vertical (sequential) axis of a single reasoning trajectory, and  $\ell \in \{\text{task, call, token}\}$  denotes the granularity of decomposition, which corresponds to parallel decoding, parallel function calling, and speculative decoding, respectively. At each step, the vertical decomposition operator  $D^{(V,\ell)}$  acts on the input task  $Q$  and the previous state  $s_{t-1}$ , partitioning the sequential reasoning process into a set of parallelizable or pipelined sub-units:

$$U_\ell(s_{t-1}) = D^{(V,\ell)}(Q, s_{t-1}) = \{T_1, \dots, T_m\}.$$

Here,  $U_\ell(s_{t-1})$  represents the vertical decomposition result at granularity  $\ell$ , and each element  $T_i$  corresponds to an independent computation unit. The outputs of these vertically decomposed units are processed by  $P_M$  in parallel or pipeline fashion and then integrated by  $A$  with order consistency. In essence, efficiency-oriented methods reinterpret  $D$  as a vertical decomposer that reorganizes the temporal granularity of inference to enable higher concurrency within a single reasoning trajectory.

## 5.1 Parallel Decoding

At the macro level, parallel decoding explores task-level or semantic-level parallelism. By decomposing forthcoming decoding tasks into semantically or structurally independent units and executing them concurrently, these approaches reduce inference latency while preserving generation quality. APAR (Liu et al., 2024c) fine-tunes LLMs to recognize hierarchical structures (e.g., list items) and fork parallel decoding threads. Multiverse (Yang et al., 2025b) adopts a Map–Process–Reduce paradigm, training models to decompose tasks, process subtasks in parallel, and merge results losslessly. ParaDecode (Wei et al., 2024) pipelines token generation by leveraging intermediate-layer

confidence to finalize tokens early and begin subsequent computations in parallel, ensuring identical outputs with standard decoding. PASTA (Jin et al., 2025b) introduces a learned annotation language to mark semantically independent text segments, which an interpreter decodes asynchronously, nearly doubling inference speed with negligible impact on output quality. ASPD (Chen et al., 2025a) detects intrinsically parallelizable branches and employs a hybrid engine to seamlessly switch between sequential and parallel modes. Hogwild (Rodionov et al., 2025b) enables multiple LLM instances to generate concurrently with shared attention caches, requiring no additional fine-tuning and delivering near-linear scaling.

## 5.2 Parallel Function Call

At the system level, parallel function calling introduces parallelism through external tool coordination. Instead of modifying the decoding process itself, it enables LLMs to schedule and execute multiple function calls simultaneously, reducing sequential bottlenecks and improving end-to-end responsiveness. AsyncLM (Gim et al., 2024b) realizes this idea by introducing asynchronous interrupts into generation, allowing the model to continue decoding while functions execute in parallel, and employing scheduling heuristics such as Longest Processing Time first to manage multiple concurrent calls, achieving latency reductions of up to 5.4× without loss of accuracy. LLMCompiler (Kim et al., 2024b), in contrast, adopts a compiler-inspired approach: it decomposes multi-call queries into optimized execution plans via a dedicated function calling planner, and dispatches tasks in parallel through a specialized executor.

## 5.3 Speculative Decoding

At the micro level, speculative decoding achieves token-level parallelism within the autoregressive process by adopting a draft-and-verify paradigm (Leviathan et al., 2023; Chen et al., 2023; Xia et al., 2023a), which accelerates generation and mitigates the sequential bottleneck of token-by-token decoding. The key challenge lies in producing high-quality drafts that can be efficiently verified by the target model. A common approach employs a smaller, faster draft model from the same family as the target LLM to ensure distributional alignment (Leviathan et al., 2023; Chen et al., 2023; Xia et al., 2023a). Acceptance can

be further improved through multi-sample generation (Yang et al., 2024) or multiple parallel draft models (Miao et al., 2024), though these designs introduce inter-model coordination overhead and retain non-trivial autoregressive costs.

Motivated by the efficiency limitations of conventional speculative decoding, recent work explores more lightweight and self-sufficient alternatives. REST (He et al., 2024) removes auxiliary models by retrieving contextually relevant tokens from a pre-built corpus and verifying them in parallel, while HD (Cho et al., 2025) organizes heterogeneous corpora hierarchically to improve draft quality. Building on this idea, CS Drafting (Chen et al., 2025b) and StagedSpec (Spector and Re, 2023) coordinate multiple draft modules with varying computational costs to reduce latency while maintaining diversity.

Another line of work performs model-internal drafting, where the LLM itself generates and verifies candidate drafts without external auxiliaries. Methods such as Draft & Verify (Zhang et al., 2024a), SWIFT (Xia et al., 2025), LayerSkip (Elhoushi et al., 2024), and Kangaroo (Liu et al., 2024a) leverage internal representations or adaptive layer skipping to accelerate decoding. Structural extensions further enhance efficiency: Blockwise Decoding (Stern et al., 2018b), MEDUSA (Cai et al., 2024b), Hydra (Ankner et al., 2024), and the Eagle/Falcon series (Li et al., 2025d, 2024, 2025c; Gao et al., 2025a) employ multi-head or multi-module architectures to parallelize token generation and verification.

Complementary approaches, including PaSS (Monea et al., 2023), SpeDec (Xia et al., 2023b), Jacobi Decoding (Santilli et al., 2023), and LADE (Fu et al., 2024), achieve acceleration purely through decoding adjustments without retraining, while Self-Draft (Gao et al., 2025b) utilizes the model’s intrinsic robustness to produce context-aware drafts from static corpus priors. Together, these advances demonstrate that efficiency-oriented speculative decoding can be realized through corpus retrieval, model-internal drafting, or architectural parallelization—all aiming to minimize redundancy and maximize throughput.

## 6 Application

Recently, parallel reasoning has emerged as a powerful paradigm for solving complex problems and

has been widely adopted by leading AI research teams. Beyond pushing the frontiers of performance on grand challenge benchmarks, the principles of parallel reasoning are being applied to enhance the reliability, efficiency, and creativity of large language models in a variety of practical settings.

**Solving Grand Challenge Problems** Gemini pioneered an advanced parallel reasoning approach, Deepthink (Google, 2025), achieving a gold medal-level performance at the International Mathematical Olympiad (IMO) and demonstrating the striking potential of these techniques. Following this, other frontier models, such as Grok4(xAI, 2025) and Claude4 (Anthropic, 2025b), have integrated similar parallel technologies to significantly boost their performance on mathematical and coding tasks. More recently, models like SEED1.6 (Seed, 2025) and Qwen3 (Qwen, 2025) have likewise employed parallel decoding techniques to achieve remarkable breakthroughs in competitive contests such as AIME, even approaching perfect scores. Beyond complex task-solving, these parallel techniques are also being applied in advanced research systems. For instance, Anthropic utilizes prompt engineering to enhance its multi-agent parallel function calling system (Anthropic, 2025a), enabling it to tackle real-world research questions with high efficiency and low latency.

**Enhancing Reliability and Factuality** A significant application of parallel reasoning is in mitigating hallucinations and improving the factual accuracy of LLM outputs. By generating multiple, diverse reasoning paths, models can identify and discard factual inconsistencies or logical fallacies that might arise in a single-path generation. This is exemplified in multi-agent (Du et al., 2024; Chen et al., 2024a; Wang et al., 2023) frameworks, where multiple models propose, critique, and refine answers to yield a final consensus that is significantly more factual and robust.

**Accelerating Complex Workflows** Through parallel function calling, models can schedule and run multiple functions concurrently instead of waiting for them to execute sequentially. This dramatically reduces bottlenecks in real-world applications requiring data retrieval or external computations. Furthermore, models can utilize optimized execution plans (Gim et al., 2024a; Kim et al., 2024a) that dispatch tool calls in parallel, leading to signifi-

cant reductions in end-to-end latency. At a more fundamental level, the philosophy of parallelism is also applied to accelerate the core token-generation process itself. Techniques such as speculative and parallel decoding (Wang et al., 2025c; Sun et al., 2024; Chu et al., 2025) are designed to parallelize parts of the autoregressive sequence generation, improving system throughput and responsiveness.

**Augmenting Creativity and Open-ended Generation** For tasks that do not have a single correct answer, such as brainstorming, creative writing, or code design, parallel reasoning is a powerful tool for exploring a wider solution space and generating more diverse outputs. By building out a structure of interconnected ideas, methods like (Yao et al., 2023; Besta et al., 2024) allow the model to explore many creative avenues before converging on a final product. Additionally, other approaches (Wang et al., 2025d) generate a diverse set of initial drafts in parallel and then integrate the best elements from all candidates to create a superior final output. This process allows for the generation of more novel and high-quality content than any single path could produce.

## 7 Challenge and Future Work

### 7.1 Challenge

**Performance Constraint** While current parallel reasoning methods show performance surpassing simple voting baselines, several key challenges remain. (1) These methods are fundamentally constrained by a Pass@k performance upper bound. Although effective at selecting the best solution from a set of generated candidates, they struggle to innovate a novel answer that surpasses the quality of the best available option in the pool. (2) The performance gains from parallel reasoning exhibit a trend of diminishing returns. As the number of parallel samples  $N$  increases, the incremental improvement in accuracy decays. This suggests that current aggregation strategies do not fully utilize all the useful information spread across the candidates, a limitation that may be rooted in the model’s intrinsic reasoning capabilities. (3) Emerging methods that integrate decomposition and aggregation into a single, adaptive model offer improvements in latency and efficiency. However, their performance is often only comparable to that of simple majority voting, and they have not yet been scaled effectively to achieve a significant performance breakthrough.

**Optimization Problem** Current parallel reasoning frameworks also face two primary optimization challenges. (1) Many two-stage or multi-agent architectures are optimized in a disjointed manner, where different components are trained separately. This approach lacks a unified, end-to-end paradigm that would allow feedback from the aggregation stage to supervise and improve the initial generation stage, hindering holistic performance. (2) The aggregation stage itself presents significant hurdles. Firstly, when presented with multiple reasoning paths as context, aggregator models often produce a concise summary instead of a novel, comprehensive reasoning trajectory. Secondly, training an aggregator using outputs from these independently generated paths introduces a complex off-policy optimization problem, which can destabilize large-scale reinforcement learning and make the approach difficult to scale effectively.

### 7.2 Future Work

**Multimodal** The potential of parallel reasoning extends beyond mathematical and coding domains, showing significant promise for boosting performance in multimodal reasoning. This paradigm is applicable not only to text-centric multimodal tasks (Wang et al., 2024c; Pramanick et al., 2025), such as multimodal question answering and named-entity recognition, but also holds great potential for image-based reasoning (Pan et al., 2025a). In this context, a model could generate multiple image variations in parallel and then perform complex reasoning based on the combination of these images, opening new avenues for creative and analytical tasks.

**End to End Optimization & Scale up** Future work in parallel reasoning faces significant challenges in training and optimization. A primary goal is to develop a unified, end-to-end training paradigm for the entire pipeline, from decomposition to aggregation, which would require a fine-grained, clear, and scalable reward signal to enable holistic optimization. Furthermore, large-scale experimentation is needed to validate that parallel reasoning can function as a continuously improving learning process, rather than merely acting as a Pass@k approximator that selects from a fixed-quality pool of answers. Finally, developing a suitable paradigm for stable, on-policy-like reinforcement learning is crucial. Such stability would enable scaling to a much larger number of parallel

samples, reducing reliance on long sequential computations and offering a clear path toward improved inference scalability and lower latency.

## 8 Discussion and Conclusion

### 8.1 Why does parallel reasoning work?

**DFS vs BFS** Sequential reasoning methods are analogous to a depth-first search (DFS). They generate solutions along a single path via iterative refinement, which risks getting trapped in a local minimum. In contrast, parallel reasoning resembles a breadth-first search (BFS), exploring multiple potential solutions simultaneously then aggregating them into a robust answer. This also indicates an interesting phenomenon: why simple repeated sampling methods can surpass the ‘plan-process-summary’ parallel method, which can be constrained by the quality of its initial plan.

**Scaling Components** Following the evolution of parallel reasoning, repeated sampling has become the most common paradigm for boosting test-time computation. A key trend in this area is the evolution of aggregation methods, which have shifted from simple self-consistency (voting) to more sophisticated ranking-based methods (scoring), and ultimately to generative approaches. This evolution marks a significant difference in the computational scaling of the aggregation stage. While voting methods rely on simple, rule-based selection of the most common answer, ranking methods like ORM/PRM require a full model forward pass to score each candidate, similar to reward models in modern reinforcement learning. Generative methods take this even further, re-reasoning over all parallel outputs to construct a final answer, which represents another level of scaling test-time compute. Thus, parallel scaling increases computation not only by decomposing a problem into several tasks but also by generating multiple solutions in parallel. This scaling of the aggregator’s computation also boosts final performance, which can be further improved by scaling the main model or the reward model’s inference time. This progression highlights a central theme: allocating more computation to all components at test-time can lead to significantly better performance.

**Tuning vs In-Context** Sequential reasoning methods, like R1, operate by first rolling out multiple reasoning trajectories. These trajectories are then rewarded based on their final answers. Finally,

paths are used to fine-tune the model’s parameters through a mechanism that reinforces positive outcomes and penalizes negative ones. In contrast, native parallel reasoning treats these rollout trajectories as a unified context by concatenating them. The model then performs its reasoning based on this combined input, effectively transforming a fine-tuning task into an in-context reasoning task. This approach also allows the model to be trained on how to best utilize this parallel information, thereby enhancing its capabilities.

### 8.2 Conclusion

This survey provides a comprehensive overview of parallel reasoning, a pivotal inference paradigm that enhances robustness by shifting from single-path sequential processing to a breadth-first exploration of a model’s capabilities. Through a unified formal framework, we categorize the field and trace its evolution from early consensus-based methods like self-consistency to sophisticated ranking models and dynamic multi-agent collaboration. Despite this progress, the field faces significant challenges, including performance constraints imposed by the initial candidate pool and the lack of unified, end-to-end optimization. Ultimately, parallel reasoning represents a fundamental shift toward more deliberate and scalable AI, and we hope this survey catalyzes further innovation in this promising domain.

## References

- Aman Madaan, Pranjali Aggarwal, Yiming Yang, and Mausam. 2023. [Let’s sample step by step: Adaptive-consistency for efficient reasoning and coding with llms](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12375–12396. Association for Computational Linguistics.
- Zachary Ankner, Rishab Parthasarathy, Aniruddha Nrusimha, Christopher Rinard, Jonathan Ragan-Kelley, and William Brandon. 2024. [Hydra: Sequentially-dependent draft heads for medusa decoding](#). *Preprint*, arXiv:2402.05109.
- Anthropic. 2025a. [How we built our multi-agent research system](#).
- Anthropic. 2025b. [Introducing claude 4](#).
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefer. 2024. [Graph of thoughts: Solving elaborate problems with large](#)



- language models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.
- Bradley C. A. 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](#). *CoRR*, abs/2407.21787.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024a. Medusa: Simple llm inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org.
- Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024b. [Medusa: Simple llm inference acceleration framework with multiple decoding heads](#). *Preprint*, arXiv:2401.10774.
- 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*.
- Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. [Accelerating large language model decoding with speculative sampling](#). *Preprint*, arXiv:2302.01318.
- Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2024a. [Reconcile: Round-table conference improves reasoning via consensus among diverse llms](#). *Preprint*, arXiv:2309.13007.
- Keyu Chen, Zhifeng Shen, Daohai Yu, Haoqian Wu, Wei Wen, Jianfeng He, Ruizhi Qiao, and Xing Sun. 2025a. [Aspd: Unlocking adaptive serial-parallel decoding by exploring intrinsic parallelism in llms](#). *Preprint*, arXiv:2508.08895.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. [Evaluating large language models trained on code](#). *CoRR*, abs/2107.03374.
- Pei Chen, Shuai Zhang, and Boran Han. 2024b. [Comm: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving](#). In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1720–1738. Association for Computational Linguistics.
- Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Ke-fan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2024c. [Universal self-consistency for large language models](#). In *ICML 2024 Workshop on In-Context Learning*.
- Ziyi Chen, Xiaocong Yang, Jiacheng Lin, Chenkai Sun, Kevin Chen-Chuan Chang, and Jie Huang. 2025b. [Cascade speculative drafting for even faster llm inference](#). *Preprint*, arXiv:2312.11462.
- Sukmin Cho, Sangjin Choi, Taeho Hwang, Jeongyeon Seo, Soyeong Jeong, Huije Lee, Hoyun Song, Jong C. Park, and Youngjin Kwon. 2025. [Lossless acceleration of large language models with hierarchical drafting based on temporal locality in speculative decoding](#). *Preprint*, arXiv:2502.05609.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2023. [Palm: Scaling language modeling with pathways](#). *J. Mach. Learn. Res.*, 24:240:1–240:113.
- Yuanlin Chu, Bo Wang, Xiang Liu, Hong Chen, Aiwei Liu, and Xuming Hu. 2025. [SSR: speculative parallel scaling reasoning in test-time](#). *CoRR*, abs/2505.15340.
- 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](#). *CoRR*, abs/2110.14168.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Yifu Ding, Wentao Jiang, Shunyu Liu, Yongcheng Jing, Jinyang Guo, Yingjie Wang, Jing Zhang, Zeng-mao Wang, Ziwei Liu, Bo Du, Xianglong Liu, and

- Dacheng Tao. 2025. [Dynamic parallel tree search for efficient LLM reasoning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 11233–11252. Association for Computational Linguistics.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. [Improving factuality and reasoning in language models through multiagent debate](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, Ahmed Aly, Beidi Chen, and Carole-Jean Wu. 2024. [Layerskip: Enabling early exit inference and self-speculative decoding](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 12622–12642. Association for Computational Linguistics.
- Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. [Complexity-based prompting for multi-step reasoning](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Yichao Fu, Peter Bailis, Ion Stoica, and Hao Zhang. 2024. Break the sequential dependency of llm inference using lookahead decoding. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org.
- Yichao Fu, Xuwei Wang, Yuandong Tian, and Jiawei Zhao. 2025. [Deep think with confidence](#). *Preprint*, arXiv:2508.15260.
- Xiangxiang Gao, Weisheng Xie, Yiwei Xiang, and Feng Ji. 2025a. Falcon: Faster and parallel inference of large language models through enhanced semi-autoregressive drafting and custom-designed decoding tree. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23933–23941.
- Zipeng Gao, Qingrong Xia, Tong Xu, Xinyu Duan, Zhi Zheng, Zhefeng Wang, and Enhong Chen. 2025b. Multi-branch self-drafting for llm inference acceleration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23942–23950.
- In Gim, Seung-Seob Lee, and Lin Zhong. 2024a. [Asynchronous LLM function calling](#). *CoRR*, abs/2412.07017.
- In Gim, Seung seob Lee, and Lin Zhong. 2024b. [Asynchronous llm function calling](#). *Preprint*, arXiv:2412.07017.
- Google. 2025. [Gemini 2.5 deep think](#).
- Lin Gui, Cristina Garbacea, and Victor Veitch. 2024. [Bonbon alignment for large language models and the sweetness of best-of-n sampling](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe 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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8154–8173. Association for Computational Linguistics.
- Chengbo He, Bochao Zou, Xin Li, Jiansheng Chen, Junliang Xing, and Huimin Ma. 2025. [Enhancing LLM reasoning with multi-path collaborative reactive and reflection agents](#). *CoRR*, abs/2501.00430.
- Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D. Lee, and Di He. 2024. [Rest: Retrieval-based speculative decoding](#). *Preprint*, arXiv:2311.08252.
- Chan-Jan Hsu, Davide Buffelli, Jamie McGowan, Feng-Ting Liao, Yi-Chang Chen, Sattar Vakili, and Dashan Shiu. 2025. [Group think: Multiple concurrent reasoning agents collaborating at token level granularity](#). *CoRR*, abs/2505.11107.
- Jenny Y. Huang, Mehul Damani, Yousef El-Kurdi, Ramon Astudillo, and Wei Sun. 2025. [Latency and token-aware test-time compute](#). *Preprint*, arXiv:2509.09864.
- Tian Jin, Ellie Y. Cheng, Zachary Ankner, Nikunj Saunshi, Blake M. Elias, Amir Yazdanbakhsh, Jonathan Ragan-Kelley, Suvinay Subramanian, and Michael Carbin. 2025a. [Learning to keep a promise: Scaling language model decoding parallelism with learned asynchronous decoding](#). *CoRR*, abs/2502.11517.
- Tian Jin, Ellie Y Cheng, Zack Ankner, Nikunj Saunshi, Blake M Elias, Amir Yazdanbakhsh, Jonathan Ragan-Kelley, Suvinay Subramanian, and Michael Carbin. 2025b. [Learning to keep a promise: Scaling language model decoding parallelism with learned asynchronous decoding](#). *arXiv preprint arXiv:2502.11517*.
- Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. 2024a. [An LLM compiler for parallel function calling](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Sehoon Kim, Suhong Moon, Ryan Tabrizi, Nicholas Lee, Michael W. Mahoney, Kurt Keutzer, and Amir Gholami. 2024b. [An llm compiler for parallel function calling](#). *Preprint*, arXiv:2312.04511.
- Dayoon Ko, Jihyuk Kim, Haeju Park, Sohyeon Kim, Dahyun Lee, Yongrae Jo, Gunhee Kim, Moontae

- Lee, and Kyungjae Lee. 2025. [Hybrid deep searcher: Integrating parallel and sequential search reasoning](#). *Preprint*, arXiv:2508.19113.
- Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. [Fast inference from transformers via speculative decoding](#). *Preprint*, arXiv:2211.17192.
- Dawei Li, Zhen Tan, Peijia Qian, Yifan Li, Kumar Satvik Chaudhary, Lijie Hu, and Jiayi Shen. 2025a. [Smoa: Improving multi-agent large language models with sparse mixture-of-agents](#). In *Advances in Knowledge Discovery and Data Mining - 29th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2025, Sydney, NSW, Australia, June 10-13, 2025, Proceedings, Part III*, volume 15872 of *Lecture Notes in Computer Science*, pages 54–65. Springer.
- Shuangtao Li, Shuaihao Dong, Kexin Luan, Xinhan Di, and Chaofan Ding. 2025b. [Enhancing reasoning through process supervision with monte carlo tree search](#). *CoRR*, abs/2501.01478.
- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024. [Eagle-2: Faster inference of language models with dynamic draft trees](#). *Preprint*, arXiv:2406.16858.
- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2025c. [Eagle-3: Scaling up inference acceleration of large language models via training-time test](#). *Preprint*, arXiv:2503.01840.
- Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2025d. [Eagle: Speculative sampling requires rethinking feature uncertainty](#). *Preprint*, arXiv:2401.15077.
- 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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Fangcheng Liu, Yehui Tang, Zhenhua Liu, Yunsheng Ni, Kai Han, and Yunhe Wang. 2024a. [Kangaroo: Lossless self-speculative decoding via double early exiting](#). *Preprint*, arXiv:2404.18911.
- Mingdao Liu, Aohan Zeng, Bowen Wang, Peng Zhang, Jie Tang, and Yuxiao Dong. 2024b. [APAR: llms can do auto-parallel auto-regressive decoding](#). *CoRR*, abs/2401.06761.
- Mingdao Liu, Aohan Zeng, Bowen Wang, Peng Zhang, Jie Tang, and Yuxiao Dong. 2024c. [Apar: Llms can do auto-parallel auto-regressive decoding](#). *Preprint*, arXiv:2401.06761.
- Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2025. [Pairjudge rm: Perform best-of-n sampling with knockout tournament](#). *Preprint*, arXiv:2501.13007.
- Tongxu Luo, Wenyu Du, Jiayi Bi, Stephen Chung, Zhengyang Tang, Hao Yang, Min Zhang, and Benyou Wang. 2025. [Learning from peers in reasoning models](#). *CoRR*, abs/2505.07787.
- Matthew Macfarlane, Minseon Kim, Nebojsa Jojic, Weijia Xu, Lucas Caccia, Xingdi Yuan, Wanru Zhao, Zhengyan Shi, and Alessandro Sordani. 2025. [Instilling parallel reasoning into language models](#). In *2nd AI for Math Workshop @ ICML 2025*.
- Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Zhengxin Zhang, Rae Ying Yee Wong, Alan Zhu, Lijie Yang, Xiaoxiang Shi, Chunshu Shi, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 2024. [Specinfer: Accelerating large language model serving with tree-based speculative inference and verification](#). In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3, ASPLOS ’24*, page 932–949. ACM.
- Giovanni Monea, Armand Joulin, and Edouard Grave. 2023. [Pass: Parallel speculative sampling](#). *Preprint*, arXiv:2311.13581.
- Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Markian Rybchuk, Philip H. S. Torr, Ivan Laptev, Fabio Pizzati, Ronald Clark, and Christian Schröder de Witt. 2024. [MALT: improving reasoning with multi-agent LLM training](#). *CoRR*, abs/2412.01928.
- Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2024. [Skeleton-of-thought: Prompting llms for efficient parallel generation](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.
- OpenAI. 2024. [Learning to reason with llms](#).
- Batu Ozturkler, Nikolay Malkin, Zhen Wang, and Nebojsa Jojic. 2023. [Thinksum: Probabilistic reasoning over sets using large language models](#). *Preprint*, arXiv:2210.01293.
- Jiadong Pan, Zhiyuan Ma, Kaiyan Zhang, Ning Ding, and Bowen Zhou. 2025a. [Self-reflective reinforcement learning for diffusion-based image reasoning generation](#). *Preprint*, arXiv:2505.22407.
- Jiayi Pan, Xiuyu Li, Long Lian, Charlie Snell, Yifei Zhou, Adam Yala, Trevor Darrell, Kurt Keutzer, and Alane Suhr. 2025b. [Learning adaptive parallel reasoning with language models](#). *CoRR*, abs/2504.15466.
- Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2025. [Spiqa: A dataset for multimodal question answering on scientific papers](#). *Preprint*, arXiv:2407.09413.



- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. [Measuring and narrowing the compositionality gap in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.
- Jianing Qi, Xi Ye, Hao Tang, Zhigang Zhu, and Eunsol Choi. 2025. [Learning to reason across parallel samples for LLM reasoning](#). *CoRR*, abs/2506.09014.
- Qwen. 2025. [Qwen3-max-thinking \(heavy\)](#).
- Gleb Rodionov, Roman Garipov, Alina Shutova, George Yakushev, Vage Egiazarian, Anton Sinitsin, Denis Kuznedelev, and Dan Alistarh. 2025a. [Hogwild! inference: Parallel LLM generation via concurrent attention](#). *CoRR*, abs/2504.06261.
- Gleb Rodionov, Roman Garipov, Alina Shutova, George Yakushev, Erik Schultheis, Vage Egiazarian, Anton Sinitsin, Denis Kuznedelev, and Dan Alistarh. 2025b. [Hogwild! inference: Parallel llm generation via concurrent attention](#). *Preprint*, arXiv:2504.06261.
- Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodola. 2023. [Accelerating transformer inference for translation via parallel decoding](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12336–12355, Toronto, Canada. Association for Computational Linguistics.
- Seed. 2025. [Seed1.6-thinking parallel decoding](#).
- Chenyang Shao, Xinyuan Hu, Yutang Lin, and Fengli Xu. 2025. [Division-of-thoughts: Harnessing hybrid language model synergy for efficient on-device agents](#). In *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pages 1822–1833. ACM.
- Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. [Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Benjamin Spector and Chris Re. 2023. [Accelerating llm inference with staged speculative decoding](#). *Preprint*, arXiv:2308.04623.
- Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018a. [Blockwise parallel decoding for deep autoregressive models](#). In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10107–10116.
- Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018b. [Blockwise parallel decoding for deep autoregressive models](#). *Preprint*, arXiv:1811.03115.
- Md Arafat Sultan and Ramón Fernandez Astudillo. 2025. [Confidence-weighted token set cover for early hypothesis pruning in self-consistency](#). *Preprint*, arXiv:2508.03979.
- Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter L. Bartlett, and Andrea Zanette. 2024. [Fast best-of-n decoding via speculative rejection](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. 2023. [Corex: Pushing the boundaries of complex reasoning through multi-model collaboration](#). *CoRR*, abs/2310.00280.
- Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijie Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. [Kimi k2: Open agentic intelligence](#). *Preprint*, arXiv:2507.20534.
- Fengwei Teng, Zhaoyang Yu, Quan Shi, Jiayi Zhang, Chenglin Wu, and Yuyu Luo. 2025. [Atom of thoughts for markov LLM test-time scaling](#). *CoRR*, abs/2502.12018.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. [Llama: Open and efficient foundation language models](#). *CoRR*, abs/2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. [Solving math word problems with process- and outcome-based feedback](#). *CoRR*, abs/2211.14275.
- Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024a. [Soft self-consistency improves language models agents](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024*, pages 287–301. Association for Computational Linguistics.
- Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2025a. [Mixture-of-agents enhances large language model capabilities](#). In *The Thirteenth*



- International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024b. [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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9426–9439. Association for Computational Linguistics.
- Qibin Wang, Pu Zhao, Shaohan Huang, Fangkai Yang, Lu Wang, Furu Wei, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2025b. [Learning to refine: Self-refinement of parallel reasoning in llms](#). *Preprint*, arXiv:2509.00084.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Zhihai Wang, Jie Wang, Jilai Pan, Xilin Xia, Huiling Zhen, Mingxuan Yuan, Jianye Hao, and Feng Wu. 2025c. [Accelerating large language model reasoning via speculative search](#). *CoRR*, abs/2505.02865.
- Ziqi Wang, Boye Niu, Zhongli Li, Linghui Meng, Jing Liu, Zhi Zheng, Tong Xu, Hua Wu, Haifeng Wang, and Enhong Chen. 2025d. [A2r: An asymmetric two-stage reasoning framework for parallel reasoning](#). *Preprint*, arXiv:2509.22044.
- Ziqi Wang, Chen Zhu, Zhi Zheng, Xinhang Li, Tong Xu, Yongyi He, Qi Liu, Ying Yu, and Enhong Chen. 2024c. [Granular entity mapper: Advancing fine-grained multimodal named entity recognition and grounding](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3211–3226, Miami, Florida, USA. Association for Computational Linguistics.
- Ziyao Wang, Muneeza Azmat, Ang Li, Raya Horesh, and Mikhail Yurochkin. 2025e. [Speculate, then collaborate: Fusing knowledge of language models during decoding](#). *CoRR*, abs/2502.08020.
- 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 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Kaiwen Wei, Rui Shan, Dongsheng Zou, Jianzhong Yang, Bi Zhao, Junnan Zhu, and Jiang Zhong. 2025a. [Mirage: Scaling test-time inference with parallel graph-retrieval-augmented reasoning chains](#). *Preprint*, arXiv:2508.18260.
- Zhepei Wei, Wei-Lin Chen, Xinyu Zhu, and Yu Meng. 2024. [Fast and accurate language model decoding via parallel token processing](#).
- Zhepei Wei, Wei-Lin Chen, Xinyu Zhu, and Yu Meng. 2025b. [Fast and accurate language model decoding via parallel token processing](#).
- Hao Wen, Yifan Su, Feifei Zhang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. 2025. [Parathinker: Native parallel thinking as a new paradigm to scale llm test-time compute](#). *Preprint*, arXiv:2509.04475.
- xAI. 2025. [Introducing grok 4](#).
- Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. 2023a. [Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3909–3925, Singapore. Association for Computational Linguistics.
- Heming Xia, Tao Ge, Peiyi Wang, Si-Qing Chen, Furu Wei, and Zhifang Sui. 2023b. [Speculative decoding: Exploiting speculative execution for accelerating seq2seq generation](#). *Preprint*, arXiv:2203.16487.
- Heming Xia, Yongqi Li, Jun Zhang, Cunxiao Du, and Wenjie Li. 2025. [Swift: On-the-fly self-speculative decoding for llm inference acceleration](#). *Preprint*, arXiv:2410.06916.
- Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. 2023. [Self-evaluation guided beam search for reasoning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Zenan Xu, Zexuan Qiu, Guanhua Huang, Kun Li, Siheng Li, Chenchen Zhang, Kejiao Li, Qi Yi, Yuhao Jiang, Bo Zhou, Fengzong Lian, and Zhanhui Kang. 2025. [Adaptive termination for multi-round parallel reasoning: An universal semantic entropy-guided framework](#). *Preprint*, arXiv:2507.06829.
- Sen Yang, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2024. [Multi-candidate speculative decoding](#). *Preprint*, arXiv:2401.06706.
- Xinyu Yang, Yuwei An, Hongyi Liu, Tianqi Chen, and Beidi Chen. 2025a. [Multiverse: Your language models secretly decide how to parallelize and merge generation](#). *Preprint*, arXiv:2506.09991.
- Xinyu Yang, Yuwei An, Hongyi Liu, Tianqi Chen, and Beidi Chen. 2025b. [Multiverse: Your language models secretly decide how to parallelize and merge generation](#). *Preprint*, arXiv:2506.09991.
- 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 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2024a. [Draft & verify: Lossless large language model acceleration via self-speculative decoding](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 11263–11282. Association for Computational Linguistics.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025a. [Generative verifiers: Reward modeling as next-token prediction](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yifan Zhang, Jingqin Yang, Yang Yuan, and Andrew C. Yao. 2025b. [Cumulative reasoning with large language models](#). *Trans. Mach. Learn. Res.*, 2025.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arik. 2024b. [Chain of agents: Large language models collaborating on long-context tasks](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Shu Zhao, Tan Yu, Anbang Xu, Japinder Singh, Aaditya Shukla, and Rama Akkiraju. 2025. [Parallelsearch: Train your llms to decompose query and search sub-queries in parallel with reinforcement learning](#). *Preprint*, arXiv:2508.09303.

Yao Zhao, Zhitian Xie, Chen Liang, Chenyi Zhuang, and Jinjie Gu. 2024. [Lookahead: An inference acceleration framework for large language model with lossless generation accuracy](#). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 6344–6355. ACM.

Tong Zheng, Hongming Zhang, Wenhao Yu, Xiaoyang Wang, Runpeng Dai, Rui Liu, Huiwen Bao, Chengsong Huang, Heng Huang, and Dong Yu. 2025. [Parallel-r1: Towards parallel thinking via reinforcement learning](#). *Preprint*, arXiv:2509.07980.